

Deteksi Gerakan Tangan menggunakan *Support Vector Machine* pada *Dumbbell* Berbasis *Raspberry Pi Zero*

ERWANI MERRY SARTIKA¹, AAN DARMAWAN¹, WILLIAM EKA JAYA¹,
ELIZABETH WIAN TO²

¹Program Studi Teknik Elektro, Universitas Kristen Maranatha, Indonesia

²Program Studi Desain Komunikasi Visual, Universitas Kristen Maranatha, Indonesia

Email: erwanimerry@gmail.com; erwani.ms@maranatha.ac.id

Received 6 Agustus 2021 | *Revised* 23 Agustus 2021 | *Accepted* 7 September 2021

ABSTRAK

Olahraga yang dilakukan di mana saja, dapat menggunakan alat sederhana seperti dumbbell. Latihan yang dilakukan harus sesuai usia dan kondisi kesehatan secara umum. Sebuah sistem diperlukan untuk mendeteksi gerakan sehingga membantu pemakai dalam menggunakan dumbbell. Dumbbell dilengkapi oleh sensor Inertial Measurement Unit (IMU), single board computer Raspberry Pi Zero W, dan LED RGB. Gerakan akan dideteksi oleh sensor Inertial Measurement Unit (IMU) yang dikirim ke Raspberry Pi Zero W untuk dilakukan preprocessing data. Algoritma Support Vector Machine (SVM) digunakan untuk mendapatkan model pendeteksi gerakan olahraga pada dumbbell. Bila gerakan terdeteksi maka Raspberry Pi Zero W akan memberi perintah LED RGB untuk menyalakan warna tertentu sesuai dengan deteksi gerakan. Berdasarkan hasil uji coba yang dilakukan kepada 5 orang dengan gerakan yang dideteksi sebanyak 6 gerakan, dumbbell pendeteksi gerakan memiliki tingkat keberhasilan sebesar 90%-94%.

Kata kunci: *dumbbell, deteksi gerakan, Raspberry Pi Zero W, SVM*

ABSTRACT

Exercises that can be done anywhere, can use simple tools such as dumbbells. Exercises should be appropriate for age and general health conditions. A system is needed to detect motion so as to assist the wearer in using the dumbbells. The dumbbell is equipped with an Inertial Measurement Unit (IMU) sensor, a single board computer Raspberry Pi Zero W, and RGB LEDs. Movement will be detected by the Inertial Measurement Unit (IMU) sensor which is sent to the Raspberry Pi Zero W for preprocessing the data. The Support Vector Machine (SVM) algorithm is used to obtain a sports motion detection model on dumbbells. If motion is detected, the Raspberry Pi Zero W will give an RGB LED command to turn on certain colors according to motion detection. Based on the results of trials conducted on 5 people with 6 movements detected, motion detection dumbbells have a success rate of 90%-94%.

Keywords: *dumbbell, motion detection, Raspberry Pi Zero W, SVM*

1. PENDAHULUAN

Olahraga perlu menjadi bagian dari kehidupan setiap orang. Kondisi Pandemi Covid-19 menyebabkan kegiatan olahraga hanya dilakukan di rumah dan tanpa bantuan instruktur. Namun latihan yang dilakukan tetap harus sesuai dengan usia dan kondisi kesehatan secara umum. Jika dirancang dengan tepat, program latihan untuk lansia dapat memberikan manfaat dan meningkatkan kualitas hidup secara keseluruhan. Selain itu program latihan juga dapat meningkatkan kekuatan tubuh, meningkatkan kesehatan, keseimbangan, stabilitas sendi, peningkatan kebugaran metabolik (termasuk toleransi glukosa dan kontrol kolesterol), mengontrol berat badan, dan pemeliharaan kepadatan tulang **(Fisher, dkk, 2017)**.

Menjaga kesehatan dengan berolahraga yang dapat dilakukan di mana saja, salah satunya adalah dengan menggunakan alat sederhana seperti *dumbbell*. Latihan yang dilakukan harus sesuai dengan usia dan kondisi kesehatan secara umum. Jika dirancang dengan tepat, program latihan beban untuk lansia dapat meningkatkan kekuatan tubuh, keseimbangan dan stabilitas sendi **(Nasrulloh, dkk, 2018)**.

Maka diperlukan sistem untuk mendeteksi gerakan sehingga dapat membantu pemakai dalam menggunakan *dumbbell*. Melalui sistem tersebut, olahraga beban menggunakan *dumbbell* dapat dilakukan dengan benar sesuai gerakan yang dipilih. Salah satu metode untuk mendeteksi gerakan adalah dengan aplikasi algoritma pengklasifikasi *Support Vector Machine* (SVM). SVM adalah sebuah metode pembelajaran mesin yang termasuk dalam kategori *supervised learning* **(Somvanshi, dkk, 2017)**. Salah satu keunggulan SVM adalah penggunaan *kernel trick* yang dapat mengatur parameter regularisasi dan *error* klasifikasi secara eksplisit **(Kumar, dkk, 2020)**. Pada kasus *gait recognition*, algoritma *Support Vector Machine* dan *Back Propagation* menghasilkan tingkat akurasi klasifikasi yang tinggi **(Wang, dkk, 2019)**. Algoritma lain yang telah digunakan adalah algoritma *Random Forrest* telah berhasil untuk mendeteksi lima gerakan dengan kinerja sebesar 78,2% **(Velloso, dkk, 2013)**. Maka pada penelitian ini algoritma SVM akan digunakan untuk meningkatkan kinerja deteksi gerakan dari hasil penelitian sebelumnya.

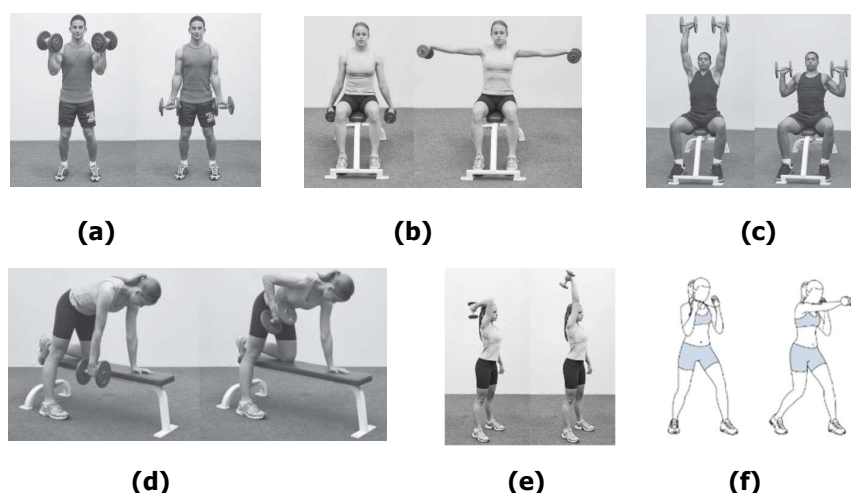
Proses aplikasi algoritma SVM dilakukan pada sebuah *Single Board Computer Raspberry Pi Zero W* **(Sukvichai, dkk, 2018)**. Alasan penggunaan *Raspberry Pi Zero W* karena ketersediaan *tools* yang umum digunakan untuk aplikasi *machine learning* seperti *python* dan *scikit-learn*, selain itu ukuran yang kecil memungkinkan untuk dapat diletakkan pada sebuah *dumbbell*. Sensor yang digunakan sebagai *input* data adalah sensor *Inertial Measurement Unit* (IMU) yang dapat membaca nilai *accelerometer*, *gyroscope*, dan *magnetometer* **(Wang, dkk, 2018)**. *Dumbbell* yang di dalamnya telah terpasang *Raspberry Pi Zero W* dan berisi algoritma SVM akan dilatih dan diuji menggunakan satu tangan. Karena kondisi Pandemi Covid-19, referensi gerakan untuk data latih masih mengacu dari literatur dan video. Pada penelitian ini diberikan indikator berupa warna LED untuk mengetahui keberhasilan eksekusi gerakan.

2. METODE PENELITIAN

Metode penelitian pada penelitian ini dimulai dengan pemilihan enam gerakan olahraga yang akan dideteksi. Ide perancangan keseluruhan sistem pendeteksi gerakan pada alat olahraga *dumbbell* dijelaskan terlebih dahulu, dilanjutkan dengan langkah-langkah perancangan algoritma SVM untuk mendapatkan parameter terbaik berdasarkan nilai *accuracy*. Perancangan *hardware* dari sistem pendeteksi gerakan juga ditunjukkan termasuk sensor, pengontrol yang digunakan, dan indikator yang menunjukkan gerakan terdeteksi.

2.1 Gerakan Olahraga Menggunakan *Dumbbell*

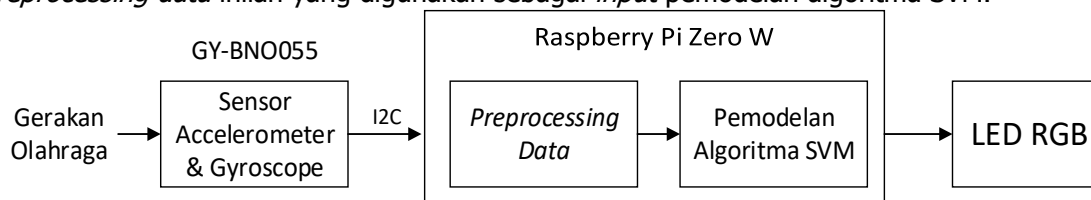
Penelitian dilakukan pada laboratorium, data latihan seharusnya diambil dari gerakan seorang instruktur olahraga. Namun pada kondisi Pandemi Covid-19 saat ini data hanya dapat diambil dari orang amatir. Program latihan *dumbbell* adalah cara yang mudah dan murah untuk melatih kekuatan di rumah tanpa perlu pergi ke sebuah *gym*. Program dasar yang baik dapat melibatkan enam gerakan berikut, yaitu *dumbbell curl*, *lateral raise*, *military press*, *seated rows*, *skull crusher* dan *forward punch* (Nasrulloh, dkk, 2018). Ilustrasi jenis-jenis gerakan dalam program latihan *dumbbell* ditunjukkan pada Gambar 1. Gambar 1 (a) menunjukkan gambar gerakan *dumbbell curl*, Gambar 1(b) menunjukkan gambar gerakan *lateral raise*, Gambar 1(c) menunjukkan gambar gerakan *military press*, Gambar 1(d) menunjukkan gambar gerakan *seated rows*, Gambar 1(e) menunjukkan gambar gerakan *tricep extension*, Gambar 1(f) menunjukkan gambar gerakan *forward punch*.



Gambar 1. (a) *Dumbbell Curl* (b) Gerakan *Lateral Raise* (c) Gerakan *Military Press* (d) Gerakan *Seated Rows* (e) Gerakan *Tricep Extension* (f) Gerakan *Forward Punch*

2.2 Perancangan Sistem Pendeteksi Gerakan

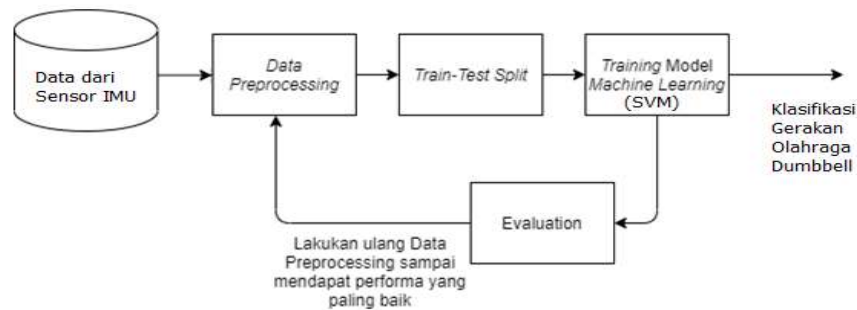
Pada sistem ini *dumbbell* digerakkan mengikuti gerakan olahraga yang diusulkan (enam gerakan dasar *dumbbell*). Pada *dumbbell* dipasang sistem pendeteksi gerakan seperti yang ditunjukkan pada Gambar 2. Gerakan akan dideteksi oleh sensor *Inertial Measurement Unit* (IMU) dan membaca nilai akselerasi dan kecepatan sudut (Sartika, dkk., 2020). Data gerakan dikirim ke *Raspberry Pi Zero W* untuk dilakukan *preprocessing data* agar data dikonversi menjadi format yang sesuai melalui ekstraksi fitur untuk mengkombinasi variabel agar dapat mendeskripsikan data lebih akurat. Pada penelitian ini ekstraksi fitur berdasarkan *maximum*, *minimum*, *range*, *median*, dan *standar deviasi* untuk mendeteksi gerakan. Hasil *preprocessing data* inilah yang digunakan sebagai *input* pemodelan algoritma SVM.



Gambar 2. Blok Diagram Sistem Pendeteksi Gerakan

Raspberry Pi Zero W diprogram agar dapat mengolah data hasil *preprocessing* menggunakan algoritma SVM untuk mendapatkan model deteksi gerakan olahraga pada *dumbbell*. Bila gerakan terdeteksi maka *Raspberry Pi Zero W* akan memberi perintah LED RGB (Tanaka, dkk., 2011) untuk menyalakan warna tertentu sesuai dengan deteksi gerakan. Sistem pendeteksi gerakan terdiri dari dua bagian, *hardware* dan *software*. Pada bagian *hardware* terdapat sensor IMU GY-BNO055 (Sensortec, 2016), *Single Board Computer Raspberry Pi Zero W*, dan LED RGB, sedangkan pada bagian *software* terdapat proses *preprocessing data* dan pemodelan algoritma SVM.

2.3 Perancangan Sistem Pendeteksi Gerakan menggunakan SVM

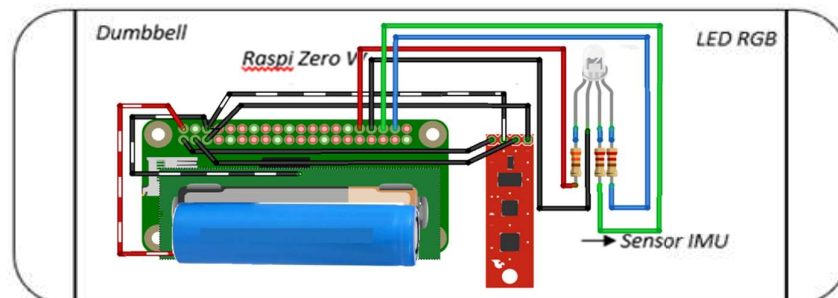


Gambar 3. Sistem Pendeteksi Gerakan menggunakan SVM

Sistem pendeteksi gerakan menggunakan SVM ditunjukkan pada Gambar 3. Data dari sensor IMU akan dilakukan proses *preprocessing* data yang mencakup normalisasi *feature* dan *feature selection* dengan tujuan agar data yang akan dimasukkan ke dalam model menjadi format yang sesuai untuk tahap selanjutnya. Setelah itu, dilakukan *train-test split* untuk memecah data menjadi *train data* dan *test data*. *Train data* adalah data yang akan dimasukkan ke dalam model, sedangkan *test data* adalah data yang akan digunakan untuk pengujian model yang dilakukan untuk mengetahui performa dari model yang dibuat. Setelah itu dilakukan *training* model menggunakan algoritma dalam *Machine Learning* yaitu SVM (Sam & Chatwin, 2015). Pengujian dilakukan untuk menentukan fungsi kernel dan parameter dari algoritma SVM. Dari berbagai jenis fungsi kernel dan parameter dari algoritma yang dipakai, akan dilakukan evaluasi untuk memilih performa dari model berdasarkan nilai *accuracy* (Medar, dkk, 2018).

2.4 Perancangan Perangkat Keras Sistem Pendeteksi Gerakan

Perangkat keras yang digunakan pada sistem deteksi gerakan olahraga menggunakan *dumbbell* terdiri dari sensor IMU GY-BNO055, *Single Board Computer Raspberry Pi Zero W*, dan LED RGB. Perancangan peletakan *hardware* sistem pendeteksi gerakan pada *dumbbell* ditunjukkan pada Gambar 4.

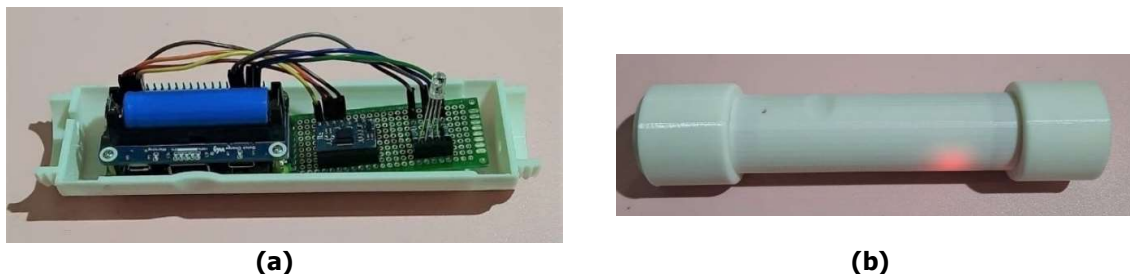


Gambar 4. Perangkat Keras Sistem Pendeteksi Gerakan pada Dumbbell

Pada perancangan perangkat keras, untuk dapat mendeteksi gerakan maka dibutuhkan sensor yang dapat membaca pergerakan. Sensor IMU GY-BNO055 dapat membaca berbagai keluaran data seperti data *accelerometer*, *gyroscope*, *magnetometer* dan *quaternion*. Parameter yang digunakan adalah *accelerometer* dan *gyroscope* untuk mendeteksi gerakan *dumbbell*. Terdapat masing-masing tiga nilai akselerasi dan kecepatan sudut yang digunakan yaitu arah x, arah y, dan arah z. Ukuran sensor IMU cukup kecil sehingga dapat disimpan dengan mudah dalam *dumbbell*.

Single Board Computer (SBC) dibutuhkan untuk mengontrol sensor, mengolah data, dan mengaplikasikan *machine learning*. Agar dapat disimpan pada alat *dumbbell*, maka dibutuhkan SBC yang berukuran kecil, sehingga digunakan *Raspberry Pi Zero W* dengan spesifikasi dimensi 66.0mm x 30.5mm x 5.0mm, prosesor 1GHz, RAM 512 MB. Keuntungan *Raspberry Pi Zero W* adalah ketersediaan *tools* yang umum digunakan untuk aplikasi *machine learning* seperti *python* dan *scikit-learn*. Namun karena kemampuan komputasi yang kecil maka terdapat keterbatasan *memory* dan *latency* untuk aplikasi *machine learning*. Hasil dari deteksi gerakan memerlukan indikator yang dapat mewakili gerakan yang terdeteksi. LED RGB digunakan untuk mengeluarkan enam warna yang berbeda sesuai enam gerakan *dumbbell*.

Realisasi perangkat keras sistem pendeteksi gerakan ditunjukkan pada Gambar 5 (a), terdapat *Single Board Computer Raspberry Pi Zero W*, yang di-*stack* dengan *li-ion battery HAT*, sensor IMU GY-BNO055, dan LED RGB. Gambar 5 (b) menunjukkan keseluruhan *dumbbell* yang telah dipasang, beban dipasang di kanan dan di kiri sekaligus untuk menutup perangkat keras elektronika dari *dumbbell*.



(a)

(b)

Gambar 5. Realisasi *Dumbbell* Pendeteksi Gerakan

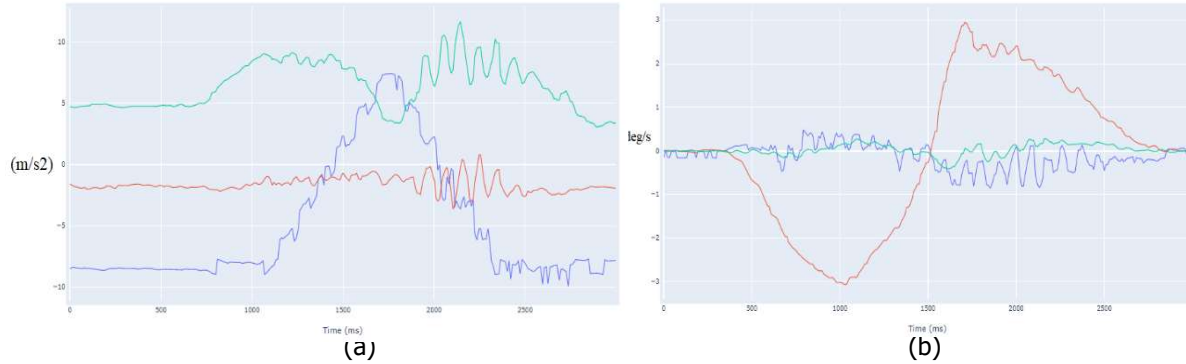
3. DATA PENGAMATAN DAN ANALISIS

3.1 Pengolahan Data Akselerasi dan Kecepatan Sudut pada Sensor IMU

Pada pengujian nilai akselerasi dan kecepatan sudut dibaca oleh sensor IMU untuk setiap gerakan yang dilakukan. Nilai akselerasi dan kecepatan sudut ditampilkan dalam bentuk grafik. Data tersebut merupakan data mentah yang selanjutnya akan diproses untuk dijadikan fitur untuk *input* pemodelan algoritma SVM.

Data setiap gerakan dari sensor IMU diberi label/kelas, terdapat tujuh kelas yaitu *dumbbell curl*, *lateral raise*, *military press*, *seated rows*, *tricep extension*, *forward punch*, dan *negative*. *Negative* adalah kelas di luar enam gerakan yang akan dideteksi (gerakan salah). *Starting number* ditentukan dimulai dari 000, dan durasi gerakan ditentukan dalam satuan detik. Selanjutnya sensor merekam data dan disimpan pada *Raspberry Pi Zero W*. Data yang diambil sebanyak 50 kali untuk setiap label gerakan.

Gambar 6 adalah grafik nilai akselerasi dan kecepatan sudut salah satu gerakan olahraga tangan yaitu gerakan *dumbbell curl*. Grafik tersebut mendeskripsikan: posisi awal *dumbbell* digenggam dengan tangan lurus dan sisi dalam pergelangan tangan menghadap depan, selanjutnya *dumbbell* digerakkan ke atas hingga siku tertekuk lalu turun kembali ke posisi awal.



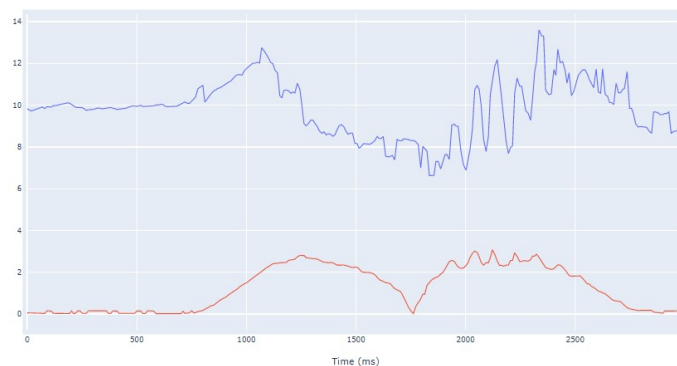
Keterangan:

- = Nilai akselerasi sumbu x dengan satuan m/s^2
- = Nilai akselerasi sumbu y dengan satuan m/s^2
- = Nilai akselerasi sumbu z dengan satuan m/s^2

Gambar 6. (a) Grafik Nilai Akselerasi dan (b) Kecepatan Sudut Sumbu X,Y,Z Gerakan *Dumbbell Curl*

Preprocessing data dilakukan pada perangkat lunak *jupyter notebook* yang diawali dengan inisialisasi library: *numpy*, *pandas* dan *scikit-learn* (Douglass, 2020). Kemudian dilakukan *import data* yang dilanjutkan dengan *normalize data* untuk data accelerometer sumbu-x, y, z, dan data gyroscope sumbu-x, y,z, sehingga diperoleh satu nilai data akselerasi dan kecepatan sudut. *Normalize data* adalah suatu proses untuk menghitung *norm* dari sebuah vektor atau matriks $i * j$. Fungsi yang digunakan adalah fungsi *Euclidian Norm*, ditunjukkan pada Persamaan (1). Gambar 7 menunjukkan grafik akselerasi dan kecepatan sudut *dumbbell-curl* yang sudah dinormalisasi.

$$||A|| = \left[\sum_{i,j} abs(a_{i,j})^2 \right]^{\frac{1}{2}} \quad (1)$$



Keterangan:

- = Nilai akselerasi setelah dilakukan *normalize data*
- = Nilai kecepatan sudut setelah dilakukan *normalize data*

Gambar 7. Grafik Nilai Akselerasi (Biru) dan Kecepatan Sudut (Merah) Sumbu X,Y,Z Gerakan *Dumbbell Curl* Sesudah *Normalize Data*

Tahap selanjutnya adalah ekstraksi fitur. Fitur yang digunakan adalah nilai *minimum*, *maximum*, *range*, *mean* dan *standard deviation* (Saloman, 2013). Fitur kemudian diekstraksi ke dalam dua variabel, yaitu *independent variable* dan *dependent variable*. Gambar 8 adalah contoh fitur-fitur yang telah diekstraksi yang terdiri dari *independent variabel*, yaitu *max_accel*, *min_accel*, *range_accel*, *mean_accel*, *std_accel*, *max_gyro*, *min_gyro*, *range_gyro*, *mean_gyro*, *std_gyro* dan *dependent variable* yaitu *output* (nama gerakan). Tahap terakhir dari *preprocessing* data adalah pembagian data menjadi *training* data dan *testing* data dengan persentase *training* data 75 % dan *testing* data 25 %. *Training* data digunakan untuk melatih model algoritma sedangkan *testing* data digunakan untuk menguji akurasi dari model yang telah dilatih.

	max_accel	min_accel	range_accel	mean_accel	std_accel	max_gyro	min_gyro	range_gyro	mean_gyro	std_gyro	output
0	13.600537	6.630422	6.970114	9.826974	1.343535	3.083480	0.011055	3.072424	1.244311	1.049440	dumbbell_curl
1	12.187691	5.491102	6.696589	9.662275	1.370167	3.077737	0.009493	3.068244	1.382631	1.023928	dumbbell_curl
2	11.661977	7.669681	3.992296	9.896116	0.880198	2.079414	0.151885	1.927529	1.471728	0.448545	dumbbell_curl
3	13.609431	5.670952	7.938478	9.977962	1.733070	3.768167	0.003333	3.764834	1.302342	1.269364	dumbbell_curl
4	13.714930	6.896709	6.818220	9.649758	1.323565	2.565824	0.004006	2.561817	1.178175	0.890796	dumbbell_curl
5	13.665848	5.005147	8.660701	9.402583	1.860009	2.548862	0.006383	2.542479	0.988625	0.789232	lateral_raise
6	12.687131	5.592620	7.094511	9.676369	1.627488	2.191748	0.018122	2.173626	1.043794	0.636189	lateral_raise
7	14.250747	4.197392	10.053355	9.822005	2.105396	2.428605	0.016518	2.412087	0.946779	0.704935	lateral_raise
8	14.022913	6.206086	7.816828	9.619721	1.667933	2.637195	0.014614	2.622581	0.994111	0.689468	lateral_raise
9	13.469321	6.423317	7.046004	9.468958	1.575755	2.228943	0.008165	2.220778	0.873819	0.705279	lateral_raise
10	12.762519	7.352585	5.409935	10.333436	1.348258	1.080211	0.039550	1.040661	0.405445	0.234709	military_press
11	12.840206	6.845583	5.994624	10.404251	1.388919	1.072265	0.046720	1.025545	0.475727	0.237297	military_press
12	12.866627	7.343439	5.523188	10.409018	1.421263	1.350562	0.017951	1.332611	0.532573	0.307888	military_press
13	15.073676	6.782551	8.291125	10.289175	1.739275	1.353011	0.031662	1.321349	0.532014	0.284354	military_press
14	12.794929	7.740788	5.054141	10.240367	1.011607	1.334800	0.067293	1.267506	0.479784	0.260937	military_press
15	11.695354	5.789024	5.906330	8.986123	1.410797	1.927595	0.015154	1.912441	0.424505	0.358021	seated_rows
16	13.782271	5.788705	7.993567	9.132039	1.570569	1.487480	0.012070	1.475411	0.468847	0.353973	seated_rows
17	11.934509	5.852051	6.082458	9.042952	1.286193	0.963229	0.009750	0.953479	0.320026	0.236104	seated_rows

Gambar 8. Contoh Fitur yang Telah Diekstraksi

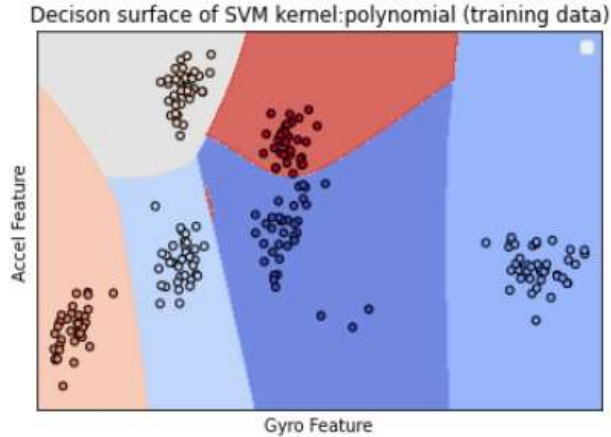
3.2 Pemilihan Kernel dan Parameter pada Model Algoritma SVM

Pengujian dilakukan untuk menentukan fungsi kernel dan parameter dari model SVM. Dari 350 data (7 kelas x 50 data), 75% digunakan untuk training yaitu sekitar 264 data dan sisanya untuk testing. Berdasarkan lima percobaan dengan kernel dan parameter yang berbeda, nilai akurasi terendah diperoleh model SVM dengan fungsi kernel *RBF*, nilai parameter $C=1$ dan $\text{Gamma}='auto'$, sedangkan nilai akurasi tertinggi tetap diperoleh model SVM yang dengan fungsi kernel *Polynomial*, nilai parameter $C=0.75$, $\text{Gamma}=0.1$ dan $\text{Degree}=3$, seperti ditunjukkan pada Tabel 1.

Tabel 1. Hasil Testing Pengujian Pemilihan Kernel dan Parameter Model SVM

Percobaan	Algoritma	Kernel	C	Gamma	Degree	Akurasi Test
1	SVM	Linear	0.75	-	-	91.10%
2	SVM	RBF	1	1.0	-	88.63%
3	SVM	RBF	0.5	0.5	-	92.04%
4	SVM	Polynomial	0.75	0.1	3	96.59%
5	SVM	Polynomial	1	1.0	2	93.18%

Dari 75% data (*training data*) dan 25% data (*testing data*) yang diambil dari satu orang, maka terpilih model SVM pada percobaan ke-4 sebagai model pendeteksi gerakan. Hal ini dikarenakan fungsi kernel *Polynomial* memiliki tingkat akurasi pengujian *test* data yang paling baik untuk *input* data yang memiliki lebih dari dua kelas/label.

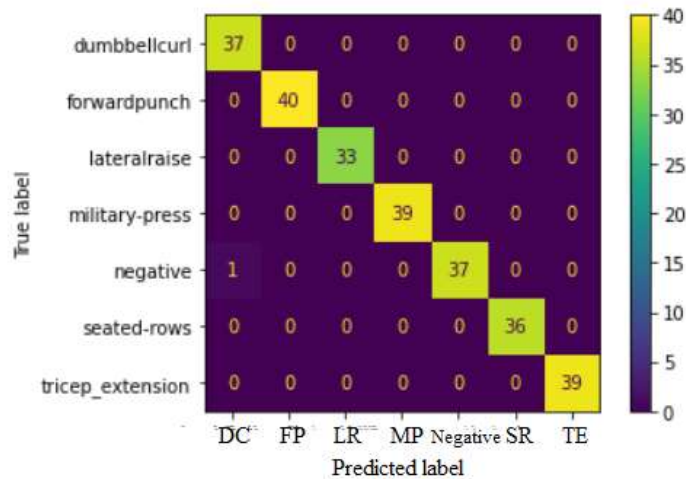


Gambar 9. Hasil Pelatihan Model SVM menggunakan *Training Data* (Fungsi Kernel *Polynomial* $C=0.75$, $\Gamma=0,1$, $Degree=3$)

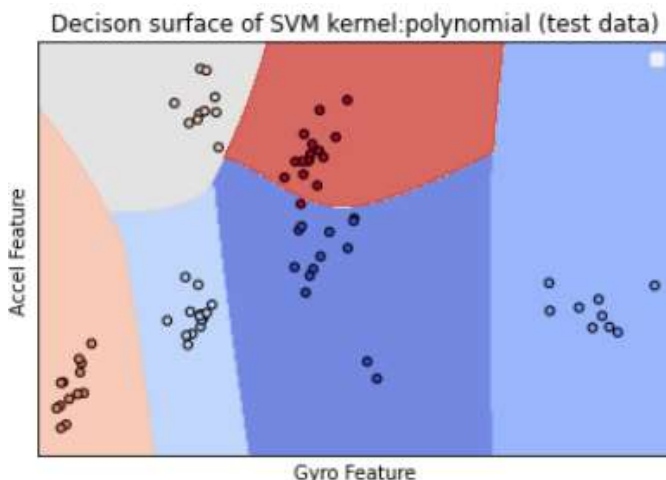
Keterangan :

- = *Dumbbell Curl*
- = *Lateral Raise*
- = *Military Press*
- = *Seated Rows*
- = *Tricep Extension*
- = *Forward Punch*

Hasil *training* model SVM yang menggunakan *training data* dengan fungsi kernel *Polynomial*, nilai parameter $C=0.75$, $\Gamma=0.1$, dan $Degree=3$ dapat dilihat pada Gambar 9. Hasil *training* yang dapat divisualisasikan terbatas, maka dari itu hanya ditampilkan hasil *training* enam kelas dari tujuh kelas, yaitu kelas *dumbbell curl*, *lateral raise*, *military press*, *seated rows*, *tricep extension* dan *forward punch*. Tabel *confusion matrix* (Marom, dkk, 2010) dari hasil *training* model SVM menggunakan *training data* dapat dilihat pada Gambar 10. Akurasi *training* model SVM adalah 99.61%.



Gambar 10. Tabel *Confusion Matrix* dari Pelatihan Model SVM menggunakan *Training Data* (Fungsi Kernel *Polynomial* $C=0.75$, $\Gamma=0,1$, $Degree=3$)

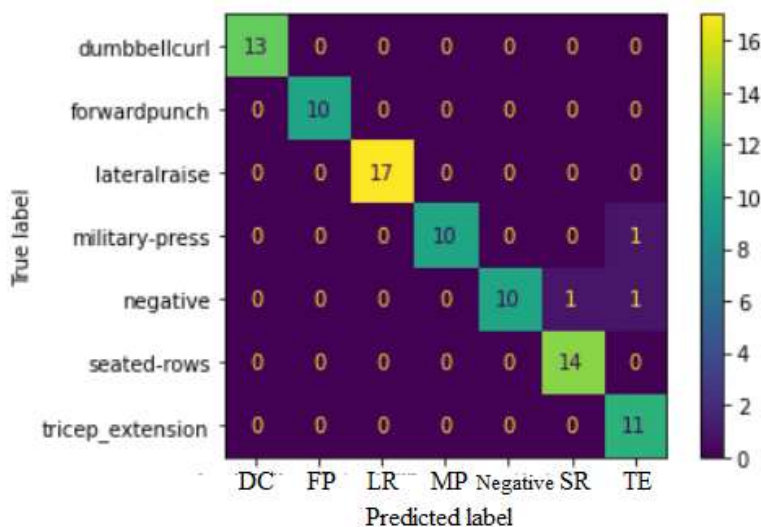


Gambar 11. Hasil Pengujian Model SVM menggunakan *Test Data* (Fungsi *Kernel Polynomial* $C=0.75$, $\Gamma=0,1$, $Degree=3$)

Keterangan :

- = *Dumbbell Curl*
- = *Lateral Raise*
- = *Military Press*
- = *Seated Rows*
- = *Tricep Extension*
- = *Forward Punch*

Hasil pengujian model SVM menggunakan *test data* dengan fungsi kernel *Polynomial*, nilai parameter $C=0.75$, $\Gamma=0.1$ dan $Degree=3$ dapat dilihat pada Gambar 11. Tabel *confusion matrix* dari pengujian menggunakan *test data* pada model SVM dengan fungsi kernel *Polynomial*, parameter $C=0.75$, $\gamma=0,1$, $degree=3$ ditunjukkan pada Gambar 12. Akurasi yang dihasilkan dari pengujian menggunakan *test data* adalah 96.59%.



Gambar 12. Tabel *Confusion Matrix* Dari Pengujian menggunakan *Test Data* (Fungsi *Kernel Polynomial* $C=0.75$, $\Gamma=0,1$, $Degree=3$)

3.3 Uji *Dumbbell* Pendeteksi Gerakan dalam Mendeteksi Gerakan

Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan sistem pendeteksi gerakan olahraga. Hasil dari pengujian *dumbbell* dalam mendeteksi gerakan yang benar, ditunjukkan pada Tabel 2.

Tabel 2. Hasil *Dumbbell* dalam Mendeteksi Gerakan Benar

	Responden 1 (24 thn)	Responden 2 (24 thn)	Responden 3 (24 thn)	Responden 4 (24 thn)	Responden 5 (24 thn)	Rata-rata
Jumlah Sampel	25	25	25	25	25	
<i>Dumbbell Curl</i>	24/25 (96%)	23/25 (92%)	25/25 (100%)	22/25 (88%)	25/25 (100%)	95.2%
<i>Lateral Raise</i>	24/25 (96%)	24/25 (96%)	21/25 (84%)	23/25 (92%)	21/25 (84%)	90.4%
<i>Military Press</i>	22/25 (88%)	23/25 (92%)	23/25 (92%)	22/25 (88%)	24/25 (96%)	91.2%
<i>Seated Rows</i>	25/25 (100%)	24/25 (96%)	25/25 (100%)	24/25 (96%)	23/25 (92%)	96.8%
<i>Tricep Extension</i>	22/25 (88%)	19/25 (76%)	17/25 (68%)	21/25 (84%)	18/25 (72%)	77.6%
<i>Forward Punch</i>	24/25 (96%)	24/25 (96%)	25/25 (100%)	24/25 (96%)	24/25 (96%)	96.8%
Total Error	9/150 (6%)	13/25 (77%)	14/150 (9.33%)	14/150 (9.33%)	15/150 (10%)	8.6%
Presentase Keberhasilan	141/150 (94%)	137/150 (91,33%)	136/150 (90.66%)	136/150 (90.66%)	135/150 (90%)	

Pengujian ini dilakukan kepada lima orang dari segi umur yang berbeda sebanyak 25 kali untuk setiap gerakan olahraga. Keberhasilan *dumbbell* dalam mendeteksi gerakan ditunjukkan dengan keberhasilan indikator warna yang menyala ketika gerakan dilakukan, daftar indikator warna untuk gerakan dapat dilihat pada Tabel 3.

Tabel 3. Daftar Indikator Warna LED RGB

Warna LED	Jenis Gerakan	Durasi (detik)	Kode Representasi (RGB)
<i>Red</i>	<i>Dumbbell Curl</i>	3	100
<i>Blue</i>	<i>Lateral Raise</i>	3	010
<i>Green</i>	<i>Military Press</i>	3	001
<i>Yellow</i>	<i>Seated Rows</i>	3	110
<i>Cyan</i>	<i>Tricep Extension</i>	3	011
<i>Pink</i>	<i>Forward Punch</i>	3	101

Total presentase keberhasilan dari pengujian *dumbbell* ini berada di atas 90% dengan presentasi tertinggi 94% dan terendah 90%. Presentase total *error* dalam pengujian ini rata-rata sebesar 6-10%. Berdasarkan hasil pengujian, gerakan *tricep extension* memiliki tingkat keberhasilan paling rendah. Hal ini disebabkan karena terdapat kemiripan nilai fitur pada gerakan *dumbbell curl* & *tricep extension* (gerakan sama-sama memiliki posisi awal tangan lurus lalu menekuk siku).

3.4 Uji *Dumbbell* Pendeteksi Gerakan dalam Mendeteksi Gerakan menggunakan Tangan Kiri

Sebagai pengujian tambahan, *dumbbell* diujikan dalam mendeteksi gerakan olahraga menggunakan tangan kiri. Pengujian tambahan ini dimaksudkan agar bila memiliki 2 buah *dumbbell*, dapat dilakukan gerakan olahraga yang memerlukan dua buah *dumbbell*. Hasil

pengujian *dumbbell* pendeteksi gerakan menggunakan tangan kiri ditunjukkan pada Tabel 4.

Tabel 4. Hasil *Dumbbell* menggunakan Tangan Kiri dalam Mendeteksi Gerakan Benar

No	Jenis Gerakan	Berhasil	Error	% Keberhasilan
1	Dumbbell Curl	24/25 (96%)	1/25 (4%)	135/150 (90%)
2	Lateral Raise	21/25 (84%)	4/25 (16%)	
3	Military Press	23/25 (92%)	2/25 (8%)	
4	Seated Rows	22/25 (88%)	3/25 (12%)	
5	Tricep Extension	22/25 (88%)	3/25 (12%)	
6	Forward Punch	23/25 (92%)	2/25 (8%)	

Pengujian dilakukan untuk mengetahui tingkat keberhasilan dari *dumbbell* pendeteksi gerakan ketika menggunakan tangan kiri. Pengujian dilakukan sebanyak 25 kali untuk setiap jenis gerakan. Jika dibandingkan dengan hasil yang ditunjukkan pada Tabel 5 terdapat penurunan tingkat keberhasilan untuk beberapa gerakan seperti *Lateral Raise*, *Seated Rows* dan *Forward Punch* (terjadi penurunan sekitar 2-3%). Namun terdapat juga kenaikan tingkat keberhasilan untuk beberapa gerakan seperti *Dumbbell Curl*, *Military Press* dan *Tricep Extension*. Secara keseluruhan tingkat keberhasilan mencapai 90%, berdasarkan hasil ini perbedaan penggunaan tangan kanan atau tangan kiri pada *dumbbell* pendeteksi gerakan tidak berpengaruh secara signifikan terhadap tingkat keberhasilan deteksi gerakan.

4. KESIMPULAN

1. Pada penelitian ini berhasil mendeteksi gerakan tangan menggunakan *Support Vector Machine* pada *dumbbell* berbasis *Raspberry Pi Zero*.
2. Berdasarkan hasil uji coba pengaturan fungsi kernel dan parameter, terpilih model SVM dengan fungsi kernel *Polynomial*, nilai parameter $C=0.75$, $\gamma = 0.1$ dan $\text{degree} = 3$, yang memiliki akurasi (96.59%)
3. Berdasarkan hasil uji coba yang dilakukan kepada lima orang, dengan 25 sampel untuk setiap enam jenis gerakan olahraga, *dumbbell* pendeteksi gerakan memiliki tingkat keberhasilan 90%-94%
4. Berdasarkan hasil pengujian, gerakan *tricep extension* memiliki tingkat keberhasilan paling rendah. Hal ini disebabkan karena terdapat kemiripan nilai fitur pada gerakan *dumbbell curl & tricep extension* (gerakan sama-sama memiliki posisi awal tangan lurus lalu menekuk siku).

DAFTAR RUJUKAN

- Douglass, M. J. J. (2020). Book Review: Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition by Aurélien Géron. In *Physical and Engineering Sciences in Medicine*, 43(3). <https://doi.org/10.1007/s13246-020-00913-z>
- Fisher, J. P., Steele, J., Gentil, P., Giessing, J., & Westcott, W. L. (2017). A minimal dose approach to resistance training for the older adult; the prophylactic for aging.

- Experimental Gerontology*, 99, 80–86. <https://doi.org/10.1016/j.exger.2017.09.012>
- Kumar, G., Banerjee, R., Kr Singh, D., Choubey, N., & Arnaw. (2020). Mathematics for Machine Learning. *Journal of Mathematical Sciences & Computational Mathematics*, 1(2), 229–238. <https://doi.org/10.15864/jmscm.1208>
- Marom, N. D., Rokach, L., & Shmilovici, A. (2010). Using the confusion matrix for improving ensemble classifiers. *2010 IEEE 26th Convention of Electrical and Electronics Engineers in Israel, IEEEI 2010*, (pp. 555–559). <https://doi.org/10.1109/IEEEI.2010.5662159>
- Medar, R., Rajpurohit, V. S., & Rashmi, B. (2018). Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning. *2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017*, (pp. 1–6). <https://doi.org/10.1109/ICCUBEA.2017.8463779>
- Nasrulloh, A., Prasetyo, Y., & Apriyanto, K. D. (2018). Dasar-dasar Latihan Beban. In *UVY Press*. <https://docplayer.info/163394993-Dasar-dasar-latihan-beban-ahmad-nasrulloh-yudik-prasetyo-krisnanda-dwi-apriyanto.html>
- Saloman, J.-B. (2013). Probability and Statistics (Fourth Edition). In *Chance*, 26(3). <https://doi.org/10.1080/09332480.2013.845457>
- Sam, K. M., & Chatwin, C. R. (2015). Evaluating the Effectiveness of Online Product Planning and Layout Tools in Online Apparel Shopping. In *Proceedings of the 2015 IEEE IEEM Perceived*, (pp. 635–639).
- Sartika, E. M., Gany, A., & Yuvens, V. (2020). Implementasi Sensor IMU untuk mengetahui Sudut Elevasi Kendaraan menggunakan Metode Least Square. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 8(2), 301. <https://doi.org/10.26760/elkomika.v8i2.301>
- Sensortec, B. (2016). *BNO055: Intelligent 9-axis absolute orientation sensor*. https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST_BNO055_DS000_14.pdf
- Somvanshi, M., Chavan, P., Tambade, S., & Shinde, S. V. (2017). A review of machine learning techniques using decision tree and support vector machine. *Proceedings - 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA 2016*. <https://doi.org/10.1109/ICCUBEA.2016.7860040>
- Sukvichai, K., Uthaisang, P., Chuengsutthiwong, P., & Maolanon, P. (2018). Hidden Dot Patterns Recognition using CNNs on Raspberry Pi Zero W. *2018 International Conference on Embedded Systems and Intelligent Technology and International Conference on Information and Communication Technology for Embedded Systems, ICESIT-ICICTES 2018*, (pp. 1–5). <https://doi.org/10.1109/ICESIT->

ICICTES.2018.8442050

- Tanaka, M., Horiuchi, T., & Tominaga, S. (2011). Color control of a lighting system using RGBW LEDs. *Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, 7866, 78660W. <https://doi.org/10.1117/12.872374>
- Velloso, E., Bulling, A., Gellersen, H., Ugulino, W., & Fuks, H. (2013). Qualitative activity recognition of weight lifting exercises. *ACM International Conference Proceeding Series, December 2014*, (pp. 116–123). <https://doi.org/10.1145/2459236.2459256>
- Wang, F., Yan, L., & Xiao, J. (2019). Human gait recognition system based on support vector machine algorithm and using wearable sensors. *Sensors and Materials*, 31(4), 1335–1349. <https://doi.org/10.18494/SAM.2019.2288>
- Wang, Y., Zhao, Y., Xia, G., Qiu, A., Shi, Q., & Liu, R. (2018). A Scale Factor Self-Calibration Method for a Batch of MEMS Gyroscopes Based on Virtual Coriolis Force. *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology*, (pp. 1–3). <https://doi.org/10.1109/ICSICT.2018.8564926>