Aplikasi Penentu Subyek Skripsi Menggunakan Metode *Support Vector Machine*

Artono Ivan Chandra[1], Yulia[2], Rudy Adipranata[3]
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131. Surabaya 60236
Telp (031) 2983455, Fax. (031) 8417658
artonoivan@gmail.com[1], yulia@petra.ac.id[2], rudya@petra.ac.id[3]

ABSTRAK

Skripsi adalah tugas yang diberikan universitas kepada mahasiswa sebagai penilaian akhir atas proses belajar yang sudah ditempuh selama beberapa semester. Setelah menyelesaikan skripsi, mahasiswa menyerahkan hasil penelitiannya kepada kampus sebagai koleksi skripsi. Pada Universitas Kristen Petra setiap skripsi yang terkumpul diberikan subyek sebagai kategori skripsi tersebut. Namun pemberian subyek ini masih manual, sehingga dibutuhkan sistem yang dapat membantu menentukan subyek skripsi.

Sistem yang dilengkapi fitur *text mining* akan membantu pihak perpustakaan dalam menentukan subyek skripsi. Langkah yang dilakukan adalah *preprocessing* yang terdiri *punctual removal, stopword removal,* dan *stemming.* Lalu proses ekstrak data teks menjadi angka menggunakan TF-IDF. Selanjutnya data akan dilatih menggunakan metode *Support Vector Machine* yang nantinya menghasilkan model dan digunakan untuk memprediksi subyek dari teks *input*. Data yang dilatih adalah data judul dan abstrak skripsi yang sudah ada.

Hasil dari penelitian yang dilakukan menunjukan dalam pembangunan model klasifikasi SVM dibutuhkan parameter TF-IDF max_df 1, n-gram (1,2), smooth_idf dan sublinear_tf *true*, kernel SVM linear dengan C 100 pada judul skripsi dan max_df 0.25, n-gram (1,1), smooth_idf dan sublinear_tf *false*, kernel SVM rbf dengan C 100 dan gamma 0.01 pada abstrak skripsi. Baik judul maupun abstrak skripsi membutuhkan *preprocessing, resample,* dan normalisasi l2.

Kata Kunci: skripsi, teks, SVM

ABSTRACT

Thesis is a task given by the university to students as a final assessment of the learning process that has been taken for several semesters. After completing the thesis, students submit their research results to the campus as a thesis collection. At Petra Christian University, every thesis collected is given a subject as the thesis category. However, giving this subject is still manual, so we need a system that can help determine the subject of the thesis.

The system that is equipped with text mining features will help the library in determining the subject of the thesis. The steps taken are preprocessing consisting of punctual removal, stopword removal, and stemming. Then the process of extracting text data into numbers using TF-IDF. Furthermore, the data will be trained using the Support Vector Machine method which will produce a model and be used to predict subjects from input text. The trained data is the title data and abstract of the existing thesis..

The results of the research conducted showed that in the construction of the SVM classification model the parameters TF-IDF max_df 1, n-gram (1,2), smooth_idf and sublinear_tf true, linear SVM kernel with C 100 for the thesis title and max_df 0.25, n-gram (1,1), smooth_idf and sublinear_tf false, rbf SVM kernel with C 100 and gamma 0.01 for the thesis abstract. Both the title and abstract of the thesis require preprocessing, resample, and l2 normalization.

Keywords: *thesis, text, SVM*

1. PENDAHULUAN

Dalam dunia perkuliahan terdapat pengujian atau tugas akhir yang digunakan untuk mengukur pencapaian seorang mahasiswa selama masa belajar di suatu universitas dalam jangka waktu tertentu. Dalam mencapai tugas akhir ada berbagai syarat yang harus dilakukan atau dicapai mahasiswa seperti jumlah sks lulus harus sesuai, telah menyelesaikan magang, lulus TOEFL dan lain sebagainya. Terdapat berbagai jenis tugas akhir untuk mahasiswa seperti magang dan yang paling sering kita dengar adalah skripsi. Skripsi sendiri menuntut mahasiswa untuk melakukan penelitian terhadap suatu topik tertentu yang digunakan untuk menyelesaikan suatu masalah.

Setelah selesai dalam menyusun skripsi, mahasiswa lalu memberikannya kepada perpustakaan universitas untuk menjadi koleksi skripsi agar dapat digunakan sebagai sumber ide atau topik untuk penelitian selanjutnya. Pada Universitas Kristen Petra, setiap skripsi yang dikumpulkan akan diberi label pada setiap data skripsi yang dinamakan subyek. Subyek ini akan menentukan jenis atau kategori pada setiap data skripsi yang terkumpul. Satu data skripsi dapat memiliki satu atau lebih subyek.

Dalam penentuan subyek ini kebanyakan masih dilakukan secara manual dan hal ini terkadang dapat membuat lama dalam pemrosesan dalam menyumpan data skripsi baru. Karena hal ini, dibutuhkan suatu sistem informasi yang dapat mengelola dan membantu menyelesaikan permasalahan yang ada. Sistem sendiri adalah jaringan kerja dari beberapa prosedur yang berkumpul bersama melakukan kegiatan [1], jadi sistem informasi adalah kumpulan informasi yang tertata rapi. Sistem informasi ini akan dilengkapi dengan fitur machine learning yang dapat membantu perpustakaan untuk menentukan subyek skripsi.

Text mining adalah cabang dari ilmu *machine learning* yang berguna untuk menambang (*mining*) sebuah data berupa teks, dan mecari beberapa kata yang mewakili isi dari teks tersebut, mecari

keterikatan, dan mengklasifikasikannya. [3] dengan menggunakan text mining maka akan dihasilkan pola tren dan ekstraksi informasi berguna yang terdapat pada suatu data teks.

Nantinya *text mining* dapat membantu perpustakaan untuk menentukan subyek skripsi baru dari hasil pembelajaran terhadap data skripsi sebelumnya. Klasifikasi akan menggunakan metode *Support Vector Machine* (SVM) yang merupakan salah satu metode terbaik dalam pengklasifikasian, prediksi, dan regresi [10]. SVM mampu bekerja pada dengan ruang kerja yang memiliki dimensi tinggi, sehingga SVM dapat membantu pengklasifikasian secara kompleks.

2. TINJAUAN STUDI

2.1 *Text Mining*

Text mining biasa disebut juga *Text Data Mining* (TDM) yaitu salah satu cabang ilmu *machine learning* yang dapat mengambil suatu informasi dalam suatu kumpulan data yang tidak teratur (unstructured) dalam bentuk teks. *Text Mining* adalah ilmu perluasan dari data mining dan *knowledge-discovery* in *database* (KDD) yang lebih umum dipakai dari pada data mining sendiri dikarenakan secara umum format penyimpanan pada database adalah dalam bentuk teks. Perbedaan *text mining* dengan data mining sendiri adalah data pada *text mining* tidak terstruktur atau tidak rapi, sedangkan dalam data mining data sudah tersusun dengan rapi.

Text mining sendiri memiliki beberapa tahapan untuk memproses data [11] tahapannya adalah:

a. Text Pre-processing

Membersihkan karakter-karakter atau kata dalam teks yang tidak berguna atau tidak memiliki informasi penting seperti kata imbuhan dan kata penghubung.

b. Text Transformation

Pengurangan dimensi kata pada dokumen, memecah dan menghitung setiap jumlah kata.

c. Feature Selecton

Tahap lanjutan dari transformasi teks, dimana tidak semua kata yang sudah dihapus mengandung informasi penting, sehingga harus dikurangi lagi dimensinya.

d. Pattern Discovery

Tahap penemuan pola atau informasi yang terdapat pada suatu dokumen teks yang telah diproses.

2.2 *Support Vector Machine*

Support Vector Machine (SVM) merupakan salah satu metode dari *machine learning* yang sering digunakan untuk memprediksi suatu data dalam hal klasifikasi maupun regresi [10]. Metode ini dikembangkan oleh Vladimir Vapnik pada tahun 1995 dan merupakan metode yang melakukan pendekatan *supervised learning*. *Supervised learning* adalah algoritma yang sering dipakai dalam klasifikasi daripada *unsupervised learning* dikarenakan memiliki data latih atau sering disebut data training yang digunakan untuk dipelajari agar dapat mengklasifikasikan atau memprediksi data baru ke dalam suatu kelas atau label tertentu.

Konsep atau cara kerja SVM adalah mencari garis pemisah terbaik untuk memisahkan kedua class yang berbeda. Garis pemisah ini biasa disebut dengan *hyperplane* yang berguna sebagai pemisah ruang vektor menjadi dua bagian menjadi dua class berbeda. Pada dasarnya SVM digunakan sebagai metode klasifikasi linear atau data yang telah tertata rapi, namun pada masalah nyata (*real world problem*) yang bersifat non-linear dimana data tidak tertata rapi dibutuhkan cara agar SVM dapat menyesuaikan. Kernel *trick* digunakan untuk memudahkan pembelajaran SVM. Beberapa macam kernel pada SVM adalah polynomial, RBF, linear. Dengan menggunakan kernel *trick* hanya cukup memahami fungsi kernel yang dipakai dan tidak perlu memahami wujud dan fungsi non-linear [4].

2.3 *Multiclass Classification*

Ada banyak metode klasifikasi yang ada seperti *Decision Tree*, *Nearest Neighbor*, dan lainya yang dapat digunakan untuk mengklasifikasikan data dalam beberapa kelas yang biasa disebut dengan *multiclass clasification*. Namun SVM hanya dapat melakukan klasifikasi biner atau dua kelas. Sementara berbagai permasalah nyata di kehidupan sehari-hari mempunyai banyak kelas, dan oleh karena itu ada beberapa metode tambahan untuk menyelesaikan *multiclass clasification* pada SVM. Beberapa diantara nya adalah *one vs all* dan *one vs one* [8].

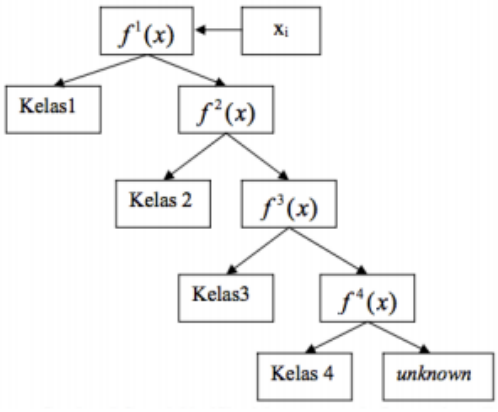
2.3.1 One vs All

Metode ini seing disebut juga *one vs rest*. Metode ini digunakan untuk membantu menentukan suatu data untuk diklasifikasikan dengan suatu kelas tertentu. Konsep dasar dari *one vs all* adalah melakukan pembagian tiap kelas, sebagai contoh misalkan ada suatu data set yang memiliki empat kelas yaitu kelas 1, 2, 3, dan 4.

Tabel 1. Contoh SVM Biner *One vs All*

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^3(x) = (w^3)x + b^3$

Seperti yang dijelaskan pada Tabel 1 suatu data yang akan diklasifikasikan akan dibandingan antara kelas 1 dan bukan kelas 1 (selain kelas 1 yaitu 2, 3, 4), apabila dia tidak masuk dalam kelas 1, maka data akan dibandingkan antara kelas 2 dan bukan kelas 2 begitu seterusnya.



Gambar 1. Contoh klasifikasi *one vs all*

Pada Gambar 1 dijelaskan alur perbandingan tiap-tiap kelas untuk klasifikasi *one vs all*.

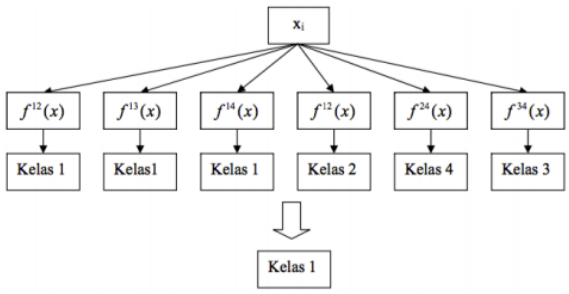
2.3.2 *One vs One*

Metode lain yang dapat digunakan SVM dalam *multiclass clasification* adalah *one vs one* dengan konsep yang mirip seperti *one vs all*. Pada dasarnya kedua metode ini hampir sama karena *one vs one* juga membandingkan data dengan dua kelas (secara biner) yang berbeda, namun perbedaan antara *one vs one* dan *one vs all* adalah perbedaan kelas yang dibandingkan. Seperti sebelum nya kita gunakan contoh *dataset* dengan empat kelas yaitu 1, 2, 3, 4.

Tabel 2. Contoh SVM Biner *One vs One*

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$

Dijelaskan pada Tabel 2 dimana *one vs one* akan membandingkan kelas 1 dengan kelas 2, lalu akan membandingkan kelas 1 dan kelas 3 lalu kelas 1 dan kelas 4 begitu seterusnya hingga seluruh kelas dibandingkan dengan kelas lainnya.



Gambar 2. Contoh klasifikasi *one vs one*

Pada Gambar 2 dapat dilihat bagaimana *one vs one* membandingkan satu per satu kelas dengan kelas lainya hingga semua selesai dibandingkan.

2.4 TF-IDF *Weighting*

TF adalah singkatan dari *Term Frequency* yang berguna untuk menghitung berapa banyak istilah kata yang ada pada suatu dokumen teks. Istilah kata pada dokumen teks ini bisa kita dapatkan dengan cara memecah-mecah teks menjadi n gabungan kata, hal ini biasa disebut dengan n-gram. Beberapa contoh n-gram adalah bigram dua gabungan kata, trigram tiga gabungan kata. Apa bila kita menggunakan bigram pada suatu kalimat "saya makan nasi goreng" maka hasilnya adalah "saya makan", "makan nasi", "nasi goreng".

IDF adalah singkatan dari *Inverse Document Frequency* yang berguna untuk menghitung *term* yang ada di keseluruhan dokumen. Semakin sedikit *term* yang ada di dokumen, maka bobot IDF akan semakin besar [6].

$$\mathbf{IDFJ=log(D/DFj)} \quad (1)$$

Penjelasan pada Rumus 1 adalah D jumlah keseluruhan dokumen dan Dfj adalah jumlah dokumen yang mengandung term pada keseluruhan dokumen. Sedangkan TF-IDF *weighting* adalah metode pembobotan kata atau *term* dengan menghitung nilai TF dan IDF nya.

$$\mathbf{Wdt=tftd*IDFtd} \quad (2)$$

Rumus 2 menjelaskan d adalah dokumen ke-d, t adalah term ke-t, W adalah bobot dari dokumen ke-d terhadap term ke-t, tf adalah banyaknya *term* yang dicari, dan IDF adalah nilai *inverse document frequency*.

2.5 *Imbalanced Dataset*

Dalam pembuatan *machine learning* sangat dibutuhkan data yang digunakan untuk di pelajari oleh mesin yang nantinya mesin memiliki kecerdasan untuk memprediksi data baru. Namun beberapa data terkadang memiliki jumlah yang tidak sesuai atau *imbalanced*. Hal ini terjadi karena beberapa subyek atau *class* data memiliki berat sebelah, ada yang memiliki banyak data dan ada yang hanya sedikit data. Hal ini sering terjadi pada berbagai data di kehidupan sehari-hari [9]. Beberapa langkah untuk mengatasi hal ini adalah:

a. Undersampling

Metode ini digunakan dengan cara mengurangi data pada class yang memiliki data terlalu banyak. Pengurangan dilakukan agar terjadi keseimbangan antara class dengan data banyak dan *class* dengan data yang sedikit. Metode ini cocok digunakan apabila dataset memiliki jumlah data yang banyak.

b. Oversampling

Metode ini adalah kebalikan dari metode undersampling yaitu dengan menduplikasi data secara acak pada *class* yang memiliki jumlah data yang sedikit agar terjadi keseimbangan. Metode ini cocok digunakan pada dataset yang memiliki jumlah data sedikit.

c. K-fold cross validation

Metode ini digunakan dengan membagi *dataset* menjadi data *training* dan data *testing* secara beberapa kali sesuai jumlah k atau lipatan. Setiap lipatan akan memiliki data training dan data testing yang berbeda. Penerapan k-fold *cross validation* dapat dilihat pada Gambar 3.

Gambar 3. Penerapan k-fold *cross validation*

Pada gambar diatas dijelaskan pada setiap lipatan akan dilakukan pengambilan data training yang berbeda. Pada intinya keseluruhan data akan dijadikan data training dan juga data testing.

2.6 *Confusion Matrix*

Dalam mencoba atau menguji suatu model klasifikasi pada pembuatan aplikasi *machine learning* dibutuhkan suatu sistem validasi yang digunakan untuk memvalidasi data yang diprediksi oleh mesin. Apakah prediksi tersebut tepat atau salah, serta mengitung seberapa besar akurasi yang didapatkan dari model dan *dataset* yang ada.

Salah satu metode validasi adalah *confusion matrix* yang sudah umum digunakan untuk memberi informasi perbandingan hasil klasifikasi [5]. Metode ini memberikan infromasi validasi berdasarkan nilai *true positive* dan *true negative* dimana model klasifikasi memprediksi dengan benar sedangkan *false positve* dan *false negative* dimana model klasifikasi salah dalam memprediksi.

Metode ini memberikan tiga informasi validasi yaitu *accuracy, precision,* dan *recall. Accuracy* adalah besar nilai akurasi dari model yang melakukan klasifikasi data. Rumus *Accuracy* pada *confusion matrix* dapat dilihat pada Rumus 3.

$$accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{TP + TN + FP + FN} \quad (3)$$

Precision memberikan informasi validasi yang berfokus pada *true positive* dari semua *class positive* yang diprediksi oleh mesin. Rumus *precision* dapat dilihat pada Rumus 4.

$$precision \quad \frac{TP}{TP + FP} \quad (4)$$

Recall adalah rasio keberhasilan mesin dalam menemukan informasi dengan membandingkan *true positive* dan jumlah *true positive* dan *false negative*. Rumus *recall* dapat dilihat pada Rumus 5.

$$recall \quad \frac{TP}{TP + FN} \quad (5)$$

2.7 Penelitian Sebelumnya

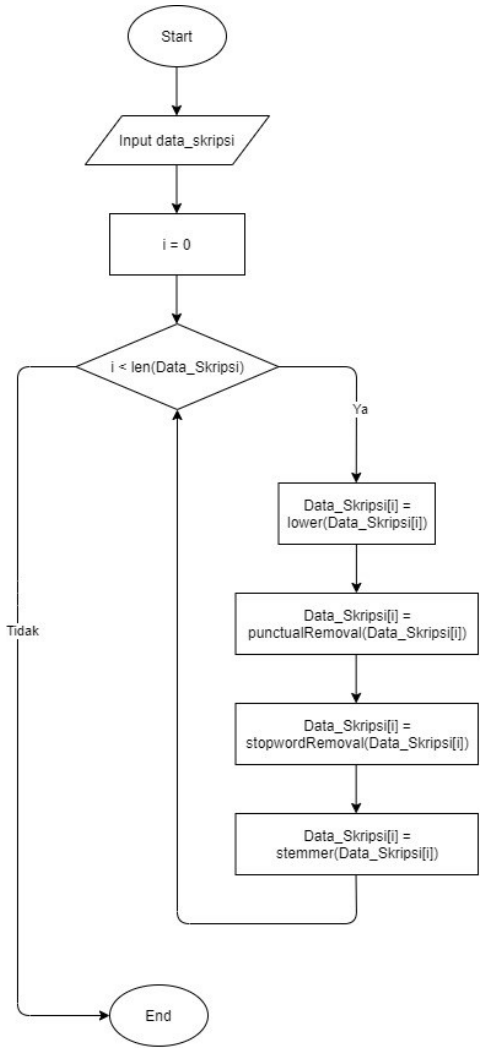
Terdapat dua penelitian sebelumnya yang juga sejenis dengan penelitian ini. Permasalahan kedua penelitian ini adalah membantu mahasiswa tingkat akhir menemukan referensi skripsi sesuai topik pilihannya, sedangkan pada penelitian ini ditujukan untuk pihak perpustakaan dalam menentukan subyek skripsi. Pada penelitian pertama dilakukan *case folding, stemming,* dan *term frequencies* dengan menggunakan metoe Naïve Bayes [7]. Pada penelitian kedua dilakukan pengkategorian manual sebanyak empat label pada *dataset* dan membandingkan metode Naïve Bayes dan SVM [2].

3. DESAIN SISTEM

3.1 *Preprocessing*

Pada tahap *preprocessing* terdapat beberapa bagian proses *preprocessing*. Tiap bagian ini memiliki fungsi tersendiri dalam hal pembersihan data (*data cleaning*) yang nantinya berguna untuk dilanjutkan pada proses *feature extraction* sehingga dapat bekerja dengan lebih baik.

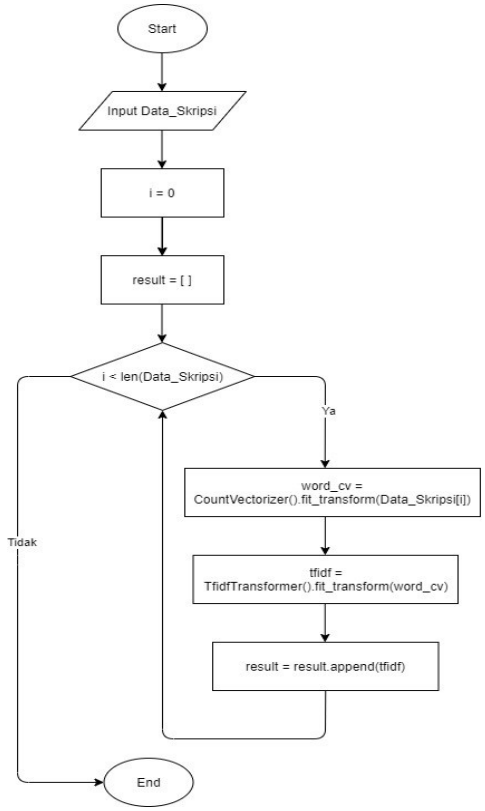
Porses ini membutuhkan *input* berupa data_skripsi. Variabel i adalah variabel penghitung atau biasa disebut *counter variable* yang digunakan untuk menghitung berapa kali proses dijalankan. Selanjutnya akan dilakukan proses perulangan atau biasa disebut *looping*. Proses perulangan ini dijalankan sebanyak jumlah total dari data_skripsi. Pada setiap proses perulangan akan dilakukan empat tahap. Tahap pertama adalah mengubah data_skripsi ke-i menjadi huruf kecil semua. Tahap kedua adalah *punctual removal* yaitu berguna untuk menghapus semua karakter-karakter pada data teks yang bukan huruf. Tahap ketiga adalah *stopword removal*, data_skripsi ke-i akan dicek apakah teks mengandung kata-kata yang tidak penting seperti “untuk”, “yang”, “dan”, dan sebagainya. Jika ada maka kata-kata tersebut akan dihapus dari teks. Tahap terakhir adalah stemming yaitu mengambil kata dasar dari data teks. Dengan stemming maka akan diperoleh teks yang lebih pendek dan lebih memiliki nilai untuk dapat diproses ke proses selanjutnya. *Flowchart preprocessing* akan ditunjukan pada Gambar 4.



Gambar 4. *Flowchart preprocessing*

3.2 *Feature Extraction*

Proses ini akan menjelaskan alur *feature extraction*. Pertama dibutuhkan *input* berupa data_skripsi yang nantinya akan diambil kolom teks nya judul skripsi ataupun abstrak skripsi. Variabel i adalah variabel penghitung atau biasa disebut *counter* dimana proses akan dijalankan beberapa kali sesuai jumlah total data_skripsi. Variabel result adalah variabel dengan tipe *array* yang menampung hasil nilai TF-IDF dari setiap data dari data_skripsi. Setelah itu dijalankan proses perulangan atau *looping* sesuai dengan jumlah total data_skripsi. Pada setiap proses perulangan akan diambil data ke-i pada data_skripsi untuk di ambil kata-kata nya. Setiap kata akan di berikan *index* atau nomor unik, proses ini biasa disebut dengan *tokenizing*. Setelah itu semua jenis kata yang sudah diberi nomor atau *index* akan disimpan pada variabel word_cv. Selanjutnya akan dilakukan perhitungan nilai TF-IDF sesuai Rumus 2. Hasil dari perhitungan TF-IDF ini nantinya akan disimpan pada variabel tfidf. Setelah itu nilai tfidf akan digabungkan sebagai elemen dengan *index* baru pada *array* result. Setelah semua perhitungan TF-IDF selesai dan jumlah i sama dengan atau lebih dari jumlah total data_skripsi, maka proses *feature extraction* akan selesai. Langkah-langkah proses *feature extraction* ini akan dijelaskan pada *flowchart feature extraction* yang ditunjukkan pada Gambar 5 dibawah ini.



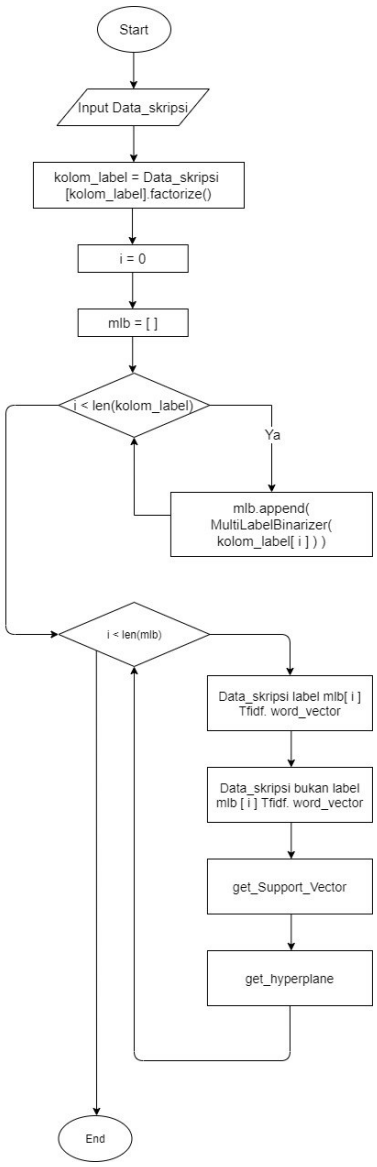
Gambar 5. ***Flowchart feature extraction***

3.3 Klasifikasi SVM

Proses selanjutnya adalah proses klasifikasi dengan menggunakan metode SVM. *Dataset* yang dimiliki adalah data dengan *multilabel dataset. Multilabel* adalah jenis *multiclass dataset* dimana *multiclass* sendiri memiliki arti *dataset* yang memiliki berbagai jenis label (lebih dari dua label). Sedangkan *multilabel* artinya adalah satu jenis label pada data memiliki lebih dari satu jenis label, semisal data pertama memiliki label A dan label B, data kedua memiliki label A dan label C dan seterusnya. Oleh karena itu label atau subyek pada setiap data skripsi akan dikonversikan menjadi biner. Lalu untuk permasalahan *multilabel* ini akan diselesaikan dengan konsep *one vs rest* yang juga salah satu metode *multiclass* juga. Pada dasarnya *one vs rest* dan *one vs one* adalah sama dan tidak terlalu berpengaruh pada akurasi, hanya saja *one vs rest* ini dipakai karena memiliki waktu yang cepat dalam pemrosesan. Konsep dari *one vs rest* ini adalah data akan dibandingkan per subyek, dari subyek A dan bukan subyek A, lalu subyek B dan bukan subyek B yang berulang sebanyak jumlah subyek yang ada.

Proses akan dilakukan dengan menerima *input* berupa data_skripsi. Data_skripsi ini akan diambil kolom-kolom subyek nya dan dilakukan proses faktorisasi dengan fungsi factorize(). Fungsi ini adalah menjadikan unik subyek-subyek yang telah terkumpul. Hasil dari faktorisasi ini akan disimpan pada variabel kolom_label dan dilakukan proses perulangan untuk mengambil nilai biner pada subyek di setiap data_skripsi. Nilai biner subyek akan disimpan pada variabel *array* mlb.

Setelah selesai mendapatkan subyek biner data, maka akan dilakukan lagi proses perulangan sebanyak jumlah subyek yang ada. Disetiap proses perulangan akan dilakukan pengambilan nilai TF-IDF, pencarian *support vector,* dan pencarian *hyperplane*. Pada setiap proses ini dilakukan dengan membandingkan data_skripsi dengan subyek ke-i dan bukan subyek ke-i (semua subyek selain subyek-i). Proses ini dijelaskan oleh *flowchart* pada Gambar 6.

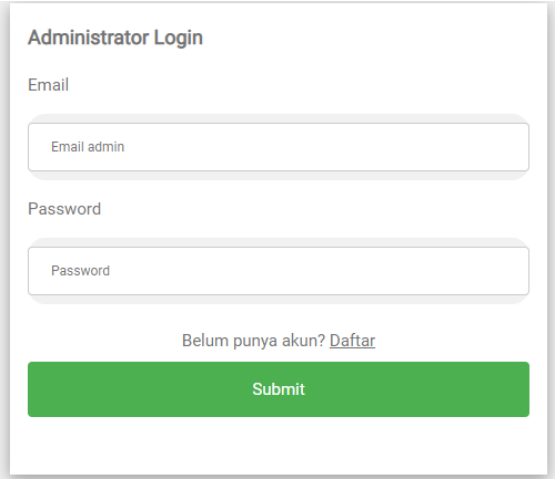


Gambar 6. ***Flowchart*** **klasifikasi SVM**

4. PENGUJIAN SISTEM

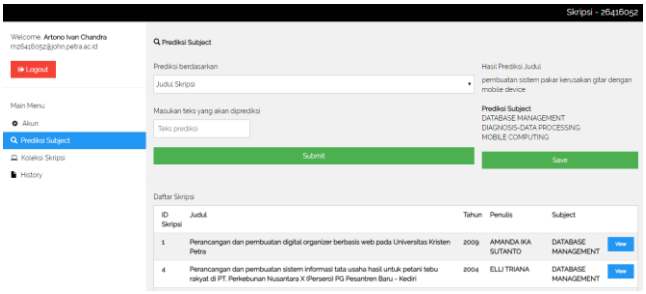
4.1 Pengujian Aplikasi

Pada tahap pengujian ini, penulis akan mencoba menjalankan *website* yang sudah dibuat untuk dijalankan atau disimulasikan untuk melihat seberapa baik *website* dapat bekerja. Pertama *user* atau admin perpustakaan akan melakukan *login* untuk dapat masuk ke *website.* Disini *user* akan memberi *input* pada *textbox* berupa *email* dan *password* dari akun yang sudah terdaftar. Apabila sudah benar maka *user* akan diarahkan ke halaman prediksi untuk memberi *input* teks skripsi untuk diprediksi subyeknya. Apabila *email* atau *password* salah maka akan muncul peringatan pada *website.* Peringatan lain juga muncul apabila *user* mencoba mengakses *website* tanpa *login*. Tampilan halaman ini dapat dilihat pada Gambar 7.



Gambar 7. Halaman *login*

Setelah melakukan *login,* maka user akan diarahkan ke halaman prediksi. Pada bagian ini *user* akan mengisi teks pada *textbox* untuk diprediksi subyek nya. Teks yang di*input user* berupa judul skripsi baru ataupun teks yang berhubungan dengan skripsi. *User* juga memilih data teks judul atau teks abstrak pada data skripsi yang sudah ada untuk memprediksi subyek skripsi baru. Nantinya *user* dapat memilih apakah akan menyimpan hasil prediksi subyek atau tidak. Tampilan halaman daftar akan ditunjukan oleh Gambar 8.



Gambar 8. Halaman prediksi

4.2 Pengujian Akurasi Model SVM

Pada pengujian ini akan ditampilkan hasil akurasi pada data *testing* yang sudah di *resample* dikarenakan ada ketidak seimbangan pada *dataset* dengan proporsi seperti berikut data *training* 618 dan *testing* sebesar 155.

4.2.1 Pengujian Model SVM Judul Skripsi

Pada tahap ini dilakukan pengujian model SVM pada judul skripsi dengan pilihan parameter terbaik yaitu kernel SVM linear, parameter C 100, max_df 1, n-gram (1,2), normalisasi l2, sublinear_tf *true*, dan smooth_idf *true*. Rata-rata akurasi didapat dengan menjumlah semua TN, TP, FN, dan FP lalu dimasukkan pada rumus 3 seperti berikut.

$$accuracy = \frac{264 + 5759}{264 + 5759 + 9 + 13}$$

Hasil rata-rata akurasi adalah 0.996360629. Sedangkan *confusion matrix* dapat dilihat pada Tabel 3.

Tabel 3. Hasil *Testing* Model Judul Skripsi

Subyek	TN	TP	FN	FP	Precision	Recall	F1-Score
ACCOUNTING-DATA PROCESSING	143	12	0	0	1	1	1
ANDROID (ELECTRONIC RESOURCE)	141	12	2	0	0.992908	0.9375	0.9631
APPLICATION SOFTWARE-DEVELOPMENT	145	10	0	0	1	1	1
ARTIFICIAL INTELLIGENCE	144	10	0	1	0.954545	0.99655	0.97446
AUDITING-DATA PROCESSING	155	0	0	0	1	1	1
BUSINESS-DATA PROCESSING	155	0	0	0	1	1	1
CELLULAR TELEPHONE SYSTEMS	155	0	0	0	1	1	1
COMPUTER GAMES-DESIGN	155	0	0	0	1	1	1
COMPUTER GAMES-PROGRAMMING	153	2	0	0	1	1	1
COMPUTER GRAPHICS	136	19	0	0	1	1	1
COMPUTER SOFTWARE-ACCOUNTING	155	0	0	0	1	1	1
COMPUTER VISION	140	15	0	0	1	1	1
CRYPTOGRAPHY-COMPUTER PROGRAM	146	9	0	0	1	1	1
DATA COMPRESSION (COMPUTER SCIENCE)	155	0	0	0	1	1	1
DATA ENCRYPTION (COMPUTER SCIENCE)	146	9	0	0	1	1	1
DATA MINING	144	11	0	0	1	1	1
DATABASE MANAGEMENT	97	51	5	2	0.956985	0.94838	0.95226
DATABASE MICROSOFT SQL SERVER	155	0	0	0	1	1	1
DIAGNOSIS-DATA PROCESSING	143	11	0	1	0.954545	0.99655	0.97446
DIGITAL IMAGE PROCESSING	128	23	4	0	0.988722	0.94	0.96238
DIGITAL VIDEO-EDITING	154	1	0	0	1	1	1
E-COMMERCE (COMPUTER PROGRAM)	143	12	0	0	1	1	1
ELECTRONIC MAIL MESSAGES	155	0	0	0	1	1	1
FINANCIAL STATEMENT	155	0	0	0	1	1	1
INFORMATION STORAGE AND INFORMATION SYSTEMS-ACCOUNTING	130	21	0	4	0.916667	0.98519	0.94703
INTERACTIVE MULTIMEDIA	155	0	0	0	1	1	1
INTERNET (COMPUTER NETWORK)	155	0	0	0	1	1	1
INTERNET PROGRAMMING	154	1	0	0	1	1	1
LIBRARY-AUTOMATION	155	0	0	0	1	1	1
MOBILE COMPUTING	150	4	0	1	0.833333	0.99338	0.89667
MULTIMEDIA SYSTEMS	155	0	0	0	1	1	1
NEURAL NETWORKS (COMPUTER SCIENCE)	154	1	0	0	1	1	1
PROGAMMING (ELECTRIC COMPUTERS)	155	0	0	0	1	1	1
PROJECT MANAGEMENT-COMPUTER PROGRAMS CONSTRUCTION	154	0	1	0	0.496774	0.5	0.49838
RAY TRACING	146	9	0	0	1	1	1
SALES-COMPUTER PROGRAMS	155	0	0	0	1	1	1
SOFTWARE ENGINEERING	152	3	0	0	1	1	1
TRANSACTION SYSTEMS (COMPUTER SYSTEMS)	155	0	0	0	1	1	1
WEBSITES DESIGN	136	18	1	0	0.996403	0.97059	0.98304

4.2.2 Pengujian Model SVM Abstrak Skripsi

Pada abstrak skripsi pilihan parameter terbaiknya kernel SVM rbf, parameter C 100, gamma 0.01, max_df 0.25, n-gram (1,1), normalisasi l2, sublinear_tf *false*, dan smooth_idf *flase*. Maka dihasilkan *confusion matrix* seperti pada Tabel 4.

Tabel 4. Hasil *Testing* Model Abstrak Skripsi

Subyek	TN	TP	FN	FP	Precision	Recall	F1-Score
ACCOUNTING-DATA PROCESSING	141	14	0	0	1	1	1
ANDROID (ELECTRONIC RESOURCE)	139	11	5	0	0.982639	0.84375	0.89857
APPLICATION SOFTWARE-DEVELOPMENT	145	10	0	0	1	1	1
ARTIFICIAL INTELLIGENCE	145	10	0	0	1	1	1
AUDITING-DATA PROCESSING	155	0	0	0	1	1	1
BUSINESS-DATA PROCESSING	155	0	0	0	1	1	1
CELLULAR TELEPHONE SYSTEMS	155	0	0	0	1	1	1
COMPUTER GAMES-DESIGN	155	0	0	0	1	1	1
COMPUTER GAMES-PROGRAMMING	155	0	0	0	1	1	1
COMPUTER GRAPHICS	135	19	0	1	0.975	0.99632	0.98533
COMPUTER SOFTWARE-ACCOUNTING	155	0	0	0	1	1	1
COMPUTER VISION	140	15	0	0	1	1	1
CRYPTOGRAPHY-COMPUTER PROGRAM	146	9	0	0	1	1	1
DATA COMPRESSION (COMPUTER SCIENCE)	155	0	0	0	1	1	1
DATA ENCRYPTION (COMPUTER SCIENCE)	146	9	0	0	1	1	1
DATA MINING	144	11	0	0	1	1	1
DATABASE MANAGEMENT	98	52	4	1	0.970958	0.95924	0.96463
DATABASE MICROSOFT SQL SERVER	155	0	0	0	1	1	1
DIAGNOSIS-DATA PROCESSING	143	12	0	0	1	1	1
DIGITAL IMAGE PROCESSING	130	22	3	0	0.988722	0.94	0.96238
DIGITAL VIDEO-EDITING	154	1	0	0	1	1	1
E-COMMERCE (COMPUTER PROGRAM)	142	13	0	0	1	1	1
ELECTRONIC MAIL MESSAGES	155	0	0	0	1	1	1
FINANCIAL STATEMENT	155	0	0	0	1	1	1
INFORMATION STORAGE AND INFORMATION SYSTEMS-ACCOUNTING	134	21	0	0	1	1	1
INTERACTIVE MULTIMEDIA	155	0	0	0	1	1	1
INTERNET (COMPUTER NETWORK)	155	0	0	0	1	1	1
INTERNET PROGRAMMING	153	2	0	0	1	1	1
LIBRARY-AUTOMATION	155	0	0	0	1	1	1
MOBILE COMPUTING	150	4	0	1	0.9	0.99669	0.94278
MULTIMEDIA SYSTEMS	154	1	0	0	1	1	1
NEURAL NETWORKS (COMPUTER SCIENCE)	155	0	0	0	1	1	1
PROGAMMING (ELECTRIC COMPUTERS)	155	0	0	0	1	1	1
PROJECT MANAGEMENT-COMPUTER PROGRAMS CONSTRUCTION	154	0	1	0	0.496774	0.5	0.49838
RAY TRACING	143	12	0	0	1	1	1
SALES-COMPUTER PROGRAMS	155	0	0	0	1	1	1
SOFTWARE ENGINEERING	152	3	0	0	1	1	1
TRANSACTION SYSTEMS (COMPUTER SYSTEMS)	155	0	0	0	1	1	1
WEBSITES DESIGN	132	17	4	2	0.932663	0.8973	0.91389

Rata-rata akurasi didapat dengan menjumlah semua TN, TP, FN, dan FP lalu dimasukkan pada rumus 3 seperti berikut.

$$accuracy = \frac{268 + 5755}{268 + 5755 + 5 + 17}$$

Hasil rata-rata akurasi adalah 0.996029777.

5. KESIMPULAN DAN SARAN

Setelah melakukan pengujian pada model dengan program yang telah dibuat, maka beberapa hal dapat disimpulkan sebagai berikut:

1. Aplikasi *website* penentu subyek skripsi berjalan dengan baik dengan beberapa *error handling* yang disediakan.
2. Penggunaan *preprocessing* dapat menghasilkan hasil yang lebih baik daripada tanpa *preprocessing* dengan menggunakan *resample data.*
3. Metode *resample* dapat meningkatkan kualitas model dengan baik pada judul dan abstrak skripsi.
4. Parameter TF-IDF terbaik pada judul skripsi adalah max_df sebesar 1, ngram (1,2), normalisasi l2, smooth_idf *true,* sublinear_tf *true*.
5. Parameter TF-IDF terbaik pada abstrak skripsi adalah max_df sebesar 0.25, ngram (1,1), normalisasi l2, smooth_idf *false,* sublinear_tf *false*.
6. Kernel SVM terbaik pada judul skripsi adalah linear, dan parameter C sebesar 100
7. Kernel SVM terbaik pada abstrak skripsi adalah rbf, parameter C sebesar 100, dan parameter gamma 0.01.
8. Rata-rata akurasi model SVM pada judul adalah 0.996360629.
9. Rata-rata akurasi model SVM pada abstrak adalah 0.996029777.
10. Beberapa nama metode menggunakan Bahasa Inggris yang menyebabkan ambigu pada beberapa data.

Dari penelitian yang telah dilakukan ada beberapa saran yang mungkin berguna untuk penelitian selanjutnya. Berikut saran yang ada:

1. Aplikasi mungkin bisa disempurnakan sehingga dapat diakses melalui *mobile device.*
2. Semakin lama data skripsi akan semakin banyak, maka diharapkan ada pertambahan jumlah data agar mesin dapat belajar lebih baik.
3. Dengan jumlah data yang semakin banyak diharapkan akan menambah jenis-jenis subyek skripsi baru agar lebih spesifik dalam proses klasifikasi
4. Mengklasifikasikan semua skripsi di semua jurusan di Universitas Kristen Petra.

6. DAFTAR REFERENSI

[1] H.M. Jogiyanto. 2005. Pengenalan Komputer. Yogyakarta, Indonesia : ANDI.

[2] Hidayatullah, A. F., Ma'arif, M. R. 2016. Penerapan Text Mining dalam Klasifikasi Judul Skripsi. Universitas Islam Indonesia, Jawa Tengah, Yogyakarta

[3] Kao, A., Poteet, S. 2005. Text mining and natural language processing: introduction for the special issue. SIGKDD Explorations, 7(1), 1-2. DOI= https://doi.org/10.1145/1089815.1089816

[4] Nugroho, A. S., Witarto, A. B., & Handoko, D. 2003. *Support Vector Machine*.

[5] Nugroho, K. S. 2019. *Retrieved* April 10, 2020, from https://medium.com/@ksnugroho/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f

[6] Pembobotan Kata TF-IDF. 2016. *Retrieved* April 10, 2020, from https://informatikalogi.com/term-weighting-tf-idf/

[7] Pradikdo, A. C., Ristyawan, A. 2018. Model Klasifikasi Abstrak Skripsi Menggunakan Text Mining untuk Pengkategorian Skripsi Sesuai Bidang Kajian. Universitas Nusantara PGRI Kediri, Jawa Timur, Indonesia.

[8] Pricila, J. M. 2016. Perbandingan Beberapa Pendekatan Multiclass SVM Klasifikasi Artikel Berbahasa Indonesia.

[9] Rumangit, Y. R. Imbalanced Dataset. n.d. *Retrieved* from https://socs.binus.ac.id/2019/12/26/imbalanced-dataset/

[10] Santosa, B. 2007. Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. Yogyakarta, Indonesia: Graha Ilmu.

[11] Sulhaerati, Kurnia, L., Kurniansyah, A., Yuliana, A. A. 2017. Analisis Tanggapan Pasar Terhadap Perusahaan Ritel Raksasa Indomart dan Alfamart. Universitas Islam Indonesia Universitas Komputer Indonesia, Jawa Timur, Indonesia.