

Rekomendasi Prioritas Penyelesaian Masalah Berdasarkan Notulensi dari Bimbingan Skripsi dengan Machine Learning

Christiano Zetro AB Sinaga¹, Anita Qoiriah²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹christiano.21033@mhs.unesa.ac.id

²anitaqoiriah@unesa.ac.id

Abstrak— Proses bimbingan skripsi sering kali menghasilkan notulensi yang memuat berbagai permasalahan yang harus diselesaikan dengan tingkat prioritas berbeda. Namun, penentuan tingkat prioritas tersebut sering kali subjektif dan memakan waktu. Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi tingkat prioritas berdasarkan notulensi bimbingan skripsi menggunakan *Machine Learning* dengan algoritma SVM. Proses pengembangan model meliputi pengumpulan *dataset* notulensi dari mahasiswa semester akhir, augmentasi data melalui *back-translation*, *preprocessing* data, pencarian parameter optimal, serta pelatihan model SVM. Data penelitian diambil dari notulensi mahasiswa dengan total 1720 data yang telah diproses menjadi tiga kelas prioritas. Evaluasi model dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*, menghasilkan akurasi tertinggi 97% dengan kernel RBF setelah penghapusan *outlier*. Sistem sederhana berbasis model SVM kemudian diuji 58 mahasiswa dari berbagai universitas untuk mengevaluasi tingkat kepuasan terhadap akurasi dan kemudahan sistem. Penilaian dilakukan melalui kuesioner berbasis skala *likert*. Hasil analisis menunjukkan bahwa pengguna memberikan penilaian positif terhadap akurasi hasil rekomendasi, kemudahan penggunaan, serta kejelasan informasi yang diberikan sistem

Kata Kunci— *Machine Learning*, SVM, Notulensi, Rekomendasi, Prioritas, Bimbingan Skripsi

I. PENDAHULUAN

Skripsi merupakan proses penelitian yang menjadi salah satu tanggung jawab dari mahasiswa tingkat akhir untuk dapat lulus dan memperoleh gelar yang diharapkannya. Dalam pelaksanaannya, mahasiswa tingkat akhir diwajibkan untuk melewati beberapa tahapan proses terlebih dahulu untuk dapat memperoleh dokumen skripsi yang dapat digunakan sebagai tugas akhir. Tahapan tersebut meliputi pengajuan proposal skripsi kepada dosen pembimbing, mengikuti seminar

proposal, memperoleh surat keputusan sebagai hasil dari seminar proposal, menjalani proses penelitian dan penulisan dokumen, dan proses terakhir yaitu melaksanakan sidang skripsi sebagai pertanggung jawaban dari seminar proposal [1][2]. Dalam proses penyusunan proposal hingga sidang skripsi, mahasiswa yang sedang menyusun skripsi juga dituntut untuk mengikuti bimbingan secara rutin dengan dosen pembimbing yang sudah ditentukan sebelumnya.

Bimbingan skripsi adalah serangkaian kegiatan rutin yang dilakukan oleh pihak mahasiswa penyusun skripsi dan dosen pembimbingnya setelah memperoleh SK (Surat Keputusan) dari diterimanya judul skripsi yang diajukan oleh mahasiswa [3]. Untuk mengoptimalkan jalannya bimbingan rutin untuk memperoleh hasil akhir skripsi yang optimal, penting bagi mahasiswa untuk membuat catatan dari setiap bimbingan skripsi yang dilakukan. Namun tidak jarang juga, mahasiswa menemukan lebih dari 1 atau 2 poin dari catatan bimbingan.

Dalam proses pencatatan untuk setiap detail dari pernyataan pernyataan yang disampaikan oleh dosen pembimbing, mahasiswa dapat mengalami kebingungan dan kehilangan fokus dalam menentukan tingkat prioritas penyelesaian dari setiap masalah yang ditemukan. Hal ini dapat memberikan permasalahan baru untuk mahasiswa tingkat akhir tersebut, yang mana penyelesaian skripsinya dapat mengalami penundaan dan mengganggu kelancaran dari proses akademik dari mahasiswa tersebut. Dengan alasan tersebut, pengembangan aplikasi bimbingan skripsi dengan bantuan penetapan prioritas pengerjaan untuk setiap poin dari notulensi bimbingan skripsinya dapat menjadi sebuah solusi yang relevan.

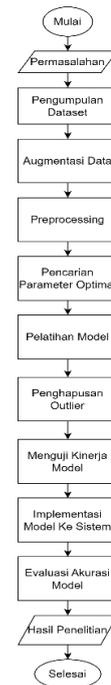
Dalam proses mengatasi permasalahan ini, teknologi *Machine Learning* (ML) menjadi salah satu pendekatan yang relevan. *Machine Learning* juga memiliki banyak algoritma atau model untuk dilatih, yang mana dalam penelitian ini algoritma *Support Vector Machine* (SVM) dipilih sebagai algoritma klasifikasi teks dalam pengembangan rekomendasi prioritas pengerjaan.

Peneliti menggunakan bantuan dari teknologi *Support Vector Machine (SVM)* dalam pengembangan klasifikasi rekomendasi ini. Pemilihan *SVM* dalam penelitian ini didasarkan pada kemampuannya yang unggul dalam mengklasifikasikan data teks dengan akurasi tinggi, khususnya pada data yang tidak selalu memiliki pola linier [4], [5], [6], [7], [8]. Dalam konteks penelitian ini, *SVM* akan dilatih untuk mengenali pola penulisan dari notulensi skripsi yang diberikan oleh pengguna dan memberikan label tingkat prioritas pengerjaannya, yaitu “paling didahulukan,” “didahulukan,” dan “tidak perlu didahulukan,” berdasarkan kombinasi dari kata kunci, frasa, atau konteks kalimat tertentu.

Terdapat beberapa penelitian sebelumnya yang mengeksplorasi terkait topik yang cukup serupa yaitu, pengembangan sistem manajemen bimbingan skripsi untuk meningkatkan efisiensi bimbingan [3], [9], [10], [11]. Meskipun demikian, penelitian ini memiliki keunikan tersendiri, yaitu penggunaan pendekatan *Native Language Processing*, yaitu *Support Vector Machine*, yang sudah banyak digunakan untuk pengolahan data.

Penelitian ini memiliki keunggulan yang terletak pada kemampuan sistem untuk memberikan bantuan berupa saran urutan prioritas pengerjaan dari catatan catatan notulensi bimbingan mahasiswa, Dengan mengembangkan suatu sistem yang dapat menganalisis notulensi bimbingan skripsi yang diberikan oleh mahasiswa sebagai user dan memberikan label prioritas pengerjaan, mahasiswa dapat terbantu dalam memajemen dan menyelesaikan tiap masalah yang muncul dari setiap bimbingan skripsi yang dijalankannya, dan menyelesaikan skripsi secara tepat waktu.

II. METODE PENELITIAN



Gbr 1 Alur penelitian

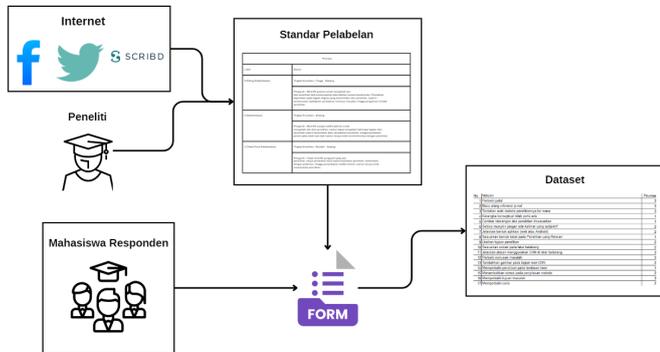
Penelitian pengembangan sistem rekomendasi prioritas berdasarkan notulensi bimbingan ini merupakan penelitian pengembangan (*Research and Development*) yang memiliki fokus pada pengembangan suatu produk, yaitu sistem rekomendasi prioritas penyelesaian berdasarkan notulensi bimbingan pada aplikasi bimbingan skripsi. Alur dari penelitian ini dapat dilihat pada Gbr 1.

A. Pengumpulan Dataset Notulensi

Dalam proses ini, dataset didapat dengan mencari notulensi bimbingan skripsi yang tersebar di internet dan menyebarkan kuisisioner yang dapat diisi oleh mahasiswa yang sudah atau sedang mengambil mata kuliah skripsi dan melaksanakan bimbingan skripsi. Dalam kuisisioner tersebut berisi daftar notulensi mahasiswa dari berbagai sesi bimbingan skripsi yang dilakukan oleh mahasiswa tersebut. Setiap notulensi tersebut juga akan disertakan dengan tingkat prioritas yang ditentukan oleh mahasiswa pengisi kuisisioner secara subjektif, mulai dari rendah, tinggi, hingga menengah. Kuisisioner disebarluaskan secara online melalui *Google Form*.

Seperti yang sudah diGbrkan pada Gbr 2, data notulensi yang sudah didapat akan dikumpulkan ke dalam bentuk file yang dapat digunakan untuk dataset pelatihan mesin, seperti .CSV. Dengan metode ini, dataset yang sudah dikumpulkan diharapkan dapat mencerminkan prioritas masalah yang dialami oleh mahasiswa selama melakukan bimbingan

skripsi. Data notulensi yang sudah terkumpul melalui kuisisioner *online* yang dibagikan oleh peneliti akan melewati tahap penyaringan yang dilakukan oleh peneliti sendiri. Penyaringan tersebut dilakukan dengan memastikan relevansi dari setiap notulensi dengan konteks bimbingan skripsi.



Gbr 2 Alur pengumpulan data notulensi

B. Augmentasi Data

Metode augmentasi data digunakan peneliti untuk menambahkan variasi dari dataset notulen yang sudah dikumpulkan pada proses sebelumnya. Tujuan dari proses ini adalah untuk memperkaya dataset notulensi supaya model memiliki kemampuan untuk mengenal lebih banyak variasi dari pola kalimat, namun tetap menjaga konteks dan makna dari teks. Metode augmentasi untuk dataset teks ialah dengan menggunakan *Back Translation*, yang pada umumnya dilakukan dengan cara menerjemahkan setiap teks ke bahasa asing lalu menerjemahkannya kembali ke bahasa awalnya, yaitu bahasa Indonesia.

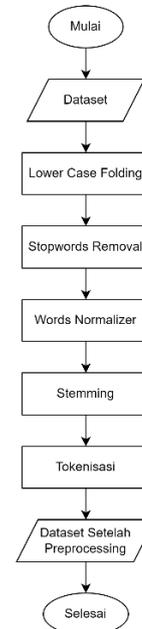


Gbr 3 Alur augmentasi data

Namun, peneliti memilih untuk melakukan metode ini dengan bantuan dari API model bahasa Gemini AI yang dapat mempermudah proses parafrase secara otomatis dan

lebih memperkaya pola kalimat dari dataset dibandingkan dengan menggunakan *Google Translate* yang hanya merubah kata menjadi kata baku. Tahapan dari proses *Back Translation* akan dijelaskan pada sub bab berikutnya, sesuai dengan alur pada Gbr 3.

C. Preprocessing Dataset Dengan NLP



Gbr 4 Alur preprocessing

Data notulensi bimbingan yang sudah didapat pada progres sebelumnya perlu diberlakukan *preprocessing* dengan metode NLP untuk memperoleh akurasi mesin yang optimal. Metode preprocessing NLP yang dilakukan adalah metode yang sudah dijelaskan pada bab sebelumnya. Urutan dari *preprocessing* dapat dilihat pada Gbr 4,

D. Pencarian Parameter Optimal Model

Pencarian parameter optimal bertujuan untuk meningkatkan performa model klasifikasi dalam mengenali pola prioritas dari notulensi bimbingan skripsi. Dalam penelitian ini, metode yang digunakan untuk mencari parameter terbaik adalah *RandomizedSearchCV*, sebuah teknik *hyperparameter tuning* yang menggunakan pendekatan acak dalam mencari kombinasi parameter terbaik.

Metode ini dipilih karena memungkinkan proses pencarian parameter berlangsung lebih cepat dibandingkan *GridSearchCV* yang mencoba semua kombinasi parameter yang mungkin. Pada penelitian ini, *RandomizedSearchCV* digunakan untuk mencari parameter optimal pada dua tahap, yaitu pada model *TF-IDF* (term frequency-inverse

document frequency) dan model *Support Vector Machine* (SVM).

Tahapan Pencarian Parameter Optimal

1. Menentukan parameter yang akan dicari. Parameter yang dicari untuk model *TF-IDF* mencakup ukuran *n-gram*, jumlah kata yang harus di-tokenize, serta penggunaan *stopwords* untuk mengurangi kata yang tidak memiliki informasi penting. Sementara itu, pada model *SVM*, parameter yang dioptimalkan adalah kernel yang digunakan (misalnya *linear* atau *rbf*), nilai *C* (parameter regularisasi), serta parameter *gamma* yang mengontrol tingkat pengaruh dari setiap data titik dalam menentukan batas pemisah. Kedua model ini akan diuji dengan berbagai kombinasi parameter tersebut.
2. Proses *RandomizedSearchCV* dijalankan dengan memberikan daftar parameter yang ingin dicoba pada masing-masing model. Teknik ini akan memilih kombinasi parameter secara acak dalam batasan tertentu. Dengan cara ini, performa setiap kombinasi parameter dapat dievaluasi dan dibandingkan dengan lebih efisien.
3. Evaluasi dan pemilihan parameter terbaik. Hasil dari *RandomizedSearchCV* adalah kombinasi parameter yang memberikan akurasi terbaik. Kombinasi parameter terbaik ini selanjutnya digunakan untuk melatih model akhir yang akan diuji kinerjanya pada data uji. Parameter ini diharapkan dapat meningkatkan akurasi prediksi prioritas, sekaligus meningkatkan stabilitas model pada data notulensi yang beragam.

E. Pelatihan Mesin *Support Vector Machine* (SVM)

Pada penelitian ini, algoritma SVM dengan kernel RBF digunakan untuk mengklasifikasi urutan prioritas pengerjaan masalah berdasarkan notulensi dari bimbingan skripsi mahasiswa. Sebelum memasuki proses pelatihan, dataset yang sudah melewati tahap *preprocessing* sebelumnya diubah menjadi bentuk vector dengan menggunakan metode *TF-IDF* (*Term Frequency-Inverse Document Frequency*). Yang mana, seperti yang sudah dijelaskan pada bab kajian teori, *TF-IDF* akan memberikan bobot pada setiap kata dalam dokumen berdasarkan frekuensi dari kemunculannya dalam dokumen dan seberapa jarang kata tersebut muncul dalam keseluruhan korpus.

- (1) Proses perubahan kata menjadi vektor dengan *TF-IDF* menggunakan formula 1 berikut:

$$TF-IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

Dengan:

- $TF(t, d)$: Frekuensi kemunculan kata t dalam dokumen d

- $IDF(t) = \log\left(\frac{N}{|\{d \in D: t \in d\}|}\right)$: Frekuensi terbalik dokumen yang berisi kata t , dengan N adalah jumlah total dokumen dalam korpus, dan $|\{d \in D: t \in d\}|$ adalah jumlah dokumen yang mengandung kata t .

- (2) Dataset yang sudah berbentuk vektor kemudian digunakan untuk melatih model SVM dengan kernel RBF dengan menggunakan formula 2.

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2)$$

- (3) Setelah model selesai dilatih, proses prediksi dilakukan dengan menggunakan *hyperplane* pemisah yang dihasilkan dari SVM. Yang mana, dengan *hyperplane* tersebut, data baru akan diklasifikasikan ke dalam kelas yang sesuai berdasarkan perhitungan pada formula 3.

$$f(x) = \text{sgn}(w \cdot x + b) \quad (3)$$

- $f(x)$: Fungsi Keputusan yang menentukan kelas dari suatu data
 - sgn : Fungsi tanda yang akan memberikan nilai +1 atau -1 untuk memutuskan kelas
- (4) Data yang diklasifikasikan akan diberikan kelas dengan nilai probabilitas paling tinggi, yaitu kelas yang memiliki margin paling jauh dengan *hyperplane*.

F. Penghapusan *Outlier*

Penghapusan *outlier* dilakukan untuk memastikan bahwa model hanya dilatih dengan data yang benar-benar relevan dan konsisten dengan pola prediksi yang diharapkan. *Outlier* dalam konteks ini adalah notulensi yang prediksi tingkat prioritasnya memiliki probabilitas rendah, yang mengindikasikan bahwa model kurang yakin terhadap prediksi tersebut. Pada penelitian ini, penghapusan *outlier* dilakukan menggunakan model *Support Vector Machine* (SVM) dengan mengaktifkan parameter *probability*, yang memungkinkan model memberikan skor probabilitas pada setiap prediksi.

Tahapan penghapusan *outlier* menggunakan probabilitas prediksi

- 1) *Pengaktifan Parameter Probability*: Parameter probability pada SVM diaktifkan untuk mengukur seberapa yakin model terhadap prediksi yang dihasilkan. Setiap notulensi yang diprediksi akan memiliki skor probabilitas yang menunjukkan tingkat keyakinan model terhadap hasil prediksi tersebut. Skor ini dapat digunakan untuk mendeteksi outlier, di mana nilai probabilitas yang rendah mengindikasikan bahwa model kurang yakin terhadap prediksi yang diberikan.
- 2) *Menentukan Threshold Probabilitas*: Beberapa skenario threshold probabilitas diuji untuk menentukan batasan nilai yang akan digunakan dalam mengidentifikasi outlier. Pada penelitian ini, threshold yang digunakan adalah nilai probabilitas kurang dari ('<') 0.85, 0.8, 0.75, 0.7, dan 0.65. Artinya, setiap notulensi yang memiliki skor probabilitas di bawah nilai threshold yang ditentukan akan dianggap sebagai outlier dan dikeluarkan dari dataset.
- 3) *Penerapan Threshold pada Dataset*: Dataset awal akan disaring menggunakan masing-masing threshold probabilitas, dan setiap notulensi yang dianggap sebagai outlier pada skenario tersebut akan dihapus dari dataset. Proses ini bertujuan untuk menghilangkan data yang tidak konsisten atau yang mungkin menyimpang dari pola yang seharusnya, sehingga model dapat fokus pada data yang relevan dan lebih mudah dalam mengidentifikasi prioritas.
- 4) *Pemilihan Threshold Optimal*: Setelah menerapkan setiap skenario threshold, performa model dievaluasi untuk menentukan threshold yang memberikan hasil terbaik. Threshold optimal adalah yang memberikan keseimbangan terbaik antara ukuran dataset dan akurasi model, sehingga data yang tersisa benar-benar mencerminkan pola prioritas yang dapat dipelajari oleh model. Dengan menggunakan metode ini, diharapkan model dapat dilatih dengan dataset yang lebih bersih dan relevan, yang pada akhirnya akan meningkatkan akurasi model dalam memberikan rekomendasi prioritas pada notulensi bimbingan skripsi.

G. Pengujian Kinerja Model Support Vector Machine

Setelah aplikasi dapat berjalan dengan baik dan benar tanpa adanya bug, peneliti akan melanjutkan ke tahapan pengujian dari model *Support Vector Machine* yang digunakan untuk memberikan rekomendasi prioritas. Proses pengujian dilakukan dengan menggunakan metode *splitting* dengan 5 jenis penjabaran, yaitu 90:10, 80:20, 70:30, 60:40, dan 50:50. Dengan menggunakan penjabaran tersebut, data testing akan diberlakukan fungsi pengujian mesin untuk memperoleh akurasi, presisi, recall, dan f1-score dengan menggunakan formula yang sudah dijelaskan pada bab sebelumnya.

H. Implementasi Model Ke Dalam Sistem

Pada tahapan ini, model SVM yang sudah dilatih pada tahap sebelumnya akan disimpan dengan menggunakan *library pickle*, kemudian diintegrasikan langsung ke dalam system rekomendasi berbasis Python yang dikembangkan dengan menggunakan *framework Streamlit*. Dengan menyimpan model dalam format *pickle*, sistem akan memberikan *response time* dan ketergantungan kepada layanan API lebih sedikit tanpa harus menggunakan API eksternal untuk mengakses model.

Proses implementasi ini diawali dengan menyimpan model yang telah dilatih pada proses sebelumnya ke dalam *format pickle*, sehingga model dapat diakses langsung oleh sistem setiap kali diperlukan untuk melakukan klasifikasi prioritas dari notulensi skripsi mahasiswa. *Framework Streamlit* digunakan sebagai antarmuka interaktif untuk pengguna yang memungkinkan pengguna untuk memberikan notulensi bimbingan skripsi ke dalam bentuk *list*, mendapatkan rekomendasi urutan prioritas, dan memberikan penilaian terhadap akurasi model. Langkah implementasi adalah sebagai berikut:

- (1) *Penyimpanan model SVM dengan library pickle*: Model SVM yang sudah dilatih dan divalidasi akurasi disimpan dalam format *pickle*. File ini nantinya akan dimuat oleh system setiap pengguna memberikan permintaan klasifikasi
- (2) *Integrasi dengan Streamlit: Framework Streamlit* digunakan oleh peneliti untuk mengembangkan antarmuka pengguna yang bersifat interaktif dan mudah diakses. *Streamlit* juga memungkinkan sistem untuk memuat file model yang sebelumnya sudah disimpan dengan format *pickle* dan menjalankan prediksi secara langsung setelah pengguna memberikan notulensi bimbingan skripsi.
- (3) *Deploy sistem dengan Streamlit*: Dalam tahap selanjutnya, pengguna akan memberikan sentimennya terhadap hasil rekomendasi yang diberikan oleh model. Oleh karena itu, model perlu memiliki kemampuan untuk dapat diakses secara publik. Dengan demikian, mahasiswa sebagai pengguna dapat memberikan sentimennya terhadap hasil rekomendasi setelah mencoba model. Dengan menggunakan layanan yang juga disediakan oleh *framework Streamlit*, sistem dapat *di-deploy* agar dapat diakses public.

I. Evaluasi Akurasi Model

Proses ini dilaksanakan setelah melewati seluruh proses sebelumnya, dengan demikian aplikasi sudah diimplementasikan dan sudah melewati tahap pengujian.

Proses terakhir adalah mengumpulkan umpan balik dari pengguna melalui kuisisioner. Kuisisioner ini sudah dirancang oleh peneliti untuk mengevaluasi aspek kemudahan penggunaan, desain antarmuka, dan bantuan yang diberikan oleh aplikasi untuk prioritas pengerjaan dari notulensi bimbingan. Pertanyaan dari kuisisioner berbentuk pertanyaan dengan skala *likert* tentang pengalaman dari pengguna aplikasi. Kuisisioner ini akan disebar ke kalangan mahasiswa tingkat akhir yang sedang melewati proses bimbingan skripsi dan sudah mencoba menggunakan aplikasi bimbingan skripsi dengan fitur rekomendasi prioritas yang sudah dikembangkan oleh peneliti. Hasil dari kuisisioner ini akan dikumpulkan dan dianalisis oleh peneliti untuk membuat kesimpulan dan saran untuk penelitian selanjutnya.

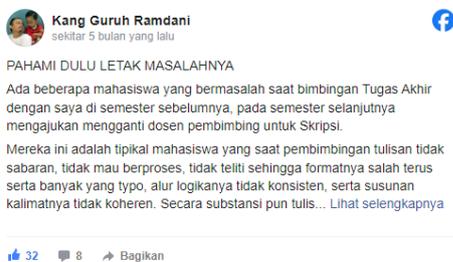
III. HASIL DAN PEMBAHASAN

Subbab ini menjelaskan hasil dari setiap proses implementasi yang sudah dijabarkan peneliti pada bab 3 sebelumnya. Yang mana, proses dalam membangun model rekomendasi prioritas notulensi dengan menggunakan model *Support Vector Machine* (SVM) diberlakukan dengan dataset yang sudah diperoleh peneliti.

A. Pengumpulan Data Notulensi

Tahap pengumpulan data adalah langkah awal dalam proses pelatihan model SVM untuk dapat memberikan rekomendasi prioritas urutan pengerjaan berdasarkan notulensi bimbingan. Pada tahap ini, data dikumpulkan oleh peneliti dari dua sumber, yakni: hasil kuisisioner dan notulensi yang tersedia secara daring, serta dari pengalaman bimbingan skripsi yang dialami langsung oleh peneliti.

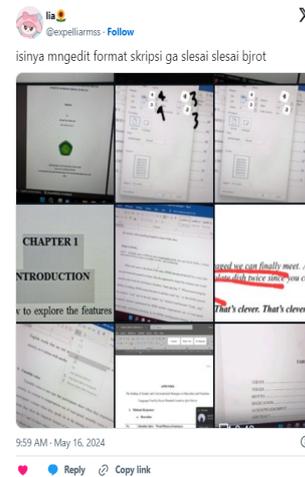
Kuisisioner disebar peneliti kepada mahasiswa yang telah atau sedang dalam masa bimbingan skripsi, dengan tujuan untuk memperoleh notulensi bimbingan skripsi dari mahasiswa. Dalam kuisisioner tersebut, mahasiswa responden diminta untuk memberikan notulensi dari hasil bimbingan skripsi dengan dosen pembimbingnya masing-masing. Bentuk notulensi dapat berupa saran, koreksi, atau masukan kepada skripsi yang sedang disusun responden.



(a) Sumber:

https://www.facebook.com/permalink.php?story_fbid=pfbid02E5WGtcEB

MLfQdtJFxyzgDz9DU3ZuHSeFANb6qHP4fYShFWS77WVussmbawPqs
veLl&id=100081681132053



(b) Sumber:

<https://x.com/expelliarmss/status/179094021053845276>



(c) Sumber:

<https://id.scribd.com/document/538188272/Tabel-Revisi-Skripsi-Popi>

Gbr 5. Contoh data yang dikumpulkan dari internet: (a) dari facebook, (b) twitter, dan (c) scribd

Selain mengandalkan data notulensi yang diperoleh dari kuisisioner, peneliti juga mencari notulensi dari berbagai sumber daring yang tersebar di internet dan mencatat notulensi dari bimbingan yang dialami oleh peneliti secara langsung. Penambahan data dari sumber daring dan pengalaman pribadi peneliti ini berfungsi untuk menambah variasi data, demi dataset dengan variasi data yang luas dan bervariasi, mengingat adanya subjektivitas dari satu responden atau sumber.

Data teks dari sumber tersebut disimpulkan dan diparafrase oleh peneliti agar dapat digunakan oleh model untuk proses pelatihan. Contoh data notulensi asli yang diambil dari internet dapat dilihat pada Gbr 5, dengan sumber data ialah *Twitter*, *Facebook*, dan *Scribd*. Data dari kuisisioner mahasiswa juga disortir kembali oleh peneliti untuk memastikan seluruh data dapat digunakan untuk melatih model.

Tabel 1. Persebaran jumlah data

Sumber Data	Jumlah
Kuisisioner	789
Internet	65
Peneliti	13
Total	867

Proses pengumpulan data dari kuisisioner dilaksanakan pada 1 Juli 2024 hingga 31 Oktober 2024. Sedangkan untuk data yang diperoleh peneliti dari pengalaman pribadi dan internet dikumpulkan pada periode 1 Mei 2024 hingga 31 Oktober 2024. Dengan jumlah data 789 data notulensi,

yang diisi oleh mahasiswa dari berbagai universitas di Indonesia. Sedangkan data tambahan yang diperoleh peneliti dari pengalaman peneliti dan internet, ialah 13 dan 65 data notulensi secara berurutan. Detail persebaran data dapat dilihat pada Tabel 1.

B. Pembersihan Data

Pada tahap ini, proses pembersihan data dilakukan untuk memastikan bahwa data notulensi yang akan digunakan sebagai dataset memiliki kualitas yang baik dan relevan. Data data yang mengandung kesalahan input dari responden dihapus oleh peneliti agar tidak mempengaruhi hasil analisis. Contoh kesalahan yang dimaksud ialah responden yang memberikan tingkat prioritas secara langsung tanpa mencantumkan notulensi hasil bimbingannya terlebih dahulu.

Data tersebut tidak dapat digunakan untuk proses pelatihan model. Dengan demikian, data harus dikeluarkan dari dataset. Proses ini bertujuan untuk memastikan hanya data notulensi yang sesuai yang digunakan dalam pelatihan model *Support Vector Machine*, sehingga hasil analisis dan sistem rekomendasi yang diberikan lebih akurat.

C. Pelabelan Data

Label dari data teks notulensi mahasiswa responden, diberikan oleh mahasiswa tersebut saat mengisi kuisioner yang disebar oleh peneliti. Sedangkan untuk data notulensi yang diperoleh peneliti dalam proses bimbingan dan pencarian di internet. Label berbentuk numerik dari skala 1, prioritas terendah hingga 3, prioritas tertinggi. Ketentuan pemberian label untuk data yang diperoleh peneliti dari internet dan pengalaman pribadi peneliti dapat dilihat pada Tabel 2.

Tabel 2. Ketentuan pemberian label oleh peneliti

Prioritas	
Label	Syarat
3 (Paling Didahulukan)	Tingkat Kesulitan : Tinggi - Sedang
	Pengaruh : Memiliki potensi untuk mengubah alur dari penelitian baik secara parsial atau bahkan secara keseluruhan. Perubahan diperlukan pada bagian bagian yang menentukan alur penelitian, seperti : penyesuaian topik/judul, perubahan rumusan masalah, hingga pergantian metode penelitian.
2 (Didahulukan)	Tingkat Kesulitan : Sedang
	Pengaruh : Memiliki sangat sedikit potensi untuk mengubah alur dari penelitian, namun dapat mengubah beberapa bagian dari penelitian seperti kesesuaian data, penjabaran penelitian, hingga perubahan penuh pada salah satu bab namun hanya untuk mensinkronkan dengan penelitian
	Tingkat Kesulitan : Rendah - Sedang

1 (Tidak Perlu Didahulukan)	Pengaruh : Tidak memiliki pengaruh pada alur penelitian, hanya perubahan kecil seperti kejelasan penulisan, kesesuaian dengan pedoman, hingga penambahan sedikit elemen namun hanya untuk memperjelas penelitian.
-----------------------------	---

Setelah seluruh data notulensi terkumpul, peneliti menggabungkan data tersebut dalam satu file *Excel* yang sebelumnya hanya digunakan untuk menyimpan data dari kuisioner. Contoh bentuk data notulensi dapat dilihat pada Tabel 3. Hal ini dilakukan peneliti untuk memperlancar proses selanjutnya yang membutuhkan satu direktori untuk penyimpanan dari dataset yang akan diproses pada tahap selanjutnya.

Tabel 3. Contoh dataset notulensi

Notulen	Prioritas
Baca ulang referensi jurnal	3
Tentukan arah metode penelitiannya ke mana	3
Sebisa mungkin jangan ada kalimat yang subjektif	2
Jelaskan bentuk aplikasi (web atau Android)	2
Kerangka konseptual tidak perlu ada	1
Gbr rancangan alur penelitian disesuaikan	1

Jumlah total dataset dan persebaran labelnya dapat dilihat pada Tabel 4, yang mana dengan total 867 data notulensi, distribusi prioritas terlihat mendominasi pada salah satu label yaitu label “Didahulukan”. Oleh karena itu, proses augmentasi data diperlukan untuk mengatasi distribusi data yang tidak merata.

Tabel 4. Jumlah dataset notulensi awal dan persebarannya

Label	Jumlah
3 (Paling Didahulukan)	269
2 (Didahulukan)	319
1 (Tidak Perlu Didahulukan)	279
Total	867

D. Augmentasi Data

Sebelum melakukan proses augmentasi, data notulensi mahasiswa dikelompokkan berdasarkan label prioritasnya, yaitu prioritas 1, 2, dan 3. Yang mana notulensi yang sebelumnya tercampur dalam proses sebelumnya dikelompokkan berdasarkan label prioritasnya. Setelah data dikelompokkan kepada berdasarkan labelnya dan disimpan ke file *Excel* yang berbeda. Masing masing file diteruskan ke model Gemini yang juga sudah dikonfigurasi pada program dengan pilihan model ‘gemini-1.5-flash’. Dengan model Gemini yang telah dikonfigurasi, data pada masing label akan dilakukan parafrase dengan tetap menjaga konteks dan makna aslinya.

Tabel 5. Contoh dataset notulensi setelah parafrase

Notulen	Prioritas
Mahasiswa diminta untuk meninjau kembali referensi jurnal yang telah dikumpulkan.	3
Mahasiswa diminta untuk menentukan metode penelitian yang akan digunakan.	3
Hindari penggunaan pernyataan yang bersifat opini pribadi.	2
Tentukan jenis aplikasi yang akan dibuat, apakah berbasis web atau Android.	2

Kerangka konseptual tidak dibutuhkan	1
Alur penelitian telah disesuaikan dengan Gbr rancangan yang dibuat.	1

Hasil *back translate* dengan menggunakan model Gemini dapat dilihat pada Tabel 5. Data notulensi baru yang merupakan hasil proses *back translate* disimpan ke file *Excel* yang juga berbeda beda masing masing labelnya dan digabungkan dengan data awal di proses selanjutnya.

Tabel 6. Persebaran data sebelum penghapusan duplikat

Label	Jumlah
3 (Paling Didahulukan)	538
2 (Didahulukan)	637
1 (Tidak Perlu Didahulukan)	558
Total	1733

Tabel 7. Persebaran data setelah penghapusan duplikat

Label	Jumlah
3 (Paling Didahulukan)	537
2 (Didahulukan)	634
1 (Tidak Perlu Didahulukan)	549
Total	1720

Dengan setiap data telah melewati proses *back translation* dengan Gemini AI, saat ini jumlah persebaran data pada masing masing label adalah dua kali lipat dari jumlah data sebelumnya. Namun dalam persebaran data ini masih terdapat beberapa duplikat pada masing masing label. Hal ini dapat diperlihatkan dengan membandingkan persebaran data pada Tabel 6 yang merupakan persebaran data awal, yaitu dua kali lipat dari data awal, dengan data pada Tabel 7 yang merupakan persebaran data setelah melewati fungsi 'drop_duplicates' dari *library pandas*. Proses menghapus data duplikat ini berfungsi untuk mencegah adanya *overfitting* pada data dengan memastikan setiap data bersifat unik dan dapat memberikan kontribusi berbeda terhadap proses pelatihan model yang akan dilakukan selanjutnya.

Tabel 8. Persebaran data setelah melakukan resampling

Label	Jumlah
3 (Paling Didahulukan)	537
2 (Didahulukan)	537
1 (Tidak Perlu Didahulukan)	537
Total	1611

Proses resampling dilakukan secara otomatis dengan menggunakan bantuan dari *library pandas* pada bahasa pemrograman *Python*. Proses resampling ini juga dilakukan dengan menghapus data pada label secara acak. Dengan demikian, persebaran data pada label saat ini merata untuk masing masing label, seperti pada Tabel 8. Setelah seluruh label memiliki persebaran data yang seimbang, tahap terakhir pada proses augmentasi ini adalah pengacakan data. Dataset yang sudah teracak ini kemudian diekspor kembali ke file berbentuk *Excel* sebagai data akhir yang akan digunakan pada proses pelatihan model selanjutnya.

E. Preprocessing

```
# %%
# Fungsi untuk membersihkan dan memproses teks
def preprocess_text(text):
    # Lowercase folding
    text = text.lower()
    # Hilangkan tanda baca
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Tokenisasi
    tokens = nltk.word_tokenize(text)
    # Normalisasi
    tokens = normalize_text(tokens)
    # Stopwords removal dan stemming
    tokens = [stemmer.stem(word) for word in tokens]
    # Gabungkan kembali token menjadi teks
    return ' '.join(tokens)
```

Gbr 6. Fungsi untuk preprocessing

Proses selanjutnya sebelum pelatihan model SVM adalah *preprocessing* untuk mempersiapkan teks notulensi agar dapat diproses untuk melatih model yang lebih baik. Proses ini mencakup beberapa langkah pengolahan teks yang bertujuan untuk menyederhanakan struktur teks, mengurangi kompleksitas kata, serta mempertahankan makna utama dari setiap kalimat notulensi, dengan urutan sesuai dengan penjelasan di bab sebelumnya pada Gbr 4.

Agar proses *preprocessing* dapat diimplementasikan ke data notulensi, fungsi perlu diinisialisasikan agar proses dapat langsung diimplementasikan ke data notulensi dengan konfigurasi dan urutan yang sudah terstruktur. Hal ini dilakukan agar proses dapat diterapkan secara otomatis dan konsisten kepada seluruh data teks notulensi. Inisialisasi fungsi dari *preprocess* dapat dilihat pada Gbr 6.

Bentuk data teks notulensi yang akan diberlakukan proses *preprocessing* dapat dilihat pada Tabel 9. Data pada tabel tersebut dapat dibandingkan pada data teks notulensi yang telah melewati seluruh tahap *preprocessing* pada Tabel 10

Tabel 9. Data teks notulensi sebelum preprocessing

Notulen
Tambahkan screenshot hasil testing dlm bab lampiran.
Jelaskan mekanisme diet vegan dlm menurunkan risiko penyakit jantung.
Pembahasan skripsi sebaiknya disimpulkan dgn menekankan pentingnya kecerdasan emosional bagi seorang pemimpin.
Harap periksa kembali hasil penelitian agar semua variabel telah diukur dgn tepat.
Tambahkan analisis tentang tantangan yg dihadapi oleh start-up baru.

Tabel 10. Data teks notulensi setelah preprocessing

clean notulensi
tambah screenshot hasil testing dalam bab lampir
jelas mekanisme diet vegan dalam turun risiko sakit jantung
bahas skripsi baik simpul dengan tekan penting cerdas emosional bagi orang pimpin
harap periksa kembali hasil teliti agar semua variabel telah ukur dengan tepat
tambah analisis tentang tantangan yang hadap oleh startup baru

F. Pencarian Parameter Optimal

Dalam tahap ini proses pencarian parameter terbaik dilakukan sebelum pelatihan model. Dalam proses ini

dilakukan pencarian untuk menemukan kombinasi dari parameter terbaik untuk model *Support Vector Machine* (SVM) dan TF-IDF agar model mampu melakukan klasifikasi prioritas dengan akurasi maksimal. Pencarian parameter ini dilakukan dengan menggunakan metode *RandomizedSearchCV*.

Tabel 11. Nilai parameter yang akan dicari optimalnya

Parameter	Nilai
C [SVM]	uniform(0.1, 10)
kernel [SVM]	linear, rbf
gamma [SVM]	scale, auto
max_df [TF-IDF]	0.75, 1.0
ngram_range [TF-IDF]	(1, 1), (1, 2)

Langkah awal yang perlu dilakukan untuk metode *RandomizedSearchCV* adalah inisialisasi parameter SVM dan TF-IDF yang akan diuji. Parameter yang akan diuji mencakup beberapa aspek penting dari suatu *pipeline*. *Pipeline* diinisialisasikan untuk mengalirkan data secara langsung dari TF-IDF menuju model SVM. Daftar dari parameter dan isinya yang akan diuji tercantum pada Tabel 11. Yang mana dalam tabel tersebut terdapat parameter untuk model SVM, yaitu parameter 'C', 'kernel', dan 'gamma'. Sedangkan untuk parameter dari TF-IDF terdiri dari 'max_df' dan 'ngram_range'.

Parameter 'C' dari model SVM merupakan parameter regulasi yang diuji dengan menggunakan rentang nilai 0.1 hingga 10. Kernel dari model SVM yang diuji adalah kernel yang juga sudah dijelaskan pada kajian teori. Sedangkan untuk parameter 'gamma' ialah skala yang menggunakan nilai 'scale' dan 'auto' untuk diuji, yang mempengaruhi jangkauan dari fungsi kernel dalam kasus kernel RBF.

Nilai 0.75 dan 1.0 dari parameter 'max_df' digunakan untuk menyeleksi kata dengan frekuensi dari data lebih dari nilai 'max_df' dari parameter TF-IDF, sehingga mengabaikan kata kata yang sifatnya umum. Sedangkan untuk parameter 'ngram_range' dari TF-IDF diuji dengan mengambil rentang nilai (1,1) dan (1,2), untuk melihat dampak penggunaan *unigram* saja dengan kombinasi dari *unigram-bigram*.

Setelah menentukan *range* parameter yang akan diuji untuk memperoleh parameter optimal, langkah selanjutnya adalah menginisialisasikan *RandomizedSearchCV* dengan meneruskan *range* parameter sebelumnya sebagai salah satu parameter dari *RandomSearch*. Parameter lain dari *RandomizedSearchCV* ialah 'n_iter' dengan nilai 50 untuk mengatur jumlah iterasi maksimal dari proses percobaan parameter secara acak. Baris kode untuk menginisialisasikan *RandomizedSearchCV* dilampirkan pada Gbr 7.

```
# Inisialisasi RandomizedSearchCV
random_search = RandomizedSearchCV(pipeline, param_dist, n_iter=50,
n_jobs=-1, verbose=2, random_state=42)
```

Gbr 7. Inisialisasi random search

Hasil dari parameter yang optimal ialah sebagai berikut:

- C=3.68, gamma='scale', kernel='rbf' untuk parameter SVM
- max_df= 0.75, ngram_range=(1,1) untuk parameter TF-IDF

G. Pelatihan Model

```
# %%
## SVM Dengan Data Bersih
## Split Test
x = df_parafrase['clean_notulensi']
y = df_parafrase['Prioritas']
for num in test_split:
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=num,
random_state=42)
    model = final_pipeline.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f"Model Accuracy {num}: {accuracy_score(y_test, y_pred)}")
```

Gbr 8. Proses pelatihan model dengan 5 skenario

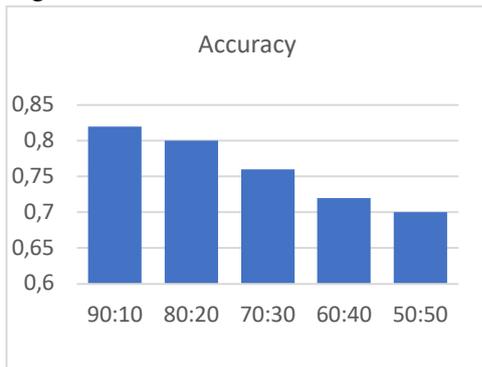
Pada proses ini, model *Support Vector Machine* (SVM) dan TF-IDF telah disimpan dalam sebuah *pipeline* final yang telah dilengkapi dengan parameter optimal yang sudah diperoleh pada proses sebelumnya. *Pipeline* final tersebut sudah termasuk dengan langkah TF-IDF dan model SVM yang masing masing telah dilengkapi dengan parameternya. Untuk melatih model dan memperoleh wawasan mengenai performa dari model dengan kondisi data yang berbeda-beda, pelatihan dan pengujian model dilakukan dengan menggunakan lima skenario pembagian data, yaitu dengan pembagian data latih dan data uji 90:10, 80:20, 70:30, 60:40, 50:50. Skenario pembagian data ini juga sudah ditetapkan pada bab sebelumnya, yaitu pada metode penelitian.

Proses implementasi untuk pelatihan dan pengujian model dapat dilihat pada Gbr 8. Yang mana, pada setiap iterasi pembagian data uji, ukuran data latih akan menyesuaikan dengan jumlah data setelah dikurangi data uji. Sebelum proses pelatihan model, fungsi *preprocessing* yang telah didapat pada proses sebelumnya juga perlu diberlakukan untuk data, serta langkah vektorisasi dengan TF-IDF yang telah termasuk dalam *pipeline* agar data berbentuk teks tersebut dapat digunakan oleh model untuk bahan pembelajaran

H. Pengujian Kinerja Model Awal

Selanjutnya peneliti akan melakukan pengujian kinerja dari model SVM dengan menggunakan proses pelatihan pada proses sebelumnya. Model akan diuji kinerjanya dengan menggunakan metode *split test* dan *cross validation*.

1) *Skenario pengujian pertama:* Dilakukan dengan menggunakan split test. Pengujian model dilakukan dengan lima variasi perbandingan ukuran data latih dan data uji, yaitu 90:10, 80:20, 70:30, 60:40, 50:50 seperti yang telah dilampirkan pada Tabel 12 dan juga telah diGbrkan dalam Gbr 9. Pada ukuran data uji 90:10, model mencapai akurasi tertinggi 82% dengan split test. Namun model memperoleh akurasi terendahnya pada skenario data uji 50:50 dengan akurasi 70%.



Gbr 9. Grafik hasil akurasi pengujian model awal

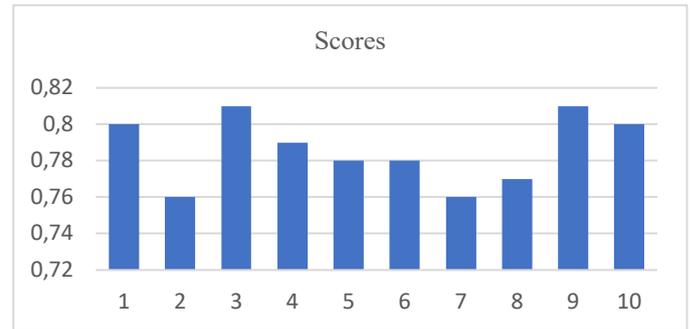
Tabel 12. Pengujian model awal dengan split test

Test Size	Accuracy	Precision	Recall	F1-Score
90:10	0.82	0.82	0.82	0.82
80:20	0.80	0.79	0.79	0.79
70:30	0.76	0.76	0.76	0.76
60:40	0.72	0.72	0.72	0.72
50:50	0.70	0.70	0.70	0.70

2) *Skenario pengujian model kedua:* Dilakukan dengan menggunakan metode cross validation untuk mengevaluasi kinerja dari model secara menyeluruh. Metode ini membagi dataset ke dalam beberapa fold yang sama besar. Dalam setiap iterasi, satu fold digunakan sebagai data uji, sedangkan untuk sisanya digunakan sebagai data latih dari model. Proses tersebut dilakukan berulang kali hingga setiap fold digunakan sebagai data uji. Hasil pengujian model dengan skenario ini dapat dilihat pada Tabel 13, hasil pengujian dalam masing masing fold juga diGbrkan dalam Gbr 10, dengan skor akurasi rata rata yang dihasilkan model ialah 79%. Akurasi tertinggi didapatkan pada fold 3 dan 9 dengan akurasi 81%, sedangkan akurasi terendahnya pada fold 7 dan 2 dengan akurasi 76%.

Tabel 13. Pengujian model awal dengan cross validation

Fold	1	2	3	4	5	6	7	8	9	10
Scores	0.8	0.76	0.81	0.79	0.78	0.78	0.76	0.77	0.81	0.8
Mean	0.79									



Gbr 10. Grafik hasil pengujian model awal dengan cross validation

I. Menghapus Outlier Setelah Pengujian Awal

Proses selanjutnya pada imlementasi pelatihan dari model SVM ialah mengidentifikasi dan menghapus *outlier* pada data. Data *outlier* merupakan data dengan tingkat kepastian prediksi yang rendah. Proses mengidentifikasi *outlier* dilakukan dengan menggunakan model dan proses yang sama, yaitu TF-IDF dan SVM. Namun, untuk mencari data *outlier* dengan model SVM diperlukan parameter baru dari model yaitu 'probability' dengan nilai *True*. Yang mana, dengan adanya parameter tersebut, model akan memiliki kemampuan untuk menggunakan fungsi 'predict_proba' yang juga telah diatur dengan *kernel* RBF. Dengan fungsi tersebut, model akan memberikan nilai probabilitas untuk masing masing kelas dari setiap sampel. Kelas dengan probabilitas paling tinggi akan disimpan sebagai indikator dari kepastian prediksi pada setiap data teks notulensi. Nilai dari probabilitas inilah yang disimpan dalam kolom tersendiri dan digunakan untuk mengidentifikasi *outlier*.

Pengindentikasi outlier dilakukan dengan menerapkan lima skenario *threshold* probabilitas. Dalam setiap skenario, data dengan nilai probabilitas kurang dari (<) *threshold* dari skenario tersebut akan diidentifikasi sebagai *outlier*, sedangkan untuk data yang nilai probabilitasnya tidak memenuhi syarat sebagai *outlier* akan dipertahankan. Jumlah outlier yang ditemukan pada masing masing skenario dapat dianalisa pada Tabel 14.

Tabel 14. Jumlah outlier dengan masing masing threshold

Threshold	Jumlah
< 0.85	626
< 0.8	468
< 0.75	360
< 0.7	262
< 0.65	202

J. Pengujian Kinerja Model Setelah Penghapusan Outlier

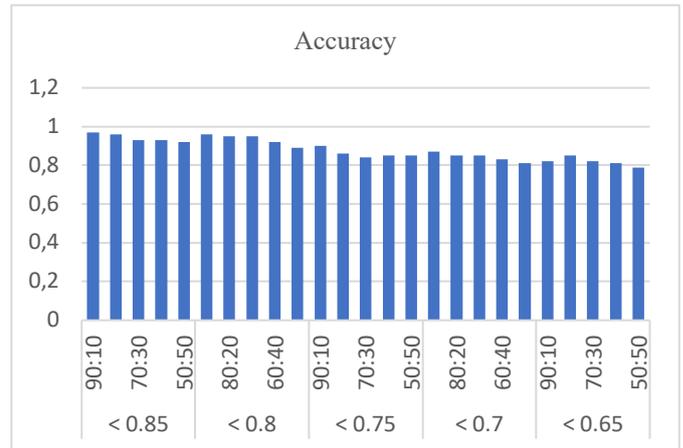
Dalam sub bab ini, peneliti melakukan pengujian dari model akhir, yaitu model yang dilatih dengan menggunakan data tanpa *outlier*. Proses pengujian akan dilakukan dengan

dua metode pengujian yaitu pengujian dengan *split test* dan *cross validation*.

- 1) *Pengujian Model Akhir Dengan Split Test*: Hasil pengujian dari skenario ini dapat dilihat pada Tabel 15. Dapat dibandingkan pada tabel tersebut, model dengan threshold < 0.85 dan perbandingan data latih dan data uji 90:10 mencapai akurasi tertinggi sebesar 97%, sementara pada ukuran data uji 50:50 dengan threshold < 0.65, akurasi terendah mencapai 79%. Pengaruh threshold terhadap akurasi juga dapat terlihat secara signifikan, dengan threshold yang lebih rendah cenderung memberikan akurasi yang lebih buruk. Perbandingan dari masing masing pengujian juga diGbrkan dalam bentuk grafik pada *Gbr 11*.

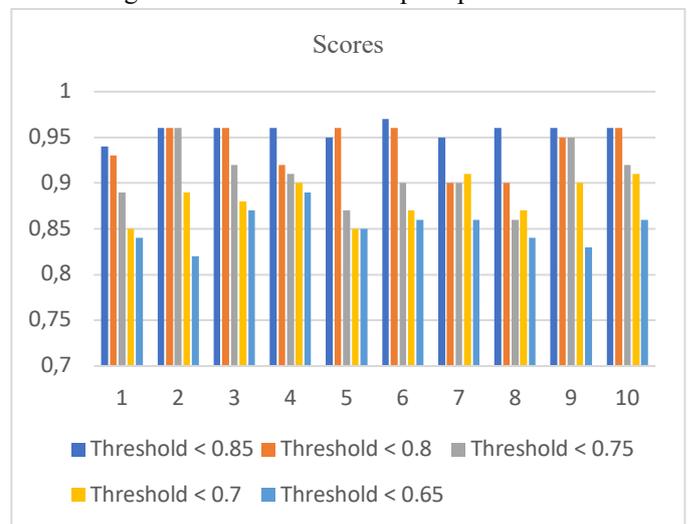
Tabel 15. Pengujian split test model akhir

Threshold	Test Size	Accuracy	Precision	Recall	F1-Score
< 0.85	90:10	0.97	0.96	0.97	0.96
	80:20	0.96	0.95	0.97	0.96
	70:30	0.93	0.92	0.93	0.93
	60:40	0.93	0.93	0.93	0.93
	50:50	0.92	0.92	0.92	0.92
< 0.8	90:10	0.96	0.97	0.97	0.97
	80:20	0.95	0.95	0.95	0.95
	70:30	0.95	0.95	0.95	0.95
	60:40	0.92	0.93	0.92	0.93
	50:50	0.89	0.90	0.88	0.89
< 0.75	90:10	0.90	0.89	0.90	0.89
	80:20	0.86	0.86	0.86	0.86
	70:30	0.84	0.84	0.84	0.84
	60:40	0.85	0.85	0.85	0.85
	50:50	0.85	0.84	0.85	0.84
< 0.7	90:10	0.87	0.87	0.86	0.86
	80:20	0.85	0.85	0.85	0.85
	70:30	0.85	0.85	0.85	0.85
	60:40	0.83	0.83	0.83	0.83
	50:50	0.81	0.81	0.81	0.81
< 0.65	90:10	0.82	0.82	0.82	0.82
	80:20	0.85	0.85	0.85	0.85
	70:30	0.82	0.83	0.82	0.82
	60:40	0.81	0.81	0.81	0.81
	50:50	0.79	0.79	0.79	0.79



Gbr 11. Grafik hasil pengujian model dengan masing masing threshold dengan split test

- 2) *Pengujian Model Akhir Dengan Cross Validation*: Dalam skenario kedua, peneliti melakukan pengujian kinerja model akhir dengan metode cross validation. Dapat dilihat pada Tabel 16 dengan grafik yang mengGbrkannya pada Gbr 12, akurasi tertinggi yang didapat juga sama dengan pengujian split test yaitu pada fold ke 6 di threshold < 0.85. Yang mana, nilai akurasi dari model cenderung menurun seiring dengan penurunan nilai dari threshold. Hal ini dikarenakan penurunan threshold memberikan data dengan tingkat ambiguitas yang tinggi kepada model, yang dapat meningkatkan resiko kesalahan pada prediksi.



Gbr 12. Grafik hasil pengujian model dengan cross validation

Tabel 16. Hasil pengujian dengan Cross Validation

Fold	Threshold < 0.85	Threshold < 0.8	Threshold < 0.75	Threshold < 0.7	Threshold < 0.65
1	0.94	0.93	0.89	0.85	0.84
2	0.96	0.96	0.96	0.89	0.82
3	0.96	0.96	0.92	0.88	0.87
4	0.96	0.92	0.91	0.9	0.89
5	0.95	0.96	0.87	0.85	0.85

6	0.97	0.96	0.9	0.87	0.86
7	0.95	0.9	0.9	0.91	0.86
8	0.96	0.9	0.86	0.87	0.84
9	0.96	0.95	0.95	0.9	0.83
10	0.96	0.96	0.92	0.91	0.86

K. Validasi Model Akhir Dengan Data Baru

Tabel 17. Data notulensi baru yang digunakan untuk validasi akurasi model

data baru	prioritas
Coba perjelas alur dokumen untuk perijinan	1
Judul sesuaikan	3
Datanya diperjelas	2
Buat diagram penelitian	1
Daftar tabel dan Gbr dipisah	1

Pengujian terakhir dilakukan pada model dengan data yang sama dengan pengujian sebelumnya, namun seluruh data digunakan untuk melatih model. Model yang telah dilatih dengan seluruh data diuji untuk mengklasifikasikan data notulensi baru, dengan contohnya pada Tabel 17, untuk mevalidasi kestabilan model secara keseluruhan. Pada skenario ini variasi *threshold* yang digunakan juga masih sama dengan pengujian sebelumnya. Hasil pengujian ini dapat dilihat pada Tabel 18.

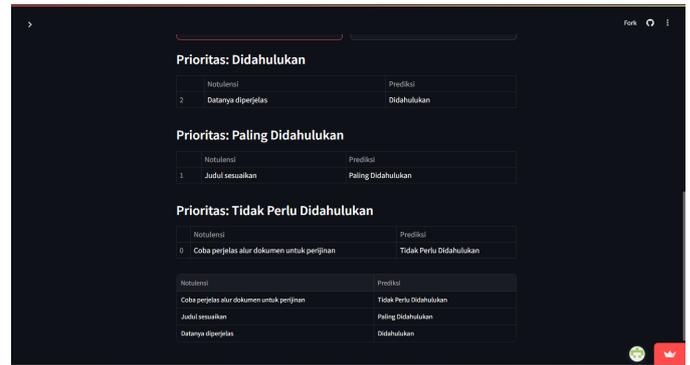
Tabel 18. Pengujian model dengan data baru dan cross validation

Threshold	New Data Accuracy
< 0.85	0.94
< 0.8	0.88
< 0.75	0.82
< 0.7	0.76
< 0.65	0.70

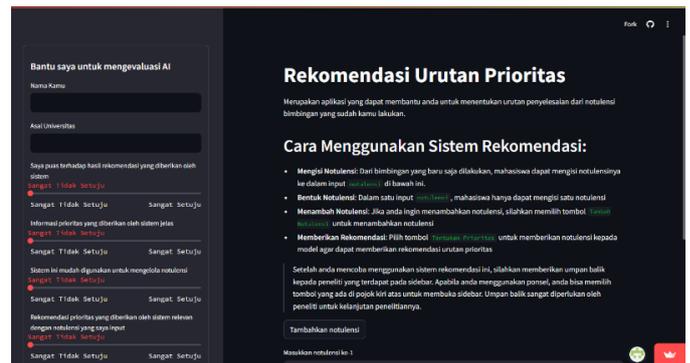
Pada pengujian tersebut akurasi tertinggi pada pengujian data baru didapat pada *threshold outlier* < 0.85, yaitu 94%. Sedangkan akurasi terendah pada pengujian data baru diperoleh pada *threshold outlier* < 0.65 dengan akurasi turun menjadi 70%. Hasil ini menunjukkan bahwa penggunaan *threshold* yang lebih tinggi lebih sesuai dalam mengidentifikasi data yang relevan dan menghasilkan akurasi yang lebih akurat.

L. Penilaian Mahasiswa Terhadap Akurasi Model

Untuk memperoleh evaluasi secara langsung, peneliti mengembangkan sistem sederhana yang memiliki model SVM terbaik dari proses sebelumnya. Model dan vektorisasi TF-IDF dengan parameternya disimpan dengan format *pickle*, sehingga dapat dipanggil langsung dalam sistem berbasis *web* menggunakan *Python Streamlit*.



Gbr 13. Tampilan sistem rekomendasi setelah memberikan prioritas



Gbr 14. Tampilan awal sistem rekomendasi

Model yang disimpan dalam sistem ialah model dengan akurasi tertinggi setelah melewati seluruh proses penelitian. Sistem ini bertindak sebagai wadah dari model, diakses oleh mahasiswa untuk memberikan penilaian terhadap kualitas dari rekomendasi yang dihasilkan serta kemudahan dari sistem. Tampilan dari sistem ini dapat dilihat pada Gbr 14 dan Gbr 13.

Setelah mengakses dan mencoba sistem rekomendasi, mahasiswa diminta untuk memberikan penilaian melalui formulir yang juga sudah disediakan dalam sistem. Formulir ini mencakup dua pertanyaan utama, yakni: pertama, penilaian mahasiswa terhadap akurasi hasil rekomendasi yang diberikan; kedua, kejelasan hasil rekomendasi yang disampaikan oleh model. Jawaban dari setiap pertanyaan menggunakan skala *likert* dari satu hingga lima. Yang mana, nilai satu menunjukkan penilaian terendah dan lima menunjukkan penilaian tertinggi. Survei diisi oleh 58 mahasiswa responden yang berasal dari berbagai universitas dalam negeri.

Pertanyaan yang diberikan oleh peneliti kepada responden dilampirkan dalam Tabel 19. Pernyataan tersebut diberikan kepada mahasiswa, yang akan menjawab dengan skala likert dari 1 (sangat tidak setuju) hingga 5 (sangat setuju). Hasil analisis dari jawaban responden terhadap pertanyaan yang diberikan, dapat dianalisis pada Tabel 20.

Secara keseluruhan, hasil kuisioner menunjukkan respon positif terhadap seluruh aspek *machine learning*. Dengan topik

Tabel 19. Kumpulan pertanyaan berdasarkan topik

Topik	Pernyataan
Hasil Rekomendasi	Saya puas terhadap hasil rekomendasi yang diberikan oleh sistem
	Informasi prioritas yang diberikan oleh sistem jelas
	Rekomendasi prioritas yang diberikan oleh sistem relevan dengan notulensi yang saya input
Penggunaan Sistem	Sistem ini mudah digunakan untuk mengelola notulensi
	Saya merasa bahwa sistem ini dapat digunakan secara mandiri tanpa panduan tambahan
Kepercayaan dan Potensi	Saya merasa bahwa rekomendasi yang diberikan oleh sistem sering membantu menyelesaikan masalah dengan lebih cepat
	Saya percaya terhadap hasil rekomendasi prioritas yang diberikan oleh sistem
	Saya merasa sistem ini berpotensi untuk membantu mahasiswa lainnya dalam mengelola notulensi bimbingan
	Saya akan merekomendasikan sistem ini kepada mahasiswa lainnya
Tampilan Antarmuka	Saya puas terhadap tampilan antarmuka dari sistem ini

Variasi dari jawaban responden juga stabil di seluruh

Hasil Rekomendasi yang memperoleh rata rata 4.28 yang artinya hasil rekomendasi yang diberikan oleh model bersifat relevan dan akurat terhadap responden. Kepercayaan dan Potensi dari model juga memperoleh rata rata 4.29 yang berarti model rekomendasi tidak hanya akurat, namun juga dapat dipercaya dan pandangan positif terhadap potensinya.

Tabel 20. Hasil analisis jawaban dari responden

Topik	Pertanyaan	Mean	Std Div	Mean Topik
Hasil Rekomendasi	P1	4.29	0.81	4.28
	P2	4.28	0.69	
	P4	4.28	0.74	
Penggunaan Sistem	P3	4.48	0.70	4.37
	P8	4.26	0.90	
Kepercayaan dan Potensi	P5	4.17	0.81	4.29
	P7	4.17	0.77	
	P9	4.43	0.70	
	P10	4.38	0.64	
Tampilan Antarmuka	P6	4.17	0.83	4.17

Variasi dari jawaban responden juga stabil di seluruh kategori, yang menunjukkan adanya keseragaman dalam pandangan responden. Pertanyaan dengan kategori sistem yaitu Penggunaan Sistem memiliki standar deviasi berkisar dari 0.7 dan 0.9, serta Tampilan Antarmuka yang memiliki standar deviasi 0.83 yang menunjukkan variasi yang lebih tinggi dibandingkan kategori model. Topik dengan kategori model *machine learning* yaitu Hasil Rekomendasi dan Kepercayaan dan Potensi model memiliki variasi data 0.64 hingga 0.81 yang menunjukkan variasi jawaban lebih konsisten.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, berikut adalah kesimpulan dari penelitian ini:

1. Proses implementasi pelatihan model prioritas berdasarkan notulensi notulensi dengan SVM melibatkan tahapan penting dengan hasil pelatihan model yang didapat ialah dengan menggunakan kernel RBF dengan 867 data sebelum augmentasi dan 1611 data setelah augmentasi.
2. Hasil pengujian kinerja dari model menunjukkan bahwa model SVM yang dikembangkan mampu memberikan performa klasifikasi, dengan akurasi tertingginya mencapai 97% pada data tanpa *outlier* dan 82% pada data dengan *outlier*. Sedangkan akurasi terendah model yaitu 79% pada data tanpa *outlier* dan 70% pada data dengan *outlier*. Model ini akurasi yang tinggi pada pengujian *split test* dan pengujian data baru.
3. Penilaian mahasiswa terhadap akurasi model menunjukkan bahwa mayoritas mahasiswa responden merasa puas dengan hasil rekomendasi yang diberikan oleh model dengan rata rata jawaban mahasiswa terhadap akurasi model mencapai 4.28 dan kepercayaan mahasiswa mencapai rata rata 4.29. Selain itu, penggunaan sistem berdasarkan mahasiswa responden mencapai rata rata tertinggi dibandingkan kategori lain yaitu 4.37 dan penilaian mahasiswa terhadap antarmuka sistem juga mencapai rata rata 4.17.

DAFTAR PUSTAKA

- [1] D. M. Sri Prastiwi M.Pd., *Pedoman Tugas Akhir Universitas Negeri Surabaya*. 2024.
- [2] W. Darmalaksana, "Panduan penulisan skripsi & tugas akhir," *Fak. Ushuluddin UIN Sunan Gunung Djati Bandung*, pp. 1–40, Feb. 2022, [Online]. Available: <https://etheses.uinsgd.ac.id/id/eprint/48611%0A>
- [3] Y. Yarpiransa, D. Saripurna, and H. Santoso, "Implementasi Metode Scrum pada Pengembangan Aplikasi Bimbingan Skripsi Online," *Hello World J. Ilmu Komput.*, vol. 2, no. 1, pp. 42–57, 2023, doi: 10.56211/helloworld.v2i1.228.

- [4] R. K. Anggraini, K. Kusriani, and H. Al Fatta, "Metode Support Vector Machine pada Klasifikasi Pengaduan Masyarakat," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, no. 1, pp. 110–116, 2023.
- [5] D. M. Abdullah and A. M. Abdulazeez, "Machine Learning Applications based on SVM Classification: A Review," *Qubahan Acad. J.*, vol. 1, no. 2, pp. 81–90, 2021, doi: 10.48161/qaj.v1n2a50.
- [6] M. T. H. K. Tusar and M. T. Islam, "A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data," in *Proceedings of International Conference on Electronics, Communications and Information Technology, ICECIT 2021*, 2021. doi: 10.1109/ICECIT54077.2021.9641336.
- [7] D. E. Cahyani and I. Patasik, "Performance comparison of tf-idf and word2vec models for emotion text classification," *Bull. Electr. Eng. Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021, doi: 10.11591/eei.v10i5.3157.
- [8] A. Maulana, Inayah Khasnaputri Afifah, Asghafi Mubarrak, Kiagus Rachmat Fauzan, Ardhan Dwintara, and B. P. Zen, "Comparison of Logistic Regression, Multinomialnb, Svm, and K-Nn Methods on Sentiment Analysis of Gojek App Reviews on the Google Play Store," *J. Tek. Inform.*, vol. 4, no. 6, pp. 1487–1494, 2023, doi: 10.52436/1.jutif.2023.4.6.863.
- [9] M. A. Kurniawan, I. Fitri, and D. Hidayatullah, "Sistem Informasi Bimbingan Skripsi Menggunakan Metode Rapid Application Development Berbasis User Centered Design," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 838, 2021, doi: 10.30865/mib.v5i3.3068.
- [10] S. Lesbassa, A. Setiawan, and L. P. Dewi, "Rancang Bangun Aplikasi Bimbingan Skripsi Online Mahasiswa Berbasis Mobile (Studi Kasus Program Studi Informatika Universitas Kristen Petra)," *J. Infra*, vol. 9, no. 2, pp. 302–307, 2021.
- [11] Z. D. Kussasih, H. Tolle, and F. Al Huda, "Pengembangan Aplikasi Bimbingan Akademik Mahasiswa Filkom Menggunakan Pendekatan Multi-Platform dengan Metode Agile System Development," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, 2024.