

Gambar 4.2 Waktu yang diperlukan untuk mencapai kondisi *steady state*

4.2 Identifikasi parameter Kontrol

Untuk implementasi loop yang diharapkan maka perlu adanya penambahan parameter kontroler pada sistem yang dirancang, pada fungsi alih sistem yaitu $G(z^{-1}) =$

$$\frac{0.03952 Z^{-1}}{1 - 0.6155 Z^{-1} - 0.3441 Z^{-2}} \quad \text{terdapat parameter estimasi yang bernilai}$$

$$a_1 = -0.6155, a_2 = -0.3441, b_1 = 0.03952$$

Nilai parameter estimasi nantinya akan digunakan untuk mencari nilai parameter control pada persamaan (2-16), (2-17) dan (2-18) dengan menggunakan program dev C++

```
#include<stdio.h>
```

```
int main(){
```

```
    double a1 = -0.6155;
```

```
    double a2 = -0.3441;
```

```
    double b1 = 0.03952;
```

```
    double a,b,c,d,e,f,g,h,I;
```

```
    float Q = 1;
```

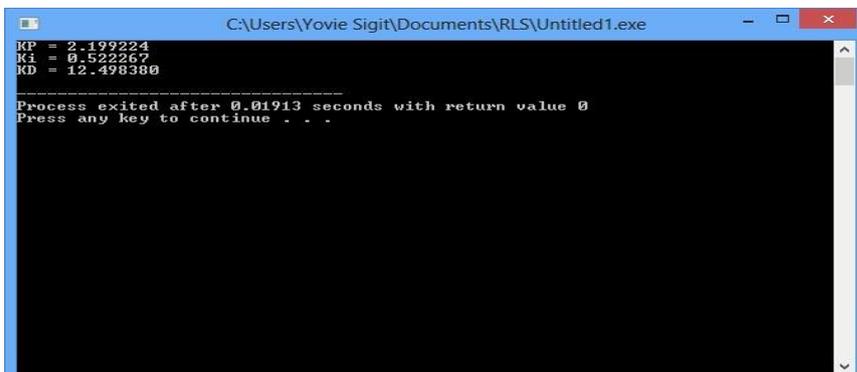
```
    double KP;
```

```
    double KI;
```

```

double KD;
double T_0 = 1;
double TD , Ti ,Ki;
a = a1+(2*a2);
b = (a/b1);
KP= -1*(b*Q);
c = T_0*a2*Q;
d = KP*b1;
TD = c/d;
e = 1/a;
f = TD/T_0;
g = e+1+f;
Ti= -1*(T_0/g);
Ki = KP/Ti;
KD = KP*TD;
printf("KP = %f\n",KP);
printf("Ki = %f\n",Ki);
printf("KD = %f\n",KD);
}

```



```

C:\Users\Yovie Sigit\Documents\RLS\Untitled1.exe
KP = 2.199224
Ki = 0.522267
KD = 12.498380
-----
Process exited after 0.01913 seconds with return value 0
Press any key to continue . . .

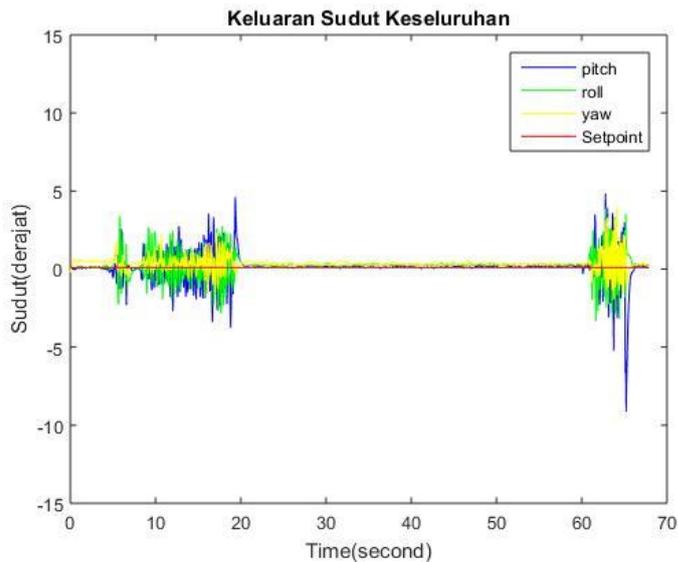
```

Gambar 4.3 Hasil perhitungan untuk mencari nilai parameter control menggunakan program DEV C++.

Hasil perhitungan parameter kontrol yang dihitung menggunakan program diatas $K_p = 2,19924$, $K_i = 0,522267$ dan $K_d = 12,54983$. nilai yang sudah didapat untuk parameter kontrol bersifat tetap di karenakan parameter estimasi di dapat secara *offline* atau tidak berubah ubah.

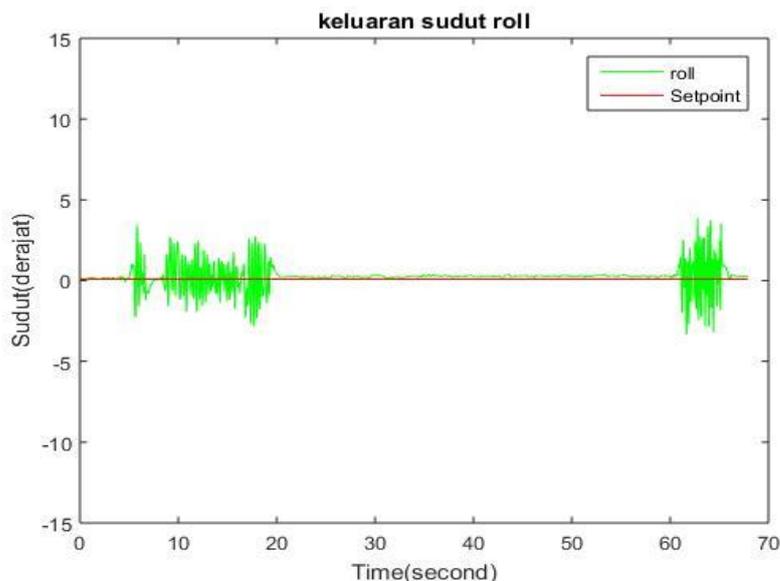
4.3 Implementasi

Implementasi sistem menggunakan waktu sampling 0,05 detik. Kemudian dengan nilai K_p , K_i dan K_d yang sudah di tentukan awal yang sudah didapatkan melalui perhitungan dengan nilai setpoint berupa sudut 0° untuk keseluruhan sudut . Data *sampling* yang diambil mulai *Quadcopter* dalam keadaan akan terbang dan sampai mendarat. Berikut analisa keseluruhan dan masing-masing sudut keluaran



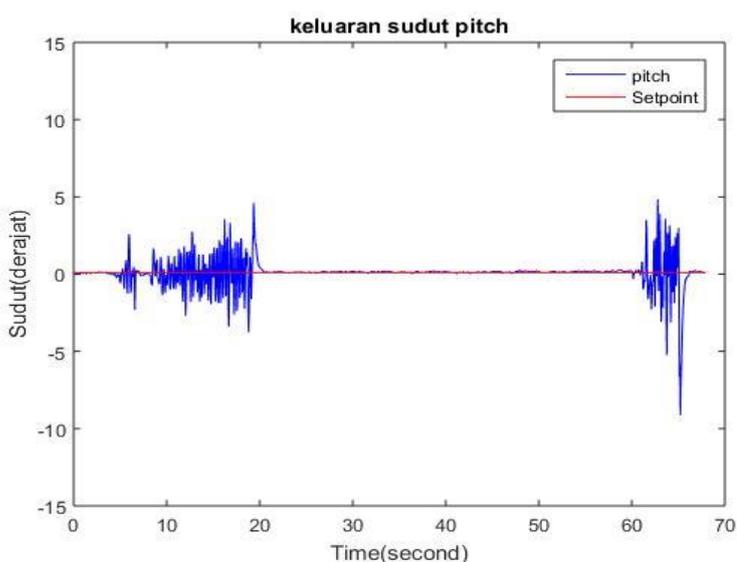
Gambar 4.4 Respon keseluruhan sudut keluaran *Quadcopter*

Dari Gambar 4.4 dapat diketahui bahwa pengontrolan kestabilan *hover* pada *quadcopter* dengan setpoint 0° . Pada awal akan terbang atau *takeoff quadcopter* banyak terjadi getaran-getaran yang disebabkan oleh kenaikan *throttle* secara bertahap yang membuat pembacaan sensor tidak stabil yang membuat banyak terjadi banyak *error*. Pada saat *hovering* sudah dapat setimbang dan tidak banyak *error*, Kemudian pada saat akan mendarat dan penurunan *throttle* secara bertahap yang membuat banyak terjadi *error*. Dapat disimpulkan bahwa perubahan *error steady state* sangat kecil saat *hovering* sehingga nilai parameter kontrol yang didapat sudah sesuai.



Gambar 4.5 Respon keseluruhan sudut keluaran *roll*

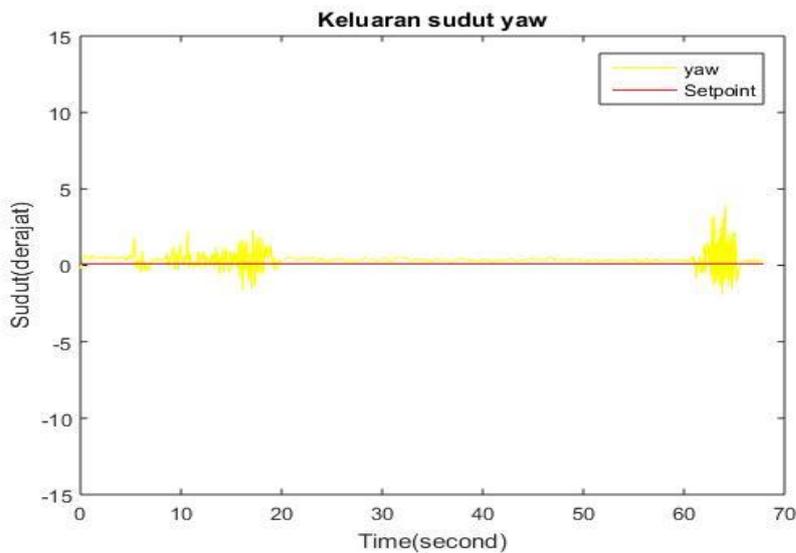
Dari Gambar 4.5 dapat diketahui atau kita lihat bahwa pengontrolan kestabilan *hover* pada sudut *roll* dengan setpoint 0° . Pada awal akan terbang atau *takeoff quadcopter* banyak terjadi getaran-getaran yang disebabkan oleh kenaikan *throttle* secara bertahap yang membuat pembacaan sudut *roll* tidak stabil yang membuat banyak terjadi banyak *error*. Pada saat *hovering* penyimpangan sudut maksimal mencapai $0,44^\circ$ dari batas maksimal penyimpangan atau *error* 10° . Sehingga nilai parameter yang di dapat sudah sesuai di harapkan dengan keluaran sudut *roll* diatas.



Gambar 4.6 Respon keseluruhan sudut keluaran *pitch*

Dari Gambar 4.6 dapat diketahui atau kita lihat bahwa pengontrolan kestabilan *hover* pada sudut *pitch* dengan setpoint 0° . Pada awal akan terbang atau *takeoff quadcopter* banyak

terjadi getaran-getaran yang disebabkan oleh kenaikan *throttle* secara bertahap yang membuat pembacaan sudut *pitch* tidak stabil yang membuat banyak terjadi banyak *error*. Pada saat *hovering* penyimpangan sudut maksimal mencapai $0,35^\circ$ dari batas maksimal penyimpangan atau *error* 10° . Sehingga nilai parameter yang di dapat sudah sesuai di harapkan dengan keluaran sudut *pitch* diatas.



Gambar 4.7 respon keseluruhan sudut keluaran *yaw*

Dari Gambar 4.7 dapat diketahui atau kita lihat bahwa pengontrolan kestabilan *hover* pada sudut *yaw* dengan setpoint 0° . Pada awal akan terbang atau *takeoff quadcopter* banyak terjadi getaran-getaran yang disebabkan oleh kenaikan *throttle* secara bertahap yang membuat pembacaan sudut *yaw* tidak stabil yang membuat banyak terjadi banyak *error*. Pada saat *hovering* penyimpangan sudut maksimal mencapai $0,42^\circ$ dari batas maksimal penyimpangan atau *error* 10° . Sehingga nilai parameter yang di dapat sudah sesuai di harapkan dengan keluaran sudut *yaw* diatas.

Dari hasil pengujian sistem yang telah dilakukan, maka dapat kita buat dalam sebuah tabel untuk memudahkan kita dalam melihat hasil performansi sistem yang telah didapatkan sebagai berikut:

Tabel 4.1 Hasil Pengujian Keseluruhan

Sudut	Setpoint(°)	<i>Error Steady State</i>(%)
Pitch	0	4,4
Roll	0	3,5
Yaw	0	4,2

