

BAB 2

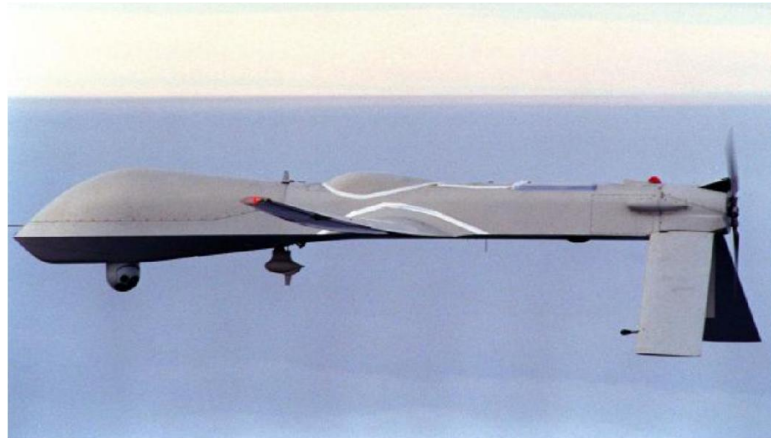
TINJAUAN PUSTAKA

2.1 *Unmanned Aerial Vehicle (UAV)*

Unmanned Aerial Vehicle (UAV) atau kendaraan udara tanpa awak adalah salah satu teknologi yang sedang mengalami perkembangan yang pesat dan memiliki potensi yang sangat besar, baik untuk keperluan militer maupun kepentingan sipil. Contoh aplikasi yang dapat diimplementasikan pada UAV adalah untuk kebutuhan survey, patroli, deteksi tambang mineral, riset, fotografi, dan keperluan lain. Kelebihan dari UAV adalah dapat digunakan pada misi-misi berbahaya tanpa membahayakan pilot / manusia.

UAV awal dibuat kebanyakan untuk kepentingan militer. Seiring berkembang teknologi dan banyak penelitian yang berkaitan dengan UAV, sekarang UAV memiliki fungsi yang berbeda-beda. Salah satu adalah Transportasi UAV yang dapat mengangkut barang pada bagian pesawat, namun untuk UAV jenis helikopter dapat mengangkut barang diluar badan pesawat. Beban yang diberikan pada UAV ini biasa beban yang tergolong ringan dan membutuhkan kecepatan pengiriman dan resiko pengiriman yang tinggi.

Pesawat tanpa awak dapat digunakan pada daerah berbahaya yang apabila dilakukan dengan pesawat berawak akan membahayakan keselamatan awak pesawat tersebut. Daerah yang berbahaya tersebut daerah pusat badai, ketinggian yang berbahaya, daerah rawan perang, dsb. Beberapa UAV dilengkapi dengan senjata yang digunakan pada kawasan perang, seperti Predator RQ-1 yang telah dapat melakukan serangan ke target-target di darat, bahkan Predator RQ-1 juga dapat melakukan perang antar pesawat di udara. Kemampuan pesawat UAV semacam ini sangat menguntungkan bagi negaranegara maju, karena resiko penggunaan pilot sebagai sandera atau tawanan perang apabila tertangkap dapat dihilangkan, contoh pesawat ditunjukkan pada Gambar 2.1 UAV Raven RQ



Gambar 2. 1 UAV Raven RQ

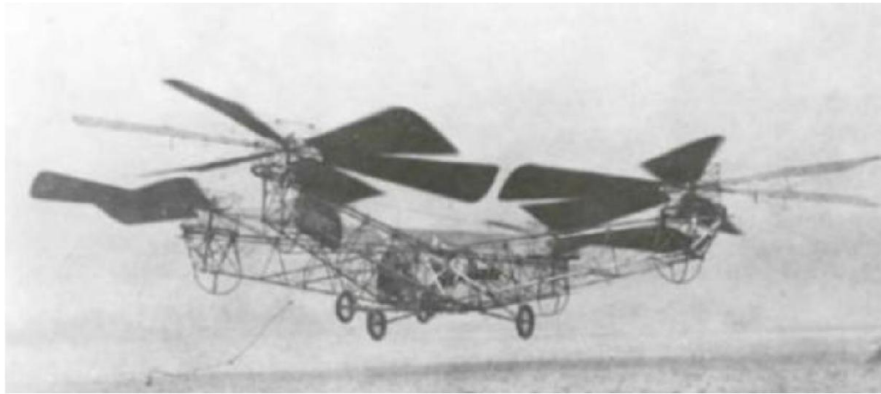
UAV juga dapat digunakan membantu peranan tim SAR (*Search and RESCue*) dalam pencarian korban dalam kecelakaan atau korban bencana alam karena UAV dapat terbang dalam cuaca buruk sekalipun. Daya tahan terbang UAV yang tidak tergantung pada pilot juga memungkinkan UAV mencari korban secara terus-menerus. Ada banyak tipe dari UAV, diantara adalah *fixed wing*, *axial wing*, *coaxial wing*, dan *quadrotor*. Dalam penelitian ini tipe pesawat yang digunakan adalah jenis *quadrotor* atau *quadcopter*. (Yulistiwa,Andry:28:2013)

2.2 Quadcopter

Quadcopter adalah salah satu jenis *rotorcraft* yang memiliki 4 buah rotor sebagai penggerak *propeller* yang menghasilkan gaya angkat. *Quadcopter* dapat melakukan *take off* dan *landing* secara *vertikal*. *Vertical Take Off Landing (VTOL) Aircraft* merupakan jenis pesawat yang dapat melakukan *take off* dan *landing* tegak lurus terhadap bumi sehingga dapat dilakukan pada tempat yang sempit. Helikopter, *tricopter*, *quadcopter*, dan beberapa *fixed wing* dengan mesin jet termasuk kategori ini.

2.2.1 Sejarah quadcopter

Quadcopter sendiri dibuat pertama kali pada tahun 1923 oleh George DeBothezat yang diberi nama *De Bothezat Quadrotor*. ditunjukkan pada Gambar 2.2 *De Bothezat Quadrotor*.



Gambar 2. 2 De Bothezat Quadrotor (1923)

Quadcopter cukup berkembang, terbukti perusahaan yang bergerak di bidang penerbangan Bell Helicopter dan Boeing menciptakan *Bell Boeing Quad TiltRotor (QTR)*. *Bell Boeing Quad TiltRotor* memiliki 4 buah rotor dapat menampung kapasitas setara dengan C-130 Hercules, dan mampu terbang pada kecepatan 250 knot. Bentuk dari *Bell Boeing Quad TiltRotor* ini ditunjukkan pada Gambar 2.3 *Bell Boeing Quad Tiltrotor (QTR)*. (Yulistiyo,Andry:32:2013)



Gambar 2. 3 Bell Boeing Quad TiltRotor (QTR)

Saat sekarang ini *quadrotor* atau *quadcopter* banyak menjadi obyek penelitian. *Quadcopter* berukuran kecil / mini yang dikendalikan melalui *remote* kontrol banyak digunakan orang untuk fotografi, pemetaan, atau sekedar hobi. Salah satu perusahaan yang memproduksi *quadcopter* mini adalah Mikrokopter.

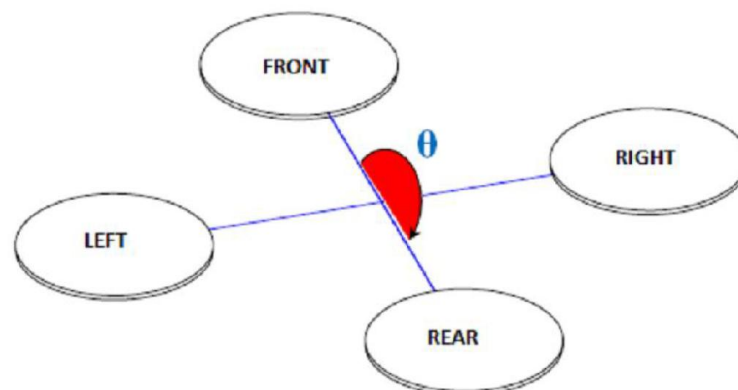
Quadcopter berukuran kecil tersebut diatur oleh sebuah chip *mikrokontroler*. Kecepatan kerja dari *mikrokontroler* yang ada mulai dari 8Mhz hingga kecepatan mendekati kecepatan komputer yaitu 400Mhz. Semakin cepat kecepatan kerja dari *mikrokontroler* yang digunakan maka proses pengambilan data sensor, penghitungan keluaran motor, keluaran sinyal motor, dan waktu pengiriman data ke *ground station* semakin cepat. Mikrokopter ditunjukkan pada Gambar 2.4 mikrokopter. (Yulistiyanto,Andry:33:2013)



Gambar 2. 4 Produk dari Mikrokopter

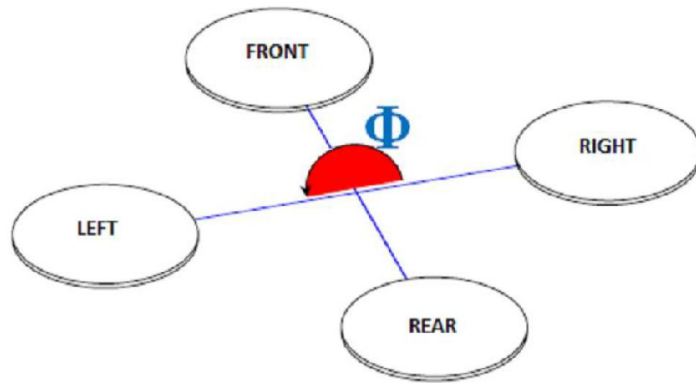
2.2.2 Pergerakan *Quadcopter*

Quadcopter dapat digambarkan sebagai kendaraan kecil dengan empat baling-baling melekat pada rotor terletak di *frame* berbentuk *plus* (+). *Frame* berbentuk *plus* (+) bertujuan untuk mempermudah mengontrol *quadcopter*. Mekanisme pergerakan *quadcopter* ditunjukkan dengan gambar 2.5 *Pitch Direction Of Quadcopter*, 2.6 *Roll Direction Of Quadcopter*, dan 2.7 *Yaw Direction Of Quadcopter*. (Yulistiyanto,Andry:33:2013)



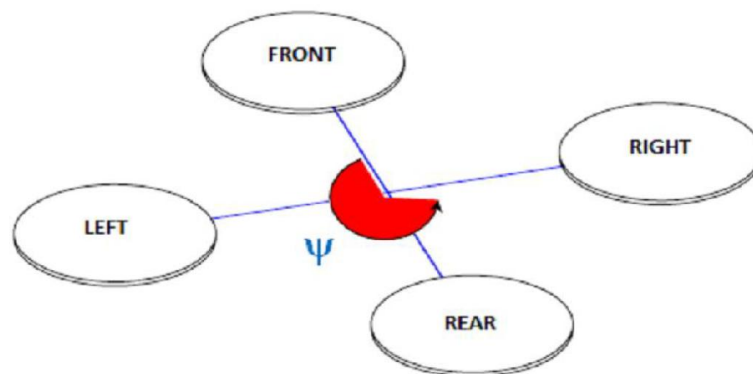
Gambar 2. 5 Pitch Direction Of Quadcopter

Sudut *Pitch* (θ) adalah sudut pada *quadcopter* yang berada pada sumbu Y seperti yang terlihat pada gambar 2.5.



Gambar 2. 6 Roll Direction Of Quadcopter

Sudut *Roll* (ϕ) adalah sudut pada *quadcopter* yang berada pada sumbu X seperti yang terlihat pada gambar 2.6.

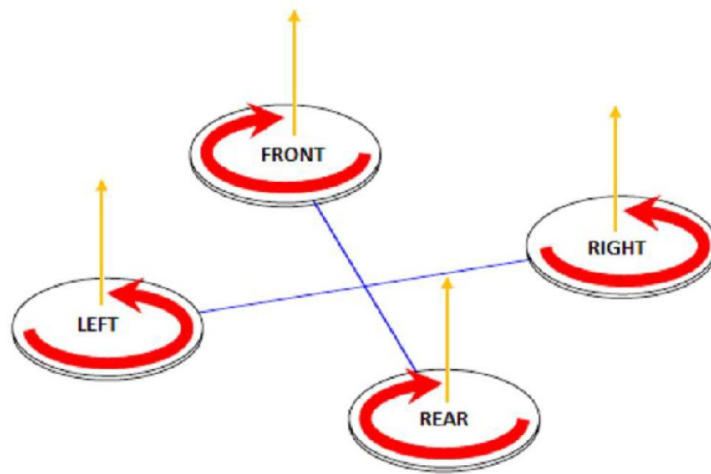


Gambar 2. 7 Yaw Direction Of Quadcopter

Sudut *Yaw* (ψ) adalah sudut pada *quadcopter* yang berada pada sumbu Z seperti yang terlihat pada gambar 2.7.

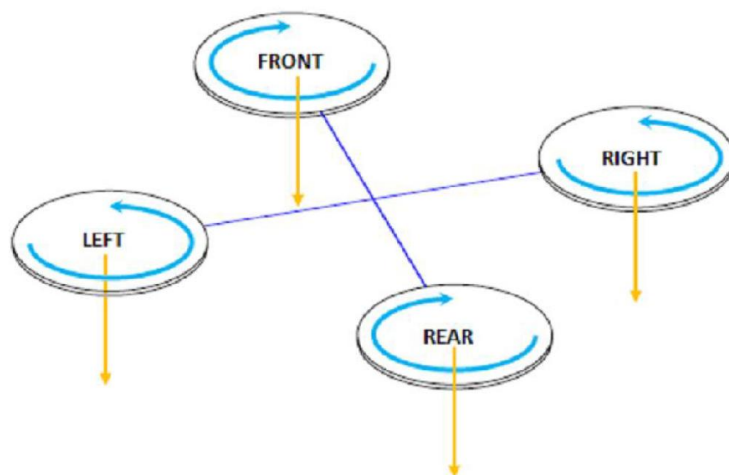
2.2.2.1 Pergerakan mekanisme *Quadcopter* lepas landas dan mendarat

Lepas landas adalah gerakan *Quadcopter* yang mengangkat dari tanah untuk posisi melayang . yang digambarkan pada gambar 2.8.



Gambar 2. 8 Menggambarkan gerak lepas landas *quadcopter*

Gerakan pendaratan adalah meningkatkan kecepatan penurunan empat motor secara bersamaan yang berarti mengubah gerakan *vertikal* dan digambarkan pada gambar 2.9.

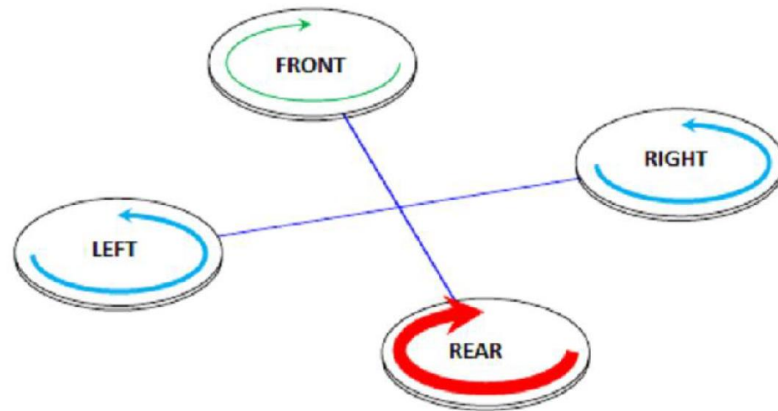


Gambar 2. 9 Menggambarkan gerak mendarat *Quadcopter*

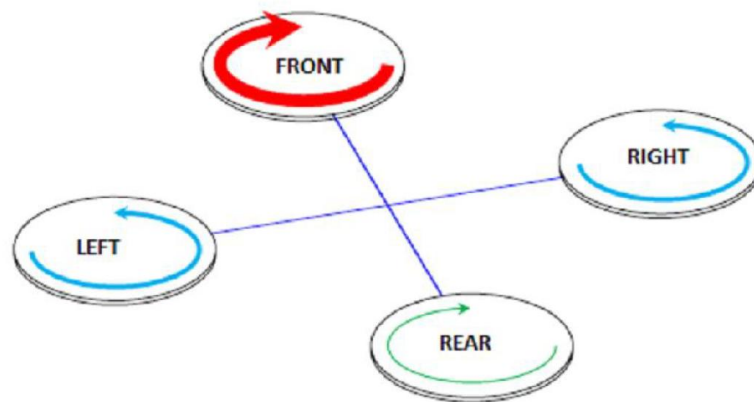
2.2.2.2 Pergerakan mekanisme *Quadcopter* maju dan mundur

Maju mundur adalah gerak kontrol dengan kenaikan, penurunan kecepatan belakang, depan. Kecepatan motor secara bersamaan akan mempengaruhi sudut

quadcopter tersebut. Gerakan maju dan mundur dari *quadcopter* yang ditunjukkan pada gambar 2.10 menggambarkan gerak maju *quadcopter* dan 2.11 Menggambarkan gerak mundur *Quadcopter*.



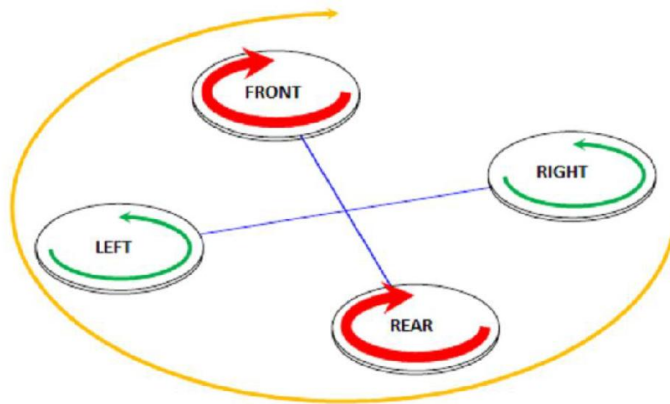
Gambar 2. 10 Menggambarkan gerak maju *quadcopter*



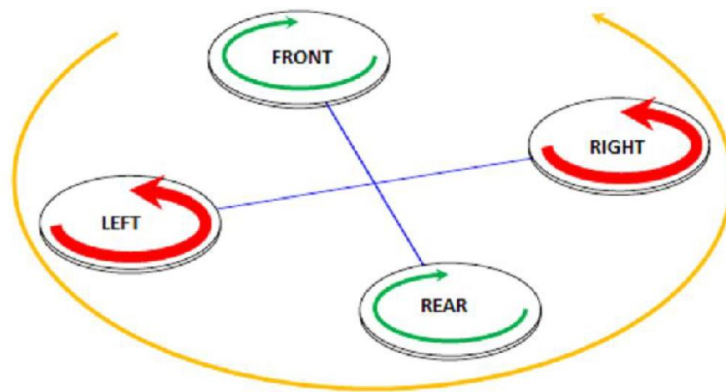
Gambar 2. 11 Menggambarkan gerak mundur *quadcopter*

2.2.2.3 Pergerakan mekanisme *Quadcopter* ke kiri dan ke kanan

Untuk gerak kiri dan kanan, dapat mengontrol dengan mengubah sudut *yaw* dari *quadcopter* yang ditunjukkan pada gambar 2.12 Menggambarkan gerak ke kanan *Quadcopter*, dan 2.13 Menggambarkan gerak ke kiri *Quadcopter*.



Gambar 2. 12 Menggambarkan gerak ke kanan *Quadcopter*



Gambar 2. 13 Menggambarkan gerak ke kiri *Quadcopter*

2.2.2.4 Pergerakan mekanisme *Quadcopter* melayang

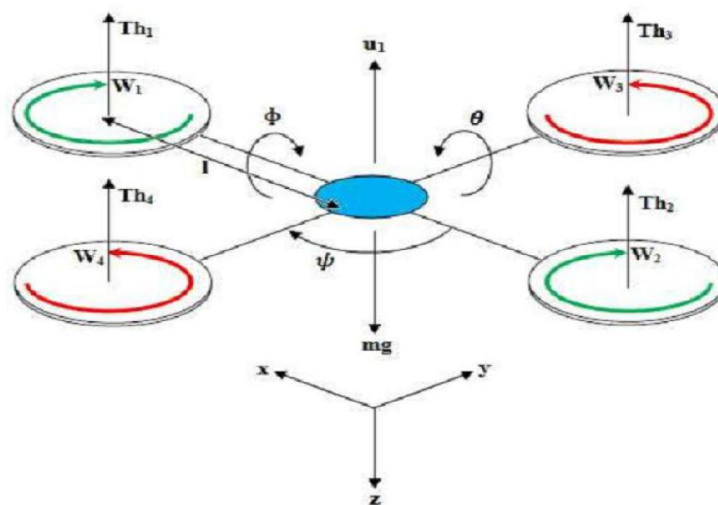
Posisi melayang atau tetap dari *quadcopter* dapat dilakukan oleh dua bagian motor yang berputar dalam searah jarum jam dan berlawanan arah jarum jam masing-masing dengan kecepatan yang sama. Dengan dua motor berputar searah jarum jam dalam posisi searah jarum jam dan kontra jumlah total torsi reaksi sama dengan nol dan *quadcopter* ini diperbolehkan dalam posisi melayang. (Yulistiyanto, Andry:37:2013)

2.2.3 Pemodelan matematika *Quadcopter*

Pemodelan secara fisik terbilang *kompleks* dan apabila tanpa ada asumsi yang digunakan untuk menyederhanakan persamaan, perhitungan menjadi tidak praktis. Beberapa asumsi yang digunakan dalam pemodelan ini adalah :

1. Struktur dari *quadcopter* dikatakan *rigid*.
2. Struktur dari *quadcopter* dikatakan *simetris*.
3. Struktur dari *propeller* dikatakan *rigid*.
4. Gaya *thrust* dan *drag* adalah proporsional dengan kuadrat dari kecepatan *propeller*.
5. Keadaan model adalah keadaan ketika *hovering quadcopter* memiliki 6 *degree of freedom* (DoF)

Gerakan skematik *quadcopter* diwakili dalam Gambar 2.14 dan berdasarkan ini skematis, pemodelan matematika *quadcopter* diperoleh dengan cara sebagai berikut :



Gambar 2. 14 Skema *Quadcopter*

Keterangan :

U_1 : Jumlah tekanan dari setiap motor

Th_1 : Dorongan yang dihasilkan oleh motor depan

Th_2 : Dorongan yang dihasilkan oleh motor belakang

Th_3 : Dorongan yang dihasilkan oleh motor kanan

Th_4 : Dorongan yang dihasilkan oleh motor kiri

m : Massa *quadcopter*

g : Percepatan gravitasi

l : Panjang setengah dari *quadcopter*

x, y, z : Tiga posisi

θ, ϕ, ψ : Tiga sudut *eurel* mewakili *pitch, roll, and yaw*

2.2.3.1 Dinamika *Quadcopter*

Dinamika *quadcopter* adalah keadaan dimana *quadcopter* berusaha mempertahankan gerakan saat di udara berdasarkan *leveling* awal. Pada dinamika *quadcopter*, terdapat persamaan kinematik dan dinamik. Kinematik berhubungan dengan posisi gerak, sedangkan dinamik berhubungan dengan posisi. untuk persamaan kinematik sebagai berikut:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \cos \theta & \sin \varphi \cos \phi + \cos \varphi \sin \theta \sin \phi & \sin \varphi \sin \phi + \cos \varphi \sin \theta \cos \phi \\ \sin \varphi \cos \theta & \cos \varphi \cos \phi + \sin \varphi \sin \theta \sin \phi & -\cos \varphi \sin \phi + \sin \varphi \sin \theta \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.1)$$

Variable ϕ, θ , dan φ adalah posisi pada *frame* bumi, sedangkan u, v , dan w adalah percepatan pada *frame* badan *quadcopter*. Selanjutnya di bawah ini adalah transfer matriks dari persamaan (2.1), dimana p, q , dan r adalah kecepatan pada masing-masing posisi.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sec \phi \sin \theta & \sec \phi \cos \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.2)$$

Untuk dinamika *quadcopter* persamaan sebagai berikut :

$$F = m \frac{dv}{dt} \quad (2.3)$$

m adalah massa dari sistem yang digunakan, a adalah percepatan gerak, dan F adalah gaya yang terjadi pada sistem. Sehingga persamaan dapat ditulis:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \frac{dv}{dt_0} = m \left(\begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \quad (2.4)$$

Sehingga percepatan *linier* dari *quadcopter* dapat dihitung:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} qw - rv \\ pw - ru \\ pv - qu \end{bmatrix} \quad (2.5)$$

M adalah torsi yang terjadi, dan h merupakan momentum putar. Dengan menggunakan persamaan coriolis didapat:

$$M = \frac{dv}{dt_0} = \frac{dh}{dt_b} + w \times h \quad (2.6)$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = 1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times 1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.7)$$

Sehingga ,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = 1^{-1} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - 1^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times 1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.8)$$

2.2.3.2 Aerodinamis *Quadcopter*

Adapun gaya aerodinamis pada setiap motor adalah sebagai berikut :

$$U_1 = b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$$

$$U_2 = \ell b (-\Omega_1^2 + \Omega_3^2)$$

$$U_3 = \ell b (-\Omega_2^2 + \Omega_4^2)$$

$$U_4 = d (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$$

$$\Omega = (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4)$$

b merupakan konstanta *thrust* (gaya angkat motor) , ℓ adalah jarak pusat *quadcopter* ke propeller, Ω adalah kecepatan sudut putar propeller, dan d adalah konstanta *drag* (gaya hambat) yang terjadi pada *quadcopter*. U_1 adalah kondisi dimana *quadcopter* ketika naik ke atas, U_2 adalah kondisi dimana terjadi gerakan maju atau mundur pada *quadcopter*, U_3 adalah kondisi dimana terjadi gerakan ke

kiri dan ke kanan pada *quadcopter*, dan U_4 adalah kondisi dimana *quadcopter* melakukan gerakan memutar.

Sehingga jika digabungkan dengan persamaan kinematika dan dinamik *quadcopter*, maka persamaan dinamika percepatan *quadcopter* dapat ditulis menjadi:

$$X = (\sin \varphi \sin \phi + \cos \psi \sin \theta \cos \phi) \frac{U_1}{m}$$

$$Y = (\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \frac{U_1}{m}$$

$$Z = -g + (\cos \theta \cos \phi) \frac{U_1}{m}$$

$$\dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{U_2}{I_{XX}}$$

$$\dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr - \frac{J_{TP}}{I_{YY}} p \Omega + \frac{U_2}{I_{YY}}$$

$$\dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq - \frac{U_4}{I_{ZZ}}$$

X merupakan percepatan *quadcopter* pada sumbu X, Y merupakan percepatan *quadcopter* pada sumbu Y, dan Z merupakan percepatan *quadcopter* pada sumbu Z. $\dot{p}$ adalah percepatan sudut di sumbu X, $\dot{q}$ adalah percepatan di sumbu Y, $\dot{r}$ dan adalah percepatan sudut di sumbu Z. (Bresciani, Tommaso: 27-35:2008)

2.3 ESC (*Electric Speed Control*)

ESC adalah rangkaian elektronik yang berfungsi sebagai pengatur kecepatan putaran motor pada pesawat atau helikopter dengan cara menterjemahkan sinyal yang diterima *receiver* dari *transmitter*. Di pasaran terdapat berbagai merk ESC dengan kekuatan arus (*current rating*) dan kekuatan voltase (*voltage rating*) serta *feature* yang ditawarkan. Contoh gambar ditunjukkan pada gambar 2.15 *Electric Speed Control*.



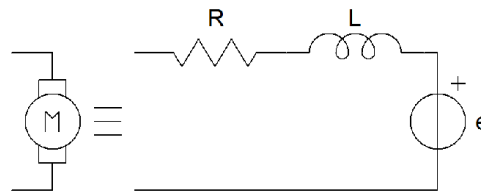
Gambar 2.15 *Electric Speed Control Quatro*

Untuk menentukan *ESC* yang akan digunakan sangatlah penting untuk mengetahui kekuatan (*peak current*) dari motor. Sebaiknya *ESC* yang digunakan memiliki *Peak Current* yang lebih besar daripada motor. Contoh, kekuatan motor adalah 12A (*ampere*) pada saat *throttle* terbuka penuh. *ESC* yang akan digunakan adalah *ESC* yang berkekuatan 18A atau 20A. Jika di paksakan menggunakan *ESC* 10A kemungkinan pada saat *throttle* dibuka penuh, *ESC* akan panas bahkan terbakar. (Yulistiyanto, Andry:58:2013)

2.4 *Brushless* DC Motor (BLDCM)

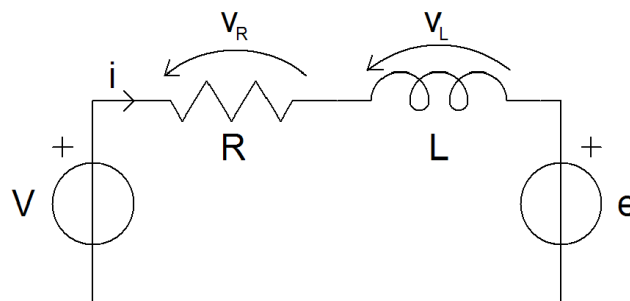
Motor DC *Brushless* adalah aktuator yang mengubah energi listrik menjadi energi mekanik dan sebaliknya. Hal ini terdiri dari dua interaktif rangkaian elektromagnetik . Yang pertama disebut rotor dan yang kedua disebut stator. Rotor adalah bagian dari motor yang bebas berputar sekitar satu detik, sedangkan stator adalah bagian dari motor yang tetap sebagai gantinya. Dalam rotor, beberapa kelompok gulungan tembaga yang terhubung secara seri dan secara eksternal dapat diakses berkat alat yang disebut komutator. Pada stator, dua atau lebih magnet permanen memaksakan medan magnet yang mempengaruhi rotor. Dengan menerapkan aliran arus DC ke gulungan, rotor berubah karena gaya yang dihasilkan oleh listrik dan selingan magnetik. Berkat rotor dan geometri komutator, motor terus berputar sementara dipasok oleh tegangan DC pada terminal.

Motor DC *Brushless* memiliki model terkenal yang mengikat besaran listrik dan mekanik. Model ini terdiri dari rangkaian resistor R [Ω], sebuah induktor L [H] dan generator tegangan e [V]. Resistor ini merupakan kerugian Joule karena aliran arus ke konduktor tembaga. Nilainya tergantung pada karakteristik geometrik dan bahan seperti kawat resistivitas, panjang dan bagian. Perilaku induktor berasal dari bentuk kabel motor yang lebar di tengah rotor. Terakhir, e Generator (disebut juga BEMF) memasok tegangan sebanding dengan kecepatan motor.



Gambar 2.16 Model Motor

Rangkaian dari motor DC dikendalikan oleh tegangan nyata Generator v [V] yang memberikan input kontrol. Secara teori, resistor lain harus ditambahkan dalam rangkaian generator tegangan mewakili kerugian pembangkit. Namun, dalam proyek yang baik, kerugian pembangkit tetap rendah karena itu adalah mungkin untuk mengabaikan mereka dalam model. Rangkaian listrik dasar ditunjukkan pada Gambar 2.17.



Gambar 2.17 Rangkaian Motor

Dimana v_R [V] adalah tegangan di resistor R dan v_L [V] adalah tegangan induktor L . Persamaan (2.9) dapat ditulis kembali sebagai dalam persamaan berikutnya.

$$v = R i + L \frac{\partial i}{\partial t} + K_E \omega_M \quad (2.9)$$

i [A] adalah motor saat ini, K_E [V s rad⁻¹] disebut konstanta bermotor dan ω_M [rad s⁻¹] adalah kecepatan sudut bermotor. Ditambahkan persamaan pertama yang telah diubah dengan menggunakan hukum Ohm $v_R = R i$, sedangkan yang kedua menggunakan persamaan $v_L = L \frac{\partial i}{\partial t}$ induktor diferensial. Anggota terakhir dari persamaan (2.9) menunjukkan bahwa mekanik dan listrik yang berkorelasi.

Kontribusi dari bagian induktor penting untuk menentukan karakteristik driver motor DC. Namun sering diabaikan dalam mekanisme perhitungan karena tiga aspek utama:

- Sebagian besar motor yang digunakan dalam robotika menunjukkan kecil induktansi berkat optimasi untuk konstruksi.
- Tiang (waktu respon) dari bagian listrik selalu jauh lebih cepat daripada yang mekanik, sehingga kecepatan sistem secara keseluruhan akan ditentukan hanya dengan kontribusi paling lambat.
- Ini jauh lebih mudah untuk memecahkan persamaan diferensial orde pertama daripada urutan kedua satu.

Untuk semua alasan ini efek induktor diabaikan dalam pekerjaan ini juga. Ada persamaan-kedepan (2.9) dapat disederhanakan sesuai dengan persamaan (2.10).

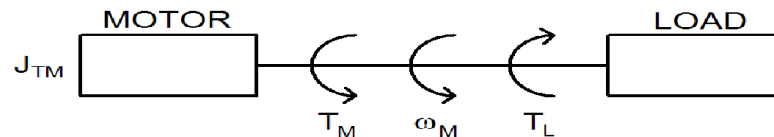
$$v = R i + K_E \omega_M \quad (2.10)$$

Dinamika motor digambarkan oleh persamaan berikut.

$$J_{TM} \omega_M = T_M - T_L \quad (2.11)$$

Dimana J_{TM} [N m s²] adalah total momen inersia motor, ω_M [rad s⁻²] adalah percepatan sudut bermotor, T_M [N m] adalah torsi motor dan T_L [N m] adalah torsi beban. Persamaan (2.18) menyatakan bahwa ketika T_M torsi motor dan beban torsi T_L tidak sama, ada percepatan (atau deselerasi) dari motor kecepatan

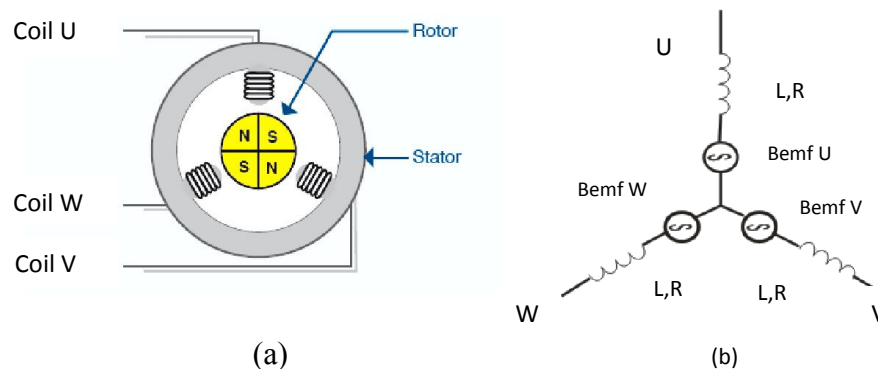
sudut ω_M . Ini variasi kecepatan juga tergantung pada total momen inersia motor J_{TM} : semakin kecil nilai J_{TM} , semakin tinggi percepatan. Gambar 2.18 menunjukkan sketsa struktur mekanik disederhanakan. (Bresciani, Tommaso :43:2008)



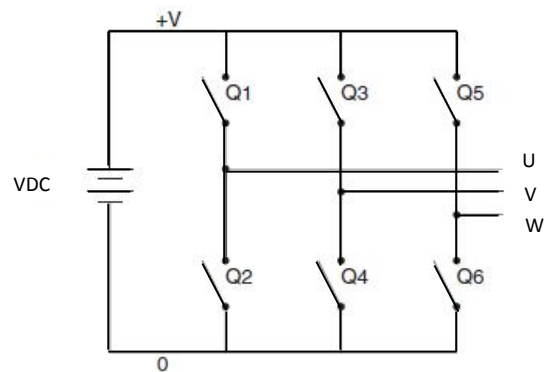
Gambar 2.18 Sistem Motor Sederhana

Motor *brushless* memiliki 3 komponen utama yaitu stator, rotor dan *hall* sensor. Tiga lilitan terletak pada stator yang meliputi induktansi, resistansi dan gaya elektromotif terangkai secara seri, sedangkan pada rotor memiliki magnet permanen dengan kutub magnet yang mempengaruhi ukuran step dan reaksi torsi dari motor.

Struktur stator dan rotor dapat ditunjukkan pada Gambar 2.19.



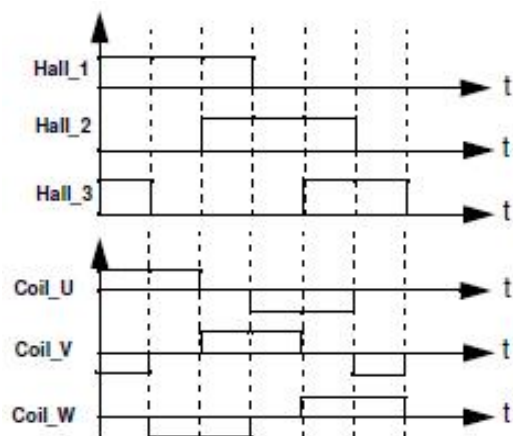
Gambar 2.19 (a) Struktur Motor *Brushless* dan (b) Konfigurasi Elektronik Stator



Gambar 2.20 Pensaklaran Motor *Brushless*

Sama halnya dengan motor AC 3 fasa, konfigurasi yang digunakan pada contoh motor *brushless* adalah konfigurasi bintang (*star*) 3 fasa. U, V dan W sebagai input untuk sumber yang disuplai dari rangkaian pensaklaran seperti pada Gambar 2.20.

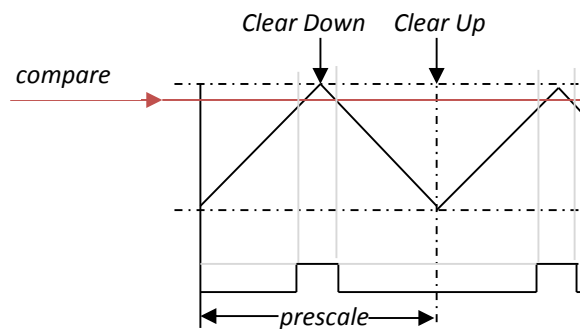
Untuk estimasi posisi rotor, motor dilengkapi tiga *hall sensor* yang ditempatkan setiap 120° sebagai penyearah dalam enam kondisi berbeda.



Gambar 2.21. Sinyal *Hall Sensor* Untuk Perputaran *propeller*

2.4.1 PWM (*Pulse Width Modulation*)

PWM merupakan suatu teknik teknik dalam mengatur kerja suatu peralatan yang memerlukan arus *pull in* yang besar dan untuk menghindari disipasi daya yang berlebihan dari peralatan yang akan dikontrol. PWM digunakan sebagai metoda untuk mengatur kecepatan perputaran motor dengan cara mengatur persentase lebar pulsa *high* terhadap perioda dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke motor sebagai sumber daya. Semakin besar perbandingan lama sinyal *high* dengan perioda sinyal maka semakin cepat motor berputar, perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri.



Gambar 2.22 Proses Pembangkitan Sinyal PWM

Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalkan suatu PWM memiliki resolusi 8 bit berarti PWM ini memiliki variasi perubahan nilai sebanyak $2 \text{ pangkat } 8 = 256$ variasi mulai dari 0 – 255 perubahan nilai. *Compare* adalah nilai pembanding, nilai ini merupakan nilai referensi *duty cycle* dari PWM tersebut dan nilai *compare* bervariasi sesuai dengan resolusi dari PWM. Pada Gambar 2.22, nilai *compare* ditandai dengan garis warna merah, dimana posisinya diantara dasar segitiga dan ujung segitiga. *Clear* digunakan untuk penentuan jenis komparator apakah komparator *inverting* atau *non-inverting*. Mikrokontroler akan membandingkan posisi keduanya, misalkan bila PWM diset pada kondisi *clear down*, berarti apabila garis segitiga berada dibawah garis merah (*compare*) maka PWM akan mengeluarkan logika 0. Begitu

pula sebaliknya apabila garis segitiga berada diatas garis merah (*compare*) maka PWM akan mengeluarkan logika 1. Lebar sempitnya logika 1 ditentukan oleh posisi *compare*, lebar sempitnya logika 1 itulah yang menjadi nilai keluaran PWM. Maka dari itu nilai *compare* inilah yang dijadikan nilai *duty cycle* PWM. *Clear Up* adalah kebalikan dari *Clear Down* pada keluaran logikanya.

Untuk perhitungan *duty cycle* dapat dilakukan dengan cara mengatur lebar pulsa “on” dan “off” dalam satu perioda gelombang melalui pemberian besar sinyal referensi *output* dari suatu PWM akan didapat *duty cycle* yang diinginkan. *Duty cycle* dari PWM dapat dinyatakan sebagai persamaan 2.12.

$$duty\ cycle = \frac{ton}{toff+ton} \times 100\% \quad (2.12)$$

dengan :

ton = lebar pulsa pada saat *on*

toff = lebar pulsa pada saat *off*

Dengan menghitung *duty cycle* yang diberikan, akan didapat tegangan *output* yang dihasilkan dengan menggunakan persamaan 2.13.

$$Average\ Voltage = \frac{a}{a+b} \times V_{full} \quad (2.13)$$

dengan :

a = nilai *duty cycle* pada saat *on*

b = nilai *duty cycle* pada saat *off*

V_{full} = Tegangan maksimum pada motor

Average voltage merupakan tegangan *output* pada motor yang dikontrol oleh sinyal PWM. *a* adalah nilai *duty cycle* saat kondisi sinyal “on”. *b* adalah nilai *duty cycle* saat kondisi sinyal “off”. *V_{full}* adalah tegangan maksimum pada motor. Dengan menggunakan rumus 2.13, maka akan didapatkan tegangan *output* sesuai dengan sinyal kontrol PWM yang dibangkitkan .

2.5 *Propeler (Baling-baling)*

Propeler (atau baling-baling) adalah baling baling untuk menjalankan pesawat terbang. Baling-baling ini memindahkan tenaga dengan mengkonversi gerakan rotasi menjadi daya dorong untuk menggerakkan sebuah kendaraan seperti pesawat terbang, untuk melalui suatu massa seperti udara, dengan memutar dua atau lebih bilah kembar dari sebuah poros utama. Sebuah propeler berperan sebagai sayap berputar, dan memproduksi gaya yang mengaplikasikan Prinsip *Bernoulli* dan Hukum gerak *Newton*, menghasilkan sebuah perbedaan tekanan antara permukaan depan dan belakang. Contoh gambar *propeller* ditunjukkan pada gambar 2.23 *Propeller* cw dan ccw. (Yulistiyanto, Andry:60:2013)



Gambar 2.23 Baling baling cw dan ccw

2.6 *Baterai Lithium Polimer*

Baterai *Lithium Polimer* atau biasa disebut dngan LiPo merupakan salah satu jenis baterai yang sering digunakan dalam dunia robot. Baterai Lipo dibuat dari beberapa kombinasi bahan: *Lithium-Cobalt*, *Lithium-Phospat* dan *Lithium-Mangan*. Kebanyakan baterai yang dijual kepada konsumen terbuat dari kombinasi *Lithium-Cobalt*. Contoh Baterai *Lithium Polimer* ditunjukkan dengan gambar 2.24 baterai *LiPo* 3sel.



Gambar 2.24 Baterai LiPo 3sell

Ada tiga kelebihan utama yang ditawarkan oleh baterai berjenis LiPo ketimbang baterai jenis lain seperti NiCad atau NiMH yaitu:

- Baterai *Lithium Polimer* memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran.
- Baterai *Lithium Polimer* memiliki kapasitas penyimpanan energi listrik yang besar.
- Baterai *Lithium Polimer* memiliki tingkat *discharge rate* energi yang tinggi, dimana hal ini sangat berguna sekali dalam bidang RC.

Selain keuntungan yang dimiliki, baterai jenis ini juga memiliki beberapa kelemahan yaitu:

- Harga baterai *Lithium Polimer* masih tergolong mahal jika dibandingkan dengan baterai jenis *NiCad* dan *NiMH*.

Performa yang tinggi dari baterai *Lithium Polimer* harus dibayar dengan umur yang lebih pendek. Usia baterai *Lithium Polimer* sekitar 300-400 kali siklus pengisian ulang. Sesuai dengan perlakuan yang diberikan pada baterai. (Susanto, Tri:28:2012)

2.7 Kamera

Kamera digunakan untuk memantau perangkat dari jauh. Dengan aplikasi *browser*, seorang dapat berkomunikasi dengan perangkat, kamera memiliki

ketajaman gambar 5 *Megapixel* dan nanti akan di letakan di bawah *frame quadcopter*.



Gambar 2.25 Kamera HD Wing

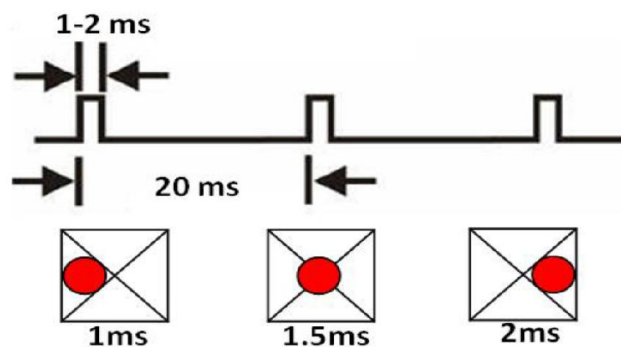
2.8 Remote Control dan Receiver 72MHz

Tx dan rx: *transmitter* dan *receiver* adalah pengirim data dan penerima data, data yang dikirim adalah data *ppm* atau *pcm*. Beberapa jenis *transmitter* berdasar dari frekuensi yang dipakai, jumlah *channel* (titik yang bias dikontrol) minimum untuk pesawat model adalah 3 *channel*, dan fasilitas penyimpanan data digital. Contoh gambar *remote control* ditunjukkan dengan Gambar 2.26 *Remote control* dan *Receiver 7 channel*.



Gambar 2.26 Remote control dan Receiver 7 channel

Receiver merupakan perangkat yang digunakan untuk menerima sinyal dari *radio transmitter* yang dikendalikan oleh pilot. Secara umum bentuk sinyal dari *radio receiver* ditunjukkan pada Gambar 2.27.



Gambar 2.27 Bentuk sinyal radio receiver dan posisi stick

Radio *transmitter* mengirimkan sinyal-sinyal sesuai dengan posisi-posisi dari tiap kanal. Bentuk sinyal yang dikirim tidak ada aturan baku yang mengatur sehingga perusahaan pembuat dapat membuat sinyal kirim sesuai dengan keinginan. Keluaran *radio receiver* dapat langsung digunakan untuk mengendalikan ESC karena sinyal merupakan sinyal standar dalam dunia RC. Pada penelitian ini keluaran *receiver* dihubungkan dengan perangkat modul mikrokontroler agar dapat digunakan untuk mengatur gerakan-gerakan *quadcopter* pada saat terbang secara manual. Remote *7 channel* ini memiliki 7 kanal, 4 buah digunakan untuk mengatur gerakan *roll*, *pitch*, *yaw*, dan *throttle*, dan 3 kanal sisa digunakan untuk *switch* dari mode manual ke *auto*. (Yulistiyanto, Andry:62:2013)

2.9 Arduino Leonardo

Arduino Leonardo adalah papan berbasis mikrokontroler pada ATmega32u4. Arduino Leonardo memiliki 20 digital input / output pin (7 dapat digunakan sebagai output PWM dan 12 sebagai input analog), 16 MHz osilator kristal, koneksi USB mikro, jack listrik, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler, terhubung ke komputer

dengan kabel USB, sumber tegangan dari adaptor AC-DC atau baterai untuk memulai. Modul Arduino Leonardo yang berbasis mikrokontroler ATmega32U4 memiliki Spesifikasi :

Mikrokontroler : ATmega32u4

Operasi Tegangan : 5V

Input Tegangan (disarankan) : 7-12V

Input Tegangan (batas) : 6-20V

Digital I / O Pins : 20

PWM Channels : 7

Analog Input Channels : 12

DC Current per I/O Pin : 40 mA

DC Current for 3.3V Pin : 50 mA

Flash Memory : 32 KB (ATmega32u4) yang 4 KB digunakan oleh bootloader

SRAM : 2,5 KB (ATmega32u4)

EEPROM : 1 KB (ATmega32u4)

Clock speed : 16 MHz (arduino.cc, diakses pada tanggal 25 april 2014)



Gambar 2.28 Arduino Leonardo

2.10 Bahasa Pemrograman C Pada Arduino

Bahasa pemrograman C merupakan bahasa utama yang digunakan pada arduino. *Coding* di dalam Arduino memiliki 3 (tiga) bagian utama, yaitu:

1. *void annexCode()*

Kode ini dieksekusi disetiap *loop* dan tidak akan mengganggu kontrol *loop* jika berlangsung kurang dari 650 mikrodetik. Di bawah ini adalah *void annexCode()* :

```
#define BREAKPOINT 1500
// PITCH & ROLL only dynamic PID adjustemnt, depending on
throttle value
if (rcData[THROTTLE]<BREAKPOINT) {
    prop2 = 100;
} else {
    if (rcData[THROTTLE]<2000) {
        prop2 = 100 - (uint16_t)conf.dynThrPID*(rcData[THROTTLE]-
BREAKPOINT)/(2000-BREAKPOINT);
    } else {
        prop2 = 100 - conf.dynThrPID;
    }
}
```

2. *void setup()*

Bagian ini merupakan inisialisasi yang diperlukan sebelum program utama dijalankan. Di bawah ini adalah *void setup()* :

```
static uint8_t rcDelayCommand; // this indicates the number of
time (multiple of RC measurement at 50Hz) the sticks must be
maintained to run or switch off motors
uint8_t axis,i;
int16_t error,errorAngle;
int16_t delta,deltaSum;
int16_t PTerm,ITerm,DTerm;
static int16_t lastGyro[3] = {0,0,0};
```

```

static int16_t delta1[3],delta2[3];
static int16_t errorGyroI[3] = {0,0,0};
static int16_t errorAngleI[2] = {0,0};
static uint32_t rcTime = 0;
static int16_t initialThrottleHold;

```

3. *void loop()*

Bagian ini menunjukkan jumlah waktu (beberapa pengukuran RC di 50Hz) yang harus dijaga untuk menjalankan atau mematikan motor. Di bawah ini adalah *void loop()* :

```

#define RC_FREQ 50
if (currentTime > rcTime ) { // 50Hz
    rcTime = currentTime + 20000;
    computeRC();
    // Failsafe routine - added by MIS
    #if defined(FAILSAFE)
        if ( failsafeCnt > (5*FAILSAFE_DELAY) && f.ARMED) {
// Stabilize, and set Throttle to specified level
            for(i=0; i<3; i++) rcData[i] = MIDRC;
// after specified guard time after RC signal is lost (in 0.1sec)
            rcData[THROTTLE] = FAILSAFE_THROTTLE;

```

Setiap modul Arduino menggunakan 2 (dua) buah koneksi dengan komputer, paralel dan serial. Komunikasi paralel digunakan untuk *burn bootloader*, program inisialisasi mikrokontroler agar dapat menggunakan koneksi serial dalam hal pemrograman, sehingga mikrokontroler dapat diprogram secara serial maupun komunikasi data antara mikrokontroler dan komputer menggunakan aplikasi Arduino. Selain itu, komunikasi paralel dapat juga digunakan untuk memprogram mikrokontroler secara paralel.

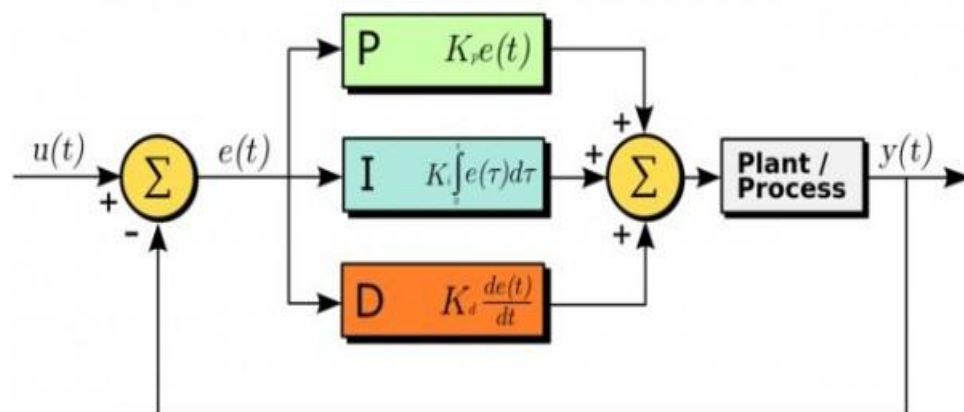
2.11 Algoritma PID (*Proportional Integral Derivate*)

Algoritma PID adalah algoritma kontrol dasar, "*PID adalah singkatan dari "proporsional, integral, turunan." Ketiga istilah menggambarkan elemen dasar dari sebuah kontroler PID. Masing-masing elemen ini melakukan tugas dan berbeda memiliki efek yang berbeda pada fungsi sistem"*(Wescot,Tim:86:2002). Algoritma PID sendiri sebenarnya merupakan gabungan dari beberapa aksi kontrol yaitu aksi kontrol proporsional ditambah integral ditambah turunan hal ini dapat dilihat pada rumus awal dari PID itu sendiri :

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

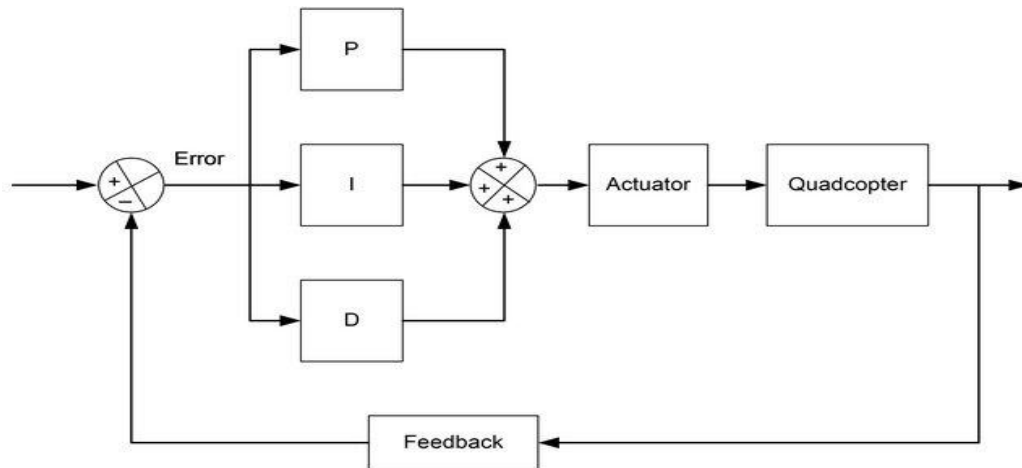
Dimana u adalah kontrol variable dan e adalah kontrol *error*. kontrol variabel merupakan hasil penambahan dari 3 bagian yaitu P(*Propotional error*), I(*integral Error*) dan D(*derrivative error*), dan parameter-paramater adalah K berfungsi sebagai parameter propotional, $\frac{K}{T_i}$ sebagai parameter integral, dan KT_d sebagai parameter *derrivative*(Astrom, Karl dan Tore Hagglund:64:2002).

Pada *quadcopter*, akan sangat memungkinkan pengguna untuk mengubah nilai-nilai konfigurasi PID untuk menyesuaikan kinerja *quadcopter* mereka. Gambar di bawah ini adalah diagram blok dari Kontroler PID.



Gambar 2.29 Diagram Blok Kontroler PID

PID (*proporsional-integral-derivatif*) merupakan sistem kontrol *loop* tertutup yang mencoba untuk mendapatkan hasil yang sebenarnya lebih dekat ke hasil yang diinginkan dengan menyesuaikan input. *Quadcopter* menggunakan PID *controller* untuk mencapai stabilitas. Di bawah ini merupakan gambar dari diagram PID *quadcopter*.



Gambar 2.30 Diagram Blok Kontroler PID Pada *Quadcopter*

2.11.1 Aksi kontrol Proposional (P)

Dalam kasus aksi kontrol proposional, hubungan antara masukan kontroler $u(t)$ dan sinyal pembangkit kesalahan $e(t)$ adalah

$$u(t) = K_p e(t)$$

atau dalam transformasi *laplace*

$$\frac{U(s)}{E(s)} = K_p$$

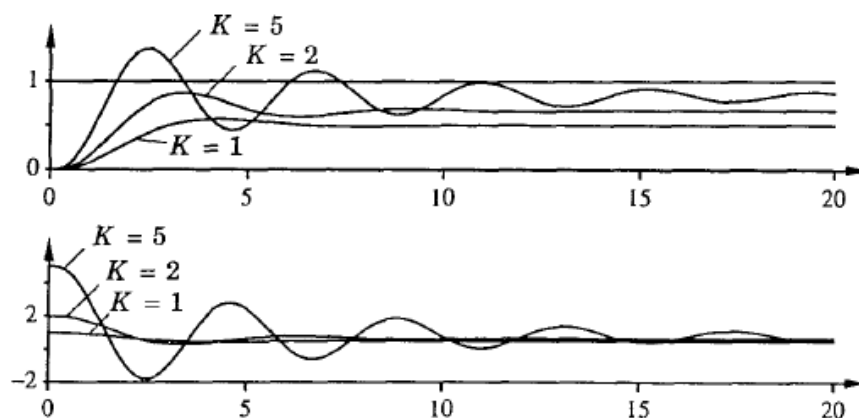
dengan K_p adalah suku penguatan Proposional.

Adapun pendapat lain dari aksi kontrol Proposional yaitu dalam kasus aksi kontrol proposional murni, persamaan nya menjadi:

$$u(t) = K_p e(t) + u_b$$

aksi kontrol pada persamaan ini hanya proportional terhadap kontrol *error*.

variabel u_b adalah nilai anggapan awal. Sehingga apabila kontro $e = 0$ maka $u(t) = u_b$. Nilai anggapan u_b biasanya dibuat *fix* dengan nilai $(u_{\max} + u_{\min})/2$. Tetapi terkadang nilai tersebut diatur secara manual (Astrom, Karl dan Tore Hagglund, 65, 2002). Pada dasarnya menurut pendapat Karl Astrom dan katsuhiko ogata hampir sama hanya terdapat perbedaan pada bagian u_b yang pada teori kedua dimasukan kedalam formula sebagai nilai set awal yang akan diberikan apabila terjadi $e=0$. Aksi kontrol propotional menggunakan parameter K_p sebagai bantuan dalam menentukan aksi respon sistem dengan tingkat error yang berbeda-beda, dengan kata lain dalam aksi kontrol proposional nilai keluaran akan menjadi berbanding lurus dengan tingkat *error* yang terjadi dengan parameter K_p dijadikan sebagai faktor pengali.



Gambar 2.31 Grafik simulasi nilai input K/K_p

pada gambar diatas adalah hasil simulasi dengan memasukan nilai K/K_p pada suatu proses (Astrom, Karl dan Tore Hagglund:66:2002)

2.11.2 Aksi Kontrol Integral

Aksi kontrol integral merupakan salah satu dari 3 aksi kontrol yang merupakan bagian dari PID. Fungsi dari aksi kontrol integral adalah memastikan *output* dari proses sesuai dengan set point saat *steady state*. Dengan aksi kontrol proposional, pada normal nya *control error* berada pada *steady state*. Dengan aksi

kontrol Integral kesalahan positif yang kecil akan selalu menaikkan nilai sinyal kontrol dan sebaliknya kesalahan negatis yang kecil akan selalu menurunkan nilai sinyal kontrol sekecil apapun kesalahan yang terjadi (Astrom, Karl dan Tore Hagglund:67:2002).

Pada kontroler dengan aksi kontrol integral nilai masukan kontroler $u(t)$ diubah pada laju proposional dari sinyal pembangkit kesalahan $e(t)$. sehingga

$$\frac{du(t)}{dt} = K_i e(t)$$

atau dalam besaran transformasi *Laplace*

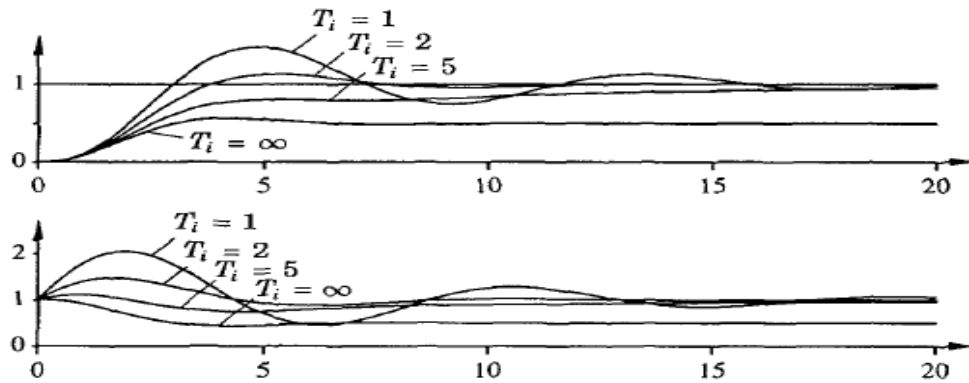
$$u(t) = K_i \int_0^t e(t) dt$$

dengan K_i adalah konstanta yang dapat diubah. Fungsi alih dari kontroler integral adalah

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

jika nilai $e(t)$ ada dua kali lipat, maka nilai $u(t)$ bervariasi dua kali secara cepat. Untuk pembangkit kesalahan nol, nilai $u(t)$ tetap konstan (Katsuhiko Ogata:202:1997).

Dalam aksi kontrol integral yang ingin diatasi adalah overshoot dari keluaran program. pada aksi kontrol proposional nilai *error* yang kecil akan memberikan perubahan yang berbanding lurus dengan keluaran sesuai dengan faktor pengali K_p pada subbab sebelum nya hal ini tentu saja akan mengakibatkan banyak perubahan-perubahan mendadak apabila terjadi perubahan kecil, karena itu lah diperlukan aksi kontrol integral untuk memberikan output proses dengan cepat pada kondisi-kondisi tersebut. berikut merupakan gambaran grafik dari proses input nilai T_i



Gambar 2.32 Grafik simulasi nilai input T_i

2.11.3 Aksi kontrol Proposional ditambah integral

Aksi kontrol ini merupakan aksi kontrol yang merupakan hasil penjumlahan dari perhitungan aksi kontrol proposional dan integral. Aksi kontrol proposional ditambah integral didefinisikan pada persamaan berikut:

$$u(t) = Kp e(t) + \frac{Kp}{T_i} \int_0^t e(t) dt$$

atau fungsi alih kontroler ini adalah

$$\frac{U(s)}{E(s)} = Kp \left[1 + \frac{1}{T_i s} \right]$$

dengan Kp penguatan proposional dan T_i disebut waktu integral. Kedua parameter yaitu Kp dan T_i bisa diinput secara manual. Pada perumusan aksi kontrol ini ada juga pendapat lain yaitu berasal dari Astrom, Karl dan Tore Hagglund yaitu:

$$u_0 = K(e_0 \frac{e_0}{T_i} t)$$

selama e_0 tidak sama dengan 0, ini tentu saja memberikan asumsi bahwa kontrol sinyal u_0 berada dalam kondisi konstan.

2.11.4 Aksi kontrol proposional ditambah turunan

Aksi kontrol proposional ditambah integral adalah gabungan dari aksi kontrol proposional dan turunan, dimana tujuan dari aksi kontrol turunan adalah untuk meningkatkan stabilitas dari *close-loop* proses sistem. ketidak stabilan dapat dijelaskan sebagai berikut. Dalam proses pergerakan dinamis, system membutuhkan waktu sebelum mendeteksi perubahan variable pada sistem kontrol hal ini mengakibatkan sistem kontrol akan terlambat dalam mengoreksi atau melakukan tindakan perubahan dalam upaya memperbaiki kesalahan, maka dalam aksi kontrol proposional ditambah *derivative*, sistem kontrol aksi proposional akan dibuat bisa memprediksikan output keluaran walaupun pada proses nya prediksi tetap akan mengalami kesalahan yang membentuk kurva (Astrom, Karl dan Tore Hagglund:69:2002). Adapun rumus dari aksi kontrol proposional ditambah turunan adalah sebagai berikut :

$$u(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right)$$

dalam ekspansi deret *taylor* $e(t + T_d)$ memberikan nilai

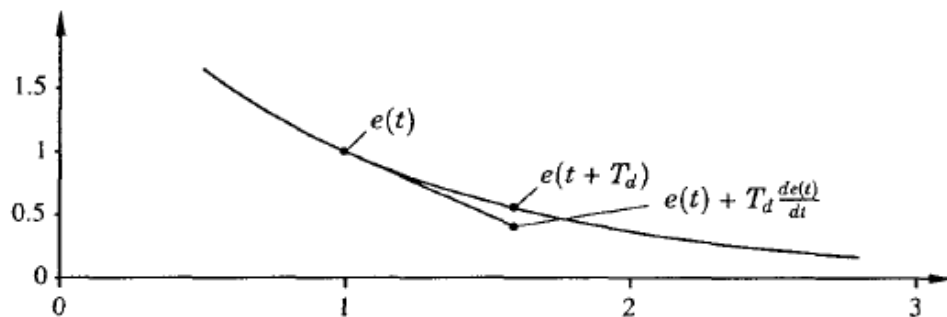
$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}$$

dan fungsi alihnya adalah

$$\frac{U(s)}{E(s)} = K_p(1 + T_d s)$$

dengan K_p adalah penguatan proposional dan T_d adalah konstanta yang disebut dengan waktu turunan. Dalam logika dasar orang awam aksi control turunan digunakan sebagai pengoreksi *error* kesalahan yang telah terjadi pada saat dilakukan aksi kontrol proposional sehingga tujuan utama dari aksi kontrol turunan adalah memperbaiki stabilitas dari aksi kontrol proposional yang terlalu responsif. Dapat diperhatikan bahwa aksi kontrol turunan tidak pernah digunakan

sendiri karena aksi kontrol ini hanya efektif selama periode transient(katsuhiko ogata, 203, 1997).



Gambar 2.33 Grafik interpretasi aksi turunan

grafik diatas adalah gambaran grafik interpretasi dari aksi turunan sebagai prediksi kontrol dimana prediksi tersebut didapatkan dari *linear extrapolation*(Astrom, Karl dan Tore Hagglund:69:2002).

2.11.5 Aksi kontrol proposional ditambah integral ditambah turunan

Aksi kontrol ini merupakan aksi kontrol gabungan antara 3 aksi kontrol, sehingga hasil nya aksi kontrol ini memiliki keuntungan dari masing-masing aksi kontrol yang sebelum nya, dan juga aksi kontrol gabungan ini juga mengurangi kekurangan dari masing-masing aksi kontrol, aksi kontrol proposional yang mengakibatkan efek ketidak stabilan pada sistem kontrol direduksi oleh aksi kontrol turunan. aksi kontrol integral yang memberikan respon yang baik juga akan meningkat efektifitas respon dari sistem.

persamaan 3 kombinasi menjadi:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

atau fungsi alih nya

$$\frac{u(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} + T_d s \right]$$

dengan Kp penguatan Proporsional, Ti waktu integral dan Td adalah waktu turunan.(katsuhiko ogata:204:1997)

pada pengembangan aksi kontrol PID terdapat banyak modifikasi salah satu nya iyalah model alternative representations.

Dalam model ini rumusan algoritma PID direpresentasikan dengan rumus berikut.

$$G(s) = Kp \left(1 + \frac{1}{sT_i} + sT_d \right)$$

pada penggunaan di komersil kontrol, aksi kontrol dideskripsikan pada rumus berikut

$$G'(s) = Kp' \left(1 + \frac{1}{sT_i'} \right) (1 + sT_d')$$

dan pada modifikasi yang lain rumus awal PID direpresentasikan sebagai berikut

$$G''(s) = Kp + \frac{Ki}{s} + sKd$$

dengan parameter yang dimodifikasi adalah

$$K_P = K_P$$

$$K_i = \frac{K}{T_i}$$

$$K_d = K T_d$$

dan dari persamaan diatas nilai yang ditentukan sebagai parameter inputan dalam proses peningkatan stabilitas, respon, dan ketepatan iyalah Kp, Ki, dan Kd dengan fungsi peningkatan seperti yang telah dijelaskan pada masing-masing aksi kontrol tersebut. (katsuhiko ogata:204: 1997)