

BAB IV HASIL DAN ANALISIS

4.1 Pengambilan data

Pengambilan data dalam penelitian ini pengumpulan data data gerakan tubuh. Proses pengambilan data bertujuan untuk memperoleh data yang akan digunakan untuk menganalisis dan menguji sistem pengendalian kursi roda yang dikembangkan.

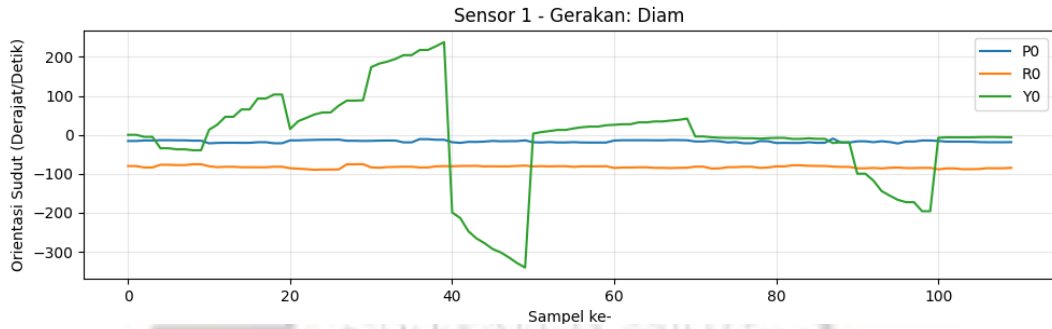
4.2 Grafik Gerakan Tubuh

Gerakan tubuh pada penelitian ini mencakup gerakan diam, fleksi, ekstensi, lateral bending kanan, dan lateral bending kiri. Masing-masing gerakan dapat dibedakan melalui perubahan pada nilai *pitch*, *roll*, dan *yaw* yang dideteksi oleh sensor MPU-6050. Pengelompokan gerakan diidentifikasi menggunakan grafik dengan sumbu Y yang menunjukkan nilai *pitch*, *roll*, dan *yaw*, dan sumbu X menunjukkan urutan data. Grafik masing-masing gerakan dijelaskan sebagai berikut.

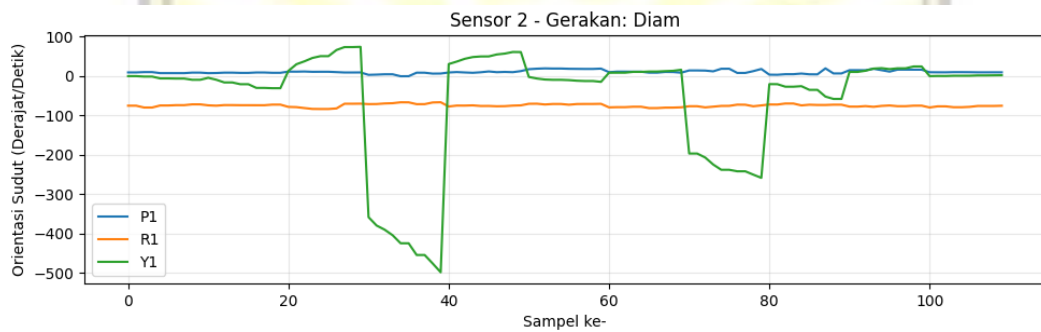
4.2.1 Grafik gerakan Diam

Gerakan diam diperoleh dari postur duduk dengan posisi yang nyaman mungkin. Gerakan diam yang dideteksi oleh sensor 1 memiliki nilai *pitch* pada rentang -26,81 hingga -9,66°, nilai *roll* pada rentang -92,29 hingga -75,32°, dan nilai *yaw* pada rentang -340,61 hingga 585,45°. Gerakan diam yang dideteksi oleh sensor 2 memiliki nilai *pitch* pada rentang -0,23 hingga 19,75°, nilai *roll* pada rentang -83,50 hingga -66,31°, dan nilai *yaw* pada rentang -498,44 hingga 74,45°. Sementara itu, gerakan diam yang dideteksi oleh sensor 3 memiliki nilai *pitch* pada rentang -7,95 hingga 11,01°, nilai *roll* pada rentang -115,00 hingga -81,91°, dan nilai *yaw* pada rentang -75,68 hingga 26,03°. Nilai *pitch*, *roll*, dan *yaw* pada gerakan diam digunakan sebagai acuan untuk membedakan gerakan-gerakan lainnya. Grafik

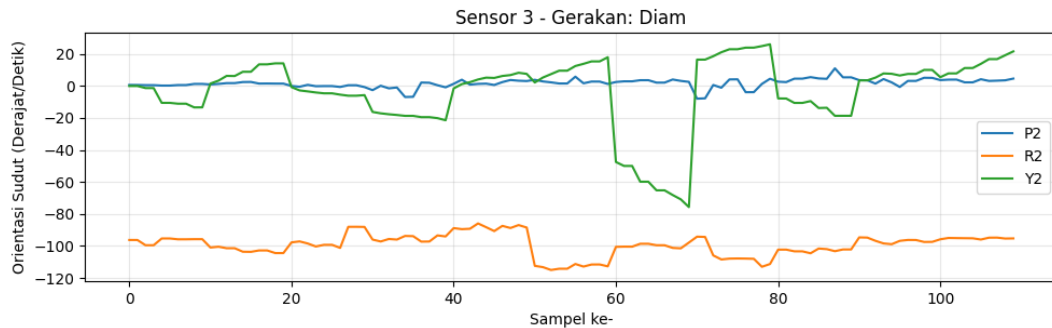
data gerakan diam yang dideteksi oleh sensor 3, 6, dan 7 ditunjukkan secara berurutan pada Gambar 4.1, Gambar 4.2, dan Gambar 4.3.



Gambar 4. 1 Grafik Gerakan Sensor 1 Diam



Gambar 4. 2 Grafik Gerakan Sensor 2 Diam

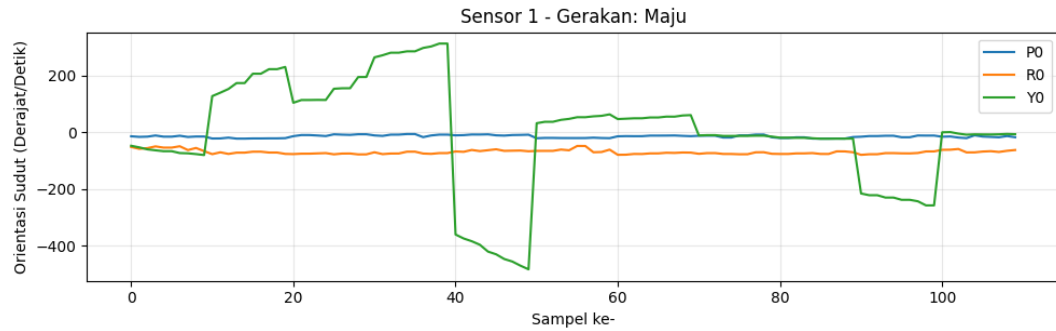


Gambar 4. 3 Grafik Gerakan Sensor 3 Diam

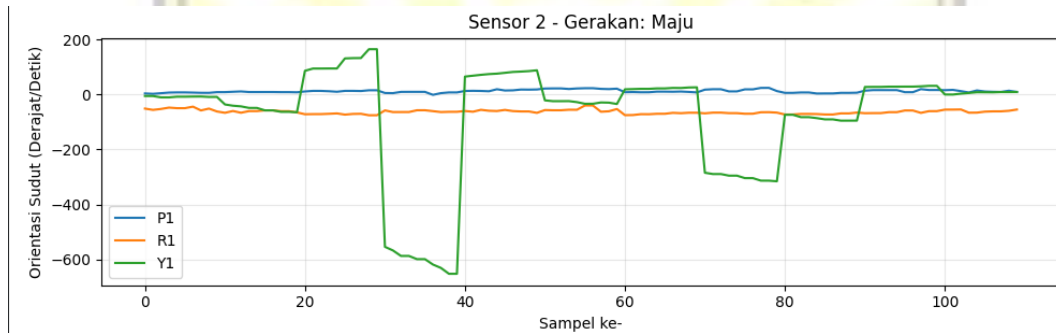
4.2.2 Grafik gerakan Fleksi

Gerakan fleksi yang dideteksi oleh sensor 1 memiliki nilai *pitch* pada rentang $-22,94$ hingga $-2,88^\circ$, nilai *roll* pada rentang $-80,17$ hingga $-48,78^\circ$, dan nilai *yaw* pada rentang $-484,01$ hingga $1236,51^\circ$. Gerakan fleksi yang dideteksi oleh sensor 2 memiliki nilai *pitch* pada rentang $-1,24$ hingga $24,11^\circ$, nilai *roll* pada rentang $-75,48$ hingga $-40,00^\circ$, dan nilai *yaw* pada rentang $-651,63$ hingga $164,64^\circ$. Sementara itu, gerakan fleksi yang dideteksi oleh sensor 3 memiliki nilai *pitch* pada rentang $-9,19$ hingga $12,34^\circ$, nilai *roll* pada rentang $-107,38$ hingga $5,70^\circ$, dan nilai *yaw* pada rentang $-261,20$ hingga $71,03^\circ$. Grafik data gerakan fleksi yang dideteksi

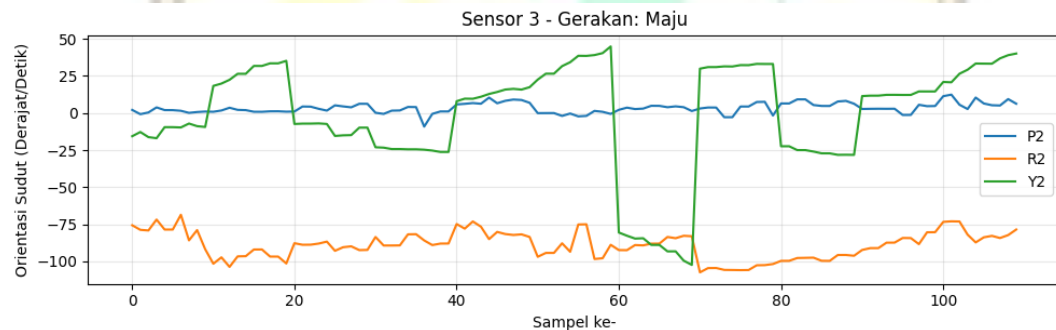
oleh sensor 1, 2, dan 3 ditunjukkan secara berurutan pada Gambar 4.4, 4.5, dan 4.6 berikut.



Gambar 4. 4 Grafik Gerakan Sensor 1 Fleksi



Gambar 4. 5 Grafik Gerakan Sensor 2 Fleksi

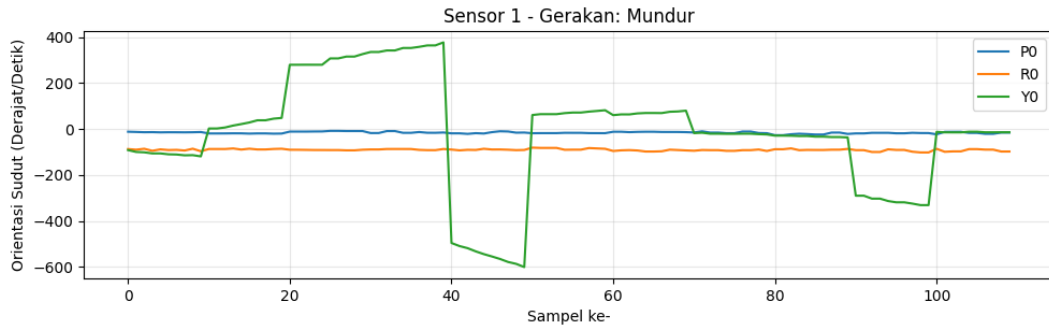


Gambar 4. 6 Grafik Gerakan Sensor 3 Fleksi

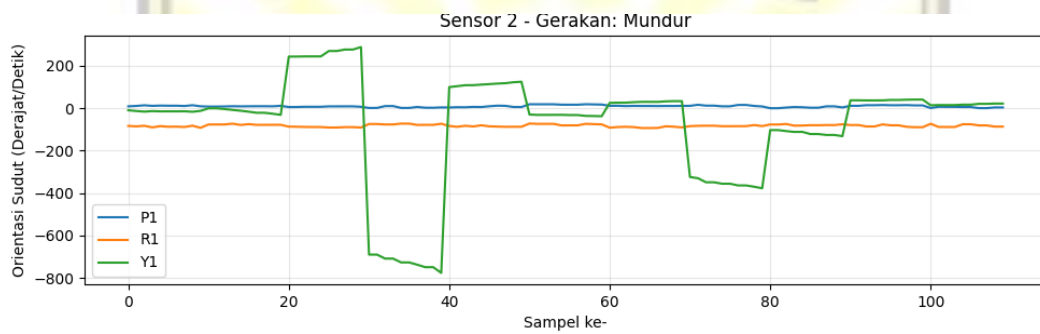
4.2.3 Grafik gerakan Ekstensi

Gerakan ekstensi yang dideteksi oleh sensor 1 memiliki nilai *pitch* pada rentang $-27,89$ hingga $-8,29^\circ$, nilai *roll* pada rentang $-104,20$ hingga $-81,04^\circ$, dan nilai *yaw* pada rentang $-600,19$ hingga $1459,05^\circ$. Gerakan ekstensi yang dideteksi oleh sensor 2 memiliki nilai *pitch* pada rentang $-0,32$ hingga $18,32^\circ$, nilai *roll* pada rentang $-93,74$ hingga $-72,49^\circ$, dan nilai *yaw* pada rentang $-775,43$ hingga $287,46^\circ$. Sementara itu, gerakan ekstensi yang dideteksi oleh sensor 3 memiliki nilai *pitch* pada rentang $-6,68$ hingga $16,44^\circ$, nilai *roll* pada rentang $-122,77$ hingga $5,69^\circ$, dan nilai *yaw* pada rentang $-350,54$ hingga $89,90^\circ$. Grafik data gerakan fleksi yang

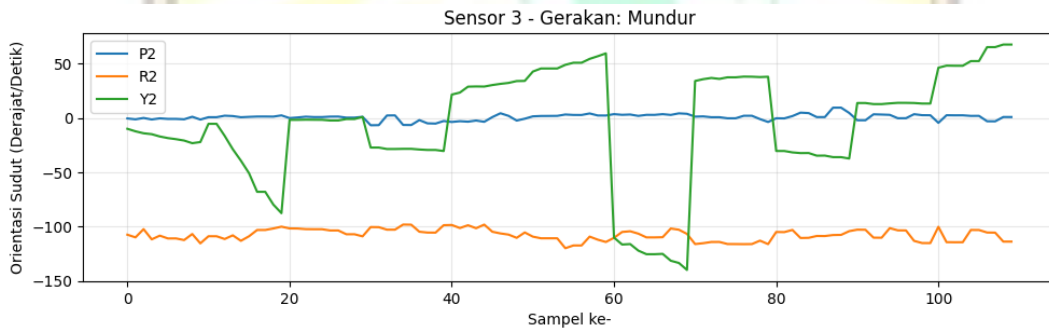
dideteksi oleh sensor 1, 2, dan 3 ditunjukkan secara berurutan pada Gambar 4.7, 4.8, dan 4.9 berikut.



Gambar 4. 7 Grafik Gerakan Sensor 1 Ekstensi



Gambar 4. 8 Grafik Gerakan Sensor 2 Ekstensi

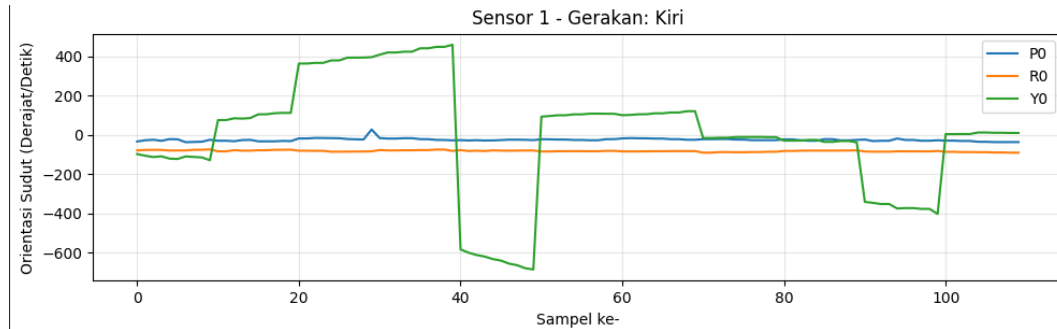


Gambar 4. 9 Grafik Gerakan Sensor 3 Ekstensi

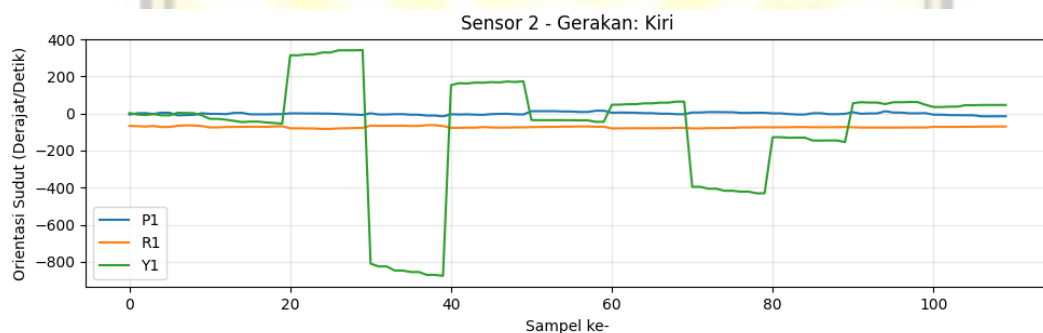
4.2.4 Grafik gerakan Lateral bending Kiri

Gerakan *lateral bending* kiri yang dideteksi oleh sensor 1 memiliki nilai *pitch* pada rentang $-38,40$ hingga $26,34^{\circ}$, nilai *roll* pada rentang $-91,77$ hingga $-69,94^{\circ}$, dan nilai *yaw* pada rentang $-686,52$ hingga $457,71^{\circ}$. Gerakan *lateral bending* kiri yang dideteksi oleh sensor 2 memiliki nilai *pitch* pada rentang $-15,21$ hingga $15,14^{\circ}$, nilai *roll* pada rentang $-83,05$ hingga $-53,97^{\circ}$, dan nilai *yaw* pada rentang $-876,26$ hingga $342,73^{\circ}$. Sementara itu, gerakan *lateral bending* kiri yang dideteksi oleh sensor 3 memiliki nilai *pitch* pada rentang $-18,56$ hingga $4,83^{\circ}$, nilai *roll* pada rentang $-114,23$ hingga $-44,34^{\circ}$, dan nilai *yaw* pada rentang $-180,16$

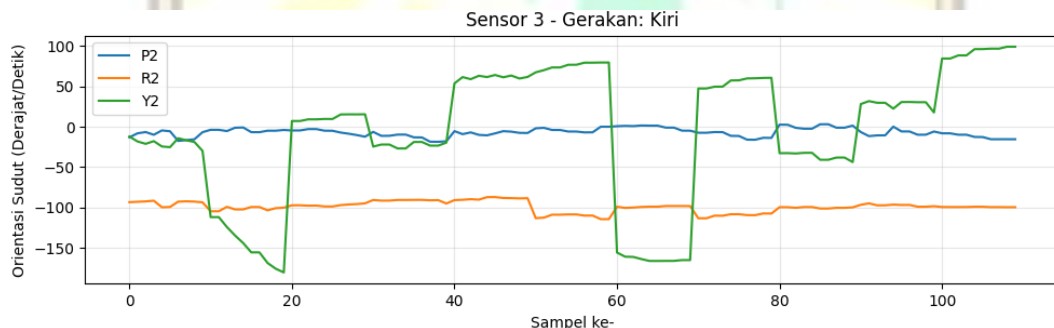
hingga $99,06^\circ$. Grafik data gerakan lateral bending kiri yang dideteksi oleh sensor 1, 2, dan 3 ditunjukkan secara berurutan pada Gambar 4.10, 4.11, dan 4.12 berikut.



Gambar 4. 10 Grafik Gerakan Sensor 1 Lateral bending Kiri



Gambar 4. 11 Grafik Gerakan Sensor 2 Lateral bending Kiri

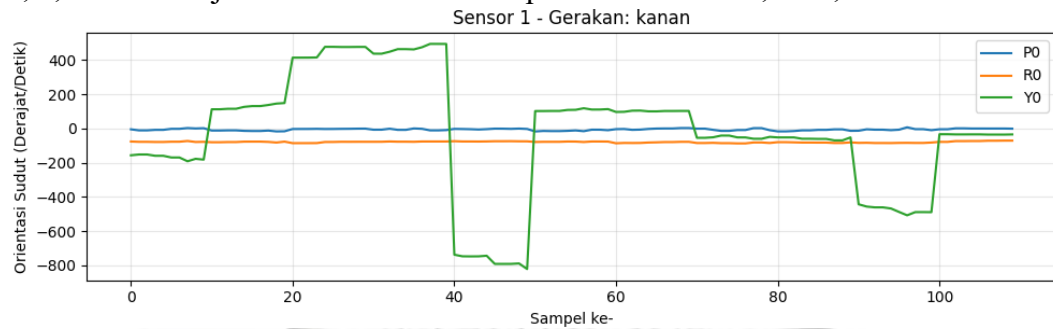


Gambar 4. 12 Grafik Gerakan Sensor 3 Lateral bending Kiri

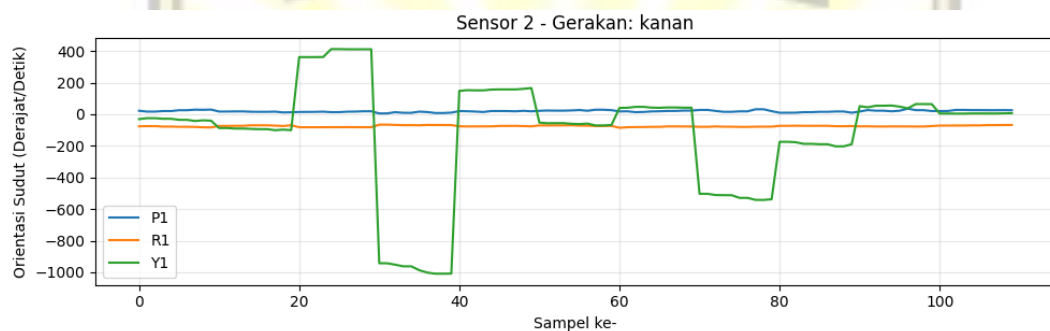
4.2.5 Grafik gerakan Lateral bending Kanan

Gerakan *lateral bending* kanan yang dideteksi oleh sensor 1 memiliki nilai *pitch* pada rentang $-17,27$ hingga $9,37^\circ$, nilai *roll* pada rentang $-87,17$ hingga $-70,82^\circ$, dan nilai *yaw* pada rentang $-821,01$ hingga $494,27^\circ$. Gerakan *lateral bending* kanan yang dideteksi oleh sensor 2 memiliki nilai *pitch* pada rentang $4,04$ hingga $35,19^\circ$, nilai *roll* pada rentang $-85,79$ hingga $-56,67^\circ$, dan nilai *yaw* pada rentang $-1009,40$ hingga $412,02^\circ$. Sementara itu, gerakan *lateral bending* kanan yang dideteksi oleh sensor 3 memiliki nilai *pitch* pada rentang $2,14$ hingga $21,65^\circ$, nilai *roll* pada rentang $-112,15$ hingga $-63,33^\circ$, dan nilai *yaw* pada rentang $-285,49$

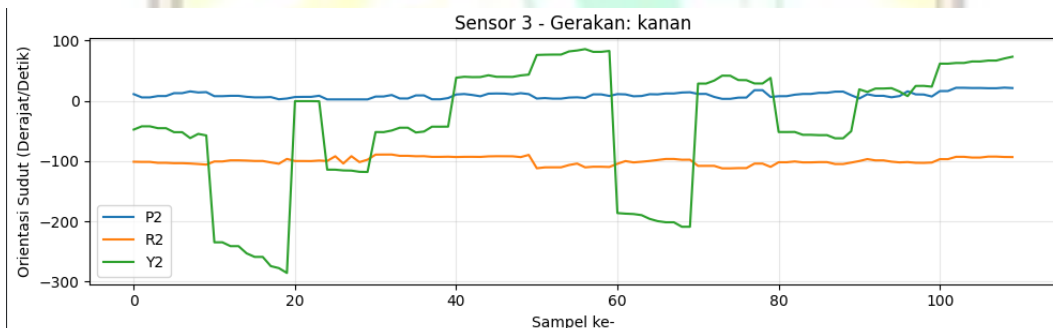
hingga $85,49^\circ$. Grafik data gerakan lateral bending kanan yang dideteksi oleh sensor 1, 2, dan 3 ditunjukkan secara berurutan pada Gambar 4.13, 4.14, dan 4.15 berikut.



Gambar 4. 13 Grafik Gerakan Sensor 1 Lateral bending Kanan



Gambar 4. 14 Grafik Gerakan Sensor 2 Lateral bending Kanan



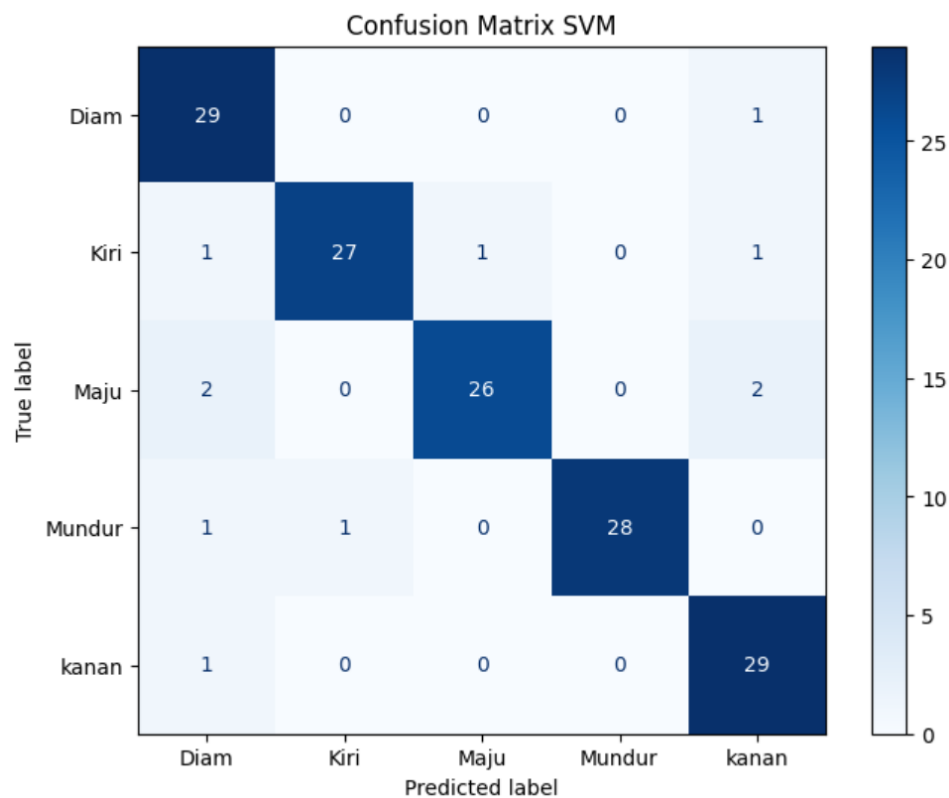
Gambar 4. 15 Grafik Gerakan Sensor 3 Lateral Bending Kanan

4.3 Model Klasifikasi Support Vector Machine

Pada model klasifikasi menggunakan SVM, tahapan implementasi dimulai dengan membaca data hasil akuisisi sensor pada Lampiran 2, kemudian membagi data menjadi data latih dan data uji. Data dibagi dengan komposisi data latih sebesar 80% dan data uji sebesar 20% menggunakan fungsi *train_test_split* dengan parameter *stratify* untuk menjaga proporsi setiap kelas tetap seimbang. Sebelum proses pelatihan dilakukan, data dinormalisasikan menggunakan metode *StandardScaler* agar setiap fitur memiliki skala yang seragam sehingga kinerja algoritma SVM dapat menjadi lebih optimal.

Model SVM yang digunakan menerapkan *Kernel Radial Basis Function* (RBF) dengan nilai parameter $C = 1$ dan $\gamma = scale$. Parameter C berfungsi untuk mengatur tingkat toleransi kesalahan klasifikasi, sedangkan parameter γ menentukan pengaruh masing-masing data terhadap pembentukan batas keputusan (*decision boundary*). Setelah proses pelatihan selesai, model digunakan untuk memprediksi data uji.

Model SVM kemudian dilatih kembali menggunakan dua komponen tersebut sehingga batas keputusan antarkelas dapat divisualisasikan dalam bidang dua dimensi. Gambar 4.16 menunjukkan hasil pengujian model klasifikasi SVM.



Gambar 4. 16 Confusion Matrix Model Klasifikasi SVM

Hasil evaluasi model klasifikasi SVM yang meliputi nilai akurasi, presisi, waktu pelatihan, dan waktu prediksi disajikan pada Tabel 4.1. Selanjutnya, analisis kinerja model untuk setiap kelas berdasarkan *confusion matrix* ditampilkan pada Tabel 4.2.

Tabel 4. 1 Hasil Evaluasi Kinerja Model SVM

Akurasi	Presisi	Waktu Pelatihan (detik)	Waktu Prediksi (detik)	Recall	F1-Score
0,9267	0,9318	0,0246	0,0107	0.9267	0.9271

Tabel 4. 2 Analisis Per Kelas Model SVM

Gerakan	Benar	Total	Persentase (%)
Diam	29	30	96,67%
Maju	27	30	90,00%
Mundur	26	30	86,67%
Kiri	28	30	96,67%
Kanan	29	30	92,67%

Berdasarkan hasil pengujian model klasifikasi *Support Vector Machine* (SVM), diperoleh nilai akurasi sebesar 92,67%, presisi sebesar 93,18%, recall sebesar 92,67%, dan *F1-Score* sebesar 92,71%. Nilai-nilai ini menunjukkan bahwa model memiliki kinerja yang sangat baik dalam mengklasifikasikan kelima kelas gerakan, yaitu Diam, Kiri, Maju, Mundur, dan Kanan, dengan tingkat kesalahan klasifikasi yang relatif kecil. Selain itu, model ini juga menunjukkan efisiensi komputasi yang sangat tinggi, ditandai dengan waktu pelatihan hanya 0,0246 detik dan waktu prediksi 0,0107 detik, sehingga model sangat cocok diterapkan pada sistem yang membutuhkan respons cepat atau *real-time*, seperti sistem kontrol gerak berbasis sinyal.

Jika ditelusuri lebih dalam melalui confusion matrix, dari total 150 sampel data uji yang terbagi rata menjadi 30 sampel untuk masing-masing kelas, model berhasil mengklasifikasikan 139 sampel dengan benar dan hanya melakukan kesalahan pada 11 sampel, atau setara dengan tingkat kesalahan sebesar 7,33%. Perhitungan metrik per kelas menunjukkan bahwa kelas Diam dan Kanan memiliki performa terbaik dengan tingkat keberhasilan masing-masing sebesar 96,67% (29 dari 30 sampel berhasil diklasifikasikan dengan benar), diikuti oleh kelas Kiri sebesar 93,33% (28 dari 30 sampel), kelas Maju sebesar 90,00% (27 dari 30 sampel), sedangkan kelas Mundur menunjukkan performa terendah dengan tingkat keberhasilan 86,67% (26 dari 30 sampel).

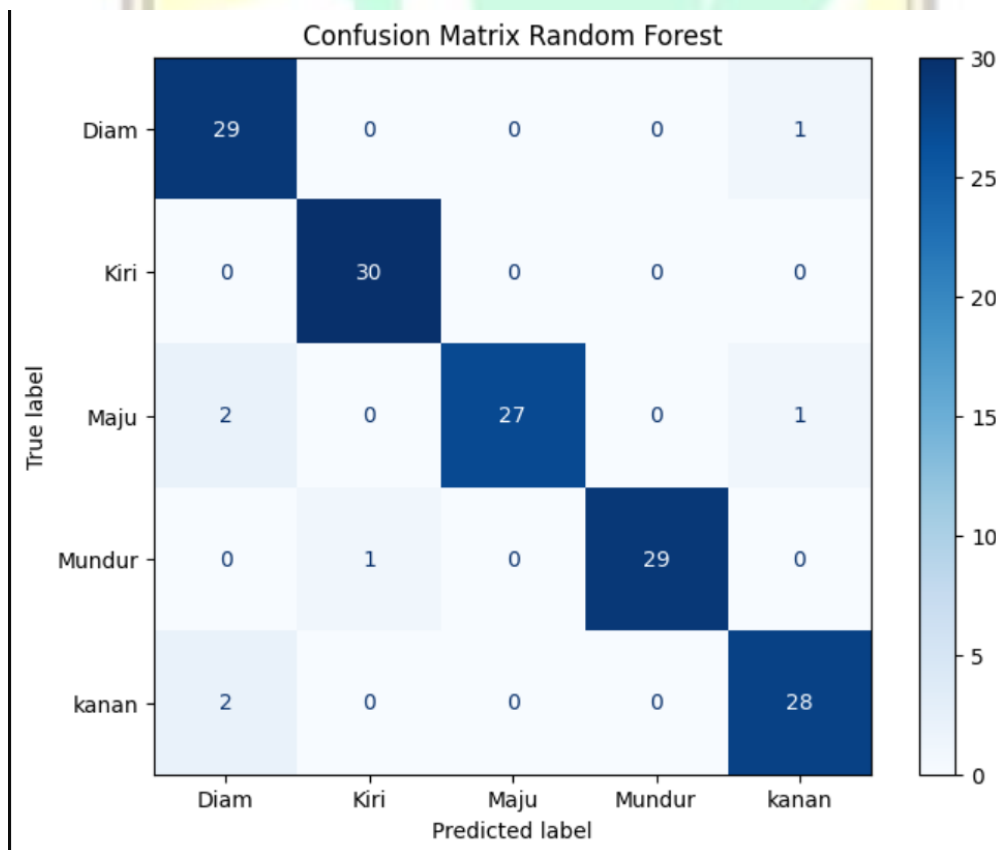
Pola kesalahan klasifikasi yang teramati pada confusion matrix mengindikasikan adanya kebingungan yang cukup signifikan antara kelas Diam dengan kelas Maju dan Mundur. Sebanyak 3 sampel yang sebenarnya berlabel Maju justru diprediksi sebagai Diam, dan 2 sampel yang sebenarnya berlabel Mundur juga diprediksi sebagai Diam, sehingga kelas Diam menjadi kelas yang paling banyak "menyerap" kesalahan prediksi dari kelas lain meskipun *recall*-nya sendiri masih tergolong baik (96,67%). Fenomena ini kemungkinan besar disebabkan oleh kemiripan karakteristik sinyal antara kondisi diam dengan fase awal gerakan maju atau mundur, di mana perbedaan pola fitur antara ketiga kelas tersebut belum cukup tegas untuk dapat dipisahkan secara sempurna oleh model. Berbeda halnya dengan kelas Kiri dan Kanan yang menunjukkan pola sinyal yang jauh lebih khas dan terpisah dari kelas lainnya, sehingga hampir tidak pernah tertukar dalam proses klasifikasi.

Secara keseluruhan, dapat disimpulkan bahwa model SVM yang dibangun sudah menunjukkan kinerja yang sangat baik dan layak digunakan, baik dari segi akurasi maupun efisiensi waktu komputasi.

4.4 Model Klasifikasi *Random Forest*

Model *Random Forest* yang digunakan dibangun dengan 100 pohon keputusan (*decision tree*) menggunakan parameter $n_estimators = 100$ dan $random_state = 42$. Parameter $n_estimators$ menentukan jumlah pohon yang dibentuk selama proses pelatihan, sedangkan $random_state$ digunakan untuk memastikan hasil eksperimen dapat direproduksi. Pada *Random Forest*, setiap pohon dilatih menggunakan sampel data yang dipilih secara acak melalui metode *bootstrap sampling*, serta menggunakan subset fitur yang berbeda. Prediksi akhir diperoleh melalui mekanisme *majority voting*, yaitu kelas yang paling banyak dipilih oleh seluruh pohon akan menjadi hasil klasifikasi.

Setelah proses pelatihan selesai dilakukan, model digunakan untuk memprediksi data uji. Model pohon *random forest* diujikan dengan data latih dan data uji. Hasil pengujian dapat dilihat pada Tabel 4.3 dan Gambar 4.17, sedangkan Tabel 4.4 menunjukkan analisis per kelas hasil model klasifikasi *random forest*.



Gambar 4. 17 *Confusion Matrix Model Klasifikasi Random Forest*

Tabel 4. 3 Hasil Evaluasi Kinerja Model Random Forest

Akurasi	Presisi	Waktu Pelatihan (detik)	Waktu Prediksi (detik)	Recall	F1-Score
0,9533	0,956	0.296403	0.010444	0.9533	0.9536

Tabel 4. 4 Analisis Per kelas Model Random Forest

Gerakan	Benar	Total	Persentase (%)
Diam	29	30	96,67%
Maju	30	30	100%
Mundur	27	30	90%
Kiri	29	30	96,67%
Kanan	28	30	93,33%

Berdasarkan hasil pengujian model klasifikasi *Random Forest*, diperoleh nilai akurasi sebesar 95,33%, presisi sebesar 95,60%, *recall* sebesar 95,33%, dan *F1-Score* sebesar 95,36%. Nilai-nilai ini menunjukkan bahwa model *Random Forest* memiliki kinerja yang sangat tinggi dan konsisten di seluruh metrik evaluasi, bahkan lebih unggul dibandingkan model SVM yang memperoleh akurasi 92,67%. Namun demikian, dari sisi efisiensi komputasi, *Random Forest* membutuhkan waktu pelatihan sebesar 0,296403 detik dan waktu prediksi sebesar 0,010444 detik, di mana waktu pelatihannya lebih lambat dibandingkan SVM yang hanya 0,0246 detik, sedangkan waktu prediksinya sedikit lebih cepat dibandingkan SVM (0,010444 detik berbanding 0,0107 detik).

Jika ditelusuri melalui confusion matrix, dari total 150 sampel data uji dengan 30 sampel untuk masing-masing kelas, model *Random Forest* berhasil mengklasifikasikan 143 sampel dengan benar dan hanya melakukan kesalahan pada 7 sampel saja, atau setara dengan tingkat kesalahan sebesar 4,67%. Angka ini lebih kecil dibandingkan tingkat kesalahan SVM yang mencapai 7,33% (11 sampel salah), menunjukkan bahwa *Random Forest* memiliki kemampuan generalisasi yang lebih baik pada data ini. Perhitungan metrik per kelas menunjukkan bahwa kelas Maju memiliki tingkat keberhasilan sempurna (100%), diikuti oleh kelas Diam dan Kiri yang masing-masing memiliki tingkat keberhasilan 96,67%, kelas Kanan sebesar 93,33%, dan kelas Mundur sebesar 90,00%.

Pola kesalahan yang teramati pada confusion matrix menunjukkan bahwa kesalahan klasifikasi masih terjadi, namun jumlahnya relatif sedikit. Terdapat 1 sampel Diam yang diprediksi sebagai Kanan, 2 sampel Maju yang diprediksi sebagai Diam, 1 sampel Maju yang diprediksi sebagai Kanan, 1 sampel Mundur yang diprediksi sebagai Kiri, serta 2 sampel Kanan yang diprediksi sebagai Diam. Pola ini menunjukkan bahwa model masih mengalami sedikit kesulitan dalam membedakan beberapa gerakan yang memiliki karakteristik sinyal serupa, terutama antara Diam dengan Maju dan Kanan. Meskipun demikian, sebagian besar sampel

pada setiap kelas tetap berhasil dikenali dengan benar, sehingga performa keseluruhan model tetap tergolong sangat baik.

Secara keseluruhan, dapat disimpulkan bahwa model *Random Forest* menunjukkan performa klasifikasi yang lebih unggul dibandingkan dengan SVM pada seluruh metrik evaluasi (akurasi, presisi, *recall*, dan *F1-Score*), dengan tingkat kesalahan yang lebih rendah dan pola kesalahan yang relatif sedikit. Namun demikian, keunggulan tersebut diperoleh dengan konsekuensi waktu pelatihan yang lebih lama dibandingkan SVM. Oleh karena itu, apabila prioritas utama adalah memperoleh akurasi klasifikasi yang tinggi, *Random Forest* merupakan pilihan yang lebih baik. Sebaliknya, apabila efisiensi waktu pelatihan menjadi pertimbangan utama, SVM masih dapat menjadi alternatif yang layak karena memiliki waktu pelatihan yang jauh lebih singkat dengan performa klasifikasi yang tetap baik.

4.5 Pengujian Performa Kursi Roda

Pengujian performa kursi roda dilakukan untuk mengevaluasi kemampuan sistem dalam mengenali gerakan tubuh, khususnya saat terjadi transisi antarggerakan. Pengujian melibatkan 15 responden yang masing-masing melakukan tiga kali percobaan di Departemen Teknik Elektro. Rincian hasil pengujian performa kursi roda disajikan pada Tabel 4.5

Tabel 4. 5 Hasil Pengujian Performa Kursi Roda *Random Forest*

No	Responden	Gerakan	Percobaan dan Waktu Pemrosesan (ms)						Jumlah
			1	2	3	4	5	6	
1	A	Diam	✓	32.591	X	0	X	0	1
		Maju	✓	32.784	✓	32.624	✓	32.064	3
		Mundur	✓	37.703	✓	33.628	✓	32.254	3
		Kiri	✓	31.466	✓	33.602	✓	33.202	3
		Kanan	✓	31.726	✓	34.249	✓	33.230	3
2	B	Diam	X	0	X	0	✓	31.817	1
		Maju	✓	31.525	✓	32.510	✓	29.655	3
		Mundur	✓	34.811	✓	32.924	✓	35.300	3
		Kiri	✓	35.503	✓	38.075	✓	43.641	3
		Kanan	✓	33.195	✓	34.860	✓	33.447	3
3	C	Diam	✓	32.233	✓	32.852	✓	45.121	3
		Maju	✓	31.071	✓	31.579	✓	31.483	3
		Mundur	✓	33.823	✓	33.146	✓	34.696	3
		Kiri	✓	32.428	✓	37.259	✓	32.652	3
		Kanan	✓	32.958	✓	36.590	✓	30.329	3
4	D	Diam	✓	33.828	✓	42.918	✓	46.192	3
		Maju	✓	33.872	✓	36.665	✓	33.730	3

		Mundur	✓	31.939	✓	29.780	✓	32.592	3
		Kiri	✓	32.111	✓	35.480	✓	31.513	3
		Kanan	✓	34.982	X	0	✓	32.560	3
		Diam	✓	32.651	✓	35.705	✓	34.500	3
		Maju	✓	34.672	✓	32.966	✓	32.473	3
5	E	Mundur	X	0	✓	32.267	X	0	1
		Kiri	✓	32.898	✓	33.200	✓	31.505	3
		Kanan	✓	31.248	✓	50.096	✓	32.779	3
		Diam	✓	31.147	✓	33.043	✓	57.230	3
		Maju	✓	32.606	✓	32.981	✓	34.121	3
6	F	Mundur	✓	32.908	✓	33.168	✓	30.755	3
		Kiri	✓	34.499	✓	35.749	✓	31.512	3
		Kanan	✓	32.033	✓	31.241	✓	33.909	3
		Diam	✓	43.322	✓	34.344	✓	34.135	3
		Maju	✓	34.708	✓	31.790	✓	32.163	3
7	G	Mundur	✓	35.177	✓	31.924	✓	31.669	3
		Kiri	✓	35.773	✓	32.650	✓	31.659	3
		Kanan	✓	35.255	✓	33.458	✓	30.599	3
		Diam	✓	34.189	✓	32.092	✓	34.274	3
		Maju	✓	32.310	✓	31.508	✓	33.508	3
8	H	Mundur	✓	33.576	✓	32.364	✓	32.308	3
		Kiri	✓	34.759	✓	34.177	✓	33.301	3
		Kanan	✓	32.261	X	0	✓	30.162	2
		Diam	✓	30.257	✓	31.932	✓	32.701	3
		Maju	✓	33.115	✓	32.961	✓	32.296	3
9	I	Mundur	✓	34.437	✓	32.853	✓	30.248	3
		Kiri	✓	39.977	✓	30.879	✓	33.681	3
		Kanan	✓	34.857	✓	34.455	✓	32.884	3
		Diam	✓	32.591	✓	35.879	✓	32.530	3
		Maju	✓	32.784	✓	32.624	✓	32.064	3
10	J	Mundur	✓	37.703	✓	33.628	✓	32.254	3
		Kiri	✓	31.466	✓	33.602	✓	33.202	3
		Kanan	✓	31.726	✓	34.249	✓	33.230	3
		Diam	✓	34.814	✓	32.528	✓	31.817	3
		Maju	✓	31.525	✓	32.510	✓	29.655	3
11	K	Mundur	✓	34.811	✓	32.924	✓	35.300	3
		Kiri	✓	35.503	✓	38.075	✓	43.641	3
		Kanan	✓	33.195	✓	34.860	✓	33.447	3
		Diam	✓	32.233	✓	31.852	✓	35.121	3
		Maju	X	0	X	0	✓	31.483	1
12	L	Mundur	✓	33.823	✓	33.146	✓	34.696	3
		Kiri	✓	32.428	✓	37.259	✓	32.652	3

13	M	Kanan	✓	32.958	✓	36.590	✓	30.329	3	
		Diam	✓	33.828	X	0	✓	36.192	2	
		Maju	✓	33.872	✓	36.665	✓	33.730	3	
		Mundur	✓	31.939	✓	29.780	✓	32.592	3	
		Kiri	✓	32.111	✓	35.480	✓	31.513	3	
14	N	Kanan	✓	34.982	✓	32.735	✓	32.560	3	
		Diam	X	0	✓	32.651	✓	34.500	2	
		Maju	✓	34.672	✓	32.966	✓	32.473	3	
		Mundur	✓	33.152	✓	32.267	✓	32.886	3	
		Kiri	✓	32.898	✓	33.200	✓	31.505	3	
15	O	Kanan	✓	31.248	✓	30.096	✓	32.779	3	
		Diam	✓	31.147	✓	33.043	✓	37.230	3	
		Maju	✓	32.606	✓	32.981	✓	34.121	3	
		Mundur	✓	32.908	✓	33.168	✓	30.755	3	
		Kiri	✓	34.499	✓	35.749	✓	31.512	3	
		Kanan	✓	32.033	✓	31.241	✓	33.909	3	
Rata-Rata Waktu Pemrosesan					31.715		31.177		32.793	31,895
Total										215
Akurasi										95,55%

Tabel 4. 6 Hasil Pengujian Performa Kursi Roda SVM

No	Responden	Gerakan	Percobaan dan Waktu Pemrosesan (ms)						Jumlah
			1	2	3	4	5	6	
1	A	Diam	✓	2.412	✓	2.145	✓	2.592	3
		Maju	✓	2.663	✓	2.131	✓	2.979	3
		Mundur	✓	2.080	✓	2.149	✓	2.029	3
		Kiri	✓	1.861	✓	2.287	✓	2.226	3
		Kanan	✓	2.189	✓	2.536	✓	2.064	3
2	B	Diam	✓	2.502	✓	2.101	✓	2.158	3
		Maju	✓	2.367	✓	2.104	✓	2.247	3
		Mundur	✓	2.079	✓	2.129	✓	2.460	3
		Kiri	✓	1.973	✓	2.025	✓	3.544	3
		Kanan	✓	2.144	✓	3.262	✓	3.876	3
3	C	Diam	✓	1.841	✓	2.352	✓	1.961	3
		Maju	✓	2.357	✓	2.115	✓	2.395	3
		Mundur	X	0	X	0	✓	3.171	1
		Kiri	✓	2.539	✓	2.521	✓	2.021	3
		Kanan	✓	2.112	✓	2.892	✓	2.015	3
4	D	Diam	✓	1.979	✓	2.033	✓	1.867	3
		Maju	✓	2.543	✓	3.280	✓	2.409	3

5	E	Mundur	✓	2.445	✓	2.160	✓	2.279	3
		Kiri	✓	1.966	✓	2.005	✓	2.497	3
		Kanan	✓	3.299	✓	2.926	✓	2.058	3
		Diam	✓	3.011	✓	2.058	✓	2.181	3
		Maju	✓	2.577	✓	2.357	✓	2.103	3
		Mundur	✓	2.053	✓	2.178	✓	2.274	3
		Kiri	✓	1.973	✓	1.978	✓	1.943	3
		Kanan	✓	2.485	✓	2.443	✓	1.911	3
		Diam	✓	2.664	✓	2.031	✓	2.531	3
		Maju	X	0	X	0	✓	2.067	1
6	F	Mundur	✓	2.396	✓	2.103	✓	2.144	3
		Kiri	✓	3.146	✓	2.196	✓	2.517	3
		Kanan	✓	2.451	✓	2.599	✓	2.189	3
		Diam	✓	2.096	✓	1.987	✓	1.967	3
		Maju	✓	2.103	✓	2.183	✓	2.101	3
7	G	Mundur	✓	2.998	✓	2.107	✓	2.671	3
		Kiri	✓	2.584	✓	2.667	✓	2.452	3
		Kanan	✓	2.157	✓	2.394	✓	2.056	3
		Diam	✓	1.970	✓	2.012	✓	1.879	3
		Maju	✓	2.131	✓	2.467	✓	2.695	3
8	H	Mundur	✓	2.327	✓	2.415	✓	2.141	3
		Kiri	✓	2.185	✓	2.215	✓	2.386	3
		Kanan	✓	1.903	✓	2.859	✓	2.371	3
		Diam	✓	2.751	✓	3.396	✓	1.830	3
		Maju	✓	3.961	✓	2.471	✓	3.339	3
9	I	Mundur	✓	2.088	✓	2.709	✓	2.412	3
		Kiri	✓	1.983	✓	1.945	✓	3.488	3
		Kanan	✓	2.555	✓	2.750	✓	2.771	3
		Diam	✓	2.412	✓	2.145	✓	2.592	3
		Maju	✓	2.663	✓	2.131	✓	2.979	3
10	J	Mundur	✓	2.080	✓	2.149	✓	2.029	3
		Kiri	✓	1.861	✓	2.287	✓	2.226	3
		Kanan	✓	2.189	✓	2.536	✓	2.064	3
		Diam	✓	2.502	✓	2.101	✓	2.158	3
		Maju	✓	2.367	✓	2.104	✓	2.247	3
11	K	Mundur	✓	2.079	✓	2.129	✓	2.460	3
		Kiri	X	0	✓	2.025	✓	3.544	2
		Kanan	✓	2.144	✓	3.262	✓	3.876	3
		Diam	✓	1.841	✓	2.352	✓	1.961	3
		Maju	✓	2.357	✓	2.115	✓	2.395	3
12	L	Mundur	✓	2.396	✓	2.156	✓	3.171	3
		Kiri	✓	2.539	✓	2.521	✓	2.021	3

13	M	Kanan	✓	2.112	✓	2.892	✓	2.015	3
		Diam	✓	1.979	✓	2.033	✓	1.867	3
		Maju	✓	2.543	✓	3.280	✓	2.409	3
		Mundur	✓	2.445	✓	2.160	✓	2.279	3
		Kiri	✓	1.966	✓	2.005	✓	2.497	3
14	N	Kanan	✓	3.299	✓	2.926	✓	2.058	3
		Diam	✓	3.011	✓	2.058	✓	2.181	3
		Maju	✓	2.577	✓	2.357	✓	2.103	3
		Mundur	✓	2.053	✓	2.178	✓	2.274	3
		Kiri	✓	1.973	✓	1.978	✓	1.943	3
15	O	Kanan	✓	2.485	✓	2.443	✓	1.911	3
		Diam	✓	2.664	✓	2.031	✓	2.531	3
		Maju	✓	2.475	✓	2.677	✓	2.067	3
		Mundur	✓	2.396	✓	2.103	✓	2.144	3
		Kiri	✓	3.146	✓	2.196	✓	2.517	3
		Kanan	✓	2.451	✓	2.599	✓	2.189	3
Rata-Rata Waktu pemrosesan				2.278		2,288		2.373	2,312
Total Gerakan									220
Akurasi									97,78%

1. Analisis Hasil Pengujian Integrasi SVM dan *Random Forest*

Berdasarkan hasil pengujian secara langsung pada kursi roda, sistem yang menggunakan algoritma *Random Forest* memperoleh tingkat akurasi sebesar 95,55%, sedangkan sistem yang menggunakan algoritma SVM memperoleh tingkat akurasi sebesar 97,78% dalam mengenali gerakan tubuh.

Algoritma *Support Vector Machine* (SVM) menunjukkan kinerja yang lebih efisien dalam menerjemahkan gestur tubuh menjadi perintah pergerakan motor. Hal ini ditunjukkan oleh rata-rata waktu pemrosesan sebesar 2,312 milidetik (ms), yang lebih cepat dibandingkan algoritma *Random Forest* dengan rata-rata waktu pemrosesan sebesar 31,895 ms. Perbedaan waktu pemrosesan tersebut menunjukkan bahwa algoritma SVM memiliki latensi yang lebih rendah, sehingga lebih sesuai untuk diterapkan pada sistem kendali kursi roda secara *real-time*. Pada tahap integrasi, waktu yang diukur adalah latensi satu siklus kendali secara utuh mulai dari akuisisi data sensor MPU-6050, transmisi data melalui ESP32, proses klasifikasi, hingga eksekusi sinyal PWM ke motor kursi roda, sehingga menghasilkan rata-rata waktu pemrosesan yang lebih besar dibandingkan dengan waktu pemrosesan pada tahap klasifikasi. Waktu pemrosesan yang lebih singkat memungkinkan sistem merespons gerakan pengguna dengan lebih cepat, sehingga dapat meningkatkan responsivitas, kenyamanan, dan aspek keselamatan selama proses pengoperasian kursi roda.

Pada pengklasifikasian data, algoritma *Random Forest* menghasilkan akurasi yang lebih tinggi (95,33%) dibandingkan dengan SVM (92,67%). Hal ini

menunjukkan bahwa hasil pengklasifikasian tidak sepenuhnya merepresentasikan performa kedua model pada kondisi penggunaan nyata.

Perbedaan performa ini erat kaitannya dengan kemampuan generalisasi masing-masing algoritma terhadap data yang berada di luar rentang sebaran data latih (*out-of-distribution*/OOD), yang lazim terjadi pada kondisi operasional nyata akibat variasi postur, kestabilan tubuh, dan *noise* sensor. *Random Forest* bekerja dengan membagi ruang fitur menjadi beberapa wilayah keputusan berdasarkan batas (*threshold*) yang dipelajari dari data latih. Karena struktur pohon keputusannya bersifat partisi tetap, setiap input akan selalu diarahkan ke satu *leaf* node tertentu, tanpa mekanisme untuk merepresentasikan ketidakpastian ketika nilai input berada jauh di luar rentang data latih. Akibatnya, ketika menghadapi data OOD, *Random Forest* tetap menghasilkan keputusan klasifikasi. Karena *leaf* node yang dipilih dibentuk berdasarkan pola data latih yang tidak merepresentasikan kondisi ekstrem tersebut. Karakteristik inilah yang menjadikan *Random Forest* lebih rentan terhadap *overfitting* terhadap karakteristik spesifik data latih, sehingga performanya menurun pada kondisi *real-time*.

Sebaliknya, SVM dengan *kernel Radial Basis Function* (RBF) bekerja berdasarkan prinsip maksimalisasi margin antarkelas, di mana keputusan klasifikasi didasarkan pada kemiripan (*similarity*) input terhadap support vector melalui fungsi eksponensial yang meluruh terhadap jarak. Ketika input berada jauh dari seluruh *support vector* (kondisi OOD), nilai fungsi keputusan pada seluruh kelas menjadi kecil dan saling berdekatan (sulit dibedakan). Namun demikian, karena *decision boundary* SVM dibentuk untuk memaksimalkan jarak pemisah antarkelas, posisi relatif input terhadap batas-batas tersebut tetap mempertahankan informasi arah klasifikasi yang paling mendekati kebenaran, meskipun tingkat keyakinannya rendah.

Perbedaan mekanisme inilah yang menjelaskan mengapa SVM menunjukkan tingkat akurasi klasifikasi gerakan yang lebih baik dibandingkan *Random Forest* pada tahap integrasi langsung ke perangkat keras kursi roda, meskipun hasil klasifikasi menunjukkan hal yang sebaliknya.

2. Perbandingan dengan Penelitian Sebelumnya

Hasil pengujian pada penelitian ini menunjukkan tingkat akurasi yang lebih tinggi dibandingkan dengan beberapa penelitian sebelumnya [8][9][10][11]. Salah satu penelitian yang dapat dijadikan sebagai pembanding adalah penelitian [10] yang memperoleh akurasi pengujian sebesar 93,33%. Nilai tersebut lebih rendah dibandingkan dengan hasil penelitian ini, yaitu sebesar 95,55% pada metode *Random Forest* dan 97,78% pada metode SVM.

Akurasi SVM dan *Random Forest* dapat lebih tinggi dibandingkan *Decision Tree* pada implementasi *real-time* karena *Decision Tree* hanya mengandalkan satu struktur pohon dengan batas keputusan kaku yang sangat sensitif terhadap *noise* dan variasi gerakan, sehingga rawan *overfitting* dan sulit menggeneralisasi saat

menghadapi data di luar pola latih pada pengujian *real-time* sebaliknya, *Random Forest* mengurangi kelemahan ini dengan menggabungkan banyak pohon keputusan melalui bagging sehingga hasil klasifikasinya lebih stabil, sementara SVM membentuk *decision boundary* berdasarkan margin maksimum antarkelas di ruang berdimensi tinggi (kernel RBF) yang mampu memisahkan pola gerakan kompleks dan tetap mempertahankan arah klasifikasi yang mendekati benar meski menghadapi variasi data baru.

Hasil penelitian ini juga menunjukkan akurasi yang lebih tinggi dibandingkan dengan penelitian [11] yang menggunakan metode KNN dan memperoleh akurasi integrasi sebesar 85,33%. Jika dibandingkan dengan hasil tersebut, akurasi integrasi pada penelitian ini mencapai 97,78% menggunakan SVM dan 95,55% menggunakan *Random Forest*, akan tetapi memiliki akurasi klasifikasi yang lebih rendah dengan KNN sebesar 99,9%.

Perbedaan performa antara KNN, SVM, dan *Random Forest* pada tahap integrasi *real-time* terjadi karena karakteristik dasar masing-masing algoritma dalam menggeneralisasi data. KNN mengklasifikasi data berdasarkan jarak *Euclidean* ke titik-titik data latih terdekat tanpa membentuk model keputusan yang general, sehingga sangat rentan terhadap variasi gerakan antarindividu, *noise* sensor akibat pergerakan kabel, dan *delay* pembacaan yang disebutkan dalam laporan KNN sendiri. Akibatnya, meski akurasi pada klasifikasi data sangat tinggi (99,9%/99,8%), performanya justru turun tajam menjadi 85,33% saat diuji langsung secara *real-time*. Nilai akurasi klasifikasi yang sangat tinggi tersebut dapat mengindikasikan bahwa model KNN cenderung mengalami *overfitting*, yaitu model terlalu menyesuaikan diri terhadap pola data dari subjek yang digunakan dalam proses pelatihan. Kondisi tersebut menyebabkan model kurang mampu mempelajari karakteristik pola gerakan yang dapat digeneralisasikan pada subjek lain. Akibatnya, ketika model digunakan untuk mengklasifikasikan pola gerakan dari subjek dengan karakteristik gerakan yang berbeda, kemungkinan terjadinya kesalahan klasifikasi menjadi lebih tinggi.