

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Simulasi

Simulasi pada evaluasi ini dibagi menjadi 3 lintasan target. Lintasan tersebut dibuat dengan mengatur track dari *virtual leader* (VL) dengan nilai dan posisi awal sebagai berikut

- Lintasan pertama, posisi awal VL berada di (250,0) dan bergerak dengan $v=1\text{ m/s}$ dan $\omega=0,5\text{ rad/s}$
- Lintasan kedua, posisi awal VL berada di (500,500) dan bergerak dengan $v=1\text{ m/s}$ dan $\omega=0.5\text{ rad/s}$
- Lintasan ketiga, posisi awal VL berada di (250, 0) dan bergerak dengan $v=1\text{ m/s}$ dan $\omega=0.5\text{ rad/s}$

4.1.1 Pengendalian berbasis Konsensus Terdistribusi

Proses simulasi dengan metode konsensus terdistribusi dimulai dengan menentukan nilai variabel kendali dan nilai *weighted value* antar robot. Kemudian simulasi akan menghasilkan lintasan yang dibentuk oleh setiap robot dan juga sinyal kendali yang diberikan kepada setiap robot. Hasil simulasi dengan metode pengendali berbasis konsensus terdistribusi untuk 3 lintasan target akan dijelaskan dibawah ini.

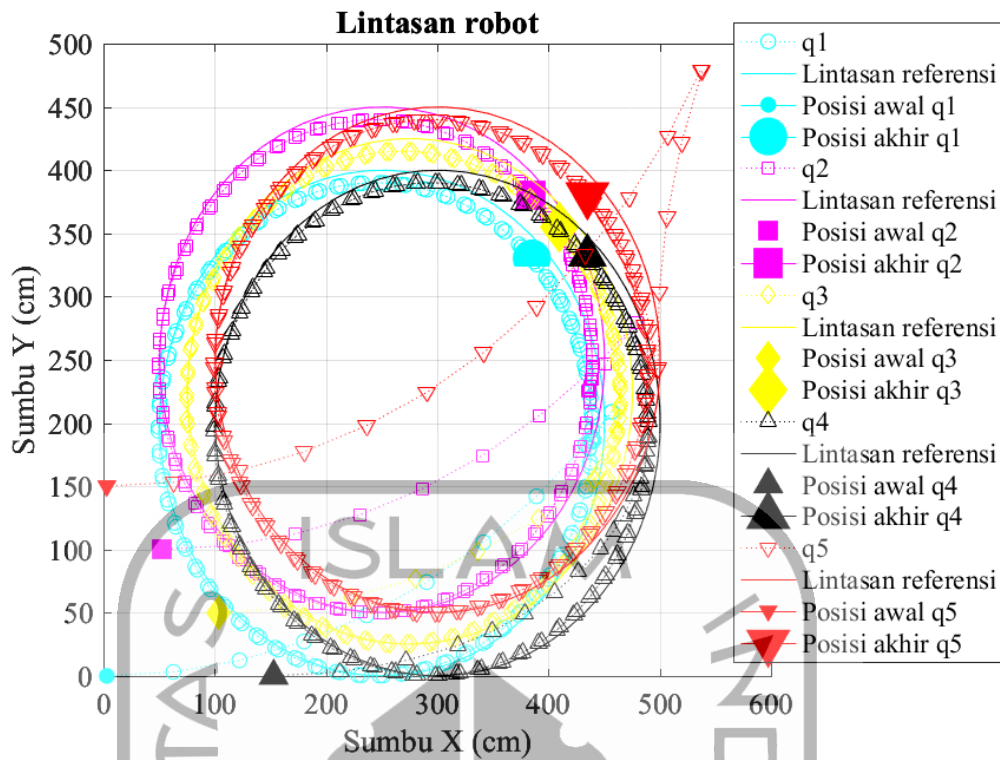
Untuk penelitian simulasi lintasan pertama dilakukan dengan $k_o = 5$, $\alpha = 10$, dan $\beta = 0.5$. Selain itu nilai *weighted value* setiap robot terhadap *virtual leader* (b_{ij}) dan *weighted value* antar robot (a_{ij}) menggunakan nilai berikut ini:

$$b_j = [10\ 10\ 10\ 10\ 10]$$

dan

$$a_{ij} = \begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & i/j \\ \hline & 0 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 0 & 1 & 1 & 1 & 2 \\ & 1 & 1 & 0 & 1 & 1 & 3 \\ & 1 & 1 & 1 & 0 & 1 & 4 \\ & 1 & 1 & 1 & 1 & 0 & 5 \end{array}$$

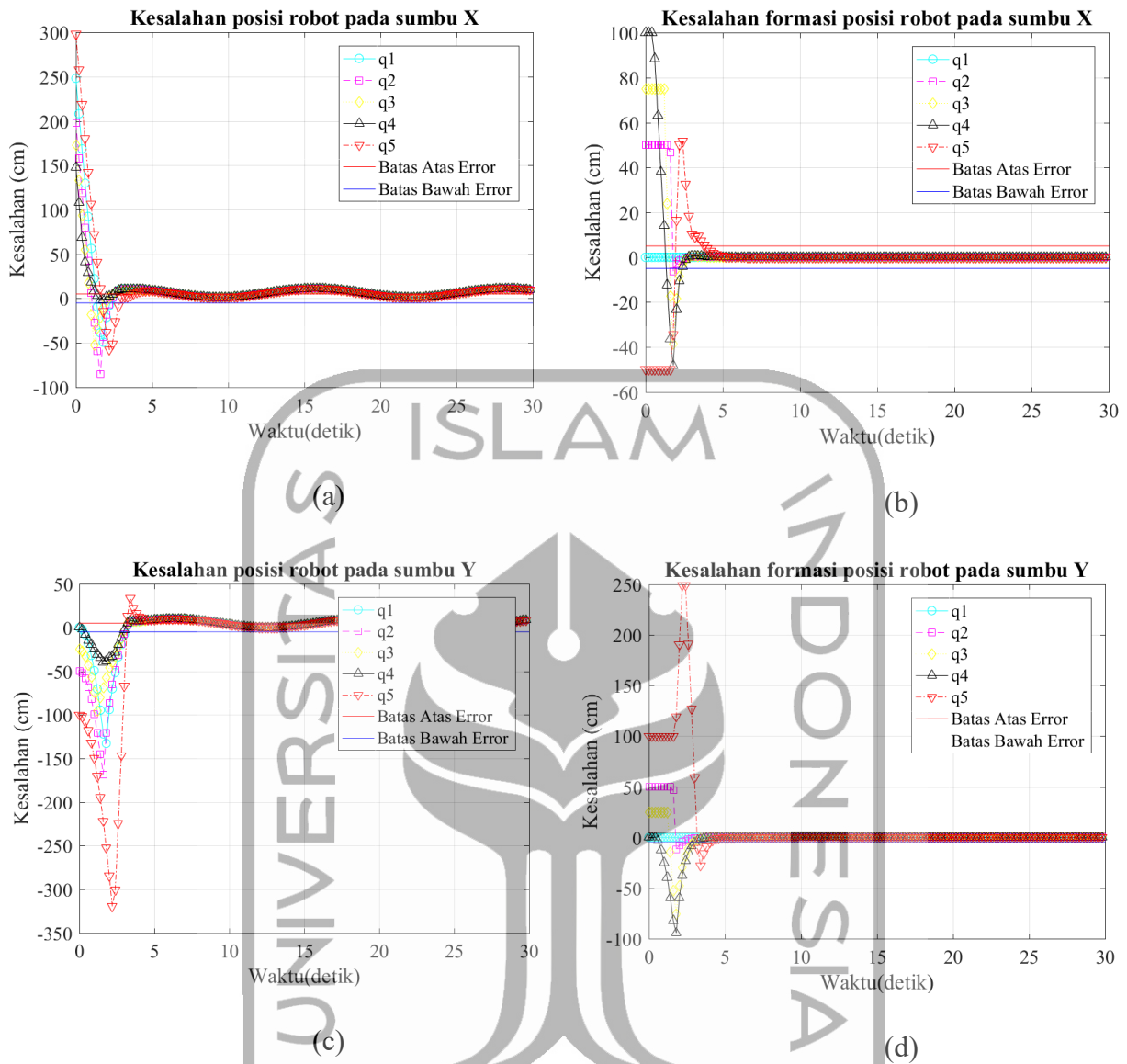
Pada proses simulasi lintasan pertama, posisi *virtual leader* di (250,0) dan bergerak $v=1\text{m/s}$ dan $\omega = 0,5\text{ rad/s}$. Bentuk lintasan yang dihasilkan dari simulasinya ditunjukkan oleh Gambar 4.1



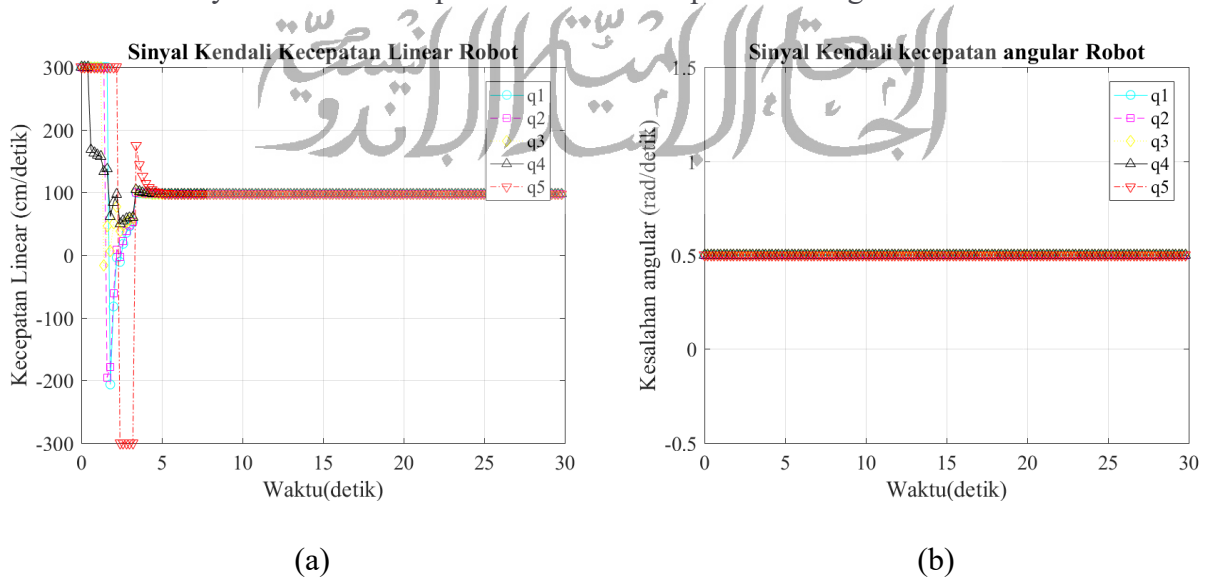
Gambar 4.1: Lintasan robot pada simulasi lintasan pertama dengan metode konsensus

Selain lintasan robot, terdapat variabel-variabel seperti kesalahan posisi dan formasi yang dihasilkan pada Gambar 4.2.

UNIVERSITAS ISLAM INDONESIA
 المعهد الإسلامي للدراسات والبحوث



Gambar 4.2: Sinyal kendali robot pada kasus lintasan pertama dengan metode konsensus



Gambar 4.3: Sinyal kendali robot pada kasus lintasan pertama dengan metode konsensus

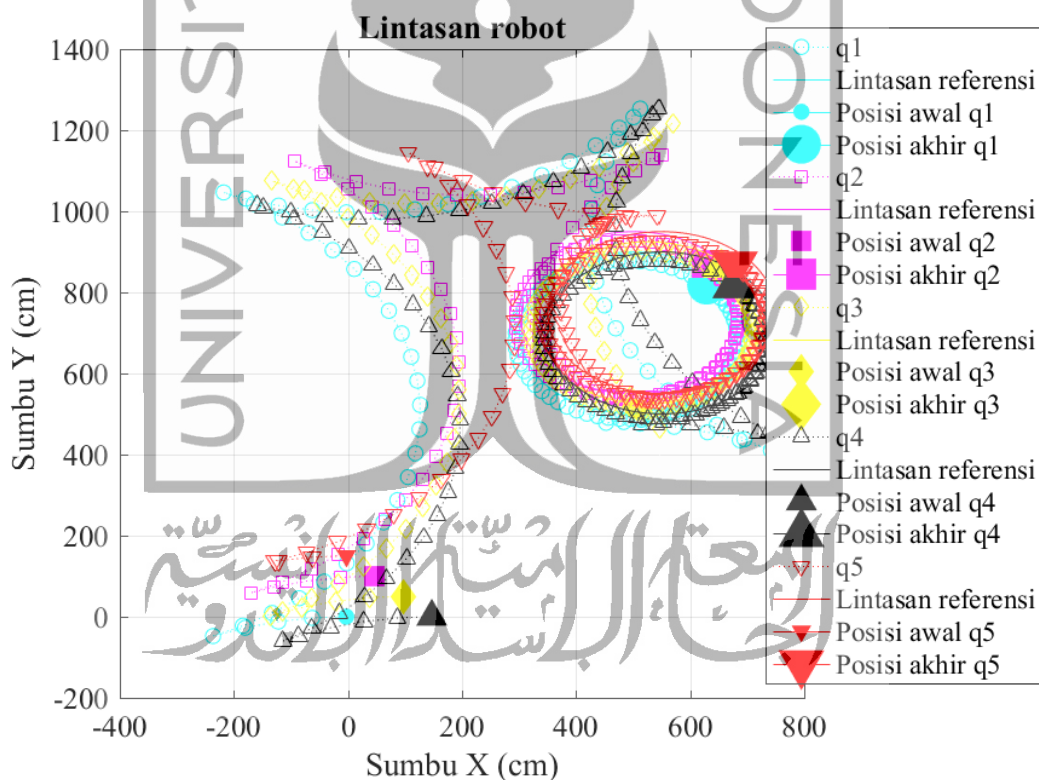
Selanjutnya untuk simulasi lintasan kedua maka variabel kendali yang digunakan adalah sebagai berikut $k_o = 5$, $\alpha = 10$, dan $\beta = 0.5$. Selain itu dengan nilai *weighted value* sebagai berikut:

$$b_j = [10 \ 10 \ 10 \ 10 \ 10]$$

dan

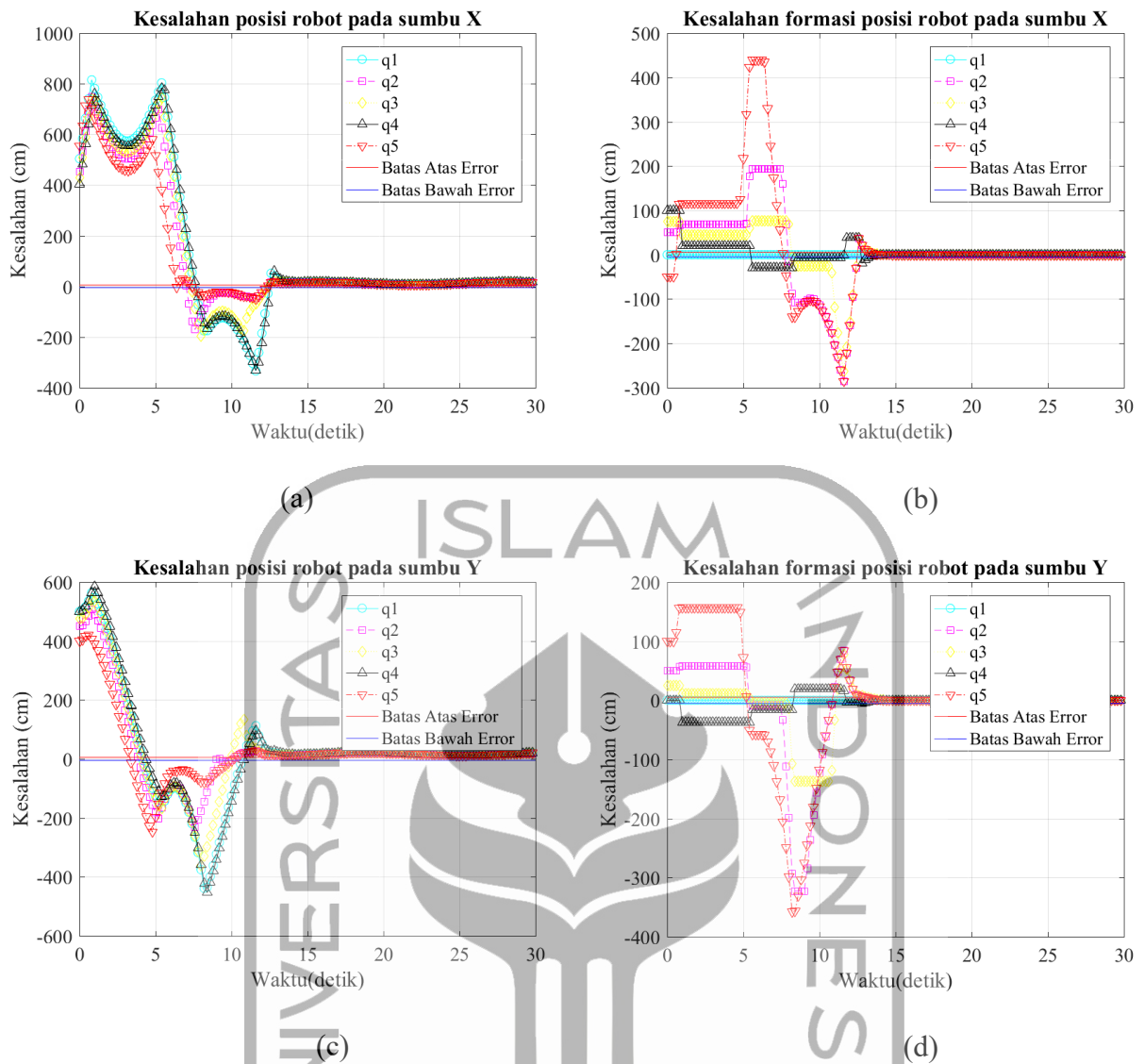
$$a_{ij} = \begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & i/j \\ \hline 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 1 & 2 \\ 3 & 1 & 1 & 0 & 1 & 1 & 3 \\ 4 & 1 & 1 & 1 & 0 & 1 & 4 \\ 5 & 1 & 1 & 1 & 1 & 0 & 5 \end{array}$$

Pada proses simulasi lintasan kedua, posisi *virtual leader* di (500,500) dan bergerak $v = 1$ m/s dan $\omega = 0.5$ rad/s, hasil lintasan robot pada simulasi lintasan kedua ditunjukkan pada Gambar 4.5.

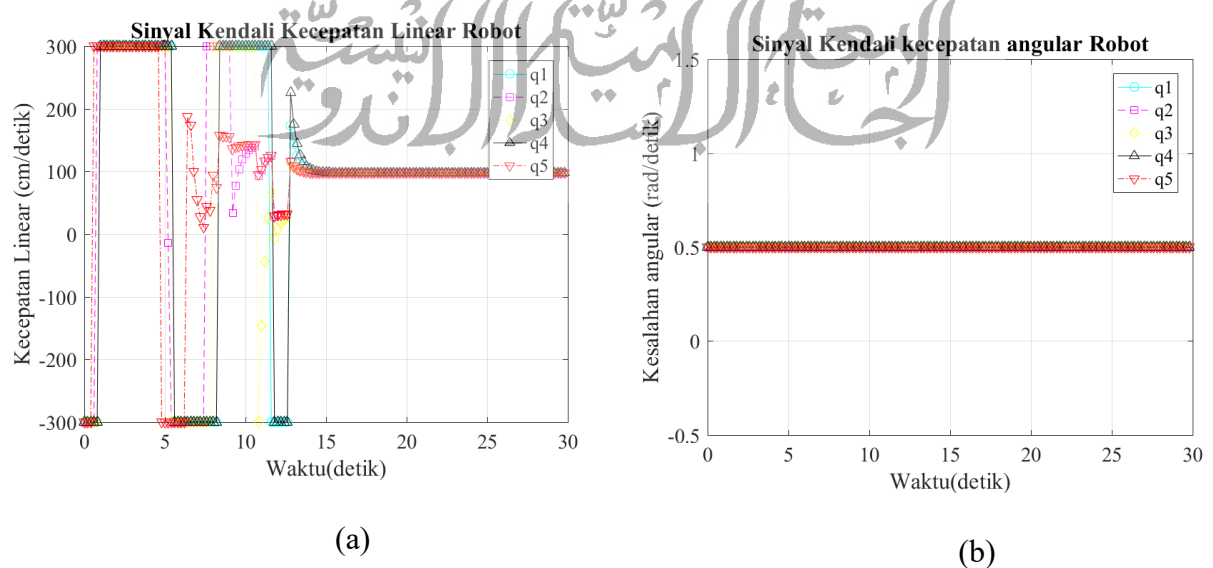


Gambar 4.4: Lintasan robot pada simulasi lintasan kedua dengan metode konsensus

Dan ilustrasi *error* pada simulasi ditunjukkan pada Gambar 4.6 dibawah ini.



Gambar 4.5 Grafik *error* pada kasus lintasan kedua dengan metode berbasis konsensus.

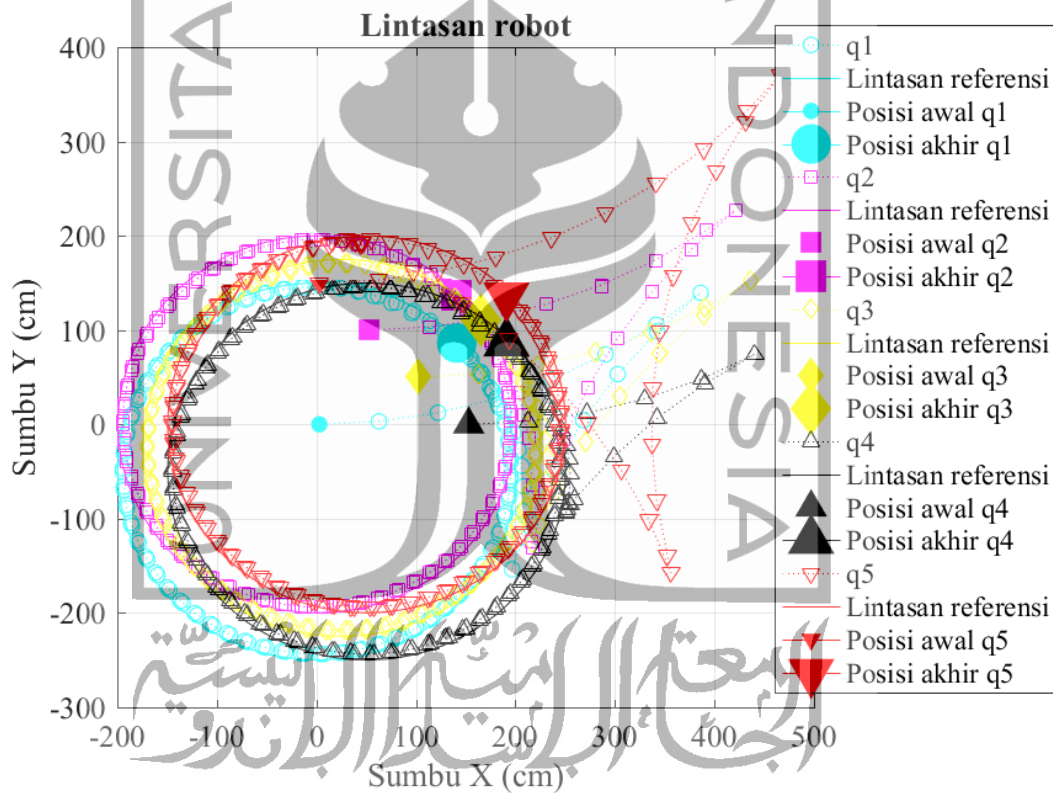


Gambar 4.6: Sinyal kendali robot pada kasus lintasan kedua dengan metode konsensus

Untuk penelitian simulasi lingkaran di sekitar posisi awal dilakukan dengan variabel kendali $k_o = 5$, $\alpha = 10$, dan $\beta = 0.5$. Selain itu, juga dengan nilai *weighted valued* di bawah ini $b_j = [10 \ 10 \ 10 \ 10 \ 10]$ dan

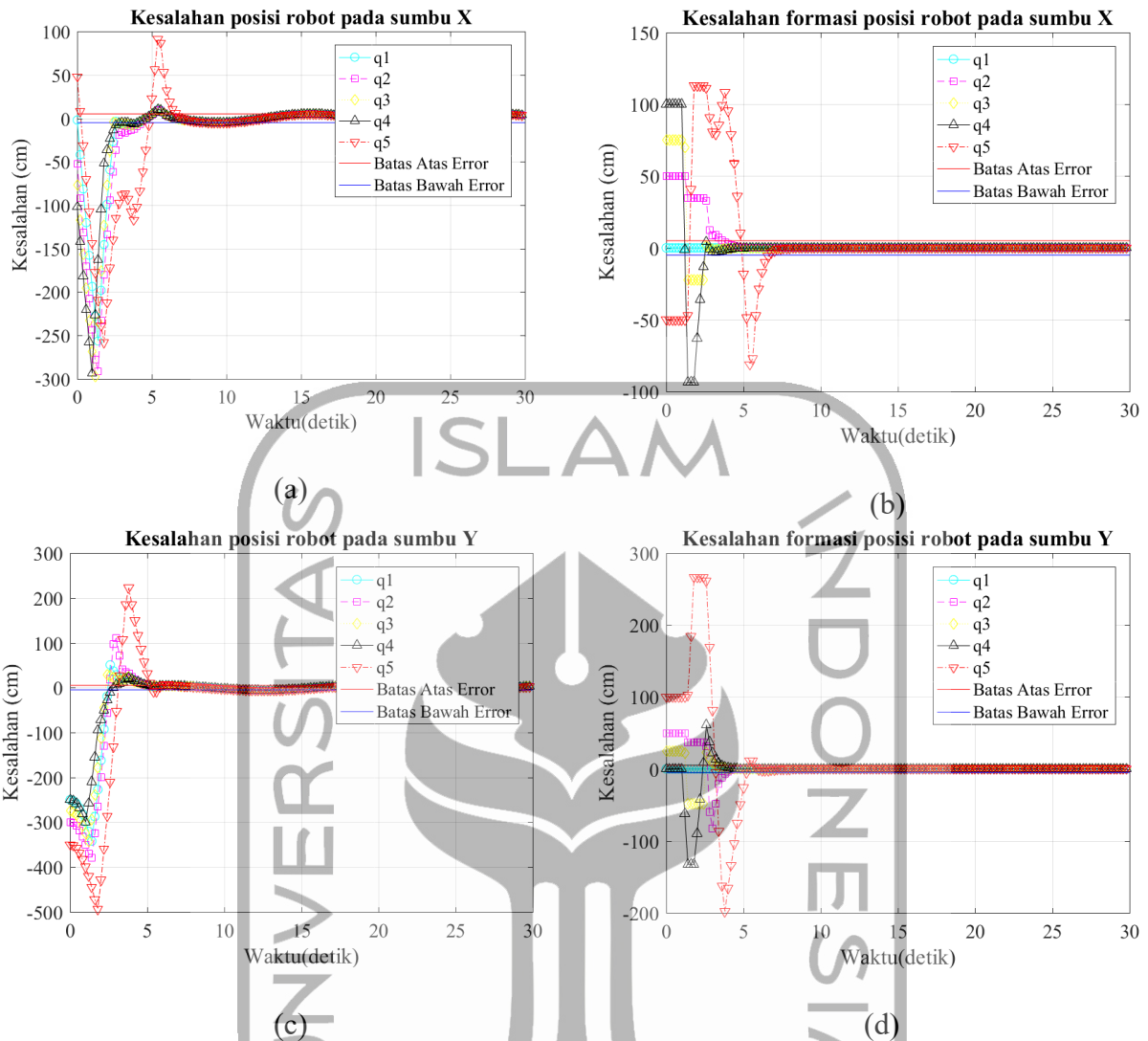
	1	2	3	4	5	<i>i/j</i>
$a_{ij} =$	0	1	1	1	1	1
	1	0	1	1	1	2
	1	1	0	1	1	3
	1	1	1	0	1	4
	1	1	1	1	0	5

Pada proses simulasi, posisi *virtual leader* di $(0, -250)$ dan bergerak $v = 1 \text{ m/s}$ dan $\omega = 0.5 \text{ rad/s}$, hasil dari simulasinya adalah sebagai berikut,

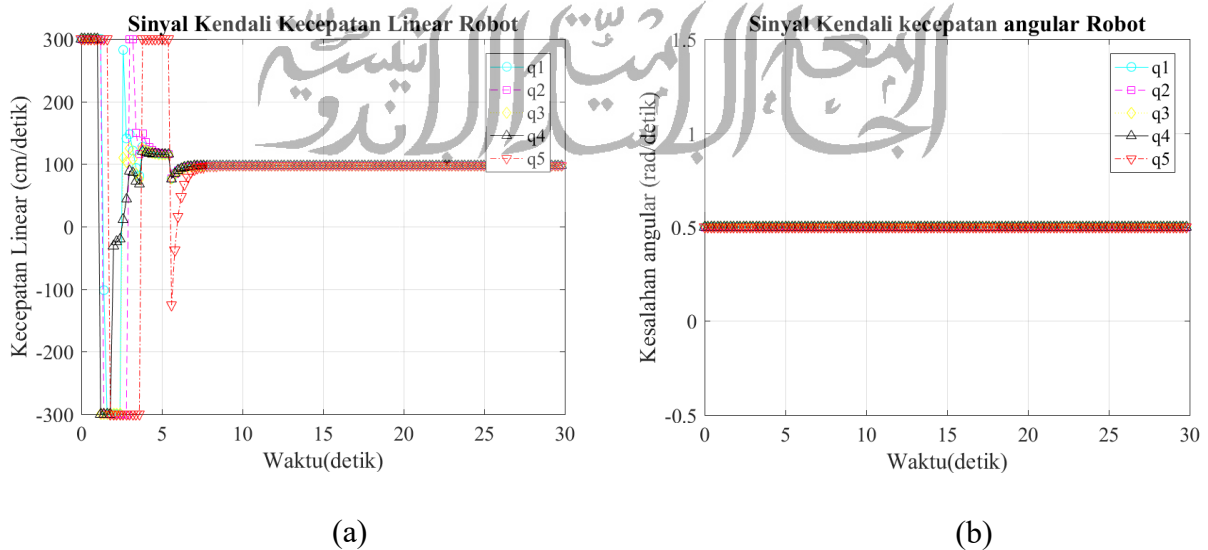


Gambar 4.7: Lintasan robot pada simulasi lintasan ketiga dengan metode konsensus

Selanjutnya berdasarkan simulasi tersebut, grafik kesalahannya adalah sebagai berikut :

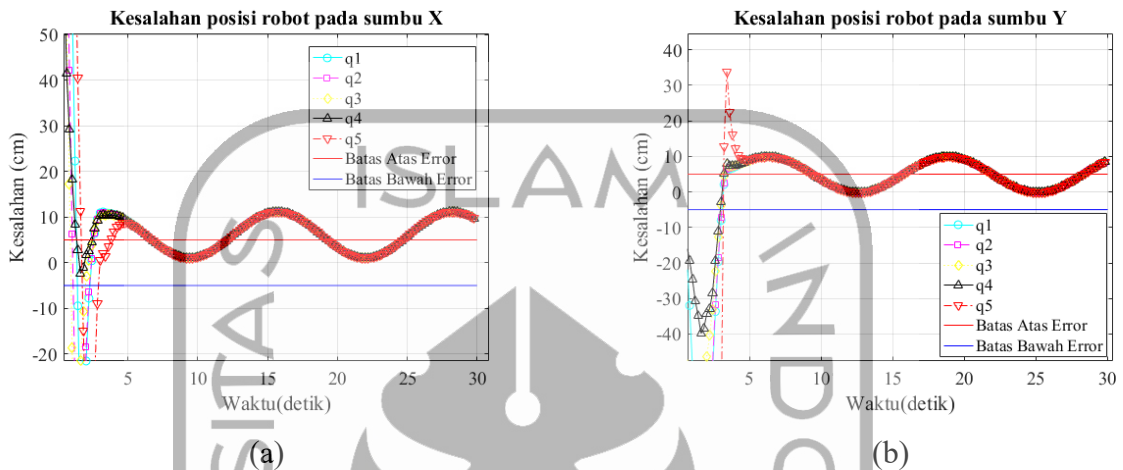


Gambar 4.8 Grafik *error* pada kasus lintasan ketiga dengan metode berbasis konsensus.



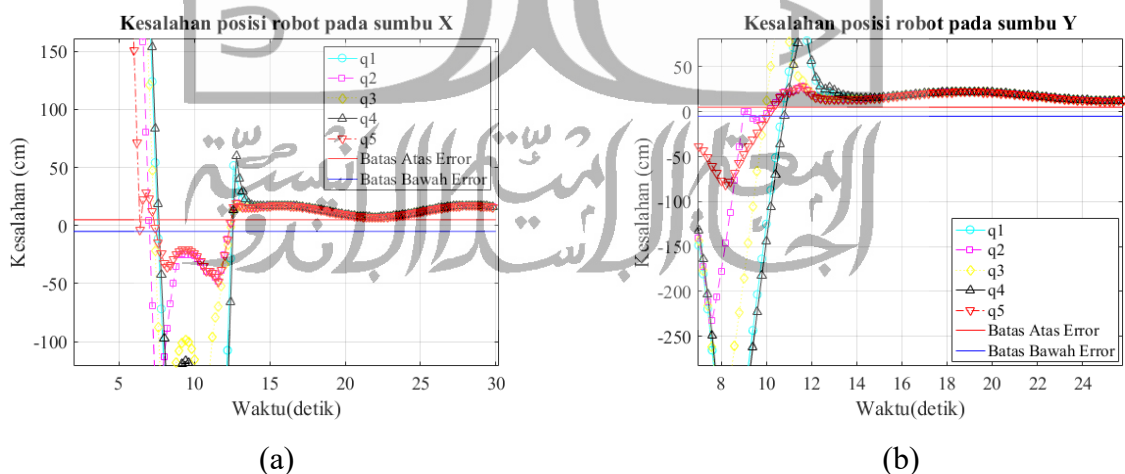
Gambar 4.9: Sinyal kendali robot pada kasus lintasan ketiga dengan metode konsensus

Hasil dari semua simulasi di atas menghasilkan keunikan dari pengendali berbasis konsensus terdistribusi pada setiap robotnya. Pada simulasi lintasan pertama dengan metode konsensus, semua robot memiliki parameter yang mencapai nilai sama dan stabil. Berdasarkan Gambar 4.1, robot tidak mencapai target yang diinginkan, Hal tersebut juga dibuktikan dengan dengan osilasi pada *error* posisi robot pada Gambar 4.4. Walaupun demikian robot mencapai nilai yang stabil jika dilihat pada Gambar 4.3, sinyal kendali pada setiap robot dan mencapai formasi yang diinginkan.

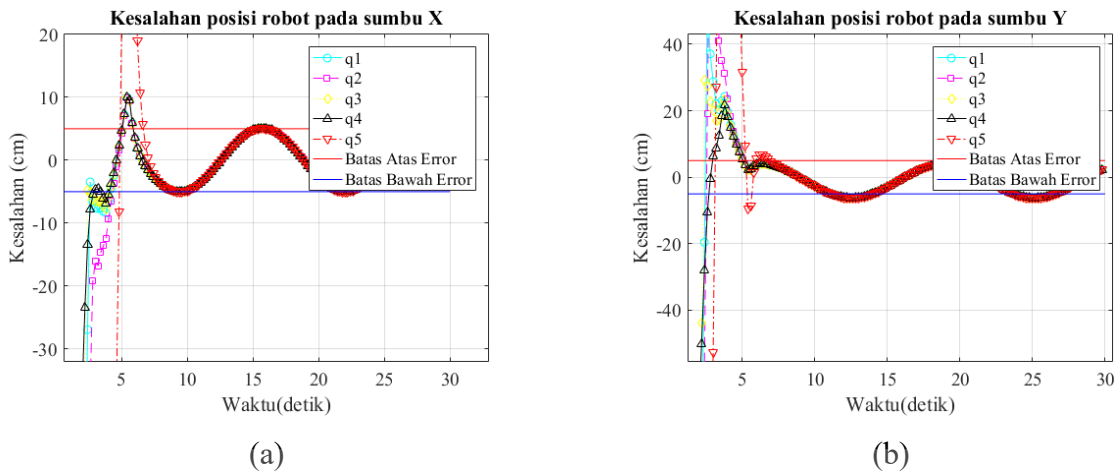


Gambar 4.10: Kesalahan posisi robot yang telah diperbesar pada simulasi lintasan ketiga dengan metode konsensus

Hal serupa juga terjadi pada simulasi lintasan kedua dan lingkaran di sekitar posisi awal berdasarkan Gambar 4.11 – 4.12. Pada simulasi lintasan kedua dan lintasan ketiga, nilai *error* tidak bernilai diantara batas atas serta batas bawah *error* yang yang dituju.



Gambar 4.11: Kesalahan posisi robot pada grafik yang telah diperbesar pada simulasi lintasan ketiga dengan metode konsensus



Gambar 4.12: Kesalahan posisi robot yang telah diperbesar pada simulasi lintasan ketiga dengan metode konsensus

Dari semua hasil diatas, pada pengendali berbasis konsensus semua mencapai formasi yang ingin dibentuk walaupun mencapai target yang diinginkan, yang bahkan sangat memiliki *error* yang cukup besar seperti pada simulasi pertama. Hal ini dikarenakan pada pengendali ini bertujuan untuk mencapai suatu nilai konsensus yang dicapai berdasarkan sinyal kendali pada setiap robot yang ditentukan oleh nilai robot-robot lainnya. Nilai konsensus yang dicapai pada pengendali ini untuk menyesuaikan jarak antar robot agar sesuai formasi yang secara matematis memperkecil kesalahan posisi pada formasi pada Persamaan (3.26) – (3.27) menjadi Persamaan (4.1) – (4.2).

$$\lim_{t \rightarrow \infty} e_{px}^i(t) = 0 \tag{4.1}$$

$$\lim_{t \rightarrow \infty} e_{py}^i(t) = 0 \tag{4.2}$$

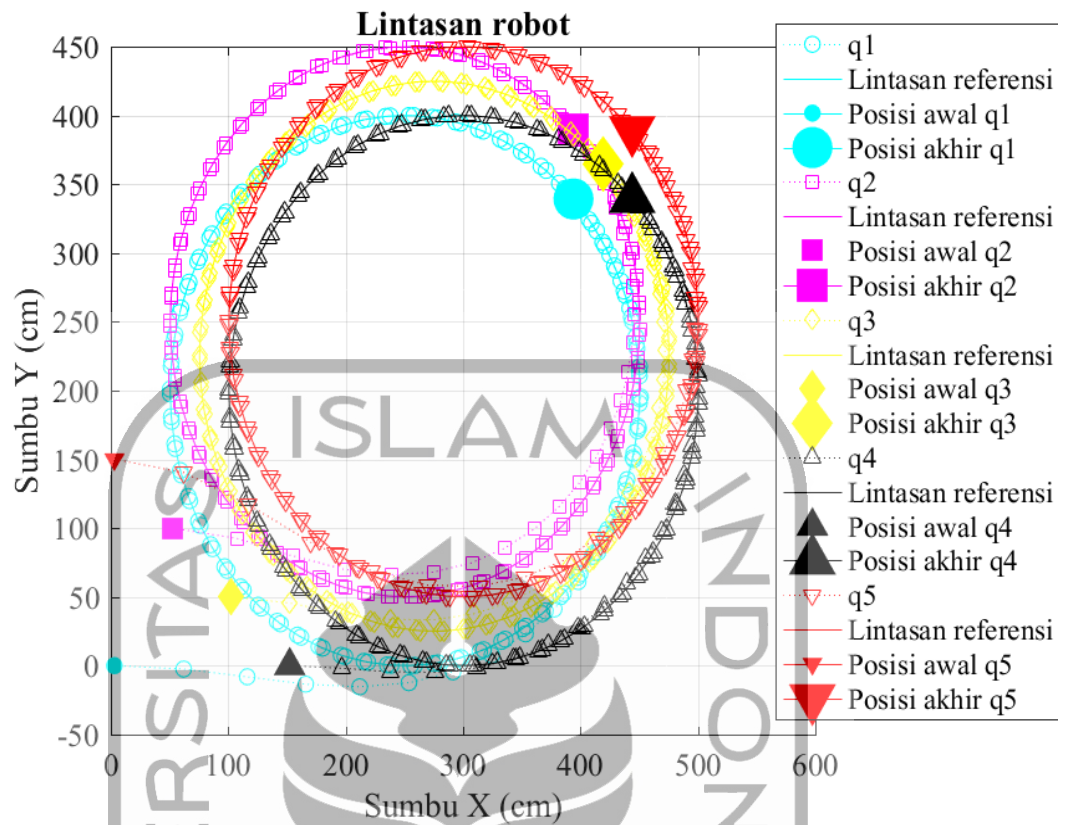
Selain itu, pengaruh oleh *weighted value* pada setiap robot juga menjadi faktor keberhasilan pada pengendalian ini. Dengan *weighted value* serta *topology* yang berbeda pada pengendalian, akan menghasilkan hasil simulasi yang sangat berbeda secara signifikan. Semua simulasi diatas menggunakan *topologi mesh* dan besaran *weighted value* yang digunakan memiliki nilai yang sama.

4.1.2 Pengendalian berbasis *Vector Field*

Sama seperti pada pengendalian berbasis konsensus terdistribusi, pada pengendalian ini juga menentukan variabel kendalinya terlebih dahulu. Kemudian, simulasi akan menghasilkan lintasan pada setiap robot beserta sinyal kendalinya untuk dianalisa. Hasil simulasi pada pengendali berbasis *vector field* akan ditampilkan dibawah ini.

Pertama, Simulasi lingkaran dekat dengan metode *vector field* dengan sinyal kendalinya berikut, $k_t = 1$, $k_d = 0.01$, $k_0 = 1$, dan $\sigma = 1$. Dan pada simulasi lingkaran dekat, posisi *virtual*

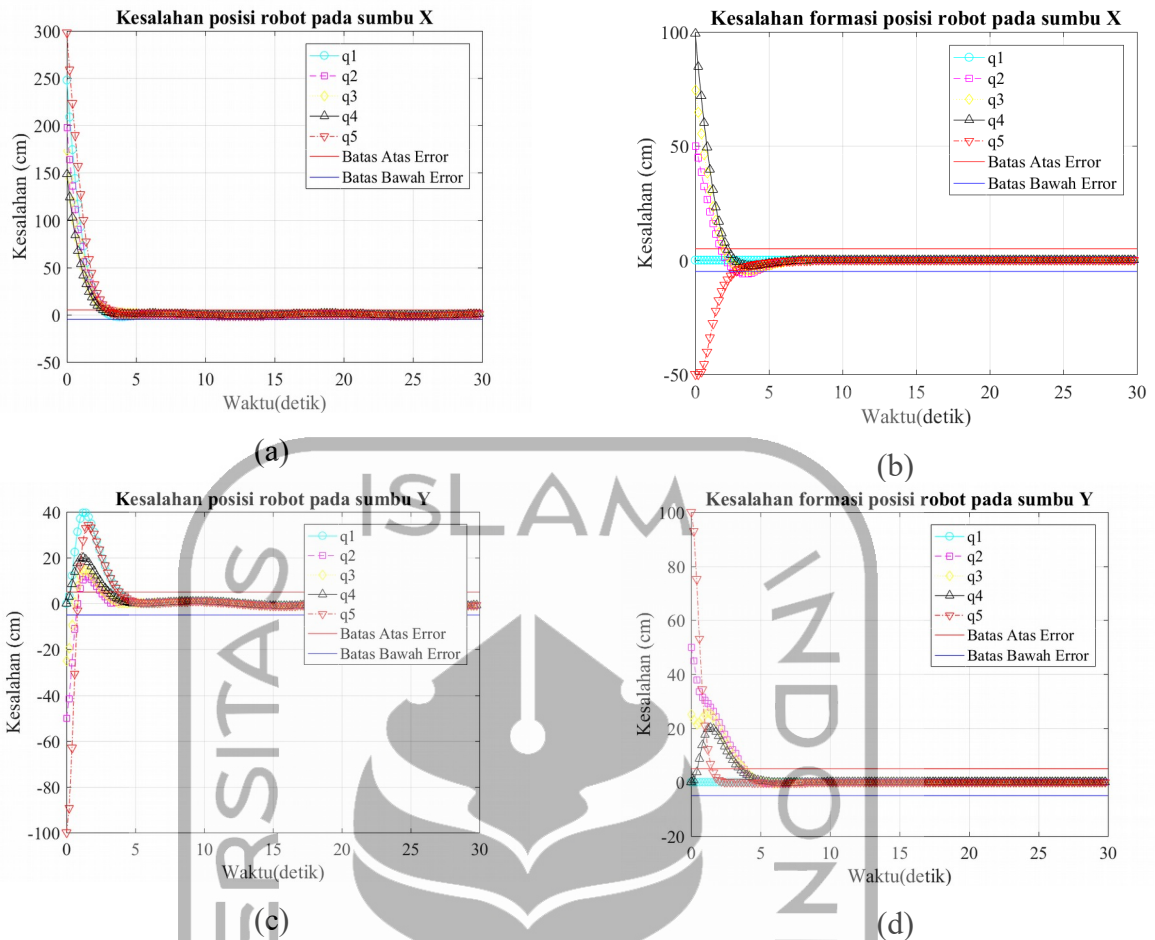
leader di (250, 0) dan bergerak $v = 1 \text{ m/s}$ dan $\omega = 0.5 \text{ rad/s}$, hasil dari simulasinya adalah sebagai berikut, Dan lintasan robot dari simulasi tersebut bisa dilihat pada Gambar 4.13



Gambar 4.13: Lintasan robot pada simulasi lintasan pertama dengan metode *vector field*

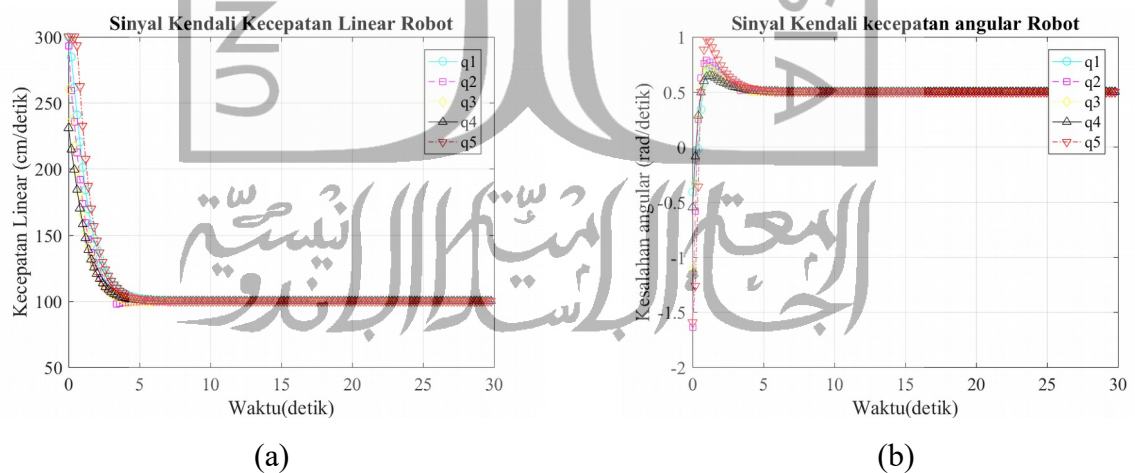
الجامعة الإسلامية
الاستد بالاندية

Selanjutnya grafik kesalahan pada robot adalah berikut :



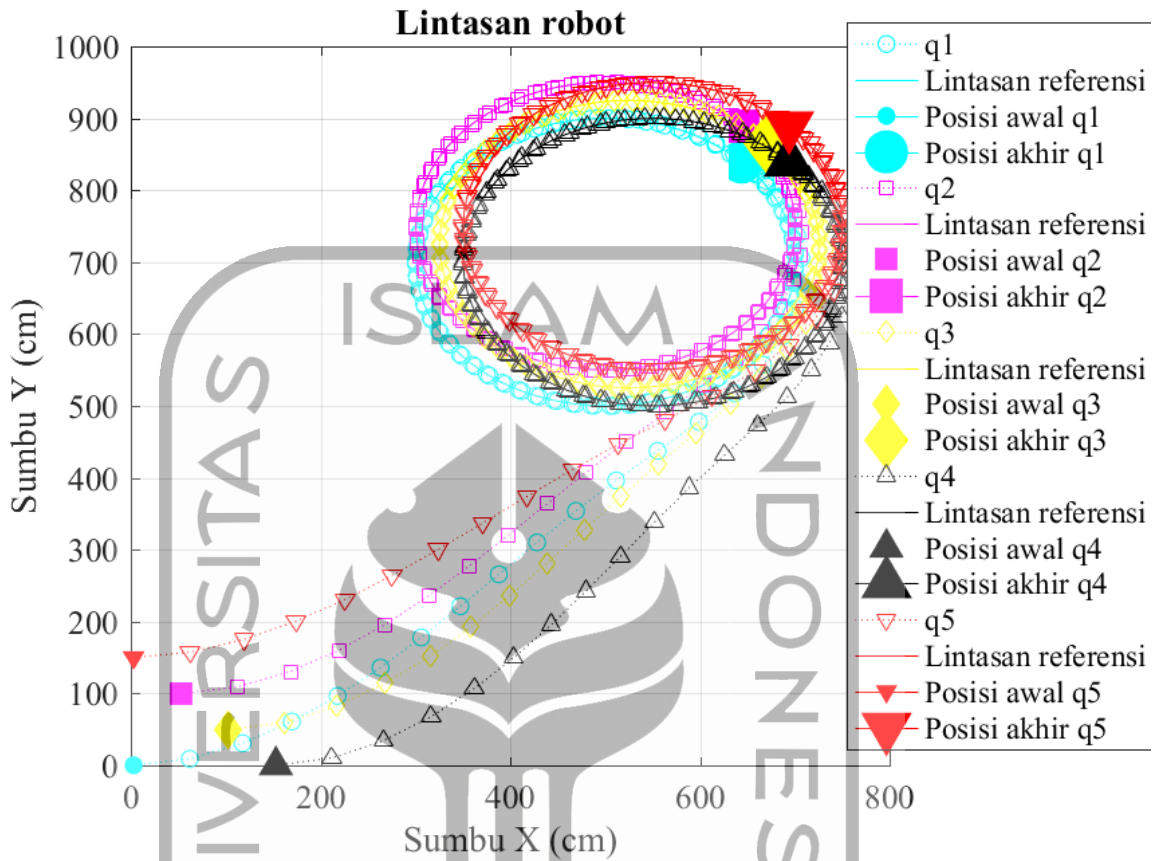
Gambar 4.14 Grafik *error* pada kasus lintasan pertama dengan metode *vector field*

Kemudian sinyal kendalinya adalah sebagai berikut:



Gambar 4.15: Sinyal kendali robot pada kasus lintasan pertama dengan metode *vector field*

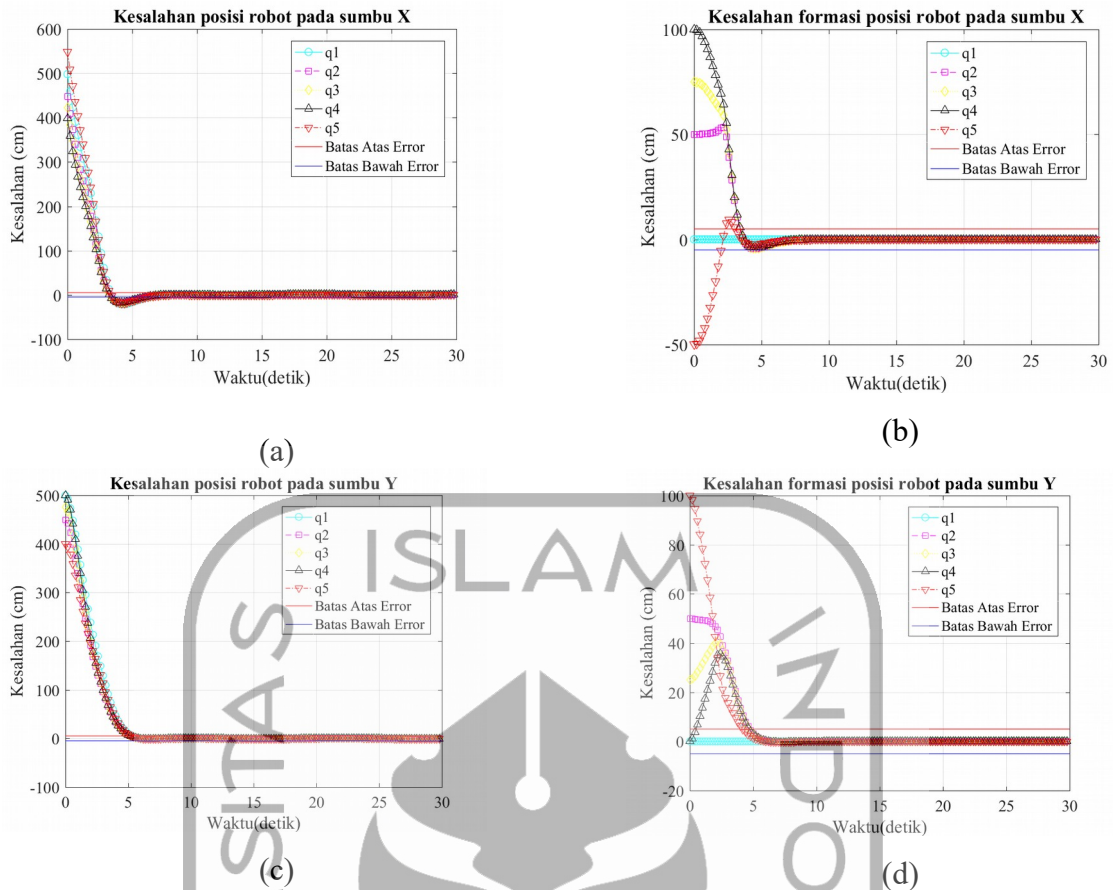
Kedua, Simulasi lintasan kedua dengan metode *vector field* dengan sinyal kendalinya berikut, $k_I = 1$, $k_d = 0.01$, $k_0 = 1$, dan $\sigma = 1$. Dan pada simulasi lingkaran dekat, posisi *virtual leader* di (500, 500) dan bergerak $v = 1 \text{ m/s}$ dan $\omega = 0.5 \text{ rad/s}$, hasil dari simulasinya adalah sebagai berikut,



Gambar 4.16: Lintasan robot pada lintasan kedua dengan metode *vector field*

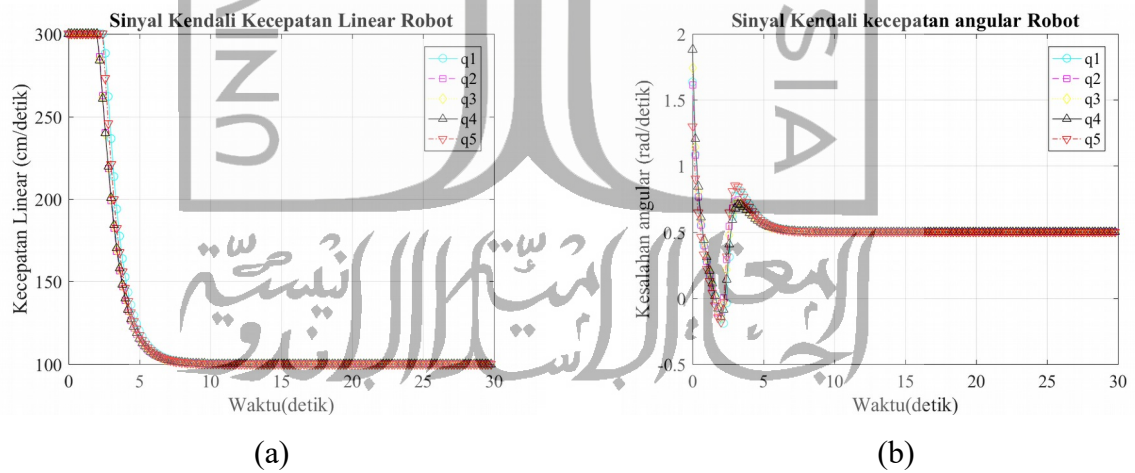
dan grafik *kesalahan* pada tiap robot ditampilkan sebagai berikut,





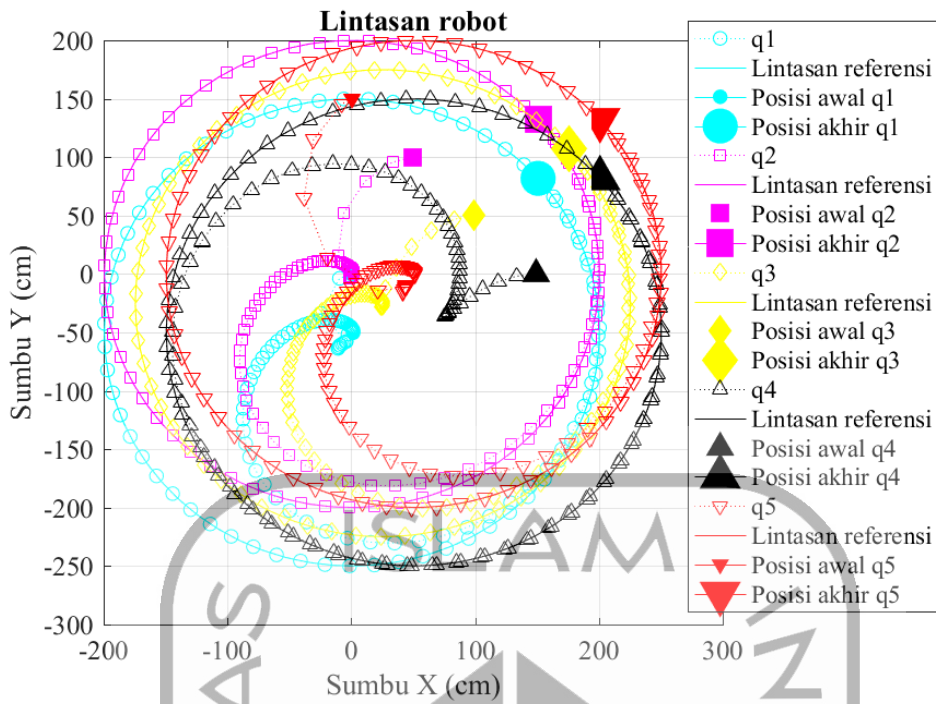
Gambar 4.17 Grafik *error* pada kasus lintasan kedua dengan metode *vector field*

Dan sinyal kendali pada simulasi lintasan kedua ditunjukkan pada Gambar 4.18 dibawah ini,



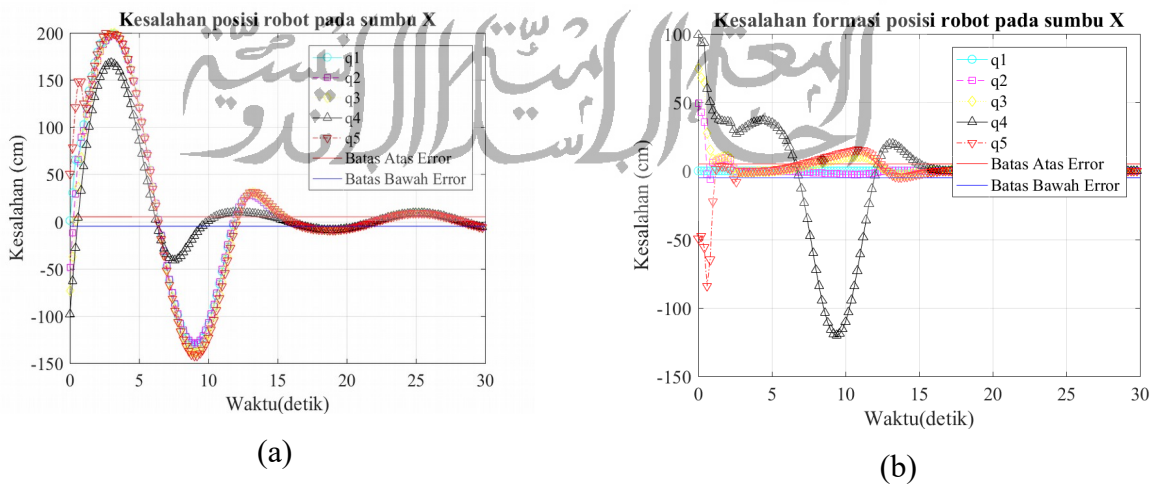
Gambar 4.18: Sinyal kendali robot pada kasus lintasan kedua dengan metode *vector field*

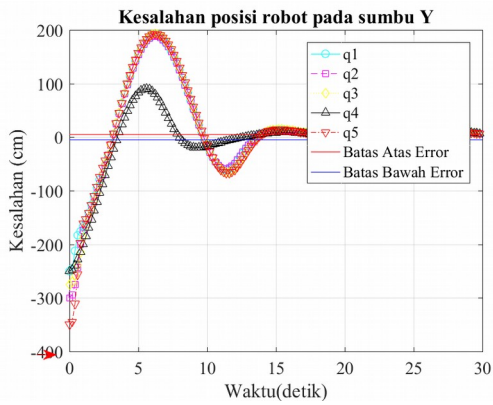
Kedua, Simulasi lintasan kedua dengan metode *vector field* dengan sinyal kendalinya berikut, $k_l = 1$, $k_d = 0.01$, $k_\theta = 1$, dan $\sigma = 0.0075$. Dan pada simulasi lingkaran dekat, posisi *virtual leader* di $(0, -250)$ dan bergerak $v = 1 \text{ m/s}$ dan $\omega = 0.5 \text{ rad/s}$, hasil dari simulasinya adalah sebagai berikut,



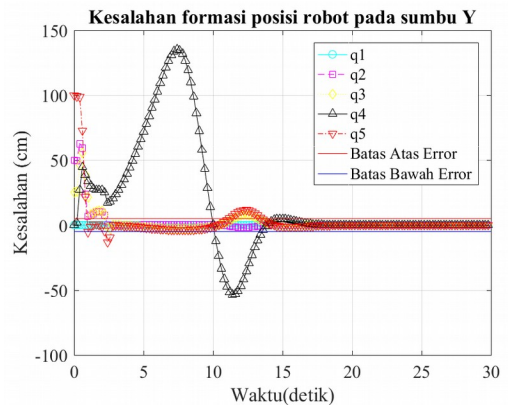
Gambar 4.19: Lintasan robot pada lintasan ketiga dengan metode *vector field*

Kemudian Grafik kesalahannya adalah sebagai berikut,



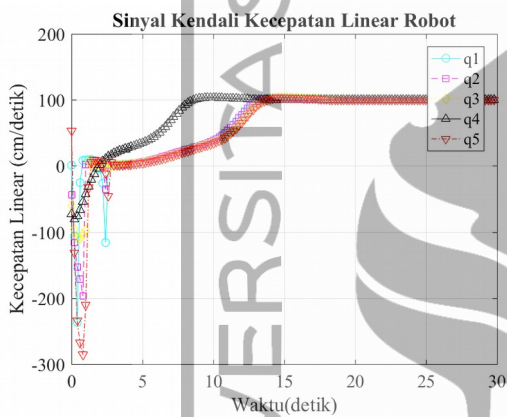


(c)

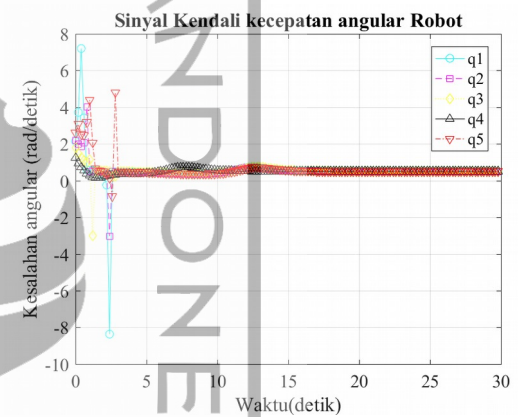


(d)

Gambar 4.20 Grafik *error* pada kasus lintasan ketiga dengan metode *vector field*



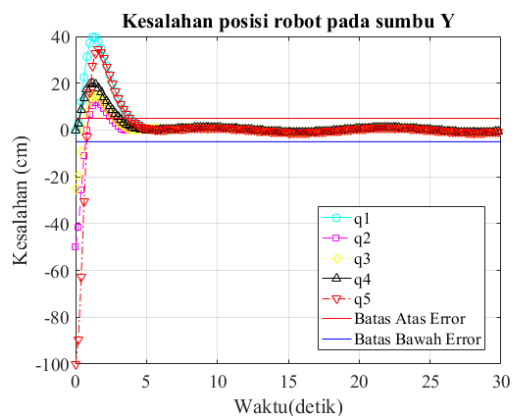
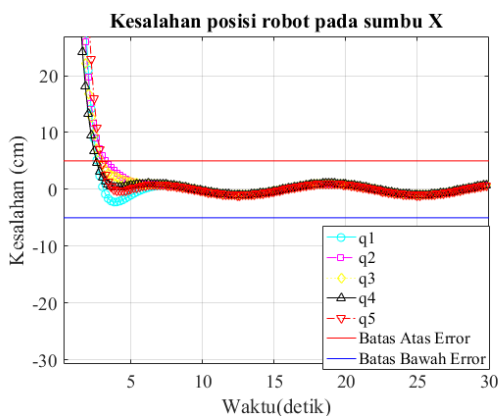
(a)



(b)

Gambar 4.21: Sinyal kendali robot pada kasus lintasan ketiga dengan metode *vector field*

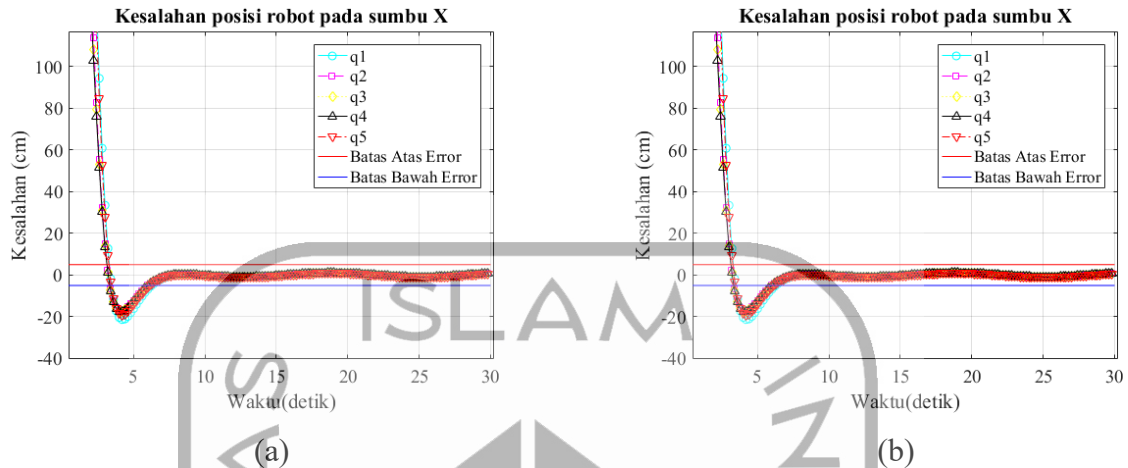
Hasil simulasi dengan pengendali berbasis *vector field* semuanya mencapai target bahkan lebih baik daripada pengendali berbasis konsensus terdistribusi seperti yang ditunjukkan pada Gambar 4.13 – 4.14, Gambar 4.16 – 4.17 dan Gambar 4.19 – 4.20. Hal ini ditunjukkan dari *error* posisi yang masih dijangkau batas atas dan batas bawah *error*. Lebih jelasnya bisa dilihat pada Gambar 4.22 – 4.24



(a)

(b)

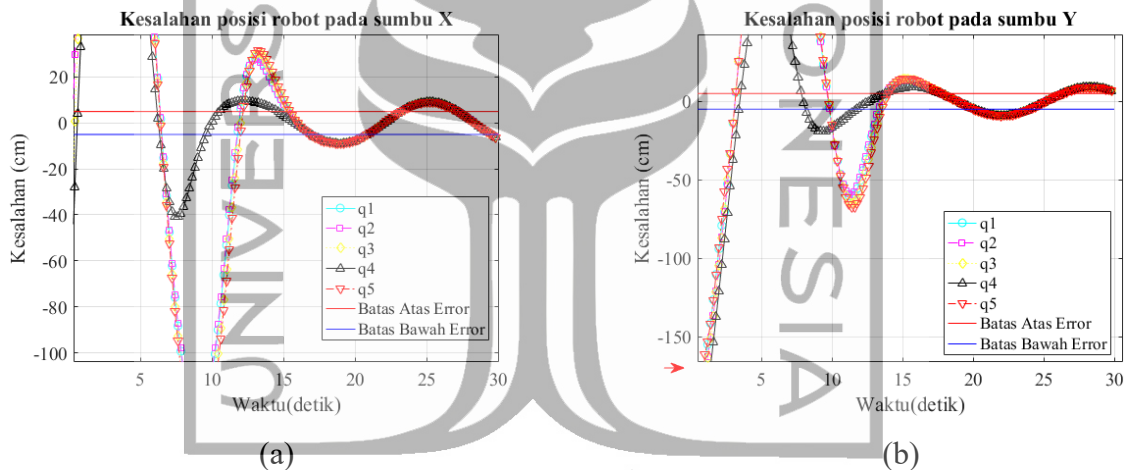
Gambar 4.22: Kesalahan Posisi robot pada simulasi lintasan ketiga dengan metode *vector field* yang diperbesar



(a)

(b)

Gambar 4.23: Kesalahan Posisi robot pada simulasi lintasan ketiga dengan metode *vector field* yang diperbesar



(a)

(b)

Gambar 4.24: Kesalahan Posisi robot pada simulasi lintasan ketiga dengan metode *vector field* yang diperbesar

Berkaitan pada semua hasil simulasi diatas, ditunjukkan bahwa pengendalian *vector field* memang cepat mencapai target. Berbeda dengan pengendali berbasis konsensus, pengendali berbasis *vector field* bertujuan untuk setiap robot mencapai posisi target yang secara perhitungan untuk memperkecil kesalahan posisi aktual dengan posisi target. Atau secara matematis bisa menjadi dituliskan pada persamaan (4.3) – (4.4)

$$\lim_{t \rightarrow \infty} e_x^i(t) = 0 \quad (4.3)$$

$$\lim_{t \rightarrow \infty} e_y^i(t) = 0 \quad (4.4)$$

Dengan tujuan pengendalian yang telah dijelaskan, maka hasil pada simulasi 1 posisi awal tidak mencapai target pada beberapa robot, namun juga ada beberapa robot yang berhasil. Hal ini dikarenakan memang sinyal kendali robot tidak dipengaruhi oleh robot lainnya.

4.2 Analisis Hasil Simulasi

Perbandingan hasil dengan metode pengendalian yang berbeda disajikan pada Tabel 4.1

Tabel 4.1 Perbandingan hasil simulasi pada kedua metode

Tabel 4.1 Hasil perbandingan simulasi pada setiap lintasan

Lintasan	Kondisi	Metode Konsensus Terdistribusi	Metode <i>Vector Field</i>
Pertama	Mencapai Target	Tidak	Ya
	Sesuai Formasi	Ya	Ya
Kedua	Mencapai Target	Tidak	Ya
	Sesuai Formasi	Ya	Ya
Ketiga	Mencapai Target	Tidak	Ya
	Sesuai Formasi	Ya	Ya

Selain itu juga, ditunjukkan bahwa dengan pengendalian *vector field* lebih cepat mencapai posisi yang diinginkan ketimbang metode pengendalian berbasis konsensus terdistribusi. Dan juga sangat pengendali *vector field* sangat mendekati posisi target daripada dengan metode pengendalian berbasis konsensus terdistribusi. Namun pada hasil simulasi ketepatan formasi pengendalian konsensus terdistribusi lebih diunggulkan daripada pengendalian *vector field*.

Hasil diatas telah dijelaskan pada subbab sebelumnya, bahwa pada pengendalian berbasis konsensus terdistribusi bertujuan untuk mencapai nilai konsensus, sehingga dari setiap robot tidak mencapai target namun mampu membentuk formasi. Sedangkan dengan metode *vector field*, hampir semua robot mencapai target tapi sebagai individu yang dibentuk dari formasi yang diinginkan.

Oleh karena itu, untuk merancang pengendalian multi robot yang bertujuan untuk mencapai suatu formasi dengan penyeragaman suatu nilai seperti sinyal kendali, atau jarak antar robot walaupun tidak mencapai posisi target pengendalian berbasis konsensus terdistribusi merupakan hal yang tepat. Namun untuk pengendalian agar sesuai dengan posisi target maka pengendalian *vector field* lebih diunggulkan.

