

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang telah dilakukan sebelumnya menggunakan sistem PID dapat dilihat dalam [8]-[15]. Pertama, [8] menunjukkan perbandingan hasil respon dari pengatur suhu ketika diberikan sistem *proportional*, *proportional integral*, dan PID yang menghasilkan grafik PID dengan nilai *steady state error* atau nilai kestabilan sistem sebesar 0,197, 0, dan 0, berturut-turut sehingga sistem yang menggunakan nilai PID lebih stabil dibandingkan dengan hanya menggunakan *proportional* dan *proportional integral*. Pada dasarnya, nilai *steady state error* mencerminkan kestabilan grafik pada PID. Ketika PID digabungkan dengan *fuzzy* dan diimplementasikan ke *brushless* motor DC, performa dapat ditingkatkan walaupun menggunakan nilai *setpoint* dan beban yang berbeda [9]. Selain itu, [10] menggunakan sistem *fuzzy*-PID untuk menggerakkan model *brushless* motor DC yang disimulasikan menggunakan *Simulink*, lalu grafik *speed tracking* ditampilkan dengan nilai $K_p = 0,2$, $K_i = 4$, dan $K_d = 0,04$. Kemudian, [11] yang mengimplementasikan PID berbasis penalaan Ziegler-Nichols dengan *hopper conveyor* sebagai media *plant* sehingga didapatkan nilai $K_p = 13,77$, $K_i = 0,25$, dan $K_d = 1,02$ yang menghasilkan kestabilan tercerni pada grafik PID.

Dalam menjalankan *multi range hydro-mechanical continuously variable transmission* (HMCVT), berat digunakan untuk mengubah nilai pada sistem *fuzzy*-PID sehingga dapat digunakan sebagai sistem transmisi yang stabil [12]. Sistem dalam [9] dan [11] juga menghasilkan grafik PID yang stabil dan telah mencapai nilai yang diinginkan (*setpoint*), serta memiliki grafik tidak bergelombang sehingga motor DC dapat berputar dengan konstan. Selanjutnya, penggunaan PID dalam mengendalikan *quadrant chopper* dengan nilai *gain* $K_p = 3,9956$, $K_i = 21,042$, dan $K_d = 0,6173$ menghasilkan putaran empat motor DC

konstan [13]. Lalu, [14] menggunakan dua roda dengan nilai PID pada masing-masing roda menghasilkan pergerakan yang stabil.

Pada penelitian kami, sistem yang digunakan hampir sama dengan [11], tetapi kami menggunakan berat untuk mengubah nilai *gain* PID dalam menggerakkan motor DC. Kemudian, kami menggunakan sistem PID yang diterapkan pada dua roda seperti [14], tetapi sistem PID yang kami implementasikan mengatur kedua roda sekaligus, tidak masing-masing atau hanya menggunakan satu nilai PID untuk mengendalikan dua roda. Kami juga menentukan nilai *gain* PID seperti dengan [11] dan [13] untuk mengatur kecepatan putaran motor DC. Sejauh yang kami ketahui, penelitian yang kami usulkan merupakan modifikasi pertama kali dan belum ada yang melakukannya.

2.2 Sistem PID

Sistem PID adalah salah satu sistem yang digunakan dalam menyelesaikan dan menemukan solusi permasalahan sistem kontrol dengan cara *proportional*, *integral*, dan *derivative* [8]. Pertama, *proportional gain* menyebabkan peningkatan respon sistem, tetapi hanya dapat meningkatkan hingga batas tertentu menyesuaikan *plant* yang digunakan. Pada hakikatnya, *plant* merupakan seperangkat alat yang menunjang kinerja pada suatu sistem. Kemudian, *integral gain* menyebabkan pengurangan nilai *steady state* atau nilai kestabilan pada sistem. Lalu, *derivative gain* menyebabkan penurunan dari variabel *output* jika *error* meningkat dan meminimalkan osilasi [5]. Fungsi transfer pada standar PID *controller* ditulis dalam bentuk persamaan umum pada (2.1) dan diturunkan dalam domain waktu pada (2.2) [8],

$$G(s) = K_p + K_i \frac{1}{s} + K_d s, \quad (2.1)$$

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right), \quad (2.2)$$

dimana K_p adalah *proportional gain*, K_i adalah *integral gain*, K_d adalah *derivative gain*, T_i adalah *integral time constant* atau pembagian nilai 1 dengan $K_p K_i$, dan T_d adalah *derivative time constant* $K_p K_d$. Selanjutnya, sistem PID terbagi menjadi dua, yaitu *linear* dan *non-linear*. Pada dasarnya, *Linear* PID (L-

PID) mengambil *error*, $e(t)$, yang merupakan selisih antara nilai aktual dengan *setpoint* terhadap waktu dan memprosesnya sesuai dengan (2.3). Berkaitan dengan itu, struktur skematik L-PID ditunjukkan pada Gambar 2.1 [5],

$$u(t) = K_p e(t) + K_i \int_{t_1}^{t_2} e(t) dt + K_d \frac{de(t)}{dt}, \quad (2.3)$$

dimana $u(t)$ merupakan sinyal kontrol dalam domain waktu. Lalu, *Non-linear* PID (NL-PID) merupakan metode dimana *error* memiliki nilai yang *non-linear* untuk mendapatkan sinyal kontrol $u(t)$ yang ditunjukkan oleh [11]

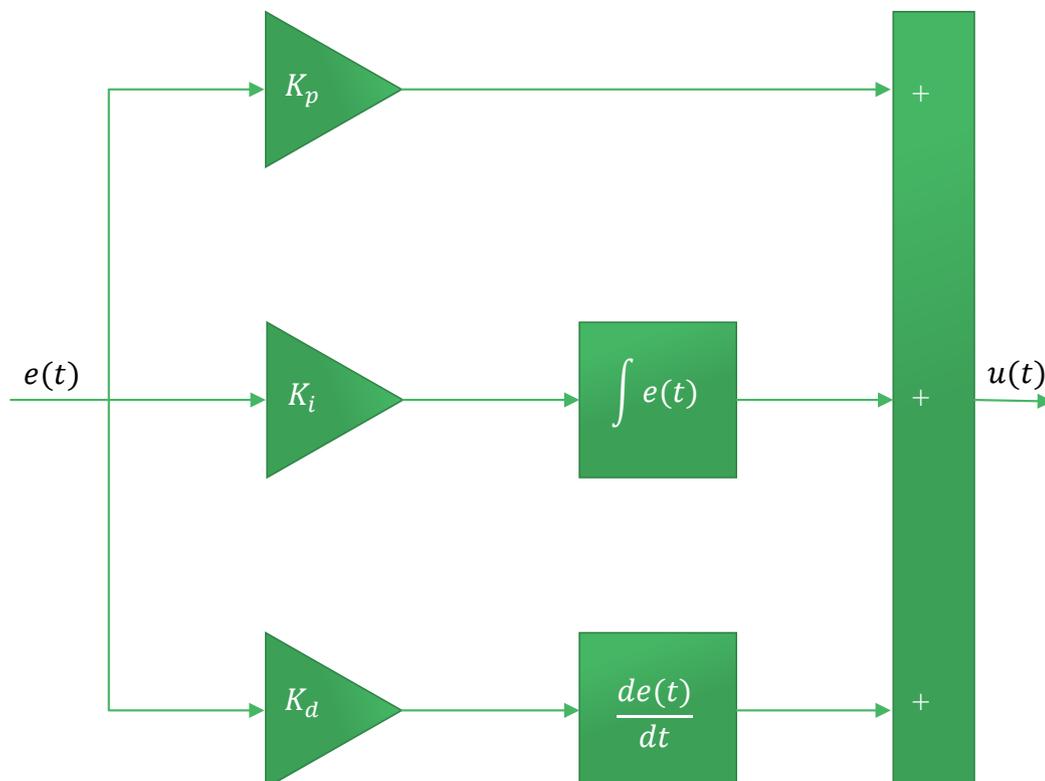
$$u(t) = e_p(t) + e_i(t) + e_d(t), \quad (2.4)$$

dimana

$$e_p(t) = f(K_p, e(t)), \quad (2.5)$$

$$e_i(t) = f(K_i, e(t)), \quad (2.6)$$

$$e_d(t) = f(K_d, e(t)), \quad (2.7)$$



Gambar 2.1 Struktur Skematik L-PID

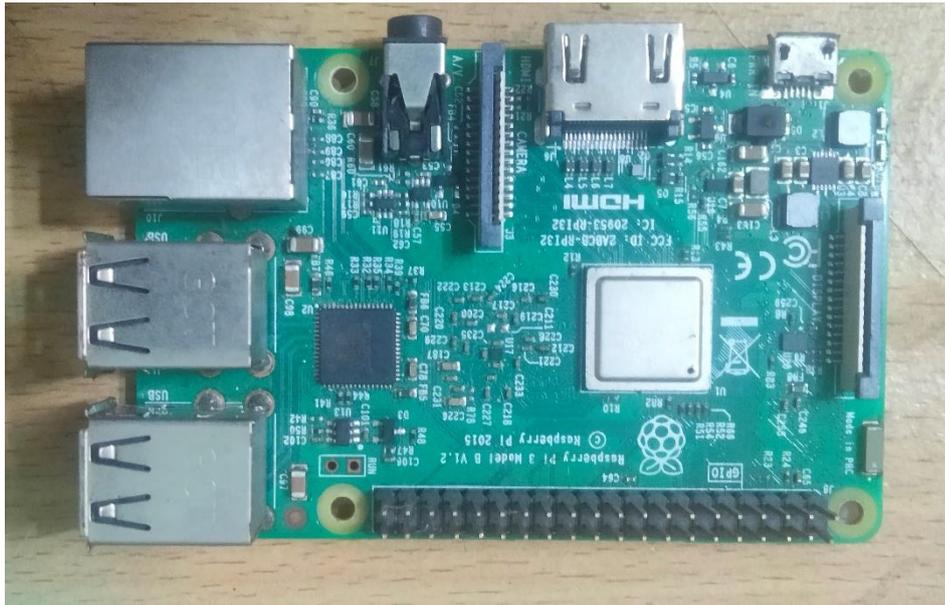
dengan $f(\cdot)$ merupakan fungsi *non-linear*, $e_p(t)$ merupakan nilai *error proportional* dalam domain waktu, $e_i(t)$ merupakan nilai *error integral* dalam domain waktu, dan $e_d(t)$ merupakan nilai *error derivative* dalam domain waktu.

2.3 Mini Personal Computer (PC)

Mini PC merupakan komputer versi mini atau kecil yang memiliki sistem yang sama seperti komputer pada umumnya. Alat ini memiliki *processor*, *Random Access Memory* (RAM), *port Local Area Network* (LAN), *port Universal Serial Bus* (USB), *port High Definition Multimedia Interface* (HDMI), *Wireless Fidelity* (Wi-Fi), Bluetooth, *micro Serial Data* (SD), *stereo output* dan memiliki pin *General Purpose Input Output* (GPIO) [16]. Pada produk kami, Raspberry Pi 3 Model B digunakan sebagai mini PC dalam menjalankan produk yang ditunjukkan pada Gambar 2.2 dan merupakan model awal dari generasi ketiga Raspberry Pi 3. Pada dasarnya, komponen ini dilengkapi dengan prosesor Quad Core 1.2 GHz Broadcom BCM2837 64 bit, 1 GB RAM, dan 40 pin GPIO yang dipakai untuk menjalankan komponen lain pada produk kami. Selanjutnya, spesifikasi lengkap Raspberry Pi 3 model B ditunjukkan pada Tabel 2.1 [17]. Mini PC Raspberry Pi 3 Model B ini dipilih karena memiliki kemampuan untuk dapat menerima data dari serial Arduino yang berupa kecepatan putaran motor DC, mengatur kecepatan motor DC melalui *driver* motor, dan melakukan perhitungan PID.

Tabel 2.1 Spesifikasi Mini PC Raspberry Pi 3 Model B

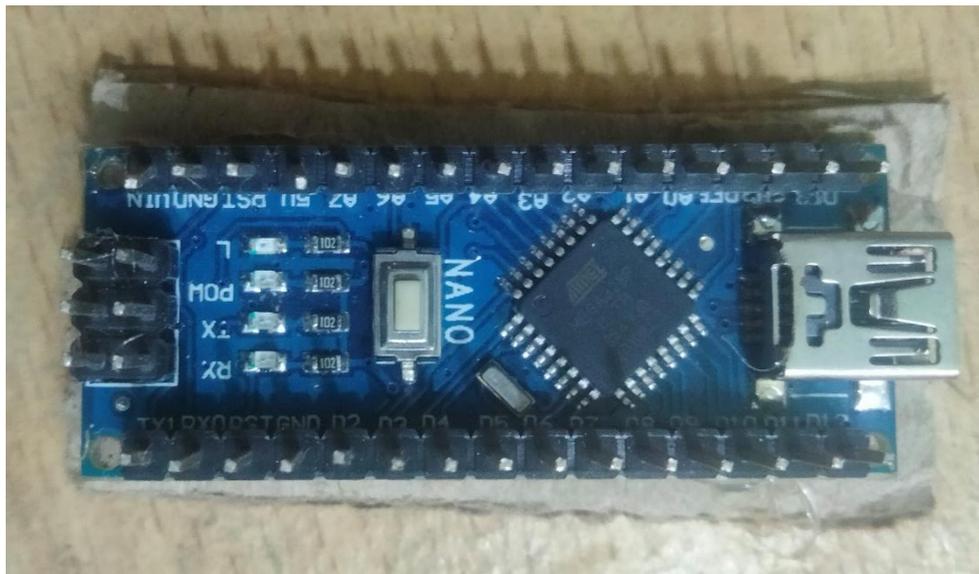
Nama Komponen	Raspberry Pi 3 Model B
Prosesor	<i>Quad Core</i> 1.2 GHz Broadcom BCM2837 64 bit
LAN	100 <i>Base Ethernet</i>
GPIO	40 pin
USB	4 USB 2 <i>ports</i>
Power Supply	5 V 2,1 A



Gambar 2.2 Mini PC.

2.3 Mikrokontroler Arduino

Arduino merupakan suatu mikrokontroler yang menggunakan sistem digital untuk mengontrol atau mengendalikan suatu peralatan elektronik. Dalam mengendalikan peralatan elektronik, Arduino dipermudah dengan menerapkan bahasa pemrograman yang dikenal dengan bahasa pemrograman Arduino [18]. Pada produk ini, Arduino dipakai untuk membaca kecepatan putaran motor DC menggunakan sensor kecepatan dengan sistem *interrupt* dimana sistem ini akan menjalankan suatu perintah tertentu ketika masih menjalankan perintah yang lain [19]. Arduino yang digunakan berjenis Nano yang ditunjukkan pada Gambar 2.3 yang berbasis *microchip* ATmega328 sebagai kendali. Mikrokontroler ini memiliki 8 pin analog dan 22 pin digital (6 *Pulse Width Modulation*). Berdasarkan hal tersebut, spesifikasi lengkap Arduino Nano yang dipakai produk ini ditunjukkan pada Tabel 2.2 [20]. Pada dasarnya, Arduino ini dipilih karena dapat mengirim data kecepatan motor DC melalui sensor kecepatan ke Raspberry Pi melalui komunikasi serial sesuai dengan kebutuhan penggunaan produk kami.



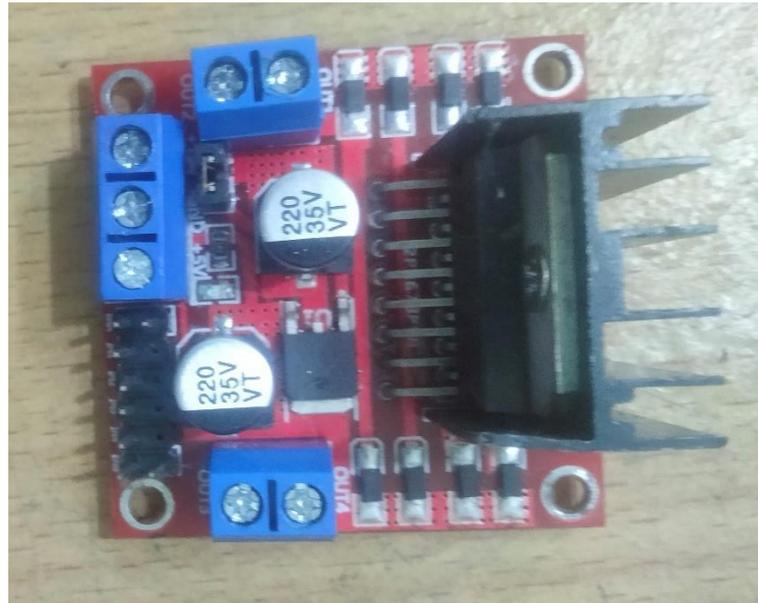
Gambar 2.3 Mikrokontroler.

Tabel 2.2 Spesifikasi Mikrokontroler Arduino Nano

Nama Komponen	Arduino Nano
Mikrokontroler	ATMega328
Catu Daya	5 V 19 mA
Pin Analog	8
Pin Digital	22
Dimensi	18 × 45 mm
Berat	7 g

2.4 Driver Motor DC

Driver motor DC merupakan *driver* dua arah yang dapat menggerakkan motor DC untuk arah maju maupun mundur. Selain itu, *driver* ini juga dapat menerima *input* berupa tegangan langsung dari baterai (*high voltage*) dan tegangan dari logika (*low voltage*). Kemudian, *driver* ini juga biasa digunakan untuk mengatur kecepatan putaran motor DC sesuai dengan logika yang diterima oleh *driver* [16]. Pada produk ini, *driver* yang digunakan berjenis L298N sebagai pengatur kecepatan putaran motor DC. Berdasarkan hal tersebut, L298N yang ditunjukkan pada Gambar 2.4 merupakan *integrated monolithic circuit* dengan *15-lead Multiwatt* dan paket *PowerSO20*. Selain itu,



Gambar 2.4 *Driver Motor*.

driver ini juga memiliki tegangan tinggi dan arus tinggi dual *full-bridge driver* untuk dapat menerima *standard TTL logic level*, dan beban seperti *relay*, *solenoid*, DC, dan *stepping motor*. Pada *driver* ini, dua masukan dapat diberikan nilai *high* atau *low* sesuai dengan masukan sinyal. Berkaitan dengan itu, spesifikasi lengkap *driver* ini dapat dilihat pada Tabel 2.3 [21]. Pada dasarnya, *driver* motor ini dipilih karena dapat mengendalikan kecepatan motor DC yang digunakan dengan keluaran arus yang sudah sesuai dengan kebutuhan motor DC dan mudah digunakan karena memiliki sistem *enable* dan *disable*, serta pengaturan arah putar pada motor DC.

Tabel 2.3 Spesifikasi *Driver Motor* L298N

Nama Komponen	L298N
<i>Power Supply Output</i>	5 – 12 V 2 A
<i>Power Supply Input</i>	1,5 – 5 V

2.5 Motor DC 12 V 400 RPM

Motor DC yang ditunjukkan pada Gambar 2.5 merupakan alat penggerak sebuah robot atau roda yang memiliki *gearbox* dan banyak digunakan di robot kapal, model mobil, peralatan rumah tangga, dan kontrol gerak linear. Kemudian, motor ini memiliki kecepatan 400 *revolution per minute* (RPM) dengan tegangan yang dibutuhkan sebesar 12 V. Berkaitan dengan itu,



Gambar 2.5 Motor DC.

spesifikasi lengkap motor DC ini dapat dilihat pada Tabel 2.4. Pada dasarnya, motor DC ini dipilih karena memiliki kecepatan linear yang dapat mengikuti pengguna dengan menggunakan perhitungan konversi pada (2.8),

$$v = \frac{2\pi r w}{60}, \quad (2.8)$$

dimana v merupakan kecepatan linear, π merupakan nilai pi atau perbandingan keliling lingkaran dengan diameternya yang bernilai 3.141592653, r merupakan jari-jari roda, dan w merupakan kecepatan putaran. Dengan menggunakan konversi (2.8) dan produk kami yang menggunakan roda berdiameter 20 cm, didapatkan kecepatan linear sebesar 4.18 m/s. Pada dasarnya, kecepatan jalan pengguna sebesar 2 km/h atau 0.55 m/s yang diteliti pada [22] sehingga kecepatan maksimum produk kami lebih cepat daripada kecepatan jalan pengguna yang menandakan bahwa produk sudah dapat mengikuti pengguna. Selain itu, dengan torsi yang dimiliki motor DC sebesar 6,5 kgcm dan produk kami menggunakan dua motor DC, maka menghasilkan total torsi 13 kgcm atau 13 kg dalam radius 1 cm dan nilai ini sudah cukup untuk dapat membawa barang 5 – 7 kg.

Tabel 2.4 Spesifikasi Motor DC 12 V 400 RPM

Nama Komponen	Motor DC 12 V 400 RPM
Tegangan Masukan	12 V
Kecepatan	400 RPM
<i>Gear</i>	1:21
Torsi	6,5 kgcm

2.6 Sensor Kecepatan

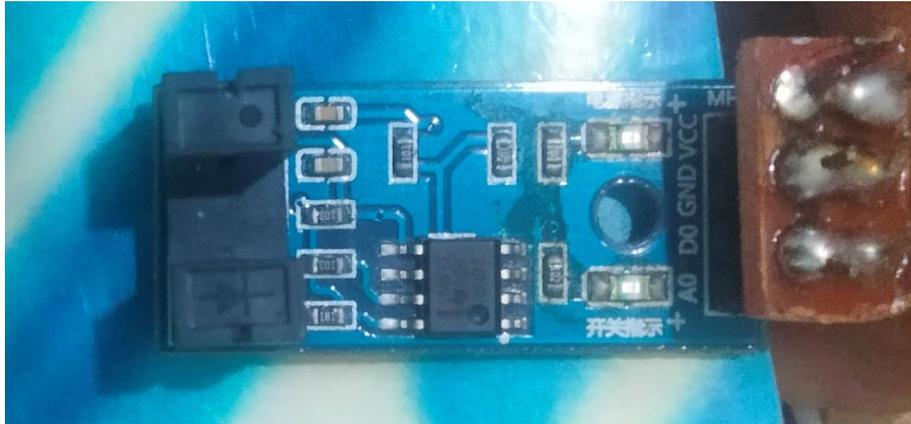
Sensor ini menggunakan *proximity sensor* dengan mengukur banyaknya putaran dari motor DC, lalu menghitung banyaknya lubang pada roda. Setelah mengetahui banyaknya lubang pada roda, data digital tersebut dikonversi menjadi data analog yang kemudian dilakukan proses perhitungan hingga menjadi nilai RPM atau putaran per menit [19]. Pada produk ini, sensor yang digunakan berjenis *infrared (IR) speed sensor* yang ditunjukkan pada Gambar 2.6 dengan menghitung banyak lubang *encoder* pada roda motor DC. Berdasarkan hal tersebut, perhitungan ditunjukkan pada (2.9),

$$RPM = \frac{n}{m} \times 60, \quad (2.9)$$

dimana n merupakan banyak lubang yang terbaca oleh sensor kecepatan pada *encode* dalam 1 detik dan m merupakan total lubang pada *encoder*. Pada dasarnya, sensor ini digunakan sebagai masukan dalam PID yang akan memberikan data berupa perubahan kecepatan motor DC ketika produk kami diberikan berat yang berbeda. Berkaitan dengan itu, spesifikasi lengkap sensor ini terdapat pada Tabel 2.5 [23].

Tabel 2.5 Spesifikasi Sensor Kecepatan

Nama Komponen	<i>IR Speed Sensor</i>
<i>Integrated Circuit</i>	LM393
<i>Power Supply</i>	3.3 – 5 V
Dimensi	32 × 14 × 7 mm



Gambar 2.6 Sensor Kecepatan.