

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Nilai Koordinat Gerakan Kendali *Quadcopter*

6.1.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui nilai koordinat yang digunakan untuk penetapan gerakan kendali navigasi *quadcopter*.

6.1.2 Pelaksanaan Pengujian

Pengujian nilai koordinat gerakan kendali *quadcopter* dilakukan dengan cara untuk gerakan *pitch* pertama memposisikan tangan tepat diatas *leap motion* kemudian menggerakkan tangan ke arah belakang dan ke arah depan *leap motion*. Pada gerakan *roll* dilakukan dengan cara memposisikan tangan tepat diatas *leap motion* kemudian menggerakkan tangan ke sisi kiri *leap motion* dan ke sisi kanan *leap motion*. Untuk gerakan *yaw* dilakukan dengan cara memposisikan tangan tepat diatas *leap motion* kemudian memutar ke arah kiri *leap motion* dan ke kanan *leap motion*. Pada gerakan *gaz* dilakukan dengan cara memposisikan tangan diatas *leap motion* kemudian bergerak ke bawah menuju *leap motion* dan ke atas menjauhi *leap motion*. Untuk gerakan *takeoff*, *hover*, mendarat dilakukan dengan cara memposisikan tangan berada tepat diatas *leap motion* lalu menggenggam untuk gerakan *takeoff*, telapak tangan membuka untuk gerakan *hover* dan jari menekan seperti menekan *keyboard* untuk gerakan mendarat. Untuk setiap gerakan dilakukan 5 kali percobaan. Dari hasil percobaan tersebut maka dijadikan pedoman untuk menentukan koordinat pada setiap gerakan tangan yang digunakan untuk mengendalikan *quadcopter* pada sub bab 5.2.

6.1.3 Prosedur Pengujian

Dalam pengujian dilakukan sesuai dengan prosedur sebagai berikut:

1. *Leap motion* dihubungkan dengan komputer menggunakan kabel *USB*.
2. Membuka aplikasi *leap motion control panel*. Lalu muncul *icon leap motion* pada di pojok kanan bawah desktop, jika *icon leap motion* berwarna hijau maka *leap motion* siap digunakan. Jika *icon leap motion* berwarna hitam maka klik kanan pada *icon* tersebut lalu pilih *resume tracking*. Jika *icon* berwarna oranye maka *leap motion* perlu dibersihkan permukaannya terlebih dahulu sebelum digunakan.
3. Membuka *command prompt node js* pada komputer lalu menjalankan program "*node sistemkontrolquadcopter.js*".
4. Pengguna memposisikan tangan di atas *leap motion*.
5. Menggerakkan tangan sesuai dengan instruksi yang telah ditentukan yaitu gerakan *pitch*, *roll*, *yaw*, *gaz*, *takeoff*, *hover*, mendarat.

6. Melihat nilai koordinat yang dihasilkan pada setiap gerakan pada *command prompt* dan nilai koordinat tersebut disalin ke dalam *microsoft excel* untuk diolah untuk dijadikan pedoman pada perancangan gerakan.
7. Mengulangi langkah ke-5 sebanyak lima kali untuk setiap pengguna.

6.1.4 Hasil Pengujian

6.1.4.1 Gerakan *Pitch*

Pada Gambar 6.1 dapat dilihat proses melakukan pengujian gerakan *pitch*. Pada bagian (a) merupakan gerakan *pitch* ke belakang yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan bergerak ke belakang sampai *leap motion* tidak mendeteksi tangan pengguna. Bagian (b) merupakan gerakan *pitch* ke depan yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan bergerak ke depan sampai *leap motion* tidak mendeteksi tangan pengguna.



Gambar 6.1 Pengujian gerakan *pitch* ke depan

6.1.4.2 Gerakan *Roll*

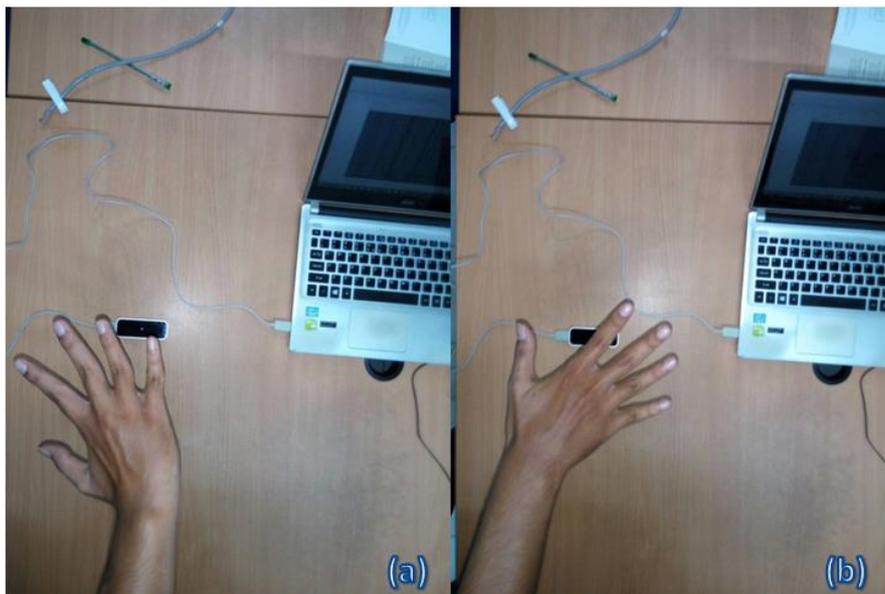
Pada Gambar 6.2 dapat dilihat proses melakukan pengujian gerakan *roll*. Pada bagian (a) merupakan gerakan *roll* ke kiri yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan bergerak ke kiri sampai *leap motion* tidak mendeteksi tangan pengguna. Bagian (b) merupakan gerakan *roll* ke kanan yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan bergerak ke kanan sampai *leap motion* tidak mendeteksi tangan pengguna.



Gambar 6.2 Pengujian gerakan *roll*

6.1.4.3 Gerakan *Yaw*

Pada Gambar 6.3 dapat dilihat proses melakukan pengujian gerakan *yaw*. Pada bagian (a) merupakan gerakan *yaw* ke kiri yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan berputar ke kiri sampai batas maksimal yang bisa dilakukan tangan pengguna. Bagian (b) merupakan gerakan *yaw* ke kanan yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan berputar ke kanan sampai batas maksimal yang bisa dilakukan tangan pengguna.



Gambar 6.3 Pengujian gerakan *yaw*

6.1.4.4 Gerakan *Gaz*

Pada Gambar 6.4 dapat dilihat proses melakukan pengujian gerakan *gaz*. Pada bagian (a) merupakan gerakan *gaz* ke bawah yaitu menggerakkan tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu tangan bergerak ke bawah sampai *leap motion* tidak mendeteksi tangan pengguna. Bagian (b) merupakan gerakan *gaz* ke atas yaitu menggerakkan tangan dengan posisi awal

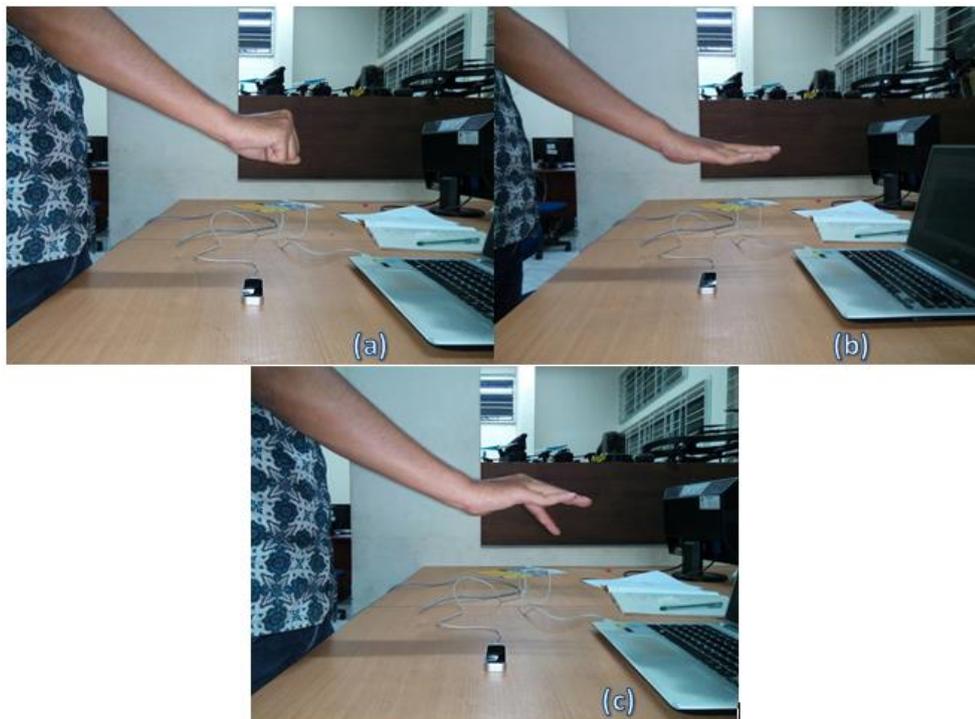
tangan tepat berada diatas *leap motion* lalu tangan bergerak ke atas sampai *leap motion* tidak mendeteksi tangan pengguna.



Gambar 6.4 Pengujian gerakan *gaz*

6.1.4.5 Gerakan *Takeoff*, *Hover* dan Mendarat

Pada Gambar 6.5 dapat dilihat proses melakukan pengujian gerakan *takeoff*, *hover* dan mendarat. Pada bagian (a) merupakan gerakan *takeoff* yaitu telapak tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu telapak tangan menggenggam. Bagian (b) merupakan gerakan *hover* yaitu telapak tangan dengan posisi awal tangan tepat berada diatas *leap motion*. Bagian (c) merupakan gerakan mendarat yaitu telapak tangan dengan posisi awal tangan tepat berada diatas *leap motion* lalu telapak tangan menekan seperti menekan sebuah tombol.



Gambar 6.5 Pengujian gerakan *takeoff*, *hover*, mendarat

6.1.5 Analisis Hasil Pengujian

Dari pengujian yang dilakukan menghasilkan berupa nilai koordinat dari semua gerakan kendali navigasi *quadcopter*. Nilai koordinat berasal dari pembacaan *leap motion* untuk koordinat x, y dan z satuannya adalah mm dan untuk yaw satuannya adalah radian. Nilai tersebut dikelompokkan sesuai dengan gerakan masing-masing seperti berikut:

Tabel 6.1 Hasil percobaan penentuan koordinat gerakan *pitch*

Gerakan (<i>pitch</i>)	Nilai			
	Koordinat x (mm)	Koordinat y (mm)	Koordinat z (mm)	Yaw (radian)
Ke belakang	-16,671	249,6388	75,71804	0,077996
Ke belakang	-11,7873	253,8842	166,6108	0,091287
Ke belakang	-10,9168	257,133	242,595	0,145312
Ke belakang	-6,39582	260,8422	303,4686	0,144418
Ke belakang	-0,30152	270,322	393,812	0,155882
Ke depan	-16,7511	251,413	-48,5957	0,073717
Ke depan	-23,3282	264,908	-302,02	0,075529
Ke depan	-23,705	261,714	-332,279	0,07732
Ke depan	-23,3597	256,1836	-358,004	0,139044
Ke depan	-28,1759	256,627	-385,824	0,156266

Pada Tabel 6.1 merupakan nilai koordinat yang dihasilkan pada gerakan *pitch*. Nilai koordinat x, y dan yaw mengalami perubahan yang tidak tetap. Untuk nilai koordinat z mengalami perubahan yang tetap yaitu semakin bertambah dan semakin berkurang dikarenakan untuk gerakan *pitch leap motion* menggunakan sumbu z sebagai tumpuan gerakan sehingga pada gerakan *pitch* diambil nilai koordinat z untuk pedoman menentukan batasan gerakan *pitch* dengan *hover* untuk koordinat z pada *leap motion* pada sub bab 5.2.

Pada Tabel 6.2 merupakan nilai koordinat yang dihasilkan pada gerakan *roll*. Nilai koordinat y, z dan yaw mengalami perubahan yang tidak tetap. Untuk nilai koordinat x mengalami perubahan yang tetap dikarenakan untuk gerakan *roll leap motion* menggunakan sumbu x sebagai tumpuan gerakan sehingga pada gerakan *roll* diambil nilai koordinat x untuk pedoman menentukan batasan gerakan *roll* dengan *hover* untuk koordinat x pada *leap motion* pada sub bab 5.2.

Pada Tabel 6.3 merupakan nilai koordinat yang dihasilkan pada gerakan yaw. Nilai koordinat x, y dan z mengalami perubahan yang tidak tetap. Untuk nilai yaw mengalami perubahan yang tetap dikarenakan untuk gerakan *yaw leap motion* menggunakan sumbu y sebagai tumpuan gerakan dan perubahan sudut ketika

melakukan gerakan *yaw* sehingga pada gerakan *yaw* diambil nilai *yaw* untuk pedoman menentukan batasan gerakan *yaw* dengan *hover* untuk perubahan nilai sudut pada *leap motion* pada sub bab 5.2.

Tabel 6.2 Hasil percobaan penentuan koordinat gerakan *roll*

Gerakan (<i>roll</i>)	Nilai			
	Koordinat x (mm)	Koordinat y (mm)	Koordinat z (mm)	<i>Yaw</i> (radian)
Ke kiri	-71,3866	257,3504	-3,87307	-0,00365
Ke kiri	-140,199	252,9392	-11,3189	-0,09788
Ke kiri	-258,103	241,6976	-12,2746	-0,27221
Ke kiri	-375,393	242,0474	10,17404	-0,60815
Ke kiri	-474,416	233,6288	-5,02401	-0,57768
Ke kanan	42,94622	234,7244	8,830496	-0,10585
Ke kanan	174,9236	248,3298	17,67776	-0,08106
Ke kanan	274,7738	247,7042	25,04316	-0,02478
Ke kanan	355,7488	250,2622	33,10138	-0,00287
Ke kanan	481,778	276,5446	48,26644	0,041784

Tabel 6.3 Hasil percobaan penentuan koordinat gerakan *yaw*

Gerakan (<i>yaw</i>)	Nilai			
	Koordinat x (mm)	Koordinat y (mm)	Koordinat z (mm)	<i>Yaw</i> (radian)
Ke kiri	-24,5127	225,7876	-14,1125	-0,34465
Ke kiri	-22,6657	228,9856	-23,2179	-0,51195
Ke kiri	-23,7032	231,5432	-30,3003	-0,70108
Ke kiri	-25,188	231,1298	-33,3918	-0,82995
Ke kiri	-25,588	230,918	-36,2319	-0,93954
Ke kanan	-17,9421	215,0194	-9,0707	0,320141
Ke kanan	-15,2496	216,1554	-9,61167	0,499895
Ke kanan	-12,6273	215,0258	-9,7554	0,660196
Ke kanan	-8,50085	214,6274	-10,001	0,864207
Ke kanan	-4,83344	212,1306	-7,86274	0,994282

Tabel 6.4 Hasil percobaan penentuan koordinat gerakan *gaz*

Gerakan (<i>gaz</i>)	Nilai			
	Koordinat x (mm)	Koordinat y (mm)	Koordinat z (mm)	Yaw (radian)
Ke bawah	-11,1545	195,7632	-12,5625	-0,04787
Ke bawah	5,020682	128,348	-10,7798	-0,04705
Ke bawah	9,152554	95,48624	-7,80281	0,017532
Ke bawah	14,03245	62,00168	-5,73005	0,099777
Ke bawah	21,37834	34,8221	-6,04442	0,142561
Ke atas	-11,3029	305,4712	-17,7881	0,046818
Ke atas	-14,3218	431,023	-42,0644	0,018658
Ke atas	-20,6565	526,7486	-54,6586	0,049317
Ke atas	-22,5122	604,1308	-65,1726	0,114024
Ke atas	-15,4397	697,164	-82,4665	0,238357

Pada Tabel 6.4 merupakan nilai koordinat yang dihasilkan pada gerakan *gaz*. Nilai koordinat x, z dan yaw mengalami perubahan yang tidak tetap. Untuk nilai koordinat y mengalami perubahan yang tetap dikarenakan untuk gerakan *gaz leap motion* menggunakan sumbu y sebagai tumpuan gerakan sehingga pada gerakan *gaz* diambil nilai koordinat y untuk pedoman menentukan batasan gerakan *gaz* dengan *hover* untuk koordinat y pada *leap motion* pada sub bab 5.2.

Pada Tabel 6.5 merupakan nilai koordinat yang dihasilkan pada gerakan *takeoff*, *hover* dan mendarat. Nilai koordinat x, y, z dan yaw tidak mengalami perubahan yang drastis dikarenakan untuk gerakan *takeoff*, *hover* dan mendarat perbedaannya pada gerakan telapak tangan. Untuk gerakan *takeoff* gerakan telapak tangan yaitu menggenggam. Untuk gerakan *hover* telapak tangan membuka dan untuk gerakan mendarat telapak tangan bergerak seperti menekan sebuah tombol sehingga pada gerakan *takeoff*, *hover*, mendarat diambil nilai koordinat x, y, z, dan yaw untuk pedoman menentukan batasan gerakan dengan gerakan *pitch*, *roll* dan yaw pada sub bab 5.2.

Tabel 6.5 Hasil percobaan penentuan koordinat gerakan *takeoff*, *hover*, mendarat

Gerakan (<i>takeoff</i> , <i>hover</i> , mendarat)	Nilai			
	Koordinat x (mm)	Koordinat y (mm)	Koordinat z (mm)	Yaw (radian)
<i>Takeoff</i>	5,44241	245,751	-26,1774	-0,00114
<i>Takeoff</i>	6,150441	245,8674	-27,7097	-0,00142
<i>Takeoff</i>	4,084202	247,1518	-28,6956	-0,00643
<i>Takeoff</i>	3,598775	246,9202	-25,1339	0,009209
<i>Takeoff</i>	4,68569	244,5102	-28,9354	0,005706
<i>Hover</i>	-1,74629	223,3518	6,001906	0,049028
<i>Hover</i>	-3,20989	222,9754	5,16865	0,047722
<i>Hover</i>	-3,63728	222,884	4,931272	0,048415
<i>Hover</i>	-4,97245	222,645	4,146183	0,047958
<i>Hover</i>	-5,42672	222,1634	4,568116	0,048955
Mendarat	-3,10575	247,9068	-0,52181	0,053428
Mendarat	-5,71061	246,7982	1,312898	0,04071
Mendarat	-4,56839	248,7728	-0,42281	0,04245
Mendarat	-4,70385	249,9172	-1,64464	0,039098
Mendarat	-5,31794	249,5516	-2,69188	0,048894

6.2 Pengujian Ketepatan Gerakan

6.2.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui apakah sistem sudah berjalan sesuai dengan perancangan serta memberikan *output* yang tepat.

6.2.2 Pelaksanaan Pengujian

Pengujian ketepatan gerakan dilakukan dengan kondisi dengan beberapa kondisi yaitu pengguna berjumlah dua orang dengan masing-masing mempunyai ukuran telapak tangan berbeda untuk pengguna A panjang telapak tangan 22 cm dan lebar 13 cm. Untuk pengguna B panjang telapak tangan 20 cm dan lebar 11 cm. Setiap pengguna melakukan pengujian sebanyak 5 kali. Dilakukan pengujian dengan ukuran telapak tangan berbeda bertujuan untuk mengetahui apakah gerakan *quadcopter* sesuai dengan instruksi yang diberikan oleh pengguna yang berbeda. Untuk setiap instruksi gerakan yang dikirim menuju *quadcopter* dapat dilihat *output* pada *command prompt* di desktop

komputer. Setelah mengamati *output* pada komputer juga mengamati apakah *quadcopter* bergerak sesuai dengan instruksi yang pengguna. Instruksi yang diberikan antara lain *roll* ke kanan, *roll* ke kiri, *pitch* ke depan, *pitch* ke belakang, *yaw* ke kanan, *yaw* ke kiri, *takeoff*, *hover* dan mendarat.

6.2.3 Prosedur Pengujian

Dalam pengujian dilakukan sesuai dengan prosedur sebagai berikut:

1. *Leap motion* dihubungkan dengan komputer menggunakan kabel *USB*.
2. Membuka aplikasi *leap motion control panel*. Lalu muncul *icon leap motion* pada di pojok kanan bawah desktop, jika *icon leap motion* berwarna hijau maka *leap motion* siap digunakan. Jika *icon leap motion* berwarna hitam maka klik kanan pada *icon* tersebut lalu pilih *resume tracking*. Jika *icon* berwarna *orange* maka *leap motion* perlu dibersihkan permukaannya terlebih dahulu sebelum digunakan.
3. Pasang baterai *quadcopter* dan tunggu sampai *quadcopter* selesai mengkalibrasi. Setelah terdengar suara *beep* sebanyak empat kali maka kalibrasi *quadcopter* sudah selesai dan siap digunakan.
4. Menghubungkan komputer dengan *quadcopter* melalui *hotspot Wi-Fi* yang ada pada *quadcopter*.
5. Membuka *command prompt node js* pada komputer lalu menjalankan program "*node sistemkontrolquadcopter.js*".
6. Membuka *command prompt node js* sekali lagi pada komputer lalu setelah terbuka jalankan program "*node navdata.js*".
7. Pengguna memosisikan tangan di atas *leap motion*.
8. Menggerakkan tangan sesuai dengan instruksi yang telah ditentukan yaitu gerakan *roll*, *pitch*, *yaw*, *gaz*, *takeoff*, *hover*, mendarat.
9. Melihat pergerakan *quadcopter* apakah sesuai dengan instruksi yang diberikan.
10. Data navigasi pada *command prompt* yang menjalankan *navdata.js* disalin lalu memindah data tersebut pada *microsoft excel* untuk menganalisis data tersebut.
11. Mengulangi langkah ke-5 sebanyak lima kali untuk setiap gerakan.

6.2.4 Hasil Pengujian

6.2.4.1 Gerakan *Pitch*

Pada Gambar 6.6 dapat dilihat proses melakukan pengujian gerakan *pitch* ke depan. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya

sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya ke depan dan *quadcopter* bergerak ke depan sesuai dengan instruksi yang diberikan oleh pengguna.



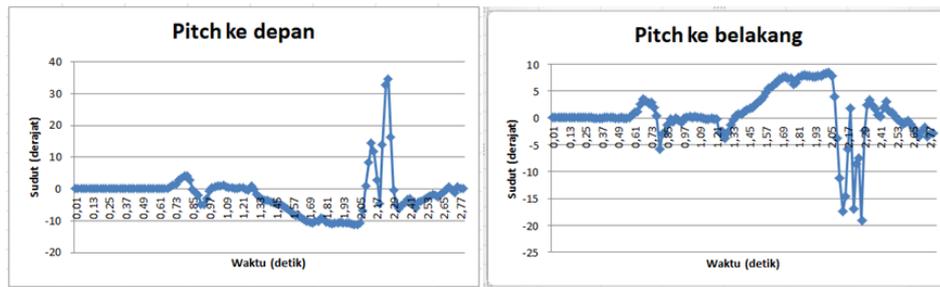
Gambar 6.6 Pengujian gerakan *pitch* ke depan

Pada Gambar 6.7 dapat dilihat proses melakukan pengujian gerakan *pitch* ke belakang. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya ke belakang dan *quadcopter* bergerak ke belakang sesuai dengan instruksi yang diberikan oleh pengguna.



Gambar 6.7 Pengujian gerakan *pitch* ke belakang

Setelah dilakukan pengujian gerakan *pitch* maka dihasilkan grafik seperti pada Gambar 6.8. Pada gambar tersebut terlihat perubahan sudut yang dialami oleh *quadcopter* ketika melakukan gerakan *pitch*. Perubahan sudut tersebut cenderung sama antara gerakan *pitch* ke depan dengan gerakan *pitch* ke belakang hanya berbeda nilai positif dan negatifnya. Hal ini dikarenakan menggunakan satu koordinat yaitu koordinat x untuk gerakan *pitch* ke depan dan ke belakang. Perubahan nilai sudut menandakan kemiringan *quadcopter* sehingga *quadcopter* bergerak sesuai sudut mana yang nilainya berubah. Untuk *pitch* ke depan nilai perubahan sudut pada koordinat x semakin negatif sedangkan untuk gerakan *pitch* ke belakang nilai perubahan sudut semakin positif.



Gambar 6.8 Grafik gerakan *pitch*

6.2.4.2 Gerakan *Roll*

Pada Gambar 6.9 dapat dilihat proses melakukan pengujian gerakan *roll* ke kiri. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya ke kiri dan *quadcopter* bergerak ke kiri sesuai dengan instruksi yang diberikan oleh pengguna.



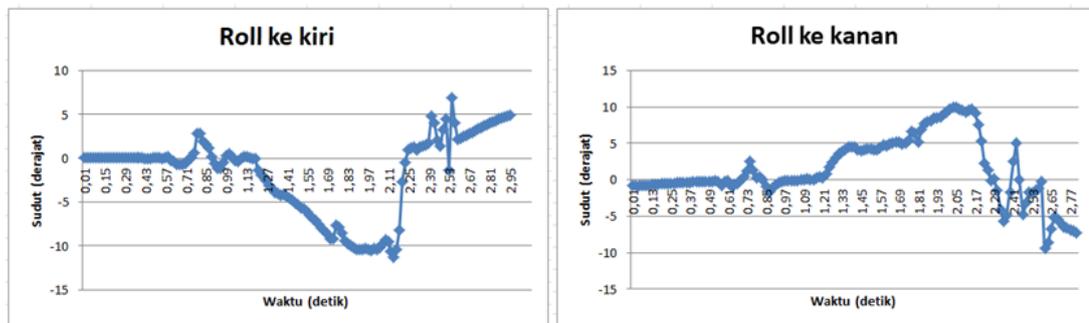
Gambar 6.9 Pengujian gerakan *roll* ke kiri

Pada Gambar 6.10 dapat dilihat proses melakukan pengujian gerakan *roll* ke kanan. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya ke kanan dan *quadcopter* bergerak ke kanan sesuai dengan instruksi yang diberikan oleh pengguna.



Gambar 6.10 Pengujian gerakan *roll* ke kanan

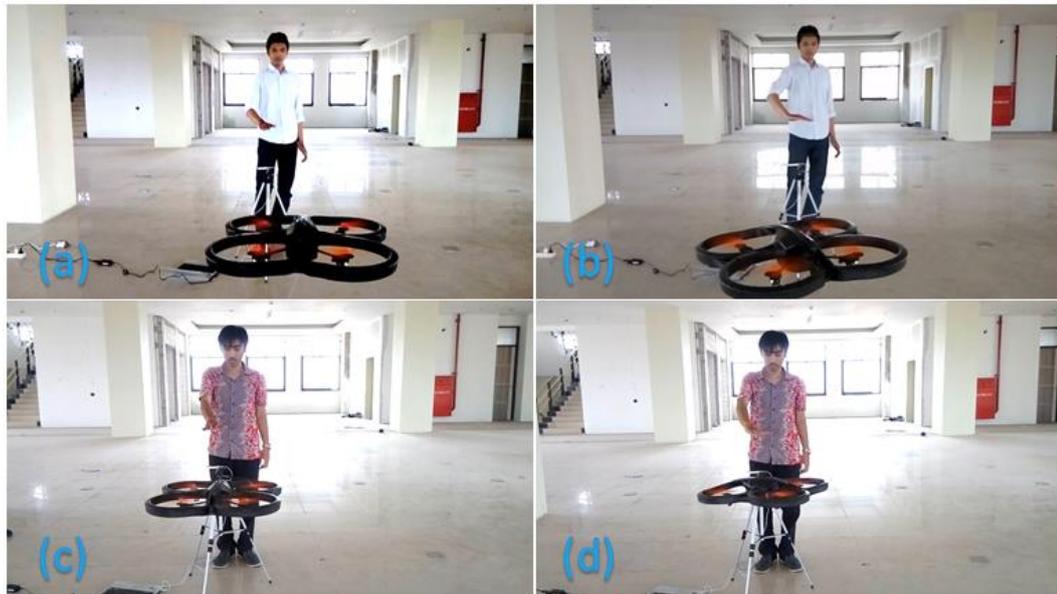
Setelah dilakukan pengujian gerakan *roll* maka dihasilkan grafik seperti pada Gambar 6.11. Pada gambar tersebut terlihat perubahan sudut yang dialami oleh *quadcopter* ketika melakukan gerakan *roll*. Perubahan sudut tersebut cenderung sama antara gerakan *roll* ke kiri dengan gerakan *roll* ke kanan hanya berbeda nilai positif dan negatifnya. Hal ini dikarenakan menggunakan satu koordinat yaitu koordinat y untuk gerakan *roll* ke kiri dan ke kanan. Perubahan nilai sudut menandakan kemiringan *quadcopter* sehingga *quadcopter* bergerak sesuai sudut mana yang nilainya berubah. Untuk *roll* ke kiri nilai perubahan sudut pada koordinat y semakin negatif sedangkan untuk gerakan *roll* ke kanan nilai perubahan sudut semakin positif.



Gambar 6.11 Grafik gerakan *roll*

6.2.4.3 Gerakan Yaw

Pada Gambar 6.12 dapat dilihat proses melakukan pengujian gerakan *yaw* ke kiri. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya putar ke kiri dan *quadcopter* bergerak putar ke kiri sesuai dengan instruksi yang diberikan oleh pengguna.



Gambar 6.12 Pengujian gerakan *yaw* putar ke kiri

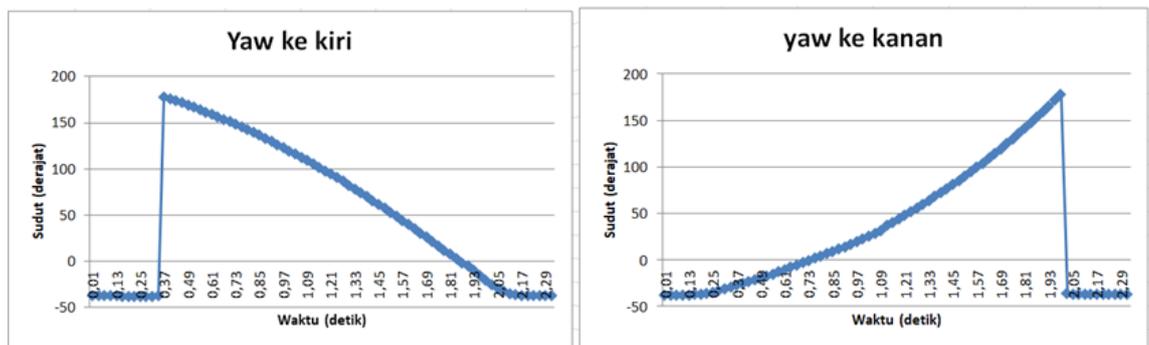
Pada Gambar 6.13 dapat dilihat proses melakukan pengujian gerakan *yaw* ke kanan. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya putar ke kanan dan *quadcopter* bergerak putar ke kanan sesuai dengan instruksi yang diberikan oleh pengguna.

Setelah dilakukan pengujian gerakan *yaw* maka dihasilkan grafik seperti pada Gambar 6.14. Pada gambar tersebut terlihat perubahan sudut yang dialami oleh *quadcopter* ketika melakukan gerakan *yaw*. Perubahan sudut tersebut cenderung sama antara gerakan *yaw* putar ke kiri dengan gerakan *yaw* putar ke kanan hanya berbeda pada perubahan nilainya. Hal ini dikarenakan menggunakan satu koordinat yaitu koordinat z untuk gerakan *yaw* putar ke kiri dan putar ke kanan. Perubahan nilai sudut menandakan kemiringan *quadcopter* sehingga *quadcopter* bergerak sesuai sudut mana yang nilainya berubah. Untuk *yaw* putar ke kiri nilai perubahan sudut pada koordinat z semakin kecil

sedangkan untuk gerakan yaw putar ke kanan nilai perubahan sudut semakin besar.



Gambar 6.13 Pengujian gerakan yaw putar ke kanan



Gambar 6.14 Grafik gerakan yaw

6.2.4.4 Gerakan Gaz

Pada Gambar 6.15 dapat dilihat proses melakukan pengujian gerakan *gaz* ke atas. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya ke atas dan *quadcopter* bergerak ke atas sesuai dengan instruksi yang diberikan oleh pengguna.

Pada Gambar 6.16 dapat dilihat proses melakukan pengujian gerakan *gaz* ke bawah. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal sebelum pengguna mulai menggerakkan tangannya

sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna menggerakkan tangannya ke bawah dan *quadcopter* bergerak ke bawah sesuai dengan instruksi yang diberikan oleh pengguna.

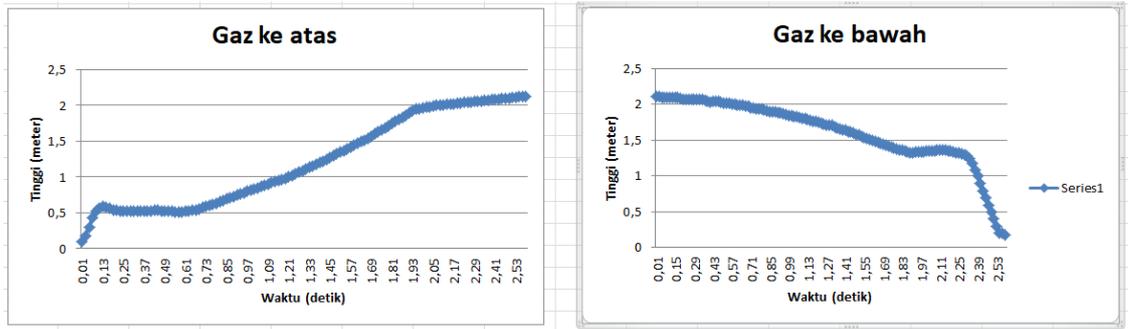


Gambar 6.15 Pengujian gerakan *gaz* ke atas



Gambar 6.16 Pengujian gerakan *gaz* ke bawah

Setelah dilakukan pengujian gerakan *gaz* maka dihasilkan grafik seperti pada Gambar 6.17. Pada gambar tersebut terlihat perubahan ketinggian yang dialami oleh *quadcopter* ketika melakukan gerakan *gaz*. Untuk *gaz* ke atas nilai ketinggian semakin bertambah sedangkan untuk gerakan *gaz* ke bawah nilai ketinggian semakin mengecil.



Gambar 6.17 Grafik gerakan gaz

6.2.4.5 Gerakan *Takeoff*, *Hover* dan *Mendarat*

Pada Gambar 6.18 dapat dilihat proses melakukan pengujian gerakan *takeoff*. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal pengguna memberikan instruksi *takeoff* yaitu dengan menggenggamkan tangan pengguna sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna memberikan instruksi dan *quadcopter* mulai bergerak terbang sesuai dengan instruksi yang diberikan oleh pengguna.



Gambar 6.18 Pengujian gerakan *takeoff*

Pada Gambar 6.19 dapat dilihat proses melakukan pengujian gerakan *hover*. Pada bagian (a) dan (b) merupakan gerakan *hover* yang dilakukan pengguna A dan B. Gerakan *hover* ini mempunyai fungsi untuk mempertahankan posisi stabil *quadcopter*. Dalam setiap pergantian pergerakan diharuskan pengguna melakukan gerakan *hover* untuk memudahkan pengguna mengendalikan *quadcopter* sesuai yang diharapkan.



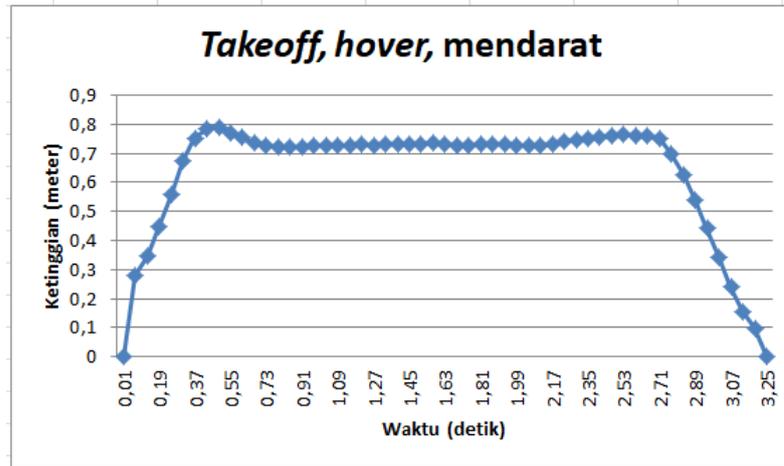
Gambar 6.19 Pengujian gerakan *hover*

Pada Gambar 6.20 dapat dilihat proses melakukan pengujian gerakan mendarat. Pada bagian (a) dan (b) merupakan gerakan yang dilakukan oleh pengguna A. Sedangkan pada bagian (c) dan (d) merupakan gerakan yang dilakukan oleh pengguna B. Untuk gerakan pengguna A, B pada bagian (a), (c) adalah gerakan awal pengguna memberikan instruksi mendarat yaitu dengan menggerakkan salah satu jari kebawah seolah-olah seperti menekan tombol *keyboard* sedangkan pada bagian (b), (d) adalah gerakan setelah pengguna memberikan instruksi dan *quadcopter* mulai bergerak mendarat sesuai dengan instruksi yang diberikan oleh pengguna.



Gambar 6.20 Pengujian gerakan mendarat

Setelah dilakukan pengujian gerakan *gaz* maka dihasilkan grafik seperti pada Gambar 6.21. Pada gambar tersebut terlihat perubahan ketinggian yang dialami oleh *quadcopter* ketika melakukan gerakan *gaz*. Untuk *gaz* ke atas nilai ketinggian semakin bertambah sedangkan untuk gerakan *gaz* ke bawah nilai ketinggian semakin mengecil.



Gambar 6.21 Grafik gerakan *takeoff*, *hover*, *mendarat*

Setelah pengguna A dan pengguna B melakukan percobaan sebanyak lima kali maka dihasilkan seperti pada Tabel 6.6 dan Tabel 6.7. Untuk Tabel 6.6 merupakan hasil pengujian pengguna A dan untuk Tabel 6.7 merupakan hasil pengujian pengguna B. Simbol (√) merupakan tanda jika percobaan berhasil, simbol (×) merupakan tanda jika percobaan tidak berhasil.

Tabel 6.6 Hasil pengujian ketepatan gerakan pengguna A

Gerakan	Percobaan ke				
	1	2	3	4	5
<i>Pitch</i> ke depan	√	√	√	√	√
<i>Pitch</i> ke belakang	√	√	√	√	√
<i>Roll</i> ke kiri	√	√	√	√	√
<i>Roll</i> ke kanan	√	√	√	√	√
<i>Yaw</i> putar ke kiri	√	√	√	√	√
<i>Yaw</i> putar ke kanan	√	√	√	√	√
<i>Gaz</i> ke atas	√	√	√	√	√
<i>Gaz</i> ke bawah	√	√	√	√	√
<i>Takeoff</i>	√	√	√	√	√
<i>Hover</i>	√	√	√	√	√
<i>Mendarat</i>	√	√	√	√	√

Tabel 6.7 Hasil pengujian ketepatan gerakan pengguna B

Gerakan	Percobaan ke				
	1	2	3	4	5
<i>Pitch</i> ke depan	√	√	√	√	√
<i>Pitch</i> ke belakang	√	√	√	√	√
<i>Roll</i> ke kiri	√	√	√	√	√
<i>Roll</i> ke kanan	√	√	√	√	√
<i>Yaw</i> putar ke kiri	√	√	√	√	√
<i>Yaw</i> putar ke kanan	√	√	√	√	√
<i>Gaz</i> ke atas	√	√	√	√	√
<i>Gaz</i> ke bawah	√	√	√	√	√
<i>Takeoff</i>	√	√	√	√	√
<i>Hover</i>	√	√	√	√	√
Mendarat	√	√	√	√	√

6.2.5 Analisis Hasil Pengujian

Dari pengujian yang dilakukan dihasilkan berupa keseluruhan pengguna dapat mengendalikan *quadcopter* dengan benar dari kelima kali percobaan. Sehingga dapat dihitung persentase ketepatan gerakan dengan Persamaan 6.1.

$$\text{Persentase ketepatan} = \frac{\text{Jumlah gerakan benar}}{\text{Jumlah pengujian}} \times 100\% \quad (6.1)$$

Sumber : (Hadi, et al., 2017)

Pada Tabel 6.8 dapat dilihat sistem ini mampu menghasilkan ketepatan gerakan 100% untuk dua pengguna dengan diameter telapak tangan berbeda.

Tabel 6.8 Hasil ketepatan gerakan

Gerakan	Jumlah pengujian	Jumlah gerakan yang benar (Pengguna A)	Jumlah gerakan yang benar (Pengguna B)	Persentase ketepatan
<i>Pitch</i> ke depan	5	5	5	100%
<i>Pitch</i> ke belakang	5	5	5	100%
<i>Roll</i> ke kiri	5	5	5	100%
<i>Roll</i> ke kanan	5	5	5	100%

Tabel 6.8 Hasil ketepatan gerakan (lanjutan)

Gerakan	Jumlah pengujian	Jumlah gerakan yang benar (Pengguna A)	Jumlah gerakan yang benar (Pengguna B)	Persentase ketepatan
<i>Yaw</i> putar ke kiri	5	5	5	100%
<i>Yaw</i> putar ke kanan	5	5	5	100%
<i>Gaz</i> ke atas	5	5	5	100%
<i>Gaz</i> ke bawah	5	5	5	100%
<i>Takeoff</i>	5	5	5	100%
<i>Hover</i>	5	5	5	100%
Mendarat	5	5	5	100%
Total ketepatan gerakan				100%

6.3 Pengujian Waktu Sistem

6.3.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui seberapa baik performa dari sistem kendali yang telah dibuat dengan melihat seberapa besar waktu terjadinya *delay* yang terjadi saat pengguna menggerakkan tangan sampai *quadcopter* dapat bergerak juga.

6.3.2 Pelaksanaan Pengujian

Pengujian ini dilakukan dengan melihat selisih *frame* antara gerakan tangan pengguna yang telah dilakukan dengan sempurna hingga *quadcopter* dapat bergerak. Untuk mengetahui selisih *frame* tersebut dengan akurat, maka akan dilakukan perekaman video pada saat pengguna mengendalikan *quadcopter*. Hasil dari video tersebut akan dianalisis menggunakan *editor video Adobe Premiere Pro CC 2015* dengan menghitung selisih *frame* gambar antara gerakan tangan pengguna dengan gerakan *quadcopter*.

6.3.3 Prosedur Pengujian

Dalam pengujian dilakukan sesuai dengan prosedur sebagai berikut:

1. Menghubungkan *leap motion* dengan komputer menggunakan kabel *USB*.
2. Membuka aplikasi *leap motion* control panel. Setelah itu muncul *icon leap motion* pada di pojok kanan bawah desktop, jika *icon leap motion* berwarna hijau maka *leap motion* siap digunakan. Jika *icon leap motion* berwarna hitam maka klik kanan pada *icon* tersebut lalu pilih *resume tracking*. Jika *icon*

berwarna oranye maka *leap motion* perlu dibersihkan permukaannya terlebih dahulu sebelum digunakan.

3. Memasang baterai pada *quadcopter* dan menunggu sampai *quadcopter* selesai mengkalibrasi. Setelah terdengar suara beep sebanyak empat kali maka kalibrasi *quadcopter* sudah selesai dan siap digunakan.
4. Menghubungkan komputer dengan *quadcopter* melalui *hotspot Wi-Fi* yang ada pada *quadcopter*.
5. Membuka *command prompt node js* pada komputer lalu menjalankan program "*node sistemkontrolquadcopter.js*".
6. Pengguna memposisikan tangan di atas *leap motion*.
7. Menggerakkan tangan sesuai dengan instruksi yang telah ditentukan yaitu gerakan *roll, pitch, yaw, gaz, takeoff, hover, mendarat*.
8. Ketika pengguna mengendalikan *quadcopter*, keseluruhan gerakan pengguna dan *quadcopter* direkam menggunakan alat rekam.
9. Mengulangi tahap ke-7 pengujian sebanyak lima kali.
10. Setelah mendapatkan video hasil pengujian, video tersebut dibuka pada *adobe premiere pro cc 2015* dan lakukan pengamatan per *frame* video untuk menghasilkan *delay* yang akurat. *Delay* dihitung berdasarkan selisih per *frame* antara pengguna selesai menggerakkan tangan dengan *quadcopter* bergerak sesuai dengan instruksi yang diberikan. Lalu hasil selisih per *frame* dibagi dengan *fps* pada video rekaman hasil pengujian sehingga menghasilkan sebuah *delay*.

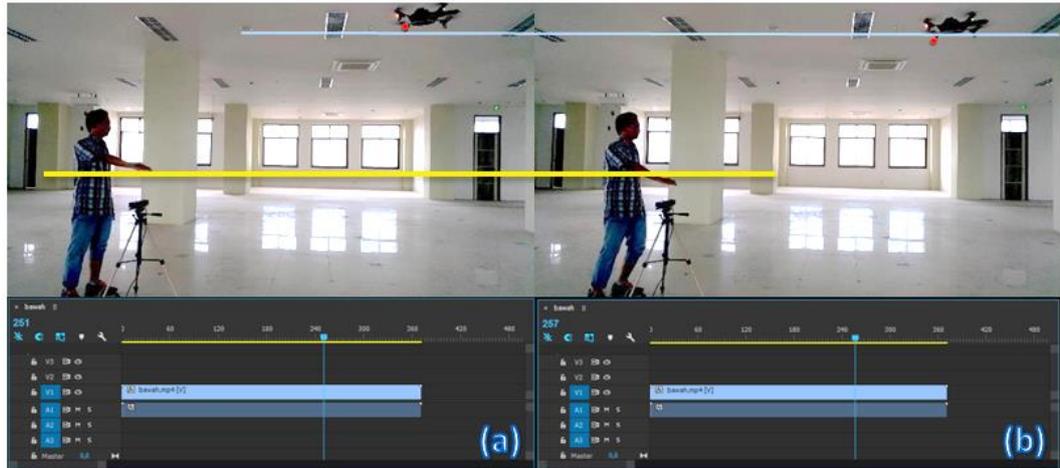
6.3.4 Hasil Pengujian

Untuk *frame per second* video (*fps*) hasil pengujian yang digunakan sebesar 29,97 *fps* karena *default* dari alat rekam yang digunakan. Setelah melakukan pengujian maka didapat hasil *delay* untuk masing-masing pergerakan *quadcopter* sebagai berikut:

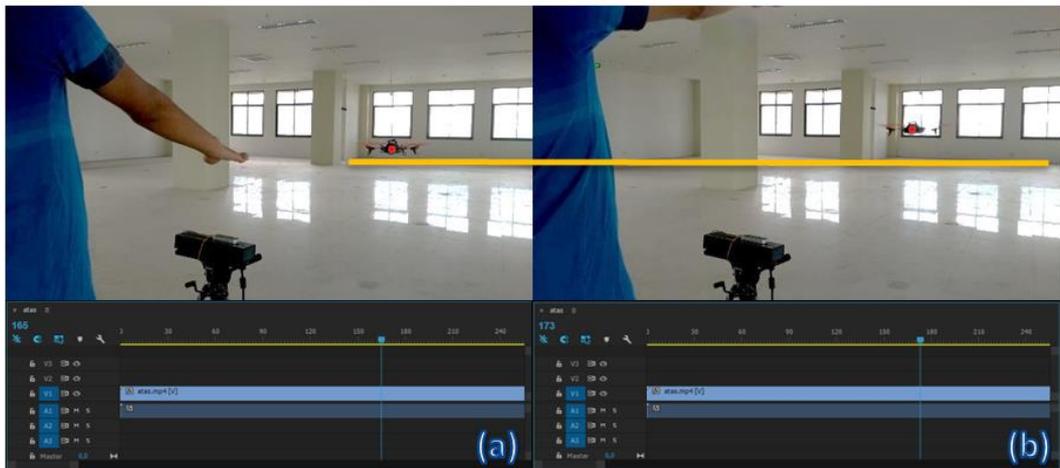
6.3.4.1 Gerakan *Gaz*

Pada saat *gaz* ke bawah seperti pada Gambar 6.22 (a) dapat dilihat bahwa *quadcopter* awalnya tepat berada di atas garis biru pada saat tangan pengguna pada posisi *hover* di atas garis kuning pada *frame* ke-251. Lalu pada Gambar 6.22 (b), diperlihatkan pada *frame* ke-257 tangan pengguna mulai menurunkan posisi dibawah garis kuning dan *quadcopter* mulai melewati garis biru atau telah melakukan gerakan *Gaz* ke bawah.

Lalu pada Gambar 6.23 diperlihatkan gerakan *gaz* ke atas. Pada Gambar 6.23 (a) pengguna masih melakukan gerakan *hover* pada *frame* ke-165. *Quadcopter* berada tepat di atas garis kuning. Kemudian pada Gambar 6.23 (b) terlihat *quadcopter* mulai bergerak ke atas menjauhi garis kuning pada *frame* ke-173 ketika tangan pengguna bergerak ke atas.



Gambar 6.22 Pengujian *delay* pada *gaz* ke bawah

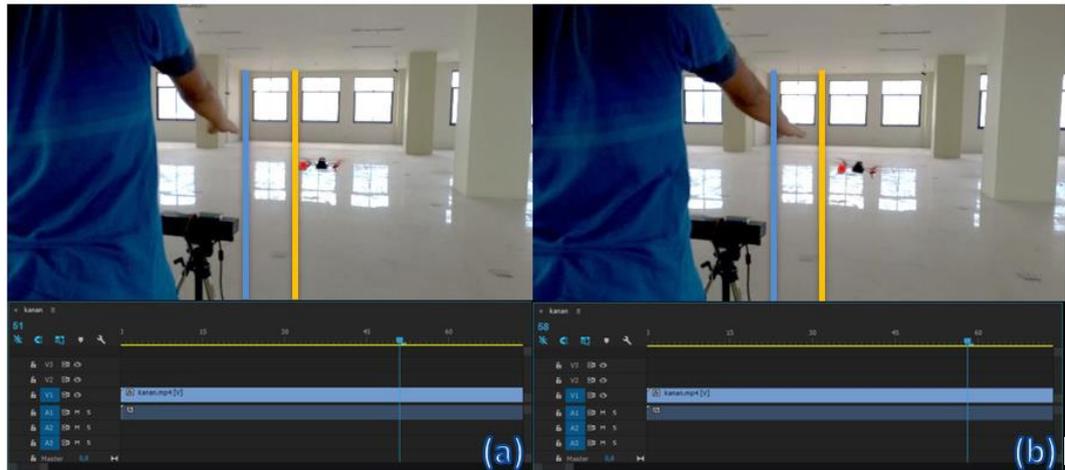


Gambar 6.23 Pengujian *delay* pada *gaz* ke atas

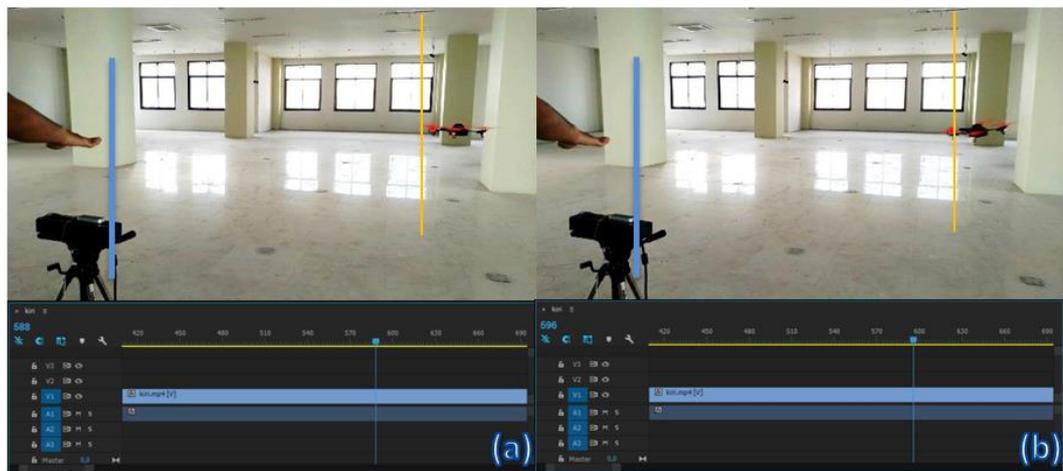
6.3.4.2 Gerakan *Roll*

Pada saat *roll* kanan seperti pada Gambar 6.24 (a) dapat dilihat bahwa *quadcopter* awalnya tepat berada di samping kanan garis kuning pada saat tangan pengguna pada posisi *hover frame* ke-51. Lalu pada Gambar 6.24 (b) diperlihatkan pada *frame* ke-58 tangan pengguna mulai menggeserkan posisi ke kanan dari posisi *hover* dan *quadcopter* mulai menjauhi garis kuning atau telah melakukan gerakan *roll* ke kanan.

Lalu pada Gambar 6.25 diperlihatkan gerakan *roll* ke kiri. Pada Gambar 6.25 (a) pengguna masih melakukan gerakan *hover* pada *frame* ke-588. *Quadcopter* berada tepat di kanan garis kuning. Kemudian pada Gambar 6.25 (b) terlihat *quadcopter* mulai bergerak ke kiri melewati garis kuning pada *frame* ke-596 ketika tangan pengguna bergerak ke kiri menjauhi garis biru.



Gambar 6.24 Pengujian *delay* pada *roll* ke kanan

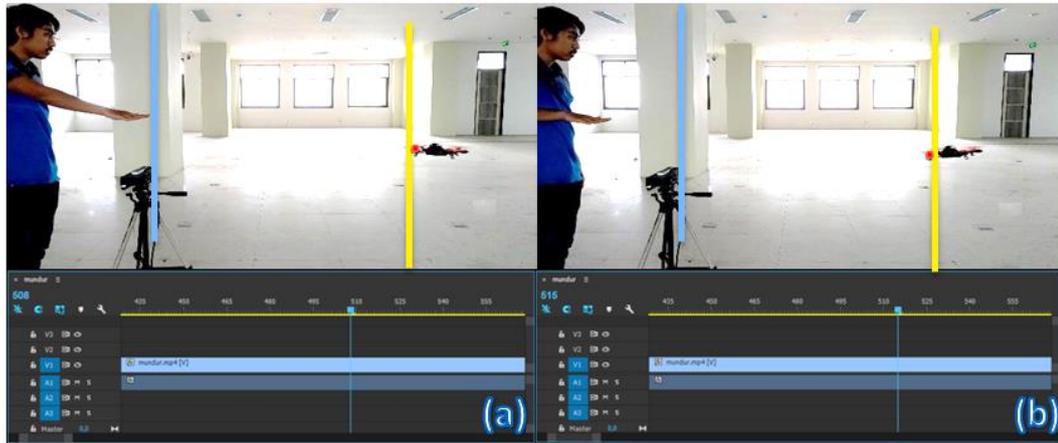


Gambar 6.25 Pengujian *delay* pada *roll* ke kiri

6.3.4.3 Gerakan *Pitch*

Pada saat *pitch* ke belakang seperti pada Gambar 6.26 (a) dapat dilihat bahwa *quadcopter* awalnya tepat berada di samping kanan garis kuning pada saat tangan pengguna pada posisi *hover frame* ke-508. Lalu pada Gambar 6.26 (b), diperlihatkan pada *frame* ke-515 tangan pengguna mulai menggerakkan ke belakang dari garis biru dan *quadcopter* mulai melewati garis kuning atau telah melakukan gerakan *pitch* ke belakang.

Lalu pada Gambar 6.27 diperlihatkan gerakan *pitch* ke depan. Pada Gambar 6.27 (a) pengguna masih dalam posisi gerakan *hover* pada *frame* ke-13. *Quadcopter* berada tepat di kiri garis kuning. Kemudian pada Gambar 6.27 (b) terlihat *quadcopter* mulai bergerak ke kanan melewati garis kuning pada *frame* ke-21 ketika tangan pengguna bergerak ke depan melewati garis biru.



Gambar 6.26 Pengujian *delay* pada *pitch* mundur



Gambar 6.27 Pengujian *delay* pada *pitch* maju

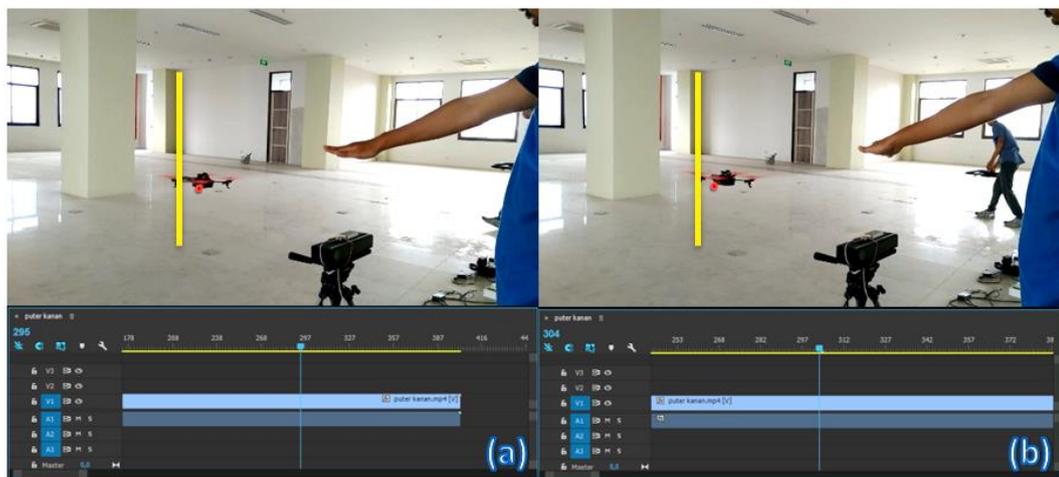
6.3.4.4 Gerakan *Yaw*

Pada saat *yaw* ke kiri seperti pada Gambar 6.28 (a) dapat dilihat bahwa *quadcopter* awalnya tepat berada di samping kanan garis kuning dan terdapat celah antara garis kuning dengan *quadcopter* pada saat tangan pengguna pada posisi *hover frame* ke-371. Lalu pada Gambar 6.28 (b), diperlihatkan pada *frame* ke-379 tangan pengguna mulai menggerakkan memutar ke kiri dari posisi *hover* dan *quadcopter* mulai mendekati garis kuning atau telah melakukan gerakan *yaw* ke kiri.

Lalu pada Gambar 6.29 diperlihatkan gerakan *yaw* ke kanan. Pada Gambar 6.29 (a) pengguna masih dalam posisi gerakan *hover* pada *frame* ke-295. *Quadcopter* berada pada sisi kanan pada garis kuning. Kemudian pada Gambar 6.29 (b) terlihat *quadcopter* mulai bergerak memutar ke kanan mendekati garis kuning pada *frame* ke-304 ketika tangan pengguna memutar ke kanan.



Gambar 6.28 Pengujian *delay* pada *yaw* ke kiri



Gambar 6.29 Pengujian *delay* pada *yaw* ke kanan

6.3.5 Analisis Hasil Pengujian

Setelah melakukan pengujian *delay* pada setiap gerakan pengguna sebanyak lima kali didapat hasil *delay* pada Tabel 6.9. Besar *delay* yang dihasilkan diambil dari selisih per *frame* pergerakan tangan pengguna dengan gerakan *quadcopter* sesuai dengan instruksi yang diberikan pengguna. Kemudian hasil selisih per *frame* dibagi dengan fps pada video rekaman pengujian yaitu 29,97 fps sehingga menghasilkan sebuah *delay*. *Delay* total didapatkan dari nilai rata-rata *delay* dalam lima kali percobaan pada setiap gerakan. Sehingga menghasilkan total *delay* 0,258 detik. Dari hasil pengujian ini *delay* yang dihasilkan lebih kecil dari pada *delay* yang dihasilkan pada penelitian Pallas, Setyawan & Prasetyo (2017) yaitu suara sebagai kendali *quadcopter* yang menghasilkan *delay* 3 detik seiring dengan bertambahnya jumlah kata yang diberikan oleh pengguna.

Tabel 6.9 Hasil *delay* dari sistem

Gerakan	Pengujian ke-					Rata-rata (detik)
	1	2	3	4	5	
<i>Gaz</i> ke bawah	0,200	0,267	0,200	0,234	0,334	0,247
<i>Gaz</i> ke atas	0,267	0,234	0,200	0,300	0,267	0,254
<i>Roll</i> ke kanan	0,234	0,267	0,300	0,300	0,200	0,260
<i>Roll</i> ke kiri	0,267	0,200	0,234	0,300	0,267	0,254
<i>Pitch</i> ke belakang	0,234	0,267	0,200	0,234	0,334	0,254
<i>Pitch</i> ke depan	0,267	0,267	0,300	0,267	0,267	0,274
<i>Yaw</i> ke kiri	0,267	0,200	0,234	0,234	0,334	0,254
<i>Yaw</i> ke kanan	0,300	0,234	0,267	0,267	0,267	0,267
Total Delay						0,258

6.4 Pengujian Fungsi Kecepatan

6.4.1 Tujuan Pengujian

Pengujian ini dilaksanakan untuk mengetahui apakah fungsi pengaturan kecepatan telah berfungsi sesuai dengan yang diharapkan.

6.4.2 Pelaksanaan Pengujian

Pengujian ini dilakukan dengan mengendalikan *quadcopter* dari titik awal menuju titik akhir. Pada saat proses perjalanan dari titik awal menuju titik akhir, gerakan yang digunakan beracuan pada koordinat x , y , z pada *quadcopter*. Untuk pengujian kecepatan pada koordinat x digunakan gerakan *pitch* ke depan, setelah itu pada koordinat y digunakan gerakan *roll* kanan, lalu pada koordinat z digunakan gerakan *gaz* ke atas serta pada *yaw* ke kanan. Untuk setiap gerakan yang diuji dilakukan perubahan gerak tangan dengan konstan contoh saat gerakan *pitch* ke depan tangan pengguna bergerak maju secara konstan sampai *quadcopter* berada pada titik akhir. Data yang diambil dari pengujian tersebut yaitu nilai kecepatan pada *quadcopter* yang terdapat pada data navigasi yang dikirim langsung oleh *quadcopter* dan nilai *input* kecepatan dari pergerakan tangan pengguna berdasarkan pada sub bab 5.4 yaitu V_{pitch} , V_{roll} , V_{gaz} serta V_{rot} menyesuaikan dengan gerakan yang dilakukan. Berdasarkan dokumentasi pihak *Parrot AR Drone* kecepatan *quadcopter* memiliki satuan mm/s. Namun pada pengujian ini diubah menjadi m/s menyesuaikan dengan kecepatan yang diberikan dari pengguna. Hal ini dikarenakan selisih nilai yang terpaut jauh antara kecepatan *quadcopter* yang dapat mencapai nilai diatas 1 mm/s sedangkan masukan dari pengguna maksimal hanya bernilai 1 saja dalam satuan *float* sesuai

pada format pengiriman *AT Command*. Kemudian data tersebut akan disimpan dalam *file* berekstensi *.xls* untuk dianalisis hasilnya.

6.4.3 Prosedur Pengujian

Dalam pengujian dilakukan sesuai dengan prosedur sebagai berikut:

1. Menghubungkan *leap motion* dengan komputer menggunakan kabel *USB*.
2. Membuka aplikasi *leap motion* control panel. Setelah itu muncul *icon leap motion* pada di pojok kanan bawah desktop, jika *icon leap motion* berwarna hijau maka *leap motion* siap digunakan. Jika *icon leap motion* berwarna hitam maka klik kanan pada *icon* tersebut lalu pilih *resume tracking*. Jika *icon* berwarna orange maka *leap motion* perlu dibersihkan permukaannya terlebih dahulu sebelum digunakan.
3. Memasang baterai pada *quadcopter* dan menunggu sampai *quadcopter* selesai mengkalibrasi. Setelah terdengar suara beep sebanyak empat kali maka kalibrasi *quadcopter* sudah selesai dan siap digunakan.
4. Menghubungkan komputer dengan *quadcopter* melalui *hotspot Wi-Fi* yang ada pada *quadcopter*.
5. Membuka *command prompt node js* pada komputer lalu menjalankan program "*node sistemkontrolquadcopter.js*".
6. Membuka *command prompt node js* sekali lagi pada komputer lalu setelah terbuka jalankan program "*node navdata.js*".
7. Pengguna memposisikan tangan di atas *leap motion*.
8. Menggerakkan tangan sesuai dengan instruksi yang telah ditentukan yaitu gerakan *roll, pitch, yaw, gaz*. Contoh gerakan *pitch* maka pengguna akan menggerakkan tangan mulai terangkat sedikit dari posisi *hover* sampai dengan terangkat dengan tinggi secara berkelanjutan, begitu pula dengan instruksi gerakan lainnya.
9. Setelah melakukan semua instruksi yang diuji maka memindah data dari *command prompt* yang menjalankan instruksi "*node navdata.js*" menuju *microsoft excel*.
10. Mengulangi tahap ke-5 pengujian sebanyak lima kali.

6.4.4 Hasil Pengujian

Setelah melakukan pengujian maka didapatkan nilai dari kecepatan *quadcopter* dan nilai kecepatan dari gerakan tangan. Berikut ini merupakan data dari pengujian tersebut.

6.4.4.1 Kecepatan Gerakan *Pitch*

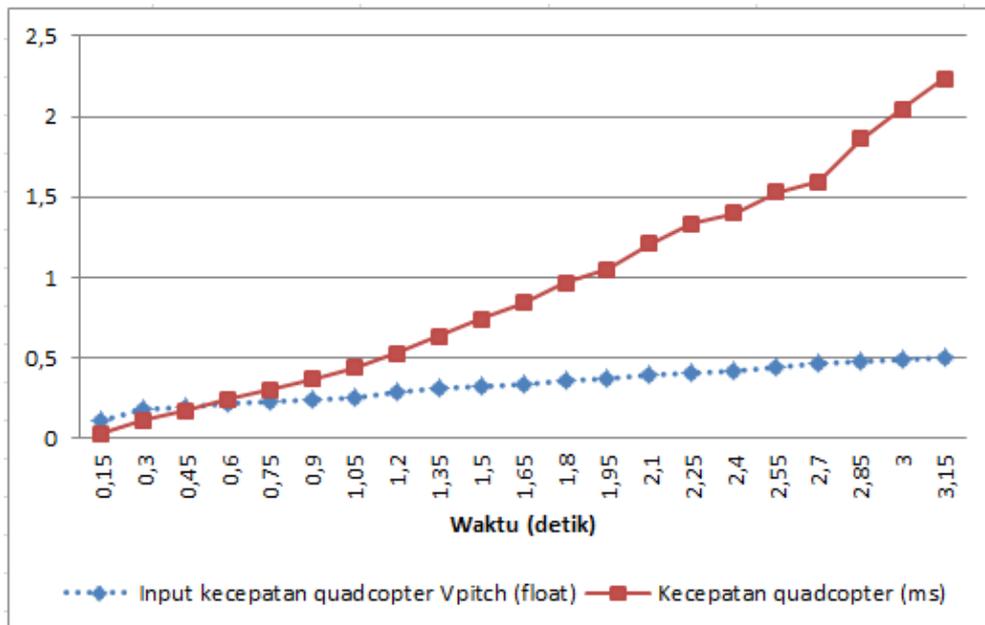
Nilai pada Tabel 6.10 dihasilkan sesuai dengan Persamaan 5.1 dalam kondisi tangan pengguna bergerak ke depan dan bergerak ke belakang. Pada saat tangan pengguna bergerak semakin ke depan maka nilai dari *Vpitch* akan bertambah dan

pada saat tangan pengguna bergerak semakin ke belakang maka nilai dari *Vpitch* akan bertambah juga.

Tabel 6.10 Data perhitungan *Vpitch*

Gerakan	<i>PositionZ</i>	<i>Vpitch</i>
Ke depan	-366,637	0,916593
Ke depan	-286,688	0,71672
Ke depan	-207,46	0,51865
Ke depan	-123,226	0,308065
Ke depan	-48,2572	0,120643
Ke belakang	74,8221	0,187055
Ke belakang	121,198	0,302995
Ke belakang	201,966	0,504915
Ke belakang	284,965	0,712413
Ke belakang	365,91	0,914775

Lalu dalam gerakan ini didapatkan nilai dari kelima kali percobaan dan dihitung rata-ratanya. Hasilnya dapat dilihat pada Tabel 6.11. Nilai gerakan *pitch* selengkapnya terdapat pada lampiran. Dari nilai pada tabel didapatkan grafik perbandingan antara kecepatan *quadcopter* dengan kecepatan yang diberikan pengguna terhadap *quadcopter* terlihat pada Gambar 6.30.



Gambar 6.30 Grafik pengaruh *input* kecepatan terhadap kecepatan *pitch* *quadcopter*

Tabel 6.11 Nilai kecepatan dari gerakan *pitch*

Waktu (Detik)	Input Kecepatan <i>Vpitch</i> (float)	Kecepatan <i>Quadcopter Vx</i> (m/s)
0,15	0,1149694	0,030799786
0,3	0,17893914	0,115816949
0,45	0,19929675	0,174738847
0,6	0,2184414	0,241628899
0,75	0,22880805	0,303187512
0,9	0,24222705	0,367505505
1,05	0,25762475	0,441977954
1,2	0,285013	0,531090222
1,35	0,3084505	0,63306051
1,5	0,323617	0,739113232
1,65	0,33988	0,840517236
1,8	0,3638115	0,967685608
1,95	0,370537	1,051508423
2,1	0,391718	1,207352661
2,25	0,4084265	1,332787622
2,4	0,42479	1,396609253
2,55	0,442995	1,526743774
2,7	0,4607705	1,59179978
2,85	0,4818925	1,860204321
3	0,4947375	2,047371631
3,15	0,502619	2,232639404

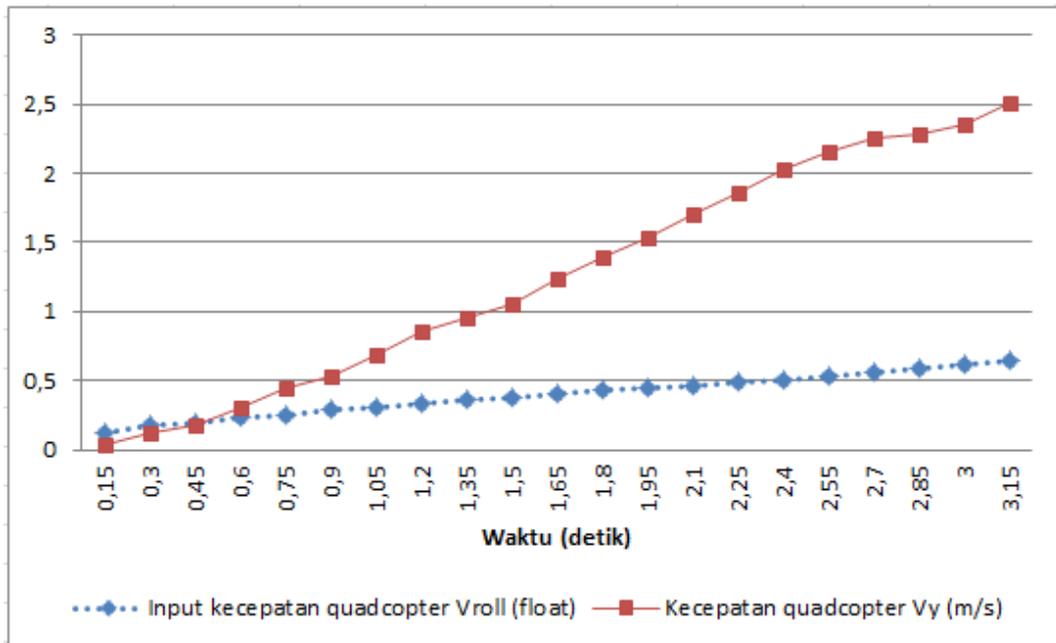
6.4.4.2 Kecepatan Gerakan *Roll*

Nilai pada Tabel 6.12 dihasilkan sesuai dengan Persamaan 5.2 dalam kondisi tangan pengguna bergerak ke kanan dan bergerak ke kiri. Pada saat tangan pengguna bergerak semakin ke kanan maka nilai dari *Vroll* akan bertambah dan pada saat tangan pengguna bergerak semakin ke kiri maka nilai dari *Vroll* akan bertambah juga.

Lalu dalam gerakan ini didapatkan nilai dari kelima kali percobaan dan dihitung rata-ratanya. Hasilnya dapat dilihat pada Tabel 6.13. Nilai gerakan *roll* selengkapnya terdapat pada lampiran. Dari nilai pada tabel didapatkan grafik perbandingan antara kecepatan *quadcopter* dengan kecepatan yang diberikan pengguna terhadap *quadcopter* terlihat pada Gambar 6.31.

Tabel 6.12 Data perhitungan *Vroll*

Gerakan	<i>PositionX</i>	<i>Vroll</i>
Ke kanan	486,937	0,973874
Ke kanan	353,169	0,706338
Ke kanan	250,107	0,500214
Ke kanan	153,995	0,30799
Ke kanan	67,639	0,135278
Ke kiri	-68,5287	0,137057
Ke kiri	-159,948	0,319896
Ke kiri	-257,964	0,515928
Ke kiri	-356,452	0,712904
Ke kiri	-475,581	0,951162



Gambar 6.31 Grafik pengaruh *input* kecepatan terhadap kecepatan *roll* quadcopter

Tabel 6.13 Nilai kecepatan dari gerakan roll

Waktu (Detik)	Input Kecepatan <i>Vroll</i> (float)	Kecepatan <i>Quadcopter</i> <i>Vy</i> (m/s)
0,15	0,119284167	0,035799786
0,3	0,178905	0,115916949
0,45	0,19602	0,171738847
0,6	0,226915	0,301187512
0,75	0,253196667	0,439977954
0,9	0,2886	0,535090222
1,05	0,306178333	0,68106051
1,2	0,337585	0,850517236
1,35	0,361243333	0,947685608
1,5	0,378075	1,057508423
1,65	0,401926667	1,230787622
1,8	0,428615	1,399609253
1,95	0,445225	1,536743774
2,1	0,462891667	1,70179978
2,25	0,48234	1,864204321
2,4	0,502619	2,032639404
2,55	0,535225	2,154680667
2,7	0,556446667	2,252208333
2,85	0,580075	2,290171667
3	0,620619	2,359734833
3,15	0,638535	2,507639404

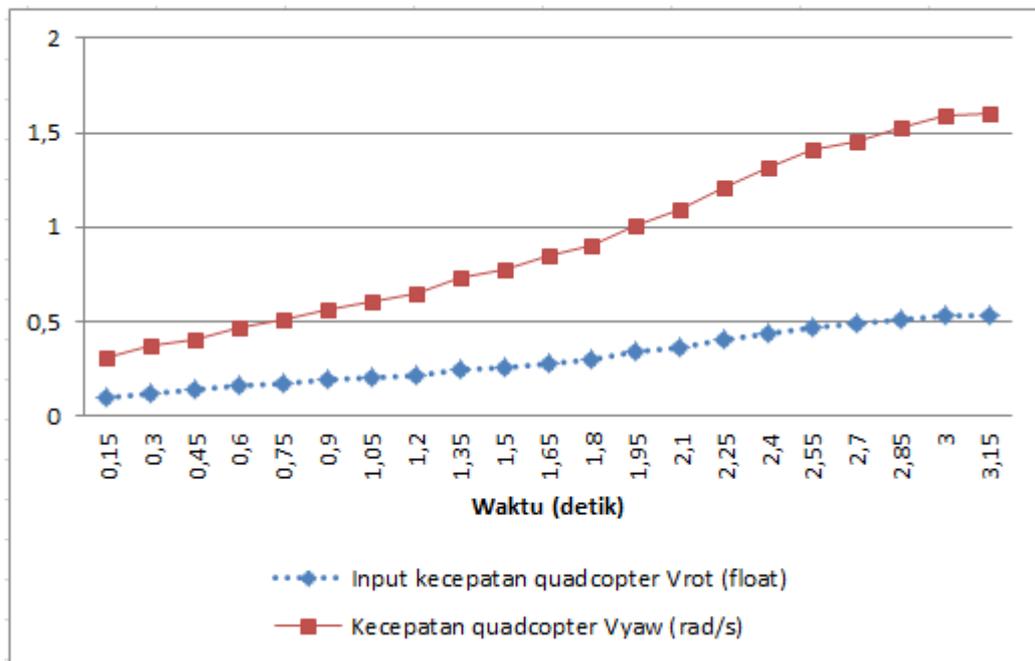
6.4.4.3 Kecepatan Gerakan *Yaw*

Nilai pada Tabel 6.14 dihasilkan sesuai dengan Persamaan 5.3 dalam kondisi tangan pengguna bergerak memutar ke kanan dan bergerak memutar ke kiri. Pada saat tangan pengguna bergerak memutar semakin ke kanan maka nilai dari *Vrot* akan bertambah dan pada saat tangan pengguna bergerak memutar semakin ke kiri maka nilai dari *Vrot* akan bertambah juga.

Lalu dalam gerakan ini didapatkan nilai dari kelima kali percobaan dan dihitung rata-ratanya. Hasilnya dapat dilihat pada Tabel 6.15. Nilai gerakan *yaw* selengkapnya terdapat pada lampiran. Dari nilai pada tabel didapatkan grafik perbandingan antara kecepatan *quadcopter* dengan kecepatan yang diberikan pengguna terhadap *quadcopter* terlihat pada Gambar 6.32.

Tabel 6.14 Data perhitungan Vrot

Gerakan	PositionYaw	Vrot
Putar ke kanan	1,965289	0,982645
Putar ke kanan	1,42293	0,711465
Putar ke kanan	1,011509	0,505755
Putar ke kanan	0,602279	0,30114
Putar ke kanan	0,330746	0,165373
Putar ke kiri	-0,35795	0,178975
Putar ke kiri	-0,63091	0,315456
Putar ke kiri	-1,04238	0,521192
Putar ke kiri	-1,4435	0,721752
Putar ke kiri	-1,95875	0,979374



Gambar 6.32 Grafik pengaruh *input* kecepatan terhadap kecepatan *yaw* quadcopter

Tabel 6.15 Nilai kecepatan dari gerakan yaw

Waktu (Detik)	Input Kecepatan Vrot (float)	Kecepatan Quadcopter VYaw (m/s)
0,15	0,101741	0,305224
0,3	0,122668	0,368004
0,45	0,136063	0,408189
0,6	0,157658	0,472973
0,75	0,170984	0,512953
0,9	0,188409	0,565226
1,05	0,200576	0,601729
1,2	0,217366	0,652097
1,35	0,24312	0,729359
1,5	0,25907	0,777209
1,65	0,281489	0,844466
1,8	0,301166	0,903498
1,95	0,335981	1,007944
2,1	0,364913	1,094738
2,25	0,404436	1,213309
2,4	0,437035	1,311104
2,55	0,469542	1,408626
2,7	0,484027	1,45208
2,85	0,509631	1,528893
3	0,529774	1,589322
3,15	0,531961	1,595883

6.4.4.4 Kecepatan Gerakan Gaz

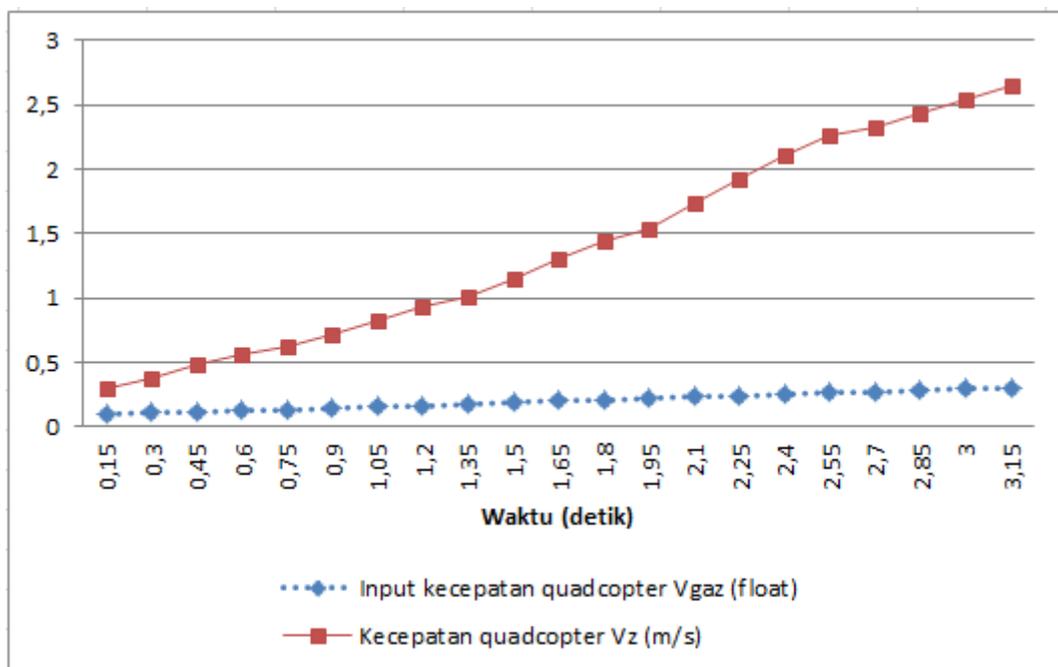
Nilai pada Tabel 6.16 dihasilkan sesuai dengan Persamaan 5.4 dalam kondisi tangan pengguna bergerak ke atas dan bergerak ke bawah. Untuk nilai V_{gaz} yang digunakan adalah V_{gaz} yang dikurangi 0,5 dikarenakan sesuai dengan kebutuhan saat tangan pengguna bergerak semakin ke atas maka nilai dari V_{gaz} akan bertambah dan pada saat tangan pengguna bergerak semakin ke bawah maka nilai dari V_{gaz} akan bertambah dan nilai maksimal V_{gaz} relatif sama untuk gerakan ke bawah dan ke atas.

Lalu dalam gerakan ini didapatkan nilai dari kelima kali percobaan dan dihitung rata-ratanya. Hasilnya dapat dilihat pada Tabel 6.17. Nilai gerakan gaz selengkapnya terdapat pada lampiran. Dari nilai pada tabel didapatkan grafik

perbandingan antara kecepatan *quadcopter* dengan kecepatan yang diberikan pengguna terhadap *quadcopter* terlihat pada Gambar 6.33.

Tabel 6.16 Data perhitungan *Vgaz*

Gerakan	<i>PositionY</i>	<i>Vgaz</i> dengan dikurangi 0,1	<i>Vgaz</i> dengan dikurangi 0,2	<i>Vgaz</i> dengan dikurangi 0,3	<i>Vgaz</i> dengan dikurangi 0,4	<i>Vgaz</i> dengan dikurangi 0,5
Ke atas	697,632	0,896617	0,796617	0,696617	0,596617	0,496617
Ke atas	634,13	0,8059	0,7059	0,6059	0,5059	0,4059
Ke atas	560,467	0,700667	0,600667	0,500667	0,400667	0,300667
Ke atas	494,868	0,606954	0,506954	0,406954	0,306954	0,206954
Ke atas	300,663	0,329519	0,229519	0,129519	0,029519	0,07048
Ke bawah	197,917	0,182739	0,082739	0,01726	0,11726	0,21726
Ke bawah	174,615	0,14945	0,04945	0,05055	0,15055	0,25055
Ke bawah	138,339	0,097627	0,002373	0,10237	0,20237	0,30237
Ke bawah	69,9229	0,00011	0,10011	0,20011	0,30011	0,40011
Ke bawah	37,4105	0,046556	0,146556	0,24656	0,34656	0,44656



Gambar 6.33 Grafik pengaruh *input* kecepatan terhadap kecepatan *gaz quadcopter*

Tabel 6.17 Nilai kecepatan dari gerakan *gaz*

Waktu (Detik)	Input Kecepatan <i>V_{gaz}</i> (float)	Kecepatan <i>Quadcopter</i> <i>V_z</i> (m/s)
0,15	0,101711	0,293
0,3	0,108757	0,369
0,45	0,116582	0,489
0,6	0,122784	0,556
0,75	0,130746	0,618
0,9	0,142327	0,713
1,05	0,15666	0,824
1,2	0,161631	0,93
1,35	0,170098	1,015
1,5	0,185003	1,145
1,65	0,204965	1,303
1,8	0,21258	1,449
1,95	0,226589	1,543
2,1	0,233177	1,73
2,25	0,243104	1,922
2,4	0,252564	2,114
2,55	0,264375	2,256
2,7	0,273861	2,317
2,85	0,281913	2,428
3	0,290919	2,537
3,15	0,303495	2,65

6.4.5 Analisis Hasil Pengujian

Pada nilai kecepatan *pitch*, *roll*, *yaw*, *gaz* diambil dalam waktu 3 detik. *Input* kecepatan yang diberikan oleh pengguna pada gerakan *pitch* yaitu 0,502619. Lalu pada gerakan *roll* adalah 0,638535. Sedangkan pada gerakan *yaw* adalah 0,531961 dan pada gerakan *gaz* yaitu 0,303495. Untuk keseluruhan dari nilai *input* tidak sampai mencapai 1 sesuai dengan perancangan. Jika *input* kecepatan mencapai 1 maka memungkinkan *quadcopter* hilang kendali dan berisiko menabrak.

Kecepatan maksimal *quadcopter* pada saat gerakan *pitch* yaitu 2,232639404 m/s. Lalu pada gerakan *roll* kecepatan maksimal *quadcopter* adalah 2,507639404 m/s. Sedangkan pada gerakan *yaw* kecepatan maksimal *quadcopter* yaitu 1,595883 rad/s dan pada gerakan *gaz* kecepatan maksimum *quadcopter* adalah

2,65 m/s. Berdasarkan pada Gambar 6.30, Gambar 6.31, Gambar 6.32, Gambar 6.33 dapat dilihat bahwa nilai kecepatan yang diberikan oleh pengguna dengan kecepatan *quadcopter* berbanding lurus. Jika *input* dari pengguna ditambah maka kecepatan *quadcopter* akan semakin bertambah. Pada penelitian Hadi, Setyawan & Maulana (2017) tentang sistem kendali navigasi *AR Drone quadcopter* dengan prinsip *natural user interface* menggunakan *microsoft kinect* mengatakan bahwa kecepatan *quadcopter* akan semakin bertambah jika pengguna memberikan *input* yang semakin besar.