SKRIPSI

KLASIFIKASI BERITA *ONLINE* BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE*

Disusun Oleh :

MUHAMMAD RISKI SAPUTRA

DBC 116 010

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PALANGKA RAYA

2020

KLASIFIKASI BERITA *ONLINE* BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE*

SKRIPSI

Sebagai salah satu syarat untuk menyelesaikan Program Strata-1 pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya

Oleh

MUHAMMAD RISKI SAPUTRA

DBC 116 010

Telah dipertahankan didepan tim penguji, pada :

Hari/Tanggal : Jumat, 16 Oktober 2020

Waktu : 15.00-16.30 WIB

1. Drs. JADIAMAN PARHUSIP, M.Kom
   NIP. 19630423 198502 1 001 : .....(Ketua)
2. SHERLY CHRISTINA, S.Kom., M.Kom
   NIP. 19810929 200604 2 001 : .....(Anggota)
3. ARIESTA LESTARI, S.Kom., M.Cs., Ph.D
   NIP. 19800322 200501 2 004 : .....(Anggota)
4. ADE CHANDRA SAPUTRA, S.Kom., M.Cs
   NIP. 19870203 201404 1 001 : .....(Anggota)
5. NAHUMI NUGRAHANINGSIH, Ph.D
   NIP. 19791009 200801 2 016 : .....(Anggota)

Mengetahui :

Fakultas Teknik
Universitas Palangka Raya
Dekan,

Ir. WALUYO NUSWANTORO, M.T.
NIP. 19651119 199302 1 001

Jurusan / Program Studi Teknik Informatika
Fakultas Teknik Universitas Palangka Raya
Ketua Jurusan,

ABERTUN SAGIT SAHAY, S.T., M.Eng
NIP. 19751212 200312 1 002

KLASIFIKASI BERITA *ONLINE* BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE*

SKRIPSI

Sebagai salah satu syarat menyelesaikan Program Strata - 1
pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya

Oleh :

MUHAMMAD RISKI SAPUTRA
NIM. DBC 116 010

Disetujui untuk diajukan dalam Ujian Skripsi

Pembimbing I,	Pembimbing II,
SHERLY CHRISTINA, S.Kom., M.Kom	**ARIESTA LESTARI,S.Kom.,M.Cs., Ph.D**
NIP. 19810929 200604 2 001	**NIP. 19800322 200501 2 004**

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020

PERNYATAAN

Dengan ini saya menyatakan dengan sebenar - benarnya bahwa dalam Skripsi ini tidak terdapat karya ilmiah yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu Perguruan Tinggi, serta tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam Skripsi ini dan disebutkan dalam Tinjauan Pustaka.

Palangka Raya, Oktober 2020

MUHAMMAD RISKI SAPUTRA
DBC 116 010

RIWAYAT PENYUSUN

Data Diri

Nama	: MUHAMMAD RISKI SAPUTRA
NIM	: DBC 116 010
Fakultas	: Teknik
Jurusan/Program Studi	: Teknik Informatika
Jenjang	: Strata 1 (S-1)
Jenis Kelamin	: Laki-Laki
Tempat, Tanggal Lahir	: Palangka Raya, 7 September 1998
Agama	: Islam
Status dalam Keluarga	: Anak Kandung
Anak ke -	: 2
Alamat	: Jl. G. Obos Induk No. 154
No. Telpon/HP	: +6285828455097

Data Orang Tua

Nama Ayah	: Indra Kusnadi
Pekerjaan Ayah	: Swasta
Nama Ibu	: Animah
Pekerjaan Ibu	: Ibu Rumah Tangga
Alamat Orang Tua	: Jl. G. Obos Induk No. 154
No. Telpon/HP	: +6281251644428

Riwayat Pendidikan *)

SD	: MIN Langkai Palangka Raya	(Tahun Lulus 2010)
SMP	: MTs Darul Amin Palangka Raya	(Tahun Lulus 2013)
SMA	: MAN Model Palangka Raya	(Tahun Lulus 2016)

Palangka Raya, Oktober 2020

MUHAMMAD RISKI SAPUTRA
DBC 116 010

Keterangan:
*) Nama, Tempat, Tahun Lulus

HALAMAN PERSEMBAHAN

~Bismillahirrahmanirrahim~

Alhamdullillah, puji serta rasa syukur kupanjatkan kepada Allah SWT, sujud syukur kusembahkan kepada-Mu ya Allah, Tuhan Yang Maha Agung dan Maha Tinggi. Atas berkah dan karunia-Mu, sehingga hamba dapat menyelesaikan Tugas Akhir Skripsi hamba dengan segala kekurangannya.

Kupersembahkan karya kecil ini, untuk orang-orang yang kukasihi dan kusayangi

Teruntuk **Mama dan Abah Tercinta,**

Terima kasih atas kasih sayang yang berlimpah dari mulai ulun lahir, hingga ulun sudah sebesar ini. Terima kasih atas kesabaran yang tiada batas diberikan dalam menghadapi ulun. Terima kasih untuk dukungan serta cinta yang diberikan yang tidak mungkin terbalaskan dengan sedikit kata cinta dari persembahan ini. Semoga ini menjadi satu langkah awal untuk membahagiakan Mama dan Abah.

Untuk **Kaka dan Adikku,**

Terima kasih atas dukungan dan doa yang luar biasa diberikan tanpa henti. Ka Heni dan Adit yang selama ini sudah menjadi saudara sekaligus sahabat.

Kepada **Dosen Pembimbing,**

Terimakasih untuk Ibu Sherly dan Ibu Ariesta, yang selalu sabar dalam membimbing saya selama mengerjakan Tugas Akhir, sehingga saya dapat menyelesaikan Tugas Akhir ini.

Juga kepada **Dosen Penguji,**

Terima kasih kepada Pak Jadiaman, Pak Ade dan Ibu Nahumi atas saran dan masukannya dalam mengevaluasi Tugas Akhir ini, sehingga penelitian saya dapat menjadi lebih baik lagi.

Untuk **Sahabat dan Teman-temanku,**

Terima kasih kepada Slis, Petra, Sinjah, Zakha, yang selalu memberikan semangat, memberikan bantuan, memotivasi ketika saya merasa bosan, dan menjadi tempat berkeluh kesah tentang perjuangan dalam mengerjakan skripsi ini. Juga kepada teman-teman GSC --Ray, Roy, Zikri, Romzy, Apri, Fris, Made, Sepri, Teguh, Caca, Andrew, Joey, Rio, Ghiraldi-- Terima kasih atas *support* yang diberikan dan lawakan yang walaupun receh tapi tetap ngakak.

Last but not least,

Terimakasih untuk teman-teman jurusan Teknik Informatika yang sudah mengisi hari-hari perkuliahanku. Semoga pertemanan kita tetap solid sampai akhir dan semuanya sukses menggapai impian dan cita-citanya masing-masing *-Aamiin-*.

SEE U ON TOP GUYS!

--MUHAMMAD RISKI SAPUTRA--

KATA PENGANTAR

Puji dan syukur peneliti panjatkan kehadirat Allah SWT, yang telah melimpahkan segala nikmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Klasifikasi Berita *Online* Berbahasa Indonesia Menggunakan Algoritma *Support Vector Machine*"** dengan sebaik-baiknya.

Penulis ingin menyampaikan terima kasih sebesar-besarnya atas bantuan dan dorongan semangat dari keluarga, dosen dan teman-teman. Terutama kepada Ibu Sherly Christina, S.Kom., M.Kom dan Ibu Ariesta Lestari, S.Kom., M.Cs., Ph.D selaku dosen pembimbing yang telah banyak membantu dalam proses penyelesaian skripsi penulis.

Penulis menyadari bahwa laporan skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan saran maupun kritik yang membangun demi kesempurnaan laporan ini. Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak, khususnya bagi mahasiswa jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.

Palangka Raya, Oktober 2020

Muhammad Riski Saputra

KLASIFIKASI BERITA *ONLINE* BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE*

MUHAMMAD RISKI SAPUTRA (DBC 116 010)
Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya
Kampus Tunjung Nyaho Jl. Yos Sudarso Palangka Raya 73112
Email : muhammadriskisaputra@gmail.com

ABSTRAK

Pengklasifikasian berita dalam suatu kategori dilakukan oleh editor secara manual. Sehingga apabila jumlah artikel berita yang tersedia sangat besar, proses pengelompokkan secara manual mungkin akan mengakibatkan *human error* karena memiliki kemungkinan untuk diklasifikasikan ke dalam kategori konten yang tidak tepat. Padahal penempatan berita berdasarkan kategori yang sesuai menjadi salah satu unsur penting dalam memenuhi minat pembaca berita. Oleh karena itu, pada penelitian ini dibuat sebuah sistem yang bertujuan untuk melakukan klasifikasi berita *online* berbahasa Indonesia secara otomatis.

Metodelogi penelitian yang digunakan meliputi metodelogi pengumpulan data, metodelogi pengembangan perangkat lunak, metode klasifikasi serta skenario pengujian. Pada penelitian ini, sistem klasifikasi berita *online* dibuat menggunakan *text mining* dengan algoritma klasifikasi *support vector machine* (SVM). Tahapan-tahapan yang diimplementasikan dalam sistem klasifikasi ini yaitu *remove punctuation*, *case folding*, tokenisasi, *stopwords removal*, tf-idf dan klasifikasi dengan SVM. Sistem klasifikasi pada penelitian ini dibangun dengan dataset berjumlah 1000, yang kemudian dibagi menjadi data latih dan data uji. Sistem klasifikasi dilatih terlebih dahulu untuk membuat model klasifikasi yang akan digunakan dalam mengkategorikan data baru dengan menggunakan data latih. Kemudian pengujian sistem klasifikasi dilakukan menggunakan data uji dengan metode *accuracy* untuk mengamati keakuratan sistem klasifikasi dalam mengelompokkan data ke dalam kategori kelas yang tepat.

Pengujian performa sistem klasifikasi dengan metode *accuracy* dilakukan dengan beberapa skenario pengujian untuk mengetahui pengaruh jumlah data latih terhadap efektifitas klasifikasi *support vector machine*. Dari pengujian yang dilakukan nilai *accuracy* tertinggi diperoleh pada skenario pengujian ke-4 sebesar 95% dengan komposisi data latih 80% dan data uji 20%. Pengujian fungsionalitas sistem dengan menggunakan *blackbox testing* menunjukkan bahwa sistem sudah dapat berjalan sesuai dengan hasil yang diharapkan. Berdasarkan hasil implementasi dan pengujian performa sistem klasifikasi pada dokumen berita *online* berbahasa Indonesia, sistem telah dapat memenuhi tujuan dalam melakukan klasifikasi pada dokumen berita *online* berbahasa Indonesia.

Kata Kunci : *Text Mining*, *Support Vector Machine*, Klasifikasi Berita.

INDONESIAN ONLINE NEWS CLASSIFICATION USING SUPPORT VECTOR MACHINE ALGORITHM

MUHAMMAD RISKI SAPUTRA (DBC 116 010)
Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya
Kampus Tunjung Nyaho Jl. Yos Sudarso Palangka Raya 73112
Email : muhammadriskisaputra@gmail.com

ABSTRACT

The editor has done news classification in a category manually. Hence, if the number of news articles available is really large, the manual grouping process may result in a human error because it can be classified into inappropriate content categories. Even though the placement of news based on the appropriate category is one of the important factors in fulfilling the interest of news readers. Therefore, in this study, the system has designed to classify Indonesian online news automatically.

The research methodology used includes data collection methodology, software development methodology, classification methods and test scenarios. In this study, text mining with a support vector machine (SVM) algorithm classification created for the online news classification system. The stages implemented in this classification system are removed punctuation, case folding, tokenization, stop words removal, TF-IDF and classification with SVM. In this study, 1000 datasets used to build the classification system, divided into training data and testing data. Training data used to train the classification system in advance for creating a classification model used in categorizing new data. Then, the accuracy method with testing data used to carry out the classification system testing for observing the accuracy of the classification system in classifying the data into appropriate class categories.

The accuracy method with several test scenarios used to test the performance of the classification system to determine the effect of the amount of training data on the effectiveness of the support vector machine classification. From the test, the fourth test scenario has obtained as the highest accuracy value of 95% with a composition of 80% training data and 20% test data. Testing system functionality used black-box testing show that the system can work according to the expected results. Based on the results of the implementation and performance testing of the classification system for Indonesian online news documents, the system has fulfilled the objectives of classifying online news documents in Indonesian.

Keywords: *Text Mining, Support Vector Machine, News Classification.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berita merupakan bagian dari komunikasi yang dapat memberikan informasi mengenai suatu kejadian atau informasi mengenai suatu fakta maupun isu yang sedang terjadi. Berita dapat disajikan dalam bentuk cetak, siaran, internet maupun melalui mulut ke mulut. Saat ini, salah satu media yang banyak digunakan dalam membaca berita adalah melalui *website*. *Website* digunakan sebagai media membaca berita karena dapat diakses kapanpun dan dimanapun. Maraknya penggunaan media *website* membuat berbagai media pemberitaan beralih ke media *online* dan membuat situsnya sendiri.

Pada umumnya, setiap situs berita *online* memiliki beberapa kategori atau topik berita, sehingga pembaca berita dapat memilih topik berita yang ingin dibaca berdasarkan minatnya. Biasanya, pengelompokkan berita dalam suatu kategori dilakukan oleh editor secara manual. Prosesnya dilakukan dengan mengetahui isi dari berita secara keseluruhan untuk selanjutnya dikelompokkan berdasarkan kategori yang tepat. Sebagai contoh untuk penggalan teks berita yang dilansir pada tanggal 12 Maret 2020 dari situs *detik.com* dengan judul "*Ngerinya Dampak Corona ke Ekonomi RI*" ditulis oleh Sylke Febrina Laucereno, seperti berikut:

> *"Perekonomian nasional diprediksi lebih rendah dibandingkan periode-periode sebelumnya. Hal ini disebabkan wabah corona yang sudah*

menyebar ke berbagai negara termasuk Indonesia. Bank sentral akan menghitung kembali proyeksi pertumbuhan ekonomi Indonesia tahun ini. Padahal sebelumnya dalam rapat dewan gubernur (RDG) BI periode Februari 2020, proyeksi ekonomi RI sudah turun menjadi 5,1% - 5,5% dari sebelumnya 5%-5,4%. Gubernur BI Perry Warjiyo menjelaskan dibutuhkan sumber perekonomian yang kuat agar tak berdampak signifikan. Selain itu industri pariwisata dan perhotelan juga telah mengalami kerugian mencapai US$ 1,5 miliar atau setara dengan Rp 21 triliun. Potensi kerugian ini dihitung dari perkiraan wisatawan China yang biasanya menghabiskan US$ 1.100 dalam satu kali perjalanan ke Indonesia. Karena itu restoran dan hotel sudah mulai merasakan dampak penurunan okupansi. Hal ini membuat perusahaan melakukan efisiensi. Direktur Riset CORE Indonesia Piter Abdullah mengungkapkan insentif yang diberikan oleh pemerintah untuk menangkal dampak corona belum ampuh untuk mendorong daya beli agar tetap stabil."

Penggalan teks berita *online* diatas, dikategorikan sebagai Ekonomi karena topik yang dibahas adalah dampak virus corona terhadap ekonomi Republik Indonesia. Namun, apabila jumlah artikel berita yang tersedia sangat besar, proses pengelompokkan secara manual mungkin akan mengakibatkan *human error* karena memiliki kemungkinan untuk diklasifikasikan sebagai konten kesehatan terkait pembahasan berita mengenai virus corona. Oleh karena itu, diperlukan suatu sistem yang dapat membantu dalam melakukan klasifikasi berita berdasarkan topik yang dibahas dengan menggunakan *text mining*.

Text mining merupakan salah satu cabang ilmu *data mining* yang menganalisis data berupa dokumen teks. Sebelum suatu teks dianalisis menggunakan metode dalam *text mining* perlu dilakukan *text pre-processing* terlebih dahulu, baru setelah itu dapat dilakukan metode klasifikasi untuk

mengelompokkannya kedalam masing-masing kategori. Berbagai macam metode klasifikasi banyak digunakan dalam melakukan klasifikasi berupa teks diantaranya adalah *Naïve Bayes Classifier* (NBC) (Mahmudy, 2015), *K-Nearest Neighbour* (KNN) (Irfa, 2018), dan *Support Vector Machine* (SVM) (Pratama, 2013). Hasil penelitian terkait klasifikasi teks yang dilakukan oleh Fatmawati dan Muhammad Affandes (2017) yang berjudul "*Klasifikasi Keluhan Menggunakan Metode Support Vector Machine (SVM) (Studi Kasus : Akun Facebook Group iRaise Helpdesk)*" menunjukkan metode SVM digunakan untuk melakukan pengelompokkan data keluhan sesuai dengan kategorinya memperoleh hasil akurasi sebesar 95,67%.

Berdasarkan latar belakang yang telah diuraikan, maka peneltian ini bertujuan untuk melakukan pengklasifikasian otomatis menggunakan algoritma *Support Vector Machine* (SVM) pada teks berita *online* berbahasa Indonesia berdasarkan topik berita, kemudian hasil dari penelitian akan diuji dengan menghitung akurasi untuk mengetahui performa pengklasifisikasian yang diimplementasikan pada sistem. Maka judul pada penelitian ini adalah "Klasifikasi Berita *Online* Berbahasa Indonesia Menggunakan Algoritma *Support Vector Machine*".

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah "Bagaimana melakukan klasifikasi berita *online* berbahasa Indonesia menggunakan Algoritma *Support Vector Machine*" ?

1.3 Batasan Masalah

Agar penelitian yang dilakukan tidak keluar dari pokok permasalahan yang dirumuskan, dalam penelitian ini terdapat beberapa batasan masalah, yaitu:

1. Data berita *online* yang digunakan adalah data berita *online* berbahasa Indonesia, berjumlah 1000 data. Data berita *online* dikumpulkan dari situs berita *online* yang tergabung dalam *Indonesia Digital Association* yaitu *detik.com*, *kompas.com* dan *okezone.com*.
2. Jumlah topik yang digunakan dalam klasifikasi adalah 5 topik yaitu Lifestyle, Olahraga, Politik, Ekonomi dan Teknologi berdasarkan konsumsi konten berita *online* terbanyak di Indonesia tahun 2019 (Utomo, 2019*)*.
3. Algoritma yang digunakan dalam klasifikasi adalah *Support Vector Machine* serta tidak membandingkannya dengan algoritma lain.
4. *Software tools* yang digunakan dalam membuat sistem klasifikasi ini adalah HTML, PHP, XAMPP dan MySQL. Serta menggunakan *localhost* sebagai web server dalam pembuatan website klasifikasi berita *online* berbahasa Indonesia dan pengujian sistem dilakukan secara *online*.

1.4 Tujuan

Tujuan dari penelitian ini adalah menerapkan algoritma *Support Vector Machine* untuk melakukan klasifikasi pada dokumen berita *online* berbahasa Indonesia berdasarkan kategori konten berita.

1.5 Manfaat

Manfaat dari penelitian ini adalah diperolehnya sebuah sistem yang dapat membantu editor berita dalam melakukan klasifikasi berita *online* berbahasa Indonesia berdasarkan kategori berita secara otomatis.

1.6 Sistematika Penulisan

Untuk memberikan gambaran secara menyeluruh terhadap penelitian yang dilakukan, maka sistematika penulisan dibagi dalam beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, sistematika penulisan dan jadwal kegiatan dalam melakukan penelitian.

BAB II LANDASAN TEORI

Bab ini memuat tentang tinjauan pustaka serta definisi dan teori-teori pendukung yang digunakan sebagai acuan atau dasar dalam melakukan penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini membahas mengenai metodologi yang digunakan dalam penelitian. Mencakup metodologi pengumpulan data, metode pengembangan perangkat lunak, metode klasifikasi dan skenario pengujian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi penjelasan hasil dan pembahasan mengenai klasifikasi pada berita *online* berbahasa Indonesia. Hasil klasifikasi yang didapatkan menggunakan algoritma *Support Vector Machine*.

BAB V KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan dari apa yang telah diuraikan pada bab sebelumnya serta saran untuk perkembangan penelitian yang akan data.

DAFTAR PUSTAKA

1.7 Jadwal Kegiatan

Tabel 1.1. Jadwal Kegiatan

KEGIATAN	BULAN DAN MINGGU																			
	Maret				April				Mei - Agustus				September				Oktober			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Penyusunan dan Seminar Proposal																				
Analisis dan Desain																				
Koding dan Pengujian																				
Pembuatan Laporan																				
Seminar Hasil																				
Seminar Tugas Akhir																				

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam melakukan penelitian ini, terdapat beberapa penelitian terkait yang telah dilakukan sebagai referensi. Penelitian-penelitian tersebut adalah sebagai berikut.

Penelitian yang dilakukan oleh Fatmawati dan Muhammad Affandes (2017), Teknik Informatika, UIN Sultan Syarif Kasim Riau dengan judul "*Klasifikasi Keluhan Menggunakan Metode Support Vector Machine (Studi Kasus: Akun Facebook Group iRaise Helpdesk)*". Penelitian ini bertujuan untuk melakukan pengklasifikasian permasalahan pada sistem iRaise berdasarkan empat kategori keluhan, yaitu login, krs, nilai dan personal dengan menggunakan metode *Support Vector Machine* (SVM). Prosesnya dilakukan dengan menggunakan fungsi LIBSVM pada aplikasi RapidMiner. Tahapan dalam *text mining* yang digunakan meliputi proses *text preprocessing*, *text transformartion* baru kemudian dilakukan klasifikasi data menggunakan metode SVM. Penelitian ini memberikan hasil akurasi tertinggi sebesar 95,67%.

Penelitian selanjutnya dilakukan oleh Delias Hendra Pratama (2013), Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor dengan judul "*Implementasi Support Vector Machine (SVM) Untuk Klasifikasi Dokumen*". Penelitian ini bertujuan untuk melakukan klasifikasi pada dokumen teks dengan mengimplementasikan algoritma *Support*

Vector Machine. Proses *text mining* yang diterapkan meliputi *preprocessing* dan pembelajaran menggunakan algoritma SVM untuk membuat model klasifikasi. Hasil akurasi pada penelitian adalah 76% dengan jumlah dokumen uji sebanyak 150 dengan data latih sebanyak 174.

Penelitian selanjutnya terkait algoritma *Support Vector Machine* dilakukan oleh Amelia Mustika dan Muhammad Affandes (2015), UIN Sultan Syarif Kasim Riau dengan judul "Penerapan Metode *Support Vector Machine* Dalam Klasifikasi Sentimen Tweet *Public Figure*". Penelitian ini bertujuan untuk melakukan klasifikasi sentimen tweet Bahasa Indonesia ke dalam kelas sentimen positif atau negatif dengan menggunakan metode *Support Vector Mahcine*. Proses SVM dilakukan dengan menggunakan fungsi LIBSVM pada aplikasi Weka. Model pembelajaran SVM diperoleh dari data latih sebanyak 800 tweet dan selanjutnya digunakan untuk mengklasifikasi 200 data uji. Dari pengujian yang telah dilakukan diperoleh akurasi klasifikasi tertinggi sebesar 72,5%.

Penelitian yang dilakukan oleh Andi Ahmad Irfa, dkk (2018), Fakultas Informatika, Universitas Telkom Bandung dengan judul "*Klasifikasi Topik Berita Berbahasa Indonesia menggunakan k-Nearest Neighbor",* melakukan klasifikasi berita sesuai dengan kategori masing-masing yang telah ditentukan dan menganalisa tingkat performa sistem yang dibangun dengan metode klasifikasi *k-nearest neighbor*. Pada penelitian ini perancangan sistem dilakukan meliputi proses pengumpulan dataset, *preprocessing data*, klafikasifikasi dengan k-nn, dan terakhir dilakukan pengujian sistem. Dalam penelitian ini sistem yang dibangun mampu menghasilkan performa *micro average f1-measure* sebesar 69,9%.

Adapun penelitian yang dilakukan oleh Wayan Firdaus Mahmudy, dkk (2015) yang berjudul "*Klasifikasi Artikel Berita Secara Otomatis menggunakan Metode Naïve Bayes Classifier yang Dimodifikasi*", melakukan klasifikasi pada artikel berita secara otomatis dengan menerapkan metode *Naïve Bayes Classifier*. Penelitian dilakukan pada 900 dokumen berita yang dibagi menjadi 9 kategori, sehingga masing-masing kategori diujikan 100 dokumen. Percobaan dilakukan dengan membandingkan proporsi pada data latih dan data uji. Secara berurutan kombinasi dokumen latih dan uji tersebut antara lain 5:95, 10:90, 15:85, 20:80, 25:75 dan 30:70. Dengan menggunakan metode Naïve Bayes Classifier didapatkan akurasi hasil klasifikasi berturut-turut 54%, 65%, 65%, 69%, 71% dan 76%.

Kemudian penelitian yang dilakukan oleh Enggal Suci Febriani (2017), Program Studi Teknik Informatika, Fakultas Teknik Universitas Nusantara PGRI Kediri yang berjudul "*Klasifikasi Konten Berita Surat Kabar Berdasarkan Judul dengan Text Mining Menggunakan Metode Naïve Bayes"*, mengklasifikasikan berita secara otomatis sehingga penyusunan surat kabar harian bisa terselesaikan tepat waktu. Tahapan *text mining* yang digunakan dalam penelitian ini meliputi *Text Preprocessing*, *Text Transformation*, *Feature Selection* dan *Pattern Discovery*. Dari pengujian klasifikasi 200 data training dan 80 data testing diperoleh jumlah data yang benar sebanyak 230 dan data yang salah sebanyak 50 dengan tingkat akurasi sebanyak 79%.

Berikut Tabel 2.1 merupakan daftar penelitian sejenis yang telah dilakukan sebelumnya yang digunakan sebagai referensi dalam melakukan penelitian ini.

Tabel 2.1. Studi Literatur Sejenis

Peneliti	Fatmawati dan Muhammad Affandes (2017)	Delias Hendra Pratama (2013)	Amelia Mustika dan Muhammad Affandes (2015)	Andi Ahmad Irfa, dkk (2018)	Wayan Firdaus Mahmudy, dkk (2015)	Enggal Suci Febriani (2017)
Judul	Klasifikasi Keluhan Menggunakan Metode *Support Vector Machine* (Studi Kasus: Akun Facebook Group iRaise Helpdesk)	Implementasi *Support Vector Machine* (SVM) Untuk Klasifikasi Dokumen	Penerapan Metode *Support Vector Machine* Dalam Klasifikasi Sentimen Tweet *Public Figure*	Klasifikasi Topik Berita Berbahasa Indonesia Menggunakan *k-Nearest Neighbor*	Klasifikasi Artikel Berita Secara Otomatis Menggunakan Metode *Naïve Bayes Classifier* Yang Dimodifikasi	Klasifikasi Konten Berita Surat Kabar Berdasarkan Judul Dengan *Text Mining* Menggunakan Metode *Naïve Bayes*

Tabel 2.1. Studi Literatur Sejenis (Lanjutan)

Metode	*Support Vector Machine* (SVM)	*Support Vector Machine* (SVM)	*Support Vector Machine* (SVM)	*K-Nearest Neighbor* (K-NN)	*Naïve Bayes Classifier* (NBC)	*Naïve Bayes*
Data	Data yang digunakan bersumber dari postingan pelanggan pada akun Facebook *Group* iRaise Helpdesk. Data yang diambil adalah data	Data yang digunakan dalam penelitian ini adalah dokumen hasil penelitian dari Jurnal Penelitian Holtikultura tahun 2002 sampai dengan tahun	Sumber data dalam penilitian diperoleh melalui data *tweet* yang di unduh dengan memanfaatkan twitter API (*Application Programming Interface*). Data	Data yang digunakan pada penelitian Tugas Akhir ini berupa teks berita yang terdiri dari bermacam-macam label kelas, seperti politik, budaya, ekonomi, otomotif,	Koleksi berita bahasa Indonesia yang diambil sebagai objek penelitian bersumber dari www.kompas. com, yang merupakan salah satu situs berita	Pada penelitian ini, data bersumber dari Radar Kediri yang merupakah sebuah lembaga surat kabar harian

Tabel 2.1. Studi Literatur Sejenis (Lanjutan)

Data	postingan yang hanya mengandung kalimat keluhan mengenai permasalahan sistem iRaise. Dataset yang digunakan berjumlah 1040 data	2009. Dokumen tersebut terdiri dari 174 dokumen latih dan 150 dokumen uji	*tweet* yang dikumpulkan antara *username* dan isi *tweet* opini yang hanya mengandung kata kunci nama-nama seperti "Jokowi", "Prabowo", "Jusuf Kala", dan "Hatta Rajasa" dengan total 1000 *tweet*	dll. Data teks berita didapatkan dari berbagai portal berita yaitu, website kompas.com, tribunnews.com, republika.com, mediaindonesia.com dan sindonews.com	berbahasa Indonesia yang banyak diakses oleh pencari berita ditanah air. Koleksi berita ini terdiri atas 900 dokumen	yang terbit di Kediri, Jawa Timur.Dengan jumlah dataset sebanyak 280 data

Tabel 2.1. Studi Literatur Sejenis (Lanjutan)

Tujuan Penelitian	Menerapkan algoritma *Support Vector Machine* untuk melakukan pengklasifikasian permasalahan pada sistem iRaise berdasarkan empat kategori keluhan, yaitu Login, KRS, Nilai dan Personal	Melakukan klasifikasi pada dokumen teks penelitian Holtikultura dengan mengimplementasikan algoritma *Support Vector Machine* berdasarkan tiga kelas, yaitu Ekofisiologi dan	Penelitian ini bertujuan untuk melakukan klasifikasi sentimen tweet Bahasa Indonesia ke dalam kelas sentimen positif atau negatif dengan menggunakan metode *Support Vector Mahcine*	Mengklasifikan berita sesuai dengan kategori masing-masing yang telah ditentukan dengan menggunakan metode klasifikasi *k-nearest neighbors* dan menganalisa tingkat performa sistem yang	Melakukan klasfikasi konten berita secara otomatis menggunakan algoritma *naïve bayes classifier* (NBC)	Mengklasifikasikan berita secara otomatis sehingga penyusunan surat kabar harian bisa terselesaikan tepat waktu

Tabel 2.1. Studi Literatur Sejenis (Lanjutan)

Tujuan Penelitian		Agronomi, Pemuliaan dan Teknologi Benih, Proteksi (Hama dan Penyakit)		dibangun dengan metode klasifikasi *k-nearest neighbors*		
Hasil Penelitian	Algoritma SVM berhasil diterapkan untuk melakukan klasifikasi data dalam kategori *multiclass* terhadap keluhan	Algoritma *Support Vector Machine* cukup baik digunakan untuk mengembangkan sistem klasifikasi dokumen teks.	Algoritma *Support Vector Machine* (SVM) dapat diterapkan untuk mengklasifikasi *tweet* sentimen *public figure*. Dari pengujian yang	Sistem yang dibangun dengan rasio perbandingan data latih dan uji 80%:20% memiliki performa paling tinggi, sehingga dapat disimpulkan	Percobaan dilakukan dengan membandingkan proporsi pada data latih dan data uji.	Dari pengujian klasifikasi 200 data training dan 80 data testing diperoleh jumlah data

Tabel 2.1. Studi Literatur Sejenis (Lanjutan)

Hasil Penelitian	pada sistem iRaise berdasarkan postingan pada akun Facebook iRaise Helpdesk. Penelitian ini memberikan hasil akurasi tertinggi sebesar 95,67%.	Hasil akurasi terbaik adalah 76 %.	telah dilakukan diperoleh akurasi klasifikasi tertinggi sebesar 72,5%.	bahwa jumlah data latih yang semakin banyak akan menambah ketepatan klasifikasi dikarenakan sistem akan banyak mendapatkan informasi dari data latih.	Dengan menggunakan metode *Naïve Bayes Classifier* didapatkan akurasi hasil klasifikasi berturut-turut 54%, 65%, 65%, 69%, 71% dan 76%	yang benar sebanyak 230 dan data yang salah sebanyak 50 dengan tingkat akurasi sebanyak 79%

Topik penelitian dalam skripsi ini mengenai klasifikasi pada teks berita *online* berbahasa Indonesia menggunakan algoritma *Support Vector Machine* berdasarkan topik berita. Data berita *online* yang digunakan berjumlah 1000 data yang dikumpulkan dengan *range* waktu dari 2016 – 2020 melalui situs berita *online* yaitu *detik.com, kompas.com* dan *okezone.com*. Data terdiri dari lima macam label kelas, yaitu Lifestyle, Olahraga, Politik, Ekonomi dan Teknologi, sehinga masing-masing kategori terdiri dari 200 data berita. Hasil akhir penelitian ini adalah sebuah *website* klasifikasi berita *online* yang dapat digunakan untuk melakukan klasifikasi kategori berita *online* serta pengelolaan sistem klasifikasi.

2.2 Berita

Dalam gambaran yang sederhana, seperti dilukiskan dengan baik oleh pakar jurnalistik, berita adalah apa yang ditulis surat kabar, apa yang disiarkan radio, dan apa yang ditayangkan televisi. Berita menampilkan fakta, tetapi tidak setiap fakta merupakan berita. Berita biasanya menyangkut orang-orang, tetapi tidak setiap orang bisa dijadikan berita. Berita merupakan sejumlah peristiwa yang terjadi di dunia, tetapi hanya sebagian kecil saja yang dilaporkan. Berita adalah laporan tercepat mengenai fakta atau ide terbaru yang benar, menarik dan atau penting bagi sebagian besar khalayak, melalui media berkala seperti surat kabar, radio, televisi, atau media *online* seperti internet (Sumadiria, 2014).

Berdasarkan riset yang dilakukan oleh IDN Media terhadap perilaku milenial di Indonesia melalui Indonesia Millenial Report 2019, terdapat 5 konten berita yang paling sering dibaca oleh remaja Indonesia pada tahun 2019. Konten

tersebut meliputi Lifestyle, Olahraga, Politik, Ekonomi dan Teknologi (Utomo, 2019).

2.2.1 Lifestyle

Lifestyle atau gaya hidup adalah perilaku seseorang yang ditunjukkan dalam aktivitas, minat dan opini khususnya yang berkaitan dengan citra diri untuk merefleksikan status sosialnya. Gaya hidup merupakan *frame of reference* yang dipakai sesorang dalam bertingkah laku dan konsekuensinya akan membentuk pola perilaku tertentu. Terutama bagaimana seseorang ingin dipersepsikan oleh orang lain, sehingga gaya hidup sangat berkaitan dengan bagaimana ia membentuk image di mata orang lain, berkaitan dengan status sosial yang disandangnya. Untuk merefleksikan image inilah, dibutuhkan simbol-simbol status tertentu, yang sangat berperan dalam mempengaruhi perilaku konsumsinya. Gaya hidup bisa dilihat dari cara berpakaian, kebiasaan, dan lain-lain. Gaya hidup bisa dinilai relatif tergantung penilaian dari orang lain. Gaya hidup juga bisa dijadikan contoh dan juga bisa dijadikan hal tabu. Contoh gaya hidup baik: makan dan istirahat secara teratur, makan makanan 4 sehat 5 sempurna, dan lain-lain. Contoh gaya hidup tidak baik: berbicara tidak sepatutnya, makan sembarangan, dan lain-lain (Hendariningrum & Susilo, 2014).

2.2.2 Olahraga

Secara umum pengertian olahraga adalah sebagai salah satu aktivitas fisik maupun psikis seseorang yang berguna untuk menjaga dan meningkatkan kualitas

kesehatan seseorang. Olahraga adalah budayasa manusia, artinya tidak dapat disebut kegiatan olahraga apabila tidak ada faktor manusia yang berperan secara ragawi/pribadi melakukan aktivitas olahraga itu. Demikianlah maka manusia adalah titik sentral dari olahraga, artinya tidak ada olahraga apabila tidak ada faktor manusia yang secara ragawi berperan melakukan olahraga itu. Oleh karena itu olahraga menuntut persyaratan-persyaratan yang harus dipenuhi oleh manusia, baik secara jasmani, rohani, maupun sosial (Griwijoyo & Sidik, 2012).

Berita olahraga mungkin menjadi salah satu berita yang membuat bayak orang tertarik untuk membacanya. Berita seputar olahraga tak hanya mencakup soal sepakbola saja. Akan tetapi juga bisa mengulas kegiatan olahraga yang lainnya. Tentu saja ada banyak jenis jenis olahraga yang sering dipertandingkan di Indonesia.

2.2.3 Politik

Politik berasal dari bahasa Belanda *politiek* dan bahasa Inggris *politics*, yang masing-masing bersumber dari bahasa *Yunani τα πολιτικά* (politika - yang berhubungan dengan negara) dengan akar katanya *πολίτης* (polites - warga negara) dan *πόλις* (polis - negara kota). Secara etimologi kata "politik" masih berhubungan dengan polisi kebijakan. Kata "politis" berarti hal-hal yang berhubungan dengan politik. Kata "politisi" berarti orang-orang yang menekuni hal politik (Faruqi, 2015).

Dalam kehidupan sehari-hari istilah “politik” sudah tidak begitu asing, karena segala sesuatu yang dilakukan atas dasar kepentingan kelompok atau

kekuasaan sering kali diatasnamakan dengan label politik. Pengangkatan atau pencopotan seorang pejabat kepala kantor misalnya, kadang dilakukan atas pertimbangan politik. Konflik yang terjadi dengan memicu pertarungan antar etnis atau agama, juga disebutkan karena politik. Gencarnya pemberitaan tentang teroris dalam media massa juga dinilai memiliki muatan politik (Cangara, 2014).

2.2.4 Ekonomi

Dalam penggunaannya di masa sekarang, istilah "ekonomi" memiliki beberapa makna. Makna-makna yang berbeda ini tidak sepenuhnya lepas satu sama lain karena masing-masing dari makna ini memiliki ide utama yang menunjukkan pendekatan yang digunakan untuk mendefinisikan pokok bahasan (*subject matter*) dari ilmu ekonomi. Yang pertama, istilah "ekonomi" kadang digunakan untuk merujuk pada cara melakukan tindakan, seperti misalnya pada kata "*economically*" (bertindak secara ekonomis atau hemat). Dalam artian ini ekonomi berarti efisiensi, pengerahan upaya minimal (dengan hasil maksimal) dan adanya adaptasi terhadap cara-cara yang digunakan untuk mencapai tujuan. Yang kedua, istilah "ekonomi" kadang juga digunakan untuk mendapatkan kebutuhan yang diinginkan atau dibutuhkan (seperti misalnya dalam produksi). Makna ini sering kali disampaikan dengan istilah "*provisioning*" (yaitu perdagangan barang dan jasa). Makna ketiga dari istilah "ekonomi" adalah merujuk pada institusi-institusi pasar. Institusi-institusi dalam pasar adalah perwujudan yang paling menyolok dari upaya pencapaian efisiensi dalam kegiatan-kegiatan yang ditujukan

untuk mendapatkan barang dan jasa yang menjadi kebutuhan (Khairani & Saleh, 2020).

2.2.5 Teknologi

Teknologi adalah keseluruhan sarana untuk menyediakan barang yang dibutuhkan untuk kelangsungan hidup dan kenyamanan hidup manusia. Penggunaan istilah 'teknologi' dalam bahasa Inggris: *technology* telah berubah secara signifikan selama 200 tahun terakhir. Sebelum abad ke-20, istilah ini tidak umum dalam bahasa Inggris, dan biasanya mengacu pada penggambaran atau seni terapan penilaian. Istilah ini sering dikaitkan dengan pendidikan teknis, seperti Massachusetts Institute of Technology (didirikan pada tahun 1861). Teknologi dalam arti ini dapat diketahui melalui barang-barang, benda-benda, atau alat-alat yang berhasil dibuat oleh manusia untuk memudahkan dan menggampangkan realisasi hidupnya di dalam dunia. Teknologi juga penerapan keilmuan yang mempelajari dan mengembangkan kemampuan dari suatu rekayasa dengan langkah dan teknik tertentu dalam suatu bidang. Teknologi merupakan Aplikasi ilmu dan *engineering* untuk mengembangkan mesin dan prosedur agar memperluas dan memperbaiki kondisi manusia atau paling tidak memperbaiki efisiensi manusia pada beberapa aspek (Kurniawan, 2019).

2.3 *Text Mining*

Text mining merupakan salah satu cabang ilmu *data mining* yang menganalisis data berupa dokumen teks. Menurut Han, Kamber, dan Pei (dalam

Prilianti dan Wijaya, 2014), *text mining* adalah satu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. Ide awal pembuatan *text mining* adalah untuk menemukan pola-pola informasi yang dapat digali dari suatu teks yang tidak terstruktur. Dengan demikian, *text mining* mengacu juga kepada istilah text data mining atau penemuan pengetahuan dari basis data teks.

Text mining adalah cabang bidang ilmu yang menggabungkan *data mining, web mining, information retrieval, information extraction, computational linguistics* dan *natural language processing* (Sumathy & Chidambaram, 2013). Gambar 2.1 menggambarkan diagram venn dari *text mining*.

Gambar 2.1. Diagram Venn pada *Text Mining*

Sumber: Jurnal Internasional Aplikasi Komputer, 2013

Saat ini, *text mining* telah mendapat perhatian dalam berbagai bidang, antara lain dibidang keamanan, biomedis, pengembangan perangkat lunak dan aplikasi, media *online*, pemasaran, dan akademik. Seperti halnya dalam *data mining*,

aplikasi *text mining* pada suatu studi kasus, harus dilakukan sesuai prosedur analisis.

2.3.1 *Information Extraction*

Information extraction adalah proses mengekstrak struktur informasi dokumen teks secara otomatis dari tidak terstruktur dan/atau semi terstruktur. Sebuah sistem *information extraction* melibatkan pengidentifikasian entitas seperti nama orang, perusahaan dan lokasi, atribut dan relasi antar entitas. *Information extraction* melakukannya berdasarkan aturan yang diakui. Penerapannya biasanya digunakan sebagai proses pencarian untuk rangkaian teks yang belum terdefinsi dalam sebuah dokumen. *Information extraction* memberikan solusi untuk perubahan sebuah kesatuan dokumen tekstual menjadi *database* yang lebih terstruktur. *Database* yang terkonstruksi oleh modul *information extraction* dapat digunakan untuk penggalian pengetahuan lebih lanjut (Sumathy & Chidambaram, 2013).

2.3.2 *Information Retrieval*

Information retrieval didefinisikan sebagai metode yang digunakan untuk representasi, penyimpanan dan akses informasi dimana informasi ditangani sebagian besar dalam bentuk dokumen tekstual seperti koran, iklan, buku yang diambil dari *database* berdasarkan permintaan atau *query* pengguna. *Information retrieval* dianggap sebagai perluasan pengambilan dokumen di mana dokumen diproses untuk disingkat ataupun diekstrak sebagian informasi yang diinginkan

pengguna. Penerapan *infortmation retrieval* akan mempercepat analisa secara signifikan dengan mengurangi jumlah dokumen untuk analisa (Sumathy & Chidambaram, 2013).

2.3.3 *Natural Language Processing*

Natural Language Processing (NLP) merupakan salah satu cabang ilmu Artificial Intelligence (AI) yang fokus pada pengolahan bahasa natural. Bahasa natural adalaah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi. Bahasa yang diterima oleh komputer harus diproses dan di pahami terlebih dahulu supaya maksud dari *user* dapat dipahami. Pada prinsipnya, bahasa natural adalah suatu bentuk representasi dari satu pesan yang ingin dikomunikasikan antar manusia.

Natural Language Processing berfokus pada *Natural Language Generation (NLG) dan Natural Language Understanding (NLU).* Kebanyakan sistem NLG termasuk *syntactic realizer* untuk memastikan aturan tata bahasa seperti kata kerja persetujuan subjek yang ditaati dan *text planner* untuk menentukan bagaimana mengurutkan kalimat, paragraf, dan bagian lainnya secara koheren. Aplikasi NLG yang terkenal adalah mesin translasi (Sumathy & Chidambaram, 2013).

2.3.4 *Data Mining*

Data mining merupakan suatu kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menentukan keteraturan, pola atau hubungan dalam set data berukuran besar. Data *mining* mengarah kepada penemuan informasi

relevan atau penemuan pengetahuan dari jumlah data yang banyak. Data *mining* mencoba untuk menemukan aturan statistik dan pola secara otomatis. *Tools* data *mining* dapat memprediksi kebiasaan dan tren ke depan yang bermanfaat bagi perusahaan untuk membuat pengetahuan positif. Tujuan keseluruhan dari data *mining* adalah untuk mengekstrak informasi dari kumpulan data dan mengubahnya menjadi struktur yang dapat dimengerti untuk analisa selanjutnya (Sumathy & Chidambaram, 2013).

Secara umum, kegunaan data *mining* dapat dibagi menjadi dua: deksriptif dan prediktif. Deskriptif berarti data *mining* digunakan untuk mencari pola-pola yang dapat dipahami manusia yang menjelaskan karakteristik data. Prediktif berarti data *mining* digunakan untuk membentuk sebuah model pengetahuan yang akan digunakan untuk melakukan prediksi. Berdasarkan fungsionalitasnya, tugas-tugas data *mining* bisa dikelompokkan ke dalam enam kelompok berikut ini (Suryanto, 2017):

1. Klasifikasi (*classification*): men-generalisasi struktur yang diketahui untuk diaplikasikan pada data-data baru. Misalkan klasifikasi penyakit ke dalam sejumlah jenis, klasifikasi email ke dalam spam atau bukan.
2. Klasterisasi (*clustering*): mengelompokkan data, yang tidak diketahui label kelasnya, ke dalam sejumlah kelompok tertentu sesuai dengan ukuran kemiripannya.
3. Regresi (*regression*): menemukan suatu fungsi yang memodelkan data dengan galat (kesalahan prediksi) seminimal mungkin.

4. Deteksi anomali (*anomaly detection*): mengidentifikasi data yang tidak umum, bisa berupa *outlier* (pencilan), perubahan atau deviasi yang mungkin sangat penting dan perlu investigasi lebih lanjut.
5. Pembelajaran aturan asosiasi (*association rule learning*) atau pemodelan kebergantungan (*dependecy modelling*): mencari relasi antar variabel.
6. Perangkuman (*summarization*): menyediakan representasi data yang lebih sederhana, meliput visualisasi dan pembuatan laporan.

2.4 Proses-proses *Text Mining*

2.4.1 *Text Pre-Processing*

Text pre-processing dilakukan untuk membuang data yang tidak konsisten, duplikasi data serta memperbaiki kesalahan data. *Text pre-processing* bertujuan untuk mempersiapkan teks menjadi data yang diproses pada tahapan berikutnya (Uysal & Gunal, 2014).

2.4.1.1 *Remove Punctuation*

Proses ini mengacu pada mengahapus pada teks berupa karakter-karakter selain alphabet yang bertujuan untuk mengurangi noise. Tahapan-tahapn dalam proses *remove punctuation* yaitu:

1. Membaca data masukan berupa teks
2. Memeriksa apakah ada karakter-karakter selain alphabet. Jika tidak ditemukan maka algoritma berakhir.
3. Jika ada hapus karakter yang bukan termasuk kedalam alphabet.

4. Algoritma selesai.

2.4.1.2 *Case Folding*

Case folding adalah tahapan pemrosesan teks dimana semua teks diubah ke dalam bentuk yang sama. Hal ini dilakukan dengan mengubah kata menjadi *lower case* atau huruf kecil (Luqyana, 2018). Pada penelitian ini semua huruf dalam teks dokumen diseragamkan menjadi huruf kecil. Tahapan-tahapan yang dilakukan yaitu :

1. Membaca data masukan berupa teks
2. Mengecek apakah ada huruf kapital (Jika menggunakan *lower case*). Jika ada maka ke tahap 3, jika tidak maka algoritma selesai.
3. Lakukan perubahan string lower pada setiap string.
4. Algoritma selesai

2.4.1.3 Tokenisasi

Langkah pertama setelah mengekstraksi konten dari sebuah berkas adalah memilih teks berukuran kecil yang bisa digunakan kembali, yang biasa disebut dengan *tokens*. *Tokens* sering kali mempresentasikan satu kata. Cara pertama paling umum dalam melakukan tokenisasi adalah dengan memilah *string* berdasarkan karakter spasi yang ditemukan (Putra, et.al. 2018). Tahapan-tahapan yang dilakukan dalam proses tokenisasi adalah sebagai berikut:

1. Membaca data masukan berupa teks

2. Memeriksa teks dengan mengecek apakah menemukan spasi. Jike bertemu spasi maka ke tahap 3. Jika tidak maka *scan* kembali.
3. Mengambil kata (term) dari teks dan membuat indeks baru untuk menandai kata (term).
4. Algoritma selesai.

2.4.1.4 *Stopword Filtering*

Stopwords adalah kata-kata yang terlalu sering diantara *file* teks yang tidak memiliki arti yang khusus dalam *Information Retrieval*. Penghilangan *stopwords* memiliki keunggulan yang penting. Mengurangi ukuran struktur pengindeksan. Untuk setiap kata pada teks jika dia adalah *stopword* maka kata tersebut termasuk kata yang sia-sia.

Tahapan-tahapan yang dilakukan yaitu :

1. Membaca data masukan berupa teks
2. Mengecek apakah ada kata atau *token* yang termasuk sebagai *stopwords*. Jika tidak ditemukan, maka algoritma berakhir. Jika ada hapus kata atau token tersebut.
3. Algoritma selesai

2.4.2 *Text Transformation*

2.4.2.1 *Feature Extraction*

Ekstraksi fitur adalah proses reduksi atribut. Tidak seperti pemilihan fitur, yang memilih dan mempertahankan atribut yang paling signifikan, Ekstraksi fitur

benar-benar mengubah atribut. Atribut atau fitur yang ditransformasikan adalah kombinasi linier dari atribut aslinya. Proses ekstraksi fitur menghasilkan seperangkat atribut yang jauh lebih kecil dan lebih kaya. Jumlah maksimum fitur dapat ditentukan pengguna atau ditentukan oleh algoritma. Secara default, algoritma menentukannya. Model yang dibangun dengan fitur yang diekstraksi bisa lebih berkualitas, karena atribut yang lebih sedikit dan lebih bermakna menggambarkan data (Wang, et.all. 2016).

Bag of words digunakan untuk menunjukkan representasi sederhana dari teks yang digunakan dalam model pengambilan dan klasifikasi. Dalam representasi ini, sebuah dokumen dipertimbangkan sebagai koleksi kata yang tidak punya hubungan, tidak *syntactic* ataupun *statistical*, atau diantaranya. Model *bag of words* mengasumsikan tidak adanya hubungan antar kata, jadi kita melihat bagaimana istilah ketergantungan dapat diambil dan digunakan dalam model linear berbasis fitur.

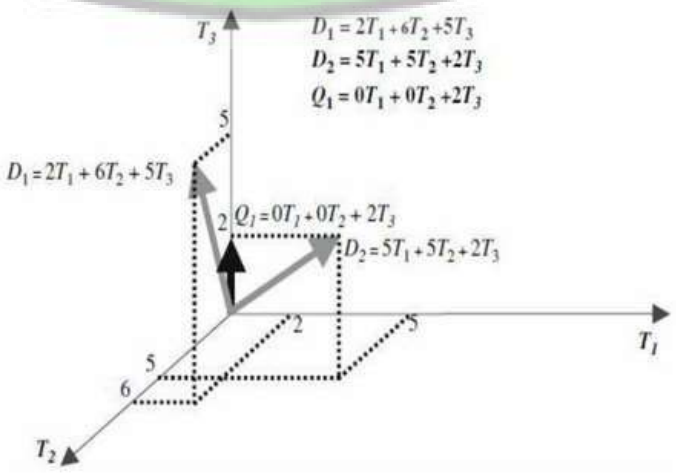
Dalam mengukur kemiripan, dokumen didefinisikan berdasarkan representasi *bag-of-words* yang dikonversi ke suatu model ruang vektor *(vector space model*, VSM). Setiap dokumen di dalam *database* dan *query* pengguna direpresentasikan oleh suatu vektor multi-dimensi. Dimensi sesuai dengan jumlah *term* dalam dokumen yang terlibat. Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks *term-document* (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokumen yang ditentukan. Nilai nol berarti bahwa *term* tersebut tidak hadir di dalam dokumen.

Pada Gambar 2.2 menampilkan contoh matriks *term-document* untuk *database* dengan n dokumen t *term*.

$$\begin{bmatrix} & T_1 & T_2 & \cdots & T_t \\ D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\ D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ D_n & w_{1n} & w_{2n} & \cdots & w_{tn} \end{bmatrix}$$

Gambar 2.2. Contoh matriks *term-document*

Keberhasilan dari model VSM ini ditentukan oleh skema pembobotan terhadap suatu *term* baik untuk cakupan lokal maupun global, dan faktor normalisasi. Pembobotan lokal hanya berpedoman pada frekuensi munculnya *term* dalam suatu dokumen dan tidak melihat frekuensi kemunculan term tersebut di dokumen lainnya. Contoh model ruang vektor (VSM) dapat dilihat pada Gambar 2.3.



Gambar 2.3. Contoh *Vector Space Models*

2.4.2.2 TF-IDF

Data yang telah melalui tahap *pre-processing* harus berbentuk numerik. Untuk mengubah data tersebut menjadi numerik yaitu menggunakan metode pembobotan TF-IDF. Metode *Term Frequency Invers Document Frequency* (TF-IDF) merupakan metode yang digunakan menentukan seberapa jauh keterhubungan kata (*term*) terhadap dokumen dengan memberikan bobot setiap kata. Metode TF-IDF ini menggabungkan dua konsep yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut (Fitri, 2013). Dalam perhitungan bobot menggunakan TF-IDF, dihitung terlebih dahulu nilai TF perkata dengan bobot masing-masing kata adalah 1. Sedangkan nilai IDF diformulasikan pada Persamaan berikut.

$$IDF(word) = log\frac{td}{df} \quad \text{.....(2.1)}$$

IDF(*word*) adalah nilai IDF dari setiap kata yang akan di cari, td adalah jumlah keseluruhan dokumen yang ada, df jumlah kemuculan kata pada semua dokumen. Untuk memperoleh TF-IDF cukup dengan mengkalikan nilai dari TF dengan IDF.

2.4.3 *Pattern Discovery*

Pattern discovery adalah penemuan pola yang terdapat pada dataset. Pola adalah kumpulan item yang seringkali muncul pada dataset dan berkolerasi secara kuat. Pola biasanya bersifat *intrinsic* dan properti yang penting dari suatu data. Oleh karena itu perlu ditemukan dan dalam hal ini *text mining* menjadi data *mining*. Metode data mining seperti *clustering, classification, information*

retrieval, dan lain-lain dapat digunakan untuk *pattern discovery*. Dalam *text mining*, tahap ini berusaha menemukan pola atau pengetahuan dari keseluruhan teks (Mahmudy, 2014).

2.5 Klasifikasi

Teknik klasifikasi digunakan untuk mempelajari sekumpulan data sehingga dihasilkan aturan yang bisa mengklasifikasi atau mengenali data-data baru yang belum pernah dipelajari. *Classification*/klasifikasi dalam pengertian komputasi, berusaha memberi label pada data. Dengan satu set fitur untuk sebuah objek, *classifier* mencoba menetapkan label pada objek itu. *Classifier* bekerja dengan belajar berdasarkan pengetahuan yang berasal dari contoh benda lain yang telah diberi label. Contoh-contoh ini disebut sebagai data pelatihan, berfungsi sebagai sumber pengetahuan sebelumnya yang digunakan oleh pengklasifikasi untuk membuat keputusan tentang objek yang sebelumnya tak terlihat (Ingersol, et.al. 2012).

Classification/klasifikasi dalam bidang komputasi adalah pencarian untuk menentukan label-label terhadap data. *Classifier* melakukan ini melalui *knowledge* yang di dapat dari label-label yang terdapat pada data contoh. Contoh-contoh tersebut disebut dengan *training data*, disajikan sebagai sumber pengetahuan yang digunakan oleh *classifier* untuk membuat keputusan terhadap objek-objek yang belum terlihat sebelumnya. Klasifikasi dapat didefinisikan secara detail sebagai suatu pekerjaan yang melakukan pelatihan/pembelajaran terhadap fungsi target *f*

yang memetakan setiap vektor (set fitur) x ke dalam satu dari sejumlah label kelas y yang tersedia (Prasetyo, 2014).

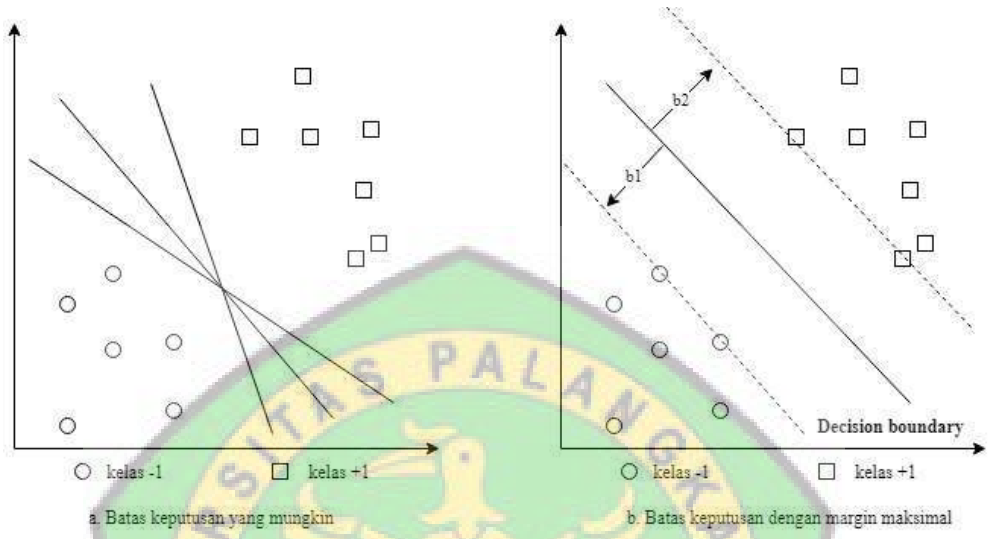
2.6 *Support Vector Machine*

2.6.1 Konsep Dasar *Support Vector Machine*

Support vector machine (SVM) adalah suatu teknik yang relatif baru untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi, yang sangat populer belakangan ini. Metode ini berakar dari teori pembelajaran statistik yang hasilnya sangat menjanjikan untuk memberikan hasil yang lebih baik dari metode lain. SVM bekerja dengan baik pada set data dengan dimensi yang tinggi. Pada SVM, hanya sejumlah data terpilih sajalah yang berkontribusi untuk membentuk model yang digunakan dalam klasifikasi yang akan dipelajari. SVM hanya menyimpan sebagian kecil saja dari data latih untuk digunakan pada saat prediksi. Hal inilah yang menjadi kelebihan SVM karena tidak semua data latih akan dipandang untuk dilatih dalam setiap iterasi pelatihannya. Data-data yang berkontribusi tersebut disebut *support vector* sehingga metodenya juga disebut *Support Vector Machine* (Prasetyo, 2014).

Ide dasar SVM adalah memaksimalkan batas *hyperplane*, yang diilustrasikan seperti pada Gambar 2.4. Pada gambar (a) ada sejumlah pilihan *hyperplane* yang mungkin untuk set data, sedangkan gambar (b) merupakan *hyperplane* dengan margin yang paling maksimal. Meskipun sebenarnya pada gambar (a) bisa juga menggunakan hyperplane sembarang, tetapi *hyperplane*

dengan margin yang maksimal akan memberikan generalisasi yang lebih baik pada metode klasifikasi (Prasetyo, 2014).

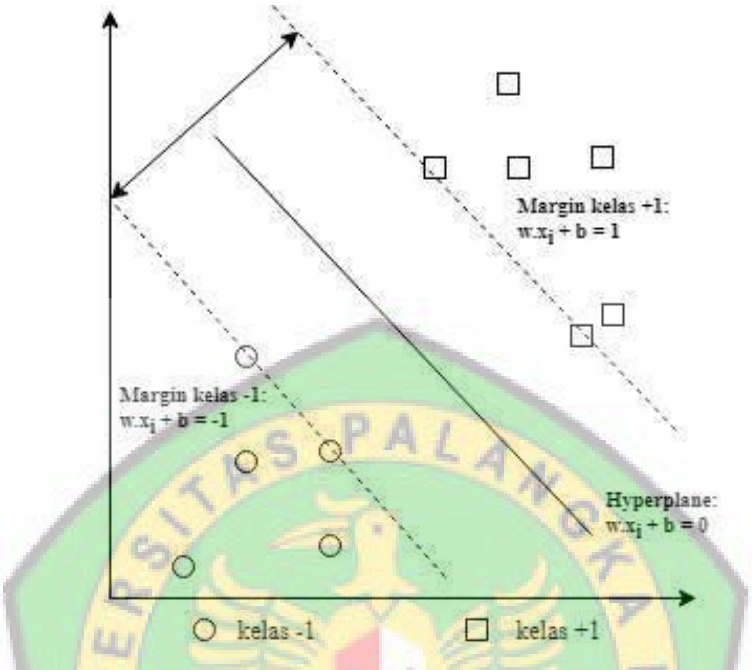


Gambar 2.4. Batas keputusan yang mungkin untuk set data

Konsep klasifikasi dengan SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas data pada *input space*. Gambar 2.4 memperlihatkan beberapa data yang merupakan anggota dari dua buah kelas data, yaitu $+1$ dan -1. Data yang tergabung pada kelas -1 disimbolkan dengan bentuk lingkaran, sedangkan data pada kelas $+1$, disimbolkan dengan bentuk kotak.

Hyperplane (batas keputusan) pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing kelas. Data yang terletak pada pada bidang pembatas

ini disebut sebagai *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pelatihan pada SVM (Prasetyo, 2014).



Gambar 2.5. Margin *Hyperplane*

2.6.2 SVM Linear

Setiap data latih dinyatakan oleh (x_i,y_i) dengan I = 1, 2, …, N, dan xi = {x_{i1}, x_{i2}, …, x_{in}}^{T} merupakan atribut (fitur) set untuk data latih ke-i. Untuk $y_i \in \{-1, +1\}$ menyatakan label kelas. Fungsi pemisah (*hyperplane*) yang digunakan dalam SVM seperti pada Gambar 2.5 adalah fungsi linear yang dapat didefinisikan sebagai berikut.

$$w.x_i + b = 0 \quad \text{..... (2.2)}$$

Parameter w dan b adalah parameter model yang dicari dalam proses pelatihan SVM. $\mathbf{w.x_i}$ merupakan *inner-product* antara w dan x_i, sehingga akan

ditemukan nilai *w* untuk setiap atribut x_i. Data x_i yang masuk ke dalam kelas -1 adalah data yang memenuhi pertidaksamaan berikut:

$$\boldsymbol{w}.\boldsymbol{x_i} + b \leq -1 \quad (2.3)$$

Sementara data x_i yang masuk kedalam kelas $+1$ adalah data yang memenuhi pertidaksamaan berikut:

$$\boldsymbol{w}.\boldsymbol{x_i} + b \geq +1 \quad (2.4)$$

Sesuai Gambar 2.5, jika ada data dalam kelas -1 (misal x_a) bertempat di *hyperplane* akan memenuhi persamaan (2.2). Untuk data kelas -1 dinotasikan:

$$\boldsymbol{w}.\boldsymbol{x_a} + b = 0 \quad (2.5)$$

Sementara kelas $+1$ (misal x_b) akan memenuhi persamaan:

$$\boldsymbol{w}.\boldsymbol{x_b} + b = 0 \quad (2.6)$$

Dengan memberikan label -1 untuk kelas pertama dan $+1$ untuk kelas kedua, maka untuk prediksi semua data uji menggunakan formula:

$$y = \begin{cases} -1, & \text{jika } w.x_i + b < 0 \\ +1, & \text{jika } w.x_i + b > 0 \end{cases} \quad (2.7)$$

Sesuai Gambar 2.5, *hyperplane* untuk kelas -1 (garis putus-putus) adalah data yang memenuhi persamaan:

$$\boldsymbol{w}.\boldsymbol{x_a} + b = -1 \quad (2.8)$$

Sementara *hyperplane* kelas $+1$ (garis putus-putus) adalah data yang memenuhi persamaan:

$$\boldsymbol{w}.\boldsymbol{x_b} + b = +1 \quad (2.9)$$

Dengan demikian maka margin dapat dihitung dengan mengurangkan persamaan (2.8) dan (2.9), didapatkan:

$$\boldsymbol{w}.(\boldsymbol{x_b} - \boldsymbol{x_a}) = 2 \quad (2.10)$$

Margin *hyperplane* diberikan oleh jarak antara dua *hyperplane* dari dua kelas tersebut. Notasi diatas dapat diringkas menjadi:

$$\|\boldsymbol{w}\| \times d = 2 \text{ atau } d = \frac{2}{\|\boldsymbol{w}\|} \quad \text{..... (2.11)}$$

2.6.3 *Hyperplane* SVM

Klasifikasi kelas data pada SVM pada persamaan (2.3) dan (2.4) dapat digabungkan dengan notasi:

$$y(\boldsymbol{w}.\boldsymbol{x_i} + b) \geq 1, i = 1,2, \ldots, \text{N} \quad \text{..... (2.12)}$$

Margin optimal dihitung dengan memaksimalkan jarak antara *hyperplane* dan data terdekat. Jarak ini dirumuskan dengan persamaan (2.11) ($\|\mathbf{w}\|$ adalah vektor bobot w). Selanjutnya masalah ini diformulasikan ke dalam problem *Quadratic Programming* (QP) dengan meminimalkan invers persamaan (2.11), $\frac{1}{2}\|\mathbf{w}\|^2$, di bawah konstrain (syarat), seperti berikut:

Minimalkan:

$$\frac{1}{2}\|\mathbf{w}\|^2 \quad \text{..... (2.13)}$$

Syarat:

$$y(\boldsymbol{w}.\boldsymbol{x_i} + b) \geq 1, i = 1,2, \ldots, \text{N} \quad \text{..... (2.14)}$$

Optimalisasi ini dapat diselesaikan dengan Lagrange multiplier:

$$Lp = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} a_i y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 \quad \text{..... (2.15)}$$

a_i adalah Lagrange multiplier yang berkorespondensi dengan x_i. Nilai a_i adalah nol atau positif.

Untuk meminimalkan Lagrangian, maka persamaan (2.15) harus diturunkan terhadap *w* dan *b* dan diset dengan nilai nol untuk syarat optimasi diatas:

Syarat 1:

$$\frac{\partial Lp}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} a_i y_i x_i \quad \text{..... (2.16)}$$

Syarat 2:

$$\frac{\partial Lp}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} a_i y_i = 0 \quad \text{..... (2.17)}$$

N adalah jumlah data yang menjadi *supporr vector*.

Karena Lagrange multiplier (α) tidak diketahui nilainya, maka persamaan di atas tidak dapat diselesaikan secara langsung untuk mendapatkan *w* dan *b*. Untuk menyelesaikan masalah tersebut, modifikasi persamaan (2.15) di atas menjadi kasus memaksimalkan dengan syarat optimal untuk dualitasnya menggunakan konstrain *Karush-Khun-Tucker* (KKT) sebagai berikut:

Syarat 1:

$$\alpha_i [y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1] = 0 \quad \text{..... (2.18)}$$

Syarat 2:

$$\alpha_i \geq 0, i = 1,2, \dots, N \quad \text{..... (2.19)}$$

Dengan menerapkan konstrain (2.18) dan (2.19) maka dipastikan bahwa nilai Lagrange multiplier sama banyaknya dengan data latih, meskipun sebenarnya banyak dari data latih yang Lagrange multiplier sama dengan nol (karena hanya beberapa saja yang akan menjadi *support vector*) ketika menerapkan syarat pertama. Konstrain di atas menyatakan bahwa Lagrange multiplier α_i harus nol kecuali data latih x_i yang memenuhi persamaan:

$$y(\boldsymbol{w}.\boldsymbol{x_i} + b) = 1 \quad \text{..... (2.20)}$$

Data latih tersebut, dengan $\alpha_i > 0$ terletak pada *hyperplane* bi_1 dan bi_2 dan disebut sebagai *support vector*. Data latih yang tidak terletak di *hyperplane* tersebut mempunyai $\alpha_i = 0$. Persamaan (2.16) dan (2.17) juga menyarankan parameter *w* dan *b* yang mendefinisikan *hyperplane* hanya tergantung pada *support vector*.

Masalah optimasi di atas masih sulit diselesaikan karena banyaknya parameter (w, b dan α_i). Untuk menyederhanakannya, persamaan optimasi (2.15) harus ditransformasikan ke dalam fungsi Lagrange multiplier itu sendiri (disebut dualitas masalah). Persamaan Lagrange multiplier (2.15) dapat dijabarkan menjadi:

$$Lp = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} a_i y_i (\boldsymbol{w}.\boldsymbol{x_i}) - b\sum_{i=1}^{N} a_i y_i + \sum_{i=1}^{N} a_i \quad \text{..... (2.21)}$$

Syarat optimasi (2.17) ada dalam suku ketiga di ruas kanan dalam persamaan (2.21), dan memaksa suku ini menjadi sama dengan 0. Dengan mengganti *w* dari syarat (2.16) dan suku $\|\mathbf{w}\|^2 = \boldsymbol{w_i}.\boldsymbol{w_j}$, maka persamaan di atas akan berubah menjadi dualitas Lagrange multiplier berupa *Ld* dan didapatkan:

Maksimalkan:

$$Ld = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N} a_i a_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} \quad \text{..... (2.22)}$$

$\boldsymbol{x_i}.\boldsymbol{x_j}$ merupakan *dot-product* dua data dalam data latih. Dengan memperhatikan persamaan (2.17) dan (2.19). Untuk mendapatkan nilai a_i langkah pertama adalah mengubah setiap dokumen teks kedalam nilai vektor (*support vector*) = $\left(\frac{x}{y}\right)$. Kemudian nilai dari setiap vektor dimasukkan ke persamaan (2.23) kernel *trick phi* berikut.

$$\varphi\begin{bmatrix}x\\y\end{bmatrix}=\begin{cases}\sqrt{x_n^2+y_n^2}>2\ maka\begin{bmatrix}\sqrt{x_n^2+y_n^2}-x+|x-y|\\\sqrt{x_n^2+y_n^2}-y+|x-y|\end{bmatrix}\\\sqrt{x_n^2+y_n^2}\leq 2\ maka\begin{bmatrix}x\\y\end{bmatrix}\end{cases} \quad \text{..... (2.23)}$$

Nilai x didapatkan dari persamaan (2.24) kernel linear untuk x berikut:

$$\sum_{i=1,j=1}^{n} x_i x_j^T, (i,j=1,2,\dots,n) \quad \text{..... (2.24)}$$

Nilai y didapatkan dari persamaan (2.25) kernel linear untuk y berikut:

$$\sum_{i=1,j=1}^{n} y_i y_j^T, (i,j=1,2,\dots,n) \quad \text{..... (2.25)}$$

Untuk memperoleh jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.23) diberi nilai bias = 1. Kemudian cari nilai paramater a_i, dengan terlebih dahulu mencari nilai fungsi setiap data menggunakan persamaan (2.26), lalu mencari nilai a_i pada persamaan lineat menggunakan persamaan (2.27) dengan memperhatikan $i,j=1,2,\dots,n$ berikut:

$$\sum_{i=1,j=1}^{n} a_i S_i^T S_j \quad \text{..... (2.26)}$$

$$\sum_{i=1,j=1}^{n} a_i S_i^T S_j = y_i \quad \text{..... (2.27)}$$

Setelah parameter a_i diperoleh, kemudian masukkan ke persamaan (2.28) berikut:

$$\widetilde{W}=\sum_{i=1}^{n} a_i S_i \quad \text{..... (2.28)}$$

Hasil yang diperoleh menggunakan persamaan (2.28) selanjutnya digunakan pada persamaan (2.29) untuk memperoleh nilai **w** dan **b** sebagai berikut:

$$y=w.x+b \quad \text{..... (2.29)}$$

Sedemikin sehingga diperoleh nilai **w** dan **b** atau nilai *hyperplane* untuk melakukan klasifikasi kedua kelas.

2.6.4 SVM Nonlinear

SVM sebenarnya adalah *hyperplane* linear yang hanya bekerja pada data yang dapat dipisahkan secara linear. Untuk data yang distribusi kelasnya tidak linear biasanya menggunakan pendekatan kernel pada fitur data awal set data. Kernel dapat didefinisikan sebagai suatu fungsi yang memetekan fitur data dari dimensi awal (rendah) ke fitur baru dengan dimensi yang relatih lebih tinggi (bahkan jauh lebih tinggi). Pendekatan ini berbeda dengan metode klasifikasi pada umumnya yang justru mengurangi dimensi awal untuk menyederhanakan proses komputasi dan memberikan akurasi prediksi yang lebih baik (Prasetyo, 2014).

Tabel 2.2 memperlihatkan beberapa pilihan fungsi kernel yang dapat digunakan dalam aplikasi:

Tabel 2.2. Fungsi Kernel

Nama Kernel	Definisi Fungsi
Linear	$K(x,y) = x.y$
Polynomial	$K(x,y) = (x.y+c)^d$
Gaussian RBF	$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{2.\sigma^2}\right)$

2.6.5 SVM pada *Information Retrieval*

Pada saat sekarang SVM telah banyak diterapkan pada bidang *information retrieval*. Sebagian besar penggunaan SVM pada *information retrieval* adalah pada pengklasifikasian dan pengkategorian dokumen, serta pada proses *searching* yang menerapkan *relevancy feedback*. Selain penggunaan pada dua hal tersebut, SVM juga digunakan untuk proses perengkingan pada *document retrieval*.

Kemampuan SVM yang cepat walaupun untuk data berdimensi tinggi kadang-kadang menjadi solusi yang tepat untuk keperluan *information retrival* yang membutuhkan kecepatan (Purnamawan, 2015).

Joachims (1998) mengenalkan SVM untuk *text categorization*. Di dalam tulisannya Joachims memberikan bukti secara teoritis dan secara eksperimen bahwa SVM sangat cocok untuk *text categorization*. Secara teoritis Joachims mengemukakan beberapa alasan mengapa SVM cocok digunakan untuk *text categorization*. Beberapa alasannya adalah sebagai berikut.

1. *High dimensional input space*: Pada *text categorization* akan didapati jumlah fitur yang sangat besar (lebih dari 10000), dan SVM cenderung tidak tergantung pada besarnya dimensi data.
2. *Few irrelevant features*: Karena sangat sedikit fitur-fitur yang tidak relevan, maka pemilihan fitur untuk tujuan mereduksi dimensi menjadi tidak efektif.
3. *Document vectors are sparse*: Vector-vector yang mewakili dokumen hanya memiliki sedikit bagian yang tidak bernilai 0. Punamawan (2015) dalam Kivinen, Warmuth, dan Auer (1995) memberikan bukti secara teoritis dan empiris, bahwa algoritma seperti SVM sangat cocok untuk menyelesaikan permasalahan seperti ini.
4. *Most text categorization problems are linearly separable*: Semua kategori dari data Ohsumed terpisah secara linear, begitu juga sebagian besar data Reuters. Ide dasar SVM adalah untuk medapatkan pemisah linear seperti itu.

2.7 Skenario Pengujian

2.7.1 Pengujian Fungsionalitas Sistem

Dilakukan setelah selesai tahap implementasi (pembuatan) program dengan menjalankan aplikasi/program dan dilihat apakah ada kesalahan atau tidak. *Testing* dilakukan dengan metode *Blackbox*. Tahap ini disebut juga sebagai tahap pengujian *alpha* (*alpha test*) dimana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri. Hal yang harus dipertimbangkan apakah *user* merasakan kemudahan serta manfaat dari aplikasi tersebut dan dapat menggunakannya sendiri (Rosa & Shalahuddin, 2013).

Dalam *testing* dan implementasi sistem dikenal 2 metode pengujian yang populer, yakni pengujian *blackbox* dan pengujian *white-box*. *Blackbox Testing* merupakan pengujian berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. Berikut ciri-ciri *Blackbox Testing* :

a. *Blackbox Testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*.
b. *Blackbox Testing* melakukan pengujian tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai *behavioral testing, spesification-based testing, input/output testing* atau *functional testing*.
c. Pada *Blackbox Testing* terdapat jeis desain tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, yang diantaranya :
    1) *Equivalence Class Partitioning*

2) *Boundary Value Analysis*

3) *State Transitions Testing*

4) *Cause-Effect Graphing*

d. Kategori kesalahan yang akan diketahui melalui *Blackbox testing :*

1) Fungsi yang hilang atau tak benar

2) Kesalahan dari antar-muka (*interface)*

3) Kesalahan dari struktur data atau akses eksternal database

4) Kesalahan dari kinerja atau tingkah laku (*behaviour)*

e. Kesalahan dari inisialisasi dan terminasi

2.7.2 Pengujian Performa Klasifikasi

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar. Akan tetapi, tidak dapat dipungkiri bahwa kinerja suatu sistem tidak bisa bekerja 100% benar. Oleh karena itu, sebuah sistem klasifikasi juga harus diukur kinerjanya. Umumnya cara mengukur kinerja klasifikasi menggunakan matriks confusion (Prasetyo, 2014).

Matriks konfusi merupakan tabel yang mencatat hasil kerja klasifikasi. Tabel 2.3 merupakan contoh matriks konfusi yang melakukan klasifikasi masalah biner (dua kelas) untuk dua kelas, misalnya kelas 0 dan 1. Setiap sel f_{ij} dalam matriks menyatakan jumlah *record*/data dari kelas *i* yang hasil prediksinya masuk ke kelas *j*. Misalnya f_{11} adalah jumlah data dalam kelas 1 yang secara benar dipetakan ke kelas 1, dan f_{10} adalah data dalam kelas 1 yang dipetakan secara salah ke kelas 0.

Tabel 2.3. Matriks konfusi untuk klasifikasi 2 kelas

f_{ij}		Kelas hasil prediksi (j)	
		Kelas = 1	Kelas = 0
Kelas asli(i)	Kelas = 1	f_{11}	f_{10}
	Kelas = 0	f_{01}	f_{00}

Berdasarkan isi matriks konfusi, maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu (f_{11} + f_{00}) dan data yang diklasifikasikan secara salah yaitu (f_{10} + f_{01}). Kuantitas matriks confusion dapat digunakan untuk mengetahui nilai akurasi.

Akurasi digunakan untuk mengetahui jumlah data yang diklasifikasikan secara benar. Untuk menghitung akurasi digunakan formula sebagai berikut:

$$Akurasi = \frac{Jumlah\ data\ yang\ dipredikasi\ secara\ benar}{Jumlah\ prediksi\ yang\ dilakukan}$$

$$= \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}} \quad \text{..... (2.30)}$$

2.8 *Software Tools*

2.8.1 HTML

HTML atau *Hyper Text Markup Language* merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman website yang dapat diakses atau ditampilkan menggunakan Web Browser. HTML sendiri secara resmi lahir pada tahun 1989 oleh Tim Berners Lee dan dikembangkan oleh *World Wide Web Consortium* (W3C), yang kemudian pada tahun 2004

dibentuklah *Web Hypertext Application Technology Group* (WHATG) yang hingga kini bertanggung jawab akan perkembangan bahasa HTML (Setiawan, 2017).

File HTML dapat dibuat dengan aplikasi *text editor* apapun di sistem operasi apapun, antara lain notepad di Windows, emasc atau vi di Unix atau SimpleText di Macintosh. File HTML ini juga bisa dibuat di aplikasi *word processor* apapun asalkan saat menyimpan file tersebut disimpan dengan format *text only*. Salah satu kelebihan dari file HTML adalah *cross platform*, artinya file HTML dapat ditampilkan di beberapa *Operating System* (OS) yang berbeda dan memiliki tampilan yang sama walaupun saat pembuatannya menggunakan satu OS tertentu saja.

2.8.2 PHP

PHP adalah bahasa pemrograman *script server side* yang di desain untuk pengembangan *web*. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan saat ini dikelola oleh The PHP Group. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *javascript* yang diproses pada *web browser* (*client*). PHP dapat digunakan dengan gratis dan bersifat open source, PHP dirilis dalam lisensi PHP license, berbeda dengan lisensi GNU *General Public License* (GPL) yang biasa digunakan untuk proyek *open source*.

Untuk membuat halaman *web*, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. PHP merupakan *website* yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan *website* statis, dimana konten dan halaman bersifat tetap. *Website* dinamis yang bias dibuat menggunakan PHP adalah situs *web* yang bisa menyesuaikan tampilan konten tergantung situasi (Setiawan, 2017).

2.8.3 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTPP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Ratna, 2014).

XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna computer di bidang pemrograman web XAMPP sebagai *server offline* yang berdiri sendiri atau localhost. XAMPP juga dilengkapi fitur manajemen *database* PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis *database* dengan mudah (Ratna, 2014).

2.8.4 MySQL

MySQL adalah sebuah *program database server* yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user* serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa digunakan adalah MySQL *FreeSoftware* yang berada dibawah Lisensi GNU/GPL (*General Public License*). MySQL merupakan sebuah *database server* yang *free*, artinya bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. Selain *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan sebagai *Client* maupun *server*. *Database* MySQL merupakan suatu perangkat lunak *database* yang berbentuk database relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language*) (Ratna, 2014).

BAB III

METODOLOGI PENELITIAN

Bab ini memaparkan proses pelaksanaan yang dilakukan dalam penelitian, mencakup penjelasan-penjelasan tentang metodologi pengumpulan data, metode pengembangan perangkat lunak, metodologi klasifikasi dan skenario pengujian yang dilakukan.

3.1 Metodologi Pengumpulan Data

Metodologi pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan untuk mencapai tujuan dalam melakukan penelitian. Pengumpulan data dalam penelitian ini menggunakan dua cara, yaitu: Studi Pustaka dan Studi Literatur Sejenis.

3.1.1 Studi Pustaka

Kepustakaan dilakukan dengan mempelajari teori-teori terkait dari hasil penelitian sebelumnya yang mendukung pemecahan masalah. Pengumpulan data dengan cara mengambil dari sumber-sumber media cetak maupun elektronik, jurnal, skripsi, *e-book* dan *browsing internet* yang dapat dijadikan acuan pemecahan masalah. Adapun data-data buku dan pencarian melalui media elektronik seperti *internet* yang digunakan dalam penelitian terdapat pada daftar pustaka.

3.1.2 Studi Literatur Sejenis

Studi literatur sejenis dilakukan dengan mempelajari penelitian sebelumnya yang memiliki keterkaitan judul dan bahasan dengan penelitian yang dilakukan.

3.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak dalam pembuatan sistem klasifikasi dokumen berita *online* berbahasa Indonesia adalah metode *Waterfall* menurut Pressman (Wahyudi, 2017). Adapun tahapan dalam pengembangan metode *waterfall* yang dilakukan adalah sebagai berikut.

1. Analisis

   Tahapan ini merupakan tahap dalam mencari informasi sebanyak-banyaknya mengenai sistem yang diteliti dengan melakukan metode-metode pengumpulan data sehingga ditemukan kebutuhan pengguna dalam sistem. Tahap ini juga dilakukan untuk mencari pemecahan masalah dan menganalisis bagaimana sistem akan dibangun.

2. Desain

   Tahap ini merupakan tahapan perancangan sistem yang dilakukan pemodelan sistem dengan menggunakan *Unified Modeling Language.*

3. Pengkodean

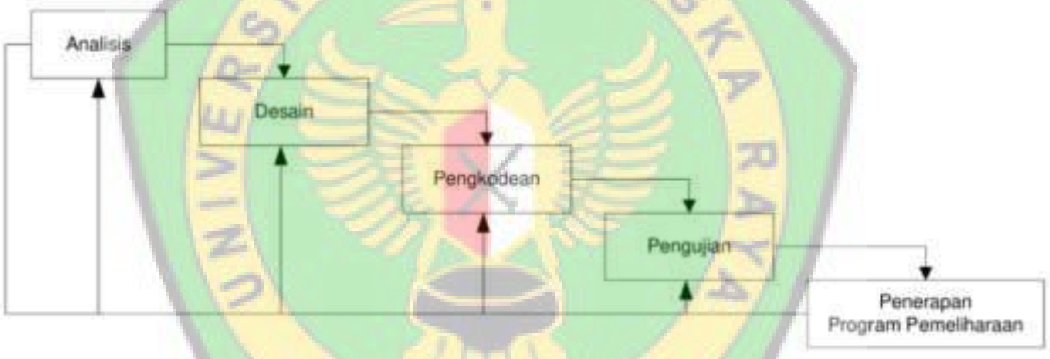
   Tahap ini merupakan tahapan dalam pengimplementasian sistem yang sudah dirancang dan dilakukan penulisan program dengan menggunakan bahasa pemrograman yang dibutuhkan.

4. Pengujian

   Tahap ini merupakan tahap pengujian sistem secara keseluruhan. Sistem yang dibangun akan diuji dengan menggunakan teknik pengujian *Blackbox.*

5. Penerapan Program Pemeliharaan

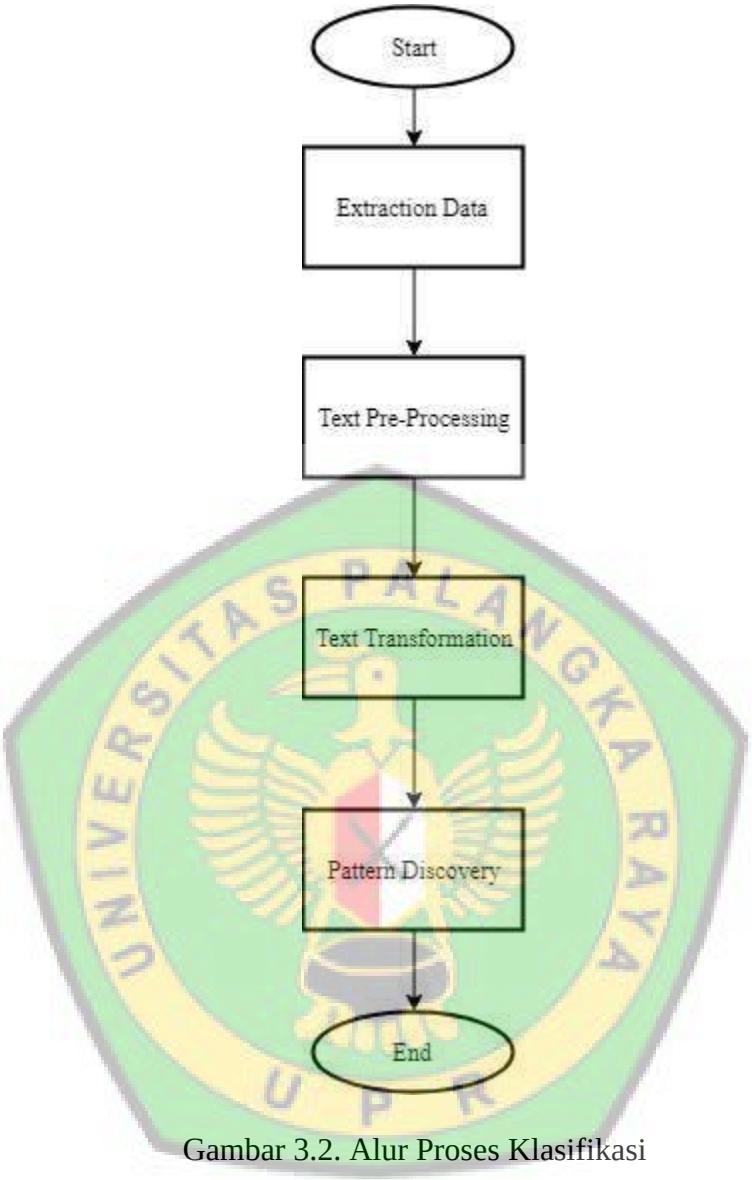
   Tahap ini merupakan tahapan penerapan penggunaan sistem oleh pengguna serta pemeliharaan sistem. Namun pada tahapan ini tidak dilakukan pemeliharaan, hanya sampai pada tahapan *testing*.



Gambar 3.1. Metode *Waterfall* (Pressman, 2010)

3.3 Metode Klasifikasi

Metode klasifikasi yang dilakukan terdapat 4 proses yaitu, *Extraction Data*, *Text Pre-Processing*, *Text Transformation* dan *Pattern Discovery* seperti yang terdapat pada Gambar 3.2.

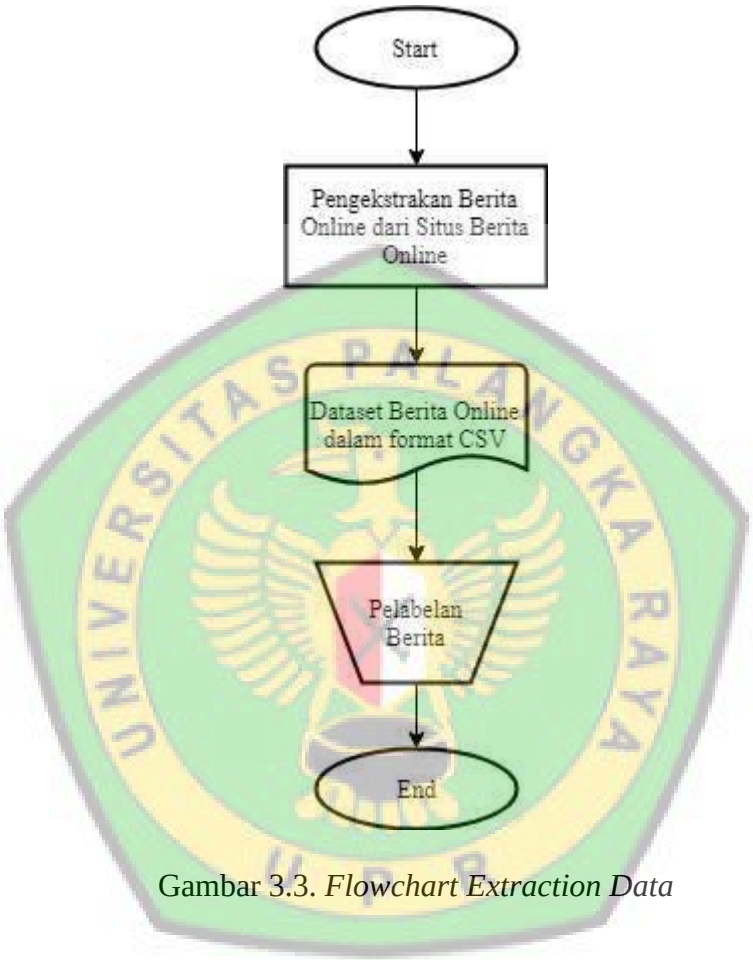


Gambar 3.2. Alur Proses Klasifikasi

3.3.1 *Extraction Data*

Extraction data yang dilakukan meliputi pengumpulan dataset berita *online* berbahasa Indonesia yang diperoleh melalui situs atau *website* berita *online* berbahasa Indonesia. Untuk memperoleh data latih (*training set*) yang tepat dan untuk mempermudah pengujian kebenaran serta keakuratan pada data uji (*testing set*), maka pemberian label kategori pada sertiap artikel berita *online* yang

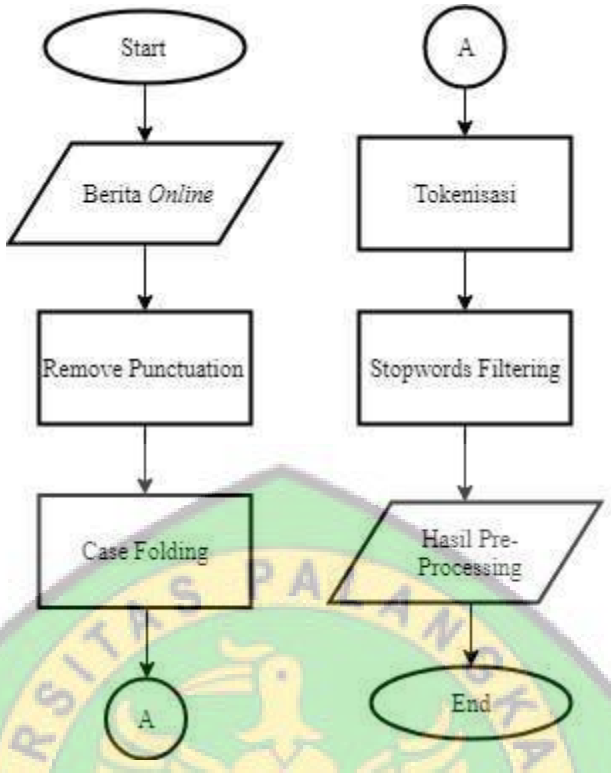
digunakan dalam penelitian telah dikelompokkan atau dikategorikan oleh situs berita *online* yang menjadi objek penelitian.



Gambar 3.3. *Flowchart Extraction Data*

3.3.2 *Text Pre-Processing*

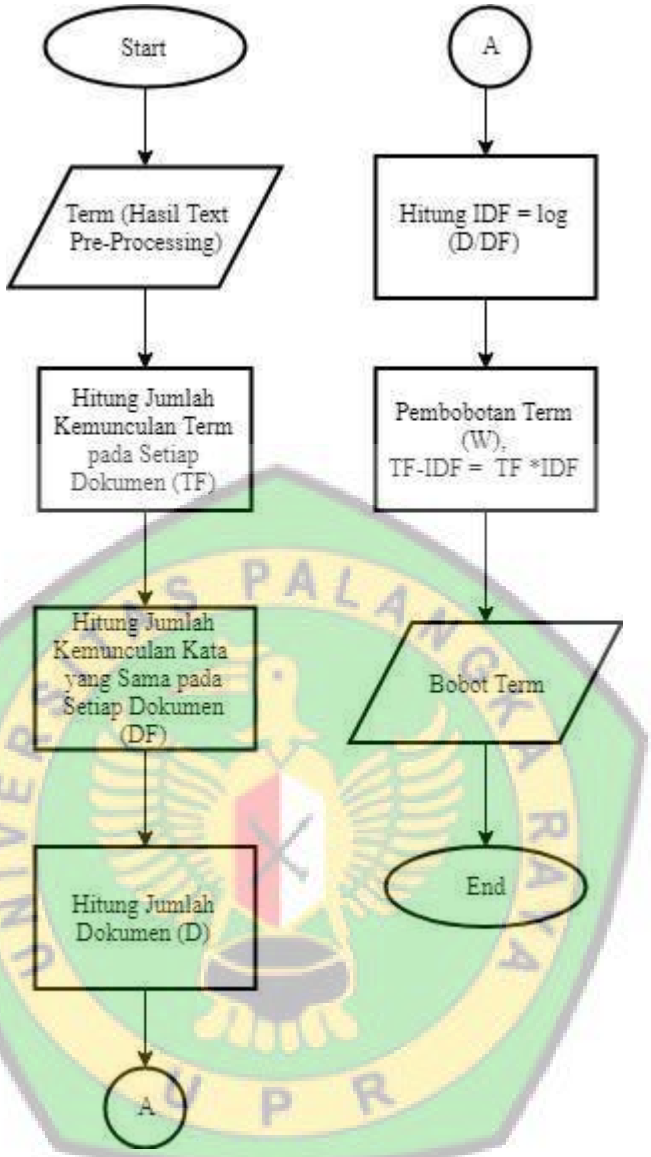
Proses ini merupakan proses awal untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Pada proses ini diterapkan *Text Vectorization* dengan metode *CountVectorizer* yang prosesnya meliputi menghilangkan tanda baca pada teks (*remove punctuation), case folding*, tokenisasi dan *stopwords filtering*.



Gambar 3.4. *Flowchart Text Pre-Processing*

3.3.3 *Text Transformation*

Pada tahap ini dilakukan pembentukan atribut yang mengacu pada proses untuk mendapatkan representasi dokumen yang diharapkan. Pendekatan yang diterapkan adalah *Feature Extraction* dengan menggunakan TF-IDF yang bertujuan untuk mendapatkan informasi dari setiap token yang ada dalam koleksi kata dari proses sebelumnya berdasarkan data bobot kemunculan kata pada seluruh dokumen.



Gambar 3.5. *Flowchart* TF-IDF

3.3.4 *Pattern Discovery*

Tahap ini merupakan tahap akhir pada proses klasifikasi yang menghasilkan model klasifikasi berdasarkan dataset yang telah di-*training* dan diuji menggunakan algoritma *Support Vector Machine*. Model ini digunakan untuk melakukan klasifikasi kategori data baru yang belum memiliki label.

3.4 Skenario Pengujian

Skenario pengujian yang dilakukan dalam penelitian ini meliputi skenario pengujian fungsionalitas sistem menggunakan metode *Blackbox Testing* dan skenario pengujian performa klasifikasi menggunakan metode *Accuracy*.

3.4.1 Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem yang dilakukan bertujuan untuk pengujian fungsionalitas sistem pada perangkat lunak yang dibuat. Metode pengujian sistem yang digunakan adalah *Blackbox Testing* (Rosa dan Shalahuddin, 2013), dimana pada metode ini hanya memfokuskan kepada fungsionalitas sistem yang dibuat.

3.4.2 Pengujian Performa Klasifikasi

Pengujian performa klasifikasi digunakan untuk mengukur kinerja terhadap sistem klasifikator dalam mengelompokkan data ke dalam label kelas yang tepat. Metode pengujian pada sistem klasifikator yang digunakan adalah *Accuracy* (Prasetyo, 2014). Metode ini digunakan untuk mengetahui tingkat keakuratan klasifikasi menggunakan algoritma *Support Vector Machine.* Untuk mengetahui pengaruh jumlah data latih terhadap efektifitas SVM *classifier* maka akan dilakukan beberapa kombinasi jumlah dokumen latih dan dokumen uji. Adapun mekanisme pengujian yang dilakukan terhadap persentase data latih dan data uji dapat dilihat pada Tabel 3.1.

Tabel 3.1. Mekanisme Pengujian

Data	
Data Pelatihan	Data Pengujian
50%	50%
60%	40%
70%	30%
80%	20%

3.5 Analisis

3.5.1 Deskripsi Sistem

Sistem pada penelitian ini merupakan sebuah sistem klasifikasi menggunakan metode *Support Vector Machine* (SVM). Sistem ini akan mengklasifikasikan berita *online* kedalam satu dari lima kelas, yaitu Lifestyle, Olahraga, Politik, Ekonomi dan Teknologi. Sistem yang dirancang dapat dijalankan melalui *website,* yang dapat digunakan untuk melakukan klasifikasi data berita *online* serta pengelolaan sistem klasifikasi.

3.5.2 Data Penelitian

Data yang digunakan pada penelitian ini berupa teks berita *online* yang terdiri dari lima macam label kelas, yaitu Lifestyle, Olahraga, Politik, Ekonomi dan Teknologi. Data teks berita *online* dikumpulkan dengan *range* waktu mulai Tahun 2016 – 2020, yang diperoleh dari situs berita *online* berbahasa Indonesia, yaitu situs *detik.com*, *kompas.com* dan *okezone.com*. Proses pengumpulan

dilakukan secara manual dan *scraping*, mulai dari masuk ke halaman *website*, mengekstrak data dari konten, dan menyimpan data ke satu file Dataset. Dataset ini digunakan untuk membangun sistem klasifikasi berita *online* menggunakan algoritma *Support Vector Machine* (SVM) berdasarkan topik berita. Hasil dari pengumpulan berita *online* tersebut kemudian dipilih sebanyak 1000 *record* data, dengan pembagian 200 *record* untuk setiap kategori kelas. Dataset disimpan dalam format *Comma-Separated Values* (CSV), terdiri dari judul dan isi berita. Tabel 3.2 menunjukkan komposisi dataset yang digunakan dalam penelitian.

Tabel 3.2. Komposisi Dataset

Situs	Jumlah Data	*Range* Waktu (Tahun)	Cara Pengumpulan
detik.com	324	2016 – 2020	manual
kompas.com	354	2016 – 2020	*scraping*
okezone.com	322	2016 – 2020	manual

3.5.3 Analisis Pengguna

Dalam sistem yang akan dibangun ada 2 entitas yang terlibat sebagai pengguna dari *website* sistem klasifikasi, yaitu Editor dan Admin. Analisis masing-masing pengguna dapat dilihat pada Tabel 3.3.

Tabel 3.3. Analisis Pengguna

Pengguna	Hak Akses
Admin	a. Kelola Data Profil b. Kelola Data Berita

Tabel 3.3. Analisis Pengguna (Lanjutan)

Pengguna	Hak Akses
	c. Kelola Data Stopwords d. Klasifikasi SVM
Editor	a. Klasifikasi Berita b. Tentang

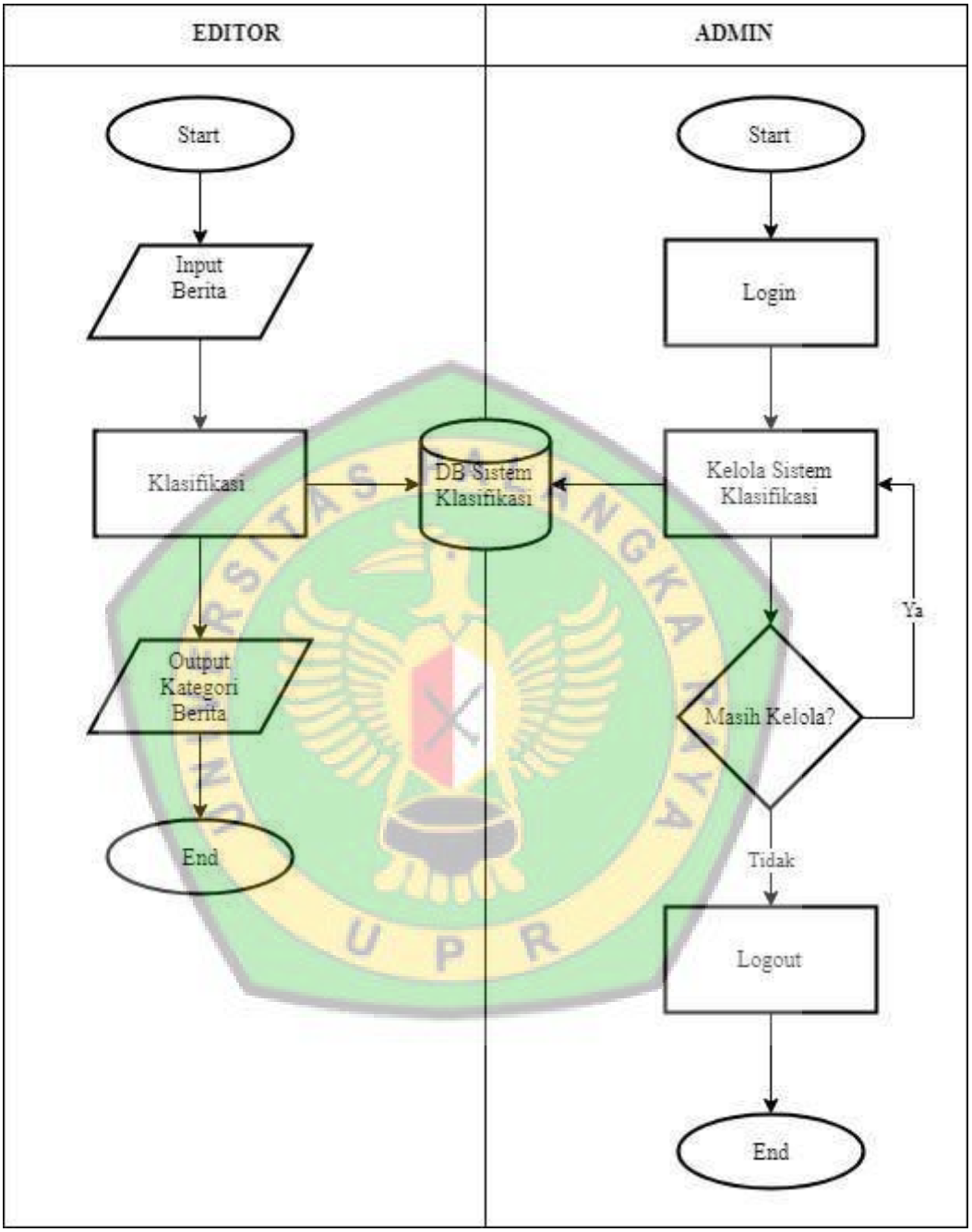
3.5.4 Proses Bisnis

Proses bisnis mengambarkan bagaimana proses yang terjadi pada *website* sistem klasifikasi, seperti mengelola data, serta fitur-fitur yang terdapat pada sistem. Proses bisnis ini juga menggambarkan bagaimana peranan admin dan editor sebagai pengguna dari *website* sistem klasifikasi. Tabel 3.4 memperlihatkan *flowchart* proses bisnis sistem klasifikasi yang dibuat.

Editor dan Admin memiliki hak akses yang berbeda. Pengguna dengan level admin memiliki hak akses yang lebih besar dibandingkan Editor. Berikut merupakan penjelasan dari hak akses masing-masing pengguna.

1. Admin, merupakan seorang yang bertugas untuk mengelola data pada *website* sistem klasifikasi. Admin memilik hak akses penuh untuk mengelola segala macam data yang ada pada dan fitur yang ada.
2. Editor, merupakan orang yang mengakses *website* sistem klasifikasi berita *online* untuk melakukan klasifikasi berita *online* melalui teks berita dengan harapan dapat memperoleh sebuah *output* mengenai kategori konten berita yang diinputkan.

Tabel 3.4. Proses Bisnis Sistem Klasifikasi



3.5.5 Fungsionalitas

Adapun fungsionalitas yang terdapat pada sistem klasifikasi berita *online* pada masing-masing pengguna adalah sebagai berikut.

1. Admin

a. *Login*

Seorang admin sebelum mulai melakukan kelola terhadap data di dalam *website* sistem klasifikasi harus login terlebih dahulu. Pada halaman *login* ini admin akan diminta untuk login terlebih dahulu dengan menggunakan *username* dan *password*.

b. Beranda Admin

Beranda admin merupakan halaman awal yang akan ditampilkan saat admin berhasil *login*.

c. Kelola Profil

Fitur ini digunakan untuk melakukan kelola terhadap akun, meliputi mengubah nama, *username* dan *password*.

d. Kelola Berita

Kelola berita adalah fitur yang digunakan untuk melakukan kelola terhadap data berita yang telah diklasifikasikan.

e. Kelola Stopwords

Kelola stopwords adalah fitur yang digunakan untuk melakukan kelola terhadap data stopwords.

f. Klasifikasi SVM

Klasifikasi SVM merupakan fitur yang digunakan untuk melakukan proses pelatihan terhadap data latih berita *online* menggunakan algoritma *Support Vector Machine* dalam membangun

model klasifikasi, kemudian model ini akan diuji dengan set data uji untuk mengetahui tingkat keakuratan model klasifikasi yang dibangun.

g. *Logout*

Fitur *logout* digunakan untuk mengakhiri aktivitas yang dilakukan admin di dalam sistem.

2. Editor

a. Beranda Editor

Beranda editor merupakan halaman awal yang akan ditampilkan saat editor membuka *website* sistem klasifikasi.

b. Klasifikasi

Fitur klasifikasi pada editor menyediakan fasilitas untuk melakukan klasifikasi konten berita terhadap berita *online* yang belum diketahui kategorinya dengan menginputkan teks berita *online* yang akan diklasifikasi, kemudian sistem akan memberikan *output* berupa kategori konten berita *online* tersebut.

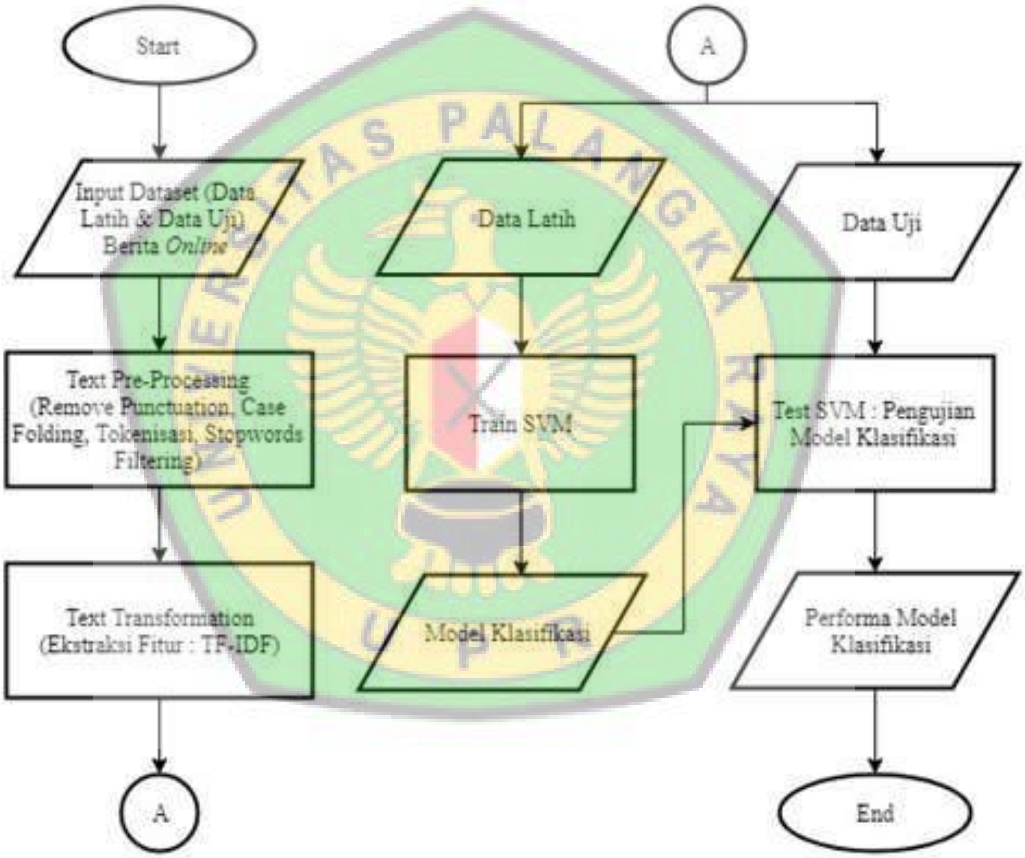
c. Tentang

Menampilkan informasi mengenai sistem klasifikasi berita *online* yang dibangun.

3.5.6 Sistem Klasifikasi

Gambar 3.6 menjelaskan proses dari sistem klasifikasi yang digunakan pada penelitian. Garis besar sistem klasifikasi pada penelitian ini adalah setelah dilakukan ekstraksi fitur yang meliputi *text pre-processing* dan *text*

transformation. Data latih akan digunakan untuk melakukan pelatihan pada algoritma *Support Vector Machine* (SVM). Data uji digunakan untuk menguji sistem klasifikasi yang dibangun. Model yang diperoleh dari pelatihan akan digunakan untuk melakukan klasifikasi terhadap artikel berita *online* baru yang belum diketahui label kelasnya.



Gambar 3.6. *Flowchart* Sistem Klasifikasi

3.5.6.1 *Text Pre-Processing*

Mula-mula dataset berita *online* yang telah diberi label akan dilakukan *text pre-processing* terlebih dahulu untuk menghilangkan data tidak penting yang

dikategorikan sebagai *noise*. Prosesnya dimulai dengan mengubah dokumen teks kedalam representasi vektor numerik, sebab algoritma *machine learning* seperti SVM beroperasi pada ruang fitur numerik. Sehingga untuk melakukan pembelajaran/pelatihan pada teks, dokumen teks perlu diubah terlebih dahulu menjadi representasi vektor, proses ini disebut sebagai *Text Vectorization*. Prosesnya meliputi tokenisasi yang didalamnya dapat diterapkan *pre-process* untuk mengurangi dimensi fitur. Berikut contoh implementasi *text pre-processing* yang dilakukan pada penelitian dengan contoh dataset berita *online* sebagai berikut.

Tabel 3.5. Dataset Berita *Online*

Dokumen	Text	Label
1	DPR: Wacana Presiden Dipilih DPR Hanya Melanggengkan Oligarki Politik.	Politik
2	Sohibul Iman Sebut Safari Politik PKS ke Partai Demokrat pada Desember.	Politik
3	Nurul Qomar Sebut Kasusnya Sarat Kepentingan Politik.	Politik
4	AS-China Tegang, Harga Emas Merangkak Naik.	Ekonomi
5	Ini Alasan Kementerian ESDM yang Belum Juga Turunkan Harga BBM.	Ekonomi
6	Kasihan Peternak, Harga Telur Ayam di Kandang Sempat Turun.	Ekonomi

1. *Remove punctuation*, digunakan untuk menghapus tanda baca dan menghapus simbol-simbol pada berita *online*.

   Hasil setelah *remove punctuation*:

Tabel 3.6. Proses *Remove Punctuation*

D1	DPR Wacana Presiden Dipilih DPR Hanya Melanggengkan Oligarki Politik
D2	Sohibul Iman Sebut Safari Politik PKS ke Partai Demokrat pada Desember
D3	Nurul Qomar Sebut Kasusnya Sarat Kepentingan Politik
D4	AS China Tegang Harga Emas Merangkak Naik
D5	Ini Alasan Kementerian ESDM yang Belum Juga Turunkan Harga BBM
D6	Kasihan Peternak Harga Telur Ayam di Kandang Sempat Turun

2. *Case folding*, digunakan untuk mengubah huruf pada berita *online* menjadi huruf kecil semua.

   Hasil setelah *case folding*:

Tabel 3.7. Proses *Case Folding*

D1	dpr wacana presiden dipilih dpr hanya melanggengkan oligarki politik
D2	sohibul iman sebut safari politik pks ke partai demokrat pada desember

Tabel 3.7. Proses *Case Folding* (Lanjutan)

D3	nurul qomar sebut kasusnya sarat kepentingan politik
D4	as china tegang harga emas merangkak naik
D5	ini alasan kementerian esdm yang belum juga turunkan harga bbm
D6	kasihan peternak harga telur ayam di kandang sempat turun

3. Tokenisasi, digunakan untuk memilah string berupa kalimat menjadi token atau kata berdasarkan karakter spasi yang ditemukan.

   Hasil setelah tokenisasi:

Tabel 3.8. Proses Tokenisasi

D1	dpr	wacana	presiden	dipilih	dpr
	hanya	melanggengkan	oligarki	politik	
D2	sohibul	iman	sebut	safari	politik
	pks	ke	partai	demokrat	pada
	desember				
D3	nurul	qomar	sebut	kasusnya	sarat
	kepentingan	politik			
D4	as	china	tegang	harga	emas
	merangkak	naik			
D5	ini	alasan	kementerian	esdm	yang
	belum	juga	turunkan	harga	bbm
D6	kasihan	peternak	harga	telur	ayam
	di	kandang	sempat	turun	

4. *Stopwords Filtering,* digunakan untuk menghilangkan kata-kata yang terlalu sering muncul diantara file teks yang tidak memiliki arti khusus.

Pada penelitian ini, *stopword* adalah daftar kata yang digunakan untuk menyaring fitur atau atribut kata yang tidak layak dimasukkan ke dalam kamus kata dalam merepresentasikan sebuah kelas. Daftar kata *stopword* yang digunakan pada penelitian ini bersumber dari situs *https://github.com/masdevid/ID-Stopwords (*Haryalesmasna, 2016), dengan jumlah awal (original) kata sebanyak 757 kata *stopword*. Daftar kata *stopword* awal (original) yang digunakan pada penelitian ini disesuaikan kembali terhadap studi kasus untuk mendapatkan atribut kata yang lebih optimal. Proses pemilihan daftar kata *stopword* dilakukan dengan observasi terhadap jumlah kemunculan sebuah fitur atau atribut kata di setiap kelasnya pada dataset berita *online* yang telah diperoleh, kemudian dibandingkan atribut kata mana yang paling sering muncul pada kelas-kelas lainnya. Dari hasil perbandingan tersebut, barulah ditetapkan fitur atau atribut kata mana yang dipertimbangkan untuk ditambahkan atau dihapus dari daftar awal kata *stopword.*

Dari hasil pengamatan yang dilakukan, diperoleh bahwa terdapat 147 kata yang ditambahkan kedalam *stopword* awal, 147 kata tersebut ditambahkan karena memiliki jumlah kemunculan yang cukup besar pada setiap dokumen di seluruh kelas. Kemudian dari daftar kata awal *stopword*, tidak terdapat kata yang dihilangkan karena daftar kata awal *stopword* yang disediakan merupakan kata umum yang memang perlu dimuat dalam

stopword list, sehingga diperoleh atribut kata yang mampu merepresentasikan sebuah kelas dengan baik. Adapun daftar kata awal (original) *stopword*, maupun daftar kata *stopword* baru yang ditambahkan sesuai dengan studi kasus pada penelitian terdapat pada bagian lampiran.

Hasil setelah *stopwords filtering*:

Tabel 3.9. Proses *Stopwords Filtering*

D1	dpr	wacana	presiden	dipilih	dpr
		melanggengkan	oligarki	politik	
D2	sohibul	iman		safari	politik
	pks		partai	demokrat	
	desember				
D3	nurul	qomar		kasusnya	sarat
	kepentingan	politik			
D4	as	china	tegang	harga	emas
	merangkak				
D5		alasan	kementerian	esdm	
			turunkan	harga	bbm
D6	kasihan	peternak	harga	telur	ayam
		kandang		turun	

3.5.6.2 *Text Transformation*

Secara keseluruhan, tahap *text transformation* dilakukan guna membentuk atribut yang mengacu pada proses untuk mendapatkan representasi dokumen yang

diharapkan seperti *Bag of Words*. Pendekatan yang diterapkan adalah *Feature Extraction* untuk mengubah dokumen kedalam bentuk yang lebih representatif, salah satunya vektor. Metode yang digunakan adalah TF-IDF yang bertujuan untuk memberikan bobot terhadap kata atau *term*.

Pembobotan diperlukan untuk mengatasi permasalahan seperti kemunculan kata yang terlalu umum karena membawa sangat sedikit informasi yang bermakna tentang konten pada dokumen teks yang sebenarnya. Jika kata seperti itu digunakan pada sistem klasifikasi maka *term* atau istilah yang sering muncul akan menutupi frekuensi istilah atau kata yang lebih jarang namun lebih menarik. Untuk menimbang ulang fitur hitungan kedalam nilai *floating* atau *real point* yang cocok digunakan pada sistem klasifikasi, digunakan metode transformasi TF-IDF. Kata atau *term* dihitung probabilitas kemunculannya dalam satu dokumen (D1 sampai D6) untuk memperoleh TF (*Term Frequency*) IDF seperti formula (2.1) dan TF-IDF. Adapun contoh perhitungan TF-IDF pada dokumen D1, D2, D3, D4, D5 dan D6 dapat dilihat pada Tabel 3.10. Pada Tabel 3.10 terdapat proses TF (*Term Frequency*) yang merupakan jumlah kemunculan atribut kata atau *term* didalam suatu dokumen teks. Kolom DF (*Document Frequency*) merupakan jumlah dokumen yang memiliki atribut kata atau *term.* IDF menunjukkan nilai kemunculan sebuah *term* dalam seluruh dokumen.

Tabel 3.10. Proses TF-IDF

Term	TF						DF	IDF	Bobot = TF*IDF					
	D1	D2	D3	D4	D5	D6			D1	D2	D3	D4	D5	D6
dpr	2	0	0	0	0	0	1	0.778	1.556	0	0	0	0	0
wacana	1	0	0	0	0	0	1	0.778	0.778	0	0	0	0	0
presiden	1	0	0	0	0	0	1	0.778	0.778	0	0	0	0	0
dipilih	1	0	0	0	0	0	1	0.778	0.778	0	0	0	0	0
melanggengkan	1	0	0	0	0	0	1	0.778	0.778	0	0	0	0	0
oligarki	1	0	0	0	0	0	1	0.778	0.778	0	0	0	0	0
politik	1	1	1	0	0	0	3	0.3010	0.3010	0.3010	0.3010	0	0	0
sohibul	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0
iman	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0
safari	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0
pks	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0

Tabel 3.10. Proses TF-IDF (Lanjutan)

Term	TF						DF	IDF	Bobot = TF*IDF					
	D1	D2	D3	D4	D5	D6			D1	D2	D3	D4	D5	D6
partai	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0
demokrat	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0
desember	0	1	0	0	0	0	1	0.778	0	0.778	0	0	0	0
nurul	0	0	1	0	0	0	1	0.778	0	0	0.778	0	0	0
qomar	0	0	1	0	0	0	1	0.778	0	0	0.778	0	0	0
kasusnya	0	0	1	0	0	0	1	0.778	0	0	0.778	0	0	0
sarat	0	0	1	0	0	0	1	0.778	0	0	0.778	0	0	0
kepentingan	0	0	1	0	0	0	1	0.778	0	0	0.778	0	0	0
as	0	0	0	1	0	0	1	0.778	0	0	0	0.778	0	0
china	0	0	0	1	0	0	1	0.778	0	0	0	0.778	0	0
tegang	0	0	0	1	0	0	1	0.778	0	0	0	0.778	0	0

Tabel 3.10. Proses TF-IDF (Lanjutan)

Term	TF						DF	IDF	Bobot = TF*IDF					
	D1	D2	D3	D4	D5	D6			D1	D2	D3	D4	D5	D6
harga	0	0	0	1	1	1	3	0.3010	0	0	0	0.3010	0.3010	0.3010
emas	0	0	0	1	0	0	1	0.778	0	0	0	0.778	0	0
merangkak	0	0	0	1	0	0	1	0.778	0	0	0	0.778	0	0
alasan	0	0	0	0	1	0	1	0.778	0	0	0	0	0.778	0
kementerian	0	0	0	0	1	0	1	0.778	0	0	0	0	0.778	0
esdm	0	0	0	0	1	0	1	0.778	0	0	0	0	0.778	0
turunkan	0	0	0	0	1	0	1	0.778	0	0	0	0	0.778	0
bbm	0	0	0	0	1	0	1	0.778	0	0	0	0	0.778	0
kasihan	0	0	0	0	0	1	1	0.778	0	0	0	0	0	0.778
peternak	0	0	0	0	0	1	1	0.778	0	0	0	0	0	0.778
telur	0	0	0	0	0	1	1	0.778	0	0	0	0	0	0.778

Tabel 3.10. Proses TF-IDF (Lanjutan)

Term	TF						DF	IDF	Bobot = TF*IDF					
	D1	D2	D3	D4	D5	D6			D1	D2	D3	D4	D5	D6
ayam	0	0	0	0	0	1	1	0.778	0	0	0	0	0	0.778
kandang	0	0	0	0	0	1	1	0.778	0	0	0	0	0	0.778
turun	0	0	0	0	0	1	1	0.778	0	0	0	0	0	0.778

3.5.6.3 *Train* SVM

Pada penelitian ini, setelah vektor-vektor fitur terbentuk menggunakan proses *Feature Extraction*, maka vektor-vektor tersebut telah siap dimasukkan ke dalam algoritma SVM untuk dijadikan data *training*. Hasil keluaran SVM dari proses *training* adalah nilai *alpha* untuk setiap vektor (data latih) dan sebuah nilai b (*bias*). Pada dasarnya SVM membagi ruang vektor kedalam 2 bagian yaitu kelas positif dan negatif dengan menggunakan *hyperplane* sebagai garis atau bidang pemisah antara dua kelas. Dalam hal ini fungsi pemisah yang digunakan adalah fungsi linear sebagaimana persamaan (2.2). Berikut merupakan perhitungan dari algoritma SVM.

Tabel 3.11 merupakan format data vektor dengan menggunakan hasil ekstraksi fitur yang telah dilakukan pada tahapan sebelumnya untuk D1, D2, D3, D4, D5, D6. Pada Tabel 3.11, kolom $x_i = \{x_1, x_2, \ldots, x_n\}$ adalah fitur atau atribut kata pada data latih ke-*i* yang telah diperoleh dari tahapan sebelumnya, untuk y_i = {-1 dan +1}, dimana label -1 adalah label kelas Ekonomi dan +1 adalah label kelas Politik. Tahap ini akan terus berulang hingga semua *term* pada berita terwakili oleh format data vektor. Untuk *term* yang sama muncul lebih dari sekali dalam sebuah berita akan diwakili sebuah data vektor saja dengan nilai bobot yang bersesuaian.

Tabel 3.11. Vektor Fitur Dataset Berita *Online*

x_i	D1	D2	D3	D4	D5	D6
x_1	1.556	0	0	0	0	0
x_2	0.778	0	0	0	0	0
x_3	0.778	0	0	0	0	0
x_4	0.778	0	0	0	0	0
x_5	0.778	0	0	0	0	0
x_6	0.778	0	0	0	0	0
x_7	0.3010	0.3010	0.3010	0	0	0
x_8	0	0.778	0	0	0	0
x_9	0	0.778	0	0	0	0
x_{10}	0	0.778	0	0	0	0
x_{11}	0	0.778	0	0	0	0
x_{12}	0	0.778	0	0	0	0
x_{13}	0	0.778	0	0	0	0

Tabel 3.11. Vektor Fitur Dataset Berita *Online* (Lanjutan)

x_i	D1	D2	D3	D4	D5	D6
x_{14}	0	0.778	0	0	0	0
x_{15}	0	0	0.778	0	0	0
x_{16}	0	0	0.778	0	0	0
x_{17}	0	0	0.778	0	0	0
x_{18}	0	0	0.778	0	0	0
x_{19}	0	0	0.778	0	0	0
x_{20}	0	0	0	0.778	0	0
x_{21}	0	0	0	0.778	0	0
x_{22}	0	0	0	0.778	0	0
x_{23}	0	0	0	0.3010	0.3010	0.3010
x_{24}	0	0	0	0.778	0	0
x_{25}	0	0	0	0.778	0	0
x_{26}	0	0	0	0	0.778	0

Tabel 3.11. Vektor Fitur Dataset Berita *Online* (Lanjutan)

x_i	D1	D2	D3	D4	D5	D6
x_{27}	0	0	0	0	0.778	0
x_{28}	0	0	0	0	0.778	0
x_{29}	0	0	0	0	0.778	0
x_{30}	0	0	0	0	0.778	0
x_{31}	0	0	0	0	0	0.778
x_{32}	0	0	0	0	0	0.778
x_{33}	0	0	0	0	0	0.778
x_{34}	0	0	0	0	0	0.778
x_{35}	0	0	0	0	0	0.778
x_{36}	0	0	0	0	0	0.778
Y	1	1	1	-1	-1	-1

Pada Tabel 3.11, terdapat data masukan yang diubah menjadi data vektor. Sebagai contoh, format data input untuk klasifikasi SVM pada penelitian ini adalah 1 1:1.556 2:0.778 3:0.778. Masukan pertama +1 atau -1 menyatakan dua label kelas awalan yang diberikan. Angka 1 sebelum tanda ":" menyatakan indeks kata dan angka 1.556 setelah tanda ":" menyatakan bobot tf-idf dari kata tersebut.

Selanjutnya harus dilakukan kernelisasi pada set data dari fitur dimensi lama sehingga mendapatkan set data dengan fitur baru berdimensi tinggi. Dengan Kernel Linear, $K(x_i x_j) = x_i x_j^T$. Dengan set data berdimensi N×2 maka akan didapatkan dimensi baru sebesar N×N, dimana N adalah banyaknya data. Dengan menggunakan informasi pada Tabel 3.11, maka untuk K(1,1), K(1,2), …,K(6,6) dihitung dari *dot-product* pada semua data. Contoh hasil perhitungan *dot-product* pada K(1,1), K(1,2), K(1,3), K(1,4), K(1,5), K(1,6) sebagai data yang pertama dapat dilihat pada Tabel 3.12.

Tabel 3.12. Proses Kernelisasi

K(1,1)	K(1,2)	K(1,3)	K(1,4)	K(1,5)	K(1,6)
2.421136	0	0	0	0	0
0.605284	0	0	0	0	0
0.605284	0	0	0	0	0
0.605284	0	0	0	0	0
0.605284	0	0	0	0	0
0.605284	0	0	0	0	0
0.090601	0.090601	0.090601	0	0	0

Tabel 3.12. Proses Kernelisasi (Lanjutan)

K(1,1)	K(1,2)	K(1,3)	K(1,4)	K(1,5)	K(1,6)
…	…	…	…	…	…
0	0	0	0	0	0
Jumlah					
5.538157	0.090601	0.090601	0	0	0

Setelah dilakukan perhitungan pada seluruh nilai x pada data berita, maka matriks kernel yang terbentuk dari hasil perhitungan $x_i x_j^T$ pada setiap K(1,1), K(1,2), …, K(6,6) adalah sebagai berikut.

$$x_i x_j^T = \begin{bmatrix} 5.538157 & 0.090601 & 0.090601 & 0 & 0 & 0 \\ 0.090601 & 4.327589 & 0.090601 & 0 & 0 & 0 \\ 0.090601 & 0.090601 & 3.117021 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.117021 & 0.090601 & 0.090601 \\ 0 & 0 & 0 & 0.090601 & 3.117021 & 0.090601 \\ 0 & 0 & 0 & 0.090601 & 0.090601 & 3.722305 \end{bmatrix}$$

Matriks di atas setiap elemennya merupakan hasil $x_i.x_j$ yang akan berkorelasi dengan $a_i.a_j$ dalam persamaan (2.22). Dengan menggunakan matriks kernel K sebagai pengganti *dot-product* $x_i.x_j$ dalam persamaan dualitas Lagrange Multiplier (2.22), diperoleh:

$$Ld = \max\ a_1 + a_2 + a_3 + a_4 + a_5 + a_6 - \frac{1}{2}(5.53817a_1^2 + 0.181202a_1a_2$$

$$+ 0.181202a_1a_3 + 4.327589a_2^2 + 0.181202a_2a_3 + 3.117021a_3^2$$
$$+ 3.117021a_4^2 + 0.181202a_4a_5 + 0.181202a_4a_6 + 3.117021a_5^2$$
$$+ 0.181202a_5a_6 + 3.722305a_6^2$$

Persamaan diatas dapat diselesaikan dengan menggunakan persamaan (2.27) untuk memperoleh nilai dari $a_1, a_2, a_3, a_4, a_5, a_6$, dengan mensubstitusikan nilai hasil dari perhitungan, sehingga bentuknya adalah sebagai berikut.

5.538157a1 + 0.090601a2 + 0.090601a3 = 1

0.090601a1 + 4.327589a2 + 0.090601a3 = 1

0.090601a1 + 0.090601a2 + 3.117021a3 = 1

3.117021a4 + 0.090601a5 + 0.090601a6 = -1

0.090601a4 + 3.117021a5 + 0.090601a6 = -1

0.090601a4 + 0.090601a5 + 3.722305a6 = -1

Dengan menyederhanakan ke-6 persamaan di atas, maka diperoleh nilai :

$$a_1 = 0.171948183, \quad a_2 = 0.221093627, \quad a_3 = 0.309472646,$$
$$a_4 = -0.304691494, \; a_5 = -0.304691494, \; a_6 = -0.253937542,$$
$$b = -0{,}160806074.$$

Setelah semua nilai α dan b ditemukan, maka bidang pemisah (*hyperplane*) yang mendiskriminasikan kelas positif dan negatif diberikan oleh persamaan (2.28). Dengan mensubstitusikan nilai, diperoleh nilai w yang akan

mendefinisikan fungsi tujuan pada setiap atribut kata yang dapat dilihat pada Tabel 3.13.

Tabel 3.13. Nilai *w* Atribut

Xi	Kata	W
x_1	dpr	0.267551372
x_2	wacana	0.133775686
x_3	presiden	0.133775686
x_4	dipilih	0.133775686
x_5	melanggengkan	0.133775686
x_6	oligarki	0.133775686
x_7	politik	0.211456851
x_8	sohibul	0.172010842
x_9	iman	0.172010842
x_{10}	safari	0.172010842
x_{11}	pks	0.172010842
x_{12}	partai	0.172010842
x_{13}	demokrat	0.172010842
x_{14}	desember	0.172010842
x_{15}	nurul	0.240769718
x_{16}	qomar	0.240769718
x_{17}	kasusnya	0.240769718

Tabel 3.13. Nilai *w* Atribut (Lanjutan)

Xi	Kata	W
x_{18}	sarat	0.240769718
x_{19}	kepentingan	0.240769718
x_{20}	as	-0.237049982
x_{21}	china	-0.237049982
x_{22}	tegang	-0.237049982
x_{23}	harga	-0.259859479
x_{24}	emas	-0.237049982
x_{25}	merangkak	-0.237049982
x_{26}	alasan	-0.237049982
x_{27}	kementerian	-0.237049982
x_{28}	esdm	-0.237049982
x_{29}	turunkan	-0.237049982
x_{30}	bbm	-0.237049982
x_{31}	kasihan	-0.197563408
x_{32}	peternak	-0.197563408
x_{33}	telur	-0.197563408
x_{34}	ayam	-0.197563408
x_{35}	kandang	-0.197563408
x_{36}	turun	-0.197563408

3.5.6.4 *Test* SVM

Setelah mendapatkan nilai *w* dan *b* atau *hyperplane* dari hasil proses *Train* SVM, selanjutnya data berita *online* baru yang belum memiliki label dapat diklasifikasikan dalam kelas positif atau negatif dengan nilai *w* dan *hyperplane* tersebut. Untuk menentukan kelas dari data baru, maka persamaan *hyperplane* yang digunakan adalah persamaan (2.7).

Sebagai contoh, untuk melakukan klasifikasi data berita dengan isi kalimat "Biang Kerok Anjloknya Harga Telur Ayam Menurut Peternak", dimana kalimat tersebut harus dirubah terlebih dahulu melalui proses *pre-processing text* dan ditransformasikan ke dalam bentuk vektor dokumen. Data hasil transformasi ke dalam bentuk vektor dokumen dapat dilihat pada Tabel 3.14.

Tabel 3.14. Klasifikasi Data Baru

$Xi_{=1,2,\ldots,36}$	$Wi_{=1,2,\ldots,36}$	Wi.Xi
0	0.267551372	0
0	0.133775686	0
0	0.133775686	0
0	0.133775686	0
0	0.133775686	0
0	0.133775686	0
0	0.211456851	0
0	0.172010842	0
0	0.172010842	0

Tabel 3.14. Klasifikasi Data Baru (Lanjutan)

$Xi_{=1,2,\ldots,36}$	$Wi_{=1,2,\ldots,36}$	Wi.Xi
0	0.172010842	0
0	0.172010842	0
0	0.172010842	0
0	0.172010842	0
0	0.172010842	0
0	0.240769718	0
0	0.240769718	0
0	0.240769718	0
0	0.240769718	0
0	0.240769718	0
0	-0.237049982	0
0	-0.237049982	0
0	-0.237049982	0
0.301	-0.259859479	-0.095622256
0	-0.237049982	0
0	-0.237049982	0
0	-0.237049982	0
0	-0.237049982	0
0	-0.237049982	0
0	-0.237049982	0

0	-0.237049982	0
0	-0.197563408	0
0.778	-0.197563408	-0.153704331
0.778	-0.197563408	-0.153704331
0.778	-0.197563408	-0.153704331
0	-0.197563408	0
0	-0.197563408	0
$\sum w_i . x_i$		**$-$ 0.539330696663669**

Dengan mensubtitusikan nilai kedalam persamaan $f(x) = \sum_{i=1} w_i . x_i + b$, dimana $b = -0.160806074$ diperoleh:

$$f(x) = -0.539330696663669 + (-0.160806074) = -0.700136770 66$$

Hasil akhir menunjukkan bahwa nilai $-0.700136770 66 < 0$, sehingga kalimat "Biang Kerok Anjloknya Harga Telur Ayam Menurut Peternak" termasuk kedalam label kelas -1 yang merupakan Berita Ekonomi.

3.5.6.5 Pengujian Performa SVM

Untuk mengecek tingkat akurasi *classifier* yang dibangun, maka digunakan metode *accuracy*. Label keluaran hasil dari pengujian pada proses *Test* SVM akan dibandingkan dengan label aslinya menggunakan persamaan (2.30). Serta untuk mengetahui pengaruh jumlah data latih terhadap efektifitas SVM *classifier* maka akan dilakukan beberapa kombinasi jumlah dokumen latih dan dokumen uji seperti pada Tabel 3.1. Sebagai contoh terdapat Tabel 3.15 yang merupakan

matriks konfusi hasil pengujian SVM *classifier* terhadap 100 data uji sebagai berikut.

Tabel 3.15. Matriks Konfusi Data Pengujian

f_{ij}		Kelas hasil prediksi (j)	
		Kelas = 1	Kelas = 0
Kelas asli(i)	Kelas = 1	$f_{11} = 46$	$f_{10} = 4$
	Kelas = 0	$f_{01} = 2$	$f_{00} = 48$

Sehingga menggunakan persamaan (2.30), diperoleh perhitungan *accuracy* sebagai berikut.

$$\frac{46 + 48}{100} = \frac{94}{100} = 0{,}94 = 94\%$$

Hasil akurasi diatas menunjukkan tingkat performa model klasifikasi dalam melakukan klasfikasi data uji yang belum diketahui label kelasnya ke dalam label kategori yang tepat dengan nilai keakuratan sebesar 94%.

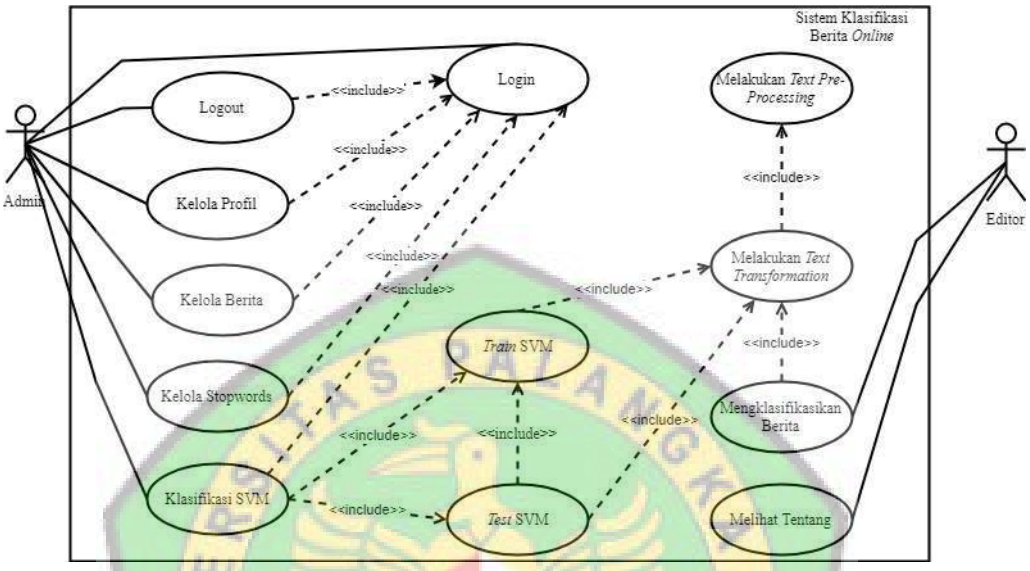
3.6 Desain

3.6.1 *Unified Modelling Language* (UML)

3.6.1.1 *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas proses yang di harapkan terjadi dari sebuah sistem. *Use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Untuk mendefinisikan skenario dari pengunaan sistem

klasifikasi berita *online*, maka kegiatan pengguna dibagi menjadi beberapa kegiatan seperti pada Gambar 3.7.



Gambar 3.7. *Use Case Diagram*

Berikut merupakan deskripsi aktor yang terdapat pada sistem klasifikasi berita *online* yang dijabarkan pada Tabel 3.16.

Tabel 3.16. Deskripsi Aktor

No	Aktor	Deskripsi
1.	Admin	Admin dapat melakukan kelola terhadap data seperti data profil, berita dan stopwords serta melakukan Klasifikasi SVM yang prosesnya meliputi *Train* SVM dan *Test* SVM, dengan mengupload dataset berita yang akan digunakan sebagai data latih (*training set*) dalam

Tabel 3.16. Deskripsi Aktor (Lanjutan)

		membuat model klasifikasi dan data uji (*test set*) untuk mengetahui keakuratan model klasifikasi yang dibangun.
2.	Editor	Editor dapat melakukan klasifikasi terhadap data berita *online* baru yang belum memiliki label dan melihat menu tentang.

Berikut merupakan deskripsi dari *use case* diagram yang terdapat pada Gambar 3.7.

Tabel 3.17. Deskripsi *Use Case*

No	*Use Case*	Deskripsi	Aktor
1.	Login	Tahap ini digunakan untuk admin sebelum mulai melakukan kelola terhadap data dan menggunakan fitur fitur di dalam *website* sistem klasifikasi dimana admin harus login terlebih dahulu	Admin
2.	Kelola Profil	Tahap ini digunakan untuk melakukan kelola terhadap data akun, seperti nama, *username* dan *password*.	Admin
3.	Kelola Berita	Tahap ini digunakan untuk melakukan kelola terhadap data berita yang telah diprediksi label kelasnya.	Admin

Tabel 3.17. Deskripsi *Use Case* (Lanjutan)

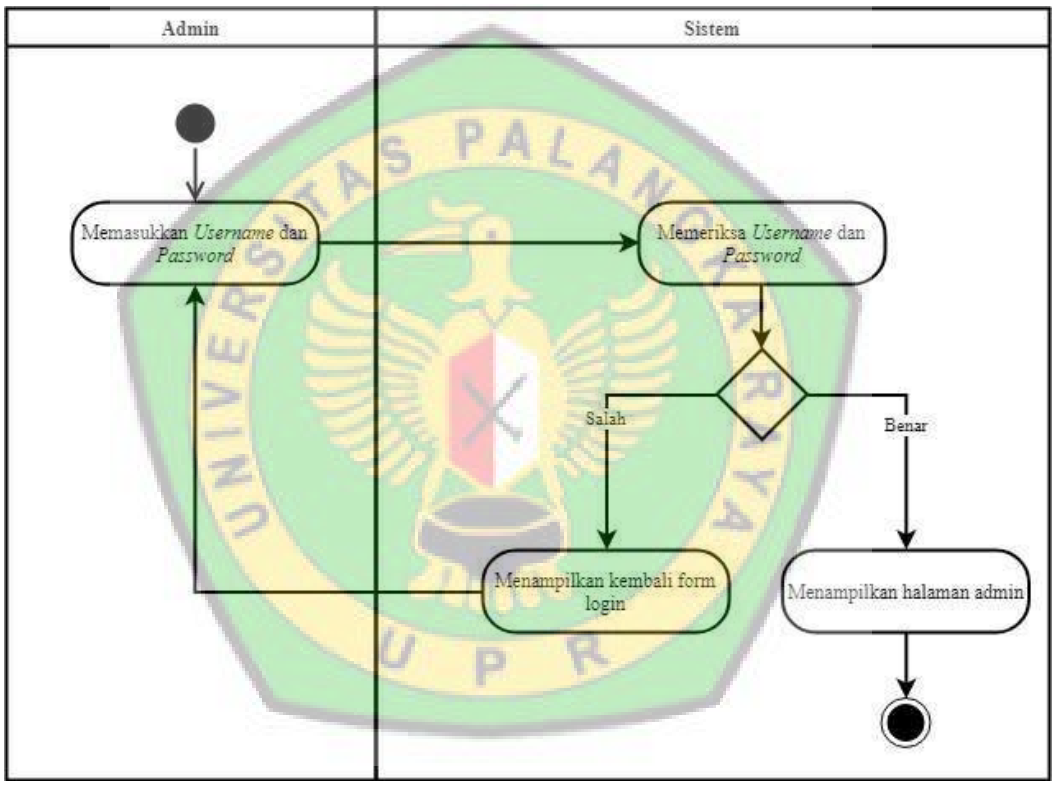
No	*Use Case*	Deskripsi	Aktor
4.	Kelola Stopwords	Tahap ini digunakan untuk melakukan kelola terhadap data stopwords, dalam hal ini adalah menambah, menghapus maupun mengubah data stopwords	Admin
5.	Klasifikasi SVM	Tahap ini digunakan untuk melakukan klasifikasi menggunakan metode SVM yang tahapannya meliputi *Train* SVM dan *Test* SVM	Admin
6.	*Train* SVM	Tahap ini akan melakukan pelatihan terhadap data latih setelah dilakukan *pre-processing text* dan *text transformation* yang menghasilkan model klasifikasi	Admin
7.	*Test* SVM	Tahap ini digunakan untuk melakukan pengujian terhadap data uji berdasarkan model klasifikasi yang telah dibangun sebelumnya pada proses *Train* SVM guna mengetahui performa dari model klasifikasi yang dibangun.	Admin
8.	*Text Pre-Processing*	Tahap ini digunakan untuk melakukan pra pemprosesan data teks agar dapat menjadi data yang lebih siap digunakan	Admin

Tabel 3.17. Deskripsi *Use Case* (Lanjutan)

No	*Use Case*	Deskripsi	Aktor
		dalam klasifikasi. Proses yang dilakukan pada tahapan ini meliputi *remove punctuation*, *case folding*, tokenisasi dan *stopwords filtering*	
9.	*Text Transformation*	Tahap ini digunakan untuk melakukan transformasi teks menggunakan metode pembobotan TF-IDF	Admin
10.	Logout	Tahap ini digunakan untuk keluar dari sistem guna mengakhiri sesi setelah melakukan kelola terhadap sistem klasifikasi	Admin
11.	Mengklasifikasikan Berita	Tahap ini digunakan untuk melakukan klasifikasi terhadap data berita *online* baru yang belum memiliki label.	Editor
12.	Melihat Tentang	Tahap ini menampilkan informasi singkat mengenai sistem klasifikasi berita *online*.	Editor

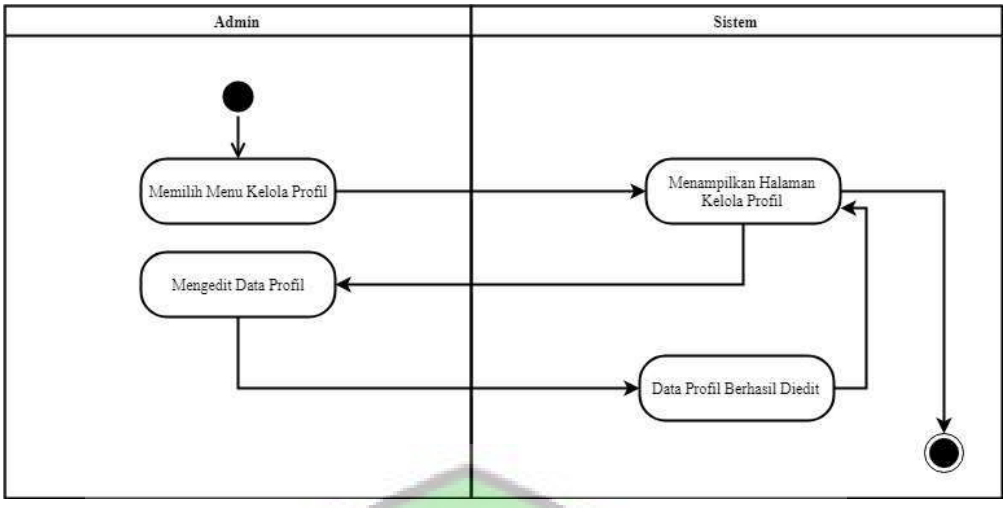
3.6.1.2 *Activity Diagram*

Activity diagram digunakan untuk memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Dalam sistem klasifikasi ini, aktifitas yang dapat digambarkan dalam *activity diagram* adalah sebagai berikut.



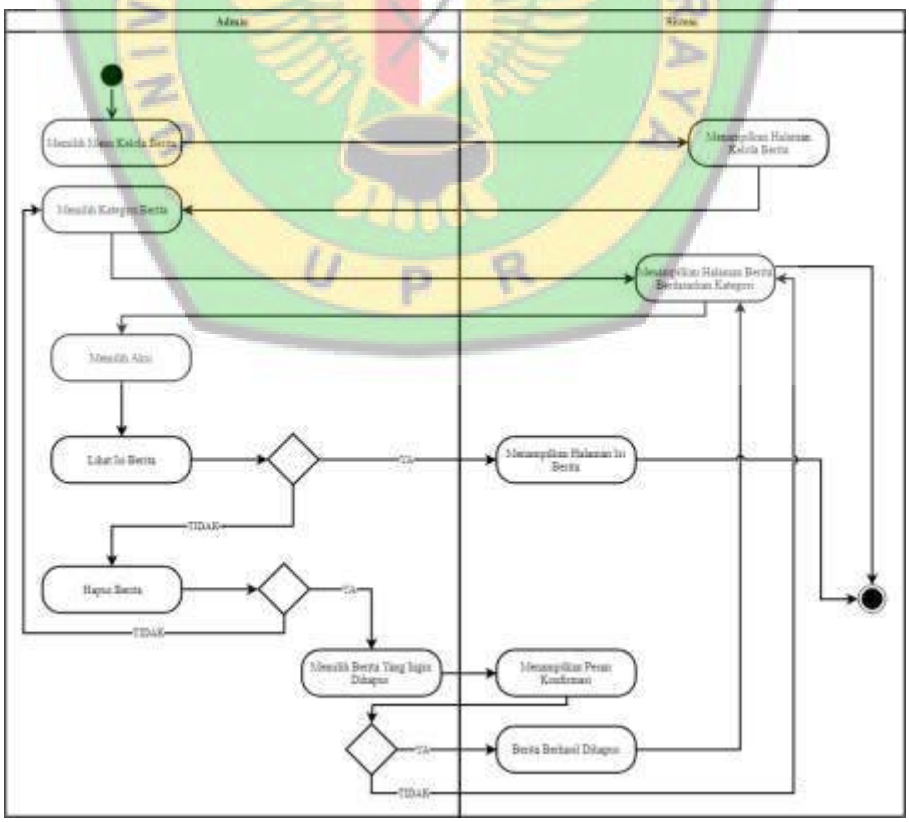
Gambar 3.8. *Activity Diagram* Login

Gambar 3.8 merupakan *activity diagram* untuk melakukan login pada sistem yang dimulai dengan admin menginputkan *username* dan *password* ke dalam sistem, apabila benar maka akan diteruskan ke halaman beranda admin.



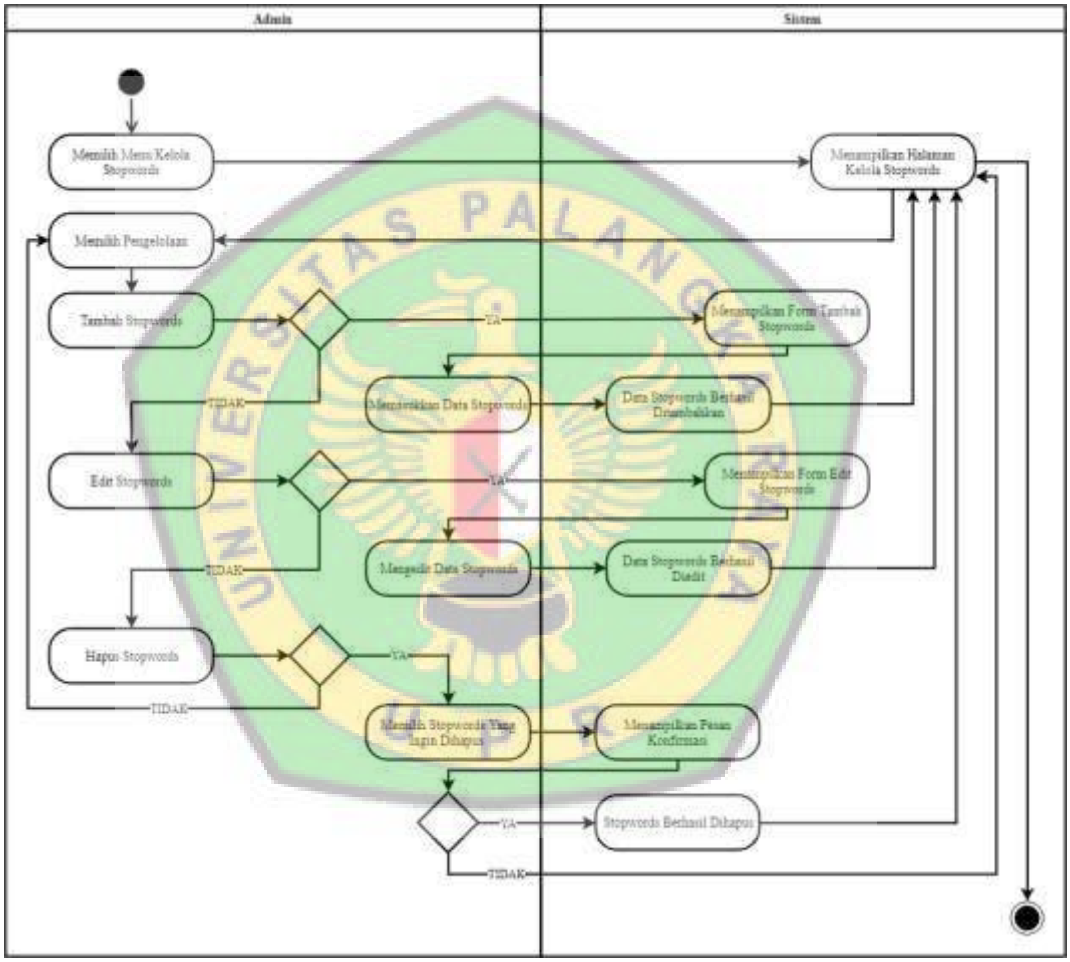
Gambar 3.9. *Activity Diagram* Kelola Profil

Gambar 3.9 merupakan *activity diagram* untuk melakukan kelola profil pada sistem, dimana admin dapat mengedit data profil.



Gambar 3.10. *Activity Diagram* Kelola Berita

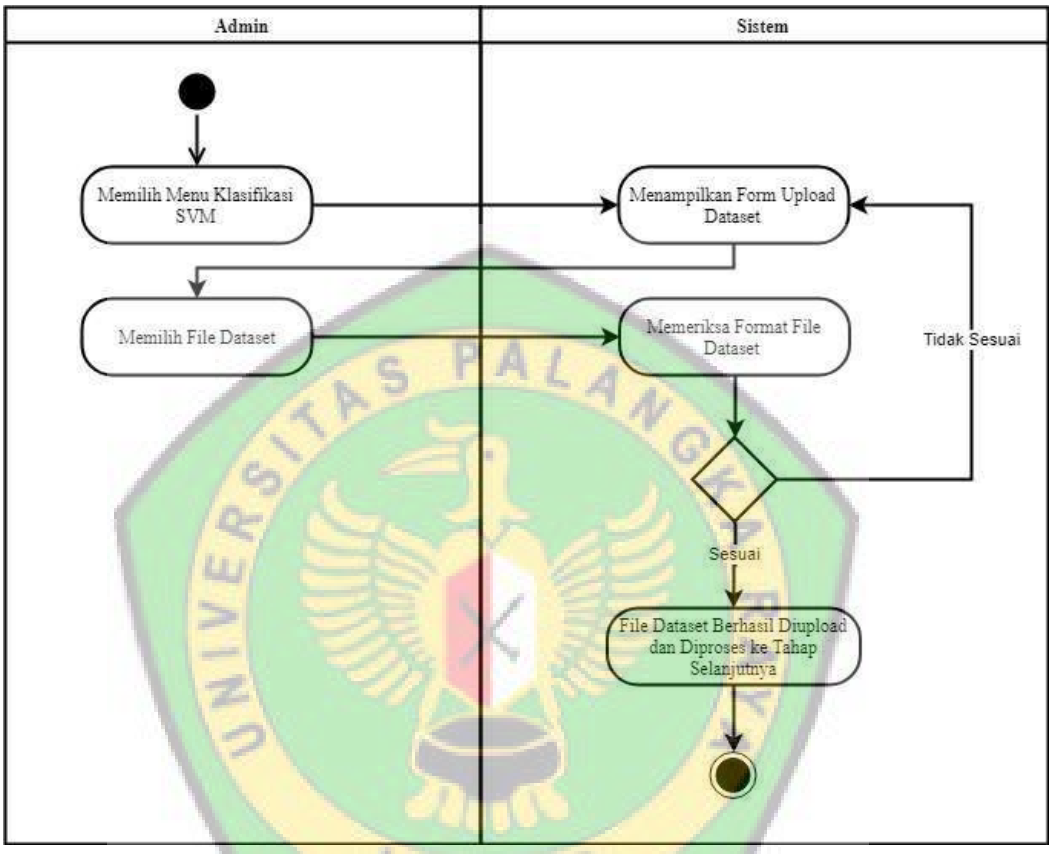
Gambar 3.10 merupakan *activity diagram* untuk melakukan kelola terhadap data berita yang telah diprediksi label kelasnya, dimulai dengan admin memilih kategori berita yang ingin dikelola, kemudian admin dapat melakukan pengelolaan sesuai dengan kebutuhan, seperti menghapus data berita.



Gambar 3.11. *Activity Diagram* Kelola Stopwords

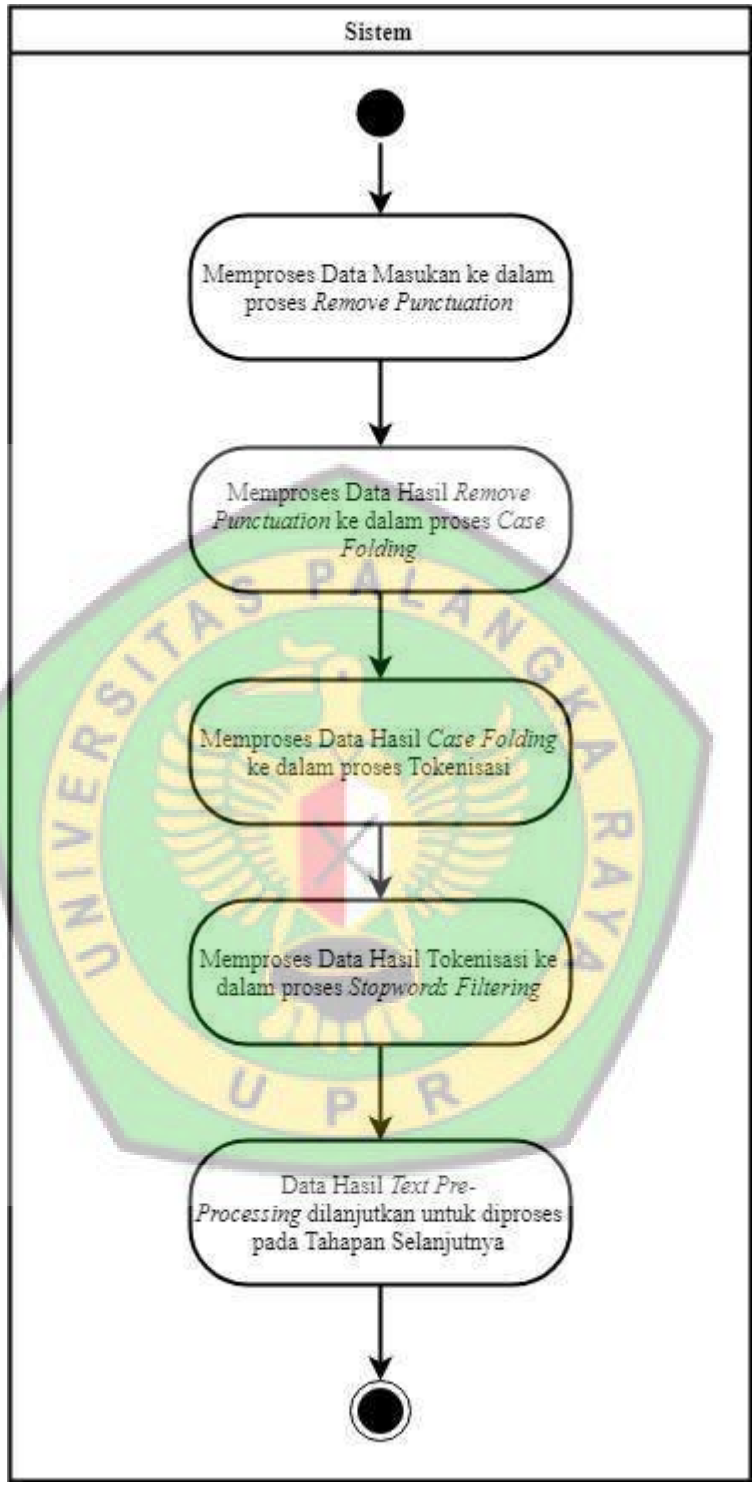
Gambar 3.11 merupakan *activity diagram* untuk melakukan kelola stopwords pada sistem yang dimulai dengan admin memilih menu kelola stopwords kemudian admin dapat melakukan pengelolaan sesuai dengan

kebutuhan, yaitu menambah stopwords, mengedit stopwords dan menghapus stopwords.



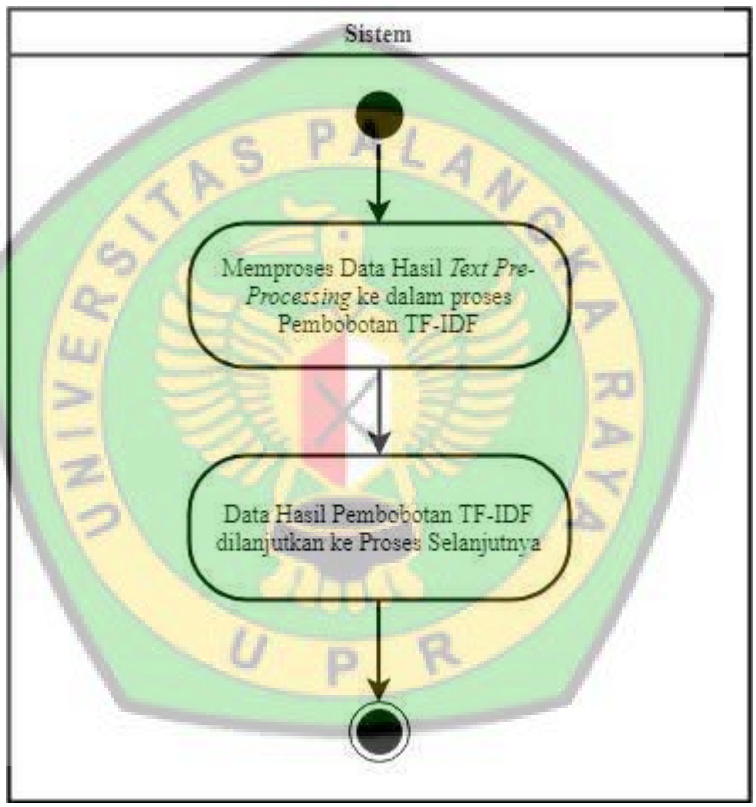
Gambar 3.12. *Activity Diagram* Klasifikasi SVM

Gambar 3.12 merupakan *activity diagram* untuk melakukan Klasifikasi SVM. Setelah admin memilih menu Klasifikasi SVM, sistem akan meminta admin untuk mengupload file sebagai dataset yang akan digunakan dalam proses pelatihan dan pengujian. Setelah sistem mengecek kesesuaian file dataset yang diupload maka dataset akan diproses ketahap pemrosesan selanjutnya.



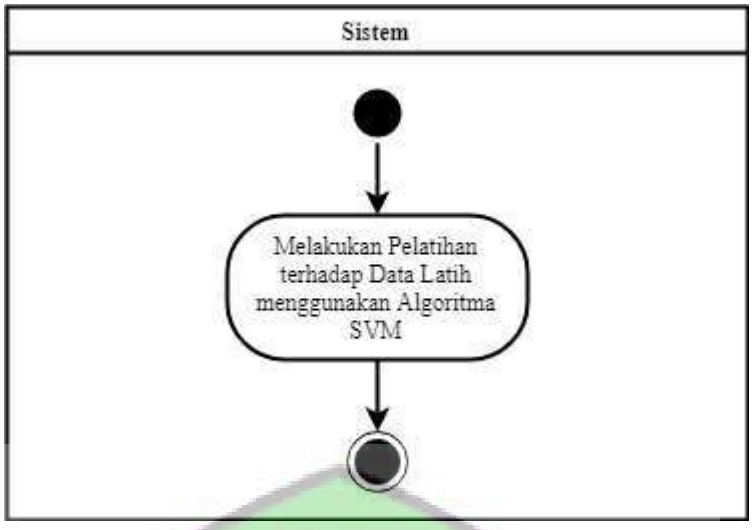
Gambar 3.13. *Activity Diagram Text Pre-Processing*

Gambar 3.13 merupakan *activity diagram* untuk melakukan *Text Pre-Processing*. Pada proses ini sistem akan melakukan beberapa tahapan proses untuk mengolah data teks kedalam format yang lebih siap digunakan. Proses yang dilakukan meliputi *remove punctuation*, *case folding*, tokenisasi dan *stopwords filtering*.



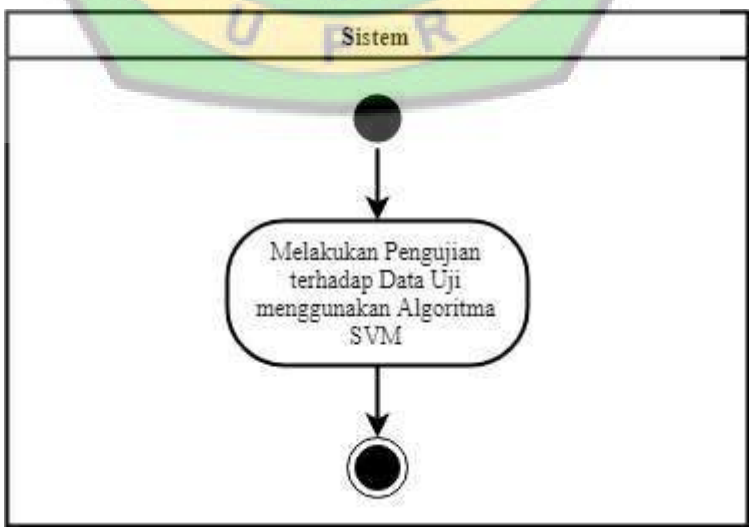
Gambar 3.14. *Activity Diagram Text Transformation*

Gambar 3.14 merupakan *activity diagram* untuk melakukan *Text Transformation.* Pada proses ini sistem akan melakukan transformasi teks kedalam format yang lebih representatif, yaitu menggunakan metode pembobotan TF-IDF.



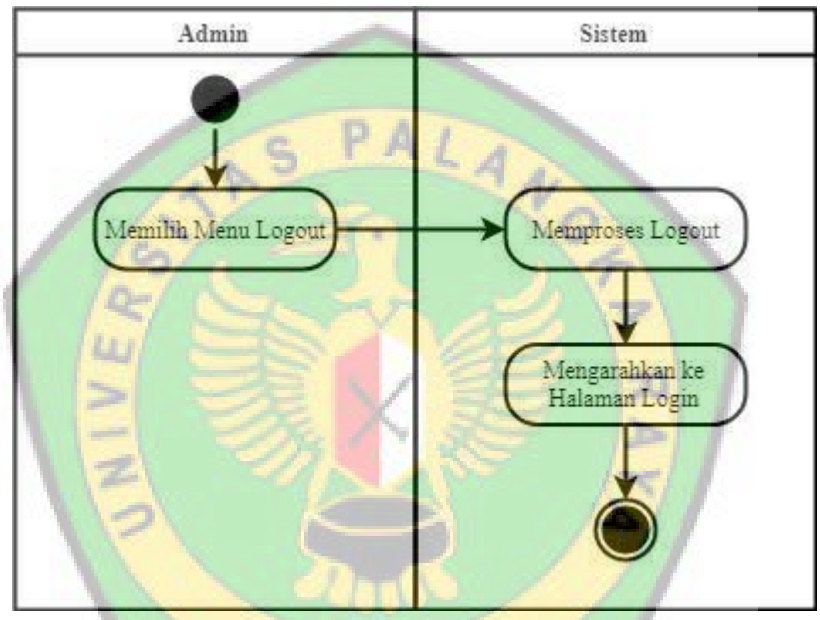
Gambar 3.15. *Activity Diagram Train* SVM

Gambar 3.15 merupakan *activity diagram* untuk melakukan *Train* SVM. Pada proses ini sistem akan melakukan pelatihan berdasarkan data latih dengan menggunakan algoritma *Support Vector Machine* dalam membangun model klasifikasi.



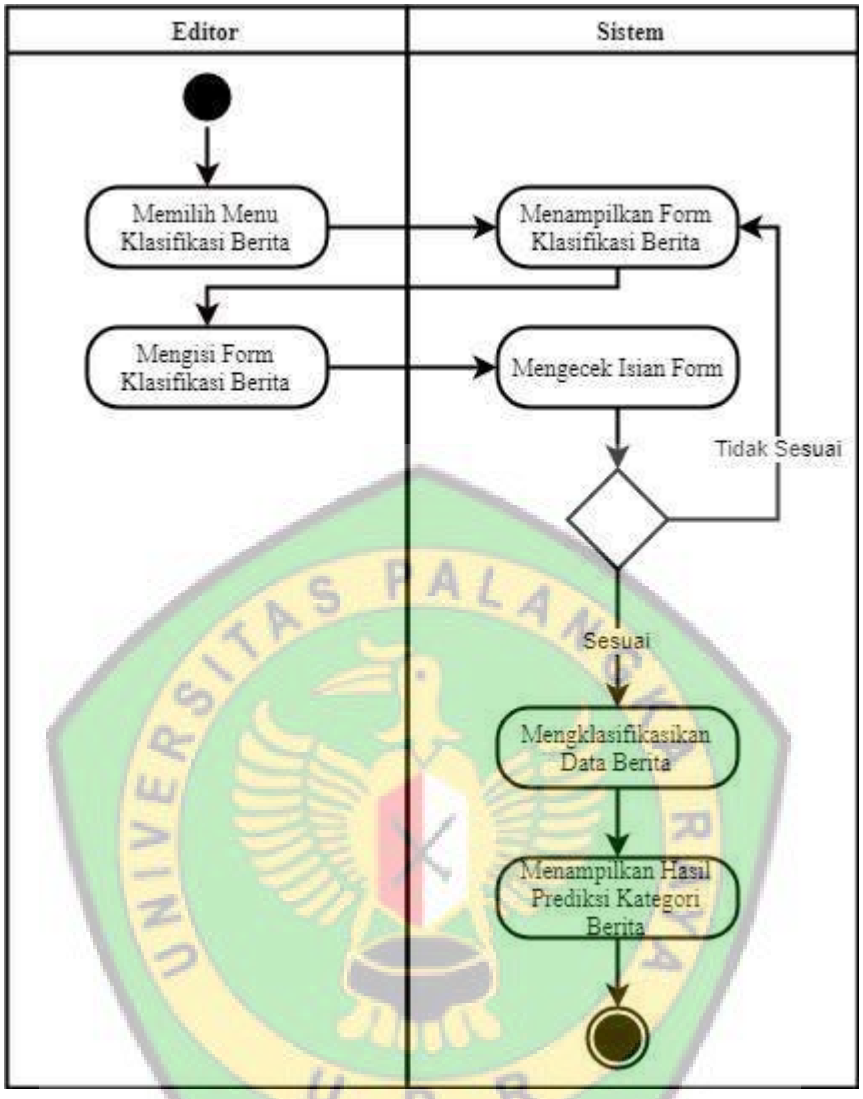
Gambar 3.16. *Activity Diagram Test* SVM

Gambar 3.16 merupakan *activity diagram* untuk melakukan *Test* SVM. Pada proses ini sistem akan melakukan pelatihan berdasarkan data uji dengan menggunakan algoritma *Support Vector Machine* dalam mengetahui performa dari sistem klasifikasi dalam mengelompokkan data berita baru kedalam label kelas yang tepat.



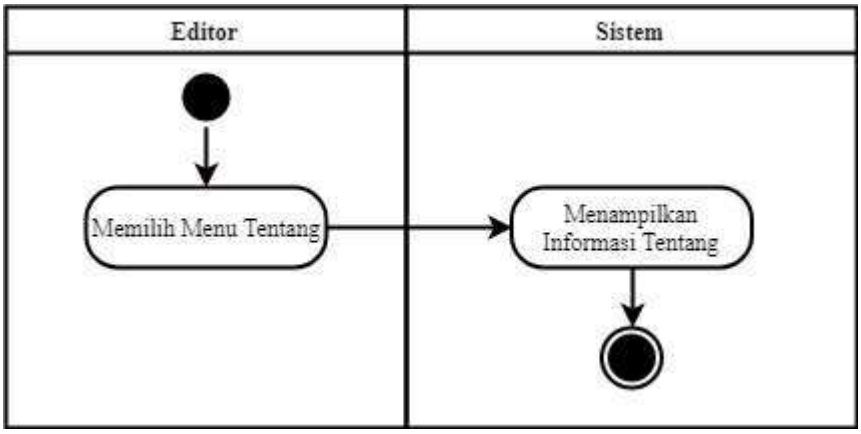
Gambar 3.17. *Activity Diagram* Logout

Gambar 3.17 merupakan *activity diagram* untuk melakukan Logout. Pada proses ini setelah admin memilih menu logout, maka sistem akan mengakhiri sesi admin dan mengarahkan admin menuju ke halaman login kembali.



Gambar 3.18. *Activity Diagram* Klasifikasi Berita

Gambar 3.18 merupakan *activity diagram* untuk melakukan Klasifikasi Berita. Pada proses ini, editor akan mengisi form klasifikasi berita yang meliputi judul dan dokumen berita yang akan diklasifikasikan kategori kontennya. Apabila form yang diisi sesuai, maka sistem akan mulai mengklasifikasikan data berita dan memberikan hasil label konten berita.

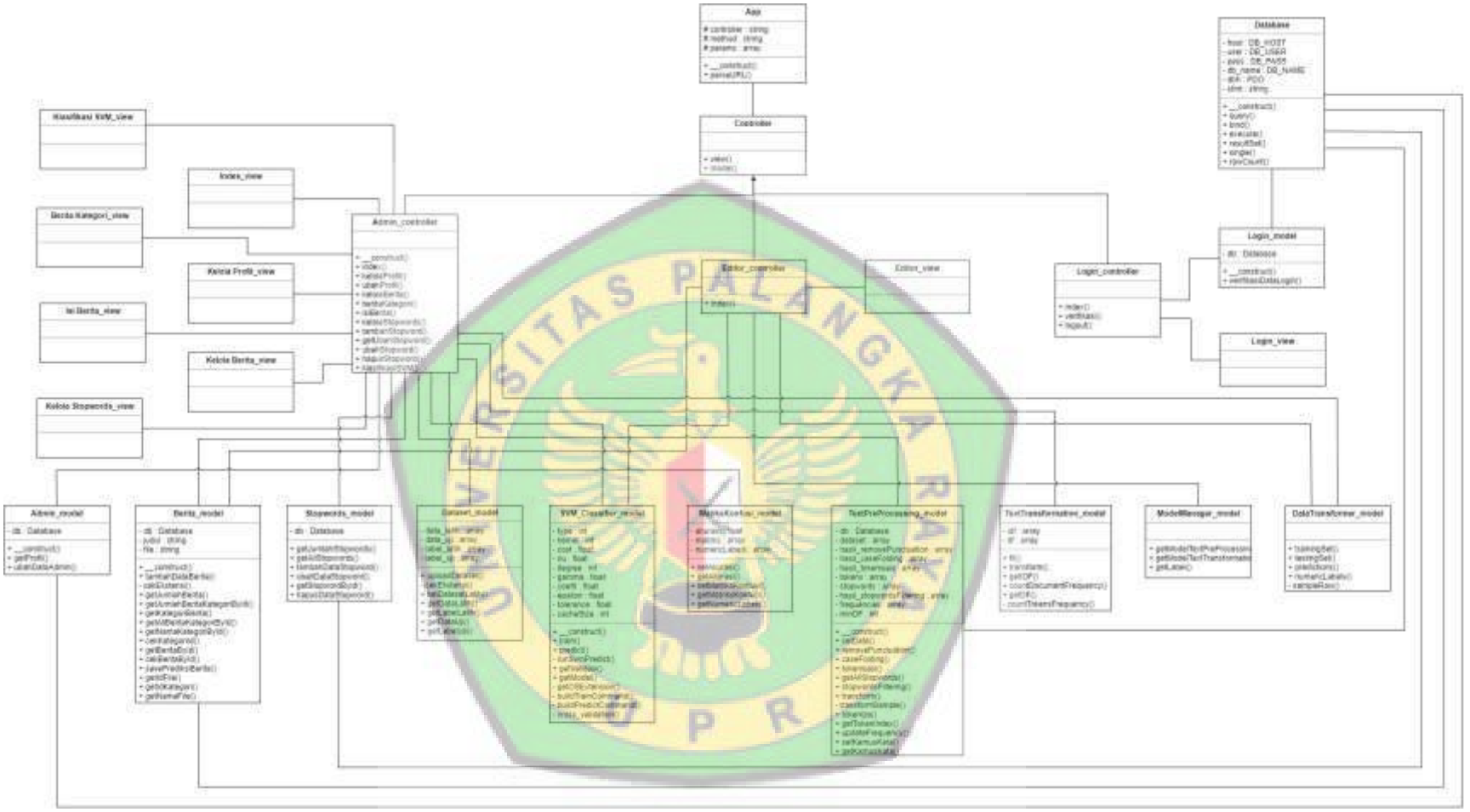


Gambar 3.19. *Activity Diagram* Tentang

Gambar 3.19 merupakan *activity diagram* ketika memilih menu Tentang. Pada proses ini, setelah editor memilih Tentang maka sistem akan menampikan informasi tentang kepada editor.

3.6.1.3 *Class Diagram*

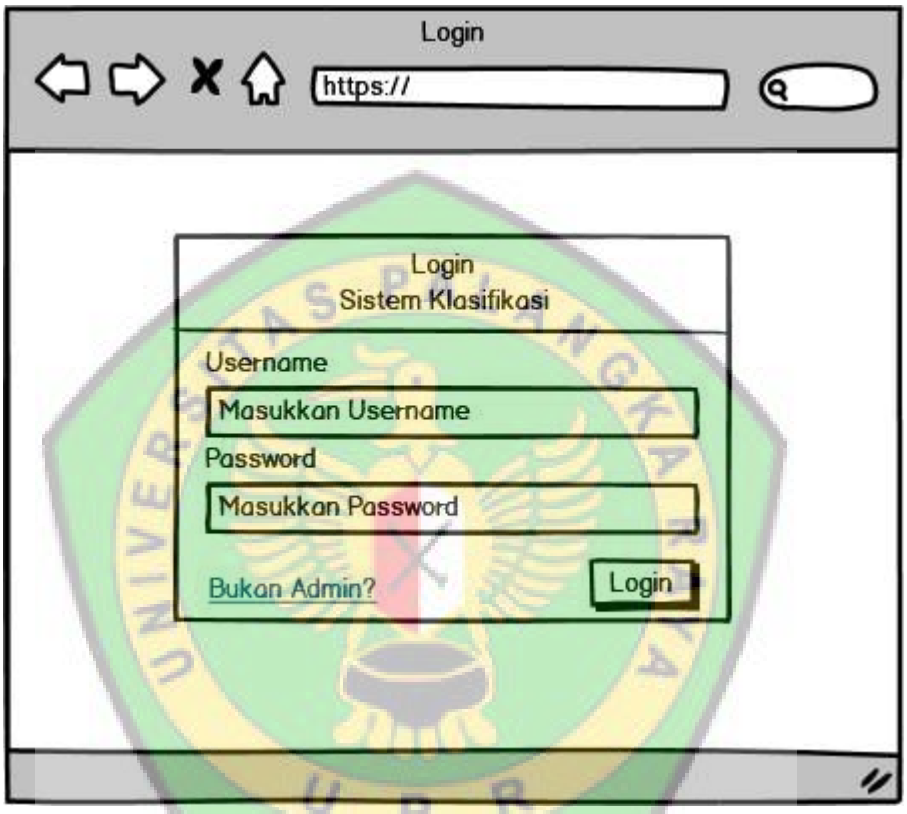
Class Diagram berguna untuk menggambarkan interaksi antar kelas didalam sistem. Adapun *class diagram* pada sistem klasifikasi berita *online* pada penelitian digambarkan pada Gambar 3.20.



Gambar 3.20. *Class Diagram* Sistem Klasifikasi

3.6.2 Desain *User Interface*

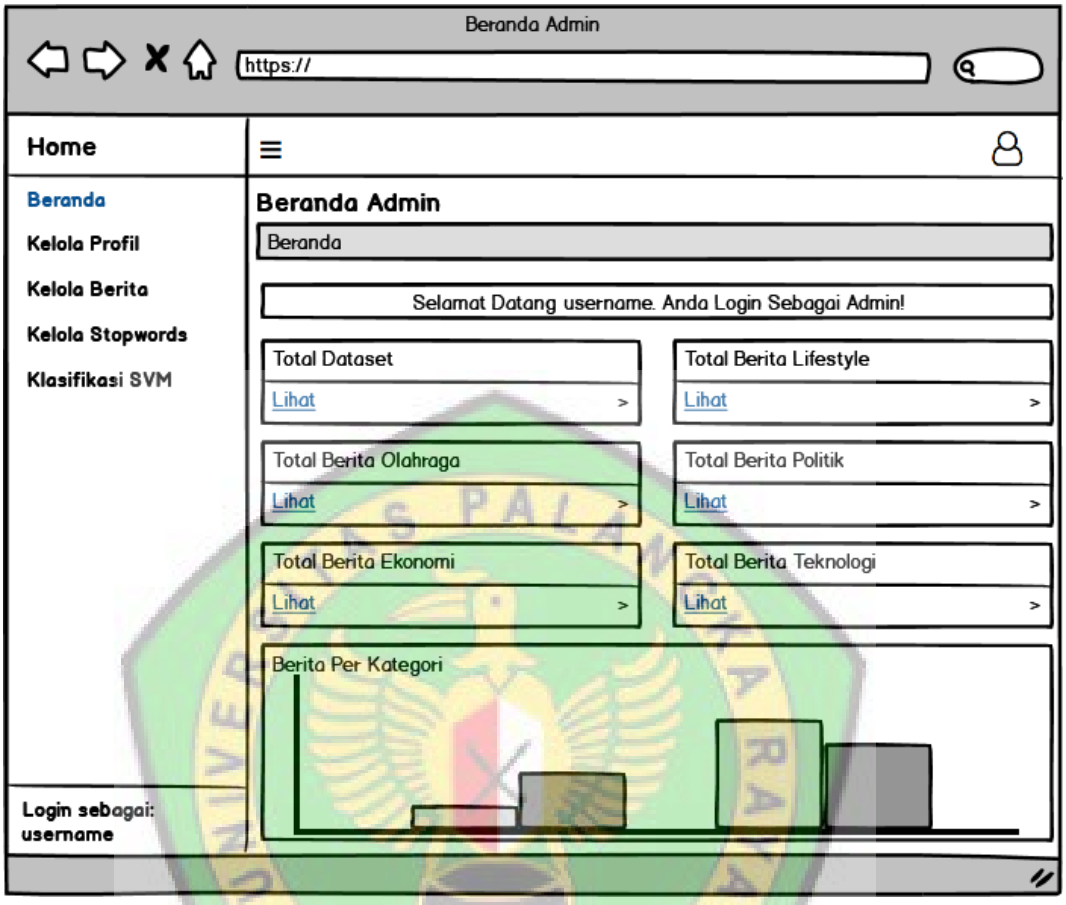
Desain *user interface* menggambarkan tampilan antar muka pada sistem klasifikasi berita *online*. Desain antar muka yang dilakukan meliputi Desain UI Admin dan Desain UI Editor.



Gambar 3.21. Halaman Login

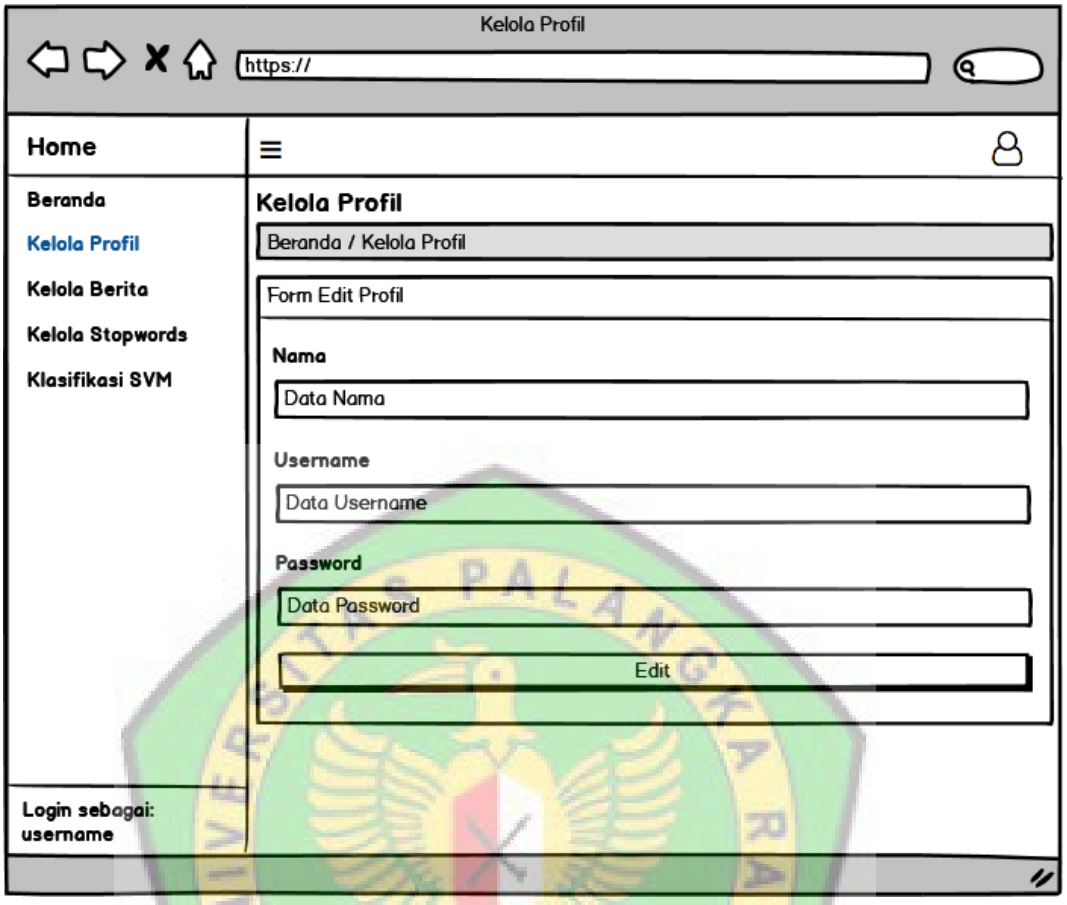
Gambar 3.21 adalah desain antar muka halaman login yang merupakan halaman pertama yang diakses apabila *user* ingin masuk sebagai Admin. *User* diharuskan menginputkan *username* dan *password* yang valid agar dapat masuk ke dalam sistem klasifikasi sebagai Admin. Jika *user* bukan admin, maka *user* dapat mengklik pada link Bukan Admin dan akan diarahkan ke halaman Beranda Editor.

3.6.2.1 Desain UI Admin



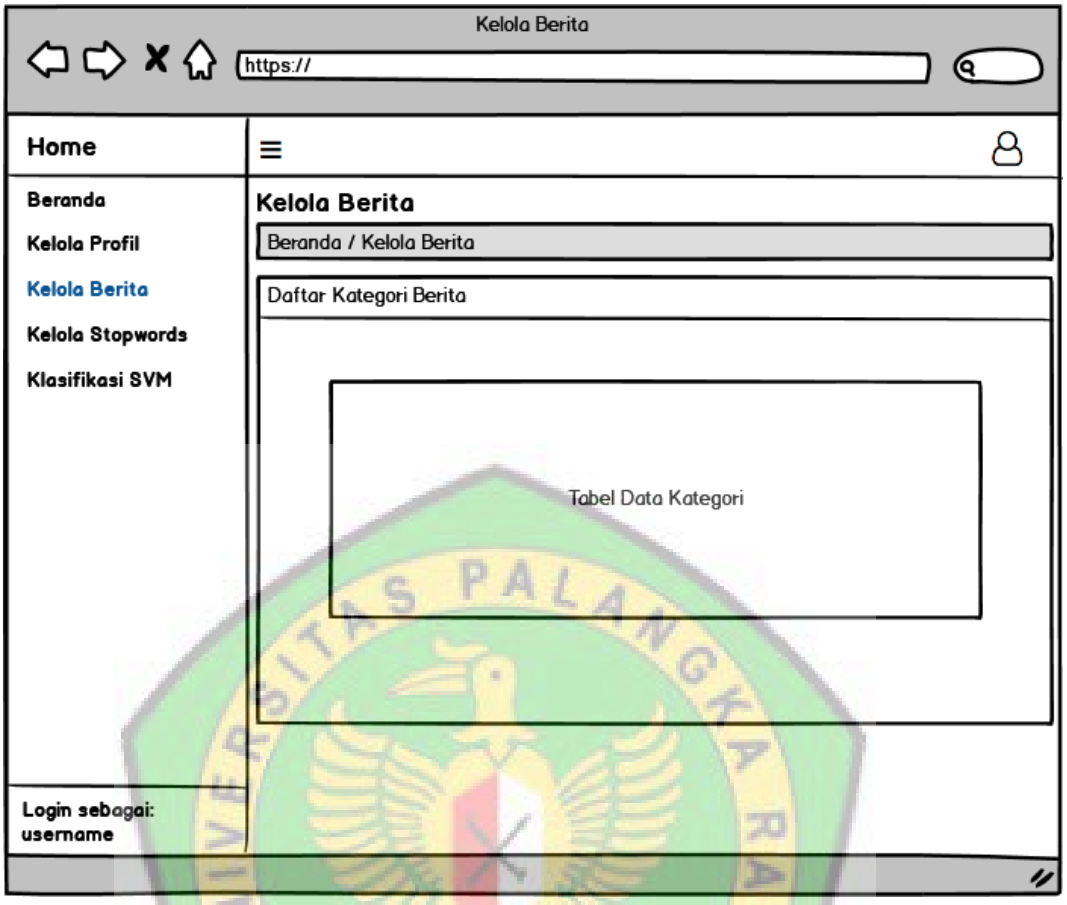
Gambar 3.22. Halaman Beranda Admin

Gambar 3.22 adalah desain antar muka halaman beranda admin. Setelah *user* menginputkan *username* dan *password* yang valid, maka *user* akan dapat login kedalam sistem sebagai Admin dan akan diarahkan ke halaman beranda admin. Halaman beranda admin akan menampilkan informasi-informasi umum tentang data yang terdapat didalam database sistem klasifikasi berita *online* seperti total data stopwords, total data berita pada setiap kategori berita dan bar chart yang akan menampilkan secara visual jumlah berita per kategori.



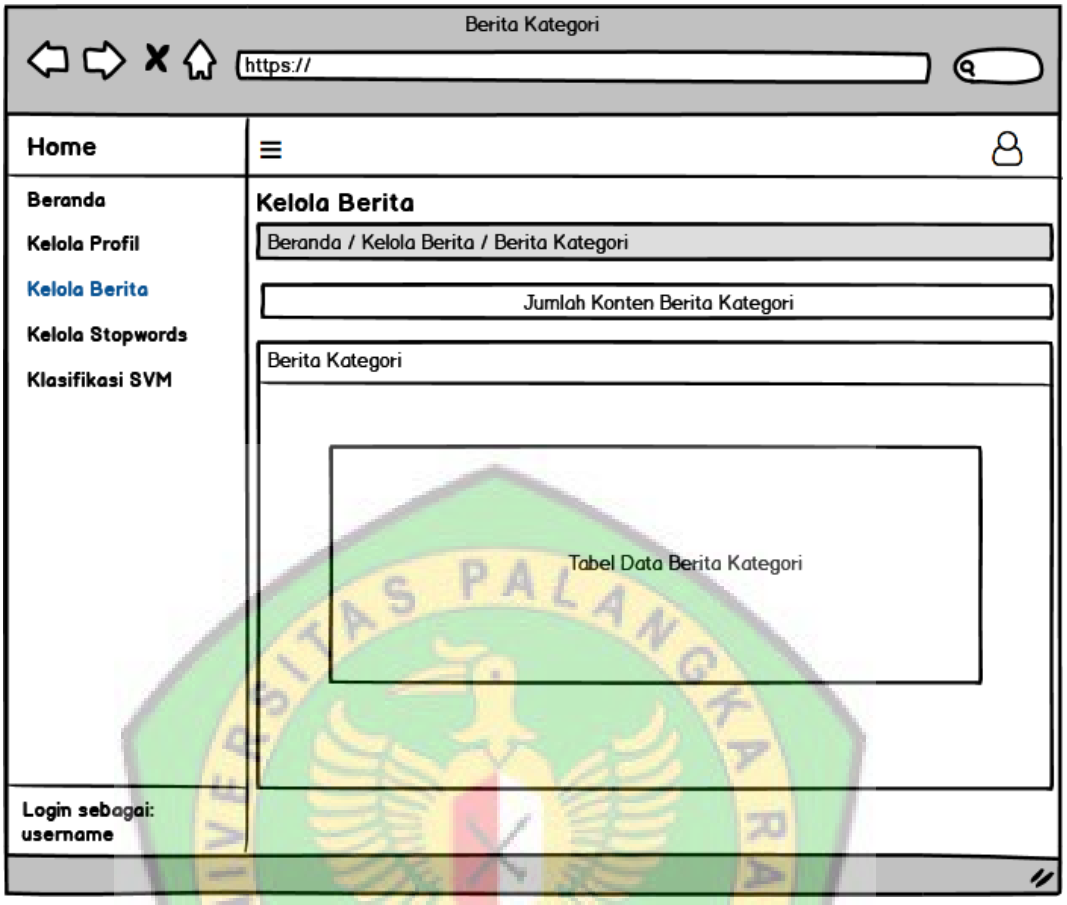
Gambar 3.23. Halaman Kelola Profil

Gambar 3.23 adalah desain antar muka halaman kelola profil. Pada halaman ini admin dapat mengubah data akun yang dimilikinya, seperti mengubah data nama, data *username* dan data *password*.



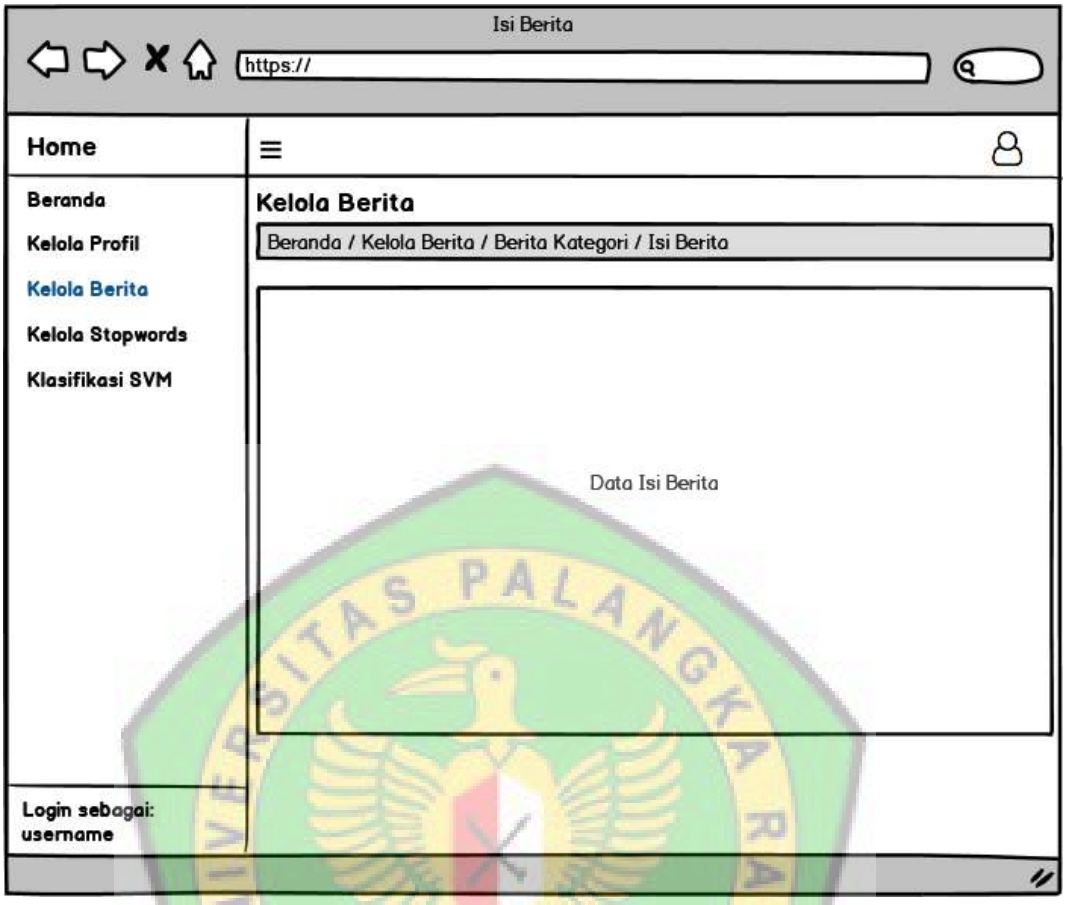
Gambar 3.24. Halaman Kelola Berita

Gambar 3.24 adalah desain antar muka halaman kelola berita. Untuk dapat mengelola berita, admin akan memilih kategori konten berita yang hendak dikelola berdasarkan tabel data kategori yang ditampilkan. Setelah memilih kategori konten berita, maka admin dapat melakukan proses kelola pada berita yang ingin dilakukan proses selanjutnya.



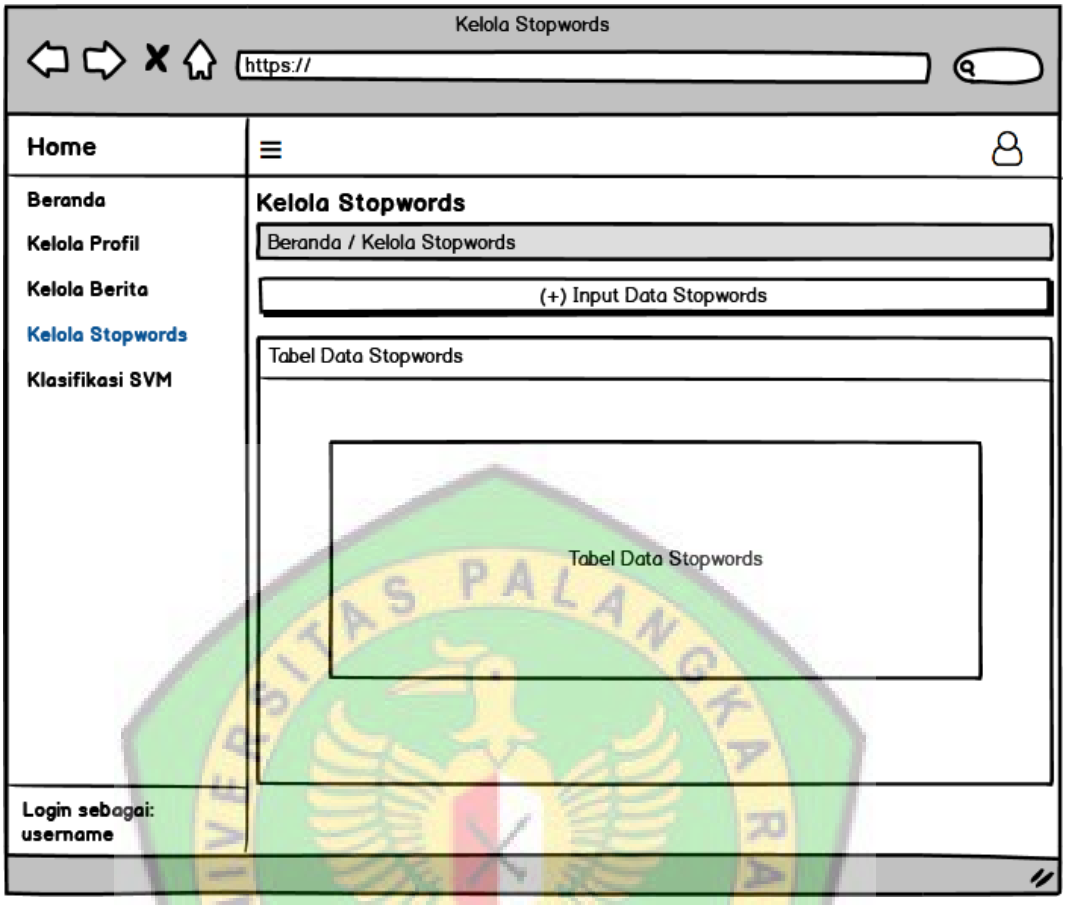
Gambar 3.25. Halaman Berita Kategori

Gambar 3.25 adalah desain antar muka halaman berita kategori. Halaman ini akan menampilkan data berita berdasarkan kategori kontennya. Pada halaman ini admin dapat memilih berita untuk dihapus.



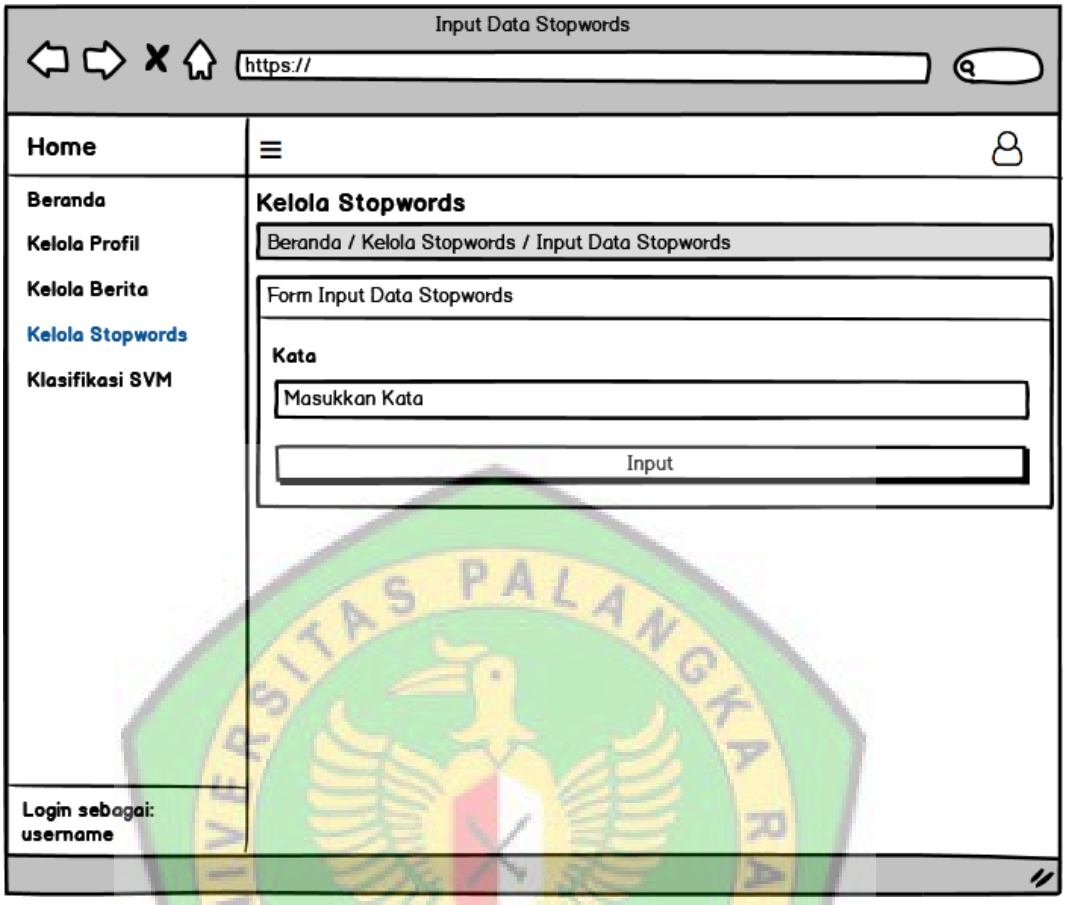
Gambar 3.26. Halaman Isi Berita

Gambar 3.26 adalah desain antar muka Halaman Isi Berita. Halaman ini akan menampilkan data isi berita berdasarkan judul berita yang dipilih.



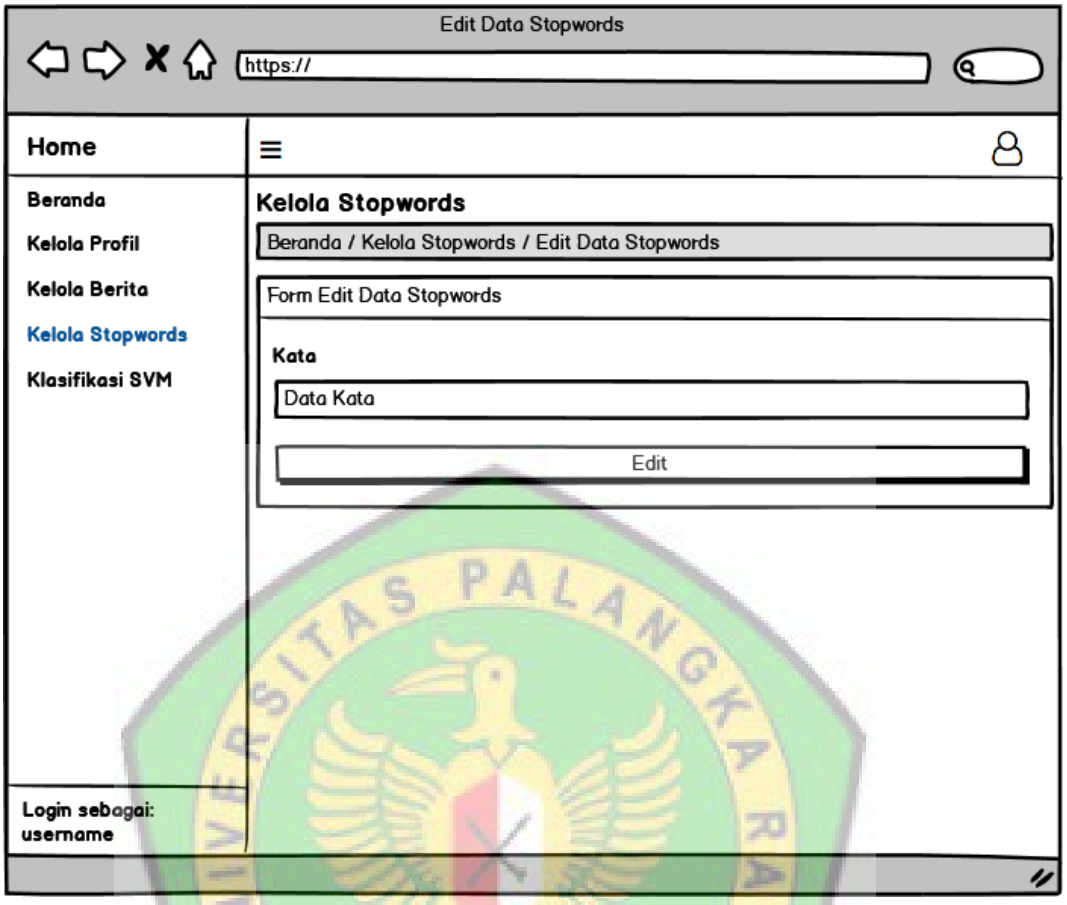
Gambar 3.27. Halaman Kelola Stopwords

Gambar 3.27 adalah desain antar muka halaman kelola stopwords. Pada halaman ini, akan ditampilkan data stopwords yang terdapat di dalam *database* sistem, kemudian admin juga dapat melakukan proses kelola terhadap data stopwords seperti menginputkan data stopwords baru, mengedit data stopwords dan menghapus data stopwords.



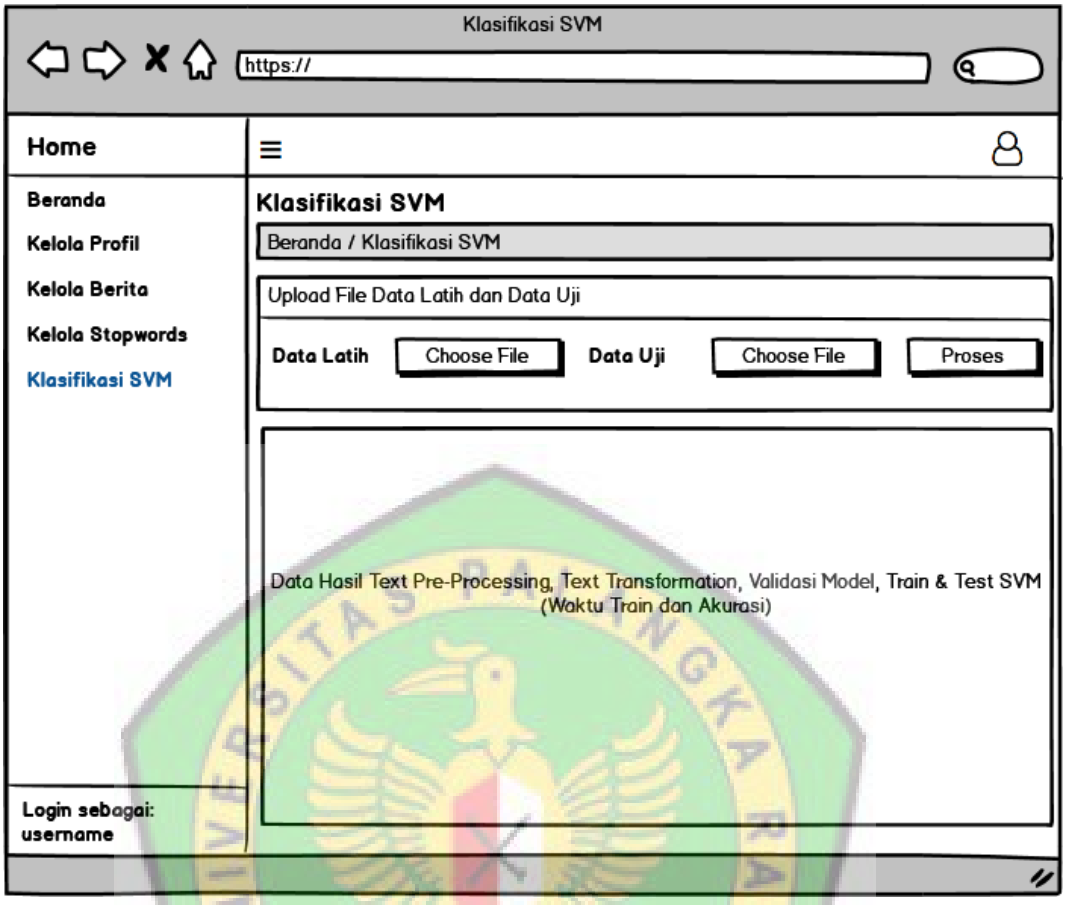
Gambar 3.28. Halaman Input Data Stopwords

Gambar 3.28 merupakan desain antar muka halaman input data stopwords. Pada halaman ini, akan ditampilkan sebuah form untuk menginputkan kata stopword yang ingin ditambahkan ke dalam *database* tabel stopwords.



Gambar 3.29. Halaman Edit Data Stopwords

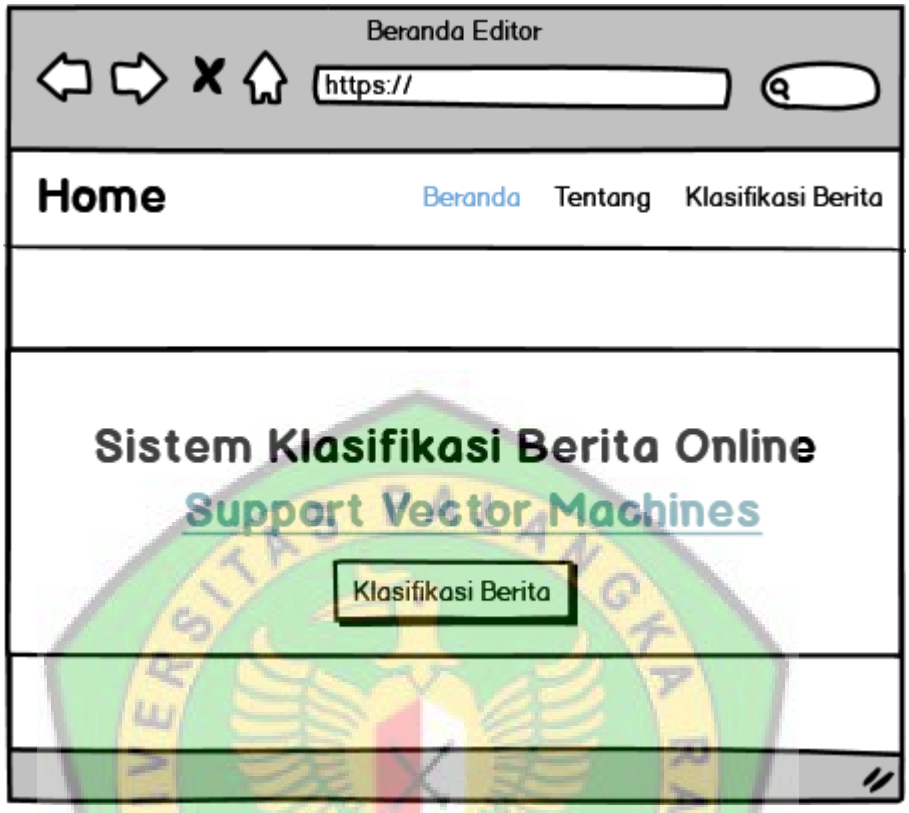
Gamabr 3.29 merupakan desain antar muka halaman edit data stopwords. Pada halaman ini, akan ditampilkan sebuah form untuk mengedit kata stopword yang ingin di ubah di dalam *database* tabel stopwords.



Gambar 3.30. Halaman Klasifikasi SVM

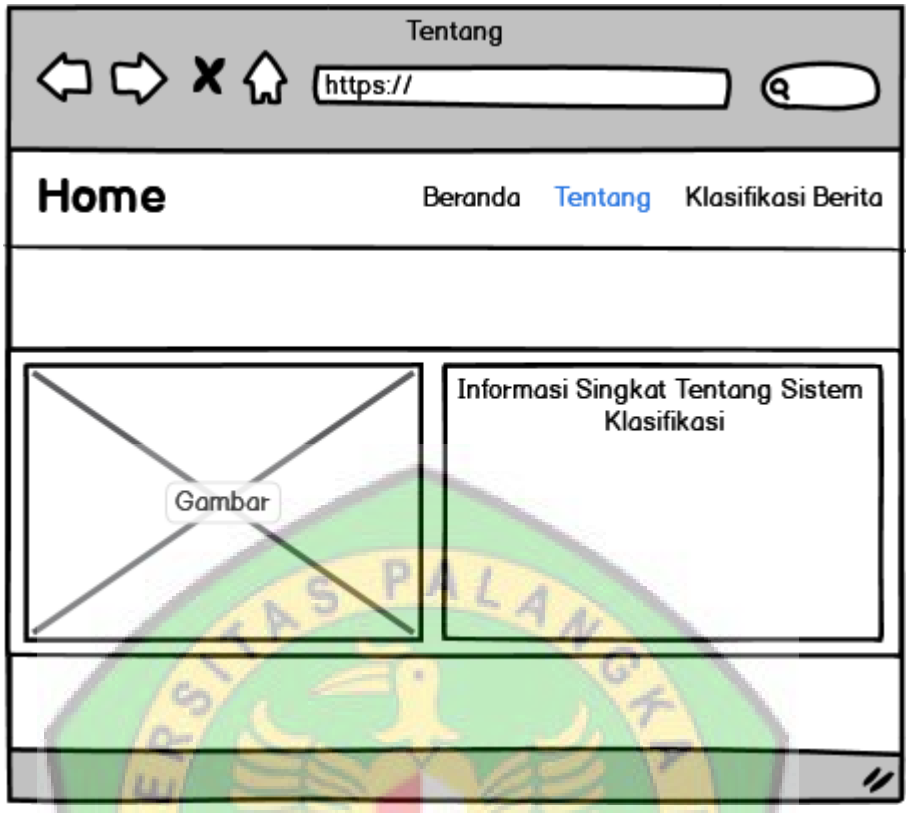
Gambar 3.30 merupakan desain antar muka Halaman Klasifikasi SVM. Pada halaman ini, untuk melakukan klasifikasi SVM, admin diharuskan untuk menginputkan dua file terlebih dahulu, yaitu file data latih dan file data uji. File data latih akan digunakan untuk melakukan proses Train SVM dalam membangun model klasifikasi sedangkan data uji akan digunakan untuk mengetahui tingkat performa model klasifikasi yang telah dibangun. Setelah menginputkan kedua file tersebut, maka sistem akan menampilkan data hasil pemrosesan.

3.6.2.2 Desain UI Editor



Gambar 3.31. Halaman Beranda Editor

Gambar 3.31 merupakan desain tampilan antar muka Halaman Beranda Editor. Halaman ini merupakan halaman utama yang akan ditampilkan saat *user* masuk ke dalam sistem klasifikasi berita *online* sebagai Editor (Bukan Admin).

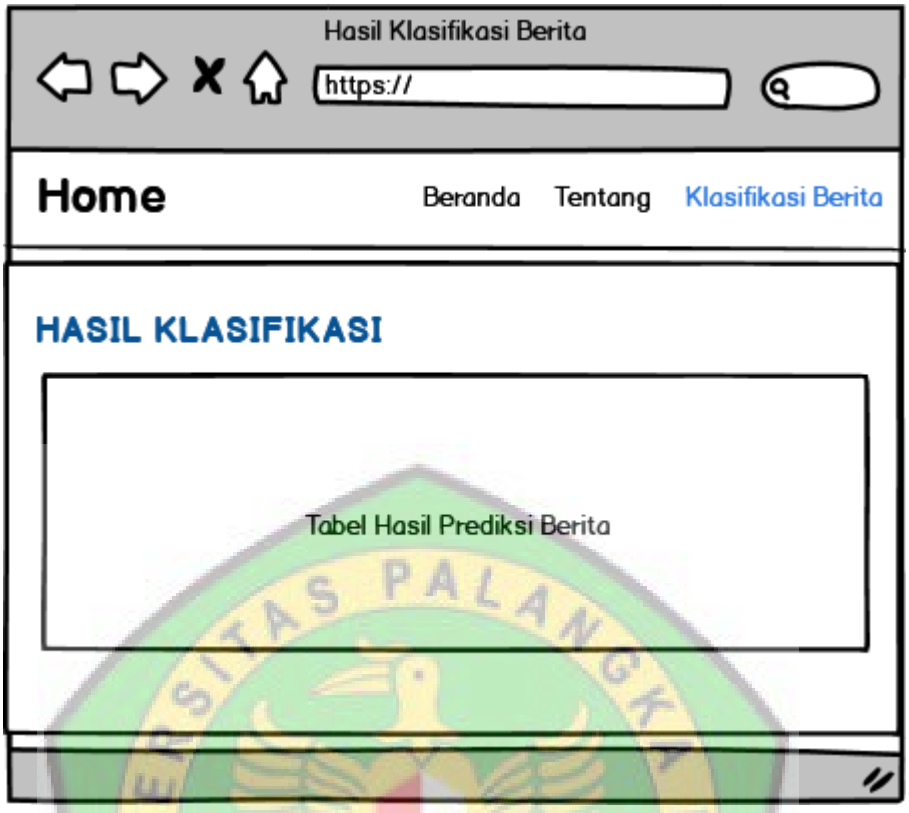


Gambar 3.32. Halaman Tentang

Gambar 3.32 merupakan desain tampilan antar muka Halaman Tentang. Halaman tentang akan menampilkan informasi singkat mengenai sistem klasifikasi berita.

Gambar 3.33. Halaman Klasifikasi Berita

Gambar 3.33 merupakan desain antar muka Halaman Klasifikasi Berita. Pada halaman ini, untuk melakukan klasifikasi berita yang belum diketahui label kelasnya, editor harus menginputkan judul berita dan file berita dengan eksteknsi file .txt terlebih dahulu.



Gambar 3.34. Halaman Hasil Klasifikasi Berita

Gambar 3.34 merukan desain tampilan antar muka Halaman Hasil Klasifikasi Berita. Pada halaman ini akan ditampilkan hasil prediksi kategori konten berita berdasarkan file berita yang diinputkan oleh editor.

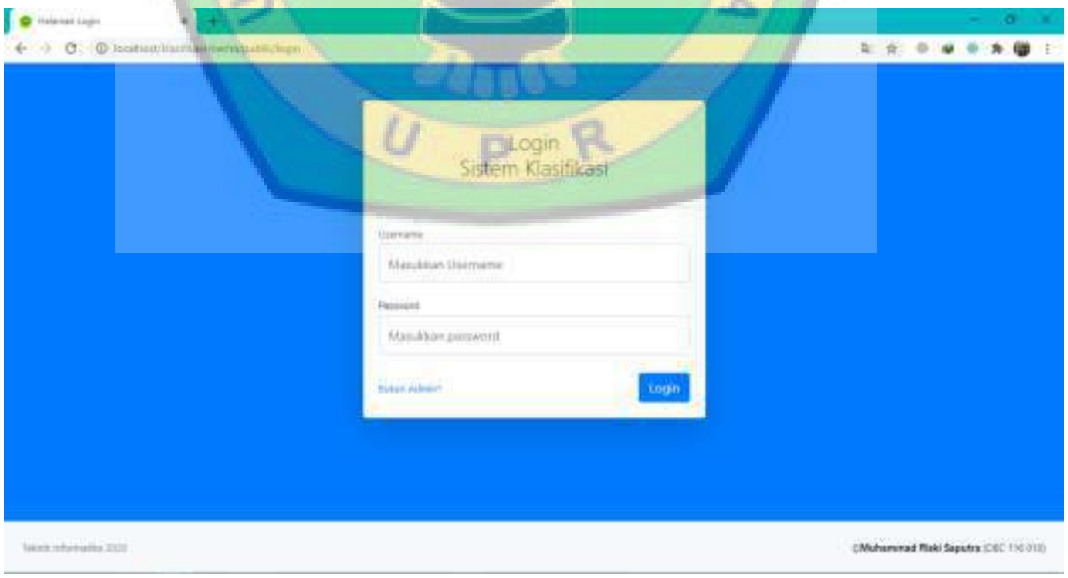
BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan memaparkan mengenai implementasi dan pengujian berdasarkan hasil analisis dan desain sistem yang telah dilakukan sebelumnya. Adapun implementasi *website* sistem klasifikasi berita *online* dapat diakses melalui *link* "klasifikasiberita.com".

4.1 Implementasi *User Interface*

Implementasi *user interface* merupakan implementasi dari desain yang sudah dirancang pada tahap sebelumnya. Semua *interface* sistem untuk semua jenis pengguna akan dibahas pada tahap ini.



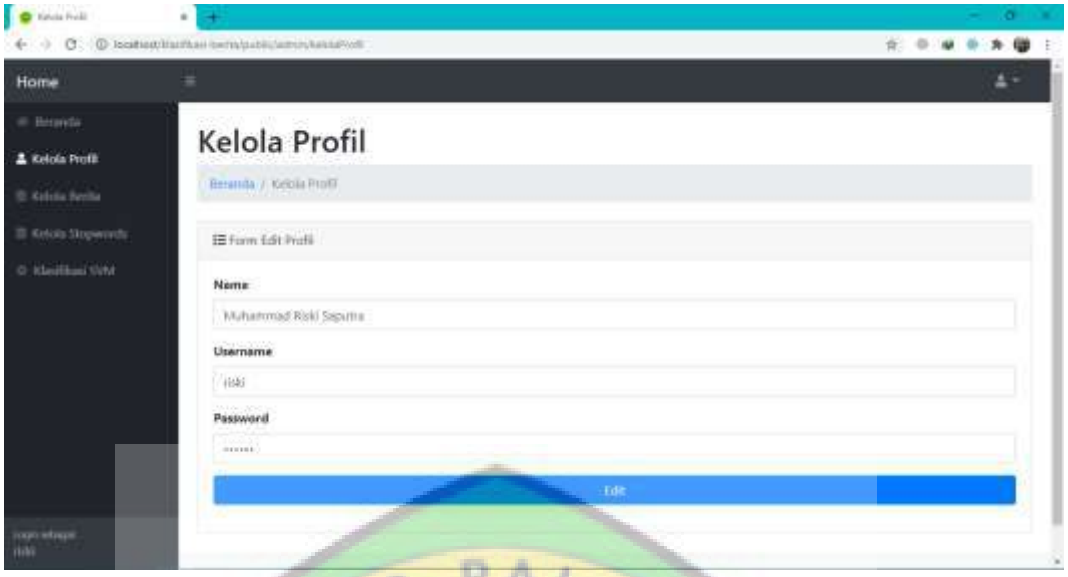
Gambar 4.1. Halaman Login

Gambar 4.1 merupakan halaman login yang akan ditampilkan pertama kali apabila pengguna ingin masuk sebagai admin pada website Sistem Klasifikasi Berita *Online*. Jika pengguna bukan Admin, maka klik pada link “Bukan Admin?” agar dapat menuju halaman Editor. Jika pengguna ingin masuk ke dalam sistem sebagai Admin, maka pengguna diharuskan untuk menginputkan *username* dan *password* yang valid.

4.1.1 *User Interface* Admin

Gambar 4.2. Halaman Beranda Admin

Gambar 4.2 merupakan implementasi antar muka Halaman Beranda Admin. Halaman ini adalah halaman pertama yang ditampilkan setelah pengguna berhasil melakukan login ke dalam sistem sebagai Admin. Pada halaman ini, akan ditampilkan informasi mengenai total data stopwords dan data berita per kategorinya.

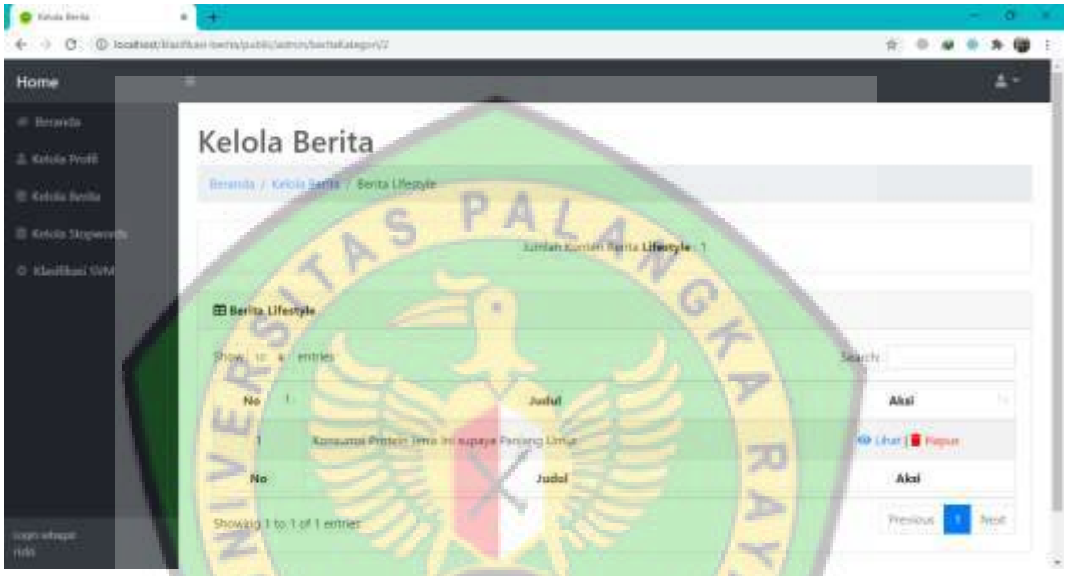


Gambar 4.3. Halaman Kelola Profil

Gambar 4.3 merupakan implementasi antar muka Halaman Kelola Profil. Pada halaman ini, admin dapat melakukan kelola profil untuk mengubah data akun admin seperti data nama, *username* dan *password*.

Gambar 4.4. Halaman Kelola Berita

Gambar 4.4 merupakan implementasi antar muka Halaman Kelola Berita. Halaman ini akan menampilkan data kategori konten berita yang terdapat didalam sistem. Pada halaman ini admin memilih kategori konten berita yang akan dikelola.

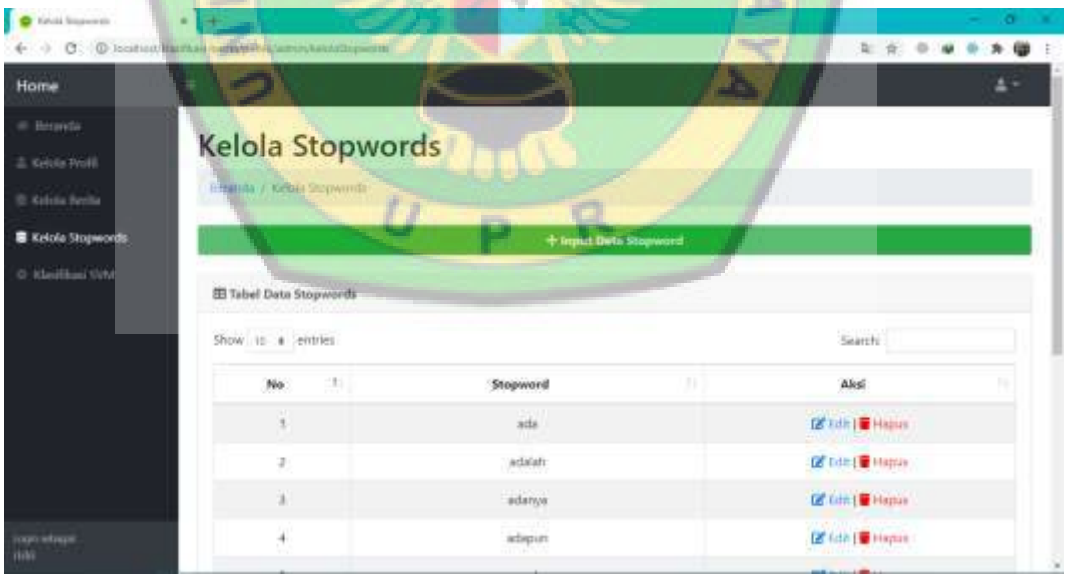


Gambar 4.5. Halaman Berita Kategori

Gambar 4.5 merupakan implementasi antar muka Halaman Berita Kategori. Halaman ini akan menampilkan data berita berupa judul berita berdasarkan kategori konten berita yang telah dipilih oleh admin untuk dikelola. Pada halaman ini admin dapat melihat dan menghapus berita.

Gambar 4.6. Halaman Isi Berita

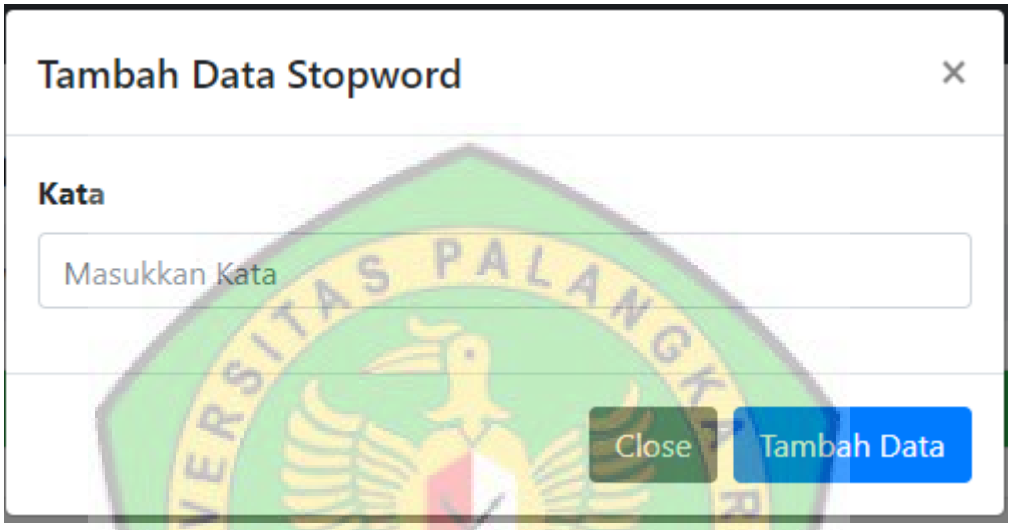
Gambar 4.6 merupakan implementasi antar muka Halaman Isi Berita. Halaman ini akan menampilkan isi dari konten berita yang dipilih untuk dilihat.



Gambar 4.7. Halaman Kelola Stopwords

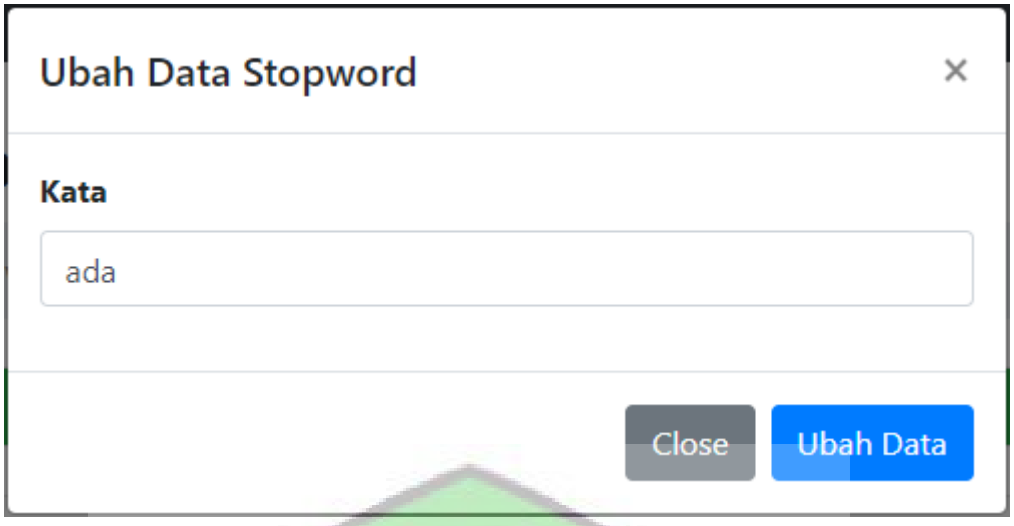
Gambar 4.7 merupakan implementasi antar muka Halaman Kelola Stopwords. Halaman ini akan menampilkan data stopwords yang terdapat didalam

database sistem. Pada halaman ini, admin dapat melakukan kelola seperti menginputkan data stopwords baru, mengedit data dan menghapus data stopwords.



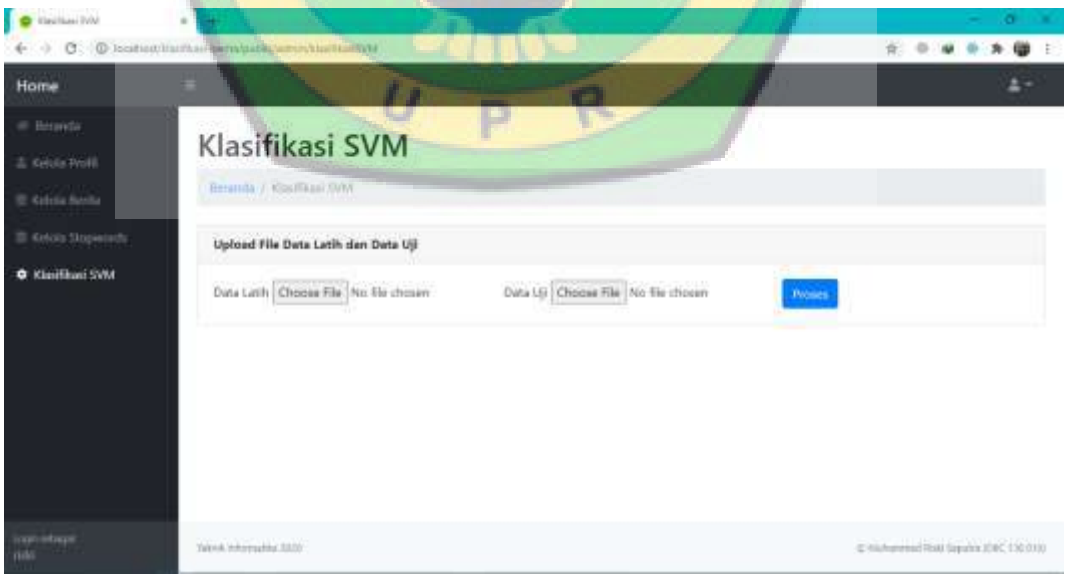
Gambar 4.8. Halaman Input Data Stopwords

Gambar 4.8 merupakan implementasi antar muka Halaman Input Data Stopwords. Pada halaman ini, admin menginputkan data stopwords yang ingin ditambahkan kedalam *database* sistem pada form input data stopwords.



Gambar 4.9. Halaman Edit Data Stopwords

Gambar 4.9 merupakan implementasi antar muka Halaman Edit Data Stopwords. Halaman ini akan menampilkan form edit data stopwords untuk diisikan data kata yang ingin diubah pada *database* sistem.



Gambar 4.10. Halaman Klasifikasi SVM

Gambar 4.10 merupakan implementasi antar muka Halaman Klasifikasi SVM. Pada halaman ini, untuk melakukan proses klasifikasi SVM admin menginputkan file data latih sebagai proses *Train* SVM dan file data uji sebagai proses *Test* SVM.

Gambar 4.11. Proses *Remove Punctuation*

Gambar 4.11 adalah proses *remove punctuation* yang dilakukan untuk menghilangkan tanda baca maupun simbol-simbol pada data berita.

Gambar 4.12. Proses *Case Folding*

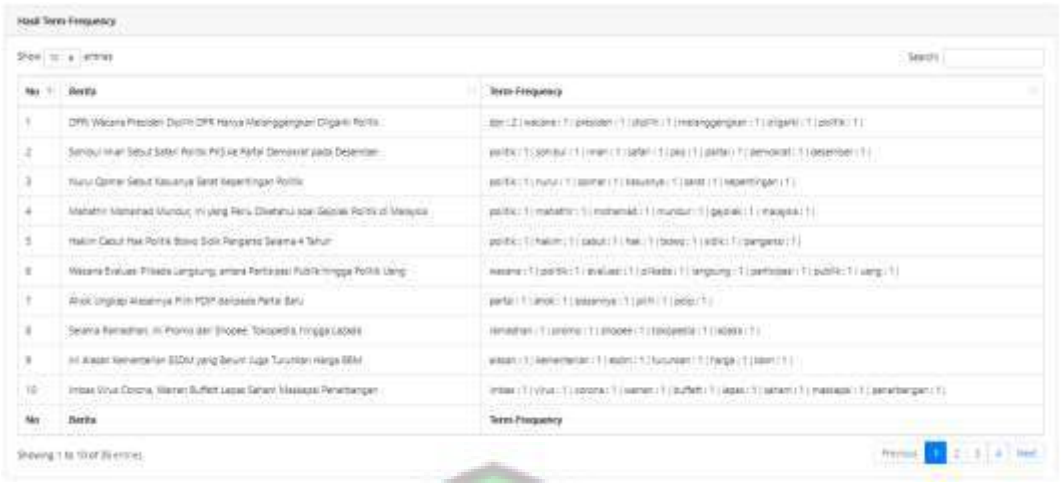
Gambar 4.12 adalah proses *case folding* yang dilakukan untuk mengubah huruf pada teks berita menjadi huruf kecil.

Gambar 4.13. Proses Tokenisasi

Gambar 4.13 adalah proses tokenisasi yang dilakukan untuk memilah kalimat dalam setiap berita menjadi kata atau token.

Gambar 4.14. Proses *Stopwords Filtering*

Gambar 4.14 adalah proses *stopwords filtering* yang dilakukan untuk menghilangkan kata atau token yang dianggap kurang penting.



Gambar 4.15. Proses *Term Frequency*

Gambar 4.15 adalah proses *term frequency* yang dilakukan untuk menghitung jumlah kemunculan sebuah *term* atau kata pada setiap dokumen berita.

Gambar 4.16. Proses TF-IDF

Gambar 4.16 adalah proses *term frequency-inverse document frequency* yang dilakukan untuk menghitung bobot sebuah *term* atau kata pada setiap dokumen berita.

Gambar 4.17. Proses Validasi Model

Gambar 4.17 adalah proses validasi model yang dilakukan untuk mengestimasikan kevalidan model klasifikasi yang dibuat.

Gambar 4.18. Hasil Train dan Test SVM

Matriks Konfusi

Fij		Kelas Hasil Prediksi (j)				
		Lifestyle	Olahraga	Politik	Ekonomi	Teknologi
Kelas Asli (i)	Lifestyle	3	0	0	0	0
	Olahraga	1	1	0	1	0
	Politik	0	0	3	0	0
	Ekonomi	3	0	0	0	0
	Teknologi	2	0	0	1	0

Gambar 4.18. Hasil Train dan Test SVM (Lanjutan)

Gambar 4.18 adalah hasil yang ditampilkan setelah proses *train* dan *test* SVM selesai dilakukan. Pada bagian ini akan ditampilkan data hasil prediksi, matriks konfusi, waktu pelatihan, serta tingkat akurasi dari model klasifikasi.

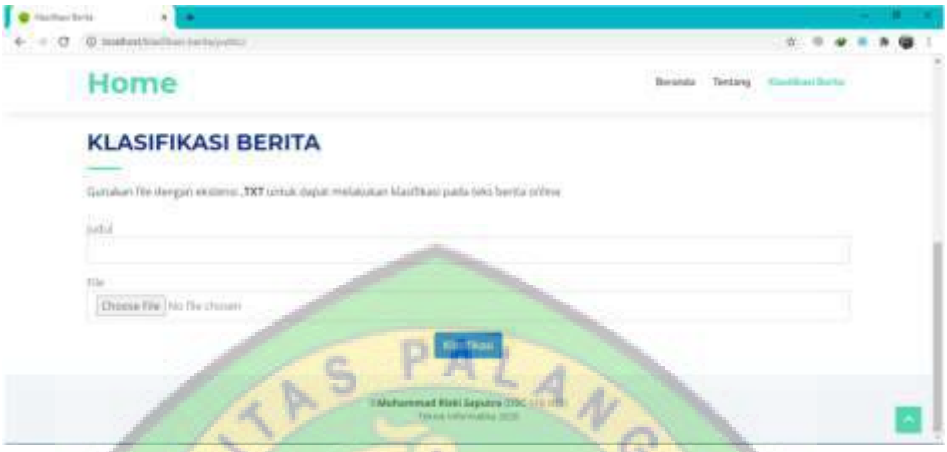
4.1.2 *User Interface* Editor

Gambar 4.19. Halaman Beranda Editor

Gambar 4.19 merupakan implementasi antar muka Halaman Beranda Editor. Halaman ini merupakan halaman pertama yang ditampilkan saat pengguna yang masuk ke sistem bukan seorang admin.

Gambar 4.20. Halaman Tentang

Gambar 4.20 merupakan implementasi antar muka Halaman Tentang. Pada halaman ini pengguna dapat melihat informasi singkat mengenai sistem klasifikasi berita *online*.



Gambar 4.21. Halaman Klasifikasi Berita

Gambar 4.21 merupakan implementasi antar muka Halaman Klasifikasi Berita. Pada halaman ini editor dapat mengklasifikasikan kategori berita dengan menginputkan judul dan file berita.

Gambar 4.22. Halaman Hasil Klasifikasi Berita

Gambar 4.22 merupakan implementasi antar muka Halaman Hasil Klasifikasi Berita. Halaman ini akan menampilkan hasil prediksi dari klasifikasi kategori berita yang diinputkan oleh editor sebelumnya.

4.2 Pengujian Sistem

4.2.1 Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas yang dilakukan menggunakan metode *Blackbox Testing*, dimana pengujian tersebut meliputi output yang dihasilkan berdasarkan input dan kondisi eksekusi yang dipilih untuk menampilkan output atau proses setiap tahapannya. Adapun pengujian fungsionalitas terbagi menjadi *Blacbox Testing* Admin dan *Blackbox Testing* Editor.

4.2.1.1 *Blackbox Testing* Admin

a. Login

Tabel 4.1. Pengujian Halaman Login

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Menginputkan *username* dan *password* yang salah	Tampilkan konfirmasi login gagal dan kembali ke halaman login	Menampilkan konfirmasi login gagal dan kembali ke halaman login	OK

Tabel 4.1. Pengujian Halaman Login (Lanjutan)

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
2.	Menginputkan *username* dan *password* yang benar	Login berhasil dan tampilkan halaman beranda admin	Login berhasil dan menampilkan halaman beranda admin	OK

b. Beranda Admin

Tabel 4.2. Pengujian Halaman Beranda Admin

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Sistem menampilkan informasi nama admin yang berhasil login	Tampilkan informasi nama admin yang berhasil login	Menampilkan informasi nama admin yang berhasil login	OK
2.	Sistem menampilkan informasi total data stopwords	Tampilkan informasi total data stopwords	Menampilkan informasi total data stopwords	OK

Tabel 4.2. Pengujian Halaman Beranda Admin (Lanjutan)

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
3.	Sistem menampilkan informasi total data berita kategori lifestyle	Tampilkan informasi total data berita kategori lifestyle	Menampilkan informasi total data berita kategori lifestyle	OK
4.	Sistem menampilkan informasi total data berita kategori olahraga	Tampilkan informasi total data berita kategori olahraga	Menampilkan informasi total data berita kategori olahraga	OK
5.	Sistem menampilkan informasi total data berita kategori politik	Tampilkan informasi total data berita kategori politik	Menampilkan informasi total data berita kategori politik	OK
6.	Sistem menampilkan informasi total data berita kategori ekonomi	Tampilkan informasi total data berita kategori ekonomi	Menampilkan informasi total data berita kategori ekonomi	OK

Tabel 4.2. Pengujian Halaman Beranda Admin (Lanjutan)

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
7.	Sistem menampilkan informasi total data berita kategori teknologi	Tampilkan informasi total data berita kategori teknologi	Menampilkan informasi total data berita kategori teknologi	OK
8.	Sistem menampilkan informasi berita per kategori dengan bar chart	Tampilkan informasi berita per kategori dengan bar chart	Menampilkan informasi berita per kategori dengan bar chart	OK

c. Kelola Profil

Tabel 4.3. Pengujian Halaman Kelola Profil

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Sistem menampilkan informasi data awal admin	Tampilkan informasi data admin sebelum diubah	Menampilkan informasi data admin sebelum diubah	OK

Tabel 4.3. Pengujian Halaman Kelola Profil (Lanjutan)

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
2.	Admin menginputkan data nama, *username* dan *password* yang ingin diubah	Data nama, *username* dan *password* berhasi di ubah	Data nama, *username* dan *password* berhasi di ubah	OK

d. Kelola Berita

Tabel 4.4. Pengujian Halaman Kelola Berita

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Sistem menampilkan data daftar kategori berita	Tampilkan data daftar kategori berita	Menampilkan data daftar kategori berita	OK
2.	Admin mengklik lihat pada kategori berita yang ingin dikelola lebih	Tampilkan halaman berita kategori berdasarkan kategori yang	Menampilkan halaman berita kategori berdasarkan kategori yang	OK

Tabel 4.4. Pengujian Halaman Kelola Berita (Lanjutan)

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
	lanjut	dipilih	dipilih	
3.	Admin mengklik pada aksi hapus berita	Berita yang dipilih terhapus	Berita yang dipilih terhapus	OK
4.	Admin mengklik pada aksi lihat berita	Tampilkan halaman isi berita	Menampilkan halaman isi berita	OK

e. Kelola Stopwords

Tabel 4.5. Pengujian Halaman Kelola Stopwords

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Sistem menampilkan data stopwords yang terdapat didalam *database*	Tampilkan data stopwords yang terdapat didalam *database*	Menampilkan data stopwords yang terdapat didalam *database*	OK
2.	Admin mengklik pada tombol input data stopwords	Tampilkan halaman input data stopwords	Menampilkan halaman input data stopwords	OK

Tabel 4.5. Pengujian Halaman Kelola Stopwords (Lanjutan)

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
3.	Admin menginputkan data stopwords baru pada halaman input data stopwords	Data stopwords berhasil di input	Data stopwords berhasil diinput	OK
4.	Admin mengklik pada edit data stopwords	Tampilkan halaman edit data stopwords	Menampilkan halaman edit data stopwords	OK
5.	Admin mengedit data stopwords pada halaman edit data stopwords	Data stopwords berhasil diedit	Data stopwords berhasil diedit	OK
6.	Admin mengklik pada hapus data stopwords	Data stopwords berhasil dihapus	Data stopwords berhasil dihapus	OK

f. Klasifikasi SVM

Tabel 4.6. Pengujian Halaman Klasifikasi SVM

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Admin menginputkan file data latih dan data uji dengan ekstensi file bukan .csv	File data latih dan data uji gagal di upload dan gagal di lakukan proses klasifikasi SVM	File data latih dan data uji gagal di upload dan gagal di lakukan proses klasifikasi SVM	OK
2.	Admin menginputkan file data latih dan data uji dengan ekstensi file .csv	File berhasil diupload dan dilakukan proses klasifikasi SVM serta ditampilkan hasil proses klasifikasi SVM	File berhasil diupload dan dilakukan proses klasifikasi SVM serta menampilkan hasil proses klasifikasi SVM	OK

4.2.1.2 *Blackbox Testing* Editor

a. Beranda Editor

Tabel 4.7. Pengujian Halaman Beranda Editor

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Editor mengklik pada tombol Klasifikasi Berita	Mengarahkan editor menuju bagian halaman Klasifikasi Berita	Editor diarahkan menuju bagian halaman Klasifikasi Berita	OK

b. Tentang

Tabel 4.8. Pengujian Halaman Tentang

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Editor mengklik pada menu Tentang	Mengarahkan editor menuju bagian halaman Tentang	Editor diarahkan menuju bagian halaman Tentang	OK

c. Klasifikasi Berita

Tabel 4.9. Pengujian Halaman Klasifikasi Berita

No	Aksi	Hasil Yang Diharapkan	Hasil Keluaran	Kesimpulan
1.	Editor mengklik pada menu Klasifikasi Berita	Mengarahkan editor menuju bagian halaman Klasifikasi Berita	Editor diarahkan menuju bagian halaman Klasifikasi Berita	OK
2.	Editor menginputkan data judul dan file berita berekstensi bukan .txt	Proses prediksi klasifikasi berita gagal	Proses prediksi klasifikasi berita gagal	OK
3.	Editor menginputkan data judul dan file berita berekstensi .txt	Proses prediksi klasifikasi berita berhasil dan ditampilkan hasil prediksi klasifikasi berita	Proses prediksi klasifikasi berita berhasil dan menampilkan hasil prediksi klasifikasi berita	OK

4.2.2 Pengujian Performa Klasifikasi

Pada pengujian sistem klasifikasi ini, untuk menguji performa dari sistem klasifikasi yang dibangun, maka digunakan metode *Accuracy* untuk mengetahui tingkat keakuratan sistem klasifikasi dalam mengklasifikasian berita *online* sesuai label kelas kategori berita yang tepat. Dalam pengujian ini juga akan dilakukan validasi terhadap model klasifikasi yang telah dibangun dengan menggunakan metode *K-Fold Cross Validation* untuk mengestimasikan tingkat generalisasi model klasifikasi yang dibuat. Pengujian performa klasifikasi dilakukan dengan beberapa skenario komposisi data latih dan data uji sebagai berikut.

Tabel 4.10. Pembagian Data Latih dan Data Uji

Komposisi Data		Jumlah Data	
Data Latih	Data Uji	Data Latih	Data Uji
50%	50%	500	500
60%	40%	600	400
70%	30%	700	300
80%	20%	800	200

Pada tabel 4.10 dijelaskan dari hasil perolehan dataset berita *online* yang berjumlah 1000 data tersebut dibagi menjadi beberapa kombinasi data latih dan data uji sebanyak 4 kali dengan mekanisme pembagian yang berbeda. Sebagai contoh mekanisme pengujian pada percobaan komposisi jumlah data latih dan data uji yang digunakan adalah 500:500, pengujian sistem klasifikasi yang telah dibuat dengan 500 data uji berita *online* tersebut dibagi menjadi 5 kategori,

sehingga masing-masing kategori akan diujikan 100 data. Berikut merupakan hasil matriks konfusi dari beberapa skenario pembagian data latih dan data uji.

1. Hasil Pengujian Skenario Pertama

Berikut merupakan hasil pengujian dengan menggunakan data latih sebanyak 500 data dan data uji sebanyak 500 data, yang disajikan menggunakan matriks konfusi dapat diihat pada Tabel 4.11 berikut.

Tabel 4.11. Matriks Konfusi Skenario Pertama

F_{ij}		Kelas Hasil Prediksi (j)				
		Lifestyle	Olahraga	Politik	Ekonomi	Teknologi
Kelas Asli (i)	Lifestyle	91	2	1	4	2
	Olahraga	4	92	2	1	1
	Politik	1	1	98	0	0
	Ekonomi	2	4	2	89	3
	Teknologi	5	3	3	1	88

Berikut perhitungan nilai akurasi dengan melihat matriks konfusi dan menggunakan persamaan 2.23:

$$Akurasi = \frac{458}{500} \times 100\% = 91.60\%$$

2. Hasil Pengujian Skenario Kedua

Berikut merupakan hasil pengujian dengan menggunakan data latih sebanyak 600 data dan data uji sebanyak 400 data, yang disajikan menggunakan matriks konfusi dapat diihat pada Tabel 4.12 berikut.

Tabel 4.12. Matriks Konfusi Skenario Kedua

F_{ij}		Kelas Hasil Prediksi (j)				
		Lifestyle	Olahraga	Politik	Ekonomi	Teknologi
Kelas Asli (i)	Lifestyle	70	2	1	5	2
	Olahraga	2	75	1	1	1
	Politik	0	1	79	0	0
	Ekonomi	2	2	2	73	1
	Teknologi	2	2	3	0	73

Berikut perhitungan nilai akurasi dengan melihat matriks konfusi dan menggunakan persamaan 2.23:

$$Akurasi = \frac{370}{400} \times 100\% = 92.50\%$$

3. Hasil Pengujian Skenario Ketiga

Berikut merupakan hasil pengujian dengan menggunakan data latih sebanyak 700 data dan data uji sebanyak 300 data, yang disajikan menggunakan matriks konfusi dapat diihat pada Tabel 4.13 berikut.

Tabel 4.13. Matriks Konfusi Skenario Ketiga

F_{ij}		Kelas Hasil Prediksi (j)				
		Lifestyle	Olahraga	Politik	Ekonomi	Teknologi
Kelas Asli (i)	Lifestyle	51	4	0	4	1
	Olahraga	1	54	1	3	1
	Politik	0	0	60	0	0
	Ekonomi	2	1	0	56	1
	Teknologi	0	1	1	1	57

Berikut perhitungan nilai akurasi dengan melihat matriks konfusi dan menggunakan persamaan 2.23:

$$Akurasi = \frac{278}{300} \times 100\% = 92.67\%$$

4. Hasil Pengujian Skenario Keempat

Berikut merupakan hasil pengujian dengan menggunakan data latih sebanyak 800 data dan data uji sebanyak 200 data, yang disajikan menggunakan matriks konfusi dapat diihat pada Tabel 4.14 berikut.

Tabel 4.14. Matriks Konfusi Skenario Keempat

F_{ij}		Kelas Hasil Prediksi (j)				
		Lifestyle	Olahraga	Politik	Ekonomi	Teknologi
Kelas Asli (i)	Lifestyle	36	1	0	2	1
	Olahraga	1	37	1	1	0
	Politik	0	0	40	0	0
	Ekonomi	1	0	0	38	1
	Teknologi	0	1	0	0	39

Berikut perhitungan nilai akurasi dengan melihat matriks konfusi dan menggunakan persamaan 2.23:

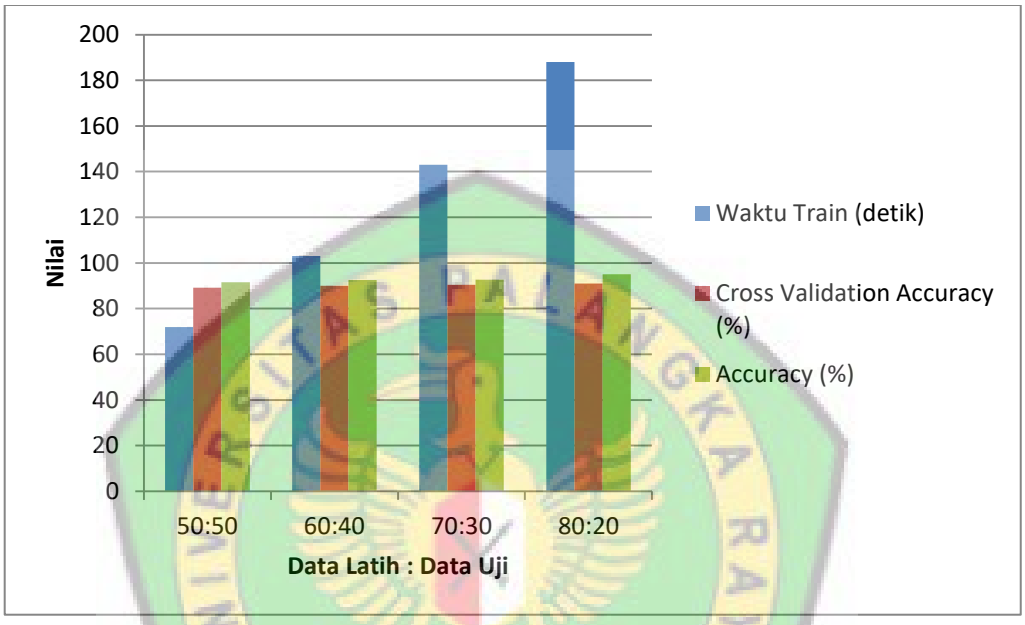
$$Akurasi = \frac{190}{200} \times 100\% = 95\%$$

Dari beberapa sekenario pengujian, berikut merupakan hasil keseluruhan pengujian performa sistem klasifikasi yang telah dilakukan.

Tabel 4.15. Hasil Pengujian Performa Klasifikasi

Komposisi Data (%)	Waktu Train (detik)	Cross Validation Accuracy (%)	Accuracy (%)
50:50	72	89.20	91.60
60:40	103	90	92.50
70:30	143	90.43	92.67
80:20	188	91	95

Tabel 4.15 menunjukkan hasil pengujian performa sistem klasifikasi dengan beberapa komposisi data latih dan data uji. Gambar 4.23 menunjukkan diagram hasil pengujian berdasarkan Tabel 4.15.



Gambar 4.23. Diagram Hasil Pengujian Performa Klasifikasi

Dari hasil pengujian performa sistem klasifikasi diketahui bahwa komposisi data latih dan data uji yang digunakan dalam membangun model klasifikasi mempengaruhi kinerja model klasifikasi. Ketika pengujian dilakukan dengan komposisi data latih dan data uji yang berbeda akan memberikan nilai yang berbeda pula, baik dari segi waktu pelatihan/*train* maupun tingkat akurasi. Penyebab perbedaan yang terjadi dikarenakan hasil yang didapatkan pada saat merepresentasikan fitur data sebagai atribut yang menggambarkan sebuah objek, dalam hal ini kategori konten berita *online*.

Jumlah fitur mempengaruhi dimensi data yang akan diperhitungkan dalam metode klasifikasi SVM. Semakin banyak jumlah data latih yang digunakan, maka akan berimbas pada waktu pelatihan/*training* yang semakin lama, namun dapat terlihat pada setiap kenaikan jumlah data latih yang digunakan dalam melakukan pelatihan juga terjadi peningkatan nilai akurasi sistem, baik dari segi validasi model yang dibangun dan tingkat akurasi model terhadap data uji.

Nilai terbaik pada pengujian performa sistem klasifikasi didapatkan pada 80% komposisi data latih dan 20% komposisi data uji. Tingkat akurasi validasi sebesar 91% menunjukkan bahwa pada proses *training* (pelatihan), model yang dibangun telah dimodelkan dengan tepat, sehingga tingkat generalisasi yang dimiliki model terhadap data uji yang diprediksi (*testing*) memiliki tingkat akurasi yang baik pula, yaitu sebesar 95%. Berdasarkan hasil pengujian performa sistem klasifikasi, model yang dipilih untuk diimplementasikan dalam sistem klasifikasi berita *online* adalah model yang dibangun dengan komposisi 80% data latih dan 20% data uji.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian pada Tugas Akhir dengan judul "Klasifikasi Berita Online Berbahasa Indonesia Menggunakan Algoritma *Support Vector Machine*", dapat ditarik kesimpulan sebagai berikut.

Website Sistem Klasifikasi Berita *Online* menggunakan metodologi pengembangan perangkat lunak *Waterfall*. Sistem Klasifikasi Berita *Online* menggunakan algoritma *Support Vector Machine* (SVM) dengan Kernel *Linear*. Klasifikasi SVM bekerja dengan menemukan *hyperplane* optimum yang memisahkan setiap kategori kelas, dimana *hyperplane* sebagai fungsi pemisah akan digunakan untuk mengetahui kelas data baru berdasarkan bidang pemisah/*hyperplane* yang ditemukan pada proses pelatihan. Sistem klasifikasi menggunakan 1000 data berita *online* berbahasa Indonesia, terbagi menjadi data latih dan data uji dengan beberapa skenario pengujian. Pengujian fungsionalitas sistem dilakukan dengan menggunakan metode *Blackbox Testing* menunjukkan bahwa sistem sudah dapat berjalan sesuai dengan hasil yang diharapkan, sedangkan pengujian performa sistem klasifikasi dilakukan dengan menggunakan metode *Accuracy*. Nilai *accuracy* tertinggi diperoleh pada skenario pengujian ke-4 sebesar 95% dengan komposisi data latih 80% dan data uji 20%. Berdasarkan hasil implementasi dan pengujian performa sistem klasifikasi pada dokumen berita *online*

berbahasa Indonesia, sistem telah dapat memenuhi tujuan dalam melakukan klasifikasi pada dokumen berita *online* berbahasa Indonesia.

5.2 Saran

Adapun saran yang dapat diberikan untuk perbaikan atau peningkatan peneltian mengenai sistem klasifikasi teks dengan menggunakan algoritma *Support Vector Machine* adalah sebagai berikut.

1. Sistem klasifikasi pada penelitian ini memiliki lima kategori konten berita. Pada penelitian selanjutnya diharapkan adanya penambahan kategori kelas agar diperoleh sistem klasifikasi berita yang lebih menyeluruh terhadap konten-konten berita lainnya.
2. Pada penelitian ini, tahap pra pemrosesan teks tidak melibatkan proses *stemming*. Diharapkan dengan menambahkan proses *stemming* dapat memberikan performa sistem klasifikasi yang lebih baik.
3. Menerapkan metode kernel lainnya (Polynomial, Gaussian RBF) dalam algoritma *Support Vector Machine*, sebagai bahan perbandingan dalam penelitian lebih lanjut.

DAFTAR PUSTAKA

Cangara, H. 2014. *Komunikasi Politik: Konsep, Teori, dan Strategi*. Jakarta: Rajawali Pers.

Caporaso, J. A., & Levine, D. P. 2008. *Teori-Teori Ekonomi Politik*. Yogyakarta: Pustaka Pelajar.

Fatmawati, F., & Affandes, M. 2018. *Klasifikasi Keluhan Menggunakan Metode Support Vector Machine (SVM) Pada Akun Facebook Group iRaise Helpdesk*. Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi, 3(1), 24-30.

Faruqi, Zahid. 2015. "*Apa sih Politik itu?"*. https://www.kompasiana.com/zahidfaruqi/552a34daf17e613c6cd623ea/apa-sih-politik-itu. Diakses pada hari Selasa, 17 Maret 2020.

Febriani, E.S. 2017. *Klasifikasi Konten Berita Surat Kabar Berdasarkan Judul dengan Text Mining Menggunakan Metode Naïve Bayes.* Program Studi Teknik Informatika, Fakultas Teknik Universitas Nusantara PGRI Kediri.

Fitri, Meisya. 2013. *Perancangan Sistem Temu Balik Informasi Dengan Metode Pembobotan Kombinasi Tf-Idf Untuk Pencarian Dokumen Berbahasa Indonesia*. Universitas Tanjungpura : Semarang.

Giriwijoyo, H. S., & Sidik, D. Z. 2012. *Ilmu Faal Olahraga (Fisiologi Olahraga)*. Bandung: PT. Remaja Rosdakarya.

Haryalesmasna, Devid. 2016. *ID-Stopwords*. https://github.com/masdevid/ID-Stopwords (2020).

Hendariningrum, R., & Susilo, M. E. 2014. *Fashion dan Gaya Hidup: Identitas dan Komunikasi*. Jurnal Ilmu Komunikasi, 6(1).

Ingersoll, G. S., Morton, T. S., & Farris, A. L. 2013. *Taming Text: How to Find, Organize, and Manipulate it*. Shelter Island: Manning.

Irfa, A.A., dkk. 2018. *Klasifikasi Topik Berita Berbahasa Indonesia menggunakan k-Nearest Neighbor*. Fakultas Informatika, Universitas Telkom Bandung.

Joachims, T. 1998. *Text categorization with support vector machines: Learning with many relevant features*. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

Khairani, R., & Saleh, R. 2020. *Kepentingan Ekonomi-Politik Media dalam Pemberitaan pada Media Cetak Serambi Indonesia*. Jurnal Ilmiah Mahasiswa Fakultas Ilmu Sosial & Ilmu Politik, 5(1).

Kurniawan, Aris. 2019. *17 Definisi, Pengertian Teknologi Menurut Para Ahli Dan Perkembangannya*. https://www.gurupendidikan.co.id/pengertian-teknologi/. Diakses pada hari Jumat, 1 Mei 2020.

Luqyana, W. A., Cholissodin, I., & Perdana, R. S. 2018. *Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 964X.

Mahmudy, W. F., & Widodo, A. W. 2015. *Klasifikasi Artikel Berita Secara Otomatis Menggunakan Metode Naive Bayes Classifier Yang Dimodifikasi*. TEKNO, 21(1).

Muslimin, Khoirul. 2019. *Jurnalistik Dasar : Jurus Jitu Menulis Berita, Feature Biografi, Artikel Populer, dan Editorial*. Yogyakarta: UNISNU.

Mustika, A., dan Affandes, M. 2015. *Penerapan Metode Support Vector Machine Dalam Klasifikasi Sentimen Tweet Public Figure*. SENTRA (Seminar Teknologi dan Rekayasa) (No. 1).

Prasetyo, Eko. 2014. *Data Mining: Mengolah Data menjadi Informasi menggunakan MATLAB*. Yogyakarta: Penerbit Andi.

Pratama, D.H. 2013. *Implementasi Support Vector Machine (SVM) Untuk Klasifikasi Dokumen*. Skripsi. Fakultas Matematika dan Ilmu Pengetahuan Alam. Departemen Ilmu Komputer. Institut Pertanian Bogor. Bogor.

Prilianti, K. R., & Wijaya, H. 2014. *Aplikasi Text Mining untuk Automasi Penentuan Tren Topik Skripsi dengan Metode K-Means Clustering*. Jurnal Cybermatika, 2(1).

Purnamawan, I. K. 2015. *Support Vector Machine Pada Information Retrieval*. Jurnal Pendidikan Teknologi dan Kejuruan, 12(2), 139-146.

Putra, S. J., Gunawan, M. N., & Suryatno, A. (2018, May). *Tokenization and N-Gram for Indexing Indonesian Translation of the Quran*. In 2018 6th International Conference on Information and Communication Technology (ICoICT) (pp. 158-161). IEEE.

Ratna, Kusuma. 2014. *Pengertian PHP dan MySQL*. Tangerang: Informasi.

Rosa, A.S dan M. Shalahuddin. 2013. *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung.

Setiawan, Didik. 2017. *Buku Sakti Pemrograman Web*. Yogyakarta: START UP.

Sumadiria, A.H. 2014. *Jurnalistik Indonesia: Menulis Berita dan Feature, Panduan Praktis Jurnalis Profesional*. Bandung: Simbiosa Rekatama Media.

Sumathy, K. L., & Chidambaram, M. 2013. *Text Mining: Concepts, Applications, Tools and Issues-An Overview*. International Journal of Computer Applications, 80(4).

Suyanto, D. 2017. *Data Mining untuk Klasifikasi dan Klasterisasi Data*. Bandung: Informatika Bandung.

Utomo, Wiliam Putra. 2019. *Indonesia Millenial Report 2019*. IDN Media: Jakarta.

Uysal, A. K., & Gunal, S. 2014. *The impact of preprocessing on text classification. Information Processing & Management*, 50(1), 104-112.

Wahyudi, R., & Aristantia, A. D. 2017. *Aplikasi Pengolahan Data Pelanggaran Siswa pada SMK Yayasan Pendidikan Teknologi 1 Purbalingga Terintegrasi dengan SMS Gateway*. Jurnal Telematika Vol, 10(2).

Wang, F., Xu, T., Tang, T., Zhou, M., & Wang, H. 2016. *Bilevel feature extraction-based text mining for fault diagnosis of railway systems*. IEEE transactions on intelligent transportation systems, 18(1), 49-58.