

Perancangan dan Implementasi Sistem Kendali PID untuk Pengendalian Gerakan *Hover* pada UAV *Quadcopter*

Ardy Seto Priambodo, Katjuk Astrowulan, Joko Susila

Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: seto_ardy@yahoo.co.id, katjuk@ee.its.ac.id, jokosus@ee.its.ac.id

Abstrak— Sistem kendali banyak memegang peranan penting dalam ilmu pengetahuan, teknologi maupun industri. Penelitian yang ditujukan untuk pengembangan sistem kendali sampai sekarang masih terus dilakukan untuk menjawab kebutuhan akan sistem kendali yang semakin kompleks. *Quadcopter* merupakan salah satu jenis *rotorcraft* yang memiliki 4 buah *rotor* pada ujung-ujung *frame*. Untuk menggerakkan *quadcopter*, yang harus dilakukan adalah mengatur kecepatan putaran tiap motor. Secara teori *quadcopter* dapat terbang melayang pada satu titik, kegiatan ini dinamakan *hover*. Untuk menghasilkan gerakan *hover* pada *quadcopter*, kecepatan motor harus dikendalikan pada daerah kecepatan nominal. Pada tugas akhir kali ini untuk mengatur gerakan *hover* dari *quadcopter* digunakan sistem kendali PD yang memiliki konstanta K_p dan K_d . Ada 3 buah kontroler individual yaitu kontroler PD *roll*, kontroler PD *pitch*, dan kontroler PD ketinggian. Perancangan sistem kendali PD pada simulasi yang digunakan untuk mempertahankan posisi *hover* pada *quadcopter* menghasilkan respon ketinggian dan sudut *roll* & *pitch* dengan besar *error steady state* = 0. Nilai parameter yang didapatkan dari hasil *tuning* terstruktur pada simulasi adalah pada kontroler *roll* & *pitch* $K_p=40$, $K_d=5$ dan pada kontroler ketinggian $K_p=35$, $K_d=17$. Respon hasil implementasi pada *quadcopter* belum sesuai pada hasil simulasi. Masih terdapat osilasi pada respon ketinggian yang besarnya +/- 15 cm disekitar *set point* dikarenakan pembacaan sensor ketinggian yang kadang tidak sesuai dengan ketinggian terukur. Selain itu koreksi sudut *roll* & *pitch* pada kontroler *roll* & *pitch* juga masih terdapat error +/- 5°.

Kata Kunci— *Quadcopter*, Kontrol PID, *Hover*, *Roll*, *Pitch*.

I. PENDAHULUAN

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 lainnya. Kelebihan dari UAV adalah dapat digunakan pada misi-misi berbahaya tanpa membahayakan pilot / manusia.

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.

Saat sekarang ini *quadrotor* atau *quadcopter* banyak menjadi obyek penelitian. *Quadcopter* berukuran kecil / mini yang dikendalikan melalui *remote control* banyak digunakan 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.

II. DASAR TEORI

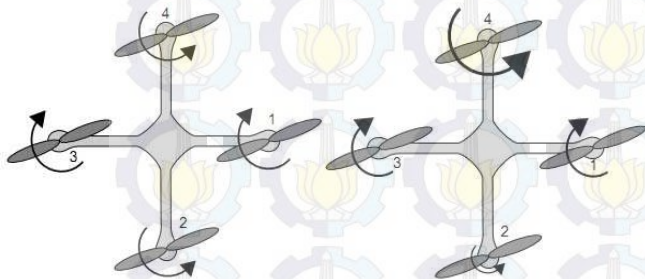
A. *Quadcopter*

Quadcopter memiliki 4 buah rotor sebagai penggerak baling-baling yang digunakan untuk menghasilkan gaya angkat. Untuk bergerak naik, diperlukan kecepatan yang sama dan cukup besar pada keempat rotornya. Secara garis besar, gerakan *quadcopter* terbagi menjadi empat.

Gerakan Hover

Pergerakan ini dilakukan dengan memberikan kecepatan motor 1-4 dengan besaran yang sama dengan kecepatan nominal. Ketika kecepatan motor 1-4 melebihi kecepatan nominal maka *quadcopter* akan bergerak naik, dan sebaliknya ketika kecepatan motor 1-4 lebih kecil dari kecepatan nominal

maka ketinggian *quadcopter* akan menurun. Pergerakan *quadcopter* pada keadaan *hover* ditunjukkan pada Gambar 1a.



Gambar. 1. (a) *Quadcopter* Keadaan *Hover* (b) *Quadcopter* Keadaan *Roll*

Gerakan Roll

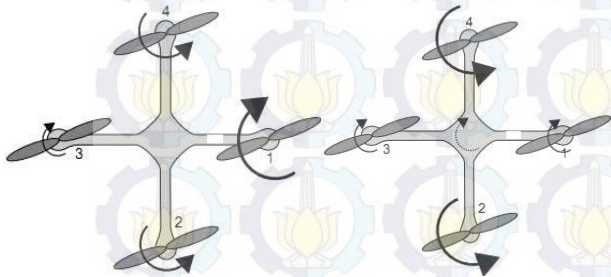
Pergerakan ini dilakukan dengan cara menambahkan (atau mengurangi) kecepatan putar motor 2 pada *quadcopter* dan bersamaan dengan itu, menurunkan (atau menaikkan) kecepatan motor 4. Gerakan ini bergerak dengan acuan pada sumbu Y. Gerakan *roll* kekanan ditunjukkan pada Gambar 1b.

Gerakan Pitch

Pergerakan ini dilakukan dengan cara menambahkan (atau mengurangi) kecepatan putar motor 1 pada *quadcopter* dan bersamaan dengan itu, menurunkan (atau menaikkan) kecepatan motor 3. Gerakan ini bergerak dengan acuan pada sumbu X. Gerakan *pitch* ke belakang ditunjukkan pada Gambar 2a.

Gerakan Yaw

Pergerakan ini dilakukan dengan cara meningkatkan (atau menurunkan) kecepatan putar motor 2 dan 4 pada *quadcopter* dan bersamaan dengan itu, menurunkan (atau menaikkan) kecepatan motor 1 dan 3. Gerakan ini berputar dengan acuan pada sumbu Z. Gerakan *yaw* searah jarum jam ditunjukkan pada Gambar 2b.



Gambar. 2. (a) *Quadcopter* Keadaan *Pitch* (b) *Quadcopter* Keadaan *Yaw*

Gerakan translasi dari *quadcopter* dapat dituliskan ke dalam persamaan seperti yang ditunjukkan pada Persamaan 1-3.

$$\ddot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m} \quad (1)$$

$$\ddot{y} = (\cos \phi \sin \theta \cos \psi - \sin \phi \sin \psi) \frac{U_1}{m} \quad (2)$$

$$\ddot{z} = -g + (\cos \phi \cos \theta) \frac{U_1}{m} \quad (3)$$

Sedangkan gerakan rotasi dari *quadcopter* yang terdiri dari rotasi sudut *roll*, *pitch*, dan *yaw* ditunjukkan oleh Persamaan 4-6.

$$\ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr + \frac{J_r}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \quad (4)$$

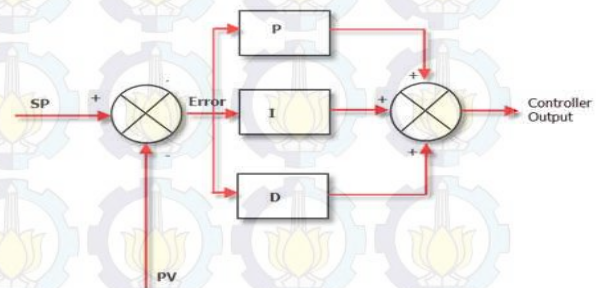
$$\ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr - \frac{J_r}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \quad (5)$$

$$\ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \quad (6)$$

B. Kontroler PID

Pengendali PID dibentuk dengan menggabungkan karakteristik yang ada pada kontroler P (Proporsional), kontroler I (Integral), dan kontroler D (Derivatif). Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel sebagai kontroler PID.

Blok diagram secara lengkap kontroler PID ditunjukkan pada Gambar 3 sedangkan persamaan matematis kendali PID ditunjukkan pada Persamaan 7.



Gambar. 3. Diagram Blok Kontroler PID

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proposional + integral + derivatif (kontroler PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar.

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de(t)}{dt} \right) \quad (7)$$

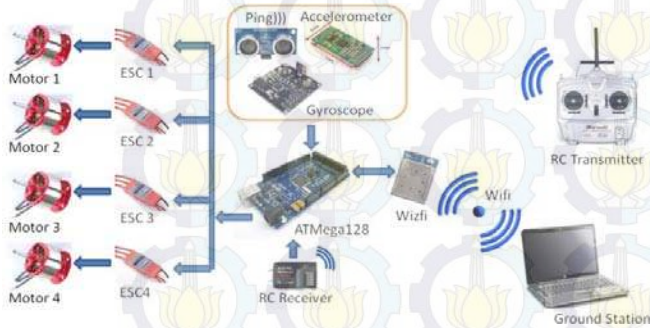
III. PERANCANGAN QUADCOPTER

A. Perancangan Sistem Elektronik

Sistem elektronik yang ada pada *quadcopter* terdiri atas mikrokontroler dan beberapa sensor yang digunakan sebagai acuan dalam menentukan gerak terbangnya. Sensor yang

digunakan adalah sensor ketinggian yang diukur menggunakan sensor ultrasonik ping))), sensor sudut yang diukur menggunakan sensor *accelerometer*, dan terakhir sensor kecepatan sudut yang diukur menggunakan sensor *gyroscope*.

Rangkaian elektronika yang dirancang harus mampu menangani jumlah *input/output* sensor dan komponen-komponen yang digunakan dalam penunjang terbang *quadcopter*. Beberapa komponen seperti sensor-sensor, ESC, dan motor merupakan modul yang dapat langsung digunakan ketika dibeli dipasaran. Perancangan keseluruhan dari sistem elektronika dari *quadcopter* dapat dilihat pada Gambar 4.



Gambar. 4. Perancangan Sistem Elektronika *Quadcopter*

B. Perancangan Kontroler

Model yang dirancang memiliki 12 keluaran yaitu 3 kecepatan translasi ($\dot{x}, \dot{y}, \dot{z}$), 3 posisi linier (x, y, z), 3 kecepatan rotasi (p, q, r) dan terakhir 3 posisi sudut (ϕ, θ, ψ). Keluaran-keluaran inilah nantinya akan digunakan sebagai umpan balik menuju kontroler agar kontroler dapat melakukan koreksi terhadap kesalahan. Dalam mempertahankan posisi *hover* ada 3 bagian penting yang harus dijaga kestabilannya, yaitu sudut *roll*, sudut *pitch*, dan ketinggian *quadcopter* sehingga tidak semua 12 keluaran dari *quadcopter* digunakan semua.

Kontroler PD digital merupakan bentuk lain dari kontroler PD yang diprogram dan dijalankan menggunakan pemrograman komputer. Untuk dapat menjalankan kontroler PD pada simulasi, maka kontroler PD analog harus diubah terlebih dahulu ke bentuk digital. Penurunan persamaan kontroler PD analog menjadi digital dapat dilihat pada Persamaan 8-10. Bentuk umum persamaan matematis dari kontroler PD ditunjukkan pada Persamaan 8.

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (8)$$

Bentuk persamaan diferensial dapat ditulis dalam bentuk diskrit seperti pada Persamaan 9.

$$\frac{de(t)}{dt} = \frac{e_k - e_{k-1}}{T} \quad (9)$$

Sehingga diperoleh kontroler PD dalam bentuk diskrit adalah seperti ditunjukkan pada Persamaan 10.

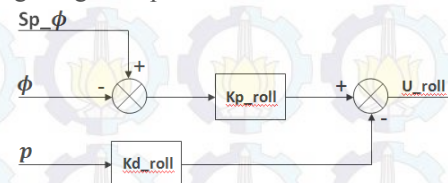
$$u(k) = K_p e_k + \frac{1}{T} K_d (e_k - e_{k-1}) \quad (10)$$

dimana T adalah *time sampling* dari sistem yang digunakan.

Kontroler Sudut Roll & Pitch

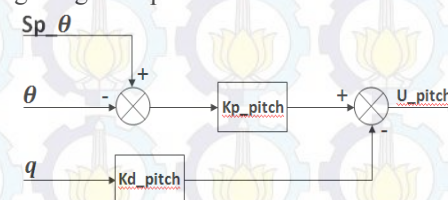
Untuk mengendalikan sudut *roll* dan *pitch* pada *quadcopter* dapat dilakukan secara independen atau sendiri-sendiri. Sesuai dengan pergerakan yang terjadi pada *quadcopter* ada 2 sudut yang diatur ketika *hover*, yaitu sudut *roll* dan sudut *pitch*. Sudut *roll* berubah ketika ada perubahan pada motor 2 dan 4 sedangkan sudut *pitch* berubah ketika ada perubahan pada motor 1 dan 3.

Pada Gambar 5 menggambarkan mengenai blok diagram kontroler pada sudut *roll*, Sp_ϕ merepresentasikan sudut *roll* yang diinginkan, ϕ adalah sudut *roll* terukur, dan p adalah kecepatan *angular* atau *roll rate*. Kp_roll dan Kd_roll adalah parameter proporsional dan derivatif dari kontroler PD. Karena kontroler D (differensial) menggunakan masukan kecepatan *angular* maka hasilnya dari kontroler ini dikurangkan. Hasil sinyal kontrol direpresentasikan pada variabel U_roll yang efeknya menambahkan kecepatan motor 2 dan mengurangi kecepatan motor 4.



Gambar. 5. Diagram Blok Kontrol Roll

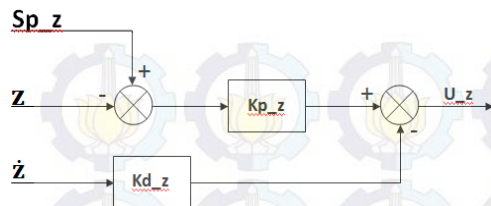
Pada Gambar 6 menggambarkan mengenai blok diagram kontroler pada sudut *pitch*, Sp_θ merepresentasikan sudut *pitch* yang diinginkan, θ adalah sudut *pitch* terukur, dan q adalah kecepatan *angular* atau *pitch rate*. Kp_pitch dan Kd_pitch adalah parameter proporsional dan derivatif dari kontroler PD. Karena kontroler D (differensial) menggunakan masukan kecepatan *angular* maka hasilnya dari kontroler ini dikurangkan. Hasil sinyal kontrol direpresentasikan pada variabel U_pitch yang efeknya menambahkan kecepatan motor 1 dan mengurangi kecepatan motor 3.



Gambar. 6. Diagram Blok Kontrol Pitch

Kontroler Ketinggian

Pada Gambar 7 menampilkan blok diagram kontroler pada ketinggian, dimana Sp_z merepresentasikan ketinggian yang diinginkan, z adalah ketinggian yang terukur, dan \dot{z} adalah kecepatan pada sumbu Z. Kp_z dan Kd_z adalah parameter proporsional dan derivatif dari kontroler PD. Karena kontroler D (derivatif) menggunakan masukan kecepatan naik/turun dari *quadcopter* maka hasilnya dari kontroler ini dikurangkan. Hasil sinyal kontrol direpresentasikan pada variabel U_z . Hasil sinyal kontrol ini dijumlahkan kepada semua motor 1,2,3, dan 4.

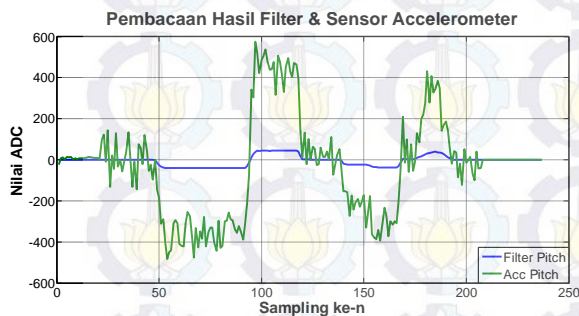


Gambar. 7. Diagram Blok Kontroler Ketinggian

IV. PENGUJIAN DAN ANALISIS

A. Pengujian Sensor Gyroscope + Accelerometer

Complementary filter disini lebih memperbaiki pembacaan sensor *gyroscope* dan *accelerometer* dari *noise* akibat getaran putaran motor. Terlihat pada Gambar 8 adalah hasil pembacaan dari *complementary filter* yang apabila dibandingkan dengan hasil pada sensor *accelerometer* saja, sudah tidak ber-*noise*. Hasil pembacaan *complementary filter* dapat dikatakan cukup baik dalam mengurangi *noise* pada pembacaan sensor *accelerometer*. Saat tidak ada pergerakan, pada keluaran sensor *accelerometer* masih terdapat nilai yang jauh dari 0, namun pada keluaran *filter* hasil pembacaannya cukup baik yaitu disekitar 0. Saat ada perubahan sudut *pitch*, hasil pembacaan pada sensor *accelerometer* sangat tidak konsisten namun hasil pembacaan *filter* konsisten dengan pembacaan sudut yang berubah.



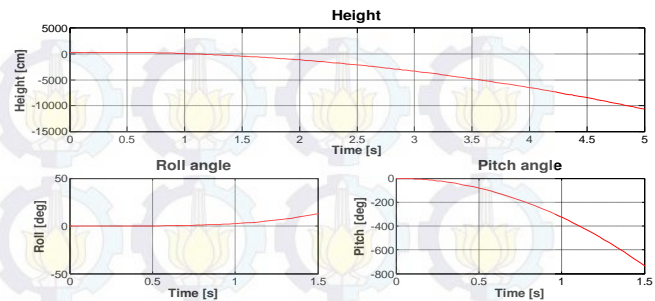
Gambar. 8. Pembacaan Hasil filter & sensor accelerometer

B. Pengujian Open Loop

Pengujian dilakukan pada simulasi dengan pemberian sinyal kecepatan motor secara langsung atau *open loop*. Hasil pembacaan dilihat pada sudut *roll*, sudut *pitch*, dan ketinggian dari *quadcopter*. Berikut adalah beberapa pengujian yang dilakukan.

Dengan member sinyal masukan pada motor 2 dan 4 sebesar 91 kemudian motor 1 sebesar 90,5 dan motor 3 sebesar 91,5 maka terlihat terjadi perubahan sudut *pitch*. Perbedaan kecepatan motor 1 dan motor 3 menyebabkan perubahan sudut *pitch*. Pada sudut *roll* juga terkena efek akibat perbedaan kecepatan motor 1 dan 3, tapi perubahan sudut *roll* tidak besar. Pada detik ke-1.5 terlihat nilai sudut *pitch* sudah hampir mencapai -800 derajat, sedangkan sudut *roll*-nya terbaca sekitar 12 derajat. Akibat sudut *pitch* berubah maka akan mempengaruhi ketinggian dari *quadcopter* menjadi menurun.

Pada detik ke-1.5 *quadcopter* sudah turun sekitar 500 cm. Pengujian dilakukan pada berbagai macam kondisi masukkan dan diperhatikan respon pada sudut *roll*, sudut *pitch*, dan ketinggian yang terjadi.

Gambar. 9. Respon ketinggian, sudut roll & pitch pada pengujian *open loop*

C. Pengujian Closed Loop

Pada simulasi *closed loop* atau simulasi model dengan kontroler ada 3 bagian yang akan dijelaskan analisisnya. Analisa pertama adalah mengenai analisa kontroler *roll quadcopter*, analisa kontroler *pitch quadcopter*, dan analisa kontroler ketinggian *quadcopter*.

Analisis Kontroler Roll & Pitch

Roll dan *pitch* dalam kasus ini adalah sama karena bentuk *quadcopter* yang simetris, hanya saja arahnya yang berbeda. Sehingga kontroler yang digunakan sejenis dengan besar parameter yang digunakan sama pula. *Set point* dari sudut *roll* diberi nilai 0 derajat dan *set point* dari sudut *pitch* diberi nilai 0 derajat. *Initial condition* dari sudut *roll* dan *pitch* adalah 15 derajat.

Karena kontroler yang digunakan adalah kontroler PD maka ada 2 parameter yang harus diatur nilainya agar respon yang dihasilkan sesuai dengan keinginan. Langkah-langkah dari *tuning* parameter *Kp* (konstanta proporsional) dan *Kd* (konstanta diferensial) dengan cara terstruktur atau eksperimen adalah sebagai berikut :

- 1) Langkah awal adalah menggunakan kontroler proporsional terlebih dahulu, sehingga *Kp* diberi suatu nilai tertentu (jangan terlalu besar) dan *Kd* diberi 0.
- 2) Naikkan pemberian nilai *Kp* hingga respon berosilasi pada sekitar set point. Tapi perlu diperhatikan, koreksi dari sudut *roll* dan *pitch* pada *quadcopter* harus cepat dan tepat sehingga respon yang diharapkan pada langkah ini harus sesuai.

Tabel 1.
Nilai *tuning* kontroler PD sudut *roll & pitch*

Kp	Kd	Respon Sudut roll & pitch
5	0	Respon lambat dan terjadi osilasi
13	0	Respon terlihat meningkat dan terjadi osilasi
27	0	Respon cukup cepat tapi belum sesuai yang diinginkan dan osilasi juga terjadi
40	0	Respon sesuai dengan kriteria yang diinginkan namun tetap terjadi osilasi
40	1	Respon cepat tapi terjadi overshoot
40	5	Respon cepat tanpa overshoot

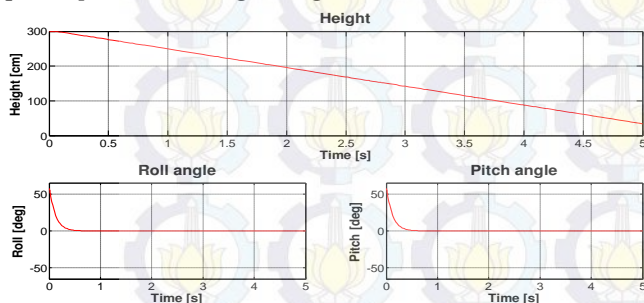
- 3) Untuk meredam osilasi, tambahkan *Kd* dengan besaran nilai tertentu (dimulai dari yang kecil) sehingga apabila

diamati respon yang terjadi pada *quadcopter* hingga stabil dan lebih responsif.

- 4) Periksa kembali performa sistem hingga mendapatkan hasil sesuai keinginan.

Pada Tabel 1 merupakan hasil *tuning* nilai-nilai parameter kontroler PD sudut *roll* dan *pitch quadcopter* dengan *initial condition* sudut *roll* dan *pitch* adalah 15 derajat, kemudian diatur *set point* sudut *roll* = 0 derajat, *set point* sudut *pitch* = 0 derajat.

Respon hasil *tuning* kontroler PD pada $K_p = 40$ dan $K_d = 5$ terlihat pada Gambar 10. Terlihat waktu yang dibutuhkan untuk mencapai keadaan tunak cukup cepat, sekitar kurang dari setengah detik. Pada analisa ini ketinggian tidak digunakan sehingga tidak ada koreksi ketinggian *quadcopter*. Koreksi sudut ini diperlukan waktu respon yang cepat agar *quadcopter* tidak mengalami penurunan secara terus menerus.



Gambar. 10. Respon sudut *roll* & *pitch* pada nilai $K_p = 40$ dan $K_d = 5$

Analisis Kontroler Ketinggian

Kondisi awal ketinggian dari *quadcopter* diatur pada ketinggian 2 m, sehingga diharapkan *quadcopter* dapat menyesuaikan *set point* yang diberikan dalam rentang waktu tertentu yang dirasa sesuai. Kontroler yang digunakan untuk mengatur ketinggian sama pada kontroler yang digunakan untuk mengatur sudut *roll* & *pitch* yaitu kontroler PD. Dimana kontroler PD memiliki 2 buah parameter yang harus di-*tuning* agar mendapatkan respon yang diharapkan.

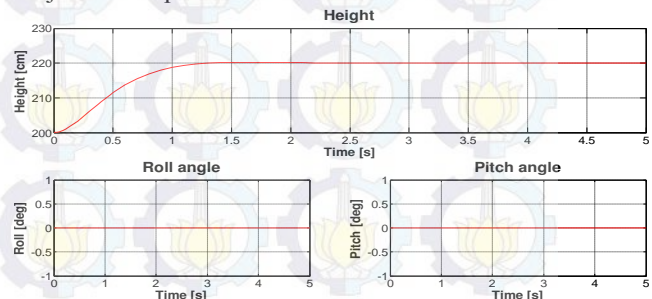
Tabel 2.

Nilai <i>tuning</i> kontroler PD ketinggian		
K_p	K_d	Respon Sudut ketinggian
5	0	Respon lambat dan terjadi osilasi
13	0	Respon terlihat meningkat namun masih terjadi osilasi
27	0	Respon cukup cepat tapi belum sesuai yang diinginkan dan osilasi juga terjadi
35	0	Respon sesuai dengan kriteria yang diinginkan namun tetap terjadi osilasi
35	1	Respon cepat tapi masih terjadi osilasi
35	10	Respon cepat dan terjadi overshoot
35	17	Respon cepat sesuai yang diinginkan dan tanpa overshoot

Untuk mendapatkan parameter K_p dan K_d dari kontroler ketinggian ini juga digunakan cara *tuning* terstruktur atau biasa disebut *tuning* secara eksperimen. Hasil *tuning* yang

didapatkan dari langkah-langkah yang dilakukan adalah $K_p = 35$ dan $K_d = 17$. Respon ketinggian yang dihasilkan menggunakan nilai parameter ini ditunjukkan pada Gambar 11.

Terlihat dengan pemberian kontroler derivatif dapat meredam osilasi yang ditimbulkan oleh kontroler proporsional sehingga respon yang didapatkan menjadi lebih baik dan menyerupai orde 1. Pemilihan nilai K_p juga tidak boleh terlalu besar karena dapat menyebabkan osilasi berlebih yang menjadikan respon tidak stabil.

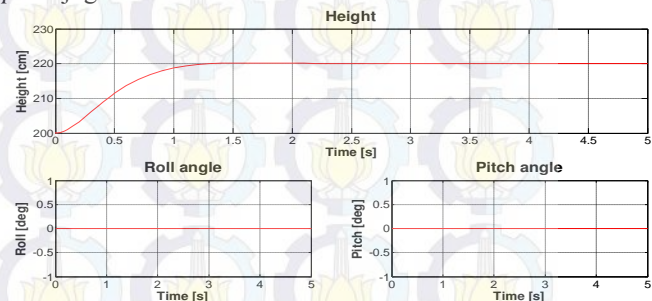


Gambar. 11. Respon ketinggian pada nilai $K_p = 35$ dan $K_d = 17$

Analisis Kontroler Roll & Pitch & Ketinggian

Dengan penggabungan kontroler ketinggian dan sudut *roll* & *pitch* diharapkan *quadcopter* dapat melakukan *hover* pada ketinggian tertentu dengan *set point* sudut *roll* & *pitch* adalah 0 derajat. Dengan kondisi awal ketinggian *quadcopter* adalah 2 meter maka *quadcopter* dengan kontroler PD mampu menjaga ketinggian menjadi 2 meter dalam waktu tertentu. Karena sudut sangat mempengaruhi ketinggian, maka kontroler PD sudut *roll* & *pitch* harus cepat mengatasi *error*. Gambar 12 menampilkan respon *quadcopter* tanpa adanya gangguan.

Untuk menguji apakah kontroler yang digunakan dapat menanggulangi gangguan, maka *quadcopter* diberi gangguan berupa gangguan sudut *roll*, sudut *pitch*, dan gangguan ketinggian berupa unit step dengan nilai tertentu. Selain itu untuk mendekati *plant* yang sesungguhnya pada sudut *roll* & *pitch* juga diberi *noise*.

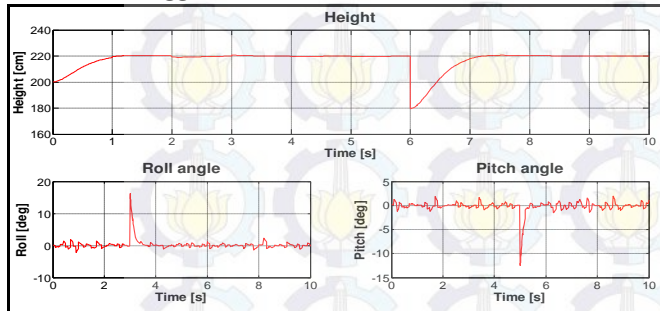


Gambar. 12. Respon ketinggian dan sudut *roll* & *pitch* dengan kontroler PD

Pada detik ke-3 pada sudut *roll* ada gangguan berupa unit step dengan besaran 16 derajat. Dengan cepat kontroler PD yang ada pada sudut *roll* bekerja memperbaiki respon dan terlihat pada detik ke 3,6 pembacaan sudut *roll* sudah pada posisi 0 derajat. Selain pada sudut *roll*, gangguan juga diberi pada sudut *pitch* di detik ke-5. Gangguan berupa unit step dengan *final value* -12 derajat. Terlihat pada detik ke-5 pembacaan sudut *pitch* mengalami penurunan hingga -12

derajat dan pada detik ke 5,4 gangguan sudah dapat diatasi dengan baik. Perubahan pada sudut *roll* & *pitch* sebenarnya mengakibatkan perubahan ketinggian dari *quadcopter*, namun karena kontroler pada sudut bekerja dengan baik dan cepat maka efek penurunan ketinggian yang dihasilkan tidak terlihat.

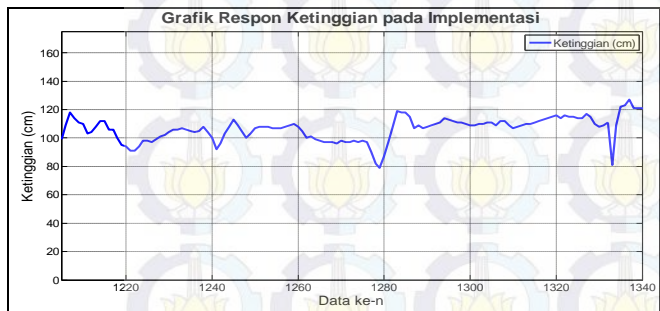
Gangguan berupa unit step juga diberikan pada ketinggian, terlihat pada Gambar 13 ketinggian di detik ke-6 mengalami penurunan sebesar 40 cm. Dalam waktu 1 detik ketinggian sudah sama sama dengan nilai *set point* yang diatur pada kontroler ketinggian.



Gambar. 13. Respon ketinggian dan sudut *roll* & *pitch* dengan kontroler PD

Analisis Implementasi

Implementasi kontroler pada *quadcopter* ini dijalankan ketika *quadcopter* sudah dalam keadaan terbang. Untuk mencapai titik tersebut *quadcopter* diterbangkan secara manual terlebih dahulu menggunakan *remote control* dan kemudian di-*switch* kedalam mode auto. Mode auto ini akan menjalankan program yang telah dirancang pada perancangan implementasi. Dari respon grafik ketinggian yang ditampilkan pada Gambar 14 terlihat *quadcopter* dapat menjaga ketinggian pada sekitar set point yang diatur sekitar ketinggian 100cm.



Gambar. 14. Respon ketinggian pada proses implementasi

V. KESIMPULAN

Dari penelitian yang dilakukan dapat diambil beberapa kesimpulan sebagai berikut :

- 1) Simulasi dari *quadcopter* yang dijalankan dapat ditampilkan secara 3D agar mempermudah dalam mempresentasikan gerakan-gerakan yang terjadi pada *quadcopter*.
- 2) Pembacaan sensor *accelerometer* terdapat *noise* yang nilainya cukup besar sehingga diperlukan suatu metode pengolahan data agar pembacaannya lebih baik dan tanpa *noise*. Dalam penelitian ini digunakan *complementary*

filter dan terbukti hasil pembacaan sensor menjadi sesuai dengan sudut terukur tanpa *noise*.

- 3) Perancangan sistem kendali PD pada simulasi yang digunakan untuk mempertahankan posisi *hover* pada *quadcopter* menghasilkan respon ketinggian dan sudut *roll* & *pitch* dengan besar *error steady state* = 0.
- 4) Nilai parameter yang didapatkan dari hasil *tuning* terstruktur pada simulasi adalah pada kontroler *roll* & *pitch* $K_p=40$, $K_d=5$ dan pada kontroler ketinggian $K_p=35$, $K_d=17$.
- 5) Respon hasil implementasi pada *quadcopter* belum sesuai pada hasil simulasi. Masih terdapat osilasi yang besarnya ± 15 cm pada sekitar *set point* pada respon ketinggian dikarenakan pembacaan sensor ketinggian yang kadang tidak sesuai dengan ketinggian terukur. Selain itu koreksi sudut *roll* & *pitch* pada kontroler *roll* & *pitch* juga masih terdapat error $\pm 5^\circ$.

DAFTAR PUSTAKA

- [1] SA,Inkyu. "Indoor visual SLAM with a Quadrotor", Queensland University Of Technology, Juli 2011.
- [2] Bresciani, Tommaso. "Modelling, Identification and Control of a Quadrotor Helicopter", Department of Automatic Control Lund University, October 2008.
- [3] David Schmidt, Michael. "Simulation And Control Of A Quadrotor Unmanned Aerial Vehicle", University of Kentucky, 2011.
- [4] W. Beard,Randal. "Quadrotor Dynamics and Control", Brigham Young University, February 19, 2008.
- [5] Fahmizal. "Implementasi Sistem Navigasi Behavior Based dan Kontroler PID pada Manuver Robot Maze", Institut Teknologi Sepuluh Nopember Surabaya, Juli 2011.
- [6] Amir,Yasir. "Modeling of Quadrotor Helicopter Dynamics", Department of Electronic and Power Engineering, National University of Sciences and Technology, Karachi, Pakistan, April 2011.
- [7] Ogata, Katsuhiko. "Teknik Kontrol Automatik – terjemahan: Ir. Edi Laksono", Erlangga, Jakarta, 1991
- [8] Colton,Shane. "The Balance Filter : A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform", Massachusetts Institute of Technology, June 25, 2007.
- [9] Bejo, Agus., "C & AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATmega 8535", Graha Ilmu, Yogyakarta, 2008.
- [10] Gamayanti, Nurlita. "Diktat Mata Kuliah Dasar Sistem Pengaturan", Institut Teknologi Sepuluh Nopember Surabaya, 2010.