

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi adalah penerapan hasil perancangan yang telah dilakukan pada tahap analisis dan perancangan sistem. Hasil perancangan diterapkan menjadi sebuah sistem untuk implementasi metode *Support Vector Machine* dalam mengklasifikasikan konten instagram berdasarkan komentar. Implementasi yang dilakukan antara lain menerapkan hasil analisis dan perancangan perangkat keras, perangkat lunak, dan antarmuka.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam membangun sistem untuk implementasi metode *Support Vector Machine* dalam mengklasifikasikan konten instagram berdasarkan komentar terdapat pada tabel 4.1:

Tabel 4.1 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	<i>Processor</i>	Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz(8 CPUs), ~ 2.4GHz
2	Memori (RAM)	8.00 GB DDR3
3	Monitor	15,6 inch dengan resolusi 1920 x 1080 pixel
4	<i>Keyboard</i>	Standar
5	<i>Mouse</i>	Standar

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun sistem untuk implementasi metode *Support Vector Machine* dalam mengklasifikasikan konten instagram berdasarkan komentar terdapat pada tabel 4.2:

Tabel 4.2 Implementasi Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 10 Pro 64-bit
2	Bahasa Pemrograman	Python 3.6.1
3	<i>Tools</i>	Pycharm, Jupyter Notebook
4	Software Pendukung	Flask

Penjelasan dari implementasi perangkat lunak di tabel 4.2:

1. Sistem Operasi yang digunakan yaitu Microsoft windows 10 Pro 64-bit. Windows 10 merupakan sistem operasi komputer pribadi yang dikembangkan oleh Microsoft sebagai bagian dari keluarga sistem operasi Windows NT.
2. Bahasa pemrograman yang digunakan yaitu python 3.6.1. Python adalah bahasa pemrograman interpretative multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python 3.6.1 dalam penelitian ini digunakan untuk mengimplementasikan program yang dibuat untuk mengklasifikasikan konten Instagram berdasarkan komentar menggunakan *Support Vector Machine*.
3. Tools yang digunakan dalam membuat program yaitu pycharm dan jupyter notebook. Untuk kedua tools tersebut digunakan sebagai *compiler* Bahasa pemrograman python.
4. Software pendukung yang digunakan yaitu library flask yang dimana berfungsi untuk menampilkan antarmuka pengguna yang semula masih berupa *console*.

4.1.3 Implementasi Antarmuka

Berikut ini adalah penjelasan implementasi antarmuka pada sistem untuk implementasi metode *Support Vector Machine* dalam mengklasifikasikan konten instagram berdasarkan komentar terdapat pada tabel 4.3:

Tabel 4.3 Implementasi Antarmuka

No	Nama Antarmuka	Deskripsi
1	Mengelola Data Latih	Menampilkan halaman untuk menambahkan dan menghapus data latih yang akan diproses sebagai model <i>Support Vector Machine</i> .
2	Mengelola Data Uji	Menampilkan halaman untuk menambahkan dan menghapus data uji. Yang akan diproses untuk menentukan hasil prediksi kelas dari data uji dengan menggunakan <i>Support Vector Machine</i> .
3	Pengujian SVM	Menampilkan halaman untuk memilih data latih dan data uji yang akan diproses dari data yang sudah dipilih sebelumnya. Pada bagian ini kita pilih tombol proses untuk menampilkan hasil dari <i>preprocessing</i> yaitu tahapan <i>case folding, cleansing, filtering, tokenizing, normalisasi Bahasa, stopword removal</i> . Setelah itu pembobotan <i>tf-idf</i> untuk data latih dan data uji. Serta menampilkan hasil klasifikasi dari data testing dan akurasinya

4.2 Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem yang dibuat sesuai dengan rancangan sistem yang telah dibuat sebelumnya. Pengujian sistem yang dilakukan meliputi tahap pengujian fungsionalitas, pengujian detail perancangan dan pengujian performansi.

4.2.1 Rencana Pengujian

Sebelum pengujian dilakukan, diperlukan rencana pengujian untuk menentukan tahap pengujian yang akan dilakukan. Berikut rencana pengujian yang akan dilakukan pada penelitian ini:

1. Rencana Pengujian
2. Skenario Pengujian
3. Hasil Pengujian
4. Evaluasi

4.2.2 Skenario Pengujian

Berdasarkan rencana pengujian yang telah dijelaskan, dilakukan skenario pengujian untuk setiap rencana pengujian. Skenario pengujian dibagi menjadi dua yaitu skenario pengujian *black box* dan skenario pengujian akurasi. Berikut penjelasan untuk setiap skenario pengujian tersebut:

1. Skenario Pengujian *Black Box*

Pengujian *black box* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Skenario pengujian *black box* dapat dilihat pada Tabel 4.4 berikut:

Tabel 4.4 Skenario Pengujian *Black Box*

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1	Input Data Latih	Menambahkan data latih dengan format csv ke dalam sistem.	<i>Black box</i>
2	Hapus Data Latih	Menghapus data latih yang sudah ada di list data latih yang terdapat dalam sistem.	<i>Black box</i>
4	Input Data Uji	Menambahkan data latih dengan format csv ke dalam sistem.	<i>Black box</i>
5	Hapus Data Uji	Menghapus data latih yang sudah ada di list data latih yang terdapat dalam sistem.	<i>Black box</i>
6	Case Folding	Mengubah data latih dan uji kedalam bentuk lowercase atau huruf kecil.	<i>Black box</i>
7	Cleansing	Menghilangkan kata yang mengandung karakter “@Username”, “URL” atau “#” yang ada pada data latih dan uji.	<i>Black box</i>
8	Filtering	Menghilangkan karakter selain a-z yang ada pada data latih dan uji.	<i>Black box</i>
9	Tokenizing	Apabila sistem menemukan “(spasi) maka sistem akan memecah kata tersebut yang terdapat pada data latih dan uji.	<i>Black box</i>

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
10	Normalisasi Bahasa	Melakukan pengecekan terhadap daftar list normalisasi bahasa, apabila ada kata yang dicek dan sesuai pada list maka kata tersebut akan diperbaiki menjadi kata yang sesuai dengan KBBI yang terdapat pada data latih dan uji.	<i>Black box</i>
11	Stopword Removal	Sistem akan melakukan pengecekan terhadap setiap kata yang terfapat pada list stopwords removal, dan jika dalam data latih dan uji terdapat kata yang sesuai dengan list stopwords removal maka akan dihapus.	<i>Black box</i>
12	Pembobotan tf-idf	Sistem akan melakukan pemberian nilai bobot terhadap setiap kata yang terdapat pada data latih dan uji, dan pembobotan kata akan dilakukan dengan menggunakan perhitungan tf-idf.	<i>Black box</i>
13	Klasifikasi SVM	Sistem akan melakukan klasifikasi dan akan mendapatkan nilai akurasi dari data uji.	<i>Black box</i>

2. Skenario Pengujian Akurasi

Pengujian akurasi merupakan tahap yang memiliki tujuan untuk mengetahui tingkat akurasi dari penggunaan metode *Support Vector Machine* dengan cara menghitung jumlah data uji yang kelasnya diprediksi secara benar. Cara mengukur kinerja dari sistem akan dilakukan menggunakan matriks konfusi. Matriks konfusi adalah tabel yang mencatat hasil kerja klasifikasi.

Pengujian dilakukan dengan menguji 100 postingan instagram yang belum diketahui kategori kelasnya menggunakan metode *Support Vector Machine*. Data uji yang digunakan sebanyak 75 data, sedangkan data latih yang digunakan sebanyak 200 data.

Pengujian akurasi dibagi menjadi 2 bagian, dimana setiap bagian tersebut akan digunakan jumlah data latih dan data uji yang berbeda. Pada pengujian akurasi pertama, data latih dan data uji yang digunakan sama dengan banyak 200 data, dan pengujian kedua 200 data latih dan 100 data uji.

4.2.3 Hasil Pengujian

Hasil pengujian merupakan hasil dari skenario pengujian yang dilakukan, sehingga hasil pengujian dibagi menjadi dua yaitu hasil pengujian *black box*, dan hasil pengujian akurasi. Berikut penjelasan untuk masing-masing pengujian:

1. Hasil Pengujian *Black Box*

Hasil pengujian *black box* berisi hasil pengujian dari setiap fungsionalitas pada struktur program. Berikut hasil pengujian *black box*:

Tabel 4.5 Hasil Pengujian Input Data Latih *Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=filename.csv	Sistem menampilkan pesan'File berhasil disimpan'	Sistem menampilkan pesan'File berhasil disimpan'	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong dan Tidak Sesuai Format)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=filename.(Format lain)	Sistem menampilkan pesan'Ekstensi file yang diperbolehkan :.csv'	Sistem menampilkan pesan'Ekstensi file yang diperbolehkan :.csv'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong	Sistem menampilkan pesan'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.6 Hasil Pengujian Hapus Data Latih *Black Box*

Hasil Pengujian (Hapus Data BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=filename.csv	Sistem menampilkan pesan 'File berhasil dihapus'	Sistem menampilkan pesan 'File berhasil dihapus'	[√] Diterima [] Ditolak

Tabel 4.7 Hasil Pengujian Input Data Uji *Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Uji=filename.csv	Sistem menampilkan pesan 'File berhasil disimpan'	Sistem menampilkan pesan 'File berhasil disimpan'	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong dan Tidak Sesuai Format)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Uji=filename.(Format lain)	Sistem menampilkan pesan 'Ekstensi file yang diperbolehkan :.csv'	Sistem menampilkan pesan 'Ekstensi file yang diperbolehkan :.csv'	[√] Diterima [] Ditolak
Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.8 Hasil Pengujian Hapus Data Uji *Black Box*

Hasil Pengujian (Hapus Data BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Uji=filename.csv	Sistem menampilkan pesan 'File berhasil dihapus'	Sistem menampilkan pesan 'File berhasil dihapus'	[√] Diterima [] Ditolak

Tabel 4.9 Hasil Pengujian *Case Folding Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=filename.csv Data Uji=filename.csv	Sistem menampilkan tahapan proses <i>Case Folding</i> .	Sistem menampilkan tahapan proses <i>Case Folding</i> .	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.10 Hasil Pengujian *Cleansing Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih= filename.csv Data Uji =filename.csv	Sistem menampilkan tahapan proses <i>Cleansing</i> .	Sistem menampilkan tahapan proses <i>Cleansing</i> .	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4.11 Hasil Pengujian *Filtering Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih= filename.csv Data Uji =filename.csv	Sistem menampilkan tahapan proses <i>Filtering</i> .	Sistem menampilkan tahapan proses <i>Filtering</i> .	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.12 Hasil Pengujian *Tokenizing Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih= filename.csv Data Uji =filename.csv	Sistem menampilkan tahapan proses <i>Tokenizing</i> .	Sistem menampilkan tahapan proses <i>Tokenizing</i> .	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.13 Hasil Pengujian *Normalisasi Bahasa Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih= filename.csv Data Uji =filename.csv	Sistem menampilkan tahapan proses <i>Normalisasi Bahasa.</i>	Sistem menampilkan tahapan proses <i>Normalisasi Bahasa.</i>	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.14 Hasil Pengujian *Stopword Removal Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih= filename.csv Data Uji =filename.csv	Sistem menampilkan tahapan proses <i>Stopword Removal</i> .	Sistem menampilkan tahapan proses <i>Stopword Removal</i> .	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.15 Hasil Pengujian Pembobotan *tf-idf* Black Box

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih= filename.csv Data Uji =filename.csv	Sistem menampilkan tahapan proses <i>Pembobotan tf-idf.</i>	Sistem menampilkan tahapan proses <i>Pembobotan tf-idf.</i>	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

Tabel 4.16 Hasil Pengujian *Klasifikasi SVM Black Box*

Hasil Pengujian (Data Masukan BENAR)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=filename.csv Data Uji=filename.csv	Sistem menampilkan tahapan proses <i>Klasifikasi dengan SVM</i> dan mendapatkan nilai akurasi dari pengujian data uji.	Sistem menampilkan tahapan proses <i>Klasifikasi dengan SVM</i> dan mendapatkan nilai akurasi dari pengujian data uji.	[√] Diterima [] Ditolak
Hasil Pengujian (Data Masukan Kosong)			
Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Data Latih=data tidak ada atau kosong Data Uji=data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih dan data uji, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=data tidak ada atau kosong Data Uji=filename.csv	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data latih, tidak boleh kosong'	[√] Diterima [] Ditolak
Data Latih=filename.csv Data Uji= data tidak ada atau kosong	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	Sistem menampilkan pesan 'Pilih data uji, tidak boleh kosong'	[√] Diterima [] Ditolak

4.2.3.1 Kesimpulan Hasil Pengujian Fungsional

Berdasarkan hasil pengujian yang telah dilakukan terhadap komponen-komponen uji, berkesimpulan bahwa pada input data latih, hapus data latih, input data uji dan hapus data uji dalam bentuk pesan yang ditampilkan sistem telah sesuai dengan apa yang diharapkan dan secara fungsional sistem yang telah dibangun bekerja dengan baik dan sesuai dengan yang diharapkan.

2. Hasil Pengujian Akurasi

Pada penelitian ini mengukur tingkat keakuratan dilakukan dengan menggunakan metode menghitung akurasi. Berikut adalah pengujian nilai akurasi yang telah dilakukan.

Pengujian akurasi yang dilakukan yaitu dengan menguji data yang tidak termasuk data latih. Sehingga dari pengujian ini diketahui tingkat pengenalan data uji dengan data latih yang telah dimasukkan. Berikut hasil pengujiannya:

a. Hasil Pengujian Akurasi Pertama

Berikut merupakan hasil kerja klasifikasi dengan menggunakan data latih dan data uji sebanyak 200 dengan data yang sama, hasilnya ada pada Tabel 4.17 berikut:

Tabel 4.17 Hasil Pengujian Model Klasifikasi

Kondisi	Linear	RBF			Polynomial		
		$\gamma=1$	$\gamma=2$	$\gamma=3$	n=1	n=2	n=3
SVM	100%	100%	100%	100%	100%	100%	100%

Hasil pengujian tersebut menunjukkan bahwa metode *Support Vector Machine* kernel linear menunjukkan akurasi yang paling besar dengan nilai akurasi 96%. Setelah itu akan dilihat hasil klasifikasi dokumen benar atau salah penempatan kedalam suatu kelas menggunakan matriks konfusi dapat dilihat pada tabel 4.18:

Tabel 4.18 Matriks Konfusi Pengujian Pertama

F _{ij}		Kelas Prediksi (j)				
		Game	Makanan	Olahraga	Teknologi	Film
Kelas Asli (i)	Game	40	0	0	0	0
	Makanan	0	40	0	0	0
	Olahraga	0	0	40	0	0
	Teknologi	0	0	0	40	0
	Film	0	0	0	0	40

Dari tabel 4.7 tersebut menunjukkan bahwa jumlah data uji yang diklasifikasi secara benar sebanyak 200 data dan hasil yang salah sebanyak 1 data. Hasil Pertama menunjukkan bahwa total data uji dikelas 1 (Game) sebanyak 40 data uji terprediksi secara benar. Hasil Kedua menunjukkan bahwa total data uji dikelas 2 (Makanan) sebanyak 40 data uji terprediksi secara benar. Hasil Ketiga menunjukkan bahwa total data uji dikelas 3 (Olahraga) sebanyak 40 data uji terprediksi secara benar. Hasil Keempat menunjukkan bahwa total data uji dikelas 4 (Teknologi) sebanyak 40 data uji terprediksi secara benar. Hasil Kelima menunjukkan bahwa total data uji dikelas 5 (Film) sebanyak 40 data uji terprediksi secara benar.

Perhitungan nilai akurasi dengan melihat matriks konfusi dan menggunakan persamaan :

$$\text{Akurasi} = \frac{\text{jumlah data yang diprediksi secara benar}}{\text{jumlah prediksi yang dilakukan}} \times 100\%$$

$$\text{Akurasi} = \frac{200}{200} \times 100\% = 100\%$$

b. Hasil Pengujian Akurasi Kedua

Berikut merupakan hasil kerja klasifikasi dengan menggunakan data latih sebanyak 200 data latih dan data uji sebanyak 100 data, hasil akurasi dapat dilihat pada Tabel 4.19 berikut:

Tabel 4.19 Hasil Pengujian Model Klasifikasi

Kondisi	Linear	RBF			Polynomial		
		$\gamma=1$	$\gamma=2$	$\gamma=3$	n=1	n=2	n=3
SVM	96%	94%	91%	89%	73%	74%	74%

Hasil pengujian tersebut menunjukkan bahwa metode *Support Vector Machine* kernel linear menunjukkan akurasi yang paling besar dengan nilai akurasi 96%. Setelah itu akan dilihat hasil klasifikasi dokumen benar atau salah penempatan kedalam suatu kelas menggunakan matriks konfusi dapat dilihat pada tabel 4.20:

Tabel 4.20 Matriks Konfusi Pengujian Kedua

F _{ij}		Kelas Prediksi (j)				
		Game	Makanan	Olahraga	Teknologi	Film
Kelas Asli (i)	Game	18	0	0	2	0
	Makanan	0	20	0	0	0
	Olahraga	0	0	20	0	0
	Teknologi	0	0	0	19	1
	Film	0	0	0	1	19

Dari tabel 4.9 tersebut menunjukkan bahwa jumlah data uji yang diklasifikasi secara benar sebanyak 96 data dan hasil yang salah sebanyak 4 data. Hasil Pertama menunjukkan bahwa total data uji dikelas 1 (Game) sebanyak 18 data uji terprediksi secara benar dan 2 data uji terprediksi salah dikelas 4. Hasil Kedua menunjukkan bahwa total data uji dikelas 2 (Makanan) sebanyak 20 data uji terprediksi secara benar. Hasil Ketiga menunjukkan bahwa total data uji dikelas 3 (Olahraga) sebanyak 20 data uji terprediksi secara benar. Hasil Keempat menunjukkan bahwa total data uji dikelas 4 (Teknologi) sebanyak 19 data uji terprediksi secara benar dan 1 data uji terprediksi salah di kelas 5.

Hasil Kelima menunjukkan bahwa total data uji dikelas 5 (Film) sebanyak 19 data uji terprediksi secara benar dan 1 data uji terprediksi salah dikelas 4.

Perhitungan nilai akurasi dengan melihat matriks konfusi dan menggunakan persamaan :

$$\text{Akurasi} = \frac{\text{jumlah data yang diprediksi secara benar}}{\text{jumlah prediksi yang dilakukan}} \times 100\%$$

$$\text{Akurasi} = \frac{96}{100} \times 100\% = 96\%$$

4.2.4 Evaluasi Pengujian

Klasifikasi konten Instagram berdasarkan komentar dengan menggunakan metode *Support Vector Machine* dengan Kernel *Linear* terhadap variasi jumlah data latih dan data uji yang telah dilakukan. Dari pengujian pertama dengan menggunakan data latih dan data uji yang sama dengan banyaknya data 200 mendapatkan nilai akurasi 100%, dan untuk pengujian kedua dengan menggunakan data latih sebanyak 200 dan untuk data uji sebanyak 100 data didapatkan hasil nilai akurasi sebesar 96%. Sebelumnya ada penelitian yang dilakukan oleh Dio Ariandi dan Kartika tentang klasifikasi dengan menggunakan metode *Support Vector Machine*[2], penelitian yang dilakukan untuk mengklasifikasikan berita yang dimana menggunakan kernel linear dan rbf, dari kedua kernel ini untuk mengklasifikasikan data memiliki nilai yang sama baik, dari pengujian yang dilakukan didapatkan nilai akurasi sebesar 88,1%. Hal ini menunjukkan bahwa penggunaan metode *Support Vector Machine* untuk mengklasifikasikan data berupa teks memiliki akurasi yang cukup baik. Tapi dari penelitian dan pengujian yang dilakukan oleh saya dengan menggunakan metode *Support Vector Machine* dengan kernel linear untuk mengklasifikasikan konten instagram berdasarkan komentar memiliki akurasi yang lebih baik dibandingkan dengan kernel yang lainnya. Dari pengujian yang dilakukan didapatkan nilai akurasi sebesar 96%. Dari hasil tersebut dapat dilihat bahwa metode *Support Vector Machine* bisa digunakan untuk mengklasifikasikan konten Instagram berdasarkan komentar.

