# BAB II

# DASAR TEORI

Dasar teori ini memuat tentang teori, hasil pemikiran dan penelitian terkait dengan quadcopter. Pembahasan pada bab ini meliputi : sekilas tentang aeromodeling, sekilas tentang quadcopter, perangkat yang digunakan, dan sistem kendali *fuzzy*

## 2.1 Sekilas Tentang Aeromodeling

Pada dunia aeromodeling, pesawat tanpa awak (*Unmanned Aerial Vehicle / UAV )* merupakan miniatur pesawat udara yang bisa di terbangkan secara *manual* maupun secara otomatis menggunakan *Remote Control* (RC). Pesawat udara aerodinamis merupakan pesawat udara yang lebih berat dari udara. Pesawat udara aerodinamis terdiri dari 2 kelompok yaitu pesawat yang bermotor dan pesawat yang tidak bermotorPesawat udara bermotor terdiri dari bersayap tetap (*Fixed wing)* dan sayap putar (*rotary wing)*. Pesawat udara udara aerodinamis bermotor bersayap tetap terdiri dari pesawat terbang dan amphibians. Pesawat udara bermotor bersayap putar terdiri dari *helicopter* dan *gyrocopter*. Pesawat udara aerodinamis tidak bermotor terdiri dari pesawat luncur (glider), pesawat layang dan layang-layang

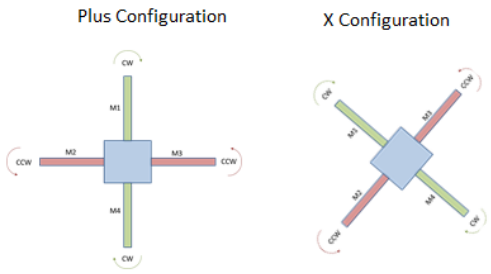



**Gambar 2.1** Gyrocopter dan Glider

## 2.2 Sekilas Tentang *Quadcopter*

*Quadcopter* adalah jenis pesawat udara aerodinamis dan bersayap putar dan salah satu jenis dari multirotor. Sebagaimana dengan namanya *Quad* yang berarti empat, sehingga pada *multirotor* ini memiliki empat buah motor *brushless* dan 4 buah baling baling.

Berdasarkan bentuk rangka (*frame*), quadcopter terdiri dari 2 model yaitu quadcopter dengan model plus (+) dan bentuk silang (x)

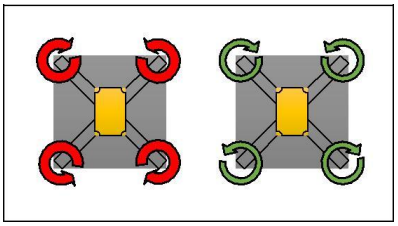



**Gambar 2.2** model quadcopter plus dan bentuk silang

## 2.3 Prinsip Kerja Q*uadcopter*

Pada ujung setiap *frame* terpasang 4 buah motor *brushless* dan *propeller*, motor *brushless* yang akan memutar *propeller*, dengan mengatur kecepatan dan mengatur arah putaran kedalam, sehingga menghilangkan gaya sentrifugal disekitar quadcopter mengakibatkan terangkat dan ber-*manuver*.

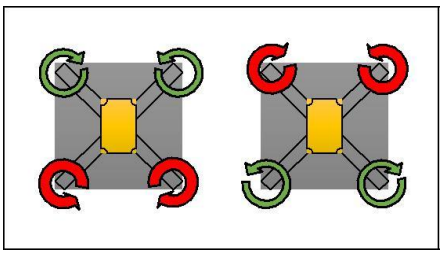
Ada 4 pengaturan dasar dasar kecepatan putaran *propeller* yaitu pengaturan *throttle propeller, pitch propeller, yaw propeller* dan *roll propeller. Throtller propeller* bertujuan untuk melakukan *vertical take off* dan *landing (VTOL)* bergerak ke atas dan ke bawah



**Gambar 2.3** *Throttle Propeller*

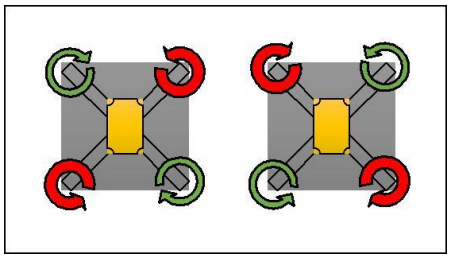
Putaran yang tebal menandakan *propeller* berputar dengan cepat yang mengakibatkan quadcopter akan terangkat (*take-off*), sedangkan putaran yang tipis menandakan *propeller* berputar dengan lambat yang mengakibatkan quadcopter akan turun/mendarat (*landing*).

Pengaturan *pitch propeller* adalah pengaturan yang diberikan agar quadcopter bergerak maju dan bergerak mundur. Berikut gambar yang menampilkan ilustrasi kecepatan putar *propeller* untuk pengaturan *pitch*.



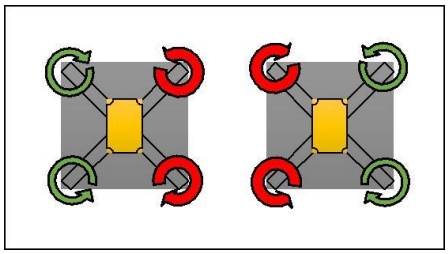
**Gambar 2.4** *Pitch propeller*

Pengaturan *yaw control* bertujuan agar quadcopter bergerak berputar kekiri dan bergerak ke kanan. Ilustrasinya dapat dilihat pada gambar berikut ini.



**Gambar 2.5** *Yaw control*

Pengaturan *roll propeller* diberikan agar *quadcopter* bergerak kekiri dan kekanan. Ilustrasi pengaturan *roll propeller* dapat dilihat pada gambar 4 di bawah ini.



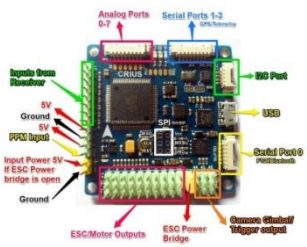
**Gambar 2.6** *Roll propeller*

## 2.4 Komponen *Quadcopter* yang digunakan

Komponen yang di gunakan ada 2 bagian yaitu bagian mekanik dan bagian elektronika. Bagian mekanik terdiri dari *frame* dan *propeller*, sedangkan bagian elektronika terdiri dari *Flight controller, Electronic Speed Controller (ESC),*catu daya dan *motor brushless*

### 2.4.1 Flight Controller

*Flight Controller* adalah sebuah mikroprosesor yang berfungsi sebagai pengatur kecepatan dan penerima sinyal dari *Remote Control. Flight Controller* menerima sinyal masukan dari *telementry* atau *Remote Control* yang berupa nilai kecepatan motor sesuai dengan *stick* ,lalu *Flight Controller* akan mengirimkan sinyal pulsa dari pin output yang akan di terima oleh *ESC.*




**Gambar 2.7** *Flight Controller CRIUS AIOP*

### 2.4.2 Electronic Speed Controller (ESC)

*ESC* adalah rangkaian elektronik yang berfungsi sebagai pengatur kecepatan putaran motor pada pesawat atau helikopter dengan cara menterjemahkan sinyal yang diterima *receiver* dari *transmitter*.




**Gambar 2.8** *Electronic Speed Controller*

### 2.4.3 Catu daya

Catu daya adalah sebuah piranti elektronika yang berguna sebagai sumber daya untuk piranti lain, terutama daya listrik. Pada dasarnya pencatu daya bukanlah sebuah alat yang menghasilkan energi listrik saja. Catu daya yang digunakan berjenis Lithium Polymer (LiPo) yang kapasitansi sebesar 5000 mAh dan memiliki 3s




**Gambar 2.9** Catu daya LiPo

### 2.4.4 Motor Brushless

Motor dc tanpa sikat atau disebut Brushless DC Motor. Brushless DC Motor adalah suatu jenis motor-sinkron. Artinya medan magnet yang dihasilkan oleh

stator dan medan magnet yang dihasilkan oleh rotor berputar di frekuensi yang sama. BLDC motor tidak mengalami Slip, tidak seperti yang terjadi pada motor induksi biasa. Motor jenis ini mempunyai permanen magnet pada bagian "rotor" sedangkan elektro-magnet pada bagian "stator"-nya. Setelah itu, dengan menggunakan sebuah rangkaian sederhana (simpel computer system), maka kita dapat merubah arus di eletro-magnet ketika bagian "rotor"-nya berputar. Motor brushless yang digunakan memiliki daya sebesar 980 kV



**Gambar 2.10** *Motor Brushless*

### 2.4.5 Sensor *Accelerometer*

Sebuah tranduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan Kendaraan, bangunan, mesin, dan juga bisa digunakan untuk mengukur getaran yang terjadi di dalam bumi, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.

Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut juga percepatan (*acceleration*). Jika kecepatan semakin berkurang dari pada kecepatan sebelumnya, disebut *deceleration*. Percepatan juga bergantung pada arah / orientasi karena merupakan penurunan kecepatan yang merupakan besaran *vector*. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula.

### 2.4.6 Sensor *Gyroscope*

Device yang berguna untuk menentukan orientasi gerak yang berotasi dengan cepat pada poros sumbu. *Gyroscope* memiliki output yang peka terhadap kecepatan sudut dari arah sumbu x yang nantinya akan menjadi sudut phi (*roll*), dari sumbu y nantinya menjadi sudut *theta* (*pitch*), dan sumbu z nantinya menjadi sudut *psi* (*yaw*).

Penggunaan *gyroscope* dalam kehidupan sehari-hari belum banyak. *Gyroscope* lazimnya digunakan pada pesawat terbang, kapal, *helicopter*, dll. Hal tersebut untuk mengurangi getaran yang ditimbulkan mesin agar keseimbangan tidak goyah. *Gyroscope* baru-baru ini juga diterapkan pada mainan *helicopter* dengan tujuan menyeimbangkan saat diterbangkan.

## 2.5 Logika Fuzzy

Teori himpunan logika fuzzy dikembangkan oleh Prof. Lofti Zadeh padatahun 1965. Ia berpendapat bahwa logika benar dan salah dari logika Boolean tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Untuk mengatasi masalah gradasi yang tidak terhingga tersebut, Zadeh mengembangkan sebuah himpunan fuzzy.

Tidak seperti logika Boolean, logika fuzzy mempunyai nilai yang kontinu. Fuzzy dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh karena sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang masukan *(input)* ke dalam suatu ruang keluaran *(output)*.

### 2.5.1 Himpunan Fuzzy

Pada himpunan tegas (crisp), nilai keanggotaaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan μA [x], memiliki dua kemungkinan, yaitu:

- satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Pada penggunaan himpunan crisp, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Himpunan fuzzy digunakan untuk mengantisispasi hal tersebut. Seberapa besar ekstensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya. Kalau pada himpunan crisp, nilai keanggotaan hanya ada dua kemungkinan, yaitu 0 atau 1, pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. apabila nilai keanggotaan fuzzy μ A[x] = 0 berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan μA[x] = 1 berarti x menjadi anggota penuh himpunan A. Himpunan fuzzy memiliki dua atribut, yaitu :

1. linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA.
2. numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti 40, 25, 30.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu.

1. Variabel fuzzy

   Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperatur, permintaan.

2. Himpunan fuzzy

   Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Contoh: Variabel umur, terbagi menjadi 3 himpunan fuzzy, yaitu: muda, parobaya, tua.

3. Semesta pembicaraan

   Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Contoh: Semesta pembicaraan untuk variabel umur: [0 + ~].

4. Domain

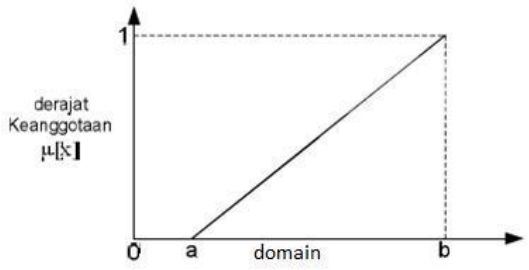
   Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Contoh: MUDA = [0, 45] ,TUA = [45, + ~].

### 2.5.2 Fungsi Keanggotaan

Fungsi keanggotaan *(membership function)* adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan). Ada beberapa fungsi yang digunakan pada perancangan ini, antara lain.

1. Representasi Linier

Pada pemetaan linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Ada dua keadaaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Gambar 2.11 dibawah ini menunjukan grafik dari kurva linier naik.

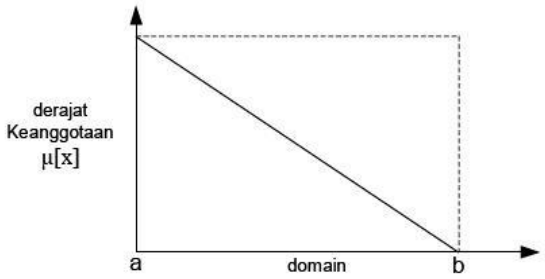



**Gambar 2.11** Kurva Linear Naik

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} \dfrac{0}{x-a} & ;x \leq a \\ \dfrac{x-a}{b-a} & ;a \leq x \leq b \\ 1 & ;x \geq b \end{cases} (pers.1)$$

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Gambar 2.12 dibawah ini menunjukan grafik dari kurva linier naik.



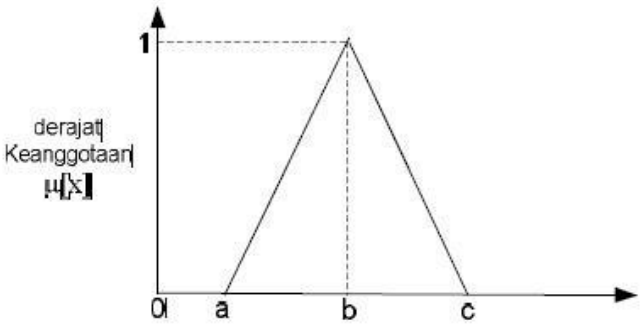

**Gambar 2.12** Kurva Linear Turun

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0 & ; x \geq b \\ \dfrac{b-x}{b-a} & ; a \leq x \leq b \ (pers.2) \\ 1 & ; x \leq a \end{cases}$$

2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis linear.Gambar 2.3 dibawah ini menunjukan grafik dari kurva segitiga.




**Gambar 2.13** Kurva Segitiga

Fungsi Keanggotaan :

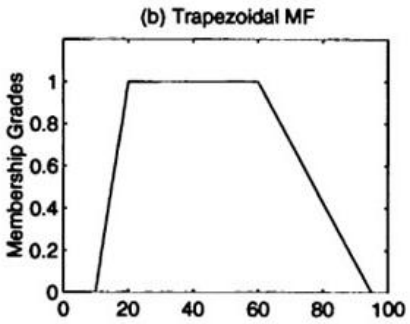
$$\mathbf{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \dfrac{x-a}{b-a}, & a \leq x \leq b. \\ \dfrac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \quad \text{(pers.3)}$$

$$\text{triangle}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad \text{(pers.4)}$$

Untuk menentukan nilai minimum dan maximum fungsi keanggotaan pada representasi kurva segitiga, dapat menggunakan persamaan 4. Parameter {a,b,c} dengan (a < b < c) menentukan sudut x dari tiga sudut penting dari fungsi keanggotaan.

3. Representasi kurva trapezoid

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan satu. Gambar 2.14 dibawah ini menunjukan grafik dari kurva trapesium.




**Gambar 2.14** kurva trapezoid

Fungsi keanggotaan :

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \le a. \\ \frac{x-a}{b-a}, & a \le x \le b. \\ 1, & b \le x \le c. \\ \frac{d-x}{d-c}, & c \le x \le d. \\ 0, & d \le x. \end{cases} \quad \text{(pers.5)}$$

$$\text{trapezoid}(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right). \quad \text{(pers.6)}$$

Untuk menentukan nilai minimum dan maximum fungsi keanggotaan pada representasi kurva trapesium, dapat menggunakan persamaan 5. Parameter {a,b,c,d} dengan (a < b < c < d) menentukan sudut x dari empat sudut penting dari fungsi keanggotaan.

### 2.5.3 Operator Fuzzy

Ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari 2 operasi disebut α-predikat. Ada tiga operator dasar yaitu.

1. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. A predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antara elemen pada himpunan-himpunan yang bersangkutan.

$\mu_{AnB} = \text{Min} (\mu_A [x], \mu_B [y]$……(pers. 7)

2. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α - predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antara elemen pada himpunan-himpunan yang bersangkutan.

$\mu_{AuB} = \text{Max} (\mu_A [x], \mu_B [y])$ ……..……(pers. 8)

3. Operator NOT

Operator ini berhubungan dengan operasi komplemen himpunan. α – predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan dengan 1.

$\mu_N = 1-\mu_A [x]$ ……..(pers. 9)

### 2.5.4 Fuzzy Inference System

Salah satu metode untuk penalaran *(Fuzzy Inference System)* adalah Metode Mamdani (Max-Min). Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan empat tahap. Dapat digambarkan dengan block dibawah ini.

1. Pembentukan himpunan Fuzzy

Pada Metode Mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

2. Fuzzyfikasi *(Fuzzification)*

Proses memetakan nilai tegas input *(crisp)* kedalam himpunan Fuzzy. Hasil dari proses ini berupa fuzzy input.

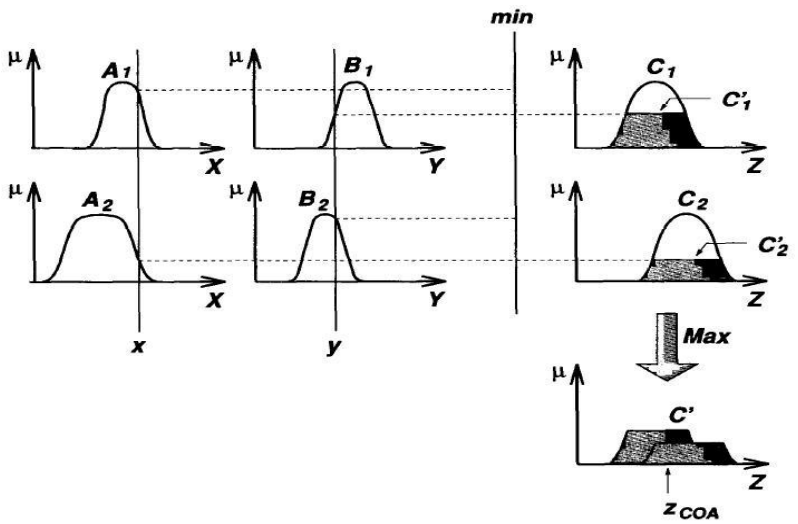
3. Rule evaluasi *(Rule evaluation)*

Proses melakukan penalaran terhadap fuzzy input yang dihasilkan oleh proses *fuzzification* berdasarkan aturan fuzzy yang telah dibuat. Proses ini menghasilkan fuzzy output. Fungsi Implikasi yang digunakan adalah Min.

*4.* Penegasan *(Defuzzification)*

Input dari proses penegasan *(defuzzification)* adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output. Ada banyak metode penegasan *(defuzzification)* yang dapat digunakan untuk aturan Mamdani, salah satunya dengan mengunakan metode *centroid of area.*

$$\boldsymbol{COA} = \frac{\int \boldsymbol{z\mu A(z)z\,dz}}{\int \boldsymbol{z\mu A(z)z\,dz}} \text{ (pers. 10)}$$

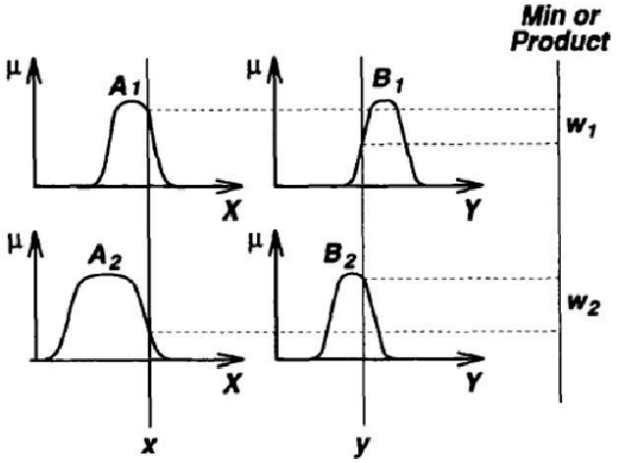



**Gambar 2.15** *Fuzzy Inference System* (Mamdani)

Salah satu alternatif metode untuk penalaran *(Fuzzy Inference System)* adalah dengan menggunakan metode penalaran fuzzy metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

Pada metode Sugeno, fuzzifikasi, operasi fuzzy, dan implikasi sama seperti metode Mamdani, perbedaannya hanya pada agregasi dan defuzzifikasi. Jika pada metode Mamdani agregasi berupa daerah di bawah kurva, maka pada metode Sugeno agregasi berupa singleton. Defuzifikasi pada metode sugeno lebih sederhana, karena hanya menghitung titik rata-rata.

$$\boldsymbol{Z} = \frac{\boldsymbol{w1z1 + w2z2}}{\boldsymbol{w1 + w2}} \text{ (pers. 11)}$$




**Gambar 2.16** *Fuzzy Inference System* (Sugeno)

### 2.6 Kendali PID

Menurut sebuah survey, bahwa 97% industri yang bergerak dibidang proses menggunakan kendali PID dalam melakukan pengontrolannya. Luasnya penggunaan kendali PID pada dasarnya dilatar belakangi beberapa hal diantaranya:

- Kesederhanaan struktur kendali. Hanya mempunyai 3 parameter utama yang perlu diatur
- Kendali PID memiliki sejarah panjang. PID telah digunakan jauh sebelum era digital berkembang (1930an)
- Kendali PID dalam banyak kasus telah terbukti menghasilkan unjuk kerja yang relative memuaskan baik digunakan sebagai sistem regulator maupun sebagai sistem servo.

Pada awalnya kendali PID umumnya diimplementasikan dengan menggunakan rangkaian elektronika *analog*. Bahkan banyak diantaranya direalisasikan dalam komponen mekanis dan pneumatic murni.

Seiring dengan berkembangnya dunia *digital* (terutama *mikroprosesor* dan *mikrokendaliler*) maka algoritma kendali PID dapat direalisasikan kedalam bentuk persamaan PID digital. Yang jika diimplementasikan hanya berupa sebuah program saja yang ditanamkan kedalam *embedded system* (*mikroprosessor* atau *mikrokontroller*).

Berikut beberapa penjelasan singkat tentang PID :

PID adalah singkatan dari (*proposional integral derivative*) yang merupakan rumus / perhitungan yang memiliki fungsi menghitung nilai error antara variable proses dengan set point yang di inginkan dan mampu memperkecil nilai error tersebut sehingga output yang dihasilkan semakin mendekati nilai yang di inginkan. Perhitungan PID sendiri memiliki rumus.

$$\boldsymbol{U\,(t) = K_P e(t) + K_i \int_0^t e\,(t)dt + K_d \frac{d}{dt} e(t)} \quad \text{.....(12)}$$

Dengan :

u(t) = sinyal output pengendali PID

Kp = *konstanta proposional*

Ti = waktu *integral*

Td = waktu *derivative*

Ki = *konstanta integral* (Kp/Ti)

Kd = *konstanta derivative* (Kp.Td)

E(t) = sinyal error = referensi – keluaran plant = set point – nilai sensor

Selain itu juga dapat menggunakan bagian-bagian rumus PID secara terpisah (dapat menggunakan Kp / Ki / Kd saja atau pun kombinasinya) tergantung kebutuhan kita. Disamping menggunakan rumus diatas untuk mencari nilai nya dari Kp, Ki, dan Kd kita juga dapat menentukan dengan *tuning manual* / *trial*

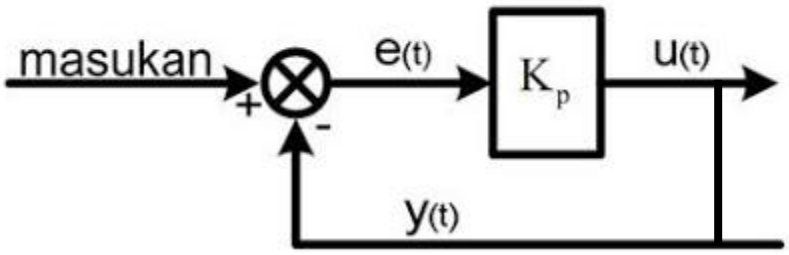
*error* (memasukan nilai Kp, Ki, dan Kd berdasarkan hasil output yang didapat), tetapi dalam menggunakan metode ini karakteristik Kp, Ki, dan Kd perlu diperhatikan sehingga kita tidak sembarangan dalam melakukan *tuning* tersebut.

### 2.6.1 Kendali Proporsional

Kendali *proposional* berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal error), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara input kendalier u(t) dengan sinyal error e(t) terlihat pada persamaan 13.

$$\boldsymbol{u(t) = K_p\, e(t)} \quad \text{.....(13)}$$

Kp adalah *konstanta proposional*. Diagram blok kendali *proposional* ditunjukan pada gambar 2.17.



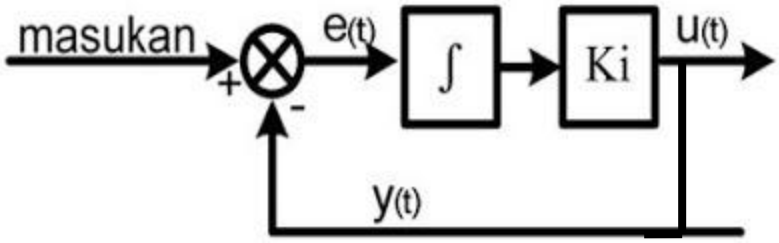

**Gambar 2.17.** Diagram Blok Kendali *Proposional*

### 2.6.2 Kendali Intergral

Kendali *integral* pada prinsipnya bertujuan untuk menghilangkan kesalahan pada keadaan stabil (offset) yang biasanya dihasilkan oleh kendali *proposional*. Hubungan antara output kendali *integral* u(t) dengan sinyal error e(t) terlihat pada persamaan 14.

$$\boldsymbol{u(t) = K_i \int_0^t e\,(t)\, dt} \quad \text{.....(14)}$$

Ki adalah *konstanta integral*. Diagram blok kendali *integral* ditunjukan pada gambar 2.18.



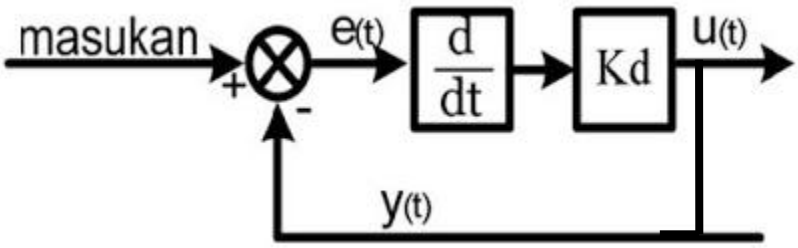

**Gambar 2.18.** Diagram Blok Kendali *Integral*

### 2.6.3 Kendali *Derivative*

Kendali *derivative* dapat disebut pengendali laju, karena output kendali sebanding dengan laju perubahan sinyal error. Hubungan antara output kendali derivative u(t) dengan sinyal error e(t) terlihat pada persamaan 15.

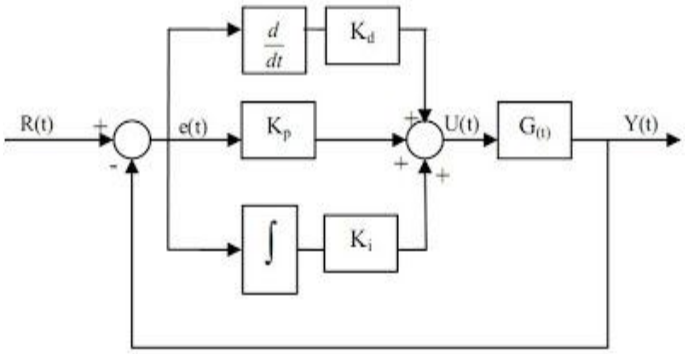
$$\boldsymbol{u(t) = k_d \frac{de(t)}{dt}} \quad \text{.....(15)}$$

Blok kendali *derivative* ditunjukan pada gambar 2.19. Kendali *derivative* tidak akan pernah digunakan sendirian, karena kendalier ini hanya akan aktif pada periode peralihan. Pada periode peralihan, kendali *derivative* menyebabkan adanya redaman pada sistem sehingga lebih memperkecil lonjakan. Seperti pada kendali *proposional*, kendali *derivative* juga tidak dapat menghilangkan offset.




**Gambar 2.19**. Diagram Blok Kendali *Derivative*

Gabungan dari ketiga kendali tersebut menjadi kendali PID. Diagram blok dari kendali PID ditunjukan pada gambar 2.20.




**Gambar 2.20**. Diagram Blok Kendali PID.

Tabel 2.1 Efek Setiap Kendali (Kp, Ki, Kd)

| **Parameter** | **Waktu naik (*Rise time*)** | ***Overshoot*** | ***Setting time*** | ***Steady-state error*** |
|---|---|---|---|---|
| Kp | Menurun | Meningkat | Perubahan Kecil | Menurun |
| Ki | Menurun | Meningkat | Meningkat | Menghilangkan |
| Kd | Perubahan Kecil | Menurun | Menurun | Tidak berpengaruh |