

## RANCANG BANGUN SISTEM KENDALI POSISI KETINGGIAN PADA *TOWERCOPTER* BERBASIS *PROPORTIONAL INTEGRAL* MENGGUNAKAN TUNING *ARTIFICIAL NEURAL NETWORK*

**Rifqi Nabila Zufar**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : rifqizufar@mhs.unesa.ac.id

**Puput Wanarti Rusimamto**

Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : puputwanarti@unesa.ac.id

### Abstrak

*Drone Singlecopter* mengalami perkembangan yang pesat pada beberapa tahun ini. *Drone* ini memiliki kelebihan dari segi massa *Drone* yang ringan sehingga energi yang dibutuhkan lebih sedikit dan juga efisien. Namun, *Drone* ini memiliki kelemahan untuk menstabilkan posisi drone pada saat lepas landas ataupun *hovering*. *Plant* yang digunakan pada penelitian ini bernama *Towercopter*. Prinsip kerja dari *plant* ini hampir sama dengan prinsip kerja dari *Vertical Take Of Landing (VTOL)* pada *Singlecopter*. *Plant* ini kemudian di ujicoba dengan mengimplementasikan sistem kendali *Proportional Integral* menggunakan tuning *Artificial Neural Network (PI-ANN)*. Hasil penelitian menunjukkan bahwa nilai  $\tau = 7.85$  detik, *Rise Time* = 13.41 detik, *Settling Time* = 52.28 detik, Maksimal *Overshoot* = 22.3% dan *Error Steady State* = 0.43%. Selanjutnya, *plant* diuji menggunakan beban. Hasil respon menunjukkan *plant* mengalami penurunan posisi sejauh 4 cm yang kemudian bergerak naik pada detik ke-173 dan respon mencapai setpoint pada detik ke- 213. Terakhir, *plant* diuji coba dengan setpoint yang diubah. Respon yang dihasilkan cenderung menyerupai respon output yang diinginkan, namun pada percobaan menggunakan setpoint 17 dan 19 cm, waktu untuk mencapai keadaan *Steady State* tidak sama dengan selisih  $\pm 20$  detik.

**Kata Kunci:** *VTOL, Towercopter, PI-ANN Controller.*

### Abstract

*Singlecopter drones* have experienced rapid development in recent years. This drone has advantages in terms of the lightweight drone mass so that the less energy needed and efficient. However, this Drone also has a disadvantage to stabilize the drone position during takeoff or hovering. The towercopter is used as a plant in this research. The working principle of this prototype is almost the same as the working principle of *Vertical Take of Landing (VTOL)* in a *Singlecopter*. This prototype is then tested by implementing an *Integral Proportional control system using Artificial Neural Network tuning (PI-ANN)*. The result shows that the value of *Rise Time* = 13.41 seconds, *Settling Time* = 52.28 seconds, *Maximum Overshoot* = 22.3% and *Steady State Error* = 0.43%. Furthermore, the plant is tested using a load. The result shows that plant response decrease in position by 4 cm which then moved up in the 173 seconds and the response reach the setpoint at the 213 seconds. Finally, the plant is tested with a modified setpoint. The response result tends to resemble desired response output, but in the experiment using a setpoint of 17 and 19 cm, the time to reach the state of *Steady-State* are not the same with difference of 20 seconds.

**Keywords:** *VTOL, Towercopter, PI-ANN Controller.*

### PENDAHULUAN

Salah satu jenis dari UAV adalah *Singlecopter Drone* atau *Drone* dengan 1 buah motor penggerak. *Drone* jenis ini memiliki kelebihan dari segi konsumsi daya yang minimal dan waktu terbang yang lebih lama. Hal ini disebabkan karena hanya ada 1 jumlah motor yang dikendalikan dan massa *Drone* yang ringan. Namun, kelemahannya adalah *Drone* sulit untuk dikendalikan karena sifatnya yang sangat rentan terhadap gangguan

angin serta cenderung tidak stabil saat lepas landas ataupun *hovering* (Moon, 2018).

Pada tahun 2010, Marta Salas García melakukan penelitian dengan judul “*Control Para La Estabilización En Suspensión De Un Cuatrimotor*”. Penelitian ini menghasilkan *prototype* dari *Singlecopter* yang diberi nama *Towercopter*. *Towercopter* sendiri merupakan sebuah alat dengan 1 buah motor brushless dan 1 buah propeller yang dipasang diantara 2 tiang penyangga.

*Towercopter* ini kemudian dijadikan *plant* untuk penelitian selanjutnya.

Pada tahun 2012, David Sánchez-Benítez, dkk melakukan penelitian dengan judul "Vertical Rotor for The Implementation of Control Laws". Pada penelitian ini, *Towercopter* dikembangkan dengan memberikan kontroler PID untuk sistem kendali ketinggiannya. Hasil yang diperoleh adalah untuk kontroler PID, nilai parameter  $K_p = 0.9471$ ,  $K_i = 0.8094$ ,  $K_d = 0.1001$  yang didapat dari metode *trial and error* dan menghasilkan respon sistem dapat mengikuti setpoint dengan rata-rata maksimal overshoot <15%.

Pada tahun 2016, Pavas Cristian, dkk melakukan penelitian dengan judul "Controlador De Posici'On Lineal Towercopter". *Towercopter* dikembangkan dengan mengubah nilai parameter PID-nya menjadi  $K_p = 0.814$ ,  $K_i = 1.935$  dan  $K_d = 0.0000569375$ . Hasil respon yang didapat adalah *plant* dapat mengikuti setpoint dengan rata-rata maksimal overshoot <12%.

Berdasarkan penelitian yang telah dilakukan terdahulu, *plant Towercopter* dapat mengikuti setpoint namun overshoot yang dihasilkan masih tergolong tinggi serta metode yang digunakan untuk mendapatkan parameter PID masih menggunakan metode *trial and error*. Metode ini tentunya memiliki kekurangan dari segi waktu yang dibutuhkan untuk mendapatkan parameter PID cenderung lama karena harus menguji coba nilai minimal hingga nilai maksimal agar respon yang dihasilkan menjadi baik. Selain itu, kontroler ini juga kurang efektif jika diterapkan pada variabel *plant* yang dapat berubah sewaktu-waktu. Pada penelitian ini, peneliti menggunakan kontroler *Proportional Integral* dengan tuning *Artificial Neural Network* (PI-ANN). Tujuan penambahan ANN ini adalah untuk mengatasi perubahan dinamika proses *plant Towercopter* sehingga respon sistem menjadi lebih stabil dengan overshoot yang lebih kecil.

**KAJIAN PUSTAKA**

**Towercopter**

*Towercopter* merupakan suatu prototype alat digunakan dalam mengembangkan sistem kendali pada quadcopter. *Towercopter* terdiri atas 1 buah BLDC beserta propelernya yang dipasang diantara 2 tiang. Prinsip kerja dari alat ini adalah motor akan bekerja meluncur sejauh sumbu Y. Gaya dorong yang dihasilkan berasal dari kecepatan putar propelernya. Semakin cepat putaran propeller, semakin besar pula daya dorong yang dihasilkan. Untuk mempertahankan tinggi motor maka perlu adanya sebuah metode kontrol yang baik (Sánchez-Benítez, 2012).

**Pemodelan Matematis Sistem**

Gaya thrust adalah gaya yang mengakibatkan towercopter bergerak naik atau turun searah sumbu Y. Gaya dorong terjadi jika  $F_{thrust} > F_{grafitasi}$  maka kerangka akan bergerak keatas.  $F_{gesek}$  berpengaruh pada  $F_{thrust}$  ataupun  $F_{grafitasi}$ . Pada saat kerangka terangkat terangkat ke atas  $F_{gesek}$  akan bernilai (-) terhadap  $F_{thrust}$ , sebaliknya jika kerangka menuju kebawah  $F_{gesek}$  bernilai (+) terhadap  $F_{thrust}$  (Sanchez-Benitez, 2012). Sehingga berdasarkan Newton's second law didapatkan persamaan sebagai berikut,

$$M \vec{a} = -M\vec{g} + \vec{F}_{gesek} + \vec{F}_{thrust} \tag{1}$$

Keterangan:

- $M$  : Massa (gr)
- $\vec{a}$  : Vector Percepatan
- $\vec{g}$  : Vector Gravitasi
- $\vec{F}_{gesek}$  : Gaya Gesek (N)
- $\vec{F}_{thrust}$  : Gaya Angkat (N)

Massa kabel yang terdapat pada kerangka juga diperhitungkan karena dapat menambah masa dari kerangka. Sehingga didapatkan,

$$M = M_{fr} + M_c = (M_{fr} + \rho_c)X \tag{2}$$

Keterangan:

- $M_{fr}$  : Massa Frame (gr)
- $M_c$  : Massa Kabel (gr)
- $\rho_c$  : Massa Jenis Kabel (gr/cm<sup>3</sup>)
- $X$  : Panjang Kabel (m)

$F_{gesek}$  *Towercopter* tergantung pada kecepatan. Semakin cepat frame bergerak pada sumbu vertikal, semakin sedikit resistensi yang yang dihadapi. Selain itu, karena gaya gravitasi diambil sebagai negatif dan dorong sebagai kekuatan positif, tanda kecepatan termasuk dalam ekspresi gesekan. Kemudian,  $F_{gesek}$  dapat dimodelkan sebagai berikut:

$$F_{gesek} = - \frac{\dot{x}\mu_s}{(1+\mu_s)} \tag{3}$$

Keterangan:

- $\dot{x}$  : Turunan Pertama dari  $x$
- $\mu_s$  : Koeffisien Gesek

Gaya yang dihasilkan oleh propeller sebanding dengan sinyal kontrol. Untuk menyederhanakan model, ini dinyatakan sebagai:

$$F_{thrust} = Ku \quad (4)$$

Keterangan:

$K$  : Konstanta Sistem

$u$  : Sinyal Kontrol

Berdasarkan persamaan 2 sampai dengan persamaan 4, dapat diringkas dan diperoleh persamaan dinamis dari *Towercopter* sebagai berikut:

$$M\ddot{x} = -Mg - \frac{\dot{x}\mu_s}{(1+\mu_s)} + F_{thrust} \quad (5)$$

Keterangan:

$\ddot{x}$  : Turunan Kedua dari  $x$

Persamaan 6 dapat dapat disederhanakan sehingga:

$$\ddot{x} = -\frac{g(M_{fr}+\rho_c)}{M_{fr}+\rho_c}x - \frac{\mu_s}{(1+\mu_s)}\dot{x} + \frac{K}{M_{fr}+\rho_c}u \quad (6)$$

Persamaan 6 kemudian dapat ditulis kedalam 2 dimensional *State Space*:

$$\dot{x}_1 = x_2 \quad (7)$$

$$\dot{x}_2 = -\frac{g(M_{fr}+\rho_c)}{M_{fr}+\rho_c}x_1 - \frac{\mu_s}{(1+\mu_s)}x_2 + \frac{K}{M_{fr}+\rho_c}u \quad (8)$$

Keterangan:

$\dot{x}_n$  : Dimensional *State Space*

$n$  : Nilai ke- $n$

### Linearisasi

Perkembangan sistem nonlinear dapat didekati dengan sistem linear dengan teori linearisasi. Teori linearisasi digunakan untuk mengatasi permasalahan ketidaklinearan sistem, sehingga sistem yang nonlinear dapat didekati dengan sistem linear. Teknik yang digunakan untuk linearisasi adalah menggunakan Matriks *Jacobi* (Wicaksono, 2016).

$$y = (c) \quad (9)$$

Keterangan:

$y$  : Keluaran Sistem

$c$  : Masukan Sistem

Sistem nonlinear merupakan permisalan dari nilai ( $x$ ), sehingga:

$$f(c) = \dot{x} = \left. \frac{df(c)}{dx} \right|_{c=x} c \quad (10)$$

Keterangan:

$f(c)$  : Keluaran Sistem

$df(c)$  : Turunan Fungsi

$dx$  : Differensial  $x$

Sistem nonlinear loop terbuka dengan jumlah state sama. Persamaan 10 menunjukkan bahwa sistem nonlinear dapat didekati menjadi sistem linear di sekitar titik ekuilibriumnya dengan  $A$  merupakan Matriks *Jacobi* dari  $f(x)$ .

$$\dot{x} = Ax + Bu \quad (11)$$

Keterangan:

$A$  : Matriks *Jacobi*  $A$

$B$  : Matriks *Jacobi*  $B$

$x$  : Jumlah *State Space*

### Analisis Karakteristik Sistem Orde Dua

Sistem orde dua mempunyai fungsi alih dengan pangkat  $s$  tertinggi dua. Respon sistem ini didapatkan dengan mengamati respon sistem terhadap sinyal uji step.

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (12)$$

Keterangan:

$K$  : Gain Overall

$\omega_n$  : Frekuensi Alami

$\xi$  : Rasio Redaman

$s$  : Penunjuk persamaan Transformasi *Laplace*

Karakteristik respon transien sistem orde dua terdiri dari:

*Rise Time* : Waktu untuk naik dari 5% ke 95% respon keadaan waktu tunak (s).

Maksimal *Overshoot* : Nilai puncak output kurva respon diukur dari satuan (%).

*Settling Time* : Waktu untuk mencapai setpoint (s).

*Error Steady-State* : *Error* pada waktu tunak (%).

### Kontroler PI (*Proportional - Integral*)

Keuntungan gabungan dari kontroler PI adalah untuk mempercepat reaksi sebuah sistem. Kontroler dengan kontrol proporsional ditambahkan dengan kontroler integral (Ogata, 1985). Aksi kontrol Proporsional Integral tersebut didefinisikan sebagai:

$$G(s) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (13)$$

$$G(s) = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (14)$$

Keterangan:

- $G(s)$  : Fungsi Kontrol
- $Kp$  : Konstanta *Proportional*
- $Ti$  : Waktu *Integral*
- $t$  : Waktu
- $dt$  : Turunan Waktu
- $\int_0^t e(t)dt$  : *Integral* sinyal *error* terhadap waktu

**Metode Tuning Artificial Neural Network**

*Artificial Neural Network* (ANN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. ANN merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi dari dalam maupun dari luar yang ada pada jaringan tersebut. ANN dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data.

Salah satu model Jaringan Syaraf Tiruan adalah *Backpropagation*. *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa walaupun tidak sama dengan pola yang dipakai selama pelatihan.

Fungsi aktivasi dalam *Backpropagation* yang dipakai harus memenuhi beberapa syarat. Salah satu fungsi aktivasi adalah fungsi *Sigmoid Biner* (Wicaksono, 2016).

$$f(si) = \frac{1}{1+e^{-z}} \tag{15}$$

Keterangan:

- $f(si)$  : Fungsi *Sigmoid*
- $e$  : Konstanta *Euler*
- $z$  : Nilai Aktifasi

**Raspberry Pi model B**

*Raspberry Pi* adalah sebuah komputer papan tunggal (single-board computer) atau SBC berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (System-on-a-chip) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk booting dan penyimpanan jangka panjang. (Agfianto, 2012).

**Sensor Ultrasonic HC-SR04**

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah di

atas gelombang suara dari 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima (Hani, 2010).

**METODE PENELITIAN**

**Pendekatan Penelitian**

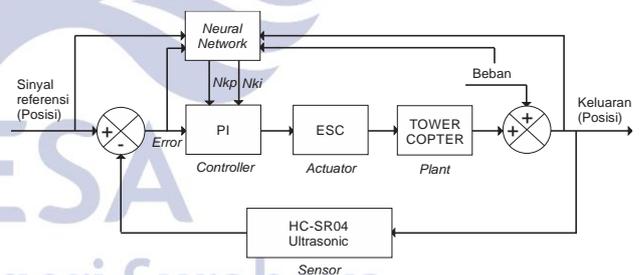
Pendekatan penelitian pada penelitian ini adalah penelitian kuantitatif. Pada penelitian ini menggunakan MATLAB 2015 sebagai pemroses data dan untuk mencari fungsi alih dari sistem kerja pada *TowerCopter*. Desain kontroler yang digunakan adalah kontroler *Proporsional-Integral* dengan tuning *Artificial Neural Network* (PI-ANN). Model kontroler ini kemudian akan diterapkan pada Instrumen kontrol menggunakan mini PC Raspberry PI 3 model B dan *HC-SR04* sebagai sensor ketinggian.

**Instrumen Pengumpulan Data**

Instrumen pengumpulan data yang digunakan dalam penelitian ini adalah berupa komputer (PC) ataupun laptop serta perangkat lunak *MATLAB* 2015 yang digunakan untuk menganalisis data keluaran sistem. Pengumpulan data kuantitatif diperoleh dari *MATLAB* 2015 yang digunakan untuk menampilkan respon simulasi sistem pada kontrol ketinggian *Towercopter*.

**Desain Sistem**

Metode kontrol yang digunakan menggunakan PI dengan metode tuning berbasis *Artificial Neural Network*. Berikut desain diagram blok sistem kendali ketinggian *TowerCopter* yang ditunjukkan pada Gambar 1.



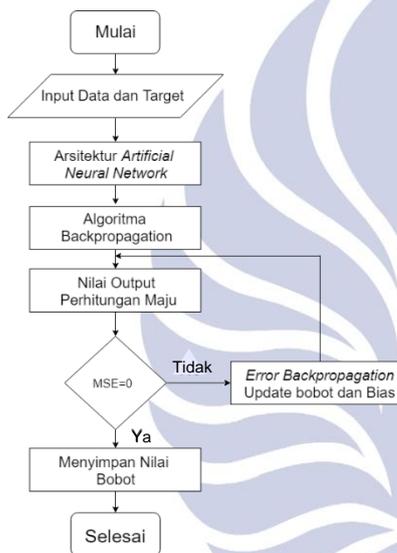
Gambar 1. Diagram Blok Sistem

Nilai dari input akan diproses pada Raspberry Pi yang didalamnya sudah terdapat Kontroler *Proporsional-Integral*. Hasil dari keluaran kontroler berupa tegangan akan masuk ke ESC sehingga ESC akan mengatur kecepatan putar pada BLDC. Sensor *Ultrasonic* akan bekerja untuk mengukur nilai dari posisi/ketiggian *Towercopter* kemudian menjadikan nilai pembacaan sebagai nilai umpan balik. Ketika sistem memiliki selisih  $\neq 0$ , maka blok *Artificial Neural Network* akan merevisi nilai error hingga mendekati 0 dan menghasilkan  $NKp$  dan  $NKi$ . Nilai tersebut akan menjadi nilai  $Kp$  dan  $Ki$  yang baru pada parameter kontroler PI.

### Perancangan Kontroler Sistem

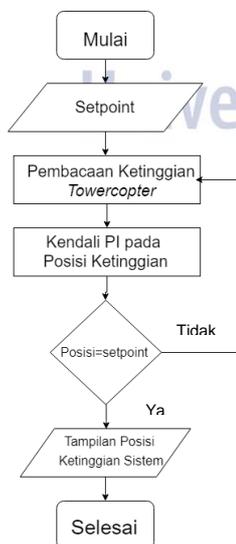
Perancangan kontroler yang pertama dilakukan adalah perancangan kontroler PI secara analitik dari parameter fungsi alih yang diperoleh dari hasil identifikasi. Dari perhitungan secara analitik tersebut, akan diperoleh parameter  $K_p$  dan  $K_i$ . Hasil perancangan kontroler PI akan digunakan sebagai target untuk proses *Learning ANN* sampai mendapatkan bobot terbaik sehingga setelah proses *Mapping ANN* mampu melakukan penalaan terhadap kontroler PI yang memiliki keluaran berupa sinyal kontrol untuk mengatur posisi ketinggian *Towercopter*.

Berikut merupakan desain algoritma pada Proses *Learning ANN* yang ditunjukkan pada Gambar 2.



Gambar 2. Proses *Learning Artificial Neural Network*

Berikut merupakan desain algoritma pada Proses *Mapping ANN* yang ditunjukkan pada Gambar 3.



Gambar 3. Proses *Mapping Artificial Neural Network*

### HASIL DAN PEMBAHASAN

Bab ini akan membahas tentang Hasil Perancangan *Plant*, Perancangan Kontroler PI-ANN dan Hasil Respon Dinamik Sistem.

#### Hasil Perancangan *Plant*

Untuk pemodelan pada *Towercopter* harus diketahui spesifikasi dari *hardware*. Berikut adalah Gambar 4. Rancang Bangun *plant Towercopter* dan Tabel 1. Spesifikasi *plant Towercopter*.



Gambar 4. Rancang Bangun *Plant Towercopter*

Tabel 1. Spesifikasi *Plant Towercopter*

No	Nama	Spesifikasi
1	Tinggi Jalur	$\pm 30$ cm
2	Motor Brushless Emax cf2812	$\pm 1500$ Kv
3	Propeller	6x 5 inc
4	Sensor Jarak	HC-Sr04
5	LCD	16 x 2
6	Instrumen Kontrol	Raspberry Pi 3B
7	Driver Motor	ESC 30A
8	Power Supply	12 V, 10A

#### Pembuatan Kontroler Sistem

Pada Penelitian ini, kontroler yang digunakan untuk mengendalikan posisi ketinggian *Towercopter* menggunakan Kontroler PI-ANN.

#### 1. Pemodelan Matematis *Plant Towercopter*

*Plant* yang sudah dibuat, kemudian diukur untuk mengetahui konstanta-konstanta yang belum diketahui yakni, Massa Frame ( $M_{fr} + \rho_c$ ) sebesar  $\pm 190$ gr, Koefisien Gesek sebesar 0.12, dan *Trust Force Koefisien* ( $K$ ) sebesar 0.8113. Konstanta yang telah didapat disubstitusikan pada model *State-Space* sebagai berikut,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g(M_{fr} + \rho_c)}{M_{fr} + \rho_c} & -\frac{\mu_s}{M_{fr} + \rho_c} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K}{M_{fr} + \rho_c} \end{bmatrix} u \quad (16)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u \quad (17)$$

Sehingga didapatkan Matriks A, B, C, D sebagai berikut,

$$A = \begin{bmatrix} 0 & 1 \\ -1.2791 & -2.2619 \end{bmatrix} \quad (18)$$

$$B = \begin{bmatrix} 0 \\ 0.8113 \end{bmatrix} \quad (19)$$

$$C = [1 \ 0] \quad (20)$$

$$D = 0 \quad (21)$$

Setelah didapatkan Matriks A, B, C, D, maka selanjutnya mengubahnya menjadi persamaan *Transfer Function* menggunakan persamaan 22.

$$G(s) = C(sI - A)^{-1}B + D \quad (22)$$

Keterangan:

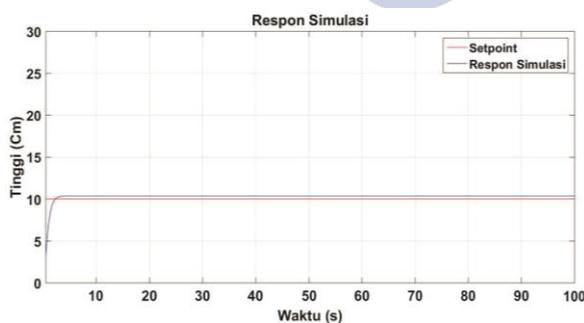
$G(s)$  : Fungsi alih pemodelan sistem

$I$  : Matriks Identitas

Sehingga didapatkan *Transfer Function* sebagai berikut,

$$G(s) = \frac{1.0373}{0.781 s^2 + 1.768s + 1}$$

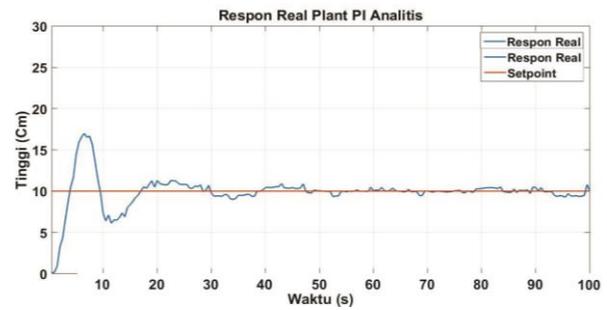
Dari model *Transfer Function* tersebut, kemudian disimulasikan pada software MATLAB untuk diketahui respon sistem tanpa kontroler.



Gambar 5. Respon Simulasi Tanpa Kontroler

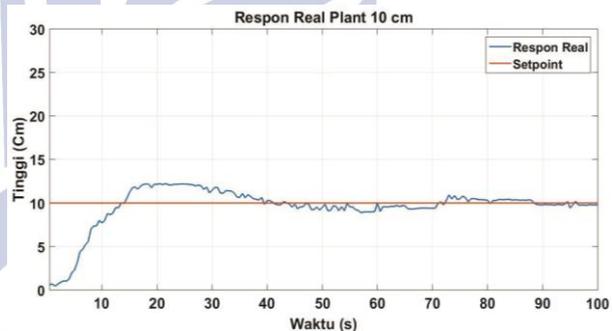
## 2. Pembuatan PI Kontroler Sebagai Target Output

*Transfer Function* yang telah didapat dari penurunan model matematis *plant*, kemudian dianalisa menggunakan pendekatan sistem orde 2 pada persamaan 12. Sehingga didapatkan  $K_p = 1.644$  dan  $K_i = 0.929$ . Berikut merupakan respon *real plant Towercopter* yang ditunjukkan pada Gambar 6.



Gambar 6. Respon *Real Plant* Hasil Pendekatan Orde 2

Hasil respon *real plant* menunjukkan bahwa respon dapat mengikuti nilai setpoint yang diberikan. Dengan *Error Steady-State* sebesar 0.36 % dan respon mulai bergerak naik pada 4.02 detik. Respon dinilai kurang baik karena masih mempunyai Maksimal *Overshoot* sebesar 69.9%. Oleh karena itu, parameter  $K_p$  dan  $K_i$  perlu untuk disesuaikan ulang dengan cara mengurangi nilai  $K_p$  dan  $K_i$  dengan tujuan untuk memperoleh respon sistem dengan Maksimal *Overshoot* <25%. Sehingga didapatkan parameter  $K_p$  dan  $K_i$  akhir adalah  $K_p = 0.16$  dan  $K_i = 0.51$ . Berikut merupakan respon *real plant Towercopter* menggunakan  $K_p$  dan  $K_i$  yang baru yang ditunjukkan pada Gambar 7.

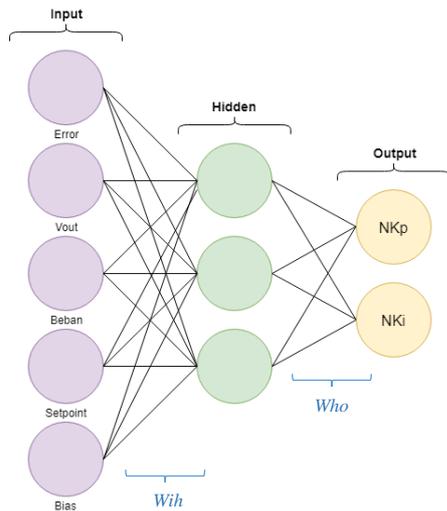


Gambar 7. Respon *Real Plant* Menggunakan  $K_p$  dan  $K_i$  Hasil Penyesuaian

Pada Gambar 7 didapatkan hasil respon *real plant* dengan *Error Steady-State* sebesar 0.43%, respon mulai bergerak naik pada 15.6 detik dengan Maksimal *Overshoot* sebesar 22.13 %. Respon yang dihasilkan sudah sesuai yang diinginkan sehingga parameter  $K_p$  dan  $K_i$  yang baru ini, kemudian dijadikan Target Output untuk pembuatan kontroler PI-ANN.

## 3. Pembuatan Kontroler PI-ANN

Untuk membuat Kontroler PI-ANN, maka terlebih dahulu merancang struktur *Artificial Neural Network* seperti pada Gambar 8.



Gambar 8. Struktur ANN BackPropagation

Proses *Learning* bertujuan untuk melatih *plant Towercopter* agar dapat beradaptasi ketika terjadi perubahan variable. Pada proses ini, input dan output ditentukan melalui *Sampling Data* sebanyak 11 kali percobaan dengan setpoint berbeda yakni, {5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}. Hasil dari *Sampling Data* percobaan dapat dilihat pada Tabel 2.

Tabel 2. *Sampling Data* Hasil Percobaan

Setpoint	Yout	Error	Beban	Bias	Nkp	Nki
5	6.25	1.25	1	1	0.85	0.23
6	7.50	1.50	1	1	0.74	0.31
7	8.75	1.75	1	1	0.66	0.38
8	10.00	2.00	1	1	0.56	0.43
9	11.25	2.25	1	1	0.26	0.49
10	12.50	2.50	1	1	0.16	0.51
11	13.75	2.75	1	1	0.10	0.55
12	15.00	3.00	1	1	0.01	0.55
13	16.25	3.25	1	1	0.005	0.61
14	17.50	3.50	1	1	0.001	0.63
15	18.75	3.75	1	1	0.0005	0.50

Keterangan:

- Setpoint* : Nilai yang Ingin Dicapai
- Yout* : Nilai Keluaran Respon
- Error* : Selisih *Yout* dan *Setpoint*
- Beban* : Beban Inputan
- Bias* : Bias *Neuron*
- Nkp* : Parameter Kp Baru
- Nki* : Parameter Ki Baru

Setelah proses *Learning* selesai, maka didapatkan bobot untuk setiap unit. Inisialisasi bobot Input menuju Hidden layer (*wih*) dapat ditunjukkan pada Tabel 3.

Tabel 3. Bobot Input Menuju Hidden Layer

	<i>Wi1</i>	<i>Wi2</i>	<i>Wi3</i>
<b><i>Wh1</i></b>	-2.752	0.284	2.122
<b><i>Wh2</i></b>	-3.862	-2.241	2.040
<b><i>Wh3</i></b>	-2.504	1.136	4.237
<b><i>Wh4</i></b>	1.651	1.812	-2.877
<b><i>Wh5</i></b>	1.662	0.740	-1.936

Inisialisasi bobot Hidden menuju Output Layer (*who*) dapat ditunjukkan pada Tabel 4.

Tabel 4. Bobot Hidden Menuju Output Layer

	<i>Wo1</i>	<i>Wo2</i>
<b><i>Wh1</i></b>	3.746	-2.963
<b><i>Wh2</i></b>	-0.834	1.365
<b><i>Wh3</i></b>	-6.042	-1.083

Bobot-bobot yang telah didapatkan, memiliki nilai MSE (*Mean Square Error*) sebesar 0.0009468.

### Hasil Respon Dinamik

Setelah perancangan kontroler PI-ANN telah selesai, langkah selanjutnya menguji-coba respon dinamik dari *plant Towercopter*. *Plant* akan diuji-coba tanpa menggunakan beban, menggunakan beban dan menggunakan setpoint yang berbeda.

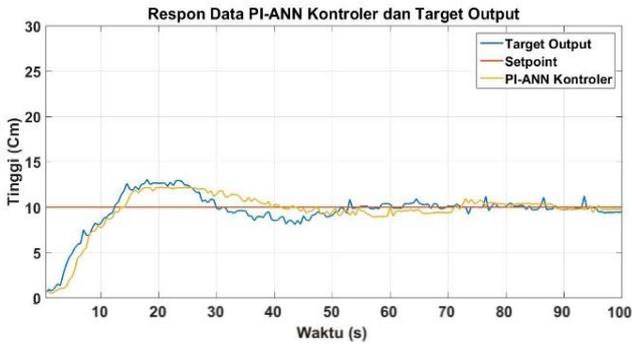
#### 1. Pengujian Tanpa Beban

Berikut merupakan respon real *plant* menggunakan kontroler PI-ANN ditunjukkan pada Gambar 9.



Gambar 9. Respon *Real Plant* menggunakan Kontroler PI-ANN

Hasil respon menunjukkan bahwa, waktu *plant* untuk mencapai keadaan tunak pada 7.85 detik dengan waktu naik sebesar 13.41 detik dan waktu untuk mencapai *Steady State* adalah 52.28 detik. Berdasarkan respon pada Gambar 9 dapat diamati bahwa *Towercopter* mampu mengikuti perubahan setpoint yang diberikan dengan *Error Steady state* sebesar 0.41 % dan maksimal *overshoot* sebesar 22.3%.



Gambar 10. Perbandingan PI-ANN Kontroler dengan Target Output

Jika dibandingkan dengan hasil respon Target Output pada Gambar 7, hasil respon PI-ANN kontroler memiliki perbedaan pada nilai *RiseTime*, *Settling Time*, Maksimal *Overshoot*, dan *Error Steady-State* masing-masing 2.21 detik, 11 detik, 0.27%, dan 0.02%.

### 2. Pengujian Menggunakan Beban

Pengujian dilakukan dengan memasang beban seberat 2.32 gr pada *plant Towercopter*.

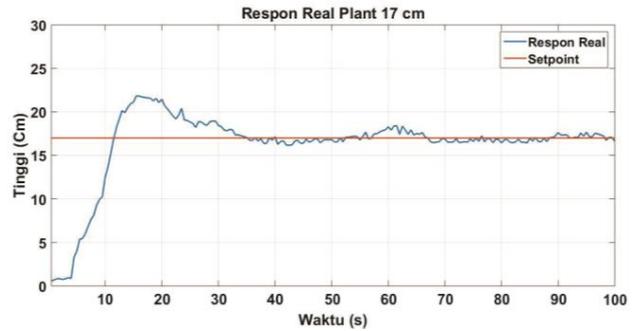


Gambar 11. Respon Towercopter Diberi Beban

Pada Gambar 11 dapat dilihat bahwa, ketika sistem sudah mencapai *Steady-State*, pada detik ke-140 pemberian beban dilakukan. *Plant* mengalami penurunan posisi sebesar 4 cm dan kemudian bergerak naik pada detik ke-173. Respon mencapai setpoint pada detik ke-213 namun masih mengalami Maksimal *Overshoot* sebesar 15%.

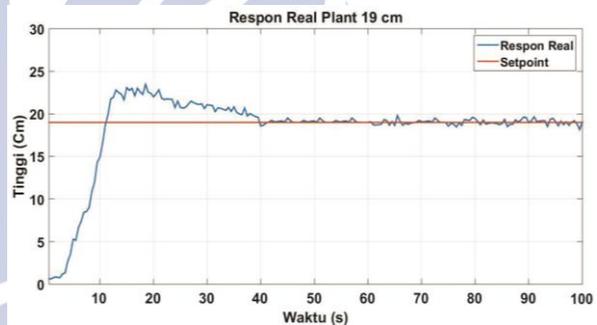
### 3. Pengujian Dengan Setpoint Berbeda

Pengujian ini dilakukan bertujuan agar respon sistem menyerupai ataupun sama dengan respon Target Output pada proses *Learning* ketika terjadi perubahan variabel. Setpoint yang diuji adalah 17cm, 19cm dan 21cm.



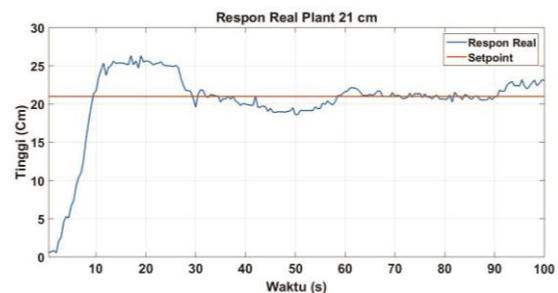
Gambar 12. Respon *Real Plant* menggunakan Kontroler PI-ANN dengan Setpoint 17cm

Hasil respon *real Towercopter* menunjukkan bahwa, waktu *plant* untuk mencapai keadaan tunak yaitu sebesar pada 9.61 detik dengan waktu naik sebesar 13.22 detik dan waktu untuk mencapai *Steady-State* adalah 35.61 detik. Berdasarkan respon pada Gambar 12 dapat diamati bahwa *Towercopter* mampu mengikuti perubahan setpoint yang diberikan dengan *Error Steady-State* sebesar 0.96 % dan maksimal *Overshoot* sebesar 23.73%.



Gambar 13. Respon *Real Plant* menggunakan Kontroler PI-ANN dengan Setpoint 19cm

Hasil respon *real Towercopter* menunjukkan bahwa, waktu *plant* untuk mencapai keadaan tunak yaitu pada 9.12 detik dengan waktu naik sebesar 13.46 detik dan waktu untuk mencapai *Steady-State* adalah 39.92 detik. Berdasarkan respon pada Gambar 13 dapat diamati bahwa *Towercopter* mampu mengikuti perubahan setpoint yang diberikan dengan *Error Steady-State* sebesar 0.20 % dan Maksimal *Overshoot* sebesar 23.52%.



Gambar 14. Respon *Real Plant* menggunakan Kontroler PI-ANN dengan Setpoint 21cm

Hasil respon *real Towercopter* menunjukkan bahwa, waktu *plant* untuk mencapai keadaan tunak yaitu pada 7.61 detik dengan waktu naik sebesar 11.64 detik dan waktu untuk mencapai *Steady State* adalah 59.36 detik. Berdasarkan respon pada Gambar 14 dapat diamati bahwa *Towercopter* mampu mengikuti perubahan setpoint yang diberikan dengan *Error Steady-State* sebesar 0.21 % dan Maksimal *Overshoot* sebesar 24.48 %.

## PENUTUP

### Simpulan

Dalam merancang sistem pengendali posisi ketinggian pada *Towercopter* menggunakan PI-ANN melalui beberapa tahapan yakni dengan penurunan model matematis *plant* untuk mengetahui *Transfer Function* dari *Towercopter*. Merancang kontroler PI untuk mendapatkan parameter  $K_p$  dan  $K_i$  sebagai parameter acuan. Parameter yang didapatkan adalah  $K_p = 0.16$  dan  $K_i = 0.51$ . Yang terakhir, adalah merancang kontroler PI-ANN. Untuk merancang kontroler ini Proses *Learning* dilakukan bertujuan untuk melatih *plant* agar dapat mengenali target sesuai yang diinginkan. Algoritma yang digunakan adalah *Backpropagation*. Kontroler ini kemudian diuji-coba sehingga dihasilkan respon *plant* pada Gambar 9 dengan nilai *Rise Time* = 13.41 detik, *Settling Time* = 52.28 detik, Maksimal *Overshoot* = 22.3% dan *Error Steady State* = 0.43%. Selanjutnya, *plant* diuji menggunakan beban seberat 2.32 gr sehingga dihasilkan respon *plant* pada Gambar 10 mengalami penurunan respon sebesar 4 cm yang kemudian bergerak naik pada detik ke-173 dan akhirnya Respon mencapai setpoint pada detik ke- 213. Terakhir, *plant* diuji coba dengan setpoint yang diubah. Hasil respon *plant* pada Gambar 11, Gambar 12 dan Gambar 13 menunjukkan bahwa kontroler PI-ANN dapat mengontrol *plant Towercopter* agar dapat beradaptasi sesuai dengan target ketika terjadi perubahan setpoint. Respon yang dihasilkan cenderung menyerupai respon target yang diinginkan namun pada percobaan menggunakan setpoint 17 dan 19 cm, waktu untuk mencapai keadaan *Steady State* tidak sama yakni dengan selisih  $\pm 20$  detik.

### Saran

Dapat dikembangkan dengan mengganti sensor *HC-SR04* dengan sensor jarak yang lebih presisi yakni sensor *Proximity* ataupun sensor *Rangefinder Parallax*. Karena pada penelitian ini sensor yang digunakan tidak presisi dan cenderung tidak stabil sehingga respon yang dihasilkan tidak bagus. Dapat dikembangkan menggunakan Kontroler PID-ANN yang bertujuan untuk mengurangi *osilasi* yang ada. Karena pada penelitian ini *osilasi* yang dihasilkan sistem masih besar.

## DAFTAR PUSTAKA

- Agfianto, T. (2012). *Monitoring Suhu Ruangan Server Berbasis Raspberry Pi Menggunakan Sensor Suhu Dht11*. Doctoral Dissertation, Politeknik Negeri Sriwijaya.
- Cristian, Pavas, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixia, F. Ruess, M. Suppa and D. Burschka. (2016). *Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue*. IEEE Robotics Automation Magazine, 19 (3) 46-56.
- García, Marta Salas, J., López-Granados, F., De Castro, A. I., & Peña-Barragán, J. M. (2010). *Configuration and specifications of an unmanned aerial vehicle (UAV) for early site specific weed management*. PloS one, 8(3), e58210.
- Hani, S. (2010). *Sensor Ultrasonik SRF05 Sebagai Memantau Kecepatan Kendaraan Bermotor*. Jurnal Teknologi, 3(2), 3.
- Moon, J. S., Kim, C., Youm, Y., & Bae, J. (2018). *UNI-Copter: A Portable Single-Rotor-Powered Spherical Unmanned Aerial Vehicle (UAV) With An Easy-To-Assemble And Flexible Structure*. Journal of Mechanical Science and Technology, 32(5), 2289-2298.
- Ogata, Katsuhiko. 1985. *Teknik Kontrol Automatik jilid 1*. Terjemahan Edi Laksono. Jakarta: Erlangga.
- Sánchez-Benítez, D., Portas, E. B., Jesús, M., & Pajares, G. (2012). *Vertical Rotor For The Implementation Of Control Laws*. IFAC Proceedings Volumes, 45(11), 224-229.
- Torres-Sánchez, J., López-Granados, F., De Castro, A. I., & Peña-Barragán, J. M. (2013). *Configuration And Specifications Of An Unmanned Aerial Vehicle (UAV) For Early Site Specific Weed Management*. PloS one, 8(3), e58210.
- Wicaksono, Prihatama Kunto (2016) *Pengendalian Pergerakan Hover Quadcopter Menggunakan Metode PID Jaringan Syaraf Tiruan*. Undergraduate thesis, Institut Teknologi Sepuluh Nopember.