BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini penulis menggunakan beberapa tinjauan pustaka yang dapat mendukung penelitian, berikut ini adalah tinjauan pustaka yang digunakan dapat dilihat pada Tabel 2.1 :

Tabel 2.1 Tinjauan Pustaka

Nomer Literatur	Penulis	Tahun	Judul
Literatur 1	Lasmadi, Adha Cahyadi, Risnauri Hidayat	2016	Implementasi Kalman Filter Untuk Navigasi Quadrotor Berbasis Sensor Accelerometer
Literatur 2	Rudi Setiawan, Hendri Himawan Triharmnto, Muhammad Fahrurozi	2021	*Gesture Control* Menggunakan IMU MPU 6050 Metode *Kalman Filter* Sebagai Kendali *Quadcopter*
Literatur 3	Rosalia H. Subrata, Raymond Tarumasely, Calvin Dwianto S.	2017	Perancangan Pengendali PID Untuk Gerakan *Pitch* dan *Roll* Pada *Quadcopter*

Literatur 4	Muhammad Rafiq, Freddy Kurniawan, Ndaru Atmi Purnami	2021	Koreksi Sudut *Attitude* Dan *Heading Quadrotor* Dengan Perubahan Matriks Kovarian Derau Pengukuran *Kalman Filter*
Literatur 5	Lasmadi, Muhammad Ali Dakir, Freddy Kurniawan, Sudarmanto	2023	Koreksi Sudut *Attitude Quadrotor* Pada Saat Dinamis Dengan Mengubah Bobot Data Sensor Pada *Kalman Filter*

2.1.1 Tinjauan Pustaka Literatur 1

Penelitian yang dilakukan oleh Lasmadi, Dkk (2016), yaitu bertujuanuntuk mengimplementasikan *kalman filter* agar dapat menapis derau dari data sensor *accelerometer* yang digunakan untuk sistem navigasi *quadrotor*. Penelitian ini dilakukan dalam 3 percobaan simulasi yang dimana *kalman filter* mampu memberikan performa yang baik dalam menapis derau proses maupun derau pengukuran. *Kalman filter* dapat diterapkan untuk sistem navigasi *quadrotor* berbasis *inersia,accelerometer* dalam menentukan posisi *quadrotor*. Pada penelitian yang dilakukan Lasmadi, dkk, *filter* menjadi tidak berarti karena adanya *derau offset* yang cukup besar. Deviasi standar galat pengukuran adalah 100 m, galat kovarian sebelum pemfilteran adalah 3.2043e+03 m, sedangkan galat kovarian setelah pemfilteran adalah 1.2251e+03 m. Pengujian yang dilakukan penulis menerapkan keseluruhan data yang tidak ada di penelitian Lasmadi, dkk dengan cara menggunakan sensor MPU 6050 dalam mencari *gyroscope, pitch, dan roll*.

2.1.2 Tinjauan Pustaka Literatur 2

Penelitian yang dilakukan oleh Setiawan, Dkk (2021), yaitu bertujuan untuk merancang dan mengimplementasikan *Inertial Measurement Unit* (IMU) MPU 6050 dengan *gesture control* menggunakan *Kalman filter* sebagai metode nya. Penelitian ini berpusat pada *gesture* tangan sebagai pokok permasalahannya untuk mengendalikaan *drone*, komunikasi sistem *wireless*, dan meningkatkan akurasi dari sensor MPU 6050. Peneitian pada sistem komunikasi data masih terjadi penundaan waktu *delay* 25.661667 ms sehingga diperlukan penelitian yang memperbaiki waktu tunda pengendalian agar hasil data yang masih menyebar tidak *konvergen* dan terjadi *drifting* pada *initial value*. Penelitian penulis dan penilitian yang dilakukan Setiawan, Dkk terdapat persamaan, yaitu sama-sama menggunakan sensor MPU 6050 dalam mencari nilai *gyroscope* dan *accelerometer*, sedangkan untuk *software* terdapat perbedaan yaitu pada aplikasi *python* dan aplikasi *putty*.

2.1.3 Tinjauan Pustaka Literatur 3

Penelitian yang dilakukan oleh Subrata, Dkk (2017) Rosalia H. Subrata, Dkk (2017), yaitu bertujuan untuk merancang pengendali PID untuk gerakan *pitch* dan *roll* pada *quadcopter*. Pada penelitian tersebut bertujuan untuk membuat dan merancang pengendali PID tanpa menggunakan *kalman filter*. Penelitian tersebut mampu membaca kondisi sudut *pitch* dan *roll* tetapi penelitian tersebut tidak mampu meningkatkan *estimasi* keadaan sistem dan mengompensasi ketidakpastian serta gangguan yang mungkin terjadi dalam sistem. Pada penelitian yang dilakukan Subrata, dkk dengan *koefisien* PID Kp = 0.0316, Ki = 0,05 dan Kd = 0,26 untuk mengendalikan Gerakan *pitch* dan *roll*, sistem pengendali *quadcopter* mampu meredam *osilasi* dan menghasilkan *error rate* ±2 derajat dari *input*. *Koefisien* pengendali PID yang dirancang tanpa *filter* hanya mampu mengembalikan kondisi sudut *pitch* dan *roll* Ketika diberi gangguan dari luar pada detik ke-3,5 hingga 4,5.

2.1.4 Tinjauan Pustaka Literatur 4

Penelitian yang dilakukan oleh Rafiq, Dkk (2021), yaitu bertujuan untuk mengkoreksi sudut *attitude* dan *heading quadrotor* menggunakan *Kalman filter* untuk mendapatkan attitude berupa sudut *orientasi roll, pitch, dan yaw* yang lebih akurat baik pada saat *statis* maupun *dinamis*. Pada penilitan yang dilakukan Rafiq, dkk diperoleh hasil orientasi pada *kalman filter* tanpa adanya perubahan nilai R masih memiliki galat yang cukup tinggi yaitu sebesar 9,31% untuk sikap *roll*, galat sebesar 0,2% untuk sikap *pitch* dan galat sebesar 26,42% untuk sikap *yaw*. Sedangkan nilai orientasi *Kalman filter* dengan adanya perubahan nilai R dapat menurunkan galat tiap sumbu menjadi 0% untuk sikap *roll*, galat sebesar 0,11% untuk sikap *pitch* dan galat sebesar 26,39% untuk sikap *yaw.*

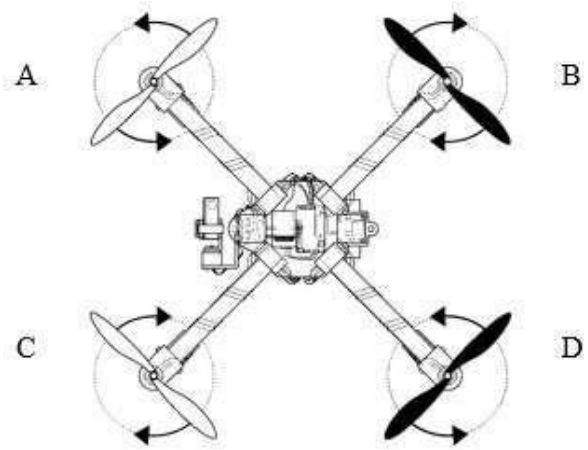
2.1.5 Tinjauan Pustaka Literatur 5

Penelitian ini dilakukan oleh Lasmadi, Dkk (2023) yaitu bertujuan untuk mengkoreksi sudut *attitude quadrotor* dinamis dengan mengubah bobot data sensor menggunakan *kalman filter*. Pada penelitian ini perubahan nilai bobot data dari kedua sensor ini dilakukan dengan mengubah nilai *matriks kovarian* derau pengukuran. Bertujuan untuk mendapatkan nilai sudut *orientasi* yang lebih akurat untuk berbagai macam kondisi. Pada penelitian yang dilakukan Lasmadi, dkk, *kalman filter* dapat digunakan dalam menentukan sudut *attitude* pada *quadrotor* dengan cara melakukan *data fusion* terhadap *gyroscope* dan *accelerometer* dengan perubahan bobot data sensor pada saat dinamis, sudut *roll* dapat diturunkan dari 9,3052% menjadi 0,0635%, dan sudut *pitch* dapat diturunkan dari 0,0021 % menjadi 0,0015 %. Namun masih terdapat kendala yaitu sudut *attitude* tidak selamanya akurat, sama seperti penelitian penulis yaitu *kalman filter* dapat digunakan pada *quadcopter* namun data cenderung lambat daripada PID.

2.2 Quadcopter

Quadcopter adalah pesawat terbang tak berawak yang mempunyai potensiuntuk lepas landas, terbang *manuver*, dan mendarat bahkan di daerah kecil. Seiring dengan perkembangan teknologi *modern*, saat ini *quadcopter* banyak digunakan sebagai pengawasan suatu wilayah, pengambilan foto/video, dan pengintaian jarak jauh. *Quadcopter* mempunyai empat buah motor di setiap sudutnya, pada setiap motornya terdapat baling-baling (*propeller*) yang membuat aliran udara sehingga akan menghasilkan tekanan kearah bawah yang akan menyebabkan *quadcopter* menghasilkan gaya angkat. Setiap motor dan baling-baling pada *quadcopter* mempunyai fungsi dalam menghasilkan gaya dorong dan torsi, empat buah motor tersebut dipasang menyilang.

Motor depan dan belakang memiliki arah putaran searah jarum jam, sebaliknya jika motor kanan dan kiri diputar berlawan dengan arah jarum jam. Hasilnya akan menuju pada gaya angkat untuk *quadcopter*, gaya angkat tersebut dapat menghasilkan *quadcopter* terbang di udara. Pada *quadcopter* terdapat tiga variable sudut yang menjadi struktur utama dalam pengendaliannya. *Roll*, *Pitch*, dan *Yaw* adalah ketiga sudut yang menjadi struktur penting dalam pengendalian *quadcopter*.



Gambar 2.1 Desain *quadcopter*

(**Sumber** : Fabiana Meijon Fadul, 2019)

Berikut adalah gerak dasar pada *quadcopter* saat melakukan manuver diudara, diantaranya :

a) Thrust

Thrust adalah gaya yang membuat *quadcopter* bergerak searah sumbu Z vertikal. Untuk memicu gaya *thrust*, diperlukan kecepatan keempat motor harus sama dan secara bersamaan motor tersebut ditambah kecepatannya agarmenimbulkan gaya angkat lebih besar dan *quadcopter* akan bergerak naik keatas begitupun sebaliknya.

b) Pitch

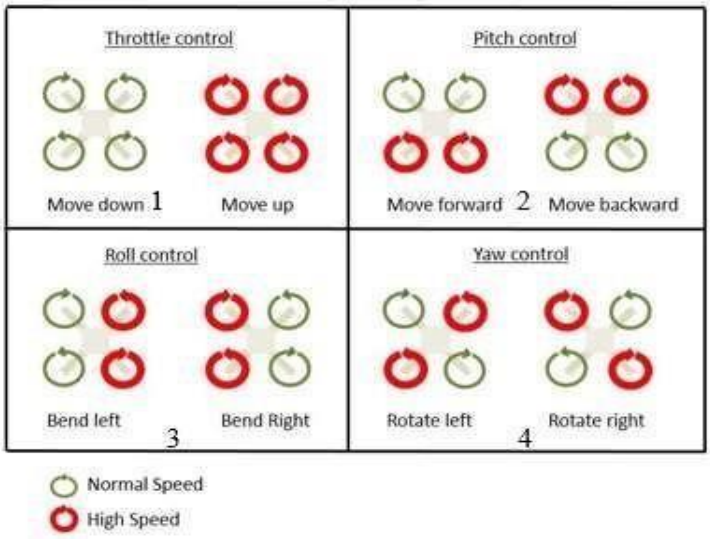
Pitch adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu Y dari bodinya. Hal ini terjadi karena dua motor berputar lebih cepat dari dua motor lainnya. Misalnya jika *quadcopter* ingin melakukan *pitch* kebelakang maka dua motor bagian depan akan berputar lebih cepat dari dua motor bagian belakang begitupun sebaliknya.

c) Roll

Roll adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu X dari bodinya. Hal ini terjadi karena dua motor berputar lebih cepat dari dua motor lainnya. Misalnya jika *quadcopter* ingin melakukan *roll* kearahkiri maka dua motor sebelah kanan akan berputar lebih cepat dari dua motor sebelah kiri begitupun sebaliknya.

d) Yaw

Yaw adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu Z dari bodinya. Pergerakan ini dipengaruhi oleh perubahan kecepatan dari keempat motor. Jika kecepatan motor depan dan belakang diperlambat dankecepatan motor kanan dan kiri dipercepat maka *quadcopter* akan bergerak menyimpang kearah kiri begitupun sebaliknya.



Gambar 2.2 Gerakan *quadcopter* berdasarkan kecepatan setiap motornya (**Sumber** : Fabiana Meijon Fadul, 2019)

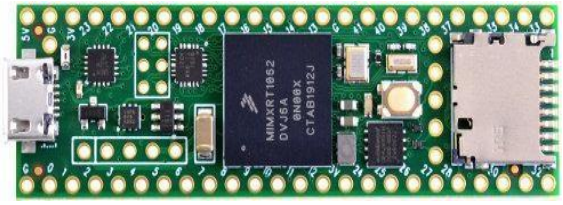
2.3 Perangkat Keras Quadcopter

Perangkat keras *quadcopter* melibatkan seluruh komponen fisik yang membentuk struktur dan sistem operasional dari drone atau *quadcopter*. Ini mencakup berbagai elemen seperti rangka, motor, baterai, kontroler penerbangan, sensor, dan komponen elektronik lainnya. Berikut adalah beberapa komponen perangkat keras utama yang terlibat dalam pembentukan dan operasi quadcopter :

2.3.1 Microcontroller Teensy

Mikrokontroler Teensy bisa digunakan dalam berbagai aplikasi, termasuk dalam dunia drone atau *quadcopter*. Pada *quadcopter*, Teensy bisa berfungsi sebagai otak pengendali atau papan kontrol yangmengontrol berbagai aspek operasi drone. Teensy dapat digunakan untuk

mengimplementasikan algoritma kontrol seperti PID (Proporsional, Integral, Derivatif) untuk menjaga kestabilan dan kontrol penerbangan quadcopter. Ini melibatkan pengolahan data dari sensor seperti *gyroscope* dan *akselerometer*, serta menghasilkan sinyal kontrol untuk motor-motor *quadcopter*.



Gambar 2.3 Microcontroller

2.3.2 Motor Brushless

Motor Brushless adalah komponen utama sebagai penggerak *quadcopter*, karena dengan putaran motor ini *quadcopter* dapat terbang. Pemilihan motor harus sebanding dengan kebutuhan karena beda jenis motor beda fungsi yang diberikan motor tersebut. Motor pada *quadcopter* menggunakan KV=RPM/Volt, ukuran KV berbanding lurus dengan kecepatan putaran motor (RPM). Jika nilai KV rendah maka RPM yang dihasilkan rendah serta untuk torsi atau daya angkat (*Throttle*) yang besar dan begitupun sebaliknya lilitan pada roto menentukan jumlah torsi yang diberikan, apabila semakin banyak lilitan pada rotor maka torsi semakin besar dan untuk RPM kecil begitu juga sebaliknya.

Gambar 2.4 Motor brushless

(**Sumber** : Febiana Meijon Fadul, 2019)

2.3.3 Propeller

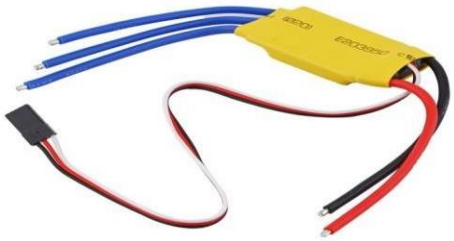
Propeller di dalam *quadcopter* termasuk dalam rotary wing atau sayap putar. Cara kerja dari alat ini yaitu mengubah putaran menjadi gayadorong untuk bergerak. Terdapat 2 jenis propeller yang dibagi berdasarkanarah putaran dan arah hembusan udara yaitu *Clockwise* (CW) dan *Counter Clockwise* (CCW). Dan material yang digunakan berbagai macam mulai dari plastic, carbon, kayu dan lain-lain.

Gambar 2.5 Propeller

(**Sumber** : Tokopedia)

2.3.4 Electronic Speed Controller

ESC adalah sebuah *device* yang bertugas sebagai *driver* motor brushless yang di mana mengatur kecepatan dan arah putaran. Selain itu tugas ESC yaitu mengubah tegangan DC ke AC 3 fasa untuk diteruskan ke motor. ESC ini terhubung langsung ke *battery* dan *flight controller* melalui kabel yang terdiri dari *signal* dan *ground*, dan komponen ini sangat berguna untuk motor agar tidak timbul kerusakan. Tipe dan besaranampere pada ESC ini bermacam-macam, paling rendah terdiri dari 12A sampai ratusan Ampere tergantung dari kebutuhan, biasanya menyesuaikan besaran motor yang digunakan. Tipe ESC diantaranya ada dua, pertama adalah ESC opto dimana tidak mempunyai output tegangan 5v sebagai tambahan, dan terakhir ESC UBEC dimana ESC ini mempunyai outputan tegangan sebesar 5v dan biasa digunakan sebagai power source.



Gambar 2.6 Electronic speed controller

(**Sumber** : Tokopedia)

2.3.5 Frame

Frame merupakan bagian rangka atau tubuh pada *quadcopter* yang menjadi tulang atau penyokong semua komponen *quadcopter* serta tempat ditempelnya semua komponen-komponen sehingga dengan adanya *frame* ini komponen dapat terintegrasi dengan baik.

Gambar 2.7 Frame *quadcopter*

(**Sumber** : Tokopedia)

2.3.6 Battery

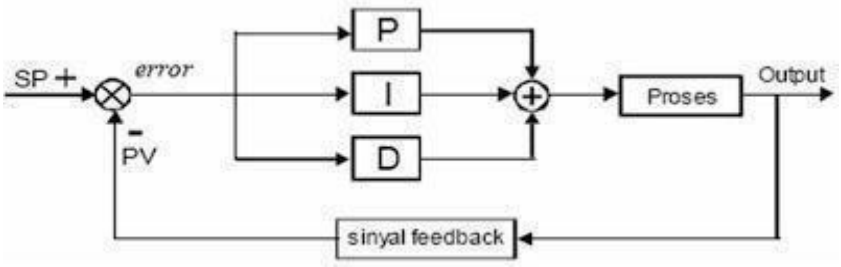
Battery merupakan sumber daya utama untuk *quadcopter*, sehingga penggunaan *battery* ini harus sesuai dengan kebutuhan, dengan melalui analisis dan perhitungan yang sudah dilakukan sebelumnya. Agar didapat hasil yang optimal saat lepas landas dengan mencapai waktu terbang maksimal. Jenis yang digunakan yaitu jenis LiPo atau *Lithium Polymer*.

Gambar 2.8 Battery *quadcopter*

(**Sumber** : Tokopedia)

2.4 Proporsional Integral Derivative (PID)

Sistem kontrol PID (*Proporsional Integral Derivative*) merupakan kontroler untuk mengetahui presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (feedback) pada sistem tersebut. Sistem kendali PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), I (*Integral*), D (*Derivative*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun digabung. Blok diagram sistem kendali PID ditunjukan pada gambar 2.9



Gambar 2.9 Blok diagram sistem kendali PID

Dengan parameter Kp, Ki, Kd yang tepat maka sistem akan memiliki respon yang sangat cepat dalam mencapai set pointnya, menghilangkan offset,mempercepat kondisi dan eliminasi error steady state. Efek dari setiap pengontrol baik itu kontrol *proporsional*, kontrol *integral* maupun kontrol *derivative* pada sistem loop tertutup disimpulkan pada Table 2.2 berikut ini :

Tabel 2.2 Tabel parameter PID

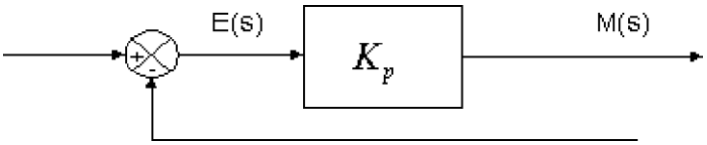
Karakteristik	Parameter	Deskripsi
Rise Time	Kp	Meningkatkan untuk mempercepat *respons*
	Ki	Meningkatkan untuk mempercepat penyesuaian dan mengurangi *overshoot*
	Kd	Meningkatkan untuk mengurangi *overshoot* dan penurunan *overshoot*
Overshoot	Kp	Meningkatkan untuk mempercepat *respons* dan mungkin meningkatkan *overshoot*
	Ki	Meningkatkan untuk mengurangi *overshoot*
	Kd	Meningkatkan untuk mengurangi *overshoot* dan penurunan *overshoot*
Settling Time	Kp	Meningkatkan untuk mempercepat *respons* dan penurunan *overshoot*
	Ki	Meningkatkan untuk mempercepat penyesuaian dan penurunan *overshoot*
	Kd	Meningkatkan untuk mempercepat penurunan *overshoot* dan penurunan *overshoot*

Steady State Error	Kp	Meningkatkan untuk mengurangi *steady state error*
	Ki	Meningkatkan untuk mengurangi *steady state error* dan penurunan *overshoot*
	Kd	Tidak berlaku untuk tujuan ini

2.4.1 Kontrol Proporsional

Kontrol *Proporsional* adalah kontroler yang memiliki output sebanding atau *proporsional* dengan input errornya, secara sederhana bahwa *output controller proporsional* adalah perkalian antara input yang berupa error dengan konstanta atau parameter proporsional.

Gambar 2.10 menunjukan diagram blok yang memperlihatkan hubungan antara besaran setting, besaran aktual dengan besaran keluaran pengontrol *proporsional*. Sinyal kesalahan (error) merupakan perbedaan antara besaran setting dengan besaran aktualnya. Perbedaan ini akan mempengaruhi pengontrol untuk membuat sinyal positif (mempercepat pencapaian harga setting) atau negative (memperlambat tercapainya harga yang diinginkan).



Gambar 2.10 Diagram kontroler propersional

Pengontrol *proporsional* mempunyai 2 parameter, pita *proporsional* (*propotional band*) dan konstanta *proporsional*. Daerah kerja kontroler efektif dicerminkan oleh pita *proporsional* sedangkan konstanta proporsional menunjukan nilai faktor penguatan sinyal

terhadap sinyal kesalahan Kp. Hubungan antara pita *proporsional* (PB) dengan konstanta *proporsional* (Kp) ditunjukkan secara presentasi oleh persamaan berikut :

$$PB = \frac{1}{Kp} \text{ x } 100\% \quad (2.1)$$

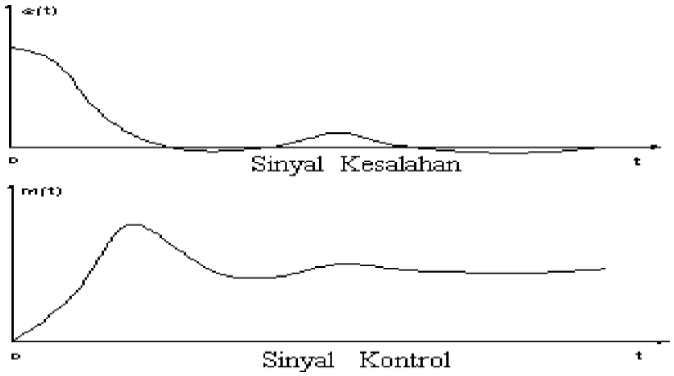
Beberapa ciri-ciri kontroler *proporsional* :

- Jika konstanta *proporsional* kecil, maka respon output sistem dari kontrol *proporsional* akan semakin lambat untuk mencapai setpointnya.
- Jika konstanta *proporsional* dinaikkan maka respon sistem akan semakin cepat untuk mencapai setpoint.
- Jika konstanta *proporsional* terlalu besar, maka output sistem akan tidak stabil atau sistem berosilasi.
- Nilai konstanta *proporsional* dapat sedemikian rupa di atur untuk mengurangi error steady state namun tidak akan bisa menghilangkannya.

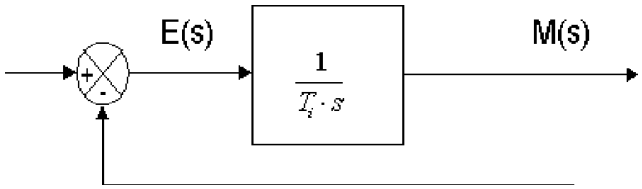
2.4.2 Kontrol Integral

Kontrol *Integral* berfungsi untuk menghilangkan error steady state yang tidak bisa dihilangkan oleh kontroler *proporsional*, kontrol *integral* ini sangat tidak disarankan bekerja sendirian karena karakteristiknya yang lambat. Prinsip dari kontroler ini mirip dengan sebuah *integral* yang mana sangat dipengaruhi oleh perubahan. *Output* dari kontroler ini adalah menjumlahkan terus menerus dari inputnya yang berupa error. Jika sinyal *error* pada inputnya tidak mengalami perubahan, maka *output* akan menjaga keadaan seperti sebelum terjadinya perubahan input.

Sinyal keluaran pengontrol *integral* merupakan luas bidang bidang yang dibentuk oleh kurva kesalahan penggerak. Sinyal keluaran akan berharga sama dengan harga sebelumnya ketika sinyal kesalahan berharga nol. Gambar 2.11 menunjukkan contoh sinyal kesalahan yang dimasukan kedalam pengontrol *integral* dan keluaran pengontrol *integral* terhadap perubahan sinyal kesalahan tersebut.



Gambar 2.11 Kurva sinyal kesalahan e(t) terhadap t pada pembangkit kesalahan nol



Gambar 2.12 Menunjukkan blok diagram antara besaran kesalahan dengan keluaran suatu pengontrol integral.

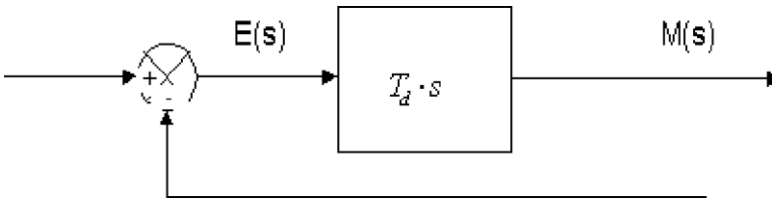
Beberapa ciri-ciri kontroler *integral* :

- Keluaran pengontrol membutuhkan selang waktu tertentu, sehingga pengontrol *integral* cenderung memperlambat respon.
- Ketika sinyal kesalahan berharga nol, keluaran pengontrol akan bertahan pada nilai sebelumnya.
- Jika input error diatas atau dibawah nol maka output akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya nilai error pada input dan parameter/konstanta integral/Ki.

- Parameter/Konstanta *Integral*/Ki yang besar akan mempercepat hilangnya offset namun semakin besar nilai kosntanta ini akan mengakibatkan osilasi pada output kontroler.

2.4.3 Kontrol Derivative

Keluaran pengontrol *Derivative* memiliki sifat seperti halnya suatu operasi differensial. Perubahan yang mendadak pada masukan pengontrol, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.13 menunjukkan blok diagram yang menggambarkan hubungan antara sinyal kesalahan dengan keluaran pengontrol.



Gambar 2.13 Blok diagram pengontrol derivative

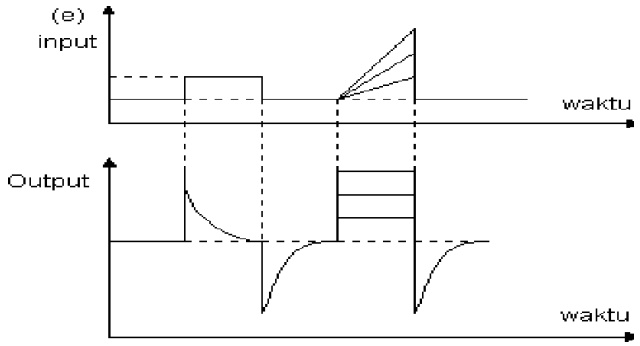
Berikut adalah persamaan dari kontroler *derivative* :

$$D_{out} = K_d \frac{de(t)}{dt} \tag{2.2}$$

Dimana Kd merupakan konstanta *derivative*.

Gambar 2.14 merupakan hubungan antara sinyal masukan dengan sinyal keluaran pengontrol *derivative*. Perubahan mendadak pada input kontrole akan mengakibatkan perubahan yang sangat besar dan cepat, apabila sinyal input berubah mendadak dan menaik (berbentuk fungsi step), output menghasilkan sinyal berbentuk impuls. Sedangkan jika sinyal *input* berubah naik secara perlahan seperti sinyal ramp maka outputnya justru seperti fungsi step yang besar magnitudnya dipengaruhi oleh kecepatan naik dari fungsi ramp dan konstanta/parameter differensialnya. Kontrol *derivative* hanya berubah nilai keluarannya ketika ada perubahan *error* sehingga ketika *error* tidak ada perubahan maka praktis kontrol ini tidak akan bereaksi. Dengan keadaan begitu, kontrol ini tidak bisa berdiri sendiri sebagai

kontroler karena sifatnya yang prediktif dan berubah ketika sinyal *input error* berubah.



Gambar 2.14 Kurva waktu hubungan input-output pengontrol *derivative*

Beberapa ciri-ciri kontroler *derivative* :

- Kontroler ini tidak dapat menghasilkan *output* bila tidak ada perubahan pada *inputnya*.
- Jika nilai *error* berubah terhadap waktu, maka *output* yang dihasilakan kontroler tergantung pada nilai Td dan laju perubahan sinyal error (*rate of error*).
- Kontrol differensial dapat menghasilkan koreksi yang sangat baik sebelum *error overshoot* menjadi sangat besar dan cenderung meningkatkan stabilitas sistem.

2.4.4 Kontrol PI

Kontrol PI merupakan gabungan antara kontrol *proporsional* dan kontrol *integral*. Kontrol PI sangat tepat digunakan pada sistem yang tidak begitu membutuhkan kestabilan sistem namun butuh akurasi pada saat kondisi baik. Dengan parameter Kp dan Ki yang tepat maka sistem akan memiliki respon yang sangat cepat dan *error steady state* bisa dieliminasi. Kekurangan kontrol ini adalah Ketika ada gangguan atau perubahan set point atau pada kondisi awal akan sedikit membutuhkan waktu untuk menuju kondisi baiknya disebabkan osilasi. Kontrol PI mempunyai persamaan matematik sebagai berikut :

$$PI(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \qquad (2.3)$$

2.4.5 Kontrol PD

Kontrol PD merupakan gabungan antara kontrol *proporsional* dan kontrol *derivative*. Kontrol PD sangat tepat digunakan pada sistem yang tidak begitu membutuhkan akurasi pada saat kondisi baik dan membutuhkan respon yang cepat, kondisi baik yang cepat dan sistem yang stabil. Dengan parameter Kp dan Kd yang tepat maka sistem akan memiliki respon yang sangat cepat dan kondisi baik yang juga cepat. Kekurangan kontrol ini adalah Ketika sistem membutuhkan akurasi terhadap *set point* yang tinggi serta tidak boleh ada *error steady state*. Kontrol PD mempunyai persamaan matematik sebagai berikut :

$$PD(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (2.4)$$

2.5 Kalman Filter

Kalman filter adalah suatu metode atau algoritma statistik yang digunakan untuk mengestimasi keadaan atau variabel suatu sistem berdasarkan serangkaian pengukuran atau observasi yang diberikan, yang mungkin terkontaminasi dengan gangguan atau ketidakpastian. Tujuan utama dari *Kalman filter* adalah untuk menghasilkan estimasi yang optimal dan akurat tentang keadaan sistem seiring waktu, sambil meminimalkan ketidakpastian.

Kalman filter pertama kali dikembangkan oleh Rudolf E. Kalman pada tahun 1960 dan telah menjadi alat yang sangat penting dalam berbagai aplikasi, termasuk navigasi, pengolahan sinyal, robotika, kendali otomatis, dan banyak lagi. Ini memiliki dua tahap utama:

- Prediksi (Proses Prediksi): Dalam tahap ini, *Kalman filter* menggunakan model sistem yang diketahui untuk memprediksi keadaan sistem pada waktu berikutnya berdasarkan estimasi sebelumnya. Ini melibatkan proyeksi maju dari keadaan sistem dan ketidakpastian.
    - Prediksi Status

$$\hat{x}(t) = F \,.\, x(t-1) + B.u(t) \quad (2.5)$$

$\hat{x}(t)$ = adalah perkiraan status variable pada waktu *t.*

F adalah matriks transisi yang menggambarkan evolusi sistem dari satu waktu ke waktu berikutnya.

$x(t-1)$ = adalah status variable pada waktu sebelumnya.

B adalah matriks control yang menghubungkan pengaruh input control $u(t)$ terhadap perubahan status variable.

$u(t)$ = adalah input control pada waktu *t.*

- Prediksi Kovarian

$$P(t) = F \,.\, P(t-1).\, F^T + Q \tag{2.6}$$

P(t) adalah matriks kovarian perkiraan pada waktu t.

Q adalah matriks kovarian noise proses yang mewakili ketidakpastian dalam model dinamis.

- Koreksi (Proses Koreksi): Pada tahap ini, *Kalman filter* memperbarui prediksi berdasarkan data pengukuran yang diterima pada waktu berikutnya. Ini dilakukan dengan membandingkan pengukuran aktual dengan prediksi dan menghitung sejauh mana pengukuran ini memengaruhi estimasi keadaan sistem dan ketidakpastian.

  - Gain Kalman

$$K(t) = P(t) \,.\, H^T \,.\, (H \,.\, P(t) \,.\, H^T + R)^{-1} \tag{2.7}$$

K(t) adalah gain Kalman, menentukan seberapa besar kita percaya pengukuran aktual dibandingkan dengan prediksi.

R adalah matriks kovarian noise pengukuran yang mewakili ketidakpastian dalam pengukuran.

  - Koreksi Status

$$\hat{x}(t) = \hat{x}(t) + K(t).\, y(t) \tag{2.8}$$

  - Koreksi Kovarian

$$P(t) = (I - K(t) \,.\, H).\, P(t) \tag{2.9}$$

I adalah matriks identitas

Keuntungan utama dari *Kalman filter* adalah kemampuannya untuk menggabungkan informasi dari model sistem dan data pengukuran untuk menghasilkan estimasi yang lebih baik daripada salah satunya secara terpisah. Ini juga mampu mengatasi gangguan atau ketidakpastian dalam data pengukuran, yang seringkali merupakan masalah dalam aplikasi dunia nyata.

Untuk penggunaan *kalman filter* pada sistem kontrol *quadcopter* dengan PID, *kalman filter* yang cocok adalah *Extended Kalman Filter* (EKF) atau *Unscented Kalman Filter* (UKF). Ini karena kedua jenis *kalman filter* ini dapat menangani kasus ketika model sistem *non-linear*, yang seringkali lebih sesuai dengan dinamika *quadcopter*.

Extended Kalman Filter (EKF):

- EKF merupakan versi linierisasi dari *Kalman Filter* yang digunakan untuk menangani sistem *non-linear*. EKF menggunakan pendekatan linierisasi melalui turunan pertama untuk mengevaluasi matriks Jacobian dari model sistem *non-linear*.
- Dalam kasus *quadcopter*, EKF dapat digunakan untuk memperkirakan keadaan (seperti posisi, kecepatan, dan orientasi) serta parameter sistem secara *real-time* dengan memanfaatkan model dinamika *non-linear quadcopter* dan pengukuran sensor.

Unscented Kalman Filter (UKF):

- UKF adalah alternatif untuk menangani sistem *non-linear* tanpa perlu melakukan linierisasi model. UKF menggunakan teknik yang disebut transformasi *unscented* untuk menangkap distribusi *nonlinearitas*.
- UKF cenderung lebih akurat daripada EKF dalam menangani sistem *non-linear* yang kompleks karena tidak memerlukan *linierisasi* yang mungkin tidak akurat pada model *quadcopter* yang rumit.

2.6 Respon Sistem Kendali

Respon sistem digunakan dalam menganalisa karakter suatu sistem kendali. Pada respon sistem biasanya mengunakan unit step dengan kondisi awal nol atau keadaan diam. Hal tersebut bertujuan mempermudah dalam

mengamati suatu perhitungan atau analisa respon dari suatu sistem kendali. Beberapa respon sistem yang digunaan antara lain meliputi :

1. *Delay time* (*td*) waktu yang dibutuhkan sistem untuk pertama kali merespon hingga mencapai setengah dari nilai yang menjadi acuan atau nilai akhir dari respon sistem.

2. *Risetime* (*tr*) waktu yang diperlukan untuk respon naik dari 10% hingga 90%, 5% hingga 95%, 0% hingga 100% dari nilai final. Nilai risetime dapat dicari mengunakan persamaan (3.1). adalah frekuensi natural sistem, yaitu frekuensi di mana sistem akan berosilasi tanpa adanya redaman. Ini adalah ukuran dari seberapa cepat sistem akan berayun saat mencapai nilai setpoint akhirnya.

$$t_r = \frac{\pi\beta}{\omega_d} \tag{3.1}$$

Dengan

$$\beta = tan^{-1}\frac{\omega_d}{-\sigma} \tag{3.2}$$

Keterangan :

t_r adalah *rise time*, yaitu waktu yang dibutuhkan sistem untuk mencapai 90% dari nilai *setpoint* akhirnya setelah terjadi perubahan *input* yang signifikan

β adalah waktu pengendalian (*time constant*) dari sistem, yang merupakan waktu yang diperlukan agar sistem mencapai 63.2% dari nilai *setpoint* akhirnya setelah terjadi perubahan *input* yang signifikan.

ω_d adalah *frekuensi natural* sistem, yaitu *frekuensi* di mana sistem akan berosilasi tanpa adanya redaman. Ini adalah ukuran dari seberapa cepat sistem akan berayun saat mencapai nilai *setpoint* akhirnya.

3. *Peak time* (tp) merupakan waktu yang dibutuhkan *sistem* untuk mencapai nilai tertinggi pada saat *overshoot* pertama.

4. Maksimum *overshoot* (Mp) merupakan nilai maksimum yang dihasilkan dari nilai respon *sistem* dan ditampilkan dalam bentuk persentase.

Overshoot maksimum didapat dari nilai tertinggi dan dihitung dari nilai *setpoint*.

1. *Settlingtime* (*ts*) waktu yang dibutuhkan sistem untuk mendapat posisi stabil dengan toleransi *error* sebesar 2% hingga 5%.(Makasudede, 1953).