BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Dalam penelitian ini, akan digunakan referensi atau tinjauan pustaka berdasarkan latar belakang yang telah diuraikan sebelumnya. Beberapa tinjauan pustaka dari penelitian terdahulu dan beberapa penelitian yang relevan dengan penelitian yang akan dilakukan oleh penulis akan digunakan untuk mendukung penulisan skripsi. Informasi lebih lanjut tentang hal ini dapat ditemukan pada Tabel 2.1 yang disajikan di bawah ini.

Tabel 2.1. Tinjauan Pustaka

No.	Judul Penelitian	Tahun	Hasil
1.	Penelitian oleh (Lamatenggo et al., 2020) dalam jurnalnya yang berjudul "Perancangan Balancing Robot Beroda Dua Dengan Metode Pengendali PID Berbasis Arduino Nano"	2020	Telah dibuat sebuah model robot beroda dua yang dapat menjaga keseimbangannya (Balancing Robot) pada permukaan datar. Metode pengendali yang digunakan adalah Pengendali Proporsional, Integral, Derivatif (PID). Komponen penyusun model balancing robot adalah Arduino Nano, Motor driver L298N, Motor DC 6V 620rpm, Gearbox 25ga370, dan Baterai Lipo 2200mAh 3S 25C XT60. Balancing robot dapat menyeimbangkan diri pada Axis Y dan Axis Z . Nilai konstanta pengendali PID diperoleh Kp = 60, Ki = 2.0 dan Kd =130 sehingga

No.	Judul Penelitian	Tahun	Hasil
			sistem ini mampu menstabilkan posisi robot.
2.	Penelitian oleh (Hendra Wijaya et al., 2018)dalam jurnalnya yang berjudul "Balancing Robot Roda Dua dengan Metode Rule base Berbasis Mikrokontroller Arduino" ."	2018	Respon robot yang dihasilkan untuk mencapai titik tegak sangat cepat. Seperti pada percobaan dengan set awal kemiringan robot -11 hingga -70 dengan rata-rata respon 421 iterasi dan set awal kemiringan robot 13 hingga 70 derajat dengan respon rata-rata untuk mencapai tegak yaitu 386 iterasi. Pada pengujian kecepatan motor robot, saat robot dimiringkan hingga kemiringan maksimal maka output dari rule base di setting dengan kecepatan maksimal motor, jika robot mulai mendekati titik tegak maka setting kecepatan motor akan lebih kecil di bandingkan saat posisi robot dimiringkan maksimal..
3.	Penelitian oleh (Fathoni et al., 2021)dalam jurnalnya yang berjudul "Implementasi Kendali Keseimbangan Gerak Two Wheels Self balancing robot Menggunakan Fuzzy Logic"	2021	Dari penelitian ini robot mampu mempertahankan posisi agar tidak terjatuh dengan sudut maksimal yang teratasi yaitu hingga $17{,}5^0$ dalam waktu 1,7 detik pada posisi stabil direntang sudut di sekitar 5^0. Penelitian ini menggunakan metode fuzzy 9x9 fungsi keanggotaan dan 81 aturan.

No.	Judul Penelitian	Tahun	Hasil
			Complementary filter digunakan untuk menyaring noise dari MPU-6050. Pemilihan ESP-32 sebagai pengendali dilakukan dengan pertimbangan antara lain karena memiliki spesifikasi lebih tinggi dari seri Arduino seperti clock speed, kapasitas memori, dan kecepatan CPU yang lebih baik. Dapat disimpulkan dari hasil pengujian seluruh sistem self balancing robot dapat berjalan secara stabil meskipun masih terdapat error akan tetapi telah mencapai performa yang lebih baik dari beberapa penelitian terdahulu. Saran untuk penelitian selanjutnya adalah meningkatkan spesifikasi hardware yang digunakan serta pemilihan motor DC dengan torsi lebih besar, sehingga dapat digunakan metode kontrol gabungan antara PID dan fuzzy, predictive control, dan lain sebagainya, agar performa sudut maksimal lebih tinggi dan error dapat teratasi. Perlu diperhatikan juga dalam penentuan offset dan pemilihan sensor yang lebih presisi agar robot dapat mencapai titik

No.	Judul Penelitian	Tahun	Hasil
			stabil 0^0. Selain itu, kendali kecepatan putaran pada kedua roda juga dapat ditambahkan agar kecepatan yang dihasilkan sama antara dua roda.
4.	Penelitian oleh (Setiawan et al., 2021)dalam jurnalnya yang berjudul "Self-Balancing Robot Beroda Dua Dengan Metode PID"	2021	Dalam pembuatan Self-Balancing Robot beroda dua dengan metode PID dapat disimpulkan, bahwa dengan menggunakan nilai konstanta PID yaitu Kp=30, Ki=75, dan Kd=0,6 berhasil membuat robot dapat berdiri dengan tegak dan seimbang, dengan error kemiringan yang didapat sebesar 1,14 derajat. Serta telah dilakukan percobaan dengan memberikan gangguan dari luar berupa dorongan, seperti menekan posisi depan atau belakang robot, dan robot tetap dapat menyeimbangkan diri kembali. Pada penelitian ini juga ditambahkan fitur robot dapat berjalan dalam keadaan seimbang. Setelah dilakukan pengujian, menghasilkan kecepatan maksimum yang dapat ditangani oleh robot adalah 15,07 cm/detik.
5.	Penelitian oleh (Mohamad Junaedi, 2018) dalam	2018	Dari perancangan, pengujian dan pengamatan yang telah di lakukan

No.	Judul Penelitian	Tahun	Hasil
	jurnalnya yang berjudul "Balancing Robot Beroda Dua Dengan Menggunakan Sensor Gyroscope Berbasis Arduino Uno"		pada robot beroda dua maka dapat di ambil kesimpulan Sensor GY-521 yang di gunakan telah mampu melakukan pembacaan sudut dari sampai. Sehingga mampu memberikan keluaran yang di inginkan pada motor DC. Sistem robot beroda dua dapat seimbang yaitu mampu memperthankan posisi berdiri dan tanpa terjatuh dalam waktu rata – rata 7,25 menit dan sudut maksimal untuk mempertahankan keseimbangan yaitu antara sampai. Penerapan kontrol fuzzy pada robot beroda dua, membuat putaran dari motor DC akan bisa di atur sesuai dengan masukan dari sensor GY-521. Mengetahui perhitungan gaya pada robot (T = r.F) maka dapat di ketahui pengaruh dimensi terhadap keseimbangan robot.

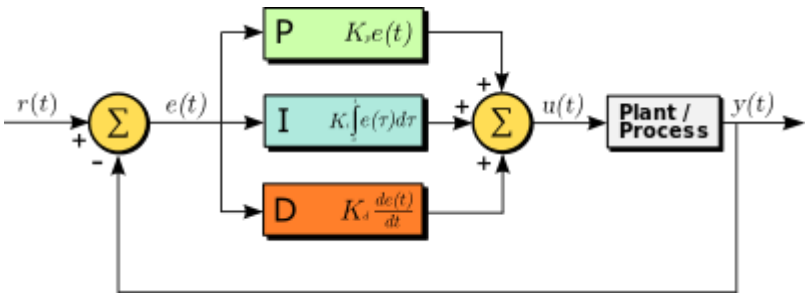
2.2. PID (Proportional, Integral, Derivative)

Kontroler PID merupakan kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri. Sebuah kontroler PID secara kontinu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur. Kontroler mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan variabel kontrol, seperti posisi keran kontrol,

damper, atau daya pada elemen pemanas, ke nilai baru yang ditentukan oleh jumlahan. Dengan Kp, Ki dan Kd, semuanya positif, menandakan koefisien untuk proporsional, integral, dan derivatif, secara berurutan (P, I, dan D). Pada model ini:

- P bertanggung jawab untuk nilai kesalahan saat ini. Contohnya, jika nilai kesalahan besar dan positif, maka keluaran kontrol juga besar dan positif.
- I bertanggung jawab untuk nilai kesalahan sebelumnya. Contoh, jika keluaran saat ini kurang besar, maka kesalahan akan terakumulasi terus menerus, dan kontroler akan merespon dengan keluaran lebih tinggi.
- D bertanggung jawab untuk kemungkinan nilai kesalahan mendatang, berdasarkan pada rate perubahan tiap waktu

  Tahapan tersebut meliputi beberapa langkah, sebagaimana yang ditunjukkan pada gambar 2.1 di bawah ini :



Gambar 2.1 Tahapan PID (Propotional, Integral, Derivative)

Karena kontroler PID hanya mengandalkan variabel proses terukur, bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan. Dengan penyesuaian (tuning) ketiga parameter model, kontroler PID dapat memenuhi kebutuhan proses. Respon kontroler dapat dijelaskan dengan bagaimana responnya terhadap kesalahan, besarnya overshoot dari setpoint, dan derajat osilasi sistem. penggunaan algoritma PID tidak menjamin kontrol optimum sistem atau bahkan kestabilannya.

PID Controller memiliki transfer function sebagai sebagai berikut:

$$H(s) = \frac{K_D s^2 + K_P s + K_I}{s^3 + K_D s^2 + K_P s + K_I}$$

PID Controller sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu P (Proportional) Controller, D (Derivative) Controller, dan I (Integral) Controller. Masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang disebut sebagai konstanta. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel di bawah ini:

Tabel 2.2 Respon PID Controller Terhadap Perubahan Konstanta

Closed-Loop Response	Rise Time	Overshoot	Settling Time	SS Error
Kp	Decrease	Increase	Small change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small change	Decrease	Decrease	Small change

Metode Ziegler-Nichols adalah salah satu metode yang umum digunakan untuk menentukan parameter tuning PID (Proporsional, Integral, Derivative). Metode ini melibatkan pengujian dan eksperimen pada sistem yang ingin dikendalikan untuk mendapatkan parameter yang optimal. Ziegler-Nichols metode osilasi penerapannya dengan memberi variasi pada nilai parameter Kp hingga mendapatkan respon sistem berupa osilasi dimana bernilai tak terhingga dan bernilai 0. Kp yang didapat dari respon sistem yang berosilasi ini bisa disebut paramater kritis (Kcr). Dari respon sistem tersebut juga didapatkan periode Kritis (Pcr). Kedua nilai ini nantinya digunakan untuk mencari nilai parameter Kp, Berikut ini adalah penalaan kontroler PID dengan tuning Ziegler Nichols telah di rumuskan secara umum yang ditunjukkan pada Tabel 2.3

Tabel 2.3 Penelaan parameter PID dengan tuning ziegler nichols

Kontroler	Kp	τ_i	τ_d
P	15	~	-
PI	13,5	0.05075	-
PID	18	0.05075	0.01269

Berdasarkan Tabel 2.3 Penalaan kontroler PID dengan tuning ziegler nichols dapat diketahui nilai perhitungan parameter PID, Kp = 0.6 Kcr = 0.5 dan Pcr =

0.125. Fungsi dari perancangan kontroler PID adalah memperbaiki respon sistem. Sehingga respon sistem mampu mencapai setpoint yang diinginkan.

2.2.1. Aksi Proporsional

Dalam kasus kontrol proporsional murni, persamaan kendali dari Persamaan (2.2) berkurang menjadi

$$u(t) = K_e(t) + u_b \cdots\cdots (2.2)$$

Aksi kontrol hanya sebanding dengan *error control*. Variabel u_b adalah bias atau reset. Ketika kesalahan kontrol $e$adalah nol, variabel kendali mengambil nilai $u(t) = u_b$. Bias u_b sering kali ditetapkan ke $(u_{max} + u_{min})/2$, tetapi kadang-kadang dapat disesuaikan secara manual sehingga kesalahan kontrol statis adalah nol pada *setpoint* yang diberikan (Åström dkk., 1995).

2.2.2. Aksi Integral

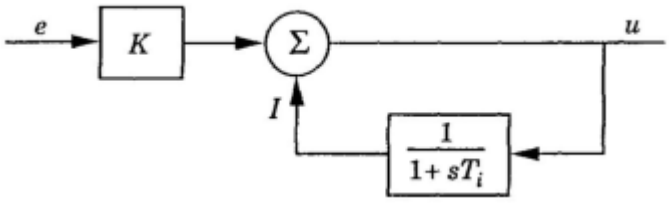
Fungsi utama dari aksi integral adalah untuk memastikan bahwa output proses sesuai dengan setpoint dalam kondisi steady state. Dengan kontrol proporsional, biasanya ada kesalahan kontrol dalam kondisi steady state. Dengan Dengan aksi integral, kesalahan positif yang kecil akan selalu mengarah pada sinyal kontrol yang meningkat, dan kesalahan negatif akan memberikan sinyal kontrol yang menurun tidak peduli seberapa kecil kesalahannya.

Argumen sederhana berikut menunjukkan bahwa kesalahan kondisi *steady state* akan selalu nol dengan aksi integral. Asumsikan bahwa sistem berada dalam kondisi *steady state* dengan sinyal kontrol konstan (u_0) dan kesalahan konstan(e_0). Dari Persamaan (2.3) dapat disimpulkan bahwa sinyal kontrol integral adalah sebagai berikut:

$$u_0 = K\left(e_0 + \frac{e_0}{T_i}t\right) \cdots\cdots (2.3)$$

Selama $e_0 \neq 0$, hal ini jelas bertentangan dengan asumsi bahwa sinyal kendali u_0 adalah konstan. Pengontrol dengan aksi integral akan selalu memberikan kesalahan kondisi tunak nol.

Aksi integral juga dapat divisualisasikan sebagai perangkat yang secara otomatis mengatur ulang istilah bias u_0, dari pengontrol proporsional. Hal ini diilustrasikan dalam diagram blok pada Gambar 2.3, yang menunjukkan kontroler proporsional dengan reset yang disesuaikan secara otomatis. Penyesuaian Penyesuaian dilakukan dengan mengumpan balik sinyal, yang merupakan nilai keluaran yang difilter, ke titik penjumlahan pengontrol. Ini sebenarnya adalah salah satu penemuan awal dari aksi integral, atau "reset otomatis".



Gambar 2.2 Diagram blok kendali Integral

2.2.3. Aksi Derivative

Tujuan dari tindakan *derivative* adalah untuk meningkatkan stabilitas loop tertutup. Karena dinamika proses, perlu beberapa waktu sebelum perubahan dalam variabel kontrol terlihat dalam output proses.

Dengan demikian, sistem kontrol akan terlambat dalam mengoreksi kesalahan. Aksi kontroler dengan aksi proporsional dan *derivative* dapat diartikan sebagai kontrol yang dibuat proporsional dengan output proses yang diprediksi, dimana prediksi dibuat dengan mengekstrapolasi kesalahan dengan garis singgung pada kurva kesalahan. Struktur dasar dari kontroler PD adalah

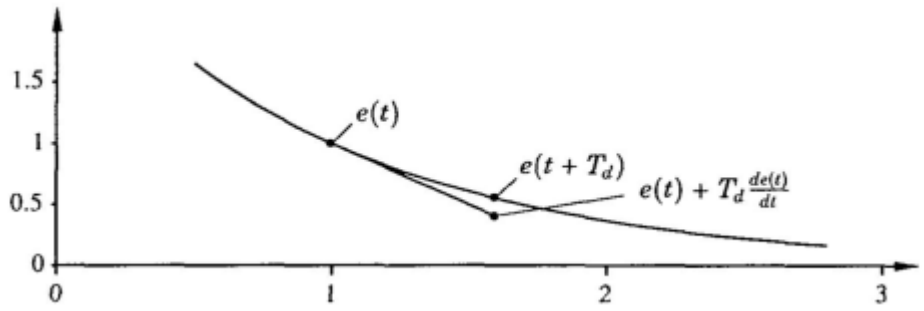
$$u(t) = K\left(e(t) + T_d \frac{de(t)}{dt}\right) \quad \text{(2.4)}$$

Pengembangan deret Taylor dari $e(t + T_d)$ memberikan

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt} \quad \text{(2.5)}$$

Sinyal kontrol dengan demikian sebanding dengan estimasi control error pada waktu T_d ke depan, di mana estimasi diperoleh dengan ekstrapolasi linier.

Sifat-sifat aksi turunan diilustrasikan pada Gambar 2.4, yang menunjukkan simulasi sistem dengan kontrol PID. Kontroler gain dan waktu integrasi dijaga konstan, $K = 3$ dan $T_i = 2$, dan waktu turunan Td diubah. Untuk Td = 0, kita memiliki kontrol PI murni. Sistem loop tertutup berosilasi dengan parameter yang dipilih. Awalnya redaman meningkat dengan bertambahnya waktu turunan, tetapi menurun lagi ketika waktu turunan menjadi terlalu besar.



Gambar 2.3 Grafik Kendali *Derivative*

2.3. NodeMCU ESP32

NodeMCU ESP32 adalah sebuah papan pengembangan (development board) yang menggunakan modul ESP32 sebagai intinya. ESP32 adalah sebuah mikrokontroler yang didesain untuk aplikasi Internet of Things (IoT) dan memiliki kemampuan Wi-Fi serta Bluetooth. NodeMCU ESP32 memungkinkan para pengembang untuk membuat berbagai proyek IoT yang terhubung ke internet.Papan pengembangan NodeMCU ESP32 menggunakan bahasa pemrograman Lua atau Arduino IDE untuk mengembangkan aplikasi yang berjalan di atasnya. Dalam penggunaan sehari-hari, NodeMCU ESP32 sering digunakan untuk membuat berbagai proyek seperti kontrol perangkat rumah tangga, pengawasan dan pengendalian jarak jauh, pemantauan lingkungan, dan banyak lagi.

Keuntungan menggunakan NodeMCU ESP32 adalah ukurannya yang kecil, konsumsi daya yang rendah, dukungan Wi-Fi dan Bluetooth yang terintegrasi, serta kemampuan pemrograman yang mudah. Hal ini membuatnya menjadi pilihan populer bagi pengembang IoT yang ingin mengembangkan prototipe atau proyek-proyek kecil dengan cepat. Dalam hal spesifikasi, NodeMCU ESP32 memiliki berbagai fitur seperti prosesor dual-core, clock speed hingga 240 MHz, RAM dan

Flash memory yang cukup besar, dukungan untuk berbagai antarmuka seperti GPIO, UART, I2C, SPI, ADC, dan banyak lagi. Papan pengembangan ini juga biasanya dilengkapi dengan USB-to-serial converter, sehingga memungkinkan untuk menghubungkannya langsung ke komputer untuk pemrograman dan debug.

Berikut adalah tabel spesifikasi dari yang dapat dilihat pada tabel 2.2 di bawah ini :

Tabel 2.4 Spesifikasi Node MCU ESP 32

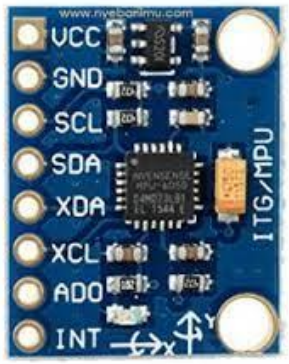
Item	**Deskripsi**
Prosessor	Xtensa Dual-Core 32 Bit LX6
Kecepatan *Clock*	240 Mhz
Tegangan Operasional	3.3 V
Flash Memori	4 Mbps
RAM	520 Kbytes
EEPROM	4096 bytes
GPIO	18 buah
Analog *Input*	18 buah
Bluetooth	Bluetooth Low Energy (BT 4.2/5.0)
USB Port	1 buah
Frekuensi *Wifi*	2.4 GHz

Gambar 2.4 NodeMCU ESP32

2.4. MPU-6050

MPU-6050 *Module* adalah sebuah modul berinti MPU-6050 yang merupakan 6 axis *Motion Processing Unit* dengan penambahan regulator tegangan dan beberapa komponen pelengkap lainnya yang membuat modul ini siap dipakai dengan tegangan supply sebesar 3-5VDC. Pada Gambar 2 terdapat modul MPU-

6050. modul ini memiliki interface I2C yang dapat disambungkan langsung ke MCU yang memiliki fasilitas I2C. Sensor MPU6050 berisi sebuah MEMS Accelerometer dan sebuah MEMS *Gyro* yang saling terintegrasi. Sensor ini sangat akurat dengan fasilitas hardware internal 16 bit ADC untuk setiap kanalnya. Sensor ini akan menangkap nilai kanal axis X, Y dan Z bersamaan dalam satu waktu. Dapat di lihat pada gambar 2.2 di bawah ini:



Gambar 2.5 MPU-6050

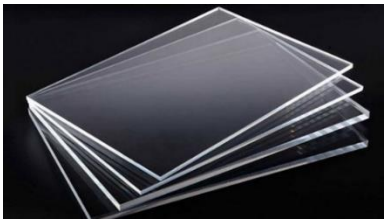
Berikut adalah tabel spesifikasi dari MPU-6050 yang dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2.5 Spesifikasi MPU-6050

Parameter	**Specification**
Chip	MPU-6050
Vin	3V-5V
Serial Interfaces Supported	I2C
Gyroscopes ranges	*+/- 250 500 1000 2000 degree/sec*
Acceleration ranges	*+/- 2g, +/- 4g, +/- 8g, +/- 16g*
Pin Spacing	*2.54mm (0.1in*)
Built-in 16bit AD converter	16 bit data Output

2.5. Akrilik

Akrilik adalah suatu bahan plastik polimer yang menyerupai kaca sehingga sering dipakai menjadi subtitusi kaca. Akrilik memiliki kejernihan yang lebih besar serta ukurannya juga lebih ringan daripada kaca. Dapat di lihat pada gambar 2.3 di bawah ini :



Gambar 2.6 Akrilik

2.6. Spencer Baut

Spancer Baut adalah baut yang batang besinya memanjang untuk menyambungkan ke spancer baut lainnya. Dapat di lihat pada gambar 2.4 di bawah ini :

Gambar 2.7 Spencer Baut

2.7. Motor Driver L298N

Motor drive adalah Salah satu part mesin produksi sebagai motor penggerak yang berfungsi untuk menggerakkan sebuah motor. Driver motor L298N merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC. Dapat di lihat pada gambar 2.5 di bawah ini:

Gambar 2.8 Motor Drive

2.8. Baterai

Baterai merupakan alat listrik-kimiawi yang menyimpan energi serta mengeluarkan tenaganya dalam bentuk listrik. Dapat di lihat pada gambar 2.6 di bawah ini:

Gambar 2.9 Baterai

2.9. Motor DC

Motor Listrik DC/DC Motor adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik yang berupa putaran atau gerak. Dapat di lihat pada gambar 2.7 di bawah ini:

Gambar 2.10 Motor Listrik DC Motor

2.10. Kabel Jumper

Kabel Jumper merupakan kabel elektrik yang mempunyai pin konektor di setiap ujungnya dan memungkinkan untuk menghubungkan dua komponen yang melibatkan Arduino tanpa memerlukan solder. Intinya, kegunaan kabel jumper ini digunakan sebagai konduktor listrik untuk menyambungkan rangkaian listrik. Dapat di lihat pada gambar 2.8 di bawah ini:

Gambar 2.11 Kabel Jumper

2.11. Kabel Data

Kabel Data adalah perangkat keras yang memungkinkan perangkat komputer dengan gawai terhubung sehingga dapat melakukan transfer data. Dapat di lihat pada gambar 2.9 di bawah ini:

Gambar 2.12 Kabel Data