

TUGAS AKHIR

SISTEM PERINGATAN DINI UNTUK MENDETEKSI UJARAN KEBENCIAN PADA KOMENTAR FACEBOOK MENGGUNAKAN SVM TF-IDF

EARLY WARNING SYSTEM TO DETECT HATESPEECH ON FACEBOOK COMMENT USING SVM TF-IDF

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Teknik Informatika



Disusun oleh:

Nama : Adiatma Muthahari
NIM : A11.2014.08469
Program Studi : Teknik Informatika – S1

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO
SEMARANG
Tahun 2018**

PERSETUJUAN SKRIPSI

Nama : Adiatma Muthahari
NIM : A11.2014.08469
Program Studi : Teknik Informatika S1
Fakultas : Ilmu Komputer
Judul Tugas Akhir : SISTEM PERINGATAN DINI UNTUK MENDETEKSI
UJARAN KEBENCIAN PADA KOMENTAR
FACEBOOK MENGGUNAKAN SVM TF-IDF

Tugas Akhir ini telah diperiksa dan disetujui,
Semarang, 15 Oktober 2018

Menyetujui
Pembimbing

Mengetahui
Dekan Fakultas Ilmu Komputer

Fahri Firdausillah S.Kom,M.CS

Dr. Abdul Syukur

PENGESAHAN DEWAN PENGUJI

Nama : Adiatma Muthahari
NIM : A11.2014.08469
Program Studi : Teknik Informatika
Fakultas : Ilmu Komputer
Judul Tugas Akhir : SISTEM PERINGATAN DINI UNTUK MENDETEKSI
UJARAN KEBENCIAN PADA KOMENTAR
FACEBOOK MENGGUNAKAN SVM TF-IDF

Tugas Akhir ini telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang tugas akhir tanggal 15 Oktober 2017. Menurut pandangan kami, tugas akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugrahan gelar Sarjana Komputer (S.Kom)

Semarang, 15 Oktober 2018

Dewan Penguji :

Ardytha Luthfiarta, M.Kom

Anggota

Adhitya Nugraha, S.Kom, M.CS

Anggota

Dr Pulung Nurtantio Andono, S.T, M.Kom

Ketua Penguji

PERNYATAAN KEASLIAN SKRIPSI

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan di bawah ini, saya :

Nama : Adiatma Muthahari

NIM : A11.2014.08469

menyatakan bahwa karya ilmiah saya yang berjudul: “SISTEM PERINGATAN DINI UNTUK MENDETEKSI UJARAN KEBENCIAN PADA KOMENTAR FACEBOOK MENGGUNAKAN SVM TF-IDF” merupakan karya asli saya (kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya). Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar saya beserta hak dan kewajiban yang melekat pada gelar tersebut. Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada tanggal : 15 Oktober 2018

Yang menyatakan

(Adiatma Muthahari)

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan di bawah ini, saya:

Nama : Adiatma Muthahari

NIM : A11.2014.08469

demi mengembangkan Ilmu Pengetahuan, menyetujui untuk memberikan kepada Universitas Dian Nuswantoro Hak Bebas Royalti Non-Eksklusif (Non-exclusive Royalty-Free Right) atas karya ilmiah saya yang berjudul:

“SISTEM PERINGATAN DINI UNTUK MENDETEKSI UJARAN KEBENCIAN PADA KOMENTAR FACEBOOK MENGGUNAKAN SVM TF-IDF” beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Dian Nuswantoro berhak untuk menyimpan, mengcopy ulang (memperbanyak), menggunakan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Dian Nuswantoro, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian surat pernyataan ini saya buat dengan sebenarnya

Dibuat di : Semarang

Pada Tanggal : 22 Oktober 2018

Yang menyatakan

Adiatma Muthahari

UCAPAN TERIMA KASIH

Dengan memanjatkan puji syukur kehadiran Allah SWT. Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya kepada penulis sehingga laporan tugas akhir dengan judul SISTEM PERINGATAN DINI UNTUK MENDETEKSI UJARAN KEBENCIAN PADA KOMENTAR FACEBOOK MENGGUNAKAN SVM TF-IDF dapat penulis selesaikan sesuai dengan rencana karena dukungan dari berbagai pihak yang tidak ternilai besarnya. Oleh karena itu penulis menyampaikan terimakasih kepada :

1. Prof., Dr., Ir. Edi Noersasongko, M.Kom, selaku Rektor Universitas Dian Nuswantoro Semarang.
2. Dr. Drs Abdul Syukur, MM selaku Dekan Fasilkom.
3. Heru Agus Santoso, P.hD, selaku Ka. ProgdI Teknik Informatika.
4. Fahri Firdausillah S.Kom,M.CS selaku pembimbing tugas akhir yang memberikan ide penelitian, memberikan informasi referensi yang penulis butuhkan dan bimbingan yang berkaitan dengan penelitian penulis.
5. Dosen-dosen pengampu di Fakultas Ilmu Komputer Teknik Informatika Universitas Dian Nuswantoro Semarang yang telah memberikan ilmu dan pengalamannya masing-masing, sehingga penulis dapat mengimplementasikan ilmu yang telah disampaikan.
6. Kedua orang tua dan keluarga yang memberi semangat, doa dan motivasi sehingga dapat menyelesaikan penelitian ilmiah.
7. Khabib, Andre, Cahyo, Muqaffi, Alif, Gilang, Febi dan Fariz yang selalu mendukung, menghibur dan siap mendengarkan suka cita penelitian ini.
8. Eri, Novan, Edi dan Mas Felix serta teman-teman LABDAS yang selalu mendukung dan membantu dalam penyelesaian penelitian ini.
9. Semua pihak yang tidak dapat disebutkan satu persatu. Terima kasih atas segala bantuan dan dukungannya.

Semoga Allah SWT memberikan balasan yang lebih besar kepada beliau-beliau, dan pada akhirnya penulis berharap bahwa penulisan laporan tugas akhir ini dapat bermanfaat dan berguna sebagaimana fungsinya.

Semarang, 15 Oktober 2018

Penulis

ABSTRAK

Jejaring sosial telah menjadi hal yang sangat erat dalam kehidupan masyarakat sebagai sebuah media untuk berkomunikasi dan bersosialisasi. Kemampuan jejaring sosial yang mampu menghubungkan banyak pengguna, menciptakan begitu banyak interaksi yang tak jarang menimbulkan permasalahan. Salah satu permasalahan tersebut adalah munculnya kiriman yang berisi ujaran kebencian. Ujaran kebencian menjadi salah satu masalah yang dihadapi oleh banyak penyedia layanan jejaring sosial. Penyedia layanan jejaring sosial berusaha menanggulangi masalah tersebut dengan membuat sistem untuk mendeteksi ujaran kebencian. Namun, dengan jumlah konten ujaran kebencian yang sangat besar sistem tersebut dinilai masih belum efektif. Pada penelitian ini peneliti berusaha untuk membuat sebuah sistem deteksi dini ujaran kebencian yang menggunakan SVM-TFIDF. Hasilnya model svm yang menggunakan kernel linear dapat menentukan suatu teks dalam waktu rata-rata 226 milidetik dengan tingkat akurasi sebesar 70,874% dibanding dengan untuk model yang menggunakan kernel rbf yang membutuhkan waktu 530,8 dengan tingkat akurasi sebesar 66,99%.

Kata kunci : jejaring sosial ,sistem peringatan dini, ujaran kebencian, SVM, TFIDF

DAFTAR ISI

JUDUL	i
PERSETUJUAN SKRIPSI	ii
PENGESAHAN DEWAN PENGUJI.....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
UNTUK KEPENTINGAN AKADEMIS	v
UCAPAN TERIMA KASIH	vi
ABSTRAK	viii
DAFTAR ISI	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Tinjauan Studi	6
2.2 Tinjauan Pustaka	7
2.2.1 Ujaran Kebencian	7
2.2.2 Sistem Peringatan Dini	9
2.2.3 Text Mining.....	11
2.2.4 Facebook Graph API	12
2.2.5 WebExtension	15
2.2.6 Term Frequency-Inverse Document Frequency.....	17

2.2.7	Support Vector Machine	18
2.3	Kerangka Pemikiran	21
BAB III	METODE PENELITIAN	22
3.1	Langkah-Langkah Penelitian.....	22
3.1.1	Pemilihan Topik Permasalahan	22
3.1.2	Pemilihan Alternatif Solusi	22
3.1.3	Pemilihan Teknik Penyelesaian Masalah.....	23
3.2	Instrumen Penelitian	24
3.2.1	Instrumen Pengumpulan Data	24
3.2.2	Instrumen Analisis Data.....	24
3.2.3	Instrumen Pengembangan Sistem.....	25
3.3	Metode yang Diusulkan	27
3.3.1	Pengumpulan Data.....	28
3.3.2	Melakukan Preproses Terhadap Data	28
3.3.3	Pemodelan menggunakan SVM	29
3.3.4	Implementasi Model Pada Mesin Filter.....	30
3.3.5	Evaluasi dan pengujian	30
3.4	Evaluasi dan Validasi.....	30
BAB IV	ANALISIS DAN PERANCANGAN SISTEM	34
4.1	Analisis Kebutuhan.....	34
4.1.1	Analisis Kebutuhan Data	34
4.1.2	Analisis Kebutuhan Pengguna	34
4.2	Desain Sistem	36
4.2.1	Use Case.....	36
4.2.2	Sequence Diagram.....	37
4.2.3	Class Diagram	46
4.3	Desain Antarmuka Sistem.....	47
4.4	Algoritma klasifikasi	55

4.4.1	Stemming dan Stopword-removal	55
4.4.2	Term weighting TF-IDF	56
4.4.3	Training dan klasifikasi SVM	58
BAB V	HASIL DAN PENGUJIAN.....	64
5.1	Source Code	64
5.1.1	library SVM	64
5.1.2	library TF-IDF	72
5.1.3	library stemming.....	74
5.2	Antarmuka Sistem	75
5.3	Pengujian Sistem	82
5.3.1	Unit Test.....	82
5.3.2	Pengujian confusion matrix	87
5.3.3	Performance Test.....	91
BAB VI	KESIMPULAN DAN SARAN	95
6.1	Kesimpulan	95
6.2	Saran	95
DAFTAR PUSTAKA	96

DAFTAR GAMBAR

Gambar 2.1. alur sistem peringatan dini	10
Gambar 2.2. Proses Pencarian pengetahuan untuk text mining	12
Gambar 2.3 struktur pengaya menggunakan webextension.....	16
Gambar 2.4. gambaran klasifikasi menggunakan support vector machine.....	18
Gambar 3.1 model pengembangan perangkat lunak waterfall.....	25
Gambar 3.2 Metode yang diusulkan.....	27
Gambar 4.1 Gambar Use Case Diagram.....	36
Gambar 4.2 Sequence Diagram login	38
Gambar 4.3 Sequence diagram untuk import dataset	39
Gambar 4.4 Sequence Diagram untuk proses hapus dataset.....	40
Gambar 4.5 Sequence Diagram untuk proses Training data.....	41
Gambar 4.6 Sequence Diagram untuk pengujian dataset	42
Gambar 4.7 Sequence Diagram untuk proses pengujian teks	43
Gambar 4.8 Sequence Diagram untuk menampilkan peringatan	44
Gambar 4.9 Sequence diagram untuk edit dan mengaktifkan model	45
Gambar 4.10 Sequence diagram untuk proses hapus model.....	45
Gambar 4.11 Gambar class diagram sistem yang akan dibuat.....	46
Gambar 4.12 Gambar desain halaman dataset	48
Gambar 4.13 Gambar Desain halaman simpan dataset	48
Gambar 4.14 Gambar desain tampilan hapus dataset.....	49
Gambar 4.15 Gambar desain halaman Training.....	50
Gambar 4.16 Gambar desain halaman simpan model	50
Gambar 4.17 Gambar Desain Halaman model.....	51
Gambar 4.18 Gambar Desain tampilan edit model	51
Gambar 4.19 Gambar desain tampilan hapus model	52
Gambar 4.20 Desain Halaman Testing	52

Gambar 4.21 Desain tampilan uji dataset	53
Gambar 4.22 Desain halaman uji teks	53
Gambar 4.23 Gambar desain halaman uji dataset	54
Gambar 4.24 Desain lanjutan halaman uji dataset	54
Gambar 4.25 Desain tampilan plugin deteksi	55
Gambar 5.1 Tampilan Halaman Login	76
Gambar 5.2 Tampilan dashboard.....	76
Gambar 5.3 Tampilan Halaman Dataset.....	77
Gambar 5.4 Tampilan simpan dataset.....	77
Gambar 5.5 Tampilan halaman training	78
Gambar 5.6 Tampilan simpan model.....	78
Gambar 5.7 tampilan halaman uji teks	79
Gambar 5.8 tampilan uji dataset.....	79
Gambar 5.9 Tampilan halaman model.....	80
Gambar 5.10 Tampilan edit model	81
Gambar 5.11 Tombol plugin	81
Gambar 5.12 tampilan dialog plugin	82
Gambar 5.13 diagram hasil pengujian pada model berkernel rbf dengan parameter C yang berbeda	88
Gambar 5.14 diagram hasil pengujian pada model berkernel linear dengan parameter C yang berbeda.....	88

DAFTAR TABEL

Tabel 3.1 contoh tabel confussion matrix	30
Tabel 3.2 Contoh test case untuk fungsi cek prima	32
Tabel 3.3 Tabel hasil unit testing	32
Tabel 4.1 tabel rincian dataset	34
Tabel 4.2 Tabel kebutuhan pengguna	35
Tabel 4.3 Contoh Data klasifikasi	56
Tabel 4.4 hasil stemming dan stopwords-removal pada data contoh.....	56
Tabel 4.5 Tabel perhitungan TF	57
Tabel 4.6 Tabel perhitungan IDF dan TFIDF	58
Tabel 4.7 hasil perhitungan alpha.....	62
Tabel 5.1 tabel spesifikasi fungsi train	82
Tabel 5.2 tabel test case untuk fungsi train	84
Tabel 5.3 Tabel hasil unit test testcase 1.....	85
Tabel 5.4 tabel hasil unit testing untuk tast case 2	86
Tabel 5.5 tabel hasil unit testing untuk testcase3	86
Tabel 5.6 tabel hasil unit testing untuk testcase4	86
Tabel 5.7 tabel hasil confussion matrix menggunakan dataset 9_lin_C10.....	89
Tabel 5.8 tabel hasil confussion matrix menggunakan dataset 9_rbf_C50.....	89
Tabel 5.9 tabel hasil pengujian menggunakan dataset testing	90
Tabel 5.10 model yang akan diujikan.....	91
Tabel 5.11 spesifikasi notebook yang digunakan untuk pengujian.....	92
Tabel 5.12 tabel hasil pengujian performance test	92

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, Internet telah menjadi kebutuhan bagi masyarakat. Hampir seluruh aspek kehidupan dalam masyarakat tidak terlepas dari pengaruh internet. Diawal kehadirannya internet dimanfaatkan hanya untuk mencari dan bertukar informasi. Seiring dengan perkembangan jaman, pemanfaatan internet juga ikut berkembang. Kini, Internet telah dimanfaatkan untuk berbagai macam hal mulai dari sebagai media hiburan, berbelanja, hingga media komunikasi yang kini telah mengubah cara masyarakat dalam berinteraksi sosial.

Perubahan sosial yang kini terjadi pada masyarakat dipengaruhi oleh munculnya media sosial. Media sosial sendiri merupakan sebuah istilah yang merujuk kepada sebuah teknologi yang dapat membuat penggunanya menciptakan dan membagikan informasi, ide, dan konten lain melalui jaringan dan komunitas virtual. Banyak media sosial yang telah populer di Indonesia mulai dari *facebook*, *instagram*, *twitter*, *whatsapp*, dll. Pada media sosial tersebut pengguna dapat saling berkomunikasi dengan berbagai cara mulai dari saling berkirim pesan, mengirim gambar hingga mengirim video. Di Indonesia pengguna media sosial telah mencapai 79 juta pada tahun 2016 dan akan terus bertambah. Sementara itu, *facebook* menjadi platform sosial media yang paling banyak digunakan oleh masyarakat indonesia[1].

Sebagai layanan jejaring sosial yang paling banyak digunakan oleh masyarakat Indonesia, *facebook* telah memiliki sebanyak 115 juta pengguna di indonesia [2]. Dengan jumlah pengguna yang sangat banyak, Jumlah konten dan interaksi antar pengguna yang dihasilkan pun juga sangat besar. Dengan berbagai macam jenis konten yang dihasilkan dan topik yang dibahas oleh pengguna, *facebook* berhasil mendapatkan keuntungan yang besar dari iklan. *Facebook* menjadi media yang efektif untuk memasarkan produk hingga mempromosikan kegiatan politik. Namun,

disisi lain hal tersebut mendatangkan permasalahan salah satunya adalah munculnya konten negatif berupa *hate speech* (ujaran kebencian).

Ujaran kebencian menjadi salah satu masalah yang dihadapi oleh banyak penyedia layanan jejaring sosial tidak terkecuali *facebook*. Sampai saat ini *facebook* masih terus berupaya menangani ujaran kebencian di banyak negara dimana jejaring sosial tersebut beroperasi. Sementara itu, Istilah ujaran kebencian mulai sering terdengar di kalangan masyarakat Indonesia pada awal tahun 2017. Hal ini tidak terlepas dari beberapa faktor salah satunya adalah faktor politik. Faktor politik yang dimaksud adalah pemilihan kepala daerah secara serentak diberbagai daerah yang diselenggarakan pemerintah Indonesia pada awal tahun 2017. Hal tersebut menyebabkan, *facebook* menjadi sebuah media untuk kampanye serta menjadi tempat mempromosikan calon kepala daerah. Tidak hanya menjadi media promosi, *facebook* juga menjadi media untuk menyerang calon lain. Mulai dari kiriman berupa kekurangan dari calon lain, kiriman berupa berita bohong (*hoax*) hingga kiriman yang mendiskriminasi golongan tertentu digunakan untuk menyerang lawan politik mereka. Hal ini menimbulkan banyak perdebatan antar pendukung calon kepala daerah. Tak jarang berakhir dengan saling memaki antar pendukung atau dibuatnya kiriman yang berisi ujaran kebencian.

Dengan banyaknya konten ujaran kebencian yang muncul di jejaring sosial *facebook*, *facebook* memerlukan sistem yang dapat menanggulangi ujaran kebencian. Di Amerika dan Eropa *facebook* telah mencoba menerapkan sistem tersebut untuk menanggulangi konten ujaran kebencian. Sementara itu di Indonesia, *facebook* sendiri menerapkan sistem yang serupa dimana *facebook* akan menyaring konten yang telah dikirim pengguna yang tak sesuai dengan ketentuan *facebook*. Namun, sistem tersebut dinilai masih belum efektif[3][4]. Masih banyak terdapat kiriman yang meresahkan dan dianggap sebagai ujaran kebencian yang muncul pada sosial media tersebut[5]. Hal ini disebabkan oleh konten ujaran kebencian yang telah dikirim, terlanjur tersebar secara luas sehingga membuat resah pengguna yang lain. Oleh karena itu, diperlukan sebuah sistem yang dapat memberikan peringatan jika sebuah konten berisi ujaran kebencian sebelum dikirimkan. Untuk

dapat menentukan sebuah kiriman berupa tulisan yang berisi ujaran kebencian secara otomatis, diperlukan sebuah proses yang disebut *text classification*.

Pada penelitian sebelumnya, William Warner dan Julia Hirschberg mencoba untuk mendeteksi ujaran kebencian yang ada pada internet menggunakan algoritma *Support Vector Machine*(SVM) yang menghasilkan akurasi rata-rata sebesar 0.94 dan nilai f-measure mendekati f-measure dari annotator sebesar 0.63[6]. Pate Burnap dan Matthew L Williams mencoba menggunakan metode n-gram yang digabungkan dengan beberapa algoritma klasifikasi seperti *Bayesian Linear Regression* (BLR), *decision tree*, dan SVM. Hasilnya, ketiga algoritma tersebut menghasilkan kinerja yang serupa dan penggunaan n-gram mampu meningkatkan tingkat akurasi[7]. Sementara itu, Nurirwan Saputra dkk melakukan analisis sentimen terhadap presiden Joko Widodo menggunakan metode SVM dan *Naive Bayes* dengan data yang ditokenisasi dalam bentuk N-gram, unigram, digram dan trigram dan telah di normalisasi serta di stemming sebelumnya. Hasilnya, Klasifikasi menggunakan SVM menghasilkan akurasi tertinggi pada dua pengujian yang dilakukan yaitu sebesar 88,70% untuk pengujian menggunakan data yang sudah dinormalisasi dan 89,26% untuk pengujian menggunakan data yang telah dinormalisasi dan stemming[8]. Lalu, Syahfitri Kartika Lidya dkk juga melakukan sentiment analisis terhadap beberapa tokoh nasional di Indonesia. Mereka menggunakan SVM dan K-NN sebagai algoritma pengklasifikasian. Hasilnya SVM menghasilkan akurasi yang lebih besar dibanding K-NN yaitu 67,90% dibanding 60,30%[9].

Sementara itu, Pinkesh Badjatiya dkk mengusulkan metode dengan 3 tahapan untuk mendeteksi ujaran kebencian yaitu dengan menggunakan *random word embedding*, proses pembelajaran *deep learning*, kemudian klasifikasi menggunakan *Gradient Boosted Decision Tree*(GBDT). Selain itu, metode yang diusulkan juga dibandingkan dengan metode klasifikasi lain seperti BLR, GBDT, SVM serta metode yang menggunakan *deep learning* saja. Hasilnya, metode yang diusulkan mampu mengungguli seluruh metode yang sudah ada. Meskipun begitu percobaan yang menggunakan SVM+TF-IDF dapat mengungguli beberapa metode

yang menggunakan *deep learning* saja. Selain itu, SVM+TF-IDF juga memiliki hasil recall dan F1 yang lebih tinggi dibanding metode klasifikasi BLR dan GBDT[10]. Dengan tingkat akurasi yang cukup bagus dan waktu pembelajaran yang relatif singkat, SVM menjadi metode yang baik untuk diaplikasikan kedalam sistem peringatan dini deteksi ujaran kebencian.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan diatas diketahui bahwa terdapat beberapa teknik untuk mengklasifikasi ujaran kabencian. Namun, teknik klasifikasi tersebut belum diterapkan ke dalam sistem yang sebenarnya. Oleh karena itu didapatkan rumusan masalah sebagai berikut

- 1 Bagaimana cara menerapkan SVM untuk melakukan peringatan secara dini ujaran kebencian?
- 2 Berapa akurasi yang dihasilkan oleh deteksi ujaran kebencian menggunakan SVM?

1.3 Batasan Masalah

Pada penelitian ini, terdapat beberapa batasan terhadap masalah yang akan diselesaikan supaya penilitian dapat dilakukan secara lebih terarah dan lebih dapat dipertanggung jawabkan. Batasan-batasan yang dimaksud antara lain :

1. Dataset berupa kumpulan teks komentar yang dikirimkan pengguna dalam bahasa Indonesia.
2. Dataset didapat dari hasil scrapping komentar publik yang dikirimkan pengguna pada halaman *facebook* Kompas.com, detikcom, dan tvOneNews.
3. Data latih yang digunakan ditokenisasi ke dalam bentuk unigram.
4. Proses klasifikasi yang digunakan yaitu preproses menggunakan stemming, ekstraksi fitur menggunakan TFIDE, dan klasifikasi menggunakan .

1.4 Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan yaitu:

1. Menerapkan metode klasifikasi SVM untuk melakukan peringatan secara dini ujaran kebencian.
2. Mengukur akurasi dari hasil deteksi dini ujaran kebencian menggunakan metode klasifikasi SVM.

1.5 Manfaat Penelitian

Diharapkan penelitian ini dapat memberikan manfaat sebagai berikut :

1. Mencegah kiriman komentar berisi ujaran kebencian yang dapat meresahkan masyarakat.
2. Menciptakan jejaring sosial yang bebas dari ujaran kebencian untuk masyarakat
3. Masyarakat dapat menggunakan jejaring sosial dengan lebih baik dan bijaksana.

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Studi

Dalam penelitian ini digunakan beberapa rujukan berdasarkan studi yang pernah dilakukan sebelumnya dengan topik identifikasi ujaran kebencian dan penggunaan SVM sebagai metode klasifikasi teks.

Pertama, William Warner dan Julia Hirschberg mencoba untuk mendeteksi ujaran kebencian yang ada pada internet menggunakan algoritma SVM. Hasilnya, didapatkan akurasi rata-rata sebesar 0.94 dan nilai f-measure mendekati f-measure dari annotator sebesar 0.63[6].

Berikutnya, Pate Burnap dan Matthew L Williams mencoba menggunakan metode n-gram yang digabungkan dengan beberapa algoritma klasifikasi seperti BLR, *decision tree*, dan SVM. Hasilnya, ketiga algoritma tersebut menghasilkan kinerja yang serupa dan penggunaan n-gram mampu meningkatkan tingkat akurasi[7].

Kemudian, Pinkesh Badjatiya dkk mengusulkan metode dengan 3 tahapan untuk mendeteksi ujaran kebencian yaitu dengan menggunakan *random word embedding*, proses pembelajaran *deep learning*, kemudian klasifikasi menggunakan GBDT. Selain itu, metode yang diusulkan juga dibandingkan dengan metode klasifikasi lain seperti BLR, GBDT, SVM serta metode yang menggunakan *deep learning* saja. Hasilnya, metode yang diusulkan mampu mengungguli seluruh metode yang sudah ada. Meskipun begitu percobaan yang menggunakan SVM+TF-IDF dapat mengungguli beberapa metode yang menggunakan *deep learning* saja. Selain itu, SVM+TF-IDF juga memiliki hasil recall dan F1 yang lebih tinggi dibanding metode klasifikasi BLR dan GBDT[10].

Nurirwan Saputra dkk melakukan analisis sentimen terhadap presiden Joko Widodo menggunakan metode SVM dan *Naive Bayes* dengan data yang ditokenisasi dalam bentuk N-gram, unigram, digram dan trigram dan telah di normalisasi serta di stemming sebelumnya. Hasilnya, Klasifikasi menggunakan

SVM menghasilkan akurasi tertinggi pada dua pengujian yang dilakukan yaitu sebesar 88,70% untuk pengujian menggunakan data yang sudah dinormalisasi dan 89,26% untuk pengujian menggunakan data yang telah dinormalisasi dan stemming[8].

Lalu, Syahfitri Kartika Lidya dkk juga melakukan sentiment analisis terhadap beberapa tokoh nasional di Indonesia. Mereka menggunakan SVM dan K-NN sebagai algoritma pengklasifikasian. Hasilnya SVM menghasilkan akurasi yang lebih besar dibanding K-NN yaitu 67,90% dibanding 60,30%[9].

2.2 Tinjauan Pustaka

2.2.1 Ujaran Kebencian

Ujaran Kebencian secara luas dapat diartikan sebagai ujaran, pernyataan ataupun hasutan yang ditujukan untuk mendiskriminasi atau melakukan kekerasan kepada seseorang atau kelompok tertentu, karena latar belakang ras, etnis, atau agama, bahkan orientasi seksual[11].

Sementara itu menurut Surat Edaran Kapolri mengenai penanganan Ujaran Kebencian, Ujaran Kebencian merupakan tindak pidana yang diatur dalam Kitab Undang-Undang Hukum Pidana(KUHP) dan ketentuan pidana lain di luar KUHP, yang diantaranya berbentuk:

1. penghinaan
2. pencemaran nama baik
3. penistaan
4. perbuatan tidak menyenangkan
5. memprovokasi
6. menghasut
7. penyebaran berita bohong

yang memiliki tujuan atau berdampak pada tindak diskriminasi, kekerasan, penghilangan nyawa, dan/atau konflik sosial[12].

Selain pengertian diatas surat edaran tersebut juga menjelaskan mengenai aspek yang dimuat dalam sebuah ujaran yang masuk kedalam kategori ujaran kebencian seperti aspek agama atau antargolongan, media yang digunakan dalam melakukan ujaran kebencian. Kemudian berdasarkan penelitian yang dilakukan oleh Linawati[13], dari bentuk-bentuk ujaran kebencian yang dijelaskan pada surat edaran tersebut, terdapat beberapa bentuk ujaran kebencian yang banyak ditemukan di media daring seperti :

1. Penghinaan

Penghinaan merupakan perbuatan menyerang kehormatan atau nama baik seseorang atau kelompok tertentu dengan tujuan supaya pihak yang diserang merasa malu atau tersinggung . kalimat berisi penghinaan cukup mudah dikenali. Penghinaan dapat dikenali dengan adanya kata kasar maupun makian yang sifatnya menghina dalam suatu kalimat, seperti anjing, bajingan, goblok.

Contoh:

“Restu Amri kalau komentar yang berkualitas lah om. Goblok dipelihara!”

2. Pencemaran Nama Baik

Pencemaran nama baik memiliki kesamaan dengan penghinaan yaitu menyerang kehormatan nama baik seseorang. Bedanya, pencemaran nama baik lebih berisi tuduhan terhadap pihak tertentu dimana hal yang dituduhkan tersebut merupakan hal yang tidak benar dan tidak sesuai dengan fakta yang ada.

Contoh:

“Anies mau ngadu domba dosen/guru dengan Ahok. Basi.”

3. Penistaan

Penistaan merupakan tuturan yang merendahkan seseorang atau kelompok tertentu. Penistaan merupakan salah satu bentuk ujaran kebencian yang banyak menyinggung tentang aspek agama, aliran agama, gender dan orientasi seksual. Tuturan yang digolongkan kedalam penistaan berisi aib atau hal yang memalukan untuk diketahui oleh umum.

Contoh:

“Ahok nggak perlu dukungan orang munafik tukang selingkuh seperti pak haji”

4. Perbuatan Tidak Menyenangkan

Perbuatan tidak menyenangkan adalah tindakan mengganggu kenyamanan dan keamanan seseorang atau kelompok tertentu. Tuturan yang digolongkan ke dalam perbuatan tidak menyenangkan biasanya berisi ancaman, paksaan, kekerasan, maupun sumpah.

Contoh:

“Keluar bui mampusin ni orang.”

5. Memprovokasi

Memprovokasi merupakan tindakan berupa tuturan maupun ujaran yang disampaikan secara berapi-api dengan maksud menciptakan keadaan yang semakin tidak kondusif, membuat orang mau melakukan tindak pidana maupun bentuk tindakan lain.

Contoh:

“Penjarakan penista agama”

6. Menghasut

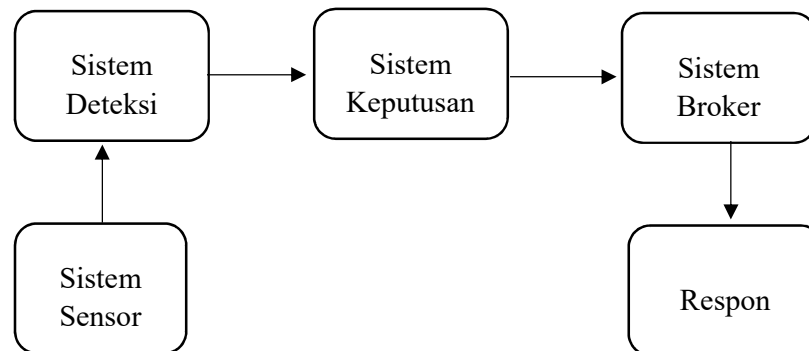
Tuturan yang berisi hasutan dilakukan dengan maksud yang hampir sama dengan provokasi. Namun, menghasut tidak disampaikan secara berapi-api.

Contoh:

“Waktu ketok kepala dan jambak anak tak bersalah sampai meninggal nggak ngompol kau, Yus? Terkutuk kau!, Nggak punya hati kau! lebih dari binatang kau!”

2.2.2 Sistem Peringatan Dini

Menurut Waidyanatha sistem peringatan dini merupakan rangkaian sistem komunikasi informasi yang terdiri dari subsistem sensor, deteksi, keputusan, serta broker yang berurutan, bekerja untuk memprediksi gangguan yang tidak diinginkan yang berpotensi dapat mengganggu stabilitas dunia nyata[14]. Diharapkan dengan adanya prediksi tersebut, dapat dilakukan penanggulangan secara efektif dan *realtime*.



Gambar II.1. alur sistem peringatan dini

Sistem Sensor merupakan bagian dimana sistem menerima informasi masukan. Sumber dan bentuk masukan sangat tergantung dengan domain dari sistem peringatan dini itu sendiri. Sebagai contoh, saat ini banyak pemanfaatan media sosial sebagai sumber masukan untuk penanggulangan bencana alam, militer, krisis ekonomi dan lain sebagainya.

Sistem Deteksi merupakan bagian dimana sistem mengekstraksi data dari kumpulan data yang didapat oleh sensor. Sebagai contoh, sistem peringatan gempa menggunakan akan membaca data dari seismograf dan mendeteksi apakah dari data seismograf tersebut menunjukkan terjadi gempa serta sistem dapat membedakan gempa sesungguhnya dengan gempa yang disebabkan oleh ledakan.

Sistem Keputusan merupakan bagian dimana sistem akan menentukan apakah sistem akan memberikan respon terhadap suatu gangguan atau tidak.

Sistem Broker merupakan sebuah sistem yang memiliki tugas untuk mengirimkan hasil keputusan ke bagian sistem respon. Sistem ini akan merubah data hasil keputusan ke dalam bentuk standard yang dapat di terima oleh sistem respon dan menentukan pesan mana yang akan dikirim.

Sistem respon merupakan keluaran dari sistem peringatan dini yang biasanya berbentuk pesan peringatan. Dalam pesan peringatan tersebut selain memberitahukan tentang resiko yang akan terjadi, diberikan juga instruksi berikutnya guna menaggulangi dampak gangguan.

2.2.3 Text Mining

Untuk dapat mengklasifikasikan suatu teks kita memerlukan sebuah teknik yang dapat mengambil informasi pada data yang berupa teks. Oleh karena itu, kita memerlukan teknik yang disebut *text mining*. *Text mining* merupakan sebuah pengembangan teknik *data mining* yang mana *text mining* dapat mencari pola tertentu pada kumpulan data tidak terstruktur (text) dalam suatu dokumen. Secara keseluruhan text mining melibatkan *data mining*, *machine learning*, *information retrieval*, dan *natural language processing*. Hasil dari sebuah proses text mining adalah sebuah pengetahuan berupa pola yang dapat diterapkan untuk menyelesaikan masalah.

Text mining memiliki alur dan proses yang hampir serupa dengan data mining pada umumnya. Yang pertama adalah *data preparation*, pada tahapan ini data yang digunakan akan dipilah terlebih dahulu. Selain itu pada tahapan ini dilakukan juga proses *preprocessing* dokumen (kategorisasi teks, ekstraksi informasi, ekstraksi istilah). terdapat beberapa hal yang dilakukan dalam preprocessing dokumen sebagai berikut, diantaranya :

1. Tokenisasi

Tokenisasi merupakan proses memecah kalimat menjadi kumpulan token atau kata. proses ini akan menghilangkan beberapa karakter seperti “”(spasi), “,”, “.” dan tanda baca lainnya.

2. *Stop word removal*

Proses ini akan menghilangkan beberapa kata yang tidak berpengaruh seperti “di”, “ke”, “yang”, dan lain sebagainya.

3. *Stemming*

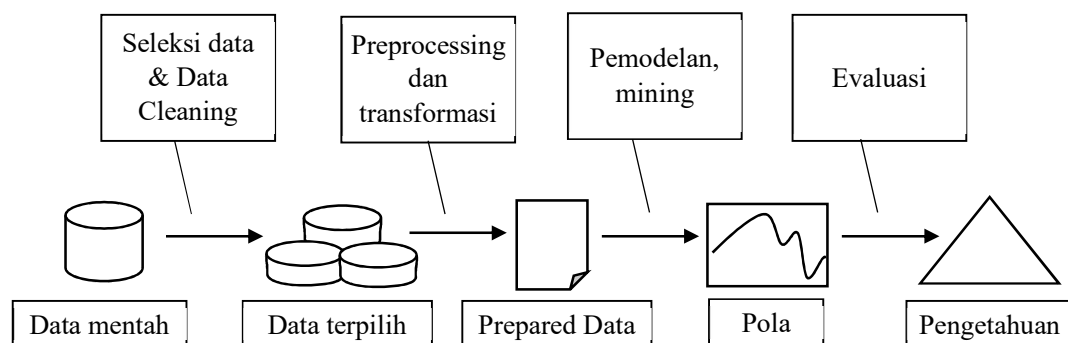
Proses ini akan merubah suatu token ke dalam bentuk dasarnya. Contohnya “menyapu” menjadi “sapu”. Bila tidak ditemukan *rule* yang mengatur bentuk dasar dari sebuah token maka token tersebut tidak akan berubah.

4. *Term weighting*

Term weighting atau pembobotan adalah sebuah proses untuk menentukan bobot dari setiap kata. pada proses ini data yang sebelumnya berupa data yang tidak terstruktur sudah berubah menjadi data *intermediate* yang lebih terstruktur.

Tahap berikutnya adalah pemodelan, pada tahap ini data yang telah di *preprocessing* telah menjadi data *intermediate* yang lebih terstruktur. lalu data *intermediate* tadi dianalisa menggunakan teknik analisa representasi *intemediate* (seperti analisis distribusi, *clustering*, analisis tren, klasifikasi dan *association rules*).

Tahap yang terakhir adalah evaluasi, pada tahap ini dilakukan visualisasi dari hasil pemodelan.



Gambar II.2. Proses Pencarian pengetahuan untuk text mining

2.2.4 Facebook Graph API

Graph API atau API Graf merupakan Application Programming Interface yang disediakan oleh facebook sebagai cara utama untuk memperoleh data di dalam dan di luar platform Facebook. API Graf adalah API berbasis HTTP yang dapat digunakan oleh aplikasi untuk mengkueri data secara terprogram, memposting

cerita baru, mengelola iklan, mengunggah foto, dan melakukan beragam jenis tugas lainnya. Nama API Graf sendiri bermula dari "graf sosial" yang merupakan representasi informasi yang digunakan di Facebook.

Untuk dapat menggunakan API Graf terdapat beberapa hal yang perlu di siapkan yaitu :

1. HTTP/1.1

Seluruh proses transfer data sesuai dengan protokol HTTP/1.1. Selain itu, seluruh endpoint juga memerlukan HTTPS.

2. URL Host

Untuk menggunakan API Graf, hampir seluruh permintaan akan diteruskan ke URL Host `graph.facebook.com`. Satu-satunya, pengecualian adalah panggilan API untuk video yang melalui URL Host `graph-video.facebook.com`.

3. Token Akses

Sebuah Token Akses diperlukan agar aplikasi dapat mengakses API Graf. Token Akses memiliki dua buah fungsi yaitu :

- Membuat aplikasi dapat menggunakan informasi pengguna tanpa memerlukan sandi
- Membuat pihak facebook dapat mengenali aplikasi yang kita buat, mengenali pengguna yang menggunakan aplikasi, dan mengetahui jenis data yangizinkan aksesnya oleh pengguna.

Seluruh endpoint graph memerlukan token akses. Oleh karena itu, sebuah Token akses harus disertakan dalam setiap permintaan untuk dapat mengakses endpoint graph.

Token Akses ini berjalan sesuai dengan protokol OAuth 2.0. Untuk mendapatkan token akses pengembang dapat melakukan permintaan pada halaman penjelajah graf atau melakukan panggilan OAuth menggunakan API pada SDK yang telah disediakan oleh facebook.

Selain itu, API Graf memiliki aturan khusus mengenai struktur query nya. Berikut adalah struktur dari API graf yang terdiri dari :

1. Node

Merupakan Objek individu seperti pengguna, Foto, Halaman, atau Komentar yang direpresentasikan dalam bentuk ID Object. Contoh panggilan graf untuk dapat membaca node halaman Coca-Cola:

```
GET https://graph.facebook.com/v2.11/820882001277849
```

Permintaan tersebut secara *default* akan mengembalikan respon berisi properti node dalam format JSON sebagai berikut :

```
{
  "name": "Coca-Cola",
  "id": "820882001277849"
}
```

2. Edge

Edge merupakan koleksi objek yang terhubung ke sebuah node. Untuk dapat mengakses sebuah edge, pengembang harus menyertakan ID node diikuti nama edge yang ingin diakses. Terdapat berbagai macam edge yang dapat diakses seperti *feed*, *events*, *friends*, *likes*, dll. Contoh menggunakan *edge feed* untuk memperoleh semua postingan di Halaman Coca-Cola:

```
GET
https://graph.facebook.com/v2.11/820882001277849/feed
```

Berikut adalah respon JSON yang akan didapatkan dari kueri di atas :

```
{
  "data": [
    {
      "created_time": "2017-12-08T01:08:57+0000",
      "message": "Love this puzzle. One of my four coke puzzles",
      "id": "820882001277849_1805191182846921"
    },
    {
      "created_time": "2017-12-07T20:06:14+0000",
      "message": "You need to add grape as a flavor for Coke in your freestyle machines.",
      "id": "820882001277849_1804966026202770"
    },
    {
      "created_time": "2017-12-07T01:29:12+0000",
```

```

      "message": "Plz play the old commercial's with the
polar bears. Would be nice to see them this holiday",
      "id": "820882001277849_1804168469615859"
    }
  ]
}

```

3. Kolom

Kolom menentukan data yang ingin Anda sertakan di dalam tanggapan. Saat mengkueri sebuah node, maka kueri tersebut akan menghasilkan sebuah set kolom secara default (). Namun, pengembang juga dapat menentukan sendiri kolom yang akan dihasilkan dengan menggunakan parameter `fields`. Contoh memperoleh kolom `about`, `fan_count`, dan `website` untuk Halaman Coca-Cola :

```

GET https://graph.facebook.com/v2.11
/820882001277849
?fields=about,fan_count,website

```

Berikut adalah respon JSON yang akan didapatkan dari kueri di atas :

```

{
  "about": "Welcome to the happiest Facebook page on, um,
Facebook.",
  "fan_count": 106714402,
  "website": "http://coca-cola.com",
  "id": "820882001277849"
}

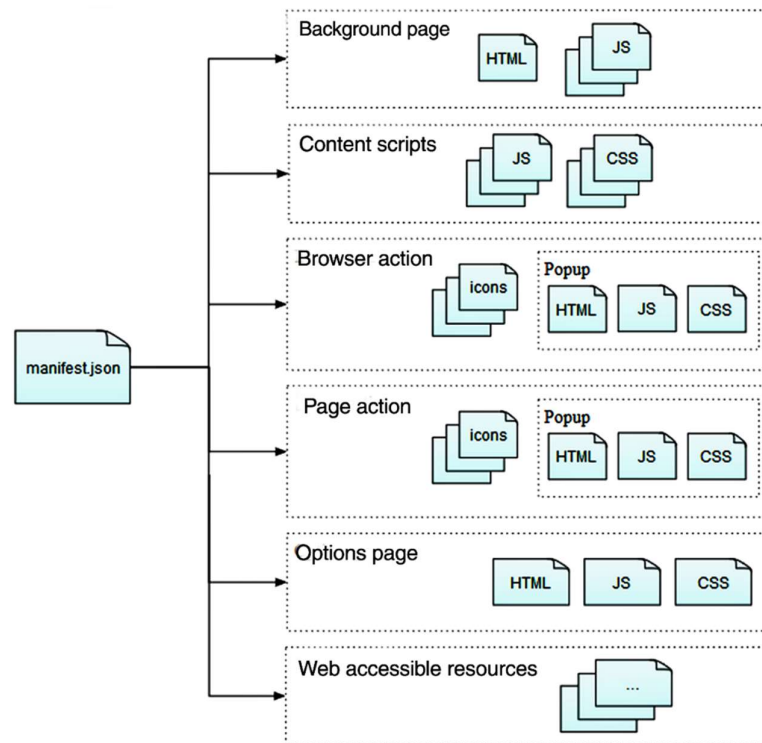
```

Selain untuk membaca informasi dalam node menggunakan metode HTTP GET, API Graf juga dapat digunakan untuk melakukan penerbitan menggunakan metode HTTP POST, penghapusan menggunakan metode HTTP DELETE, serta pembaruan menggunakan metode HTTP UPDATE.

2.2.5 WebExtension

WebExtension Merupakan standar yang digunakan untuk mengembangkan pengaya suatu web browser. WebExtension merupakan standar yang dapat digunakan di berbagai browser (cross-browser) seperti firefox, chrome dan opera. Untuk menambahkan fitur dan fungsi pada browser, WebExtension menggunakan

teknologi yang umum digunakan pada pengembangan sebuah halaman web seperti Javascript, CSS, dan HTML. Selain dapat memanfaatkan teknologi seperti javascript, sebuah pengaya dapat menggunakan akses yang disediakan oleh WebExtension API pada browser.



Gambar II.3 struktur pengaya menggunakan webextension

Sebuah pengaya yang menggunakan WebExtension memiliki anatomi sebagai contoh gambar di atas. Pada sebuah pengaya WebExtension terdapat kumpulan file yang mungkin terdapat pada sebuah paket pengaya sebagai berikut :

1. Manifest

File manifest merupakan file yang wajib ada pada sebuah pengaya. File ini berisi metadata dasar seperti nama pengaya, versi, serta hak akses yang dibutuhkan. File ini juga memetakan file lain yang digunakan oleh pengaya.

2. Background Script

Background script merupakan bagian bengaya yang akan berjalan pada background. Hal ini bertujuan supaya sebuah fungsi yang membutuhkan proses yang cukup lama dapat berjalan secara independen pada bagian background.

3. Browser Action

Browser action mendeskripsikan tampilan pada jendela browser serta aksi yang dilakukan oleh pengaya tersebut.

4. Content Script

Content Script merupakan bagian dimana pengaya dapat mengakses serta melakukan manipulasi halaman web.

5. Web Accessible Resource

Merupakan file resource seperti gambar yang dapat digunakan oleh content script.

2.2.6 Term Frequency-Inverse Document Frequency

Term Frequency(TF) dan *Inverse Document Frequency*(IDF) adalah sebuah algoritma pembobotan yang banyak digunakan dalam *text mining*. TF-IDF ini merupakan algoritma yang cukup sederhana namun efektif untuk menentukan bobot dari sebuah kata yang terdapat pada sebuah dokumen. TF-IDF merupakan perhitungan statistik untuk menentukan seberapa penting suatu kata terhadap suatu dokumen atau *corpus* dengan rumus sebagai berikut :

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

Pada dasarnya TF-IDF adalah hasil perkalian nilai TF serta IDF suatu kata t pada dataset D . Pada rumus diatas t adalah kata yang ingin dihitung bobotnya, d adalah dokumen, dan D merupakan kumpulan dokumen (Dataset). Sementara itu, untuk penelitian ini nilai TF yang digunakan merupakan frekuensi kemunculan suatu kata pada sebuah dokumen.

$$tf(t, d) = \log(1 + f_{t,d})$$

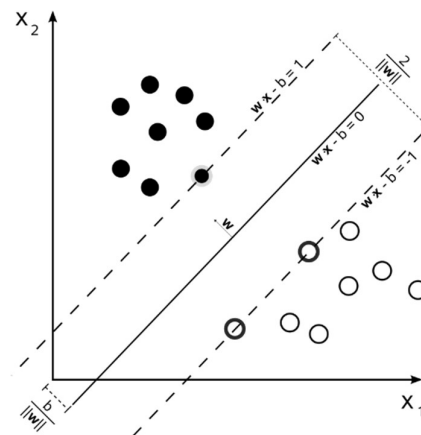
Kemudian, nilai IDF suatu kata didapatkan dari hasil perhitungan logaritma dari jumlah seluruh dokumen berbanding dengan berapa banyak dokumen dimana kata tersebut ditemukan.

$$idf(t, D) = \log \left(1 + \frac{N}{n_t} \right)$$

Dimana N merupakan total dokumen dan n_t merupakan jumlah dokumen dimana kata t ditemukan.

2.2.7 Support Vector Machine

Support Vector Machine merupakan supervised learning model yang digunakan untuk melakukan analisa data dalam proses klasifikasi atau regresi linear. Pada proses klasifikasi metode pembelajaran ini akan memetakan data pada suatu bidang dan membedakan kategori yang terpisah secara jelas dengan jarak yang selebar-lebarnya. Kategori yang berbeda tersebut dipisahkan oleh *hyperplane*. Berikut adalah gambaran klasifikasi menggunakan svm.



Gambar II.4. gambaran klasifikasi menggunakan support vector machine

Hyperplane merupakan sebuah garis atau bidang yang memisahkan 2 buah kategori yang berbeda. Sebuah *hyperplane* dapat kita cari dengan menggunakan rumus sebagai berikut .

$$w \cdot x_i - b = 0 \quad (1)$$

w pada persamaan diatas merupakan vektor yang menentukan bobot. Sementara itu x_i merupakan vektor sampel data dan b merupakan bias. Pada klasifikasi biner, hasil klasifikasi akan menghasilkan true (+1) dan false (-1). Untuk klasifikasi tersebut berlaku rumus sebagai berikut:

$$f(x) = \begin{cases} +1, & \text{jika } w \cdot x_i - b \geq 0 \\ -1, & \text{jika } w \cdot x_i - b \leq 0 \end{cases} \quad (2)$$

Namun, agar hasil klasifikasi menjadi lebih baik kita perlu memaksimalkan nilai margin. *Margin* sendiri merupakan jarak antara titik-titik positif dan negatif terdekat di sekitar *hyperplane*. titik-titik positif dan negatif terdekat di sekitar *hyperplane* ini biasa disebut dengan *support vector*. Untuk menentukan margin dapat digunakan persamaan berikut

$$f(x) = \begin{cases} +1, & \text{jika } w \cdot x_i - b \geq 1 \\ -1, & \text{jika } w \cdot x_i - b \leq -1 \end{cases} \quad (3)$$

Pada persamaan di atas batas atas dan batas bawah tidak lagi 0 seperti persamaan. Hal ini dikarenakan $-1 < w \cdot x_i - b < 1$ menyatakan margin. Persamaan dapat disederhanakan menjadi

$$y_i(w \cdot x_i - b) \geq 1 \quad (4)$$

Pada umumnya dataset yang digunakan dalam klasifikasi teks dapat dipisahkan secara linear. Namun, tidak menutup kemungkinan bahwa dataset yang kita miliki tidak dapat dipisahkan secara linear. Oleh karena itu kita perlu membuat model yang dapat menghiraukan beberapa data tidak sesuai guna menambah akurasi. Untuk melakukan nya kita dapat menambahkan sebuah variabel *slack* (ξ). Teknik ini disebut soft margin.

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i \quad \xi_i > 0 \quad (5)$$

Dari persamaan di atas, agar nilai margin menjadi maksimum kita harus meminimalkan problem sebagai berikut.

$$\min \quad \frac{1}{2} w \cdot w + C \sum_{i=1}^n \xi_i$$

$$s. t. \quad y_i(w \cdot x_i - b) \geq 1 - \xi_i \quad \xi_i > 0 \quad (6)$$

Pada persamaan diatas C merupakan parameter yang menentukan seberapa banyak data yang dibiarkan *misclassify* .Kita dapat mempermudah penyelesaian persamaan di atas dengan mengubahnya kedalam bentuk lagrange sebagai berikut:

$$L(w, \xi, b, \alpha) = \frac{1}{2}w \cdot w + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i [1 - \xi_i - y_i(w \cdot x_i - b)] \quad (7)$$

α_i pada persamaan di atas merupakan lagrange multiplier yang memiliki nilai 0 atau positif. Dari penyelesaian lagrange di atas kita bisa mendapatkan w sebagai berikut.

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (8)$$

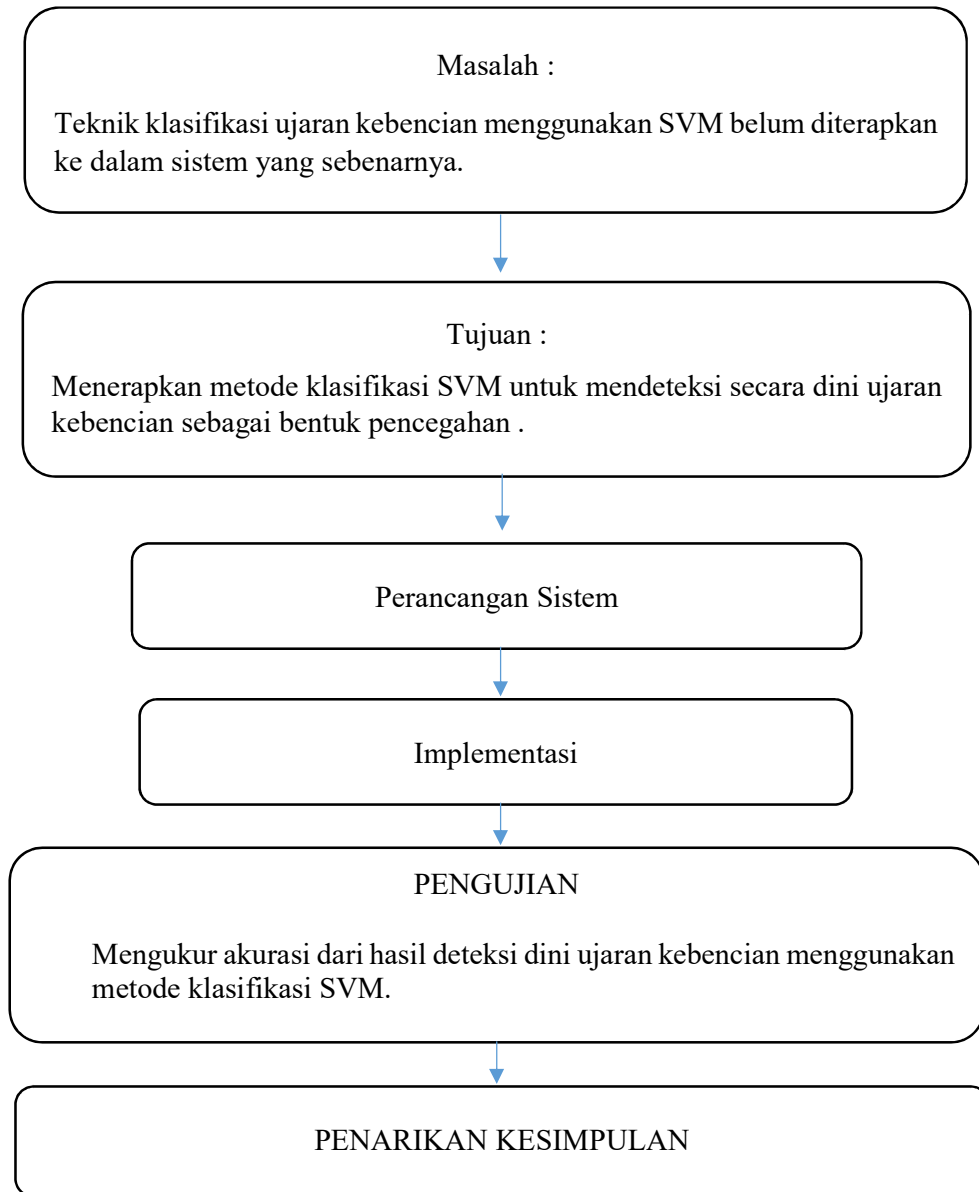
jika $0 < \alpha_i < C$ maka x_i merupakan support vector. kemudian Nilai w dan b yang didapat digunakan pada persamaan (2) untuk proses klasifikasi. Jika nilai w pada persamaan (8) kita substitusikan dengan w yang ada pada persamaan (2) maka didapatkan fungsi objektif sebagai berikut :

$$f(x) = \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle - b \quad (9)$$

Persamaan diatas merupakan fungsi objektif untuk menyelesaikan masalah secara linear. Untuk masalah yang tidak dapat di selesaikan secara linear (non-linear), kita dapat mengganti perhitungan dot product x_i, x dengan sebuah fungsi kernel lain seperti kernel *radial base function* atau *sigmoid*. Fungsi kernel ini dapat digunakan untuk membantu memecahkan masalah non-linear.

2.3 Kerangka Pemikiran

Berdasarkan teori yang telah dipaparkan sebelumnya, penelitian akan dilakukan dengan kerangka pemikiran sebagai berikut:



BAB III

METODE PENELITIAN

3.1 Langkah-Langkah Penelitian

3.1.1 Pemilihan Topik Permasalahan

Pembuatan penelitian yang dilakukan ini diawali dengan pencarian masalah. topik permasalahan didapat dari hal yang dialami dan dirasakan langsung oleh penulis sebelum memutuskan untuk melakukan penelitian mengenai ujaran kebencian. Keputusan untuk melakukan penelitian ini berawal dari ketidaknyamanan yang dirasakan saat menggunakan sosial media dalam hal ini adalah *Facebook*. Rasa tidak nyaman yang dirasakan disebabkan oleh banyaknya kiriman baik berupa status, video, gambar, tautan maupun komentar yang berisi ujaran kebencian. Oleh karena itu, ujaran kebencian dijadikan sebagai topik yang akan dibahas pada penelitian ini.

Setelah menentukan topik permasalahan, hal yang dilakukan berikutnya adalah mencari masalah apa yang akan diberikan solusi serta seperti apa solusi yang akan diberikan. Untuk masalah yang akan diberikan solusi, dipilih ujaran kebencian pada kiriman berupa komentar. Hal ini dikarenakan jumlah kiriman berupa komentar lebih banyak dibanding status baik yang berisi ujaran kebencian maupun tidak. Dengan jumlah komentar yang lebih banyak, maka lebih banyak juga data yang dapat dianalisis. Selain itu, kiriman berupa komentar dipilih karena komentar lebih mudah untuk didapatkan dibanding kiriman berupa status.

3.1.2 Pemilihan Alternatif Solusi

Kemudian untuk solusi yang akan diajukan, menimbang beberapa alternatif solusi sebagai berikut :

1. Menampilkan dialog yang memberikan peringatan kepada pengguna bahwa komentar yang dikirimkan mengandung unsur ujaran kebencian dan menyarankan pengguna untuk mengubah komentarnya agar lebih sesuai.
2. Menyembunyikan komentar yang dibuat pengguna lain yang terindikasi sebagai ujaran kebencian.

3. Menghapus komentar pengguna yang telah dikirimkan sebelumnya, jika komentar tersebut kemudian teridentifikasi sebagai ujaran kebencian.

Dari ketiga alternatif solusi diatas, menampilkan dialog peringatan dipilih. Hal ini dikarenakan dengan menampilkan dialog, sistem dapat membuat pengguna untuk berfikir kembali untuk mengirim konten yang terindikasi ujaran kebencian serta tidak membatasi hak pengguna untuk bersosial media meski terjadi sedikit interupsi ketika dialog tersebut harus ditampilkan[15]. Sementara untuk menyembunyikan komentar pengguna lain diperlukan diperlukan lebih banyak proses klasifikasi karena tiap komentar perlu diperiksa apakah termasuk ujaran kebencian atau tidak. Selain itu, penghapusan komentar pengguna dikemudian hari tidak dipilih dikarenakan alternatif solusi ini pernah diujicoba oleh facebook namun tidak lagi dilanjutkan. Hal ini dikarenakan terdapat insiden dimana terjadi kesalahan deteksi ujaran kebencian pada sebuah status seorang pengguna facebook yang menimbulkan rasa tidak puas pada pengguna tersebut. Dengan diberikan peringatan sebelum komentar dikirimkan, pengguna dapat memperbaiki tulisannya supaya dapat diterima oleh sistem serta tidak mengurangi hak pengguna dalam berpendapat. Oleh karena itu, dengan alternatif solusi yang diajukan, diharapkan proses deteksi dan penanganan menjadi lebih cepat serta lebih dapat diterima bagi pengguna.

3.1.3 Pemilihan Teknik Penyelesaian Masalah

Untuk dapat menampilkan dialog segera sesaat setelah pengguna mengirimkan komentar diperlukan sistem deteksi dini yang dapat berjalan secara *realtime* (waktu nyata). Agar sistem berjalan secara waktu nyata diperlukan teknik klasifikasi yang cepat serta cukup akurat. SVM merupakan salah satu algoritma yang banyak digunakan untuk klasifikasi teks dan memiliki akurasi serta kinerja dari segi waktu yang cukup baik[5][6][8][9]. Oleh karena itu, SVM dipilih sebagai algoritma klasifikasi untuk sistem yang akan dibuat.

3.2 Instrumen Penelitian

Berikut adalah instrumen-instrumen yang digunakan dalam melakukan penelitian ini:

3.2.1 Instrumen Pengumpulan Data

Data yang akan digunakan didapatkan dari komentar pengguna pada kiriman yang dibuat oleh halaman facebook portal berita. Halaman portal berita yang menjadi objek pengumpulan data nya adalah Kompas.com, detikcom dan tvOneNews. Komentar dari halaman tersebut dikumpulkan dengan cara disimpan dan diolah kedalam perangkat lunak spreadsheet seperti microsoft excel .Selain mengumpulkan data secara manual, data komentar dapat dikumpulkan menggunakan API Graf yang telah disediakan *Facebook*. Untuk melakukan panggilan API menggunakan ekstensi Curl pada PHP. token akses yang digunakan merupakan token akses yang disediakan pada halaman penjelajah API Graf untuk melakukan panggilan API pada proses pengumpulan data. Data yang telah dikumpulkan terlebih dahulu diberi label untuk membedakan mana data yang termasuk ke dalam ujaran kebencian maupun tidak untuk masing-masing data. Kemudian, data akan disimpan ke dalam basis data sistem untuk diproses lebih lanjut.

3.2.2 Instrumen Analisis Data

Pada penelitian ini dilakukan analisis terhadap data yang diperoleh pada komentar *Facebook* dengan langkah-langkah sebagai berikut:

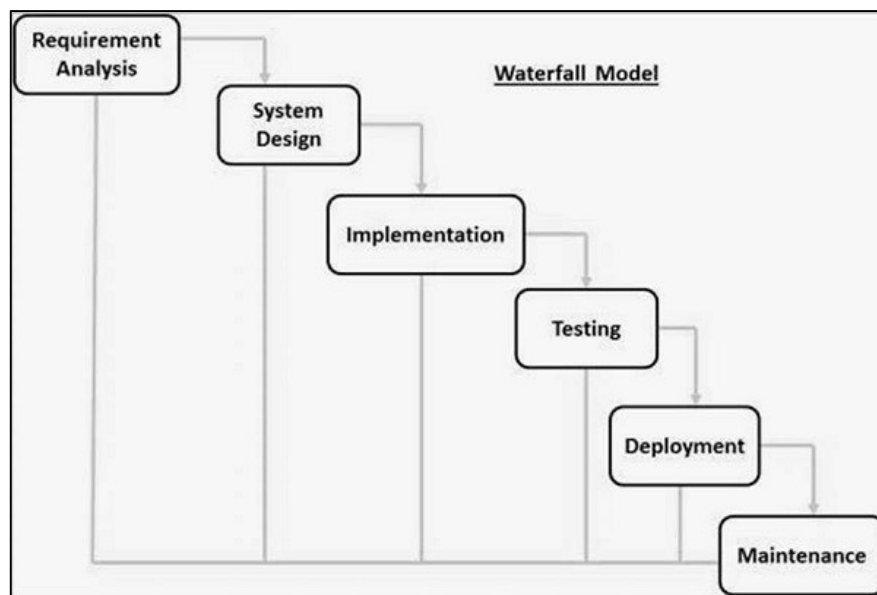
1. Komentar-komentar yang telah didapatkan dari proses scrapping akan dipilah kembali untuk menentukan komentar mana saja yang akan digunakan. Menghilangkan komentar yang tidak relevan sama sekali dengan kiriman, komentar duplikat maupun komentar yang terindikasi sebagai spam.
2. Data yang digunakan akan dibagi menjadi dua yaitu 80% sebagai data latih dan 20% sebagai data uji.

3. Data yang telah dipilih akan masuk ke tahap preprocessing yang terdiri dari stop-word removal, tokenisasi serta stemming.
4. Data yang telah dipreproses akan diambil bobot nya menggunakan TF-IDF.
5. Implementasi algoritma *Support Vector Machine* pada hasil pembobotan TF-IDF untuk membangun sistem peringatan dini pada sebuah kiriman Facebook.

3.2.3 Instrumen Pengembangan Sistem

Selain beberapa instrumen yang telah disebutkan pada bagian sebelumnya, Terdapat beberapa instrumen yang berkaitan dengan proses pengembangan sistem.

1. Model Waterfall akan digunakan sebagai acuan dalam tahapan pengembangan aplikasi. Waterfall termasuk dalam sekuensial linear sistematis yang mana terdiri dari analisis, desain, implementasi, pengujian.



Gambar III.1 model pengembangan perangkat lunak waterfall

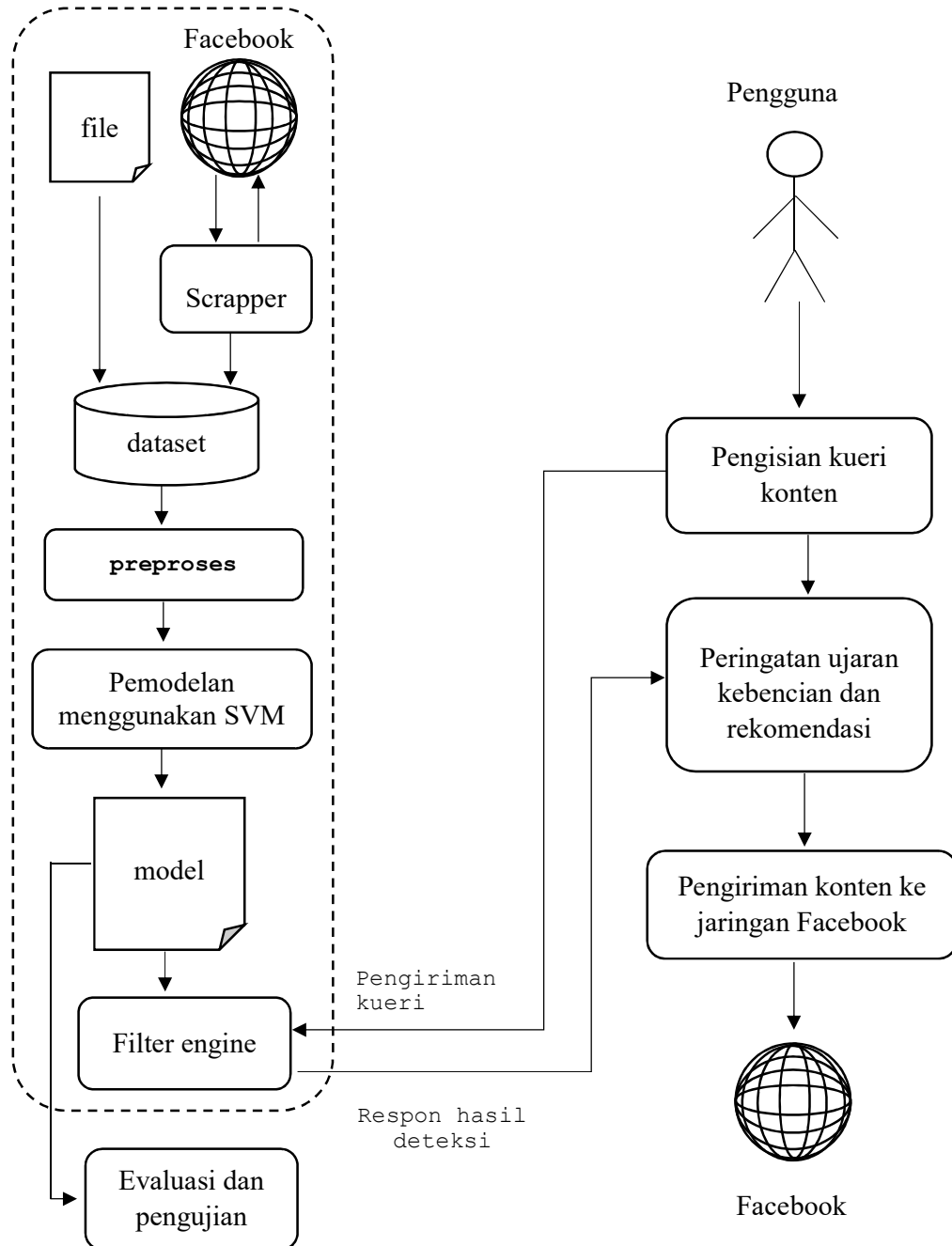
Proses analisis kebutuhan akan mendokumentasikan kebutuhan yang diperlukan oleh sistem yang akan digambarkan menggunakan use case. Proses desain akan mendokumentasikan tentang spesifikasi sistem yang digambarkan menggunakan class diagram, sequence diagram dan mockup.

Proses implementasi akan menterjemahkan desain yang telah dibuat kedalam kode bahasa pemrograman. Proses pengujian sistem akan menguji apakah sistem telah dibuat sesuai dengan spesifikasi dengan menggunakan unit test.

2. Sistem yang akan dibuat memerlukan akses ke facebook. Oleh karena itu, sebuah akses token diperlukan untuk melakukan panggilan API Graf. Untuk mendapatkan token tersebut kita perlu mendaftarkan diri sebagai pengembang aplikasi facebook. Selain itu, versi API Graf yang dapat digunakan adalah sesuai dengan rekomendasi dari facebook yaitu versi 2.6 ke atas. Dalam hal ini, penelitian ini menggunakan API Graf versi 3.0.
3. Untuk mengimplementasikan SVM ke dalam sistem, digunakan algoritma *Sequential Minimum Optimization* (SMO) yang diperkenalkan oleh John Platt[16]. SMO digunakan karena SMO menghasilkan waktu training yang lebih singkat dibanding algoritma lain seperti *chunking*.
4. Untuk proses *stemming* dan *stop word removal* menggunakan *library stemming* sastrawi untuk bahasa pemrograman PHP.
5. Pada penelitian ini koneksi internet diperlukan agar sistem dapat melakukan panggilan Graf.

3.3 Metode yang Diusulkan

Berikut adalah gambaran dari metode yang diusulkan:



Gambar III.2 Metode yang diusulkan

Dari gambar di atas, metode yang diusulkan dapat dibagi menjadi tahapan-tahapan dengan penjelasan sebagai berikut :

3.3.1 Pengumpulan Data

Pada tahapan ini data akan dikumpulkan dari halaman facebook. Pengumpulan data dapat dilakukan secara manual maupun otomatis. Pengumpulan data secara manual dilakukan dengan cara menyalin seluruh atau sebagian komentar berupa teks yang terdapat pada kiriman facebook ke dalam perangkat lunak spreadsheet. Kiriman yang dijadikan sumber data merupakan kiriman yang memiliki lebih dari 30 komentar. Selain teks komentar dikumpulkan juga url kiriman facebook, halaman facebook yang membuat kiriman dan *caption*/judul tautan yang dilampirkan pada kiriman tersebut.

Sementara untuk melakukan pengumpulan data secara otomatis menggunakan scrapper. Namun, untuk penelitian ini pengumpulan data dilakukan secara manual seperti yang sudah dijelaskan sebelumnya.

3.3.2 Melakukan Preproses Terhadap Data

Pada tahapan ini data akan di preproses lebih lanjut dengan melakukan seleksi, *cleaning* dan transformasi data.

Pada proses seleksi dan *cleaning*, Data yang telah dikumpulkan akan dipilih lagi secara semi-acak dengan batasan maksimal 30 komentar yang akan diambil untuk tiap post hingga didapat sejumlah 509 data komentar. Sementara itu, atribut yang akan digunakan dari data yang dikumpulkan adalah atribut teks komentar. Semi-acak yang dimaksud adalah data yang dipilih masih relevan dengan kriteria:

1. Merupakan komentar berbahasa indonesia baku maupun tidak baku. Komentar dengan sedikit kosakata bahasa daerah masih dipilih. Bukan merupakan komentar yang seluruhnya dalam bahasa asing atau daerah tertentu.
2. Isi komentar membahas dan sesuai dengan konten yang dikirim
3. Bukan merupakan komentar spam atau iklan
4. Komentar belum dipilih sebelumnya (bukan duplikat)

Kemudian data yang telah dipilih akan diberi label dengan ketentuan sebagai berikut:

1. Komentar diberi label -1 bila memenuhi unsur ujaran kebencian seperti Penghinaan, pencemaran nama baik, penistaan ,memprovokasi, dan menghasut [13].
2. Selain itu data diberi label 1.

Data yang merupakan ujaran kebencian diberi label -1 dan data yang bukan ujaran kebencian akan diberi label 1. Sehingga data yang didapatkan memiliki 2 atribut yaitu teks dan label. Data kemudian dibagi menjadi 2 untuk dijadikan dataset latih dan dataset uji dengan perbandingan 80% data untuk data latih dan 20% data untuk data uji. Setelah itu, baik dataset latih maupun dataset uji disimpan ke dalam basis data.

Kemudian Transformasi data dilakukan dengan cara *stemming*, *stop word removal*, tokenisasi, dan pembobotan. Proses *stemming* dan *stop word removal* akan menggunakan library Sastrawi . Proses tokenisasi akan menggunakan fungsi yang telah disediakan PHP. Kemudian pembobotan akan menggunakan teknik pembobotan TF-IDF.

3.3.3 Pemodelan menggunakan SVM

Data yang telah di preproses akan masuk ke tahapan pemodelan. Pada tahap pemodelan, data akan dilatih menggunakan metode klasifikasi SVM. Proses pelatihan data akan menghasilkan sebuah model yang dapat digunakan untuk proses klasifikasi pada tahapan berikutnya. Model yang dihasilkan akan berisi beberapa informasi seperti :

1. Nilai bobot w jika menggunakan kernel linear
2. Nilai α
3. Label dan data yang termasuk support vector
4. Paramater yang digunakan saat training.

Yang mana nilai nilai tersebut akan digunakan saat proses prediksi.

3.3.4 Implementasi Model Pada Mesin Filter

Model yang telah dihasilkan pada proses training akan disimpan pada server dan diterapkan pada mesin filter. Mesin filter merupakan layanan perantara (*service*) yang akan menentukan apakah sebuah permintaan yang dikirimkan pengguna termasuk ujaran kebencian atau tidak berdasarkan model yang telah dihasilkan sebelumnya. Pengguna akan mengirimkan permintaan menggunakan perangkat lunak yang terpasang pada sisi klien dengan memanfaatkan *Asynchronous JavaScript and XMLHTTP* (AjaX) untuk mengirimkan permintaan. Service yang dibuat merupakan *Representational State Transfer*(REST) web service yang akan memberikan respon berupa file *JavaScript Object Notation*(JSON). Kemudian sisi server akan memprediksi teks berdasarkan permintaan yang didapat lalu mengirimkan pesan berdasarkan hasil prediksi. Sistem pada sisi client kemudian akan melakukan aksi berdasarkan pesan yang diterima dari mesin filter.

3.3.5 Evaluasi dan pengujian

Model yang telah dihasilkan akan diuji menggunakan data testing dan dievaluasi lebih lanjut.

3.4 Evaluasi dan Validasi

1. Confussion Matrix

Evaluasi dan validasi hasil menggunakan *confussion matrix*. Dari *confussion matrix* tersebut akan dihitung nilai akurasi, *precision*, dan *recall*. *Confussion matrix* sendiri merupakan tabel yang berisi jumlah data yang dikategorikan secara benar dan dikategorikan secara salah. Berikut adalah contoh sebuah *confussion matrix* :

Tabel III.1 contoh tabel *confussion matrix*

		Label prediksi	
		Benar	salah
Label aktual	Benar	14 (<i>true-positive</i>)	3 (<i>false-negative</i>)
	salah	4 (<i>false-positive</i>)	9 (<i>true-negative</i>)

Formula untuk menghasilkan akurasi yaitu dengan membagi jumlah data yang diprediksi dengan benar (*True Positive* dan *True Negative*) dengan total keseluruhan data testing.

$$\begin{aligned} \text{akurasi} &= \frac{\text{true positive} + \text{true negative}}{\text{total data}} \\ &= \frac{14 + 9}{14 + 9 + 3 + 4} \\ &= \mathbf{0.821} \end{aligned}$$

Nilai *precision* dihitung dengan membagi jumlah data benar yang bernilai positif (*True Positive*) dibagi dengan jumlah data yang benar bernilai positif (*True Positive*) dan data yang salah bernilai positif (*False Positive*).

$$\begin{aligned} \text{precision} &= \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \\ &= \frac{14}{14 + 4} \\ &= \mathbf{0.777} \end{aligned}$$

Nilai *Recall* dihitung dengan membagi data yang benar bernilai positif (*True Positive*) dengan jumlah data data yang benar bernilai positif (*True Positive*) dan data yang salah bernilai negatif (*False Negative*).

$$\begin{aligned} \text{recall} &= \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \\ &= \frac{14}{14 + 3} \\ &= \mathbf{0.823} \end{aligned}$$

2. Unit Test

Selain *confussion matrix*, Unit Test digunakan untuk mengevaluasi aplikasi yang digunakan berdasarkan spesifikasi yang telah ditentukan. Unit Test akan menguji apakah sebuah unit pada program seperti fungsi atau modul berjalan sesuai dengan yang diinginkan. Dalam sebuah pengujian menggunakan unit testing, beberapa test case akan dibuat berdasarkan alur program, masukan serta hasil yang diharapkan. Berikut adalah contoh test case yang digunakan dalam unit test.

Tabel III.2 Contoh test case untuk fungsi cek prima

No	Input	Output diharapkan
1.	Number = 0	False
2.	Number = 13	True
3.	Number = 27	false

Contoh diatas merupakan sebuah tabel yang berisi test case untuk sebuah fungsi cek prima. Sebuah unit test akan menentukan nilai input yang diujikan berdasarkan alur proses yang sudah kita ketahui pada sebuah modul (*white-box*). Sementara test case dipaparkan dalam bentuk *input-output driven test (black-box)* seperti berikut.

Tabel III.3 Tabel hasil unit testing

No	Input	Output diharapkan	Output dihasilkan	Keterangan
1.	Number = 0	False	False	1 dari 1 pernyataan ok
2.	Number = 13	True	True	1 dari 1 pernyataan ok
3.	Number = 27	False	true	Terjadi kesalahan

3. Performance Test

Performance testing ditujukan untuk menguji sistem dari sisi performa atau efisiensi, yang biasanya mengacu pada waktu respon (*response time*) atau *throughput* saat sistem menjalankan kerja dan konfigurasi tertentu. Dalam penelitian ini metrik yang digunakan untuk menguji performa adalah *response time*. waktu respon akan dipaparkan

dalam satuan waktu milidetik atau detik. Pengujian performa akan dilakukan beberapa kali untuk menemukan minimum, maksimum, dan rata-rata waktu respon yang dibutuhkan.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Kebutuhan

4.1.1 Analisis Kebutuhan Data

Pada penelitian ini terdapat data yang diperlukan berupa kumpulan sampel komentar facebook yang telah diberi label ujaran kebencian maupun bukan. Untuk mendapatkan data tersebut peneliti melakukan pengumpulan data secara manual. Selain itu, peneliti menyiapkan *crawler* yang dapat mengumpulkan komentar dari facebook secara otomatis. Data yang telah didapat kemudian diseleksi dan dipilih kembali untuk mengurangi data yang dianggap tidak relevan.

Setelah melalui proses seleksi didapatkan 509 data komentar yang akan diberi label. Lalu, data diberi label menggunakan nilai -1 dan 1 dimana -1 merupakan label untuk ujaran kebencian dan 1 untuk bukan ujaran kebencian. Dalam penelitian ini pemberian label akan dilakukan oleh penulis berdasarkan rujukan yang digunakan tentang ujaran kebencian. Data yang telah diberi label akan digunakan untuk melakukan proses klasifikasi. Untuk memenuhi kuota 80% data sebagai data latih maka diambil 406 dari keseluruhan data sebagai data latih. Sementara 103 data akan diambil sebagai data uji dengan rincian sebagai berikut.

Tabel IV.1 tabel rincian dataset

Dataset	Ujaran kebencian	Bukan ujaran kebencian	jumlah
Dataset latih	161	245	406
Dataset uji	41	62	103

4.1.2 Analisis Kebutuhan Pengguna

Pada kebutuhan pengguna, kebutuhan dibedakan menjadi 2 jenis yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Untuk masing-masing pengguna kebutuhan masih dibagi lagi berdasarkan jenis pengguna yang

menggunakan yaitu admin dan pengguna akhir (*end-user*). Berikut adalah gamabran mengenai kebutuhan dari pengguna.

Tipe kebutuhan	Kebutuhan
Fungsional	<p>Admin :</p> <ol style="list-style-type: none"> 1. Pengguna dapat menambahkan dataset 2. Pengguna dapat menjalankan proses pembelajaran 3. Pengguna dapat memilih model yang akan digunakan 4. Pengguna dapat melakukan proses evaluasi terhadap model yang dihasilkan <p>End-User :</p> <ol style="list-style-type: none"> 1. Sistem dapat menampilkan peringatan jika komentar terindikasi mengandung ujaran kebencian
Non-fungsional	<p>Admin :</p> <ol style="list-style-type: none"> 1. Keamanan - terdapat mekanisme login untuk melindungi pengaturan mesin. 2. Ergonomy – sistem memiliki antarmuka yang mudah digunakan <p>End-User :</p> <ol style="list-style-type: none"> 1. Response-time – peringatan harus segera muncul dengan waktu maksimal 4 detik sejak pengguna menekan tombol kirim

Tabel IV.2 Tabel kebutuhan pengguna

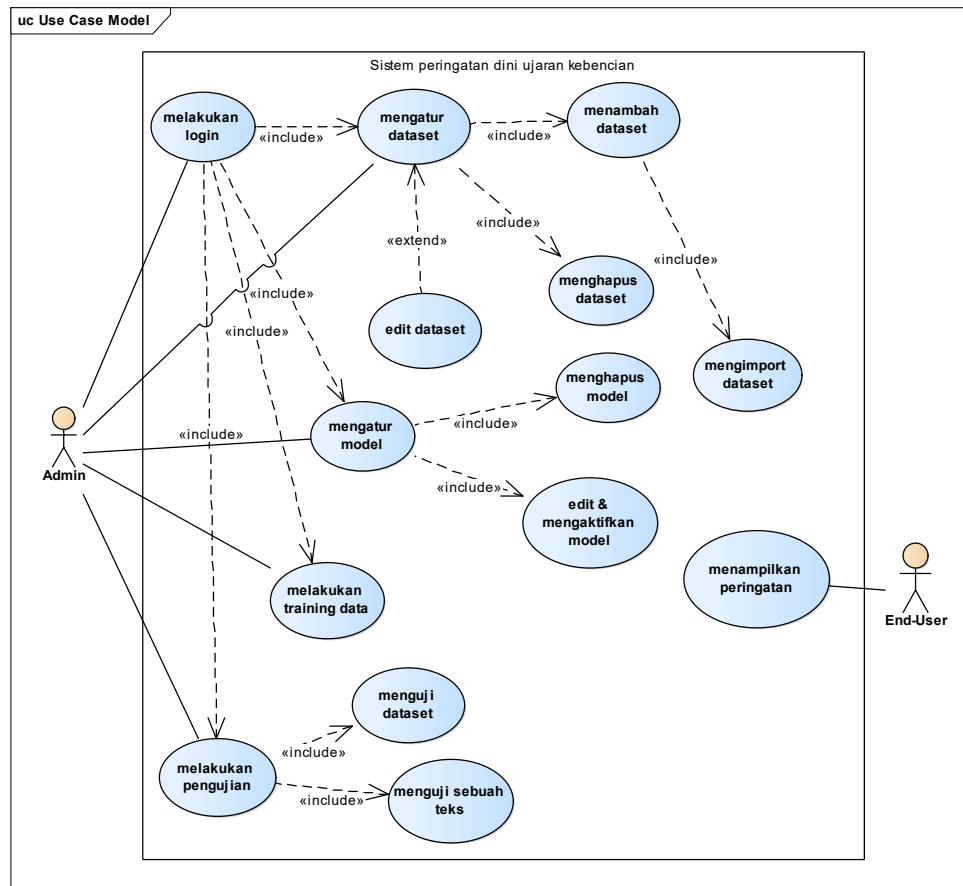
Pada bagian kebutuhan non-fungsional untuk end-user terdapat kebutuhan response time yang kurang dari 4 detik. Hal ini dikarenakan supaya end-user tetap fokus pada program dan merasa bahwa program berjalan sebagaimana mestinya. 4

detik dianggap merupakan jeda waktu yang masih dalam batas toleransi dimana 1 detik merupakan waktu bagi kebanyakan orang merasakan adanya jeda dan 10 detik merupakan waktu bagi kebanyakan orang untuk tetap menggunakan program tersebut [17] .

4.2 Desain Sistem

Pada bagian desain, proses desain digambarkan menggunakan diagram UML (*Unified Modeling language*) seperti *Use Case Diagram*, *Class Diagram*, dan *Sequence Diagram*.

4.2.1 Use Case



Gambar IV.1 Gambar Use Case Diagram

Gambar 4.1 merupakan Use Case yang menggambarkan fungsi apa saja yang akan dibuat dalam sistem. Pada sistem yang akan dibuat terdapat 4 fitur utama

untuk admin yang digambarkan pada use case mengatur dataset, mengatur model, melakukan training, melakukan pengujian. 4 use case tersebut memiliki kegunaan masing-masing sebagai berikut :

1. Mengatur dataset

admin dapat mempersiapkan dataset yang akan digunakan untuk proses training maupun testing. Untuk mengatur dataset sistem harus dapat menambahkan, menghapus serta mengubah dataset.

2. mengatur model

admin dapat menentukan model mana yang akan digunakan sistem untuk mengklasifikasikan masukan dari end-user.

3. melakukan training

pada use case ini admin dapat melakukan proses training serta menentukan opsi training yang ingin dilakukan. Selain itu admin dapat menyimpan hasil pembelajaran menjadi sebuah model.

4. melakukan pengujian

pada use case ini pengguna dapat mengevaluasi model yang dihasilkan pada proses training.

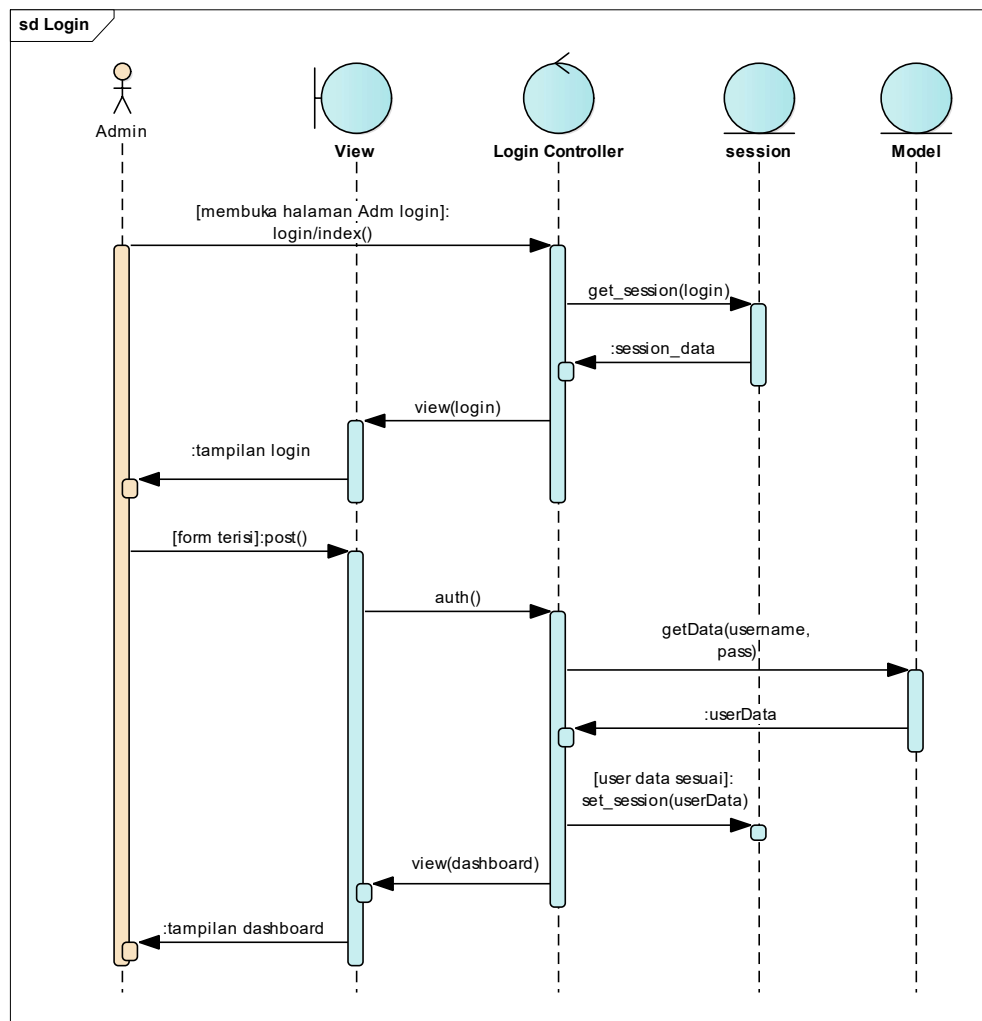
Selain itu, terdapat satu buah use case yang merupakan fitur utama yang akan didapat oleh end-user yaitu menampilkan peringatan. Pada use case ini end-user dapat memberikan masukan teks dan kemudian sistem akan menampilkan peringatan bila teks yang dikirim terdeteksi merupakan ujaran kebencian.

4.2.2 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan urutan proses dalam sebuah fungsi dan menggambarkan interaksi antar komponen dalam sistem. Berikut adalah sequence diagram yang ada dalam sistem

a. Sequence Diagram Login

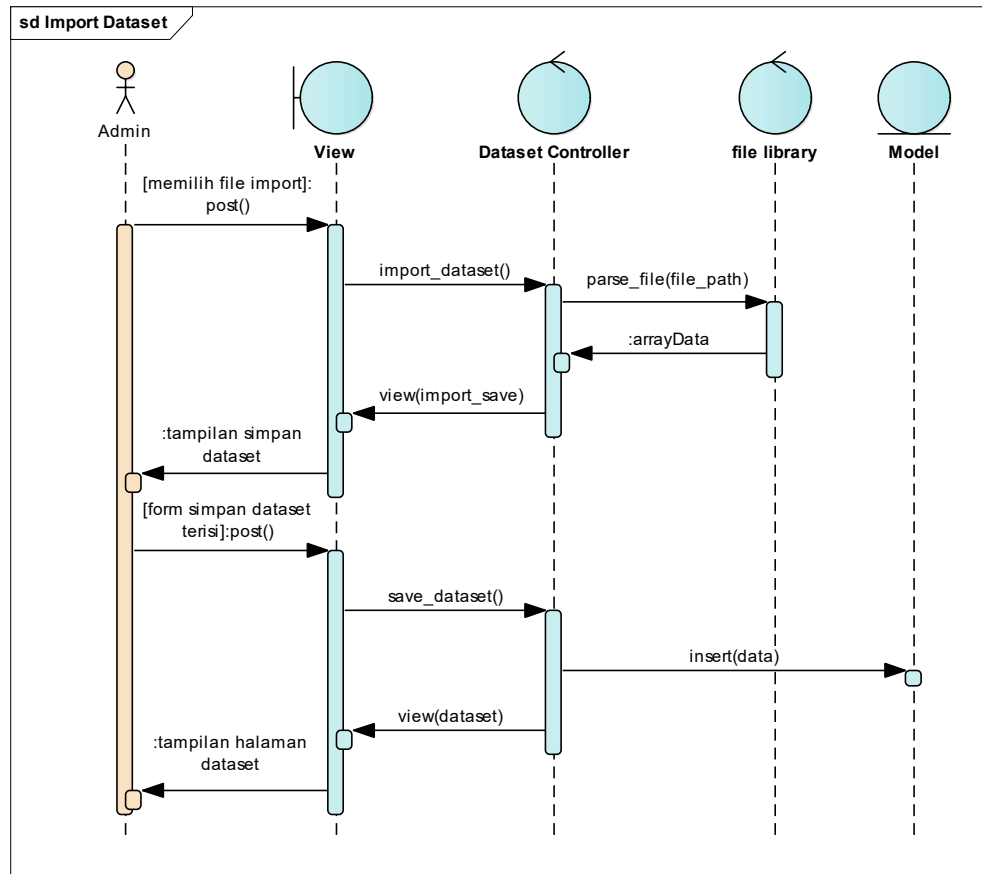
Pada proses login sistem akan memeriksa apakah pengguna memiliki sesi yang sedang aktif atau tidak. Bila tidak sistem akan menampilkan tampilan login. Pada tampilan login terdapat form yang harus diisi. Kemudian sistem akan memeriksa apakah data pengguna yang diberikan tersedia pada basis data. Lalu sistem akan membuat sesi baru dan mengarahkan kehalaman *dashboard*.



Gambar IV.2 Sequence Diagram login

b. Sequence Diagram Import Dataset

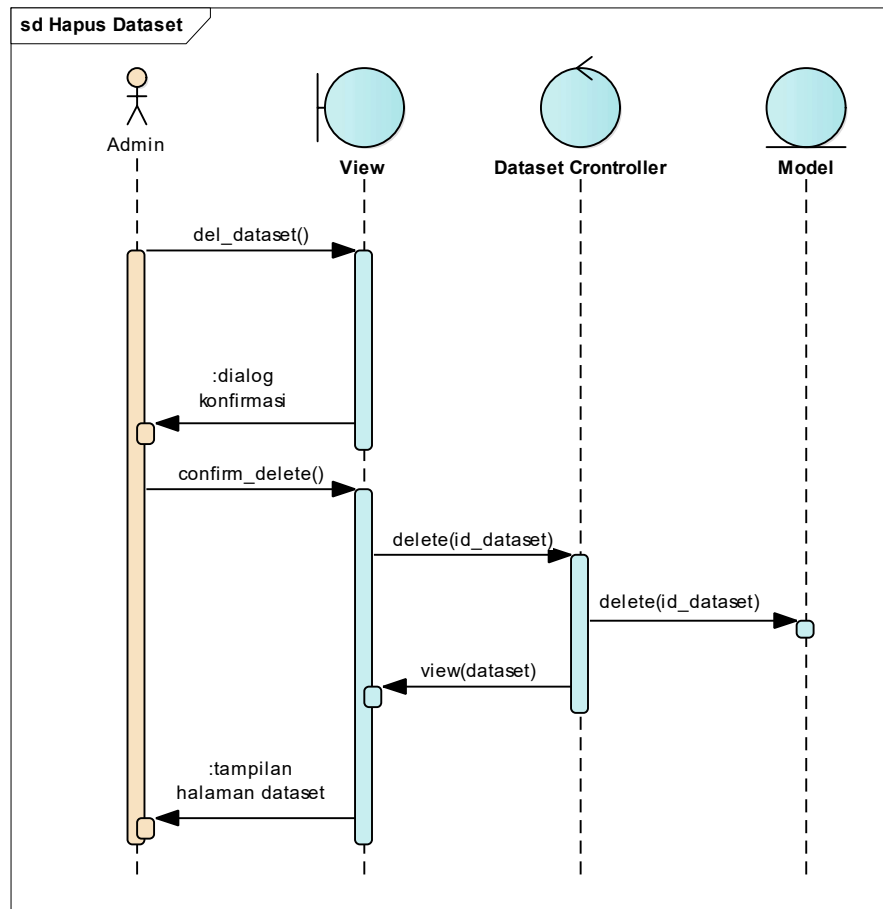
Pada proses import dataset, sistem akan meminta pengguna untuk memilih file yang akan diunggah. Sistem akan melakukan *parsing* terhadap file yang telah diunggah untuk kemudian sistem akan menampilkan halaman konfirmasi. Setelah pengguna melakukan konfirmasi untuk menyimpan data, sistem akan menyimpan data tersebut ke dalam basis data



Gambar IV.3 Sequence diagram untuk import dataset

c. Sequence Diagram Hapus Dataset

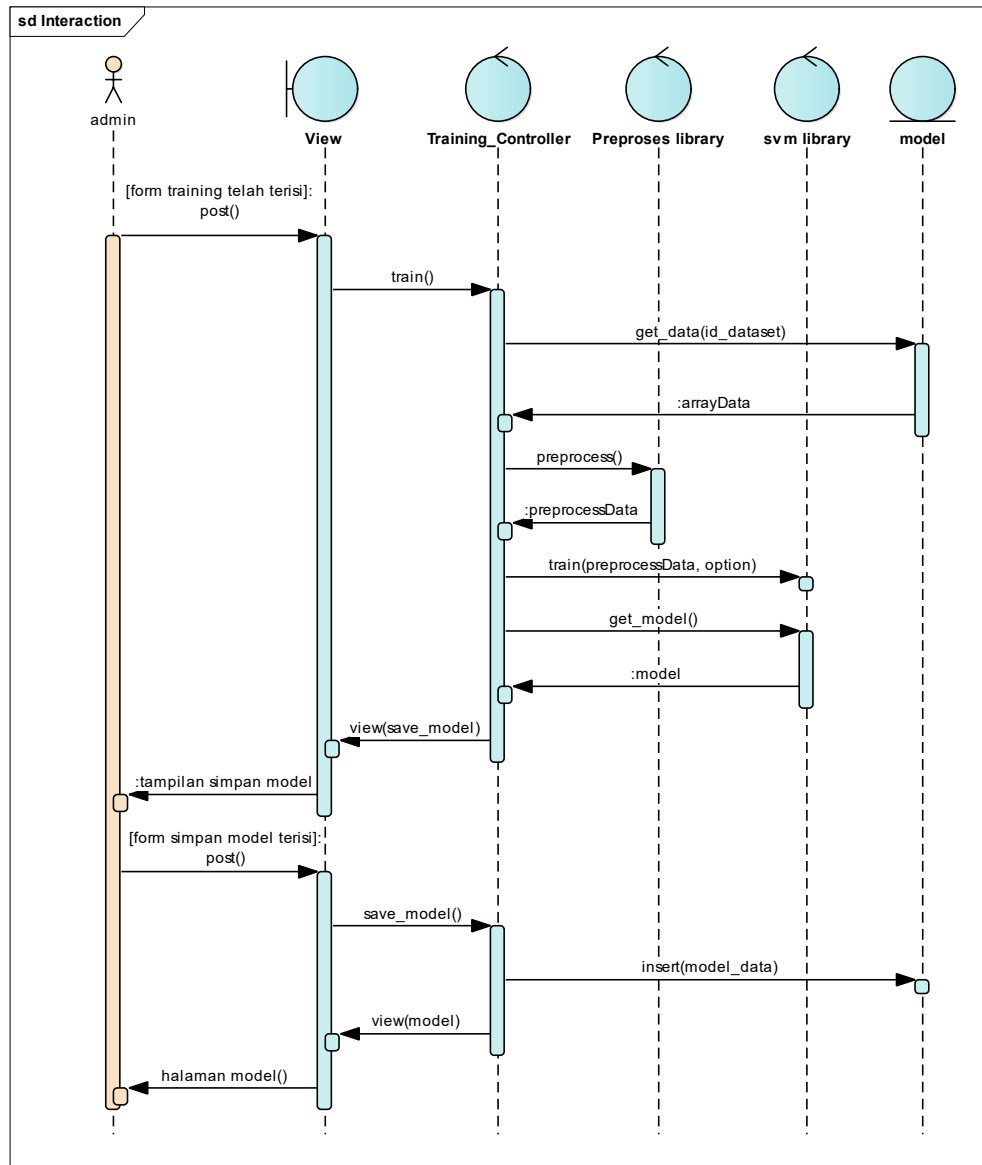
Pada proses hapus dataset, pengguna terlebih dahulu mengkonfirmasi dataset mana yang akan dihapus. Kemudian sistem akan menghapus dataset yang ada di basis data sesuai id_dataset yang dikirimkan.



Gambar IV.4 Sequence Diagram untuk proses hapus dataset

d. Sequence Diagram Melakukan Training

Pada proses training pengguna akan mengisi form training terlebih dahulu. Lalu, sistem akan mengambil data dari basis data, melakukan preproses terhadap data yang diambil, kemudian melakukan *training* menggunakan *library svm*. Hasil training akan ditampilkan untuk mendapat konfirmasi dari pengguna. Setelah pengguna melakukan konfirmasi, model akan disimpan ke dalam basis data.

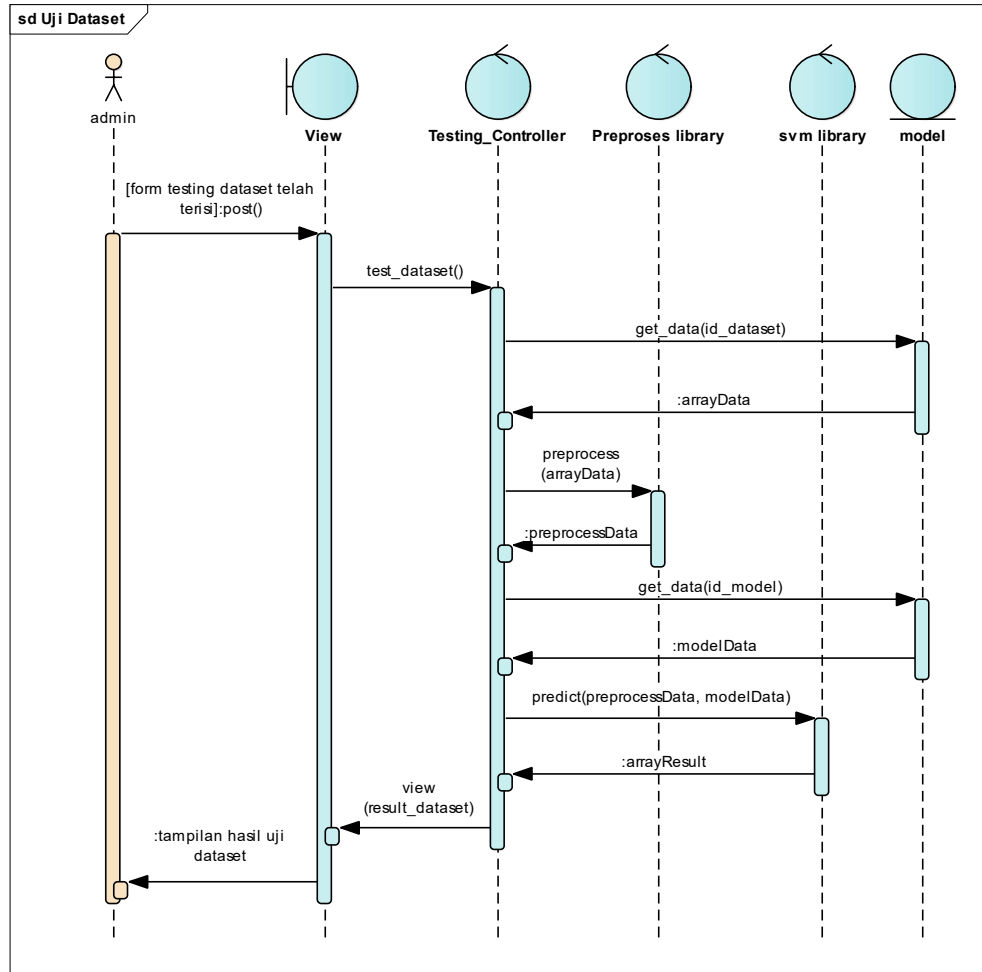


Gambar IV.5 Sequence Diagram untuk proses Training data

e. Sequence Diagram Menguji Dataset

Pada proses pengujian dataset, sistem akan memuat dataset yang ada pada basis data berdasarkan dataset pilihan pengguna untuk diuji. Dataset kemudian akan dipreproses dan diklasifikasikan berdasarkan model yang

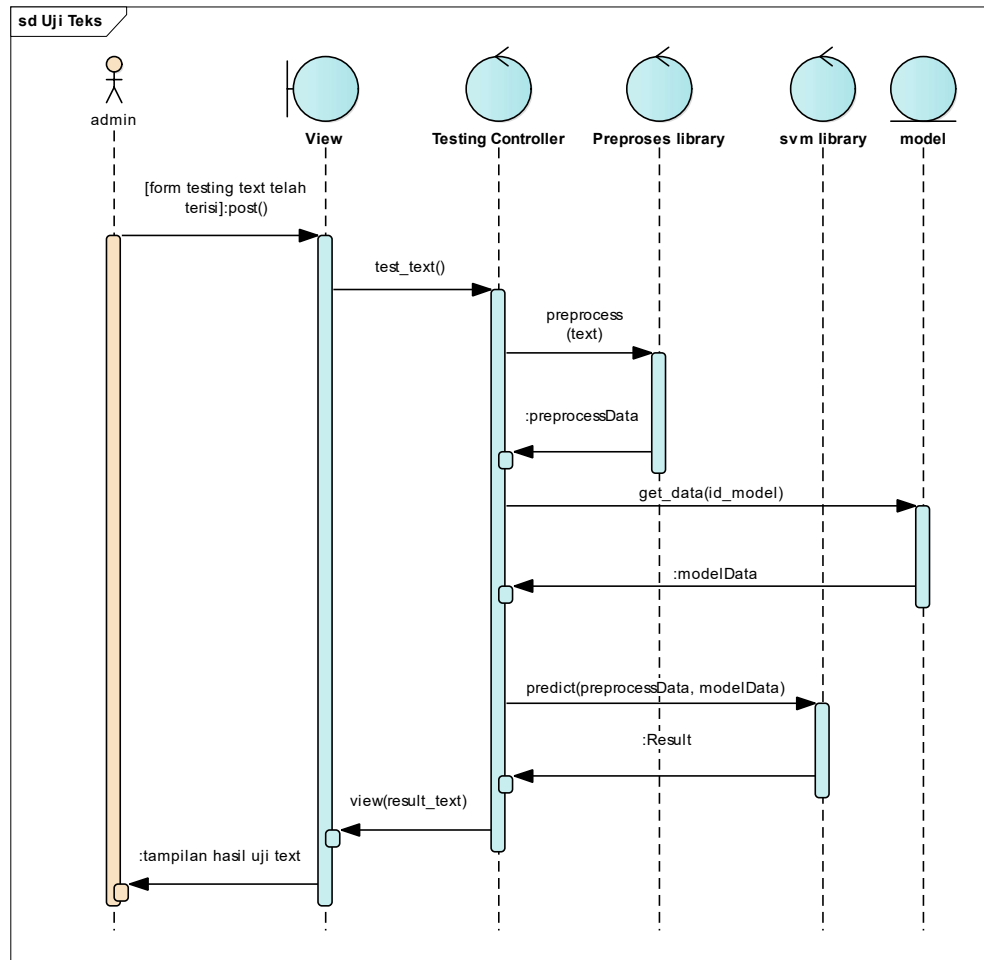
dipilih pengguna. Lalu, hasilnya akan ditampilkan pada halaman hasil uji dataset.



Gambar IV.6 Sequence Diagram untuk pengujian dataset

f. Sequence Diagram Menguji Sebuah Teks

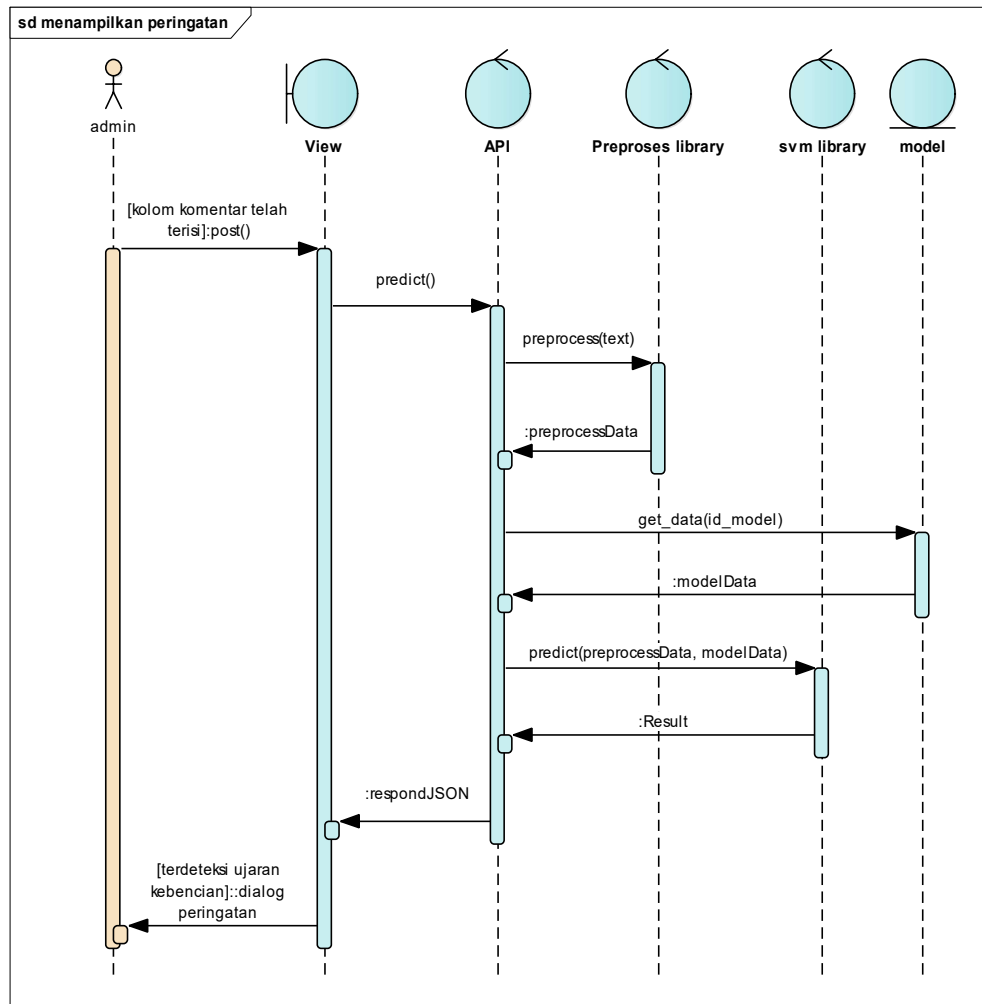
Pada proses uji teks, sistem akan melakukan proses yang hampir sama dengan uji dataset. Namun data yang akan diujikan merupakan teks yang dimasukan oleh pengguna.



Gambar IV.7 Sequence Diagram untuk proses pengujian teks

g. Sequence Diagram Menampilkan Peringatan

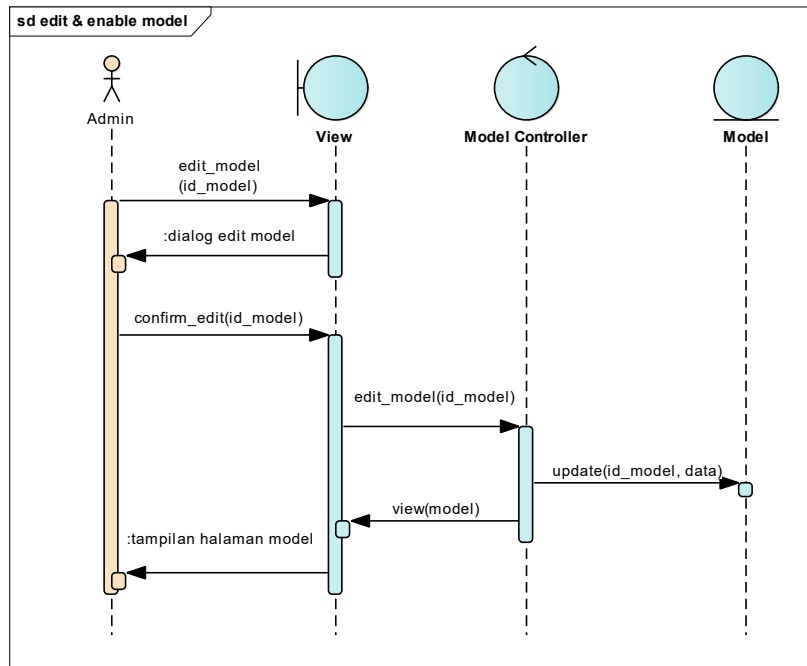
Pada proses menampilkan peringatan, komentar yang ditulis pengguna akhir akan dikirimkan ke sistem peringatan dini sebelum dikirimkan ke *facebook* untuk dievaluasi terlebih dahulu. Sistem akan memberikan *respond* sesuai dengan hasil evaluasi. sistem akan menampilkan peringatan bila hasil evaluasi menyatakan positif ujaran kebencian.



Gambar IV.8 Sequence Diagram untuk menampilkan peringatan

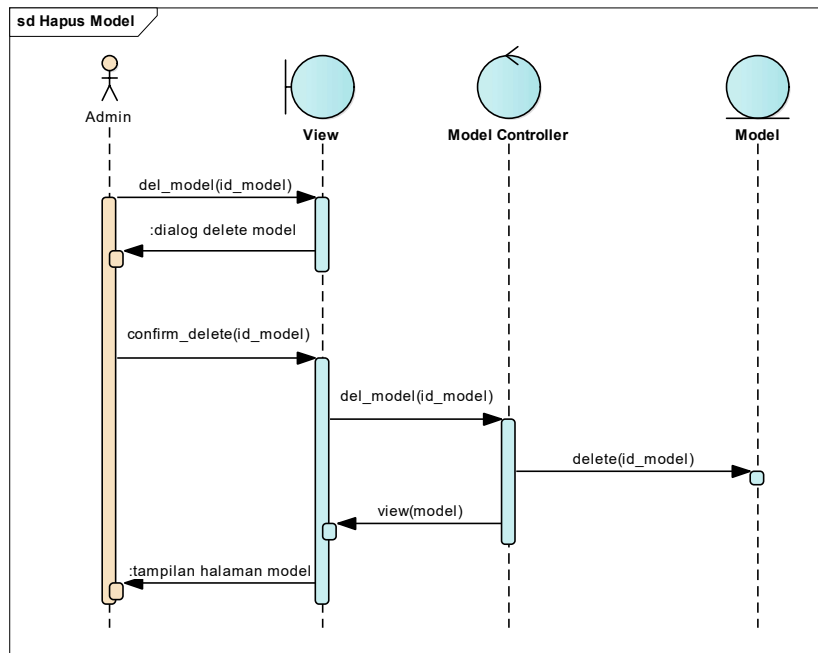
h. Sequence Diagram Mengedit Dan Mengaktifkan Model

Pada proses edit model, dialog yang berisi form edit akan muncul. pengguna dapat mengubah isi form tersebut kemudian melakukan konfirmasi untuk melakukan perubahan. Setelah pengguna melakukan konfirmasi, sistem akan mengubah data model yang ada pada basis data.



Gambar IV.9 Sequence diagram untuk edit dan mengaktifkan model

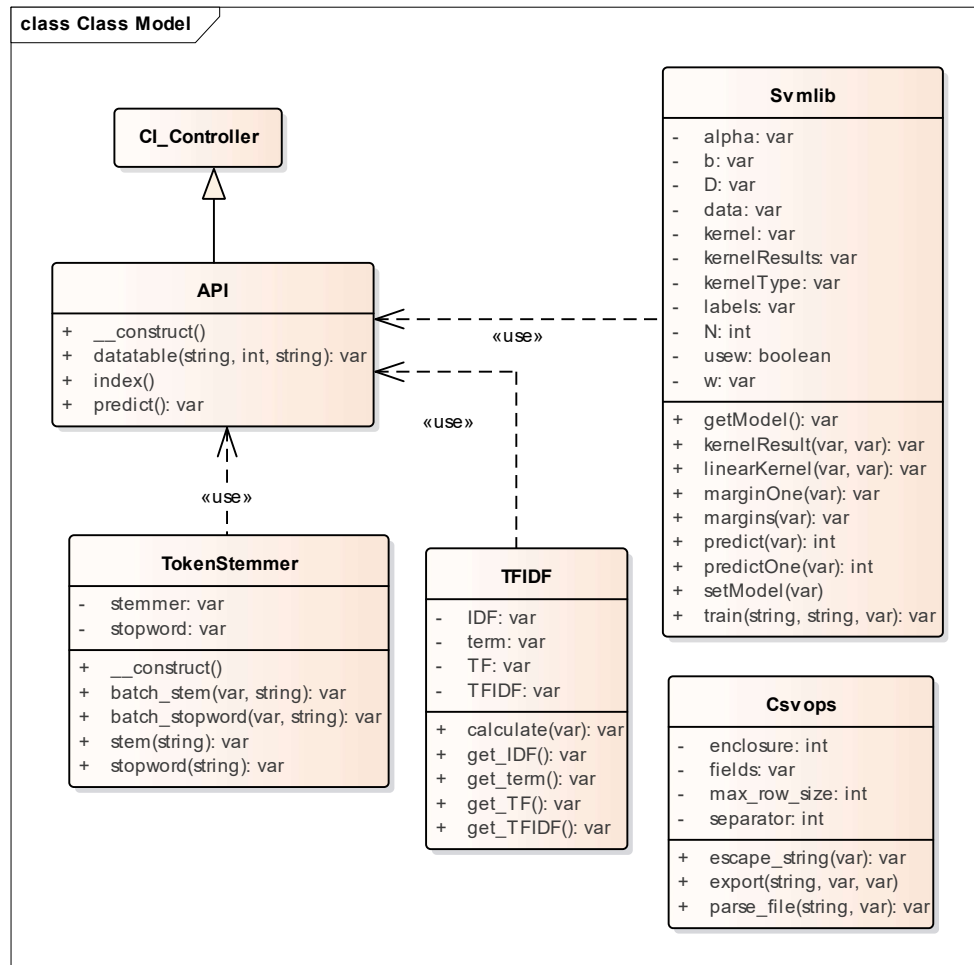
i. Sequence Diagram Menghapus Model



Gambar IV.10 Sequence diagram untuk proses hapus model

Pada proses hapus model, pengguna terlebih dulu mengkonfirmasi model yang akan dihapus. Kemudian, Sistem akan menghapus model dari basis data.

4.2.3 Class Diagram



Gambar IV.11 Gambar class diagram sistem yang akan dibuat

Pada gambar diatas merupakan *class diagram* yang menggambarkan class yang akan digunakan sistem untuk mengklasifikasi teks. Pada *class diagram* tersebut terdapat dua jenis class yaitu *class controller* dan *class library*. *Class Controller* merupakan subclass/child dari class *CI_Controller*. Sementara *class library* merupakan class yang berdiri sendiri. *Class controller* akan mengatur alur proses sistem baik pada sisi admin dan *end-user*.

Kemudian class diagram diatas merupakan sebagian class dari keseluruhan sistem yang dibuat yang dianggap cukup krusial untuk dibahas. Selain itu, pada class diagram di atas terdapat satu Class controller yaitu class API dan 4 buah class library yaitu class Svmlib, TFIDF, Csvops, serta TokenStemmer .

Class API merupakan class yang akan menangani permintaan API oleh user yang menggunakan AJAX(Asynchronous JavaScript and XMLHTTP). Class ini akan digunakan baik ketika sistem dijalankan oleh admin maupun *end-user*. Fungsi yang dapat digunakan pada class ini yaitu adalah meminta data yang akan ditampilkan pada halaman admin, dan mengklasifikasikan teks yang dikirim oleh *end-user*.

Brikutnya, Class Svmlib merupakan class yang digunakan untuk melakukan pembelajaran dan klasifikasi dengan teknik SVM. Kemudian, class TFIDF adalah sebuah class yang berfungsi untuk mendapatkan bobot tiap kata pada sebuah dataset kalimat berdasarakan perhitungan TF-IDF.

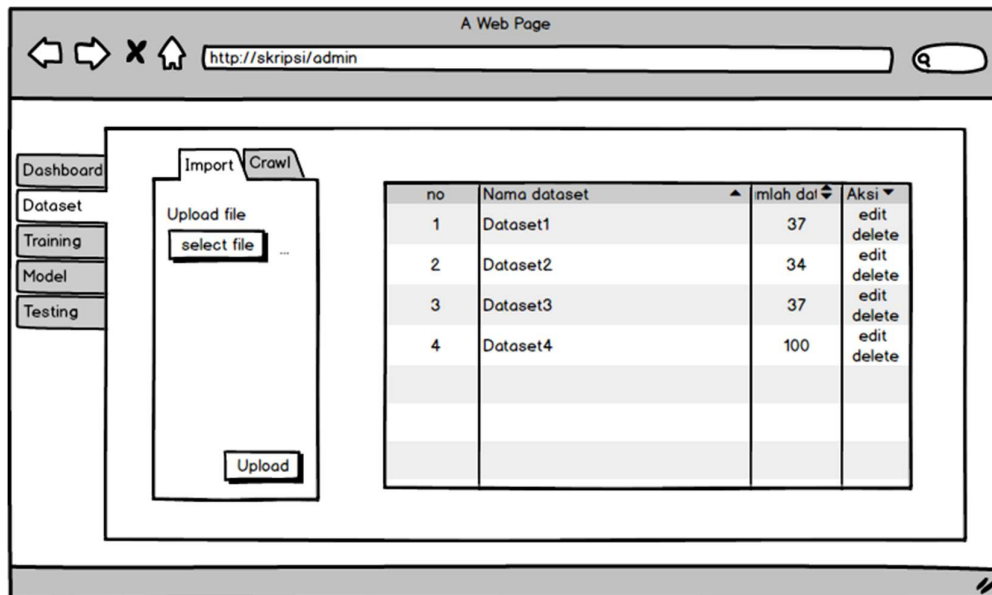
Lalu, class TokenStemmer merupakan class yang dapat digunakan untuk melakukan proses stemming dan stopwords-removal sebuah data teks maupun kumpulan data teks dalam bentuk array. Class Csvops merupakan sebuah library yang akan membantu untuk membaca file .csv dan merubahnya ke dalam bentuk array.

Class Svmlib, TFIDF, dan TokenStemmer merupakan class yang akan digunakan setiap kali saat proses pembelajaran maupun klasifikasi dilakukan. Smentara class Csvops hanya akan digunakan ketika admin melakukan import file .csv ke database

4.3 Desain Antarmuka Sistem

Untuk memberikan gambaran serta konsep mengenai antarmuka yang akan didapat oleh pengguna, Berikut adalah desain dari antarmuka berupa *mockup* yang akan dibuat :

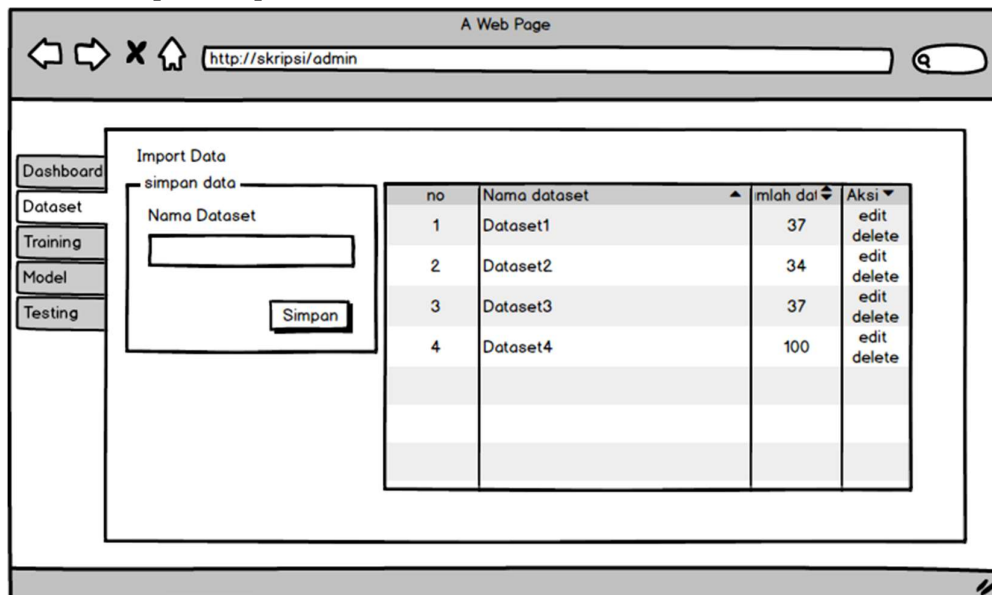
a. Halaman Dataset



Gambar IV.12 Gambar desain halaman dataset

Pada halaman dataset terdapat dua bagian yaitu yang bagian import dataset dan list dataset. Pada bagian import dataset admin dapat memilih file untuk mulai melakukan proses import file kemudian klik tombol upload. Pada bagian list dataset pengguna dapat menghapus dataset yang sudah ada.

b. Tampilan Import Dataset

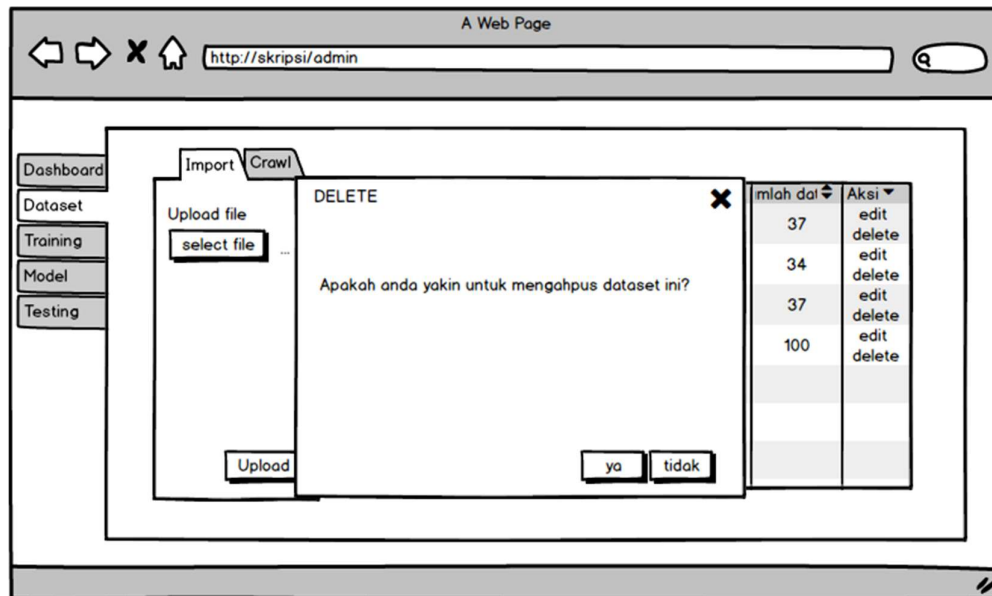


Gambar IV.13 Gambar Desain halaman simpan dataset

Setelah melakukan upload file dataset pengguna perlu mengkonfirmasi untuk menyimpan dataset tersebut ke dalam *database*. Admin cukup mengisi form kemudian menekan tombol simpan.

c. Tampilan Hapus Dataset

Pada bagian list dataset pengguna dapat melakukan proses hapus dataset dengan menekan tombol delete yang tersedia pada kolom aksi. Ketika tombol tersebut ditekan tampilan seperti di bawah akan muncul. Admin cukup mengkonfirmasi jika ingin menghapus dataset tersebut.



Gambar IV.14 Gambar desain tampilan hapus dataset

d. Halaman Training

Pada halaman training terdapat beberapa form yang dapat diubah dan pilihan dataset yang harus diisi oleh admin. Kemudian jika admin telah selesai menentukan parameter yang diinginkan, admin dapat menekan tombol untuk segera melakukan training.

A Web Page

http://skripsi/admin

Dashboard
Dataset
Training
Model
Testing

Training

preproses

☐ stem
☐ stop word remove

svm

parameter 1
3

parameter 2
0.01

dataset

mulai

Gambar IV.15 Gambar desain halaman Training

e. Halaman Hasil Training

A Web Page

http://skripsi/admin

Dashboard
Dataset
Training
Model
Testing

Hasil Training

Simpan Model

Nama Model

simpan

Hasil Pembelajaran

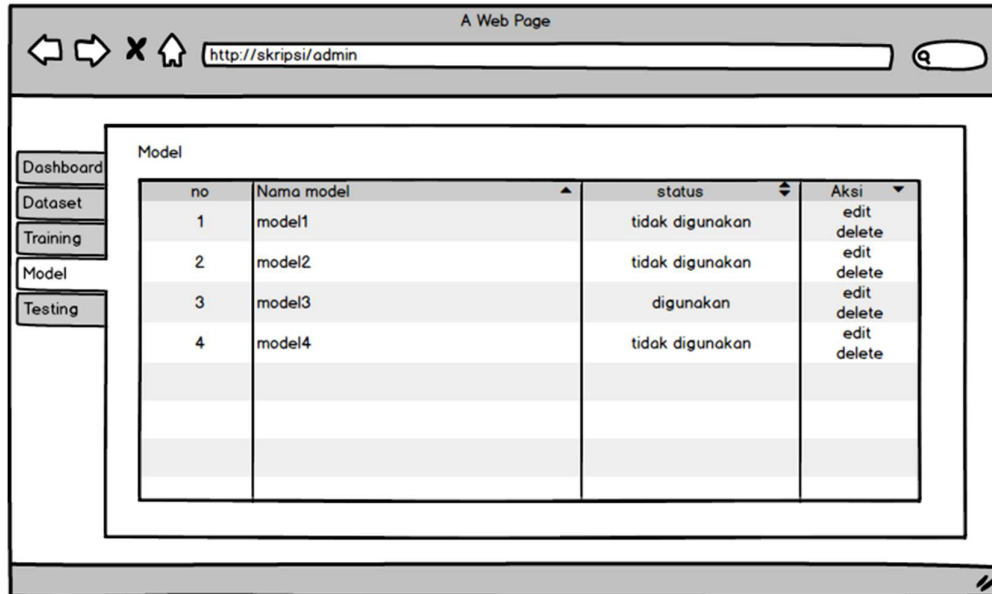
Log

Perhitungan

Gambar IV.16 Gambar desain halaman simpan model

Pada Halaman Hasil training, admin dapat menyimpan model yang dihasilkan oleh hasil training dengan cara mengisi form nama model dan menekan tombol simpan.

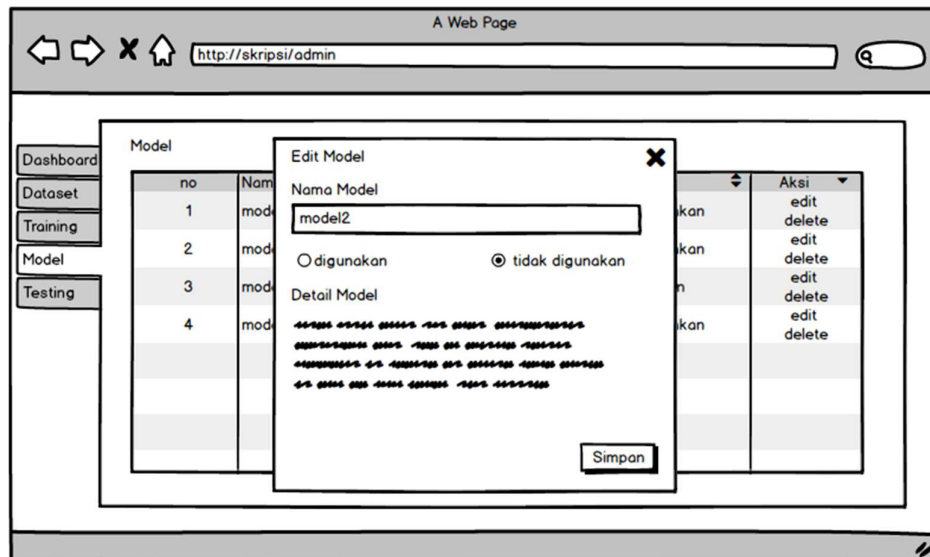
f. Halaman Model



Gambar IV.17 Gambar Desain Halaman model

Pada halaman model pengguna akan diberikan daftar model yang ada pada database dalam bentuk tabel.

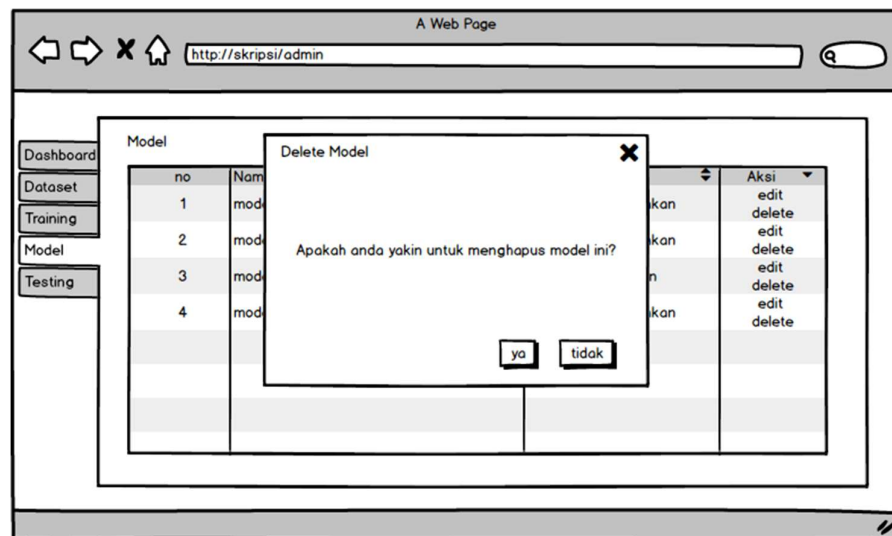
g. Tampilan Edit Model



Gambar IV.18 Gambar Desain tampilan edit model

Di atas merupakan desain tampilan untuk proses edit model. Akan muncul sebuah dialog yang berisi form yang nilainya dapat diubah oleh admin bila admin menekan tombol edit model.

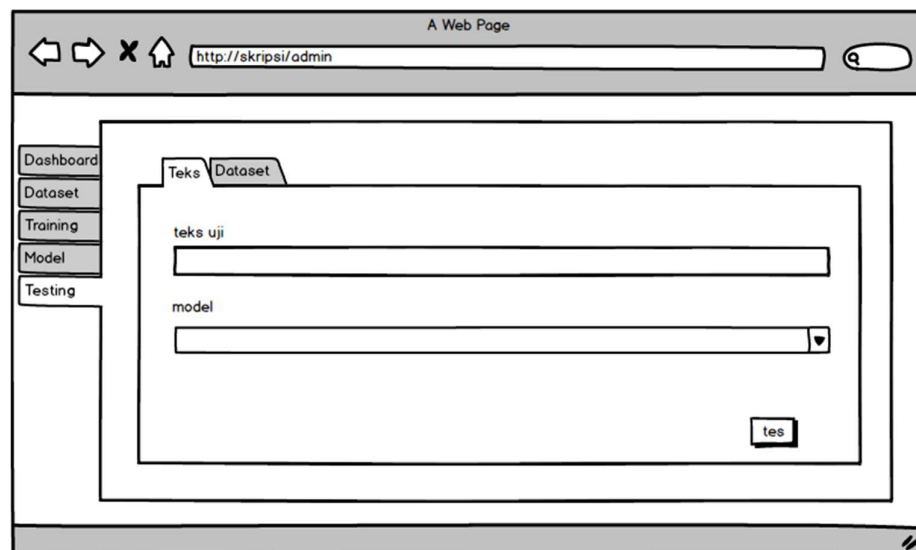
h. Tampilan Hapus Model



Gambar IV.19 Gambar desain tampilan hapus model

Di atas merupakan desain tampilan untuk proses hapus model. Akan muncul sebuah dialog konfirmasi bila admin menekan tombol *delete* model.

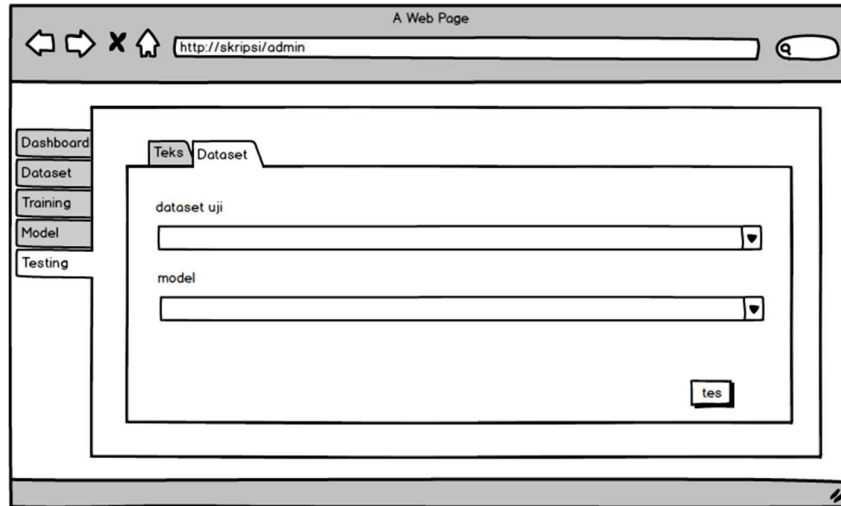
i. Halaman Testing



Gambar IV.20 Desain Halaman Testing

Pada halaman testing admin akan langsung di berikan tampilan berupa tab yang berisi form pengujian teks seperti diatas

j. Tampilan Uji Dataset

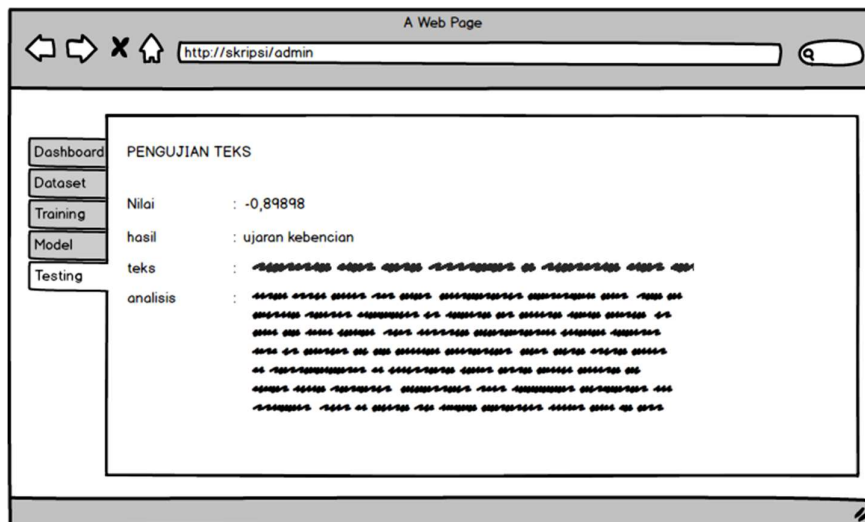


Gambar IV.21 Desain tampilan uji dataset

Untuk uji dataset, tampilan akan menyerupai uji teks. Namun, form yang perlu diisi bukan berupa masukan teks melainkan pilihan dataset.

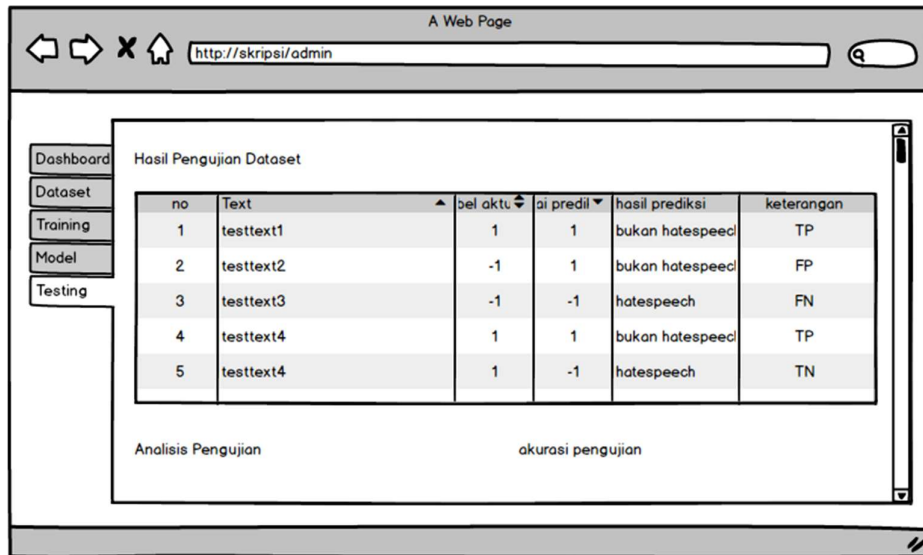
k. Halaman Hasil Uji Teks

Pada halaman hasil uji teks akan ditampilkan data pengujian dari teks yang dimasukkan oleh admin.



Gambar IV.22 Desain halaman uji teks

1. Halaman Hasil Uji Dataset

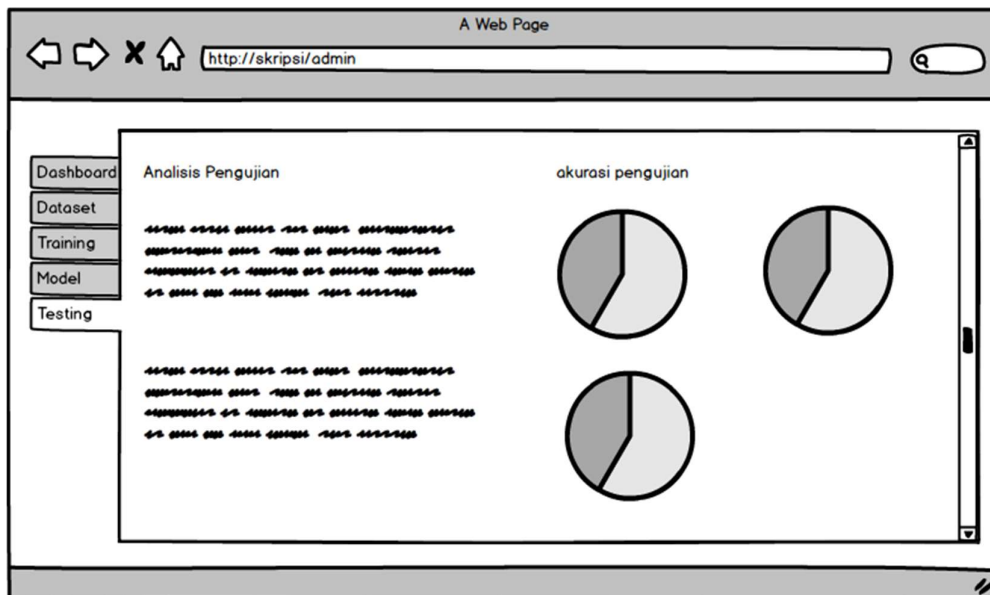


no	Text	bel aktu	ai predil	hasil prediksi	keterangan
1	testtext1	1	1	bukan hatespeech	TP
2	testtext2	-1	1	bukan hatespeech	FP
3	testtext3	-1	-1	hatespeech	FN
4	testtext4	1	1	bukan hatespeech	TP
5	testtext4	1	-1	hatespeech	TN

Analisis Pengujian akurasi pengujian

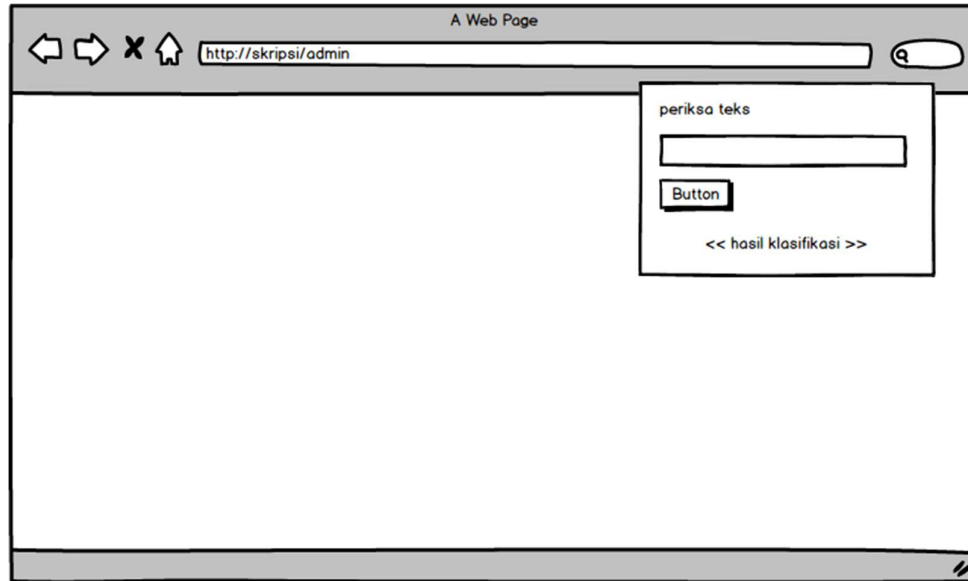
Gambar IV.23 Gambar desain halaman uji dateset

Pada halaman hasil uji dataset, hasil pengujian akan ditampilkan dalam bentuk tabel seperti di atas. Selain itu terdapat juga indikator pengujian lain seperti tingkat akurasi yang akan ditampilkan dalam bentuk chart.



Gambar IV.24 Desain lanjutan halaman uji dataset

m. Tampilan end user



Gambar IV.25 Desain tampilan plugin deteksi

Sebuah dialog akan muncul pada browser jika pengguna telah memasang plugin deteksi ini. Dialog yang muncul akan berisi form yang dapat pengguna isi dengan teks yang akan dievaluasi

4.4 Algoritma klasifikasi

Bagian ini akan menjelaskan tentang proses yang akan dilalui untuk dapat mengklasifikasikan sebuah teks ujaran kebencian.

4.4.1 Stemming dan Stopword-removal

Untuk proses pembelajaran hal pertama yang dilakukan adalah menghilangkan kata hubung (*stopword removal*) dan mengubah kata pada kalimat menjadi kata dasar (*stemming*). Misal terdapat data teks dengan label seperti berikut:

Tabel IV.3 Contoh Data klasifikasi

Nomor	Teks	Label
D1	Ketika pabrik Hoax bicara Hoax.	1
D2	Ya Tuhan, ampuni semua dosa-dosa nya dan dosa kami	1
D3	Partai kafir calonnya kafir haram di pilih umat muslim kecuali kafir	-1
D4	INDONESIA NEGARA IBLIS	-1
D5	Tapi lebih busuk mulut mu ahoq	-1

Pada data di atas label bernilai 1 diberikan pada data yang bukan merupakan ujaran kebencian sementara label bernilai -1 diberikan pada data yang merupakan ujaran kebencian. Teks pada data di atas kemudian akan mengalami proses stemming dimana kata pada kalimat akan diubah ke dalam bentuk bakunya, misal *ampuni* menjadi *ampun* dan *calonnya* menjadi *calon*. Kemudian teks akan melalui proses *stopword-removal* untuk menghilangkan kata hubung seperti *di*, *dan*, *yang*. Setelah melalui proses tersebut maka teks yang dihasilkan akan berubah seperti berikut

Tabel IV.4 hasil stemming dan stopwords-removal pada data contoh

Nomor	Teks	Label
D1	pabrik hoax bicara hoax	1
D2	tuhan ampun semua dosa nya dosa	1
D3	partai kafir calon kafir haram pilih umat muslim kafir	-1
D4	indonesia negara iblis	-1
D5	lebih busuk mulut mu ahoq	-1

4.4.2 Term weighting TF-IDF

Setelah data melalui proses stemming dan stopwords removal, data kemudian akan di hitung bobot tiap katanya menggunakan TF-IDF. Berikut adalah

proses pembobotan TF-IDF dengan menggunakan data yang telah diproses sebelumnya.

Tabel IV.5 Tabel perhitungan TF

No	term	Term Count ($f_{t,d}$)					Term Frequency ($\log(1 + f_{t,d})$)				
		D1	D2	D1	D2	D5	D1	D2	D3	D4	D5
1	pabrik	1	0	0	0	0	0.30103	0	0	0	0
2	hoax	2	0	0	0	0	0.47712	0	0	0	0
3	bicara	1	0	0	0	0	0.30103	0	0	0	0
4	tuhan	0	1	0	0	0	0	0.30103	0	0	0
5	ampun	0	1	0	0	0	0	0.30103	0	0	0
6	semua	0	1	0	0	0	0	0.30103	0	0	0
7	dosa	0	2	0	0	0	0	0.47712	0	0	0
8	nya	0	1	0	0	0	0	0.30103	0	0	0
9	partai	0	0	1	0	0	0	0	0.30103	0	0
10	kafir	0	0	3	0	0	0	0	0.60206	0	0
11	calon	0	0	1	0	0	0	0	0.30103	0	0
12	haram	0	0	1	0	0	0	0	0.30103	0	0
13	pilih	0	0	1	0	0	0	0	0.30103	0	0
14	umat	0	0	1	0	0	0	0	0.30103	0	0
15	muslim	0	0	1	0	0	0	0	0.30103	0	0
16	indonesia	0	0	0	1	0	0	0	0	0.30103	0
17	negara	0	0	0	1	0	0	0	0	0.30103	0
18	iblis	0	0	0	1	0	0	0	0	0.30103	0
19	lebih	0	0	0	0	1	0	0	0	0	0.30103
20	busuk	0	0	0	0	1	0	0	0	0	0.30103
21	mulut	0	0	0	0	1	0	0	0	0	0.30103
22	mu	0	0	0	0	1	0	0	0	0	0.30103
23	ahq	0	0	0	0	1	0	0	0	0	0.30103

Diatas merupakan perhitungan TF. Berikutnya menghitung nilai IDF suatu dokumen dan TFIDF semua kata untuk setiap dokumen seperti pada tabel berikut :

Tabel IV.6 Tabel perhitungan IDF dan TFIDF

term	DF (n_i)	IDF ($\log(1+(N/n_i))$)	TF-IDF (TF*IDF)				
			D1	D2	D3	D4	D5
pabrik	1	0.77815125	0.23425	0	0	0	0
Hoax	1	0.77815125	0.37127	0	0	0	0
Bicara	1	0.77815125	0.23425	0	0	0	0
Tuhan	1	0.77815125	0	0.23425	0	0	0
ampun	1	0.77815125	0	0.23425	0	0	0
semua	1	0.77815125	0	0.23425	0	0	0
Dosa	1	0.77815125	0	0.37127	0	0	0
Nya	1	0.77815125	0	0.23425	0	0	0
Partai	1	0.77815125	0	0	0.23425	0	0
Kafir	1	0.77815125	0	0	0.46849	0	0
Calon	1	0.77815125	0	0	0.23425	0	0
Haram	1	0.77815125	0	0	0.23425	0	0
Pilih	1	0.77815125	0	0	0.23425	0	0
Umat	1	0.77815125	0	0	0.23425	0	0
muslim	1	0.77815125	0	0	0.23425	0	0
indonesia	1	0.77815125	0	0	0	0.23425	0
negara	1	0.77815125	0	0	0	0.23425	0
Iblis	1	0.77815125	0	0	0	0.23425	0
Lebih	1	0.77815125	0	0	0	0	0.23425
Busuk	1	0.77815125	0	0	0	0	0.23425
Mulut	1	0.77815125	0	0	0	0	0.23425
mu	1	0.77815125	0	0	0	0	0.23425
Ahoq	1	0.77815125	0	0	0	0	0.23425

4.4.3 Training dan klasifikasi SVM

Dari hasil pembobotan TF-IDF, data yang didapatkan akan bisa digunakan untuk proses training maupun prediksi menggunakan teknik klasifikasi SVM. Pada proses training SVM algoritma yang digunakan adalah algoritma *SMO* yang dikenalkan oleh *Platt* [16], [18].

Pertama kita perlu menentukan dua buah pengali lagrange α_i dan α_j yang akan dievaluasi dan dilakukan optimalisasi. Pada algoritma yang menerapkan SMO secara penuh, teknik *heuristic* akan digunakan untuk menentukan α mana yang akan dievaluasi. Namun, pada implemementasi simple SMO kita cukup melakukan iterasi untuk α_i dimana $i=1, \dots$ banyaknya data. Kemudian jika α_i tidak memenuhi kondisi KKT(Karush-Kuhn-Tucker) kita akan memilih α_j secara acak.

Kemudian kita harus mencari nilai L (low) dan H (hi) yang merupakan variabel batasan yang memenuhi $L \leq \alpha_j \leq H$ yang juga memenuhi $0 \leq \alpha_j \leq C$. Sesuai dengan ketentuan berikut :

- Jika $y^i \neq y^j$, $L = \max(0, \alpha_j - \alpha_i)$, $H = \min(C, C + \alpha_j - \alpha_i)$ (1)
- Jika $y^i = y^j$, $L = \max(0, \alpha_i + \alpha_j - C)$, $H = \min(C, \alpha_i + \alpha_j)$ (2)

Jika nilai α_j tidak memenuhi batasan terhadap L dan H , maka kita akan menghitung nilai α_j yang baru dengan ketentuan sebagai berikut :

$$\alpha_j := \alpha_j - \frac{y^j(E_i - E_j)}{\eta} \quad (3)$$

Dimana

$$E_i = f(x^i) - y^i \quad (4)$$

$$\eta = 2\langle x^i, x^j \rangle - \langle x^i, x^i \rangle - \langle x^j, x^j \rangle \quad (5)$$

$$f(x) = \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle + b \quad (6)$$

Ketika menghitung nilai η kita dapat menggunakan fungsi kernel yang kita inginkan untuk menggantikan inner product pada persamaan (5). Berikutnya kita menentukan nilai baru dari α_j agar nilai berada diantara batas $[L, H]$.

$$\alpha_j := \begin{cases} H, & \text{jika } \alpha_j > H \\ \alpha_j, & \text{jika } L < \alpha_j < H \\ L, & \text{jika } \alpha_j < L \end{cases} \quad (7)$$

Pada kondisi umum Persamaan (3) dan (7) berlaku jika nilai η bernilai lebih kecil dari 0. Sementara itu pada kondisi khusus dimana nilai $\eta \geq 0$ maka kita harus mencari nilai dari fungsi objektif L dan H sebagai berikut :

$$f_i = y^i(E_i + b) - \alpha_i K(x^i, x^i) - y^i y^j \alpha_j K(x^i, x^j) \quad (8)$$

$$f_j = y^j(E_j + b) - y^i y^j \alpha_i K(x^i, x^j) - \alpha_j K(x^j, x^j) \quad (9)$$

$$L_i = \alpha_i + y^i y^j (\alpha_j - L) \quad (10)$$

$$H_i = \alpha_i + y^i y^j (\alpha_j - H) \quad (11)$$

$$L_{obj} = L_i f_i + L f_2 + \frac{1}{2} L_i^2 K(x^i, x^i) + \frac{1}{2} L^2 K(x^j, x^j) + y^i y^j L L_i (x^i, x^j) \quad (12)$$

$$H_{obj} = H_i f_i + H f_2 + \frac{1}{2} H_i^2 K(x^i, x^i) + \frac{1}{2} H^2 K(x^j, x^j) + y^i y^j H H_i (x^i, x^j) \quad (13)$$

$K(x^i, x^j)$ merupakan fungsi kernel pada x^i, x^j Setelah kita mencari nilai objektif L dan H maka kita menentukan nilai baru dari α_j berdasarkan kondisi berikut :

$$\alpha_j := \begin{cases} H, & \text{jika } L_{obj} > H_{obj} \\ \alpha_j, & \text{jika } L_{obj} = H_{obj} \\ L, & \text{jika } L_{obj} < H_{obj} \end{cases} \quad (14)$$

Setelah menentukan nilai α_j kita akan menentukan nilai α_i dari α_i dengan ketentuan sebagai berikut:

$$\alpha_i := \alpha_i + y^i y^j (\alpha_j^{(lama)} - \alpha_j) \quad (15)$$

Dimana $\alpha_j^{(lama)}$ merupakan nilai α_j sebelum dioptimisasi oleh persamaan (3) dan (7). Berikutnya , kita perlu menghitung nilai bias b dengan ketentuan sebagai berikut:

$$b := \begin{cases} b_1 & \text{jika } 0 < \alpha_i < C \\ b_2 & \text{jika } 0 < \alpha_j < C \\ \frac{b_1 + b_2}{2} & \text{jika selainnya} \end{cases} \quad (16)$$

Dimana

$$b_1 = b - E_i - y^i(\alpha_i - \alpha_i^{(lama)})\langle x^i, x^i \rangle - y^j(\alpha_j - \alpha_j^{(lama)})\langle x^i, x^j \rangle \quad (17)$$

$$b_2 = b - E_j - y^i(\alpha_i - \alpha_i^{(lama)})\langle x^i, x^j \rangle - y^j(\alpha_j - \alpha_j^{(lama)})\langle x^j, x^j \rangle \quad (18)$$

Berikut adalah notasi algoritma untuk keseluruhan proses yang telah dijelaskan di atas.

INPUT

- C: parameter regularisasi
- tol: variabel toleransi numerik
- numpasses : banyak nya iterasi terhadap α tanpa terjadi perubahan
- $(x^i, y^i), \dots, (x^n, y^n)$: data training

OUTPUT

- α : pengali lagrange
- b : nilai bias

PROSES

- Alpha[1...n] \leftarrow 0 , b \leftarrow 0, passes \leftarrow 0
- **while** (passes < numpasses) do
 - num_changed_alpha \leftarrow 0
 - **for** i [1...n]; i+1
 - $E_i \leftarrow f(x^i) - y^i$
 - **if** (($y^i E_i < -\text{tol} \ \&\& \ \alpha[i] < C$) || ($y^i E_i > \text{tol} \ \&\& \ \alpha[i] > 0$))
 - pilih j \neq i secara acak
 - $E_j \leftarrow f(x^j) - y^j$
 - old_ai \leftarrow alpha[i] , old_aj \leftarrow alpha[j]
 - hitung L dan H berdasarkan persamaan (1) dan (2)
 - **if** (L==H)
 - **continue**
 - hitung Eta (η) berdasarkan persamaan (5)
 - **if** (Eta \geq 0)
 - hitung nilai baru dari alpha[j] menggunakan persamaan (8) hingga (14)

- **else**
 - hitung nilai baru dari $\alpha[j]$ menggunakan persamaan (3) dan (7)
 - $\text{if}(\text{abs}(\alpha[j] - \text{old_aj}) < 10^{-4})$
 - **continue**
 - tentukan nilai $\alpha[i]$ berdasarkan persamaan (15)
 - hitung b_1 dan b_2 berdasarkan persamaan (17) dan (18)
 - hitung b berdasarkan (16)
 - $\text{num_changed_alpha}++$
 - **endif**
- **endfor**
- **if** ($\text{num_changed_alpha} == 0$)
 - $\text{passes}++$
- **else**
 - $\text{passes} \leftarrow 0$
- **endwhile**

Dari algoritma di atas, menggunakan data contoh yang sudah diolah sebagai data training dengan $C = 5$, $\text{tol} = 0,0001$ dan $\text{numpasess} = 2000$ nilai b yang didapat adalah -0.2610 dan nilai α sebagai berikut :

Tabel IV.7 hasil perhitungan alpha

Dokumen	D1	D2	D3	D4	D5
Alpha	5	3.529209	1.346615	4.489122	2.693473

Kemudian untuk mengklasifikasikan suatu data kita dapat menggunakan persamaan (6). Berikut adalah proses klasifikasi dengan menggunakan D2 sebagai data yang akan di prediksi kategori nya.

$$f(x) = (5*1*(0,23425*0+0.37127*0+0.23425*0+0*0.23425+0*0.23425$$

$$\begin{aligned}
& 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0)) + (3,529209*1*(0*0,0 + \\
& 0*0 + 0*0 + 0.23425*0.23425 + 0.23425*0.23425 + 0.23425*0.23425 + 0.37127 \\
& *0.37127 + 0.23425*0.23425 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + \\
& 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0)) + (1,346615*-1*(0*0 + 0*0 + 0*0 + \\
& 0*0.23425 + 0*0.23425 + 0*0.23425 + 0*0.37127 + 0*0.23425 + 0.23425*0 + \\
& 0.46849*0 + 0.23425*0 + 0.23425*0 + 0.23425*0 + 0.23425*0 + 0.23425*0 + 0*0 \\
& + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0)) + (4,489122*-1*(0*0 + 0*0 + 0*0 + \\
& 0*0.23425 + 0*0.23425 + 0*0.23425 + 0*0.37127 + 0*0.23425 + 0*0 + 0*0 + 0*0 \\
& + 0*0 + 0*0 + 0*0 + 0*0 + 0.23425*0 + 0.23425*0 + 0.23425*0 + 0*0 + 0*0 + 0*0 \\
& + 0*0 + 0*0)) + (2,693473*-1*(0*0 + 0*0 + 0*0 + 0*0.23425 + 0*0.23425 + \\
& 0*0.23425 + 0*0.37127 + 0*0.23425 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 \\
& + 0*0 + 0*0 + 0*0 + 0.23425*0 + 0.23425*0 + 0.23425*0 + 0.23425*0 + \\
& 0.23425*0)) + (-0.2610)
\end{aligned}$$

$$f(x) = 0 + 1,26109 + 0 + 0 + 0 - 0,2610$$

$$f(x) = 1,00009$$

Dari hasil diatas nilai $f(x)$ adalah positif ($f(x) > 0$) maka label prediksi untuk D2 adalah 1 atau bukan ujaran kebencian.

BAB V

HASIL DAN PENGUJIAN

5.1 Source Code

Dari sistem yang dibuat dihasilkan library svm, tfidf dan stemming yang akan membantu melakukan klasifikasi. Berikut dipaparkan source code dari library tersebut.

5.1.1 library SVM

Berikut adalah source code library yang melakukan proses pembelajaran dan klasifikasi menggunakan teknik svm

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Svmlib
{
    protected $alpha;
    protected $b;
    protected $D;
    protected $data;
    protected $kernel;
    protected $kernelResults;
    protected $kernelType;
    protected $labels;
    protected $N;
    protected $sew_;
    protected $gamma;
    protected $w;

    public function train($data, $labels, $options = [])
    {
        // menyimpan data ke atribut class untuk fungsi bantuan lain
        $this->data = $data;
        $this->labels = $labels;
        $this->gamma = array_key_exists('gamma', $options) ? $options['gamma'] : 1;
        // parameters
        // hyperparameter C. kurangi untuk lebih banyak regularisasi
        $C = array_key_exists('C', $options) ? $options['C'] : 1.0;
        // toleransi numerik. Don't touch unless you're pro
        $tol = array_key_exists('tol', $options) ? $options['tol'] : 1e-4;
        // toleransi untuk alpha
        $alphatol = array_key_exists('alphatol', $options) ? $options['alphatol'] : 1e-7; //
        epsilon
    }
}
```

```

// banyaknya iterasi
$maxiter = array_key_exists('maxiter', $Options) ? $Options['maxiter'] : 10000;
// banyaknya data yang tidak dicek dan tidak berubah nilai sebelum proses
berhenti
$numpasses = array_key_exists('numpasses', $Options) ? $Options['numpasses'] :
20;

// inisialisasi kernel
$this->kernelType = array_key_exists('kernel', $Options) ? $Options['kernel'] :
"linear";
$kernel = $this->select_kernel($this->kernelType);

if (array_key_exists('kernel', $Options)) {
    if (is_string($Options['kernel'])) {
        // kernel merupakan string
        if ($Options['kernel'] === 'linear') {
            $kernel = [$this, 'linearKernel'];
            $this->kernelType = 'linear';
        }
    }

    if (is_callable($Options['kernel'])) {
        // kernel merupakan fungsi
        $kernel = $Options['kernel'];
        $this->kernelType = 'custom';
    }
}

// inisialisasi
$this->kernel = $kernel;
$this->N = $N = count($data);
$this->D = $D = count($data[0]);
$this->alpha = array_fill(0, $N, 0);
$this->b = 0.0;
$this->usew_ = false;

// perhitungan Cache kernel untuk menghindari perhitungan ulang.
// dapat menghabiskan banyak memori jika nilai N sangat besar.
if (array_key_exists('memoize', $Options) && $Options['memoize']) {
    $this->kernelResults = array_fill(0, $N, []);

    for ($i = 0; $i < $N; $i++) {
        $this->kernelResults[$i] = array_fill(0, $N, []);

        for ($j = 0; $j < $N; $j++) {
            $this->kernelResults[$i][$j] = $kernel($data[$i], $data[$j]);
        }
    }
}

```

```

// menjalankan SMO
$iter = 0;
$passes = 0;

while ($passes < $numpasses && $iter < $maxiter) {
    $alphaChanged = 0;

    for ($i = 0; $i < $N; $i++) {
        $Ei = $this->marginOne($data[$i]) - $labels[$i];

        if (($labels[$i] * $Ei < -$tol && $this->alpha[$i] < $C)
            || ($labels[$i] * $Ei > $tol && $this->alpha[$i] > 0)
        ) {
            // alpha_i perlu diperbarui!
            // pilih j != i secara random untuk memperbarui nilai alpha
            $j = $i;

            while ($j == $i) {
                $j = rand(0, $this->N - 1);
            }

            $Ej = $this->marginOne($data[$j]) - $labels[$j];

            // hitung batas L and H untuk j
            $ai = $this->alpha[$i];
            $aj = $this->alpha[$j];
            $L = 0;
            $H = $C;

            if ($labels[$i] == $labels[$j]) {
                $L = max(0, $ai + $aj - $C);
                $H = min($C, $ai + $aj);
            } else {
                $L = max(0, $aj - $ai);
                $H = min($C, $C + $aj - $ai);
            }

            if (abs($L - $H) < 1e-4) {
                continue;
            }

            $eta = (2 * $this->kernelResult($i, $j)) - $this->kernelResult($i, $i) -
                $this->kernelResult($j, $j);
            $S = $labels[$i] * $labels[$j];

            if ($eta >= 0) {

                // fungsi objektif untuk menghitung new alpha_j
                // jika ternyata eta >= 0

```



```

    $f1 = $labels[$i] * ($Ei + $this->b) - $this->alpha[$i]*$this-
>kernelResult($i, $i) - $S* $this->alpha[$j] * $this->kernelResult($i, $j);
    $f2 = $labels[$j] * ($Ej + $this->b) - $S* $this->alpha[$i]*$this-
>kernelResult($i, $j) - $this->alpha[$j] * $this->kernelResult($j, $j);

    $Li = $this->alpha[$i] + $S * ($this->alpha[$j] - $L);
    $Hi = $this->alpha[$i] + $S * ($this->alpha[$j] - $H);

    $Lobj = $Li * $f1 + $L*$f2 + (pow($Li, 2)* $this->kernelResult($i,
$i))/2 + (pow($L, 2)* $this->kernelResult($j, $j))/2 + $S * $L * $Li * $this-
>kernelResult($i, $j);
    $Hobj = $Hi * $f1 + $H*$f2 + (pow($Hi, 2)* $this->kernelResult($i,
$i))/2 + (pow($H, 2)* $this->kernelResult($j, $j))/2 + $S * $H * $Hi * $this-
>kernelResult($i, $j);

    if($Lobj>$Hobj + $alphatol)
    {
        $newaj = $H;
    }elseif ($Lobj<$Hobj - $alphatol)
    {
        $newaj = $L;
    }else
    {
        $newaj = $aj;
    }

} else
{
    $newaj = $aj - (($labels[$j] * ($Ei - $Ej)) / $eta);

    if ($newaj > $H) {
        $newaj = $H;
    }

    if ($newaj < $L) {
        $newaj = $L;
    }

}

// kemudian menghitung alpha_i berdasarkan new alpha_j.

if (abs($aj - $newaj) < $tol) {
    continue;
}

$this->alpha[$j] = $newaj;
$newai = $ai + $S * ($aj - $newaj);
$this->alpha[$i] = $newai;

```

```

        // update nilai bias
        $b1 = $this->b - $Ei - $labels[$i] * ($newai - $ai) * $this-
>kernelResult($i, $i)
        - $labels[$j] * ($newaj - $aj) * $this->kernelResult($i, $j);

        $b2 = $this->b - $Ej - $labels[$i] * ($newai - $ai) * $this-
>kernelResult($i, $j)
        - $labels[$j] * ($newaj - $aj) * $this->kernelResult($j, $j);

        $this->b = 0.5 * ($b1 + $b2);

        if ($newai > 0 && $newai < $C) {
            $this->b = $b1;
        }

        if ($newaj > 0 && $newaj < $C) {
            $this->b = $b2;
        }

        $alphaChanged++;
    }
}

$iter++;
$passes = ($alphaChanged == 0) ? $passes + 1 : 0;
}

// jika menggunakan kernel linear
// maka nilai w akan dihitung dan mengubah flag usew_ menjadi true
// sebagai tanda bahwa menggunakan nilai w untuk klasifikasi
if ($this->kernelType === 'linear') {
    // compute weights and store them
    $this->w = array_fill(0, $this->D, 0);

    for ($j = 0; $j < $this->D; $j++) {
        $s = 0.0;

        for ($i = 0; $i < $this->N; $i++) {
            $s += $this->alpha[$i] * $labels[$i] * $data[$i][$j];
        }

        $this->w[$j] = $s;
        $this->usew_ = true;
    }
} else {
    // jika kernel yang digunakan bukan linear
    // kita perlu menyimpan data yang merupakan support vektor
    // berdasarkan nilai alpha[i] yang masih dalam batas alphasol
    $newdata = [];

```

```

$newlabels = [];
$newalpha = [];

for ($i = 0; $i < $this->N; $i++) {
    if ($this->alpha[$i] > $alphatol) {
        $newdata[] = $this->data[$i];
        $newlabels[] = $this->labels[$i];
        $newalpha[] = $this->alpha[$i];
    }
}

// store data and labels
$this->data = $newdata;
$this->labels = $newlabels;
$this->alpha = $newalpha;
$this->N = count($this->data);
}

return $this->get_model();
}

public function select_kernel($kernel)
{
    if($kernel == "linear")
    {
        return [$this, 'linearKernel'];
    } else if ($kernel == "rbf")
    {
        return [$this, 'rbfKernel'];
    }
}

// inst merupakan array[D]. mengembalikan margin dari argumen yang diberikan

public function marginOne($inst)
{
    $f = $this->b;

    // jika kernel linear digunakan dan bobot w sudah dihitung dan disimpan
    // atribut usew_ akan bernilai true.
    if ($this->usew_) {
        // jika usew_ bernilai true hitung margin berdasarkan persamaan
        // margin = w.inst + b
        for ($j = 0; $j < $this->D; $j++) {
            $f += $inst[$j] * $this->w[$j];
        }
    } else {
        // jika tidak menggunakan w maka menggunakan persamaan
        // margin = alpha[i] * y[i] * K(x,x[i]) + b
        if(is_callable($this->kernel))

```

```

        {
            $kernel = $this->kernel;
        } else
        {
            $kernel = $this->select_kernel($this->kernelType);
        }
        for ($i = 0; $i < $this->N; $i++) {

            $f += $this->alpha[$i] * $this->labels[$i] * $kernel($inst, $this->data[$i]);
        }
    }
    return $f;
}

public function predictOne($inst)
{
    return $this->marginOne($inst) > 0 ? 1 : -1;
}

// data merupakan array NxD . mengembalikan array of margins.
public function margins($data)
{
    $N = count($data);
    $margins = array_fill(0, $N, 0);
    for ($i = 0; $i < $N; $i++) {
        $margins[$i] = $this->marginOne($data[$i]);
    }
    return $margins;
}

protected function kernelResult($i, $j)
{
    if ($this->kernelResults) {
        return $this->kernelResults[$i][$j];
    }
    $kernel = $this->kernel;

    return $kernel($this->data[$i], $this->data[$j]);
}

// data merupakan array NxD array. mengembalikan array of 1 atau -1 sebagai
prediksi
public function predict($data)
{
    $margs = $this->margins($data);
    for ($i = 0; $i < count($margs); $i++) {
        $margs[$i] = $margs[$i] > 0 ? 1 : -1;
    }

    return $margs;
}

```

```

}

protected function linearKernel($v1, $v2)
{
    $s = 0;

    for ($q = 0; $q < count($v1); $q++) {
        $s += $v1[$q] * $v2[$q];
    }
    return $s;
}

protected function rbfKernel($v1, $v2)
{
    $s = 0;

    for ($q = 0; $q < count($v1); $q++) {
        $s += pow(($v1[$q] - $v2[$q]), 2);
    }
    $s = exp(-($this->gamma) * $s);
    return $s;
}

public function get_model()
{
    $model = array(
        "alpha" => $this->alpha,
        "b" => $this->b,
        "D" => $this->D,
        "kernelResults" => $this->kernelResults,
        "kernelType" => $this->kernelType,
        "labels" => $this->labels,
        "N" => $this->N,
        "usew_" => $this->usew_,
        "gamma" => $this->gamma,
    );
    if ($this->usew_ == true)
    {
        $model["w"] = $this->w;
    }
    else
    {
        $model["data"] = $this->data;
    }
    return $model;
}

public function set_model($array)
{

```

```

$this->alpha = $array['alpha'];
$this->b = $array['b'];
$this->D = $array['D'];
$this->kernelResults = $array['kernelResults'];
$this->kernelType = $array['kernelType'];
$this->kernel = $this->select_kernel($this->kernelType);
$this->labels = $array['labels'];
$this->N = $array['N'];
$this->usew_ = $array['usew_'];
$this->gamma = array_key_exists('gamma', $array) ? $array['gamma'] : null;
$this->w = array_key_exists('w', $array) ? $array['w'] : null;
$this->data = array_key_exists('data', $array) ? $array['data'] : null;
    }
}
?>

```

5.1.2 library TF-IDF

Berikut adalah source code untuk library yang menjalankan proses perhitungan tf-idf

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class TFIDF
{
    private $TFIDF;
    private $TF;
    private $IDF;
    private $term;
    public function calculate($dataQuery)
    {
        // TF-IDF
        $alltoken = [];
        $doccount = 0; ///banyak nya seluruh dokumen

        foreach ($dataQuery as $key ) {
            $doccount++;
            $tokenDoc = explode(" ", $key['data_text']);
            $alltoken = array_merge($alltoken, $tokenDoc);
        }
        $totalterm = array_count_values($alltoken); ///term yang ada pada teks
        $TF = []; ///nilai TF seluruh dokumen
        $TFraw = [];
        foreach ($dataQuery as $key ) {
            $TFdoc = []; ///nilai TF pada satu dokumen

```

```

        $tokenDoc = array_count_values(explode(" ",
$key['data_text']));

        foreach ($totalterm as $key => $value) {
            if(array_key_exists($key, $tokenDoc))
            {
                $TFdoc[$key] = $tokenDoc[$key];
                $TFlog[$key] = log(1 + $tokenDoc[$key],10);
            }else
            {
                $TFdoc[$key] = 0;
                $TFlog[$key] = log(1 + 0);
            }
        }
        array_push($TF, $TFlog);
        array_push($TFraw, $TFdoc);
    }
    $IDF = [];
    $DF = [];
    foreach ($totalterm as $key => $value) {
        $df = 0;
        foreach ($TFraw as $key2) {
            if($key2[$key]>0)
            {
                $df++;
            }
        }
        $DF[$key] = $df;
        $IDF[$key] = log(1+($doccount/$df),10);
    }

    $TFIDF = [];
    $i =0;
    $y=0;
    foreach($TF as $key => $value){
        $TFIDFdoc = [];
        $y=0;
        foreach ($value as $key1 => $value1) {
            $TFIDFdoc[$y] = $TF[$key][$key1]*$IDF[$key1];
            $y++;
        }
        $TFIDF[$i] = $TFIDFdoc;
        $i++;
    }

    $this->TFIDF = $TFIDF;
    $this->TF = $TF;
    $this->IDF = $IDF;
    $this->term = $totalterm;
    return $TFIDF;
}

```

```

function get_tf()
{
    return $this->TF;
}
function get_idf()
{
    return $this->IDF;
}
function get_tfidf()
{
    return $this->TFIDF;
}
function get_term()
{
    return $this->term;
}
}

```

5.1.3 library stemming

Berikut adalah source code untuk library yang mengatur proses stemming dan tokenisasi

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
require_once FCPATH. '/vendor/autoload.php';
class TokenStemmer {

    private $stemmer;
    private $stopWord;
    function __construct(){
        $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
        $this->stemmer = $stemmerFactory->createStemmer();
        $stopwordFactory = new
\Sastrawi\StopWordRemover\StopWordRemoverFactory();
        $this->stopWord = $stopwordFactory->createStopWordRemover();
    }

    public function stem($text)
    {
        return $this->stemmer->stem($text);
    }

    public function batch_stem($arr, $index)
    {
        foreach ($arr as $key => $value) {
            foreach ($value as $key2 => $value2) {

```



```

        if($index == $key2)
        {
            $arr[$key][$key2] = $this->stemmer-
>stem($value2);
        }
    }
    }
    return $arr;
}

public function stopword($text)
{
    return $this->stopWord->remove($text);
}

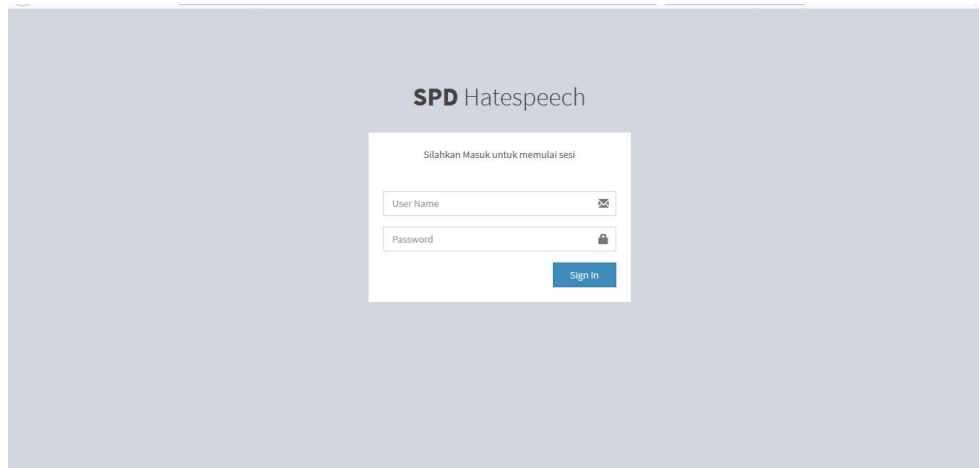
public function batch_stopword($arr, $index)
{
    foreach ($arr as $key => $value) {
        foreach ($value as $key2 => $value2) {
            if($index == $key2)
            {
                $arr[$key][$key2] = $this->stopWord-
>remove($value2);
            }
        }
    }
    return $arr;
}
}

```

5.2 Antarmuka Sistem

Dari penelitian ini dihasilkan beberapa hal yang salah satunya adalah sebuah sistem yang dapat digunakan oleh seorang pengguna untuk memanfaatkan sistem peringatan dini serta mengatur model yang digunakan. Untuk menjelaskan sistem yang telah dibuat, tangkapan layar dari sistem yang dibuat akan diberikan disertai dengan penjelasan bagaimana sistem tersebut dijalankan sebagai berikut.

a. Melakukan Login



Gambar V.1 Tampilan Halaman Login

Langkah-langkah untuk melakukan login :

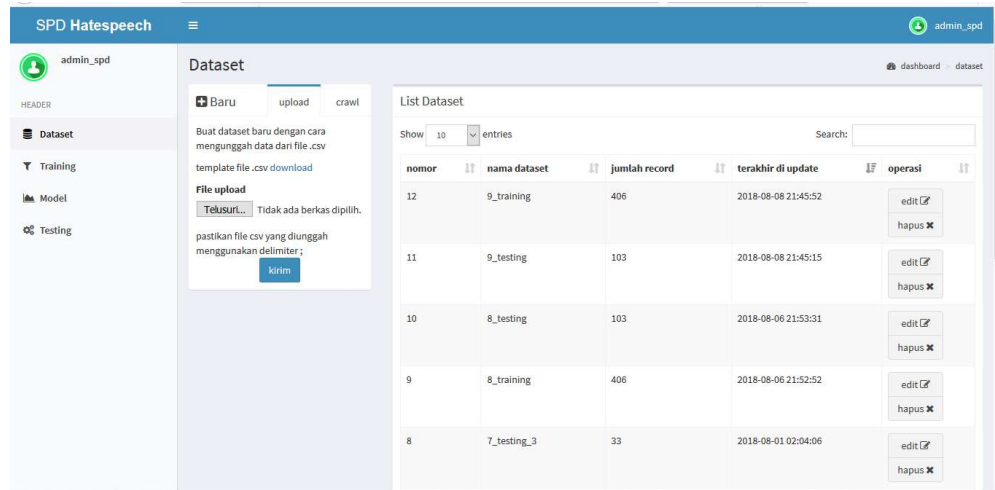
1. Membuka browser
2. Mengunjungi halaman web dengan menuliskan domain web ke dalam bagian url pada browser
3. Mengisi form username dan password
4. Menekan tombol sign in

Setelah login berhasil pengguna akan diarahkan ke halaman dashboard seperti berikut dan proses login dianggap telah berhasil.



Gambar V.2 Tampilan dashboard

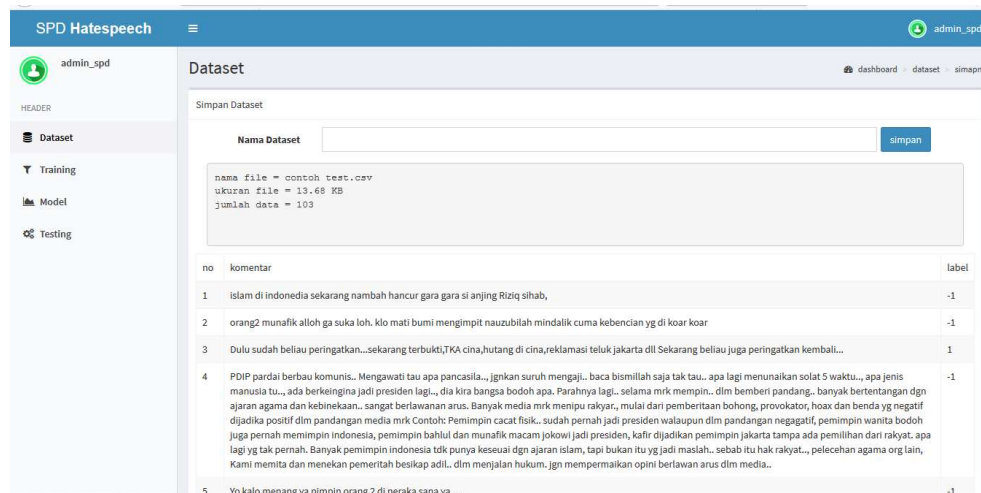
b. Menambahkan Dataset



Gambar V.3 Tampilan Halaman Dataset

Untuk menambahkan dataset admin dapat melakukan langkah sebagai berikut :

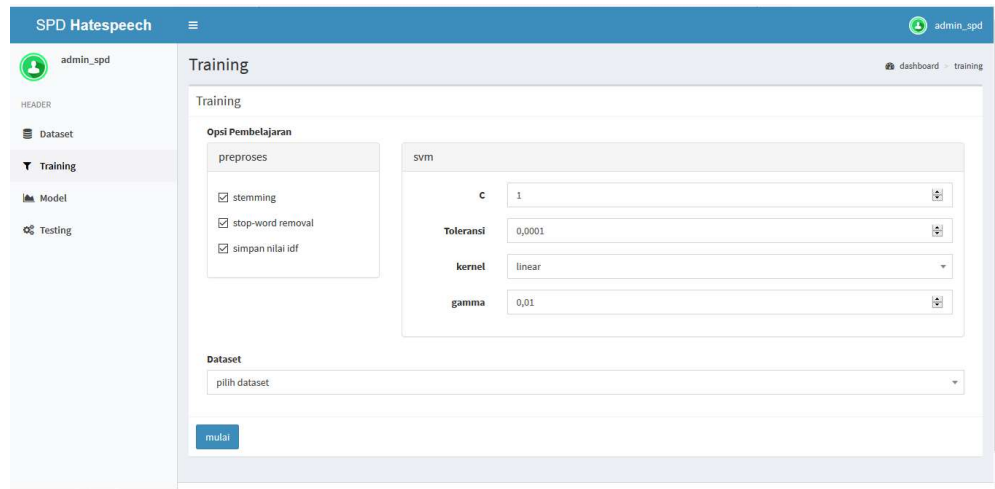
1. klik menu dataset
2. Pada bagian baru, klik tombol telusuri
3. Pilih file .csv yang akan diimport
4. Klik open, kemudian tunggu hingga tampilan berikut muncul



Gambar V.4 Tampilan simpan dataset

5. Isi form nama dataset
6. Klik tombol simpan

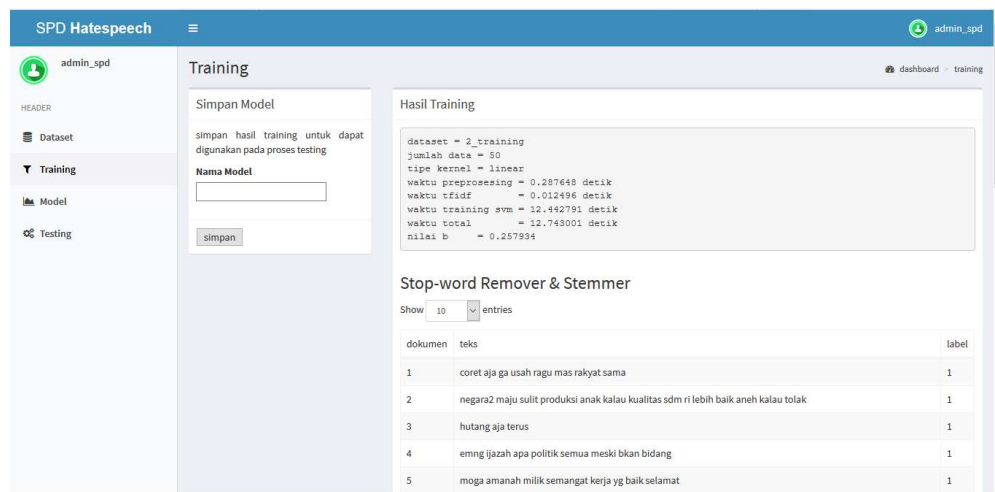
c. Melakukan Training



Gambar V.5 Tampilan halaman training

Untuk melakukan training admin dapat melakukan langkah-langkah sebagai berikut :

1. Klik menu training
2. Ubah parameter yang digunakan untuk training (optional) yang tersedia
3. Pilih dataset yang akan digunakan
4. Klik mulai, kemudian tunggu hingga halaman berikut muncul

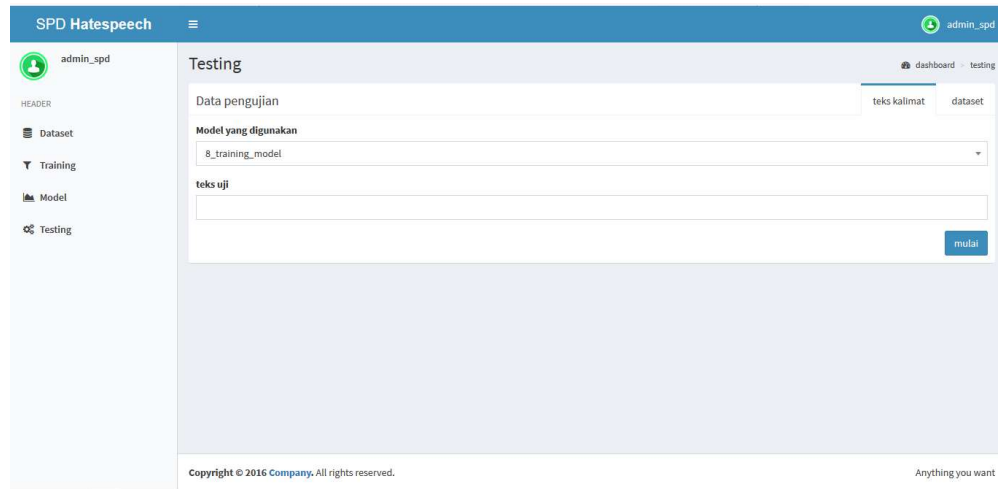


Gambar V.6 Tampilan simpan model

5. Isi form nama model
6. Klik simpan

d. Melakukan Pengujian

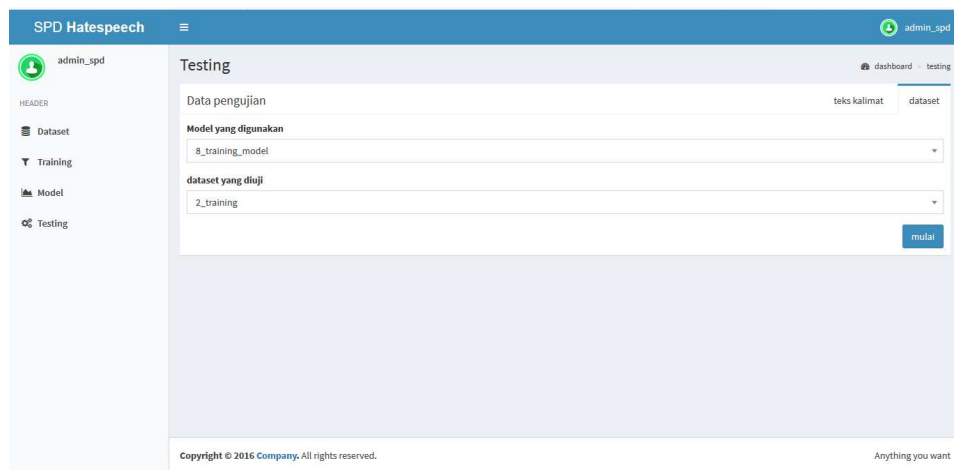
Dalam melakukan pengujian, sistem memberikan 2 jenis pengujian yaitu menguji teks dan menguji dataset.



Gambar V.7 tampilan halaman uji teks

Untuk melakukan pengujian teks admin dapat melakukan langkah-langkah sebagai berikut :

1. Klik menu testing
2. Klik tab teks kalimat
3. Pilih model yang akan digunakan
4. Isi teks yang akan diujikan
5. Klik tombol mulai.

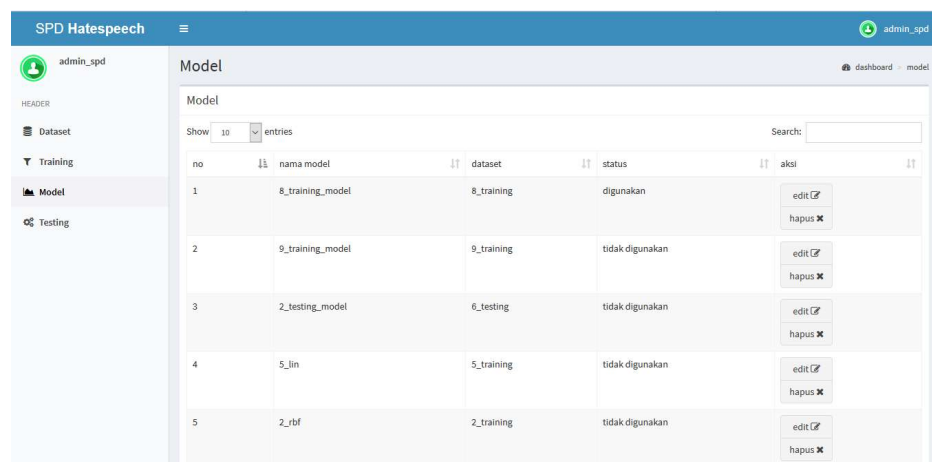












Gambar V.8 tampilan uji dataset

Untuk melakukan pengujian dataset admin dapat melakukan langkah-langkah sebagai berikut :

1. Klik menu testing
2. Klik tab dataset
3. Pilih model yang akan digunakan
4. Pilih dataset yang akan digunakan
5. Klik tombol mulai.

e. Mengaktifkan Model yang Digunakan

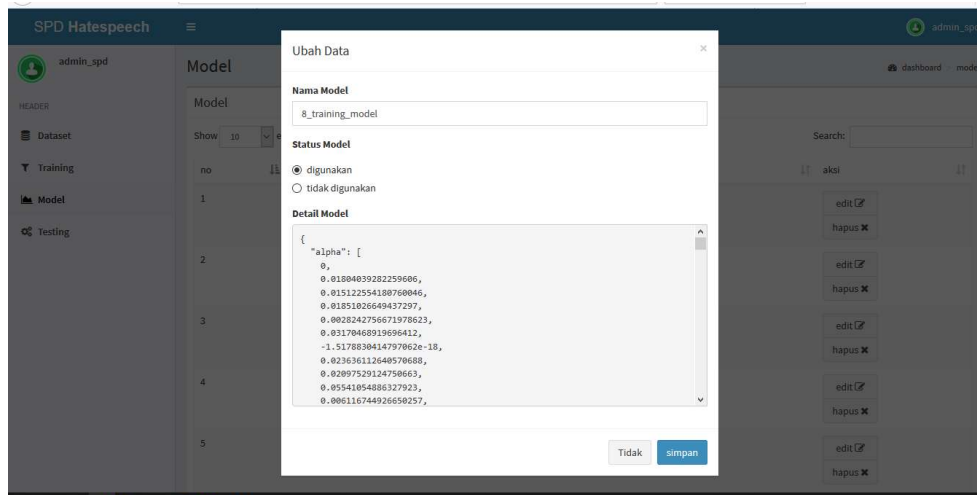


no	nama model	dataset	status	aksi
1	8_training_model	8_training	digunakan	edit  hapus 
2	9_training_model	9_training	tidak digunakan	edit  hapus 
3	2_testing_model	6_testing	tidak digunakan	edit  hapus 
4	5_lin	5_training	tidak digunakan	edit  hapus 
5	2_rbf	2_training	tidak digunakan	edit  hapus 

Gambar V.9 Tampilan halaman model

Untuk mengaktifkan model yang digunakan, admin dapat melakukan langkah-langkah sebagai berikut :

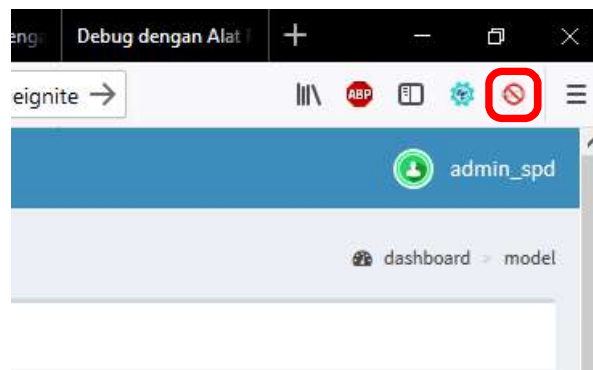
1. Klik edit pada model yang diinginkan, kemudian tampilan berikut akan muncul



Gambar V.10 Tampilan edit model

2. Klik opsi digunakan pada bagian status model
 3. Klik simpan
- f. Menggunakan Aplikasi Klien

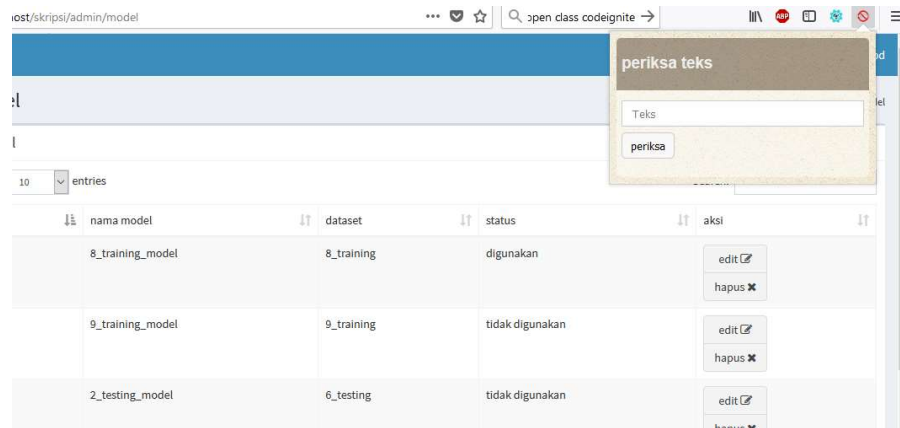
Aplikasi klien merupakan aplikasi yang berbentuk plugin browser yang harus dipasang terlebih dahulu pada browser pengguna. Setelah plugin berhasil terpasang maka akan muncul tombol disamping bilah menu seperti pada tampilan berikut.



Gambar V.11 Tombol plugin

Untuk menggunakan aplikasi tersebut pengguna dapat melakukan langkah-langkah sebagai berikut :

1. Klik tombol plugin yang muncul pada browser hingga muncul tampilan sebagai berikut.



Gambar V.12 tampilan dialog plugin

2. Isi form teks
3. Klik periksa, kemudian aplikasi akan menentukan apakah teks yang dimasukkan termasuk ujaran kebencian atau bukan.

5.3 Pengujian Sistem

5.3.1 Unit Test

Pada pengujian menggunakan Unit Test, pengujian dilakukan menggunakan tool pengujian PHPUnit. Sementara itu, fungsi yang akan diuji adalah fungsi train modul library SVM sebagai berikut :

Tabel V.1 tabel spesifikasi fungsi train

- Fungsi Train	
Input	
	<ul style="list-style-type: none"> - Data : array[N][D] ; N jumlah Data, D jumlah atribut/ kata - Labels : array[N] ; N jumlah data - Options : array{ <ul style="list-style-type: none"> o C ; parameter regularisasi o Tol ; toleransi o AlphaTol ; toleransi suport vektor o Maxiter ;maximal iterasi yang dilakukan o Numpasses ; banyak pengecekan untuk alpha yang tidak berubah untuk berhenti o Kernel ; tipe kernel o Memoize ; menyimpan matriks kernel o Gamma ; nilai gamma untuk kernel non-linear

Proses	
1	- data \leftarrow options[data]
2	- labels \leftarrow options[labels]
3	- if (options[C] != null)
4	o C \leftarrow options[C]
5	- else
6	o C \leftarrow 1
7	- if (options[tol] != null)
8	o tol \leftarrow options[tol]
9	- else
10	o tol \leftarrow 0,0001
11	- if (options[alphatol] != null)
12	o alphatol \leftarrow options[alphatol]
13	- else
14	o alphatol \leftarrow 10^{-7}
15	- if (options[maxiter] != null)
16	o maxiter \leftarrow options[maxiter]
17	- else
18	o maxiter \leftarrow 10000
19	- if (options[numpasses] != null)
20	o numpasses \leftarrow options[numpasses]
21	- else
22	o numpasses \leftarrow
23	- if (options[kernel] != null)
24	o kernel \leftarrow select_kernel(options[kernel]);
25	- else
26	o kernel \leftarrow select_kernel('linear');
27	- D \leftarrow length(data)
28	- N \leftarrow length(data[0])
29	- if (options[memoize] != null && options[memoize])
30	o kernelResult[N][N]
31	o for i[0...N-1]; i+1
32	▪ for j[0...N-1]; i+1
33	• kernelResult[i][j] = kernel(i,j)
34	- Menjalankan algoritma SMO seperti pada subbab 4.4
35	- if(options[kernel]=='linear')
36	o w[0...D-1] \leftarrow 0
37	o for j[0...D-1]; j+1
38	▪ for i[0...N-1]; i + 1
39	• w[j] \leftarrow w[j] + alpha[i]*labels[i]*data[i][j]
40	o usew_ \leftarrow true
41	- else
42	o newdata[], newlabels[], newalphas[]

43	○ $j \leftarrow 0$
44	○ for $i[0 \dots N-1]; i + 1$
45	▪ if ($\alpha[i] > \alpha_{\text{phatol}}$)
46	• $\text{newdata}[j] \leftarrow \text{data}[i]$
47	• $\text{newlabels}[j] \leftarrow \text{labels}[i]$
48	• $\text{newalpha}[j] \leftarrow \alpha[i]$
49	• $j++$
50	○ $\text{data} \leftarrow \text{newdata}, \text{labels} \leftarrow \text{newlabels}$
51	○ $\alpha \leftarrow \text{newalphas}$
52	○ $N \leftarrow \text{length}(\text{newdata})$
53	○ $\text{usew_} \leftarrow \text{false}$
54	- return model

Output	
-	Model : array{
	○ N
	○ D
	○ α : array [N]
	○ b : float
	○ w : array[D] ; nilai bobot ,jika menggunakan kernel linear
	○ usew_ ;
	○ data ; jika menggunakan kernel selain linear data dapat berubah
	○ labels ;
	○ kernelResult : array[N][N] ; jika menggunakan memoize
	○ kernel ; tipe kernel
	}

Dari spesifikasi dan alur fungsi di atas maka terdapat beberapa kondisi yang mempengaruhi output model secara signifikan seperti pada line 29 dan 35. Dari beberapa kondisi tersebut didapatkan testcase sebagai berikut :

Tabel V.2 tabel test case untuk fungsi train

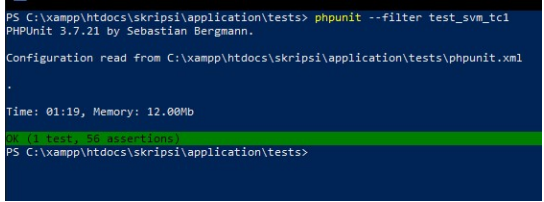
no	kondisi	Hasil yang diharapkan
1.	options[kernel] = linear parameter lain disesuaikan	<ul style="list-style-type: none"> - usew_ = true - $w \neq \text{null} \ \& \ w = \text{array}[D]$ - data tidak terdapat dalam model output - labels tidak berubah dari semula - hasil output lain sesuai dengan perhitungan SMO

no	kondisi	Hasil yang diharapkan
2	options[kernel] = rbf parameter lain disesuaikan	<ul style="list-style-type: none"> - usew_ = false - w tidak terdapat pada model output - nilai data, labels dapat berubah dari semula - hasil output lain sesuai dengan perhitungan SMO
3	options[memoize] = null atau options[memoize] = false parameter lain disesuaikan	<ul style="list-style-type: none"> - kernelResult = null
4	options[memoize] = true parameter lain disesuaikan	<ul style="list-style-type: none"> - kernelResult = array[N][N]

Pada pengujian untuk menentukan kesesuaian hasil perhitungan SMO (b dan α), diberlakukan pengujian dalam rentang nilai tertentu dengan rentang nilai yang kecil yaitu ± 0.001 . Hal ini dikarenakan algoritma yang digunakan menggunakan fungsi random. Selain itu data yang diujikan merupakan data yang berjumlah 50 teks dengan term sebanyak 582 setelah melalui preproses. Berikut adalah hasil pengujian unit testing dari test case di atas

a. TestCase 1

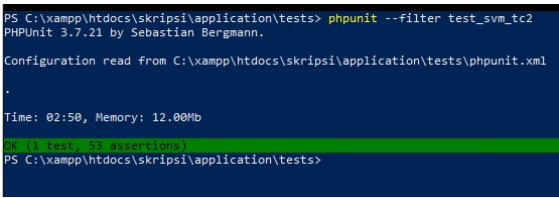
Tabel V.3 Tabel hasil unit test testcase 1

Testcase	Test svm_tcl
Kondisi	options[kernel]= linear parameter lain disesuaikan
Hasil yang diharapkan	<ul style="list-style-type: none"> - usew_ = true - w != null & w = array[D] - data tidak terdapat dalam model output - labels tidak berubah dari semula - hasil output lain sesuai dengan perhitungan SMO
Hasil unit test	 <pre> PS C:\xampp\htdocs\skripsi\application\tests> phpunit --filter test_svm_tcl PHPUnit 3.7.21 by Sebastian Bergmann. Configuration read from C:\xampp\htdocs\skripsi\application\tests\phpunit.xml . Time: 01:19, Memory: 12.00Mb OK (1 test, 56 assertions) PS C:\xampp\htdocs\skripsi\application\tests> </pre>

keterangan	1 test 56 assertion OK
-------------------	------------------------

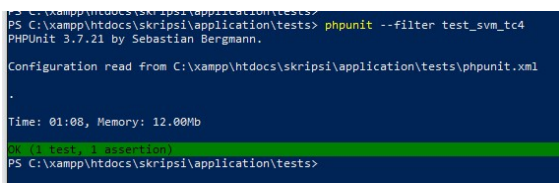
b. Testcase 2

Tabel V.4 tabel hasil unit testing untuk test case 2

Testcase	Test_svm_tc2
Kondisi	options[kernel] = rbf parameter lain disesuaikan
Hasil yang diharapkan	<ul style="list-style-type: none"> - usew_ = false - w tidak terdapat pada model output - hasil output lain sesuai dengan perhitungan SMO
Hasil unit test	 <pre> PS C:\xampp\htdocs\skripsi\application\tests> phpunit --filter test_svm_tc2 PHPUnit 3.7.21 by Sebastian Bergmann. Configuration read from C:\xampp\htdocs\skripsi\application\tests\phpunit.xml . Time: 02:50, Memory: 12.00Mb OK (1 test, 53 assertions) PS C:\xampp\htdocs\skripsi\application\tests> </pre>
keterangan	1 test 53 assertion OK

c. Testcase 3

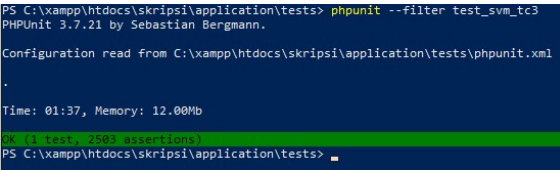
Tabel V.5 tabel hasil unit testing untuk testcase3

Testcase	Test_svm_tc3
Kondisi	options[memoize] = false parameter lain disesuaikan
Hasil yang diharapkan	- kernelResult = null
Hasil unit test	 <pre> PS C:\xampp\htdocs\skripsi\application\tests> phpunit --filter test_svm_tc3 PHPUnit 3.7.21 by Sebastian Bergmann. Configuration read from C:\xampp\htdocs\skripsi\application\tests\phpunit.xml . Time: 01:08, Memory: 12.00Mb OK (1 test, 1 assertion) PS C:\xampp\htdocs\skripsi\application\tests> </pre>
keterangan	1 test 1 assertion OK

d. Testcase 4

Tabel V.6 tabel hasil unit testing untuk testcase4

Testcase	Test_svm_tc4
Kondisi	options[memoize] = true parameter lain disesuaikan
Hasil yang diharapkan	- kernelResult != null

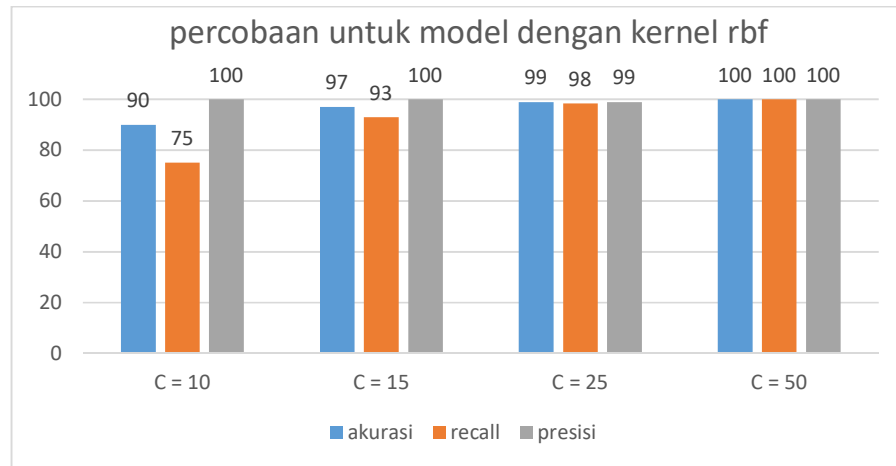
	<ul style="list-style-type: none"> - kernelResult = array[50][50] - nilai kernelresult harus sesuai
Hasil unit test	 <pre> PS C:\xampp\htdocs\skripsi\application\tests> phpunit --filter test_svm_tc3 PHPUnit 3.7.21 by Sebastian Bergmann. Configuration read from C:\xampp\htdocs\skripsi\application\tests\phpunit.xml Time: 01:37, Memory: 12.00Mb OK (1 test, 2503 assertions) PS C:\xampp\htdocs\skripsi\application\tests> </pre>
Keterangan	1 test 2503 assertion OK

Dari keseluruhan pengujian di atas, tidak ditemukan kesalahan terkait implementasi svm.

5.3.2 Pengujian confussion matrix

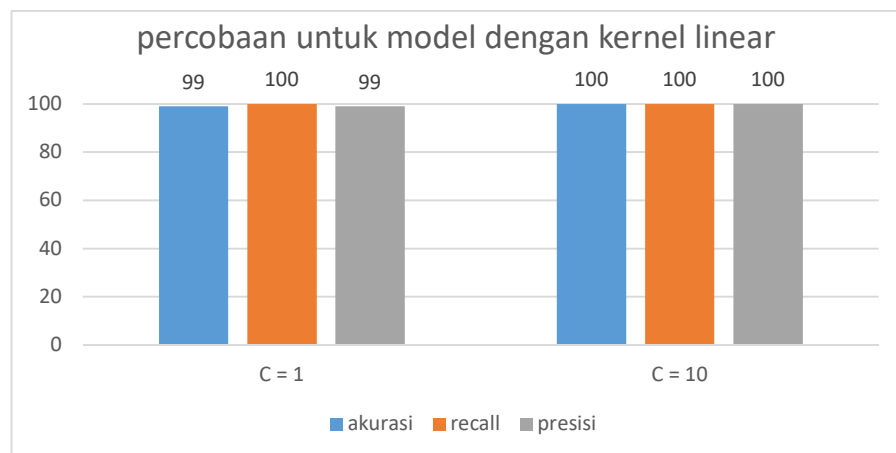
Pada pengujian berikutnya model yang dihasilkan akan diuji menggunakan confusion matrix. Pada pengujian ini model akan diukur tingkat keakuratan nya untuk mengklasifikasikan suatu teks. Sementara itu, baik model yang menggunakan kernel linear maupun kernel rbf akan diuji pada pengujian ini. Kemudian, dataset yang akan diujikan adalah dataset khusus untuk testing yang memiliki jumlah data sebanyak 103 data dengan data berlabel ujaran kebencian berjumlah 41 data dan data berlabel bukan ujaran kebencian sebanyak 62 data. Hal ini dilakukan dengan tujuan untuk mengetahui bagaimana kemampuan model dalam mengklasifikasikan data yang belum diketahui dan dipelajari sebelumnya.

Sebelum menentukan model yang akan dianalisis lebih jauh, beberapa model dievaluasi untuk menemukan model dengan nilai parameter C yang mampu menghasilkan akurasi, *recall*, serta presisi yang paling baik untuk mengklasifikasikan kembali data training. Berikut adalah grafik hasil pengujian dengan nilai paramater C yang berbeda untuk pengujian yang berkernel rbf :



Gambar V.13 diagram hasil pengujian pada model berkernel rbf dengan parameter C yang berbeda

Berikut adalah grafik hasil pengujian dengan nilai parameter C yang berbeda untuk pengujian yang berkernel linear :



Gambar V.14 diagram hasil pengujian pada model berkernel linear dengan parameter C yang berbeda

Setelah melakukan beberapa percobaan dengan nilai C yang berbeda didapatkan model dengan nilai C = 10 (9_lin_C10) untuk kernel linear dan model dengan nilai C = 50 (9_rbf_C50) untuk kernel rbf dengan hasil terbaik. Oleh karena itu, kedua model tersebut selanjutnya akan dibahas secara lebih lanjut. Berikut adalah tabel confusion matrix dan perhitungan akurasi, presisi, dan *recall* untuk pengujian menggunakan model 9_lin_C10, dimana TP merupakan *True-Positive*, FN adalah *False-Negative*, FP adalah *False-Positive* dan TN adalah *True-Negative*:

Tabel V.7 tabel hasil confusion matrix menggunakan dataset 9_lin_C10

		Label prediksi	
		Ujaran kebencian (true)	Bukan ujaran kebencian (false)
Label aktual	Ujaran kebencian (true)	20 (TP)	21 (FN)
	Bukan ujaran kebencian (false)	9 (FP)	53 (TN)

$$\begin{aligned}
 akurasi &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{20 + 53}{20 + 53 + 9 + 21} \\
 &= 70,874
 \end{aligned}$$

$$\begin{aligned}
 recall &= \frac{TP}{TP + FN} \\
 &= \frac{20}{20 + 21} \\
 &= 48,78
 \end{aligned}$$

$$\begin{aligned}
 presisi &= \frac{TP}{TP + FP} \\
 &= \frac{20}{20 + 9} \\
 &= 68,966
 \end{aligned}$$

Kemudian berikut adalah tabel confusion matrix dan hasil perhitungan akurasi, presisi, dan recall untuk pengujian yang menggunakan model 9_rbf_C50:

Tabel V.8 tabel hasil confusion matrix menggunakan dataset 9_rbf_C50

		Label prediksi	
		Ujaran kebencian (true)	Bukan ujaran kebencian (false)
Label aktual	Ujaran kebencian (true)	14 (TP)	27 (FN)
	Bukan ujaran kebencian	7 (FP)	55 (TN)

	(false)		
--	---------	--	--

$$\begin{aligned}
 akurasi &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{14 + 55}{14 + 55 + 7 + 27} \\
 &= 66,99
 \end{aligned}$$

$$\begin{aligned}
 recall &= \frac{TP}{TP + FN} \\
 &= \frac{14}{14 + 27} \\
 &= 34,146
 \end{aligned}$$

$$\begin{aligned}
 presisi &= \frac{TP}{TP + FP} \\
 &= \frac{14}{14 + 7} \\
 &= 66,667
 \end{aligned}$$

Lalu, berikut adalah tabel perbandingan akurasi, recall dan presisi untuk pengujian yang menggunakan model 9_lin_C10 dan 9_rbf_C50

Tabel V.9 tabel hasil pengujian menggunakan dataset testing

Model	Akurasi	Recall	Presisi
9_lin_C10	70,874	48,78	68,966
9_rbf_C50	66,99	34,146	66,667

Dari hasil di atas, didapatkan bahwa sistem dapat mengklasifikasikan data dengan tingkat akurasi sebesar 66 - 70 % data yang belum pernah dipelajari. Kemudian dari hasil diatas diketahui juga bahwa model yang menggunakan kernel linear lebih baik untuk mengklasifikasikan data yang belum pernah dipelajari.

Sementara itu, untuk hasil recall dan presisi model berkernel linear memiliki nilai yang lebih baik. Namun baik model berkernel linear dan rbf menghasilkan nilai recall yang kurang baik. Hal ini dapat disebabkan oleh tidak seimbangnya jumlah data yang berlabel ujaran kebencian dengan data yang berlabel bukan ujaran

kebencian pada data latih. Jumlah data berlabel bukan ujaran kebencian yang lebih banyak menyebabkan kecenderungan sistem untuk mengklasifikasikan suatu teks yang belum diketahui labelnya sebagai bukan ujaran kebencian. hal ini membuat jumlah data *false-negatif* bertambah dan Mengurangi nilai recall.

5.3.3 Performance Test

Selain pengujian menggunakan unit test, dipaparkan juga hasil pengujian menggunakan performance test. Pada pengujian ini, hal yang akan diuji adalah berapa waktu yang dibutuhkan sistem untuk merespon permintaan dari pengguna dalam menentukan kategori teks.

Skenario yang akan digunakan pada pengujian ini adalah :

1. Pengujian akan membandingkan performa dari model yang menggunakan kernel linear dan kernel non-linear (rbf).
2. Waktu respon dihitung sejak permintaan dikirim hingga respon diterima oleh browser pengguna akhir.
3. Akan dilakukan 20 kali percobaan untuk tiap model nya, kemudian diambil nilai rata-rata, minimum dan maksimumnya.

Model yang akan diuji performa nya merupakan model yang dibahas pada pengujian confusion matrix yaitu model 9_lin_C10 dan 9_rbf_C50. Berikut adalah detail model yang akan diujikan pada pengujian ini:

Tabel V.10 model yang akan diujikan

Nama Model	Atribut	Nilai
9_lin_C10	Kernel	Linear
	C	10
	Tol	0.0001
	N (jumlah data)	406
	D (jumlah term)	2115
	Labels	Array[406]
	Alpha	Array[406]
	B (bias)	0,27079664890131705

Nama Model	Atribut	Nilai
9_rbf_C50	Usew_	True
	W	Array [2115]
	Kernel	Rbf
	C	50
	Tol	0,0001
	Gamma	0,01
	N (jumlah data)	384
	D (jumlah term)	2115
	Labels	Array[384]
	Alpha	Array[384]
	B (bias)	-1,365369589597491
	Usew_	False

Kemudian pengujian ini, akan dijalankan pada laptop dengan spesifikasi sebagai berikut :

Tabel V.11 spesifikasi notebook yang digunakan untuk pengujian

Jenis komponen	Spesifikasi komponen
Prosesor	Intel core i5 4200u 2/4 (core/thread) 2.4 GHz
RAM	8 GB ddr4
Penyimpanan	HDD 1 TB
Sistem operasi	Windows 10 pro – 64 bit
Server PHP	XAMPP dengan PHP 7.2.1
Browser	Mozilla Firefox 59.0.2

Berikut adalah hasil pengujian *performance test*:

Tabel V.12 tabel hasil pengujian performance test

	Waktu (milidetik)	
	9_lin_C10	9_rbf_C50
1	234	553
2	218	516

	Waktu (milidetik)	
	9_lin_C10	9_rbf_C50
3	219	563
4	218	516
5	219	547
6	203	547
7	219	515
8	218	567
9	234	513
10	297	516
11	250	518
12	274	549
13	235	531
14	218	547
15	211	513
16	219	550
17	203	508
18	189	532
19	218	515
20	224	500
MIN	189	500
AVG	226	530.8
MAX	297	567

Dari hasil di atas, model 9_lin_C10 menghasilkan waktu respon yang lebih baik. Hal ini dikarenakan model yang menggunakan kernel linear melakukan perhitungan dot product x dengan w dan tidak melakukan proses perhitungan kernel antara sampel test dengan support vector saat proses prediksi seperti pada model yang menggunakan kernel rbf. Selain itu baik sistem yang menggunakan kernel linear maupun kernel rbf mampu menghasilkan waktu respon dibawah 1 detik.

Meskipun begitu, waktu respon ini tidak termasuk dengan waktu transfer yang dibutuhkan jika telah digunakan secara publik karena pengujian yang dilakukan secara lokal. Sehingga model yang menghasilkan waktu respon yang lebih kecil lebih direkomendasikan untuk digunakan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan, dapat diambil kesimpulan dari penelitian yang dilakukan sebagai berikut :

1. SVM TF-IDF dapat diimplementasikan pada sistem peringatan dini ujaran kebencian. Sistem peringatan dini ini direalisasikan ke dalam bentuk sebuah pengaya web browser.
2. Sistem yang telah dibuat menggunakan SVM sebagai teknik klasifikasi telah diimplementasikan secara benar dan sesuai dengan hasil yang diharapkan.
3. Sistem dengan model yang menggunakan kernel linear lebih layak digunakan dalam sebuah sistem peringatan dini dengan waktu respon kurang dari satu detik (minimum = 189 milidetik; rata-rata = 226 milidetik; maksimum = 297 milidetik) dan tingkat akurasi sebesar 70,874% pada pengujian menggunakan dataset uji . Sementara untuk model yang menggunakan kernel rbf juga membutuhkan waktu kurang dari 1 detik (minimum = 500 milidetik; rata-rata= 530,8 milidetik ; maksimum = 567 milidetik) dan memiliki akurasi sebesar 66,99% pada pengujian menggunakan dataset uji.

6.2 Saran

Berdasarkan hasil yang sudah didapatkan pada penelitian ini, terdapat beberapa saran yang dapat dipertimbangkan untuk kedepannya seperti :

1. Melakukan validasi data yang digunakan untuk pengujian dan pelatihan dengan bantuan anotator yang ahli pada bidang bahasa dan komunikasi.
2. Melakukan optimisasi pada proses pembelajaran seperti menambahkan seleksi fitur seperti particle swarm optimization untuk mempercepat waktu prediksi.

DAFTAR PUSTAKA

- [1] J. Balea, "The latest stats in web and mobile in Indonesia (INFOGRAPHIC)," 2016. [Online]. Available: <https://www.techinasia.com/indonesia-web-mobile-statistics-we-are-social>. [Accessed: 16-May-2018].
- [2] A. Setyo Wardani, "Facebook Kini Punya 115 Juta Pengguna Aktif di Indonesia," 2017. [Online]. Available: <http://tekno.liputan6.com/read/3057940/facebook-kini-punya-115-juta-pengguna-aktif-di-indonesia>. [Accessed: 18-Feb-2018].
- [3] T. H. Putri, "Banyak Hoax dan Ujaran Kebencian, Facebook Gagal Pahami Budaya Indonesia," 2018. [Online]. Available: <https://www.idntimes.com/news/indonesia/teatrika/lamban-tangani-konten-hoax-dan-ujaran-kebencian-facebook-gagal-pahami-budaya-indonesia-1/full>. [Accessed: 30-Aug-2018].
- [4] Agustin Setyo Wardani, "Gagal Atasi Ujaran Kebencian, Facebook Cs Akan Didenda Rp 76 M," 2017. [Online]. Available: <https://www.liputan6.com/tekno/read/3008171/gagal-atasi-ujaran-kebencian-facebook-cs-akan-didenda-rp-76-m>. [Accessed: 30-Jul-2018].
- [5] Retia Kartika Dewi, "Inilah Kasus Ujaran Kebencian yang Melibatkan PNS dan Pegawai BUMN," 2018. [Online]. Available: <https://regional.kompas.com/read/2018/05/22/17205521/inilah-kasus-ujaran-kebencian-yang-melibatkan-pns-dan-pegawai-bumn?page=2>. [Accessed: 30-Aug-2018].
- [6] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," *Proceeding LSM '12 Proc. Second Work. Lang. Soc. Media*, no. Lsm, pp. 19–26, 2012.
- [7] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy and Internet*, vol. 7, no. 2, pp. 223–242, 2015.
- [8] N. Saputra, B. T. Adji, and A. E. Permanasari, "Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming Menggunakan Metode Naive Bayes dan SVM," *J. Din. Inform.*, vol. 5, no. November, 2015.
- [9] S. K. Lidya, O. S. Sitompul, and S. Efendi, "Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (SVM) dan K-Nearest Neighbor(K-NN)," *Semin. Nas. Teknol. dan Komun.* 2015, vol. 2015, no. Sentika, pp. 1–8, 2015.
- [10] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweet," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, no. 2, pp. 759–760.

- [11] M. C. Anam, “Surat Edaran Kapolri Tentang Penanganan Ujaran Kebencian (Hate Speech) dalam Kerangka Hak Asasi Manusia,” pp. 341–364.
- [12] Kapolri, *Surat Edaran Nomor:SE/06/X/2015 tentang Penanganan Ujaran Kebencian (Hate Speech)*. Indonesia, 2015.
- [13] Linawati, “Tindak Tutur Ujaran Kebencian Dalam Komentar Pembaca pada Surat Kabar Online Tribunnews.Com,” *Bhs. Dan Sastra Indones. - SI Univ. Negeri Yogyakarta*, vol. 6, pp. 606–614, 2017.
- [14] N. Waidyanatha, “Towards a typology of integrated functional early warning systems,” *Int. J. Crit. Infrastructures*, vol. 6, no. 1, pp. 31–51, 2010.
- [15] K. Whitenton, “Overuse of Overlays: How to Avoid Misusing Lightboxes,” 2015. [Online]. Available: <https://www.nngroup.com/articles/overuse-of-overlays/>. [Accessed: 19-Oct-2018].
- [16] J. C. Platt, “Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines,” *Adv. kernel methods*, pp. 185–208, 1998.
- [17] J. Nielsen, *Usability Engineering*. 1993.
- [18] Stanford University, “CS229 Simplified SMO Algorithm,” *Cs*, vol. 229, pp. 1–5, 2009.