

BAB II TINJAUAN PUSTAKA

2.1. Udang

Udang Air Tawar merupakan hewan air yang hidup di sungai dan di rawa-rawa serta danau. Hewan ini mampu hidup pada selang parameter air yang lebar, tidak toleran terhadap kandungan oksigen terlarut yang rendah bahkan pada air keruh. Udang ini juga memiliki nilai ekonomis tinggi karena dapat digunakan sebagai makanan alternatif untuk dikonsumsi maupun udang hias. Kandungan gizi tinggi, pemasaran yang mudah, tidak memerlukan lahan luas dan biaya pemeliharaan relatif kecil menyebabkan udang ini diminati banyak orang untuk dibudidayakan. Jenis udang yang digunakan pada penelitian ini adalah udang merah *red swamp crayfish* (*Procambarus clarkii*) yang di tunjukkan pada gambar di bawah ini merupakan spesies udang yang begitu invasif. Ciri khasnya adalah seluruh tubuhnya berwarna merah bata untuk jantan, sedangkan betina orange kemerah-merahan. Ukuran tubuhnya lebih kecil dibandingkan dengan *cherax sp.* *Procambarus clarkii* dewasa panjang tubuh hanya berkisar 12 cm dengan berat 75-100 gram per ekor. (Abdul Qolik, 2014). Gambar udang *red swamp crayfish* seperti yang diperlihatkan pada Gambar 2.1.



Gambar 2.1 Udang merah *red swamp crayfish* (*Prochambarus clarkii*)
Sumber: <http://tutorialvideo.info/tag/cara-budidaya-udang-hias-air-tawar> (2012)

2.2. Sensor pH

pH adalah pengukuran dari kadar keasaman atau kadar alkali sebuah larutan. Asam alkali adalah sebuah zat yang mengandung hydrogen dan menyerahkan ion-ion hydrogen dalam larutan atau disebut donor proton. Basa mengandung OH dan menyerahkan ion-ion hidoksida dalam larutan encer atau disebut penerima proton.

Pada perancangan ini digunakan sensor pH kit SKU SEN0161 yang dirancang untuk arduino, yang ditunjukkan pada Gambar 2.2. Sensor pH kit SKU: SEN0161 memiliki LED (Light Emitting Diode) yang bekerja sebagai indikator power, bnc konektor, dan antarmuka sensor



Gambar 2. 2 Sensor pH SKU SEN0161
Sumber: www.sensorex.com/pH_Sensor (2013)

2.3. Pompa

Pompa adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan dari suatu tempat ke tempat yang lain melalui suatu media perpipaan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus. Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara masukan (*suction*) dengan keluaran (*discharge*). Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran, bentuk pompa dapat dilihat pada gambar 2.3 berikut



Gambar 2. 3 Pompa
Sumber: <https://fajar-khurniawan.blogspot.co.id/2016/12/wiper-dan-washer-pend-teknik-otomotif.html> (2016)

2.4. Motor Direct Current (DC)

Motor DC merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya, memutar *impeller* pompa, *fan* atau *blower*, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (*mixer*, bor listrik, *fan* angin) dan di industri. Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorenz, yaitu membangkitkan fluksi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Metode *Pulse Width Modulation* (PWM) adalah salah satu metode yang dapat digunakan untuk mengatur kecepatan dari motor DC.

Motor DC banyak digunakan sebagai aktuator karena memiliki banyak keunggulan. Keunggulannya antara lain pengoperasian dan perawatan mudah, batas kecepatan kerja yang luas, dapat dikendalikan secara otomatis maupun manual, mempunyai bentuk lebih ringkas dan tidak menimbulkan polusi (Gatot, 2000). Contoh bentuk motor DC dapat dilihat dalam Gambar 2.4

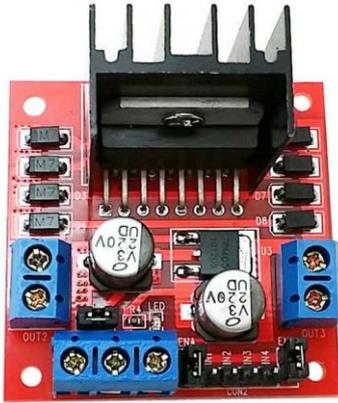


Gambar 2. 4 Motor direct current (DC)

Sumber: <http://www.globalsources.com/manufacturers/12V-DC-Motor.html> (2015)

2.5. Driver Motor

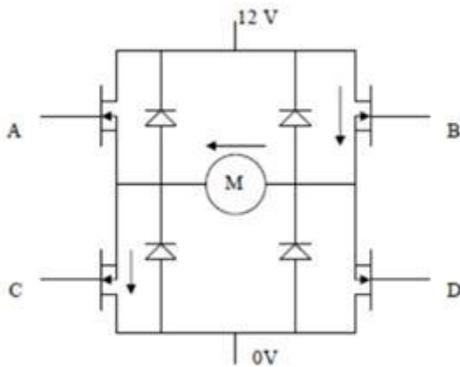
Driver motor berfungsi sebagai mengubah sinyal PWM dari mikrokontroler menjadi tegangan. *Driver* motor L298N merupakan *driver* yang digunakan untuk mengontrol atau mengendalikan kecepatan serta arah motor DC. Dalam aplikasinya *driver* motor biasanya tersusun dari rangkaian transistor-transistor yang tersusun sedemikian rupa sehingga mampu mengendalikan arah putar dan kecepatan motor berdasarkan arah loop dan tegangan kutub motor. *Driver* yang akan digunakan adalah driver motor. L298N dapat dilihat dalam Gambar 2.5



Gambar 2. 5 Driver motor L298N

Sumber: <http://yujum.com/rangkaian-dan-skematik-driver-motor-l298n> (2014)

H-bridge adalah sebuah perangkat keras berupa rangkaian yang berfungsi untuk menggerakkan motor. Rangkaian ini diberi nama *H-bridge* karena bentuk rangkaiannya yang menyerupai huruf H seperti pada Gambar berikut:



Gambar 2. 6 Konfigurasi *H-bridge* MOSFET

Rangkaian ini terdiri dari dua buah MOSFET kanal P dan dua buah MOSFET kanal N. Prinsip kerja rangkaian ini adalah dengan mengatur mati-hidupnya ke empat MOSFET tersebut. Huruf M pada gambar adalah motor DC yang akan dikendalikan. Bagian atas rangkaian akan dihubungkan dengan sumber daya kutub positif, sedangkan bagian bawah rangkaian akan dihubungkan dengan sumber daya kutub negatif. Pada saat MOSFET A dan MOSFET D on sedangkan MOSFET B dan MOSFET C off, maka sisi kiri dari gambar motor akan terhubung dengan kutub positif dari catu daya, sedangkan sisi sebelah kanan motor akan terhubung dengan kutub negatif dari catu daya sehingga motor akan bergerak searah jarum jam, jika kita melakukan proses sebaliknya maka arah putar akan berlawanan arah jarum jam

2.6. Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega328 (lihat Gambar 2.7). *Board* ini memiliki 54 digital *input/output* pin (14 pin dapat digunakan sebagai *output* PWM), 16 *input* analog, 16 MHz osilator kristal, USB koneksi, jack listrik, *header* ICSP, dan tombol reset.



Gambar 2. 7 Arduino mega 2560

Sumber: www.electroschematics.com

Arduino Mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Eksternal (non-USB) daya dapat berasal baik dari AC ke adaptor DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2,1 mm konektor *power*. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor power. Arduino dapat beroperasi dengan catu daya eksternal 6 sampai 20 volt. Namun jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak papan. Kisaran yang disarankan adalah 7 sampai 12 volt.

ATmega 2560 memiliki 256 KB (dengan 8 KB digunakan untuk *bootloader*), 8KB dari SRAM dan 4 KB EEPROM. Masing-masing dari 54 pin digital di Arduino Mega 2560 dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (), beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara default terputus) dari 20-50 Kohm. Selain itu, beberapa pin memiliki fungsi khusus:

- *Serial*: 0 (RX) dan 1 (TX); *Serial* 1: 19 (RX) dan 18 (TX); *Serial* 2: 17 (RX) dan 16 (TX); *Serial* 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.

- *Eksternal interrupts*: 2 (*interrupt 0*), 3 (*interrupt 1*), 18 (*interrupt 5*), 19 (*interrupt 4*), 20 (*interrupt 3*), dan 2 (*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
- PWM: 0 - 13. Menyediakan output PWM 8-bit dengan fungsi *analogWrite ()*.
- *SPI*: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan [SPI library](#).
- *LED*: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai *high* LED menyala dan ketika pin bernilai *low* LED mati.
- *I²C*: 20 (SDA) dan 21 (SCL). Dukungan *I²C* (TWI) komunikasi menggunakan *wire*.

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. Arduino Mega 2560 menyediakan 4 UART TTL (5V) untuk komunikasi serial. Sebuah Arduino Mega 2560 sebagai saluran komunikasi serial melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan driver USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun pada Windows diperlukan sebuah file inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari papan Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah [Software Serial Library](#) memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board Arduino Mega 2560. Arduino Mega 2560 juga mendukung *I²C* (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan kawat untuk menyederhanakan penggunaan bus *I²C*.

2.7. Pulse Width Modulation (PWM)

PWM (*Pulse Width Modulation*) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (*duty cycle*) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi high kemudian berada di zona transisi ke kondisi *low*. Pengaturan lebar pulsa tersebut dilakukan untuk menentukan nilai *duty cycle*. *Duty cycle* adalah rasio lebar pulsa high dibagi dengan total lebar pulsa high dan *low*. Nilai

duty cycle mulai dari 0 untuk nilai PWM = 0, hingga 100 untuk nilai PWM = 5V (Artanto, 2009).

2.8. Kontroler

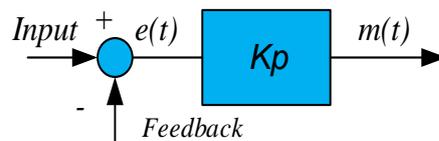
Kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Hal itu disebabkan oleh tidak dapat dirubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, perubahan perilaku sistem hanya dapat dilakukan melalui penambahan kontroler.

Salah satu fungsi kontroler adalah mengurangi sinyal *error*, sinyal *error* adalah perbedaan nilai *setpoint* dengan nilai *output plant*. *Setpoint* adalah nilai referensi atau nilai yang diinginkan, sedangkan *output plant* adalah nilai aktual yang terukur pada *output plant*. Semakin kecil nilai sinyal *error* maka kinerja sistem kontrol dinilai semakin baik.

Prinsip kerja kontroler adalah membandingkan nilai *output plant* dengan nilai *setpoint*, menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan *error* (Ogata K, 1997).

2.8.1. Kontroler *Proportional*

Kontroler *proportional* memiliki *output* yang besarnya sebanding dengan besarnya sinyal *error*. *Output* kontroler merupakan perkalian antara penguatan proporsional dengan sinyal *error*. Gambar 2.8 menunjukkan diagram blok kontroler *proportional* dan Persamaan 2.1 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2. 8 Diagram blok kontroler *proportional*

Sumber: Ogata K. (1997)

$$m(t) = K_p e(t) \dots \dots \dots (2.1)$$

dimana:

K_p = adalah penguatan *proporsional*

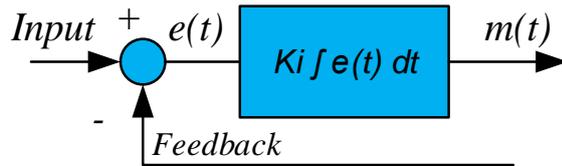
$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

Penambahan K_p akan mempercepat kecepatan respon *transient* dan mengurangi kesalahan keadaan mantap.

2.8.2. Kontroler Integral

Kontroler *integral* memiliki karakteristik seperti sebuah operasi *integral*, *output* kontroler dipengaruhi oleh perubahan yang sebanding dengan perubahan nilai sinyal *error*. *Output* kontroler merupakan penjumlahan terus menerus dari perubahan sinyal *error*. Gambar 2.9 menunjukkan diagram blok kontroler *proportional* dan Persamaan 2.2 dan Persamaan 2.3 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2. 9 Diagram blok kontroler *integral*

Sumber: Ogata K. (1997)

$$\frac{dm(t)}{dt} = Ki e(t) \dots\dots\dots(2.2)$$

$$m(t) = Ki \int e(t) dt \dots\dots\dots(2.3)$$

dimana:

Ki = adalah penguatan *integral*

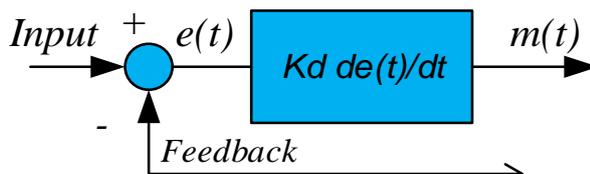
e(t) = sinyal *error*

m(t) = *output* kontroler

Aksi kontrol *integral* digunakan untuk menghilangkan sinyal *error* dalam keadaan mantap.

2.8.3. Kontroler Derivative

Kontroler *derivative* memiliki sifat seperti suatu operasi *derivative*. Perubahan yang mendadak pada masukan kontroler mengakibatkan perubahan yang sangat besar dan cepat. Kontroler ini tidak akan menghasilkan *output* saat sinyal *error* konstan sehingga tidak akan mempengaruhi keadaan mantap. Gambar 2.10 menunjukkan diagram blok kontroler *derivative* dan Persamaan 2.4 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2. 10 Diagram blok kontroler *derivative*

Sumber: Ogata K. (1997)

$$m(t) = Kd \frac{de(t)}{dt} \dots \dots \dots (2.4)$$

dimana:

Kd = adalah penguatan *derivative*

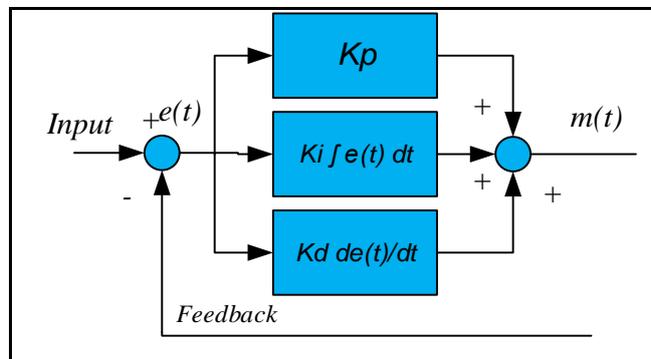
$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

Kontroler ini digunakan untuk memperbaiki atau mempercepat respon *transient*.

2.8.4. Kontroler *Proportional Integral Derivative (PID)*

Gabungan aksi kontrol *proportional*, *integral*, dan *derivative* yang terlihat dalam Gambar 2.11 mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut (Persamaan 2.5):



Gambar 2. 11 Diagram blok kontroler PID

Sumber: Ogata K. (1997)

$$m(t) = Kp \cdot e(t) + Ki \cdot \int e(t) dt + Kd \frac{de(t)}{dt} \dots \dots \dots (2.5)$$

dimana:

Kp = adalah penguatan *proportional*

Ki = adalah penguatan *integral*

Kd = adalah penguatan *derivative*

$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

2.9. Root Locus

Root Locus (tempat kedudukan akar) merupakan suatu analisis dalam keilmuan *control engineering* yang menggambarkan pergeseran letak pole-pole suatu sistem loop tertutup dari perubahan besarnya penguatan loop terbuka dengan gain adjustment. Analisis ini digunakan sebagai salah satu dasar untuk mendesain suatu sistem kendali sesuai dengan karakteristik dan spesifikasi yang diinginkan. Analisis *root locus* ini dapat menentukan apakah suatu sistem stabil atau tidak. Selain itu dapat menentukan besarnya rentang penguatan loop terbuka agar suatu sistem masih dapat dikatakan stabil (tetapi tidak bisa menstabilkan suatu sistem

tidak stabil secara utuh menjadi sistem yang stabil). Plot kurva root locus berada pada bidang-s (domain frekuensi).

Sifat-sifat dan beberapa catatan mengenai root locus :

- *Root locus* bermula dari pole-pole $G(s)H(s)$ untuk penguatan (K) sama dengan nol dan berakhir di zero-zero $G(s)H(s)$ untuk $K \rightarrow \infty$ (termasuk zero-zero pada titik tak hingga).
- Spesifikasi transien dapat diatur dengan mengatur nilai K untuk mendapatkan respon waktu yang diinginkan. Mengubah bentuk root locus berarti mengubah respon transien (biasanya dengan kompensasi fasa maju yang mengakibatkan adanya efek penambahan zero)
- Keakuratan sistem dapat diperbesar dengan menambahkan pole di origin bidang-s yang berarti menambah tipe sistem yang mengakibatkan konstanta galat tak hingga dan galat dapat menjadi nol. Hal ini dapat pula diimplementasikan dengan kompensasi fasa mundur (memperbesar gain tanpa mengubah kurva *root locus*)

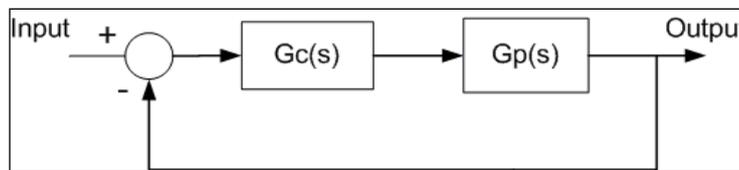
Prosedur penggambaran *root locus* dapat digambar secara manual ataupun digambar dengan menggunakan Matlab. Berikut ini adalah prosedur penggambaran kurva Root Locus secara manual :

1. Menentukan *closed loop transfer function* dan meletakkan pole-pole dan zero-zero pada bidang-s
2. Menentukan interval terdapatnya root locus pada sumbu nyata. Bila interval daerah sumbu real mempunyai jumlah pole dan atau zero di sebelah kanannya bernilai ganjil, maka daerah tersebut terdapat *root locus*.
3. Menentukan jumlah asimtot, sudut asimtot, dan perpotongan asimtot dengan sumbu real
4. Menentukan titik pencar dan temu pole-pole (*break away point* dan *break in point*)
5. Menentukan titik potong kurva *root locus* dengan sumbu imajiner (jika ada)
6. Menentukan sudut datang (untuk zero) dan sudut berangkat (untuk pole)
7. Sketsa *root locus* dari data-data yang telah didapatkan.

Pencarian nilai parameter PID dengan menggunakan metode Root Locus:

Rancangan sistem kendali loop tertutup menggunakan *root locus* memungkinkan untuk mengatur sekurang-kurangnya beberapa letak *pole* sistem loop tertutup

sehingga dapat mengatur tanggapan transien pada tingkat tertentu dan pengaruhnya terhadap tanggapan keadaan mantap (Philips,1996:209). Prosedur analitis perancangan kontroler PID menggunakan metode *root locus* yang dijelaskan dalam *Feedback Control System* oleh Charles L. Philips dan Royce D. Harbour dapat dilihat dalam Gambar 2.12 berikut:



Gambar 2. 12 Sistem kendali loop tertutup
 Sumber: Charles L. Philips dan Royce D. Harbour (1996)

Untuk sistem tersebut, persamaan karakteristik diberikan persamaan 2.6:

$$1 + Gc(s)Gp(s) = 0 \dots\dots\dots(2.6)$$

Misalkan diinginkan lokus akar melalui $s = s_1$, maka hasil dari persamaan ditunjukkan pada persamaan 2.7:

$$Gc(s_1)Gp(s_1) = -1$$

$$Gc(s_1)|Gp(s_1)|e^{j\psi} = 1e^{j\pi} \dots\dots\dots(2.7)$$

Fungsi alih kontroler PID setelah ditransformasi laplace dinyatakan oleh persamaan 2.8:

$$Gc(s) = Kp + \frac{Ki}{s} + Kd s \dots\dots\dots(2.8)$$

Perhitungan dari persamaan 2.10 ditunjukkan pada persamaan 2.9:

$$Gc(s_1) = \frac{1}{|Gp(s_1)|} e^{j(\pi-\psi)} \dots\dots\dots(2.9)$$

Substitusi persamaan 2.9 pada persamaan 2.5 didapatkan persamaan 2.10:

$$Kd s_1^2 + Kp s_1 + Ki = \frac{e^{j(\pi-\psi)}}{|Gp(s_1)|} \dots\dots\dots (2.10)$$

Dengan

$$s_1 = |s_1|e^{j\beta} \dots\dots\dots(2.11)$$

Hasil dari substitusi persamaan 2.11 ke persamaan 2.10 didapatkan pada persamaan 2.12:

$$Kd|s_1|^2(\cos 2\beta + j \sin 2\beta) + Kp|s_1|(\cos \beta + j \sin \beta) + Ki \dots\dots\dots (2.12)$$

Menyamakan real dengan real dan imajiner dengan imajiner, didapatkan hasil pada persamaan 2.13:

$$\begin{bmatrix} |s_1|^2 \cos 2\beta & |s_1| \cos \beta \\ |s_1|^2 \sin 2\beta & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} \frac{|s_1|}{Gp(s_1)} \cos(\beta + \Pi + \psi) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\beta + \Pi + \psi) \end{bmatrix} \dots\dots\dots (2.13)$$

Atau dapat ditunjukkan pada persamaan 2.14:

$$\begin{bmatrix} |s_1|^2 \cos 2\beta & |s_1| \cos \beta \\ |s_1|^2 \sin 2\beta & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} -\frac{|s_1|}{Gp(s_1)} \cos(\psi - \beta) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\psi - \beta) \end{bmatrix} \dots\dots\dots(2.14)$$

Dari persamaan dapat dilihat bahwa untuk perancangan kontroler PID, satu dari tiga penguatan Kp, Ki, Kd, harus ditentukan dahulu.