

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini akan membahas secara umum mengenai penerapan teknologi yang digunakan pada penelitian ini, baik teori perangkat keras, perangkat elektronik maupun perangkat lunak yang digunakan dalam perancangan robot tersebut. Dengan referensi dari penelitian sebelumnya, metode serta teori yang disampaikan akan merujuk pada definisi secara umum yang berkaitan dengan penelitian ini.

#### **2.1 Perangkat Keras**

Dalam merancang sebuah sistem kontrol PID pada mesin bensin ini dibutuhkan perangkat keras yang terdiri atas bagian mekanik berupa fisik sistem kontrol pada mesin bensin. Di samping itu dibutuhkan pula bagian elektronik berupa sensor-sensor serta komponen penunjang lainnya.

##### **2.1.1 Mesin Bensin**

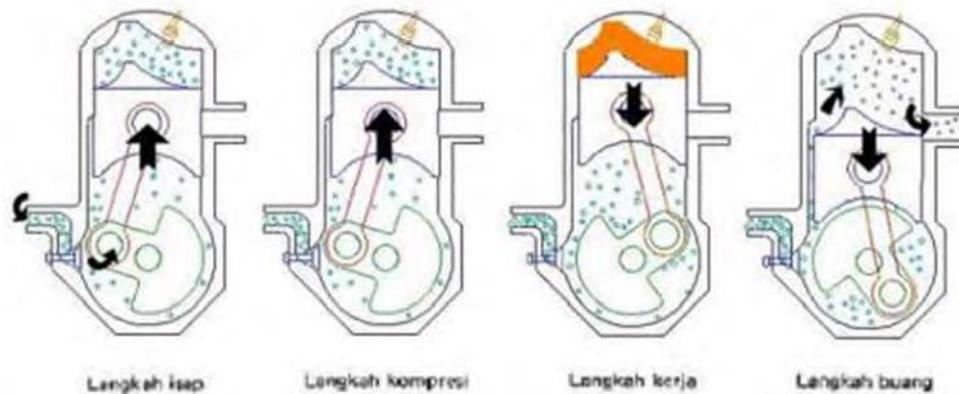
Motor bakar bensin ialah salah satu tipe motor bakar yang memakai bensin selaku bahan bakarnya. Bensin itu sendiri ialah bahan bakar yang gampang dibakar serta gampang menguap. Kecepatan pembakaran umumnya berkisar antara 15- 20 meter/ detik, temperatur hawa bertambah sampai 1500oC (1773 K) dan tekanannya menggapai kisaran 30- 40 kilogram/ cm<sup>2</sup> (0, 03- 0, 04 N/ m<sup>2</sup>). Motor bakar bensin banyak digunakan di bidang otomotif. Dengan kata lain, banyak digunakan selaku penggerak untuk bermacam tipe kendaraan. Motor bensin, biasanya digolongkan ke dalam 2 tipe, ialah motor bensin 2 langkah serta motor bensin 4 langkah [7].

##### **2.1.1.1 Prinsip Kerja Mesin Bensin**

Motor bakar bekerja dengan gerakan piston bolak- balik (*translasi*). Motor bensin bekerja bagi prinsip 4 langkah serta 2 langkah. Yang diartikan dengan sebutan ‘langkah’ di mari biasa kita tahu dengan istilah “Tak” merupakan perjalanan piston dari titik mati atas ke titik mati dasar ataupun kebalikannya. Energi ataupun tenaga yang menggerakkan piston tersebut diperoleh dari tekanan gas hasil pembakaran bahan bakar serta hawa. Setelah itu gerakan bolak- balik dari piston diganti ke dalam wujud gerak berputar (rotasi) oleh poros engkol. Gerak putaran poros engkol inilah yang menggambarkan *output* dari motor bakar bensin,

yang kemudian bisa dimanfaatkan guna bermacam keperluan mekanis, salah satunya guna menggerakkan kendaraan [8].

Pada dasarnya prinsip kerja motor bensin merupakan sama, proses *intake*, *compression*, *power*, *exhaust* cuma saja pada motor bensin dua langkah seluruh proses cuma dicoba dalam dua langkah (*upward* serta *downward*) piston [8]. Skema gerakan piston pada motor bensin dua langkah bisa dilihat pada Gambar 2.1.



Gambar 2. 1 Skema Gerakan Piston 2 Langkah

a) Langkah Pertama (*Upward Stroke*):

Ialah langkah kompresi, di mana piston bergerak ke atas, kombinasi bahan bakar udara dimampatkan serta setelah itu terbakar pada dikala piston sudah menggapai titik mati atas. Perihal ini menyebabkan penyusutan tekanan (*vacuum*) pada *crankcase*, sehingga kombinasi bahan bakar udara dari karburator mengalir ke dalam *crankcase* [8].

b) Langkah Kedua (*Downward Stroke*):

Ialah langkah usaha, piston didorong ke dasar oleh tekanan pembakaran, kombinasi bahan bakar hawa di dalam *crankcase* dikompresikan apabila piston menutup lubang pemasukan [8].

c) Langkah Ketiga (Pembilasan)

Pembilasan berlangsung apabila piston melewati titik mati dasar, kombinasi bahan bakar hawa hendak mengalir dari *crankcase* lewat saluran bilas ke dalam silinder menghasilkan gas sisa pembakaran [8].

### 2.1.1.2 Komponen Mesin Bensin

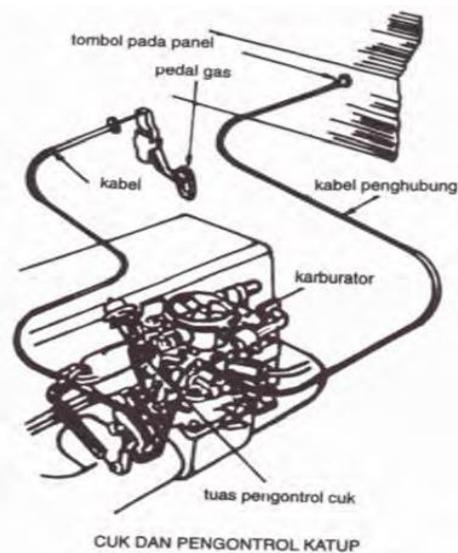
#### a) Karburator

Karburator berperan selaku tempat bercampurnya bahan bakar bensin dengan udara jadi wujud kombinasi semacam kabut, serta lebih gampang terbakar. Karburator dipasang pada saluran hisap. Suatu gasket digunakan buat menghindari hawa keluar dari salurannya serta blok penyekat umumnya dipasang buat menyekat ataupun membatasi perpindahan panas dari mesin ke karburator [8].

#### b) Tuas Pengontrol Kecepatan

Kabel ataupun kawat *throttle* berperan selaku pengontrol kecepatan dari mesin dengan menempatkan jumlah kombinasi udara serta bensin yang dimasukkan ke silinder mesin.

Dalam kendaraan, kabel *throttle* akan tersambung dengan pedal gas, pedal gas ini dihubungkan ke karburator dengan tuas serta batang penarik ataupun kabel. Sepanjang pedal ditekan, tuas membuka katup percepatan dalam karburator [8].

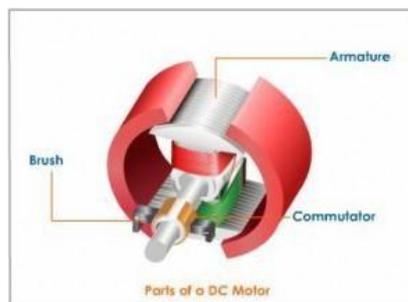


Gambar 2. 2 Skema Komponen Pada Mesin Bensin

Kebutuhan hendak sistem kontrol pada mesin bensin dikala ini sangat dibutuhkan, mengingat masih terdapatnya keperluan menimpa pengontrolan mesin bensin. Guna saat ini riset mengenai pengontrolan pada mesin bensin tidak sebanyak pengontrolan pada motor listrik.

### 2.1.2 Motor Listrik DC

Motor listrik DC ataupun motor arus searah ialah salah satu mesin listrik yang mengganti tenaga listrik jadi tenaga mekanik. Motor arus searah banyak sekali digunakan dalam produk elektronik yang kerap kita jumpai tiap hari, semacam pada mainan kanak-kanak, pemutar kaset, kipas pendingin pc, serta lain-lain. Gerak ataupun putaran yang dihasilkan oleh motor arus searah diperoleh dari interaksi 2 buah medan yang dihasilkan oleh bagian ‘jangkar’ (*armature*) serta bagian ‘medan’ (*field*) dari motor arus searah. Pada riset ini, motor listrik DC hendak digunakan guna menarik kabel *throttle* [9]. Bagian motor arus DC ditunjukkan pada Gambar 2.3.



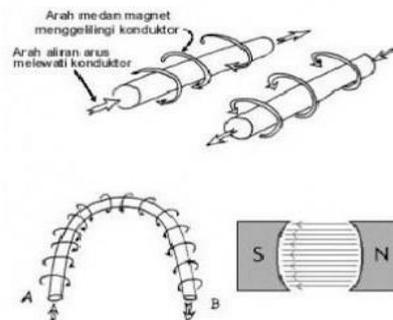
Gambar 2. 3 Bagian Motor Listrik DC

Pada Gambar 2.3, bagian medan berupa sesuatu kumparan yang tersambung ke sumber arus searah. Sebaliknya bagian jangkar diarahkan selaku magnet permanen (U- S), bagian jangkar ini tidak wajib berupa magnet permanen, dapat pula berupa lilitan yang hendak jadi elektromagnet apabila memperoleh sumber arus searah [9].

Bagian lain yang tidak kalah berarti pada motor arus searah merupakan terdapatnya ‘komutator’ (*comutator*). Keuntungan dari motor DC ini merupakan bisa melindungi pasokan energi dengan metode mengatur kecepatan. Pengendalian ini bisa dicoba dengan metode mengganti tegangan dinamo, apabila dinaikkan maka akan menaikkan kecepatan, sebaliknya apabila diturunkan maka akan mengurangi kecepatan. Ada pula metode yang lain ialah mengganti arus medan, peningkatan arus medan sebanding dengan peningkatan kecepatan [9].

### 2.1.2.1 Prinsip Kerja Motor Listrik DC

Bila arus melewati suatu batang konduktor, hingga akan muncul medan magnet di dekat batang konduktor. Prinsip kerja motor DC dimana medan magnet yang membawa arus mengelilingi konduktor, ditunjukkan pada Gambar 2.4.

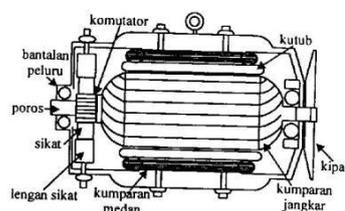


Gambar 2. 4 Prinsip Kerja Motor Listrik DC

Pada Gambar 2.4, menampilkan medan magnet yang tercipta di dekat konduktor berganti arah karena bentuk U. Medan magnet cuma terjalin di dekat suatu konduktor bila terdapat arus mengalir pada konduktor tersebut. Jika konduktor berupa U (angker dinamo) diletakkan diantara kutub utara serta selatan yang kokoh medan magnet konduktor hendak berhubungan dengan medan magnet kutub [9].

### 2.1.2.2 Komponen Utama Motor Listrik DC

Motor DC digunakan pada pemakaian khusus, di mana dibutuhkan torsi yang besar ataupun percepatan yang pas untuk kisaran kecepatan yang luas [9]. Komponen motor DC ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Komponen Motor Listrik DC

Motor DC mempunyai tiga komponen utama:

a) Kutub Medan

Motor listrik DC mempunyai kutub medan yang stasioner serta dinamo yang menggerakkan *bearing* pada ruang di antara kutub medan. Motor DC sederhana mempunyai 2 kutub medan yaitu kutub utara serta kutub selatan.

b) Dinamo

Apabila arus masuk mengarah dinamo, hingga arus ini hendak jadi elektromagnet. Dinamo yang berupa silinder, dihubungkan ke as penggerak guna menggerakkan beban.

c) Komutator

Komutator bermanfaat guna membalik arah arus listrik dari dalam dinamo. Komutator juga membantu dalam transmisi arus antara dinamo serta sumber daya.

### 2.1.2.3 PWM (*Pulse Width Modulation*)

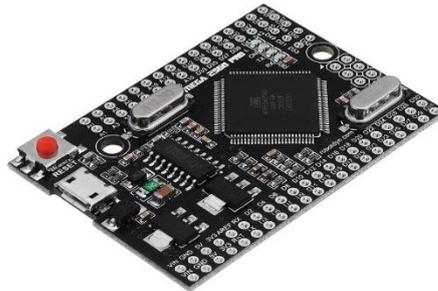
Modulasi lebar pulsa (PWM) biasanya ialah tata cara memanipulasi lebar sinyal yang diwakili oleh pulsa dalam satu siklus guna memperoleh tegangan rata-rata yang berbeda. Sebagian contoh aplikasi PWM merupakan modulasi informasi yang digunakan guna mengendalikan daya ataupun tegangan jadi beban, pengatur tegangan, serta aplikasi yang lain. Aplikasi PWM berbasis *mikrokontroler* umumnya berbentuk kontrol kecepatan motor DC serta kontrol motor servo dan lainnya.

PWM merupakan teknologi yang mendapatkan sinyal analog dari perangkat digital. Sinyal PWM umumnya mempunyai amplitudo serta frekuensi fundamental yang tetap, namun lebar pulasanya berbeda. Lebar pulsa PWM sebanding dengan amplitudo sinyal asli yang tidak termodulasi. Dengan kata lain, sinyal PWM mempunyai frekuensi gelombang yang tetap, namun *duty cycle*-nya bermacam-macam (antara 0% serta 100%) [10].

### 2.1.3 Arduino Mega 2560

Arduino Mega 2560 merupakan papan pengembangan *mikrokontroler* berbasis arduino, memakai *chip* ATmega2560. *Board* ini mempunyai *pin* I/O yang lumayan banyak, 54 *pin* I/O digital( 15 diantaranya merupakan PWM), 16 *pin* input

*analog*, serta 4 *pin* UART( *serial hardware*). Arduino Mega 2560 dilengkapi dengan osilator 16 Mhz, *port* USB, konektor listrik DC, konektor ICSP serta tombol reset. *Board* sangat lengkap serta telah mempunyai seluruh yang diperlukan untuk *mikrokontroller*. Penggunaannya cukup mudah, cukup sambungkan catu daya dari USB ke PC. Papan arduino mega 2560 ditunjukkan pada Gambar 2.6.



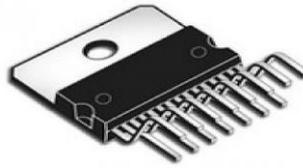
Gambar 2. 6 Papan Arduino Mega

*Chip* ATmega2560 pada arduino mega 2560 mempunyai memori 256 KB, di mana 8 KB digunakan buat *bootloader*. Kapasitas SRAM merupakan 8KB, EEPROM merupakan 4KB, serta perpustakaan EEPROM bisa digunakan untuk membaca serta menulis selama pemrograman.

Arduino Mega 2560 mempunyai jumlah *pin* paling banyak di seluruh papan pengembangan arduino. Mega 2560 mempunyai 54 *pin digital*, yang bisa digunakan sebagai 23 *input* ataupun *output*. Arduino Mega 2560 mempunyai bermacam sarana untuk berkomunikasi dengan PC serta arduino lain ataupun *mikrokontroller* yang lain. *Chip* Atmega 2560 menyediakan komunikasi serial UART TTL (5V) pada pin 0 (RX) serta pin 1 (TX) [11].

#### 2.1.4 Driver Motor

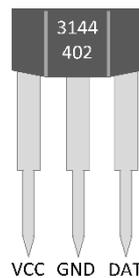
Pada IC L298 terdapat rangkaian *H-Bridge* transistor NPN. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari output *mikrokontroller*. Transistor digunakan sebagai *switching* yang berfungsi untuk mengatur arah putaran motor [12]. L298N dapat mengontrol 2 buah motor dc, tegangan yang dapat dikeluarkannya bisa mencapai tegangan 46 Vdc dan arus mencapai 2 A untuk setiap kanalnya *Driver* motor dapat ditunjukkan pada Gambar 2.7.



Gambar 2. 7 IC Driver Motor L298N

### 2.1.5 Sensor Magnet

Sensor *Hall-Effect* dirancang untuk merasakan adanya objek magnetis dengan perubahan posisinya. Perubahan medan magnet yang terus menerus menyebabkan timbulnya pulsa yang kemudian dapat ditentukan frekuensinya, sensor jenis ini biasa digunakan sebagai pengukur kecepatan. Sensor *hall effect* digunakan untuk mendeteksi kedekatan (*proximity*), kehadiran atau ketidakhadiran suatu objek magnetis yang menggunakan suatu jarak kritis. Sensor ini mempunyai tipe *on-off* untuk digunakan sebagai *limit switch*, sensor keberadaan (*presence sensors*) [13]. Sensor medan magnet dapat ditunjukkan pada Gambar 2.8.



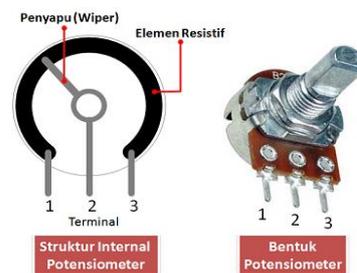
Gambar 2. 8 Sensor Medan Magnet

Sensor ini memiliki tiga buah *pin*, dan membutuhkan catu daya antara 4,5V sampai 6V. Apabila tidak terdapat medan magnet di dekatnya, maka *output* dari sensor ini akan menghasilkan tegangan setengahnya dari tegangan catu daya [13].

### 2.1.6 Potensiometer

Potensiometer merupakan resistor 3 *pin* dengan sambungan geser yang membentuk pembagi tegangan bisa disetel. Bila cuma 2 *pin* yang digunakan salah satu *pin* tetap dan *pin* geser, potensiometer berperan selaku resistor variabel ataupun *rheostat*. Potensiometer biasanya digunakan guna mengatur piranti elektronika seperti pengendali suara pada penguat. Potensiometer yang dioperasikan oleh sesuatu mekanisme bisa digunakan sebagai transduser, misalnya selaku sensor *joystick*. Potensiometer tidak sering digunakan untuk mengatur energi besar (lebih dari 1 watt) secara langsung. Potensiometer digunakan guna menyetel taraf isyarat

analog (misalnya pengendali suara pada piranti audio), serta sebagai pengendali masukan buat sirkuit elektronik. Sebagai contoh, sebuah peredup lampu memakai potensiometer buat mengendalikan pensaklaran suatu TRIAC, jadi secara tidak langsung mengatur kecerahan lampu. Komponen potensiometer adalah sensor analog yang paling sederhana namun sangat berguna untuk mendeteksi posisi putaran, misalnya kedudukan sudut poros *aktuator* berdasarkan nilai resistansi pada putaran porosnya [14]. Struktur internal potensiometer dan juga bentuknya ditunjukkan pada Gambar 2.9.



Gambar 2. 9 Struktur Internal Potensiometer dan Bentuknya

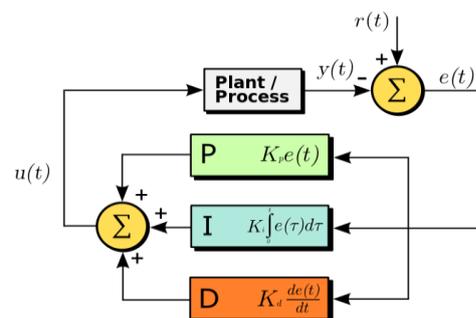
## 2.2 Perangkat Lunak

Perangkat lunak digunakan untuk mengontrol seluruh perangkat keras yang telah dirancang dan terintegrasi. Perangkat lunak ini berupa algoritma gerak dan tugas *aktuator* dalam bentuk *listing* program yang ditanamkan di dalam *mikrokontroller*. Program tersebut dapat berbentuk bermacam-macam sesuai dengan spesifikasi dan *mikrokontroller* yang digunakan.

### 2.2.1 Kontrol PID (*Proportional – Integral – Derivative*)

Kontrol PID merupakan gabungan dari kendali *proporsional*, *integral* dan *derivatif*. Maksud dari penggabungan tersebut karena apabila masing-masing pengendali berdiri sendiri, ketika terdapat kekurangan pada pengendali, maka pengendali tersebut tidak mampu untuk memperbaiki kekurangan tersebut. Oleh karena itu, penggabungan ketiga elemen kendali akan saling mengisi setiap kekurangan yang dihasilkan. Keunggulan dari kendali PID ini didasarkan pada masing-masing kendali P, I dan D yaitu untuk mempercepat reaksi sistem, menghilangkan *offset*, dan membuat sistem menjadi lebih stabil dengan cara meredam osilasi [15].

PID controller merupakan controller yang berarti yang kerap digunakan dalam industri. Pengontrol PID akan memberikan aksi berdasarkan *error* yang telah diperoleh, *error* disini merupakan perbedaan dari *set point* yang telah ditentukan. Keluaran dari *kontroller* PID adalah hasil penjumlahan dari ketiga komponen tersebut. Karakteristik *kontroller* PID ini dipengaruhi oleh ketiga komponen P, I, dan D, penyetelan nilai setiap konstanta  $K_p$ ,  $K_i$ , dan  $K_d$  akan mengakibatkan penonjolan sifat dari masing-masing komponen. Ketiganya bisa dipakai bersamaan ataupun sendiri-sendiri tergantung dari reaksi yang kita mau terhadap sesuatu *plant*. Pada perancangan sistem kontrol PID, biasanya dilakukan dengan metode *trial & error*. Hal ini disebabkan karena parameter  $K_p$ ,  $K_i$  dan  $K_d$  tidak *independent*, untuk mendapatkan aksi kontrol yang baik diperlukan langkah *trial & error* dengan kombinasi antara P, I dan D sampai ditemukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  seperti yang diharapkan [16]. Skema sistem kontrol PID dapat ditunjukkan pada Gambar 2.10.



Gambar 2. 10 Skema Kontrol PID

Masukan *kontroller* ke *plant* menciptakan *output kontroller*, bersumber pada hasil pada domain waktu dari *feedback* yang berbentuk nilai *error*. Rumus umum PID diformulasikan dalam persamaan dibawah ini:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Keterangan:

$u(t)$  = Output dari pengontrol PID

$e(t)$  = Error (selisih antara *set point* dengan kecepatan)

$K_p$  = Konstanta *Proportional*

$K_i$  = Konstanta *Integral*

$K_d$  = Konstanta *Derivative*

### 2.2.1.1 Kontrol *Proportional*

Pemakaian kontrol P mempunyai bermacam keterbatasan sebab sifat kontrol yang tidak dinamik ini. Meski demikian dalam aplikasi-aplikasi dasar yang simpel kontrol P ini lumayan sanggup untuk memperbaiki reaksi *transien* khususnya *rise time* serta *settling time*. Kekurangan dari pengendalian *proportional* merupakan munculnya *overshoot*, di mana besarnya *overshoot* bergantung pada besarnya gain elemen- elemen di dalam *loop* dan *Proportional Band* (PB). Perihal ini diakibatkan oleh watak dasar pengendali *proportional* yang memerlukan *error* yang menciptakan *output*. Pengontrol *proportional* mempunyai keluaran yang sebanding/ sepadan dengan besarnya sinyal kesalahan (selisih antara besaran yang di idamkan dengan harga aktualnya) [3].

### 2.2.1.2 Kontrol *Integral*

Pengontrol *Integral* berperan menciptakan reaksi sistem yang mempunyai kesalahan kondisi mantap nol (*Error Steady State*= 0). Pengontrol *Integral* pula sanggup melenyapkan *overshoot* yang dicoba oleh proporsional. Bila suatu pengontrol tidak mempunyai faktor *integrator*, pengontrol *proportional* tidak sanggup menjamin keluaran sistem dengan kesalahan kondisi mantapnya nol. Kontrol I bisa membetulkan sekaligus melenyapkan reaksi *steady-state*, tetapi pemilihan  $K_i$  yang tidak pas bisa menimbulkan reaksi transien yang besar sehingga bisa menimbulkan ketidakstabilan sistem. Pemilihan  $K_i$  yang sangat besar malah bisa menimbulkan output berosilasi sebab menaikkan orde sistem [3].

Keluaran pengontrol adalah hasil penjumlahan terus menerus dari perubahan masukannya. Jika sinyal *error* tidak berubah, maka keluaran akan mempertahankan keadaan sebelum masukan berubah.

### 2.2.1.3 Kontrol *Derivative*

Sebab lambatnya kontrol pengendali PI dalam pengendalian sistem. Upaya membetulkan tanggapan didapat dengan memakai unit kontrol *differential* ataupun *derivative*, disingkat D. *Output* pengendali D ialah *differential* dari guna *input*. Sayangnya faktor D tidak bisa menghasilkan *output* (apabila tidak terdapat pergantian *input*). Sebab sifat ini, pengendali D tidak sempat dipakai sendirian. *Unit* pengendali D senantiasa dipakai dalam kombinasinya dengan P serta I, jadi pengendali PD ataupun PID. Tidak hanya itu, pengendali D tidak bisa dipakai buat

proses *variable* yang memiliki *noise*. Sebab banyaknya hambatan dalam pengendali D, populasi pengendali PID serta PD jadi tidak sebanyak pengendali P ataupun PI [17].

### 2.2.2 *Tuning PID Metode Trial and Error*

Dalam menerapkan pengontrol PID pada suatu *plant*, perlu adanya nilai yang harus dimasukkan pada masing-masing pengontrol  $K_p$ ,  $K_i$  dan  $K_d$ . Nilai yang dimasukkan pada pengontrol harus pas, agar pengontrol PID dapat bekerja dengan baik. Salah satu cara yang paling sering digunakan dalam menentukan nilai parameter PID yaitu dengan menggunakan metode *trial and error*. Metode *trial and error* adalah metode mencari nilai parameter dengan mencoba-coba suatu nilai tertentu hingga didapat sebuah performansi kontrol PID yang terbaik. Metode *trial and error* ini tidak memerlukan perhitungan matematis dan kompleks, namun metode *tuning* ini mempunyai kekurangan yaitu sangat memerlukan banyak waktu dan tenaga karena perlu mencoba memasukkan banyak nilai sampai didapatkan hasil respon pengontrol PID yang terbaik [2].

Meskipun mencari nilai parameter PID menggunakan metode *trial and error* tidak memerlukan perhitungan secara matematis, tetapi perlu dilakukan juga dengan memperhatikan setiap karakteristik masing-masing parameter. Karakteristik untuk setiap parameter  $K_p$ ,  $K_i$  dan  $K_d$  dapat dilihat pada Tabel 2.1 berikut ini.

Tabel 2. 1 Karakteristik Masing-masing Parameter

<i>Response Parameter</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Error Steady State</i>
<i>Proportional</i>	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan
<i>Integral</i>	Menurunkan	Meningkatkan	Meningkatkan	Menghilangkan
<i>Derivative</i>	Perubahan Kecil	Menurunkan	Menurunkan	Perubahan Kecil