

## BAB 4

### HASIL PENELITIAN

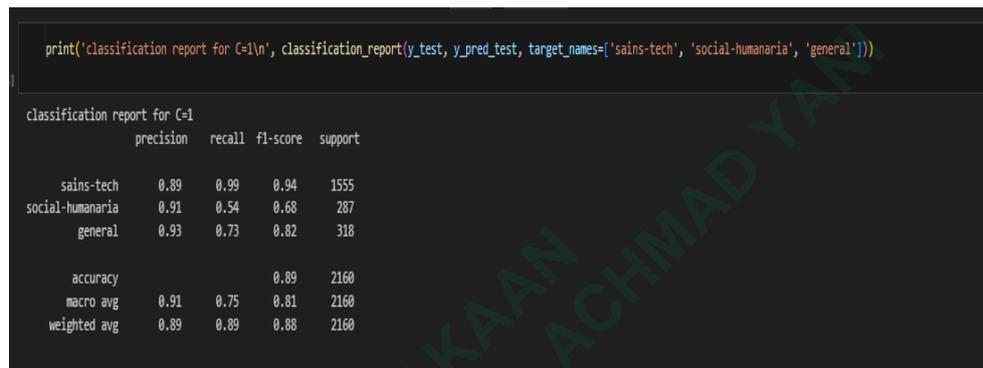
#### 4.1 RINGKASAN HASIL PENELITIAN

Penelitian rekomendasi buku ini diperoleh berdasarkan kumpulan data dari *platform*. Jumlah data yang disimpan yaitu 8702 data berlabel *sains-tech*, *social-humanaria*, dan *general*. Bagian data yang menjadi fitur dari penelitian adalah judul buku yang telah melewati serangkaian pra-pemrosesan seperti mengubah kalimat menjadi *lowercase* atau ukuran kecil, menghilangkan unsur angka dan karakter yang tidak memiliki fungsi, pemecahan kata dalam kalimat serta menghilangkan kata sambung atau kata yang memiliki informasi yang rendah, setelah itu data dipisah dengan modul *train\_test\_split* yang terdapat pada *sklearn.model\_selection* oleh bahasa pemrograman Python dengan jumlah 75% total data sebagai data latih sedangkan 25% menjadi data uji dalam penelitian ini.

Kemudian dilakukannya pembangunan *pipeline* dengan parameter pembobotan kata menggunakan TF-IDF dan penyesuaian model SVM menggunakan implementasi *libsvm SVC*. Berdasarkan referensi dari landasan teori bahwa *SVC* memiliki parameter *C* untuk mengoptimalkan SVM dalam menghindari kesalahan klasifikasi dari setiap contoh data pelatihan. Dalam penelitian ini, ada perbandingan ketika menggunakan parameter *C* dengan nilai yang berbeda yaitu  $C = 0.05$  dan  $C = 1$  dari data latih sejumlah 6477 data. Berikut uraian untuk perbandingan nilai  $C = 0.05$  yaitu memberikan nilai akurasi **72.2%** sedangkan  $C = 1$  yaitu memberikan peningkatan akurasi **88.4%**. Ini menunjukkan bahwa menurut Scikit Learn (2023) bahwa nilai *C* yang besar akan memilih *hyperplane* dengan *margin* yang lebih kecil pada pengklasifikasian nilai positif, sedangkan nilai *C* yang kecil akan memilih *hyperplane* dengan *margin* yang lebih besar pada pengklasifikasian nilai negatif bahkan untuk kasus ini apabila data latih dapat dipisahkan secara *linear* adalah benar karena adanya peningkatan kompleksitas kelas hipotesis yang bergantung kepada besarnya nilai *C*. Kemudian model yang sudah dibentuk dan disesuaikan oleh parameter fitur dan label,

diprediksi menggunakan data latih serta memberikan keluaran hasil berupa label untuk memberikan rekomendasi pada sistem. Untuk mempermudah presentasi model dan menentukan rekomendasi buku maka dibangun sebuah sistem yang dinamakan Sistem Rekomendasi Buku.

Gambar 4.1 dan Gambar 4.2 merupakan laporan klasifikasi dari hasil evaluasi model menggunakan *Confusion Matrix*.

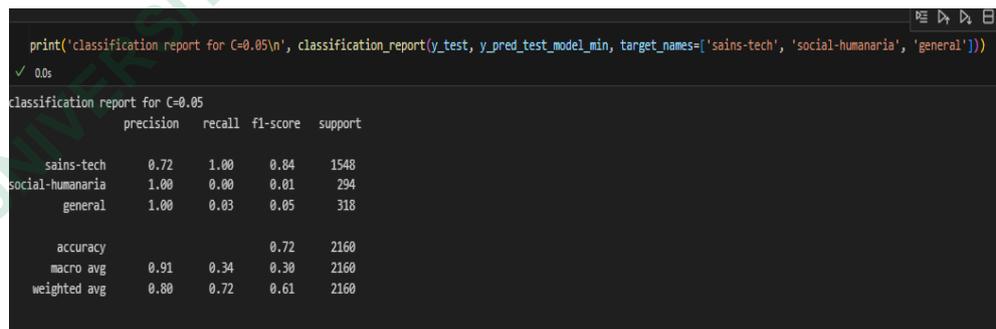


```
print('classification report for C=1\n', classification_report(y_test, y_pred_test, target_names=['sains-tech', 'social-humanaria', 'general']))
```

classification report for C=1				
	precision	recall	f1-score	support
sains-tech	0.89	0.99	0.94	1555
social-humanaria	0.91	0.54	0.68	287
general	0.93	0.73	0.82	318
accuracy			0.89	2160
macro avg	0.91	0.75	0.81	2160
weighted avg	0.89	0.89	0.88	2160

Gambar 4.1 Evaluasi model  $C = 1$

Gambar 4.1 menjelaskan bahwa dengan parameter  $C=1$  memberi hasil evaluasi yang lebih baik yaitu akurasi 88%, *precision* 91%, *recall* 75%, *f1-score* 81% untuk pengujian data berjumlah 2160 data.



```
print('classification report for C=0.05\n', classification_report(y_test, y_pred_test_model_min, target_names=['sains-tech', 'social-humanaria', 'general']))
```

classification report for C=0.05				
	precision	recall	f1-score	support
sains-tech	0.72	1.00	0.84	1548
social-humanaria	1.00	0.00	0.01	294
general	1.00	0.03	0.05	318
accuracy			0.72	2160
macro avg	0.91	0.34	0.30	2160
weighted avg	0.80	0.72	0.61	2160

Gambar 4.2 Evaluasi model  $C = 0.05$

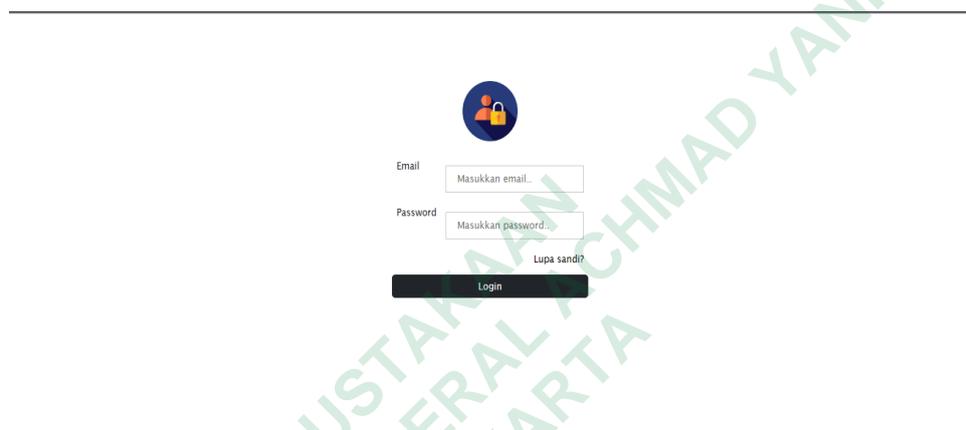
Gambar 4.2 menjelaskan bahwa dengan parameter  $C=0.05$  memberi hasil evaluasi yaitu akurasi 72%, *precision* 91%, *recall* 34%, *f1-score* 30% untuk pengujian data berjumlah 2160 data. Dari hasil perbandingan, maka parameter regulasi-L2 atau  $c$  yang dipilih adalah  $C=1.0$

## 4.2 IMPLEMENTASI DESAIN ANTARMUKA

Berikut ini tampilan dari implementasi desain antarmuka Sistem Rekomendasi Buku.

### 1. Halaman *Login*

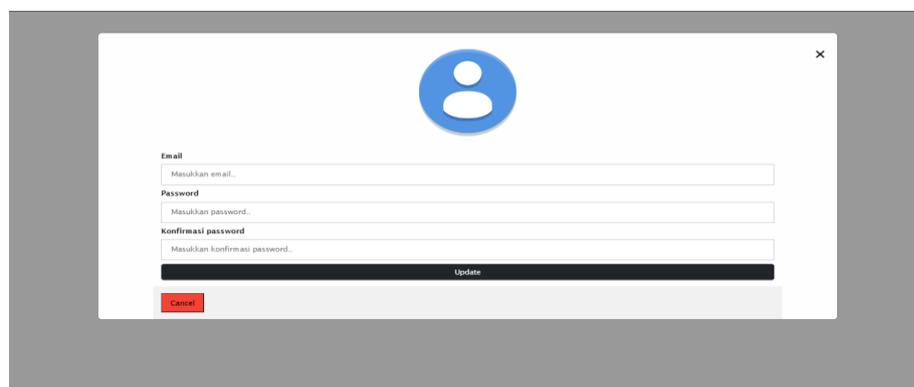
Pengguna diminta untuk memasukkan email dan password yang sudah terdaftar sebagai pelajar oleh sistem untuk proses pengenalan autentikasi dan otorisasi pengguna. Berikut Gambaran implementasi melalui Gambar 4.3



Gambar 4.3 implementasi halaman *Login*

### 2. Kotak dialog formulir lupa kata sandi

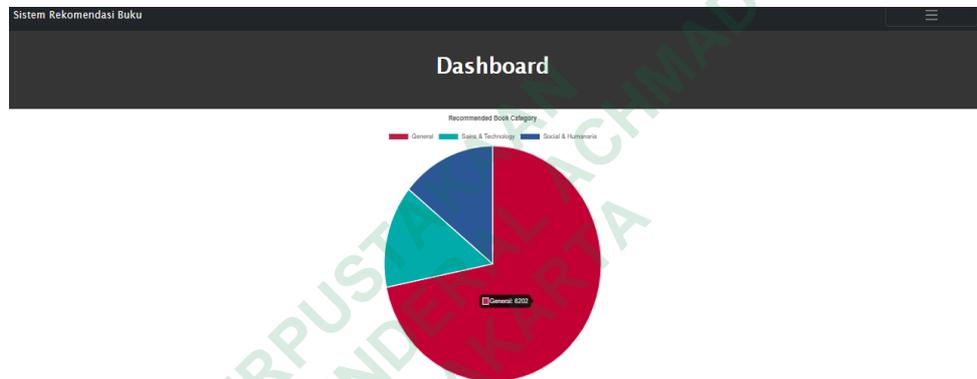
Apabila pengguna melupakan kata sandi yang sudah terdaftar maka pengguna diminta untuk memasukkan email, password serta mengkonfirmasi password baru. Tujuan dari memasukkannya email yang sudah terdaftar sebagai pelajar oleh sistem untuk mengenali apakah email telah tersedia pada basis data sistem. Berikut Gambaran implementasi melalui Gambar 4.4



Gambar 4.4 implementasi Kotak Dialog

### 3. Halaman *Dashboard*

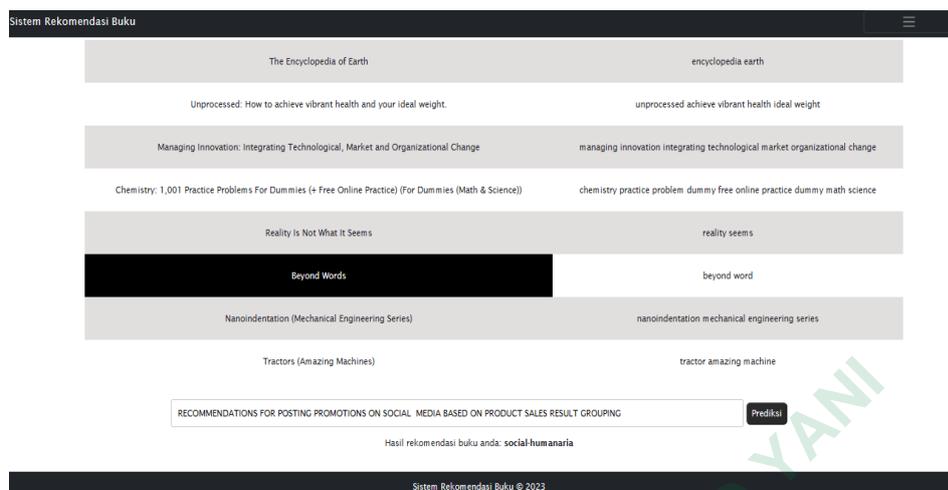
Implementasi dari halaman dashboard ini menampilkan grafik berupa *Pie Charts* yang memberikan informasi terkait jumlah dari kategori buku, kemudian dibawah dari grafik terdapat formulir seleksi yang dapat dilakukan pengguna untuk memilih kategori apa yang ingin dipastikan dalam pra-pemrosesan data pada penelitian serta terdapat formulir masukkan guna menginisiasi klasifikasi oleh pengguna. Berikut Gambaran implementasi melalui Gambar 4.5, Gambar 4.6, dan Gambar 4.7



Gambar 4.5 Implementasi grafik *dashboard*

Judul	Hasil pra-pemrosesan
The 5 Minute Motivator: Learn the Secrets to Success, Health, and Happiness	minute motivator learn secret success health happiness
Challenging Global Inequality: Development Theory and Practice in the 21st Century	challenging global inequality development theory practice st century
The Encyclopedia of Earth	encyclopedia earth
Unprocessed: How to achieve vibrant health and your ideal weight.	unprocessed achieve vibrant health ideal weight
Managing Innovation: Integrating Technological, Market and Organizational Change	managing innovation integrating technological market organizational change
Chemistry: 1,001 Practice Problems For Dummies (+ Free Online Practice) (For Dummies (Math & Science))	chemistry practice problem dummy free online practice dummy math science
Reality Is Not What It Seems	reality seems
Beyond Words	beyond word
Nanninfantation (Merkhaniral Fominaeinn Series)	nanninfantation merkhaniral fominaeinn series

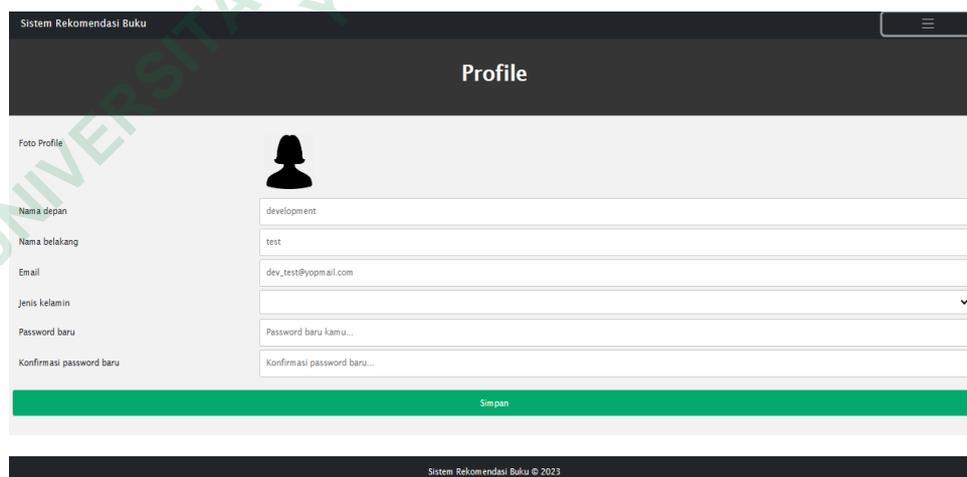
Gambar 4.6 Implementasi *Dashboard* Tabel kategori



Gambar 4.7 Implementasi *Dashboard* inisiasi rekomendasi

#### 4. Halaman *Profile*

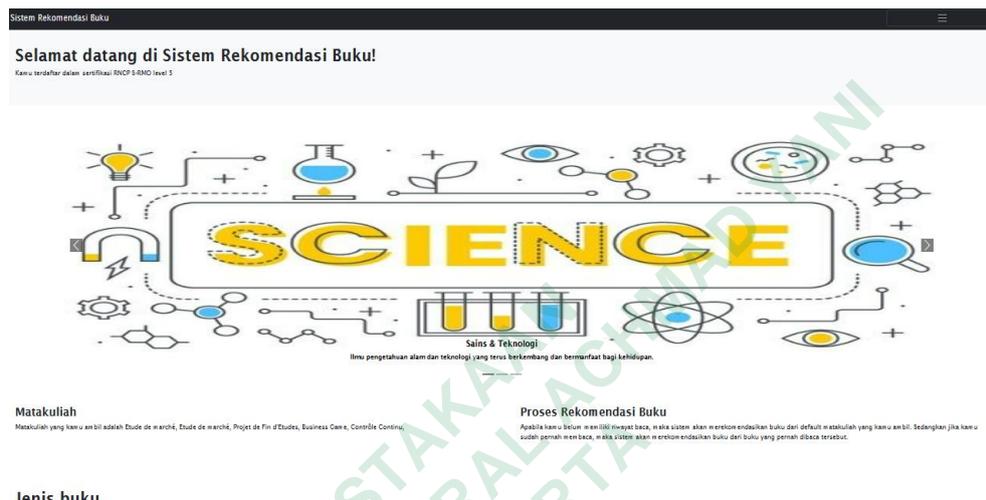
Implementasi halaman *Profile* berguna untuk pengguna dalam mengubah identitas diri seperti email, password, jenis kelamin, nama depan serta nama belakang. Dalam perubahan password, sistem meminta untuk pengguna memasukkan konfirmasi password sebagai validasi sistem dalam menyimpan data. Berdasarkan implementasinya, sistem belum mengimplementasikan perubahan foto identitas. Berikut Gambaran implementasi melalui Gambar 4.8



Gambar 4.8 Implementasi halaman *Profile*

## 5. Halaman Informasi Umum

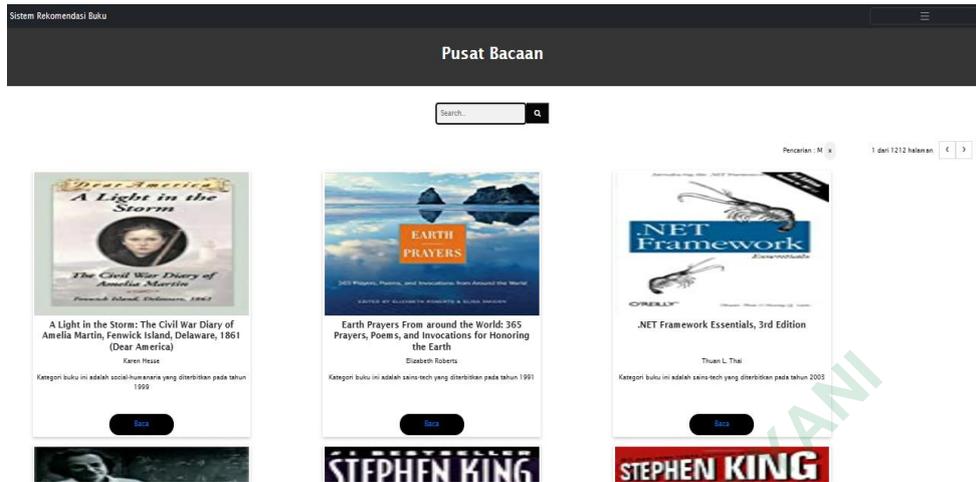
Halaman ini berguna dalam menampilkan informasi terkait sistem dan nama kelas, tingkatan, nama RNCP dan matakuliah dari pengguna melalui otorisasi token yang disimpan di dalam penyimpanan local *browser*. Berikut Gambaran implementasi melalui Gambar 4.9



Gambar 4.9 Implementasi halaman Informasi Umum

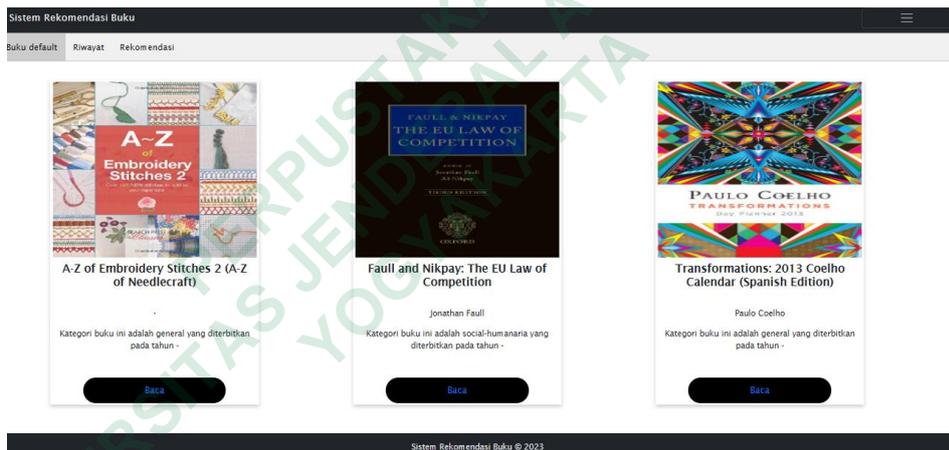
## 6. Halaman Pusat Bacaan

Implementasi halaman Pusat Bacaan adalah menampilkan keseluruhan buku yang berjumlah 18.000 (delapan belas ribu) dari berbagai topik dan kategori. Pengguna dapat mencari buku melalui formulir pencarian dengan pola kata karena algoritma pencarian pada sistem ini menggunakan modulasi *Regular Expression*. Setiap buku yang ditampilkan terdapat tombol untuk pengguna baca, dan sistem akan mengarahkan ke laman dari Google Books berdasarkan *International Standard Book Number* atau ISBN dari buku tersebut, selain mengarahkan pengguna ke dalam laman Google Books, sistem juga menyimpannya sebagai riwayat bacaan pengguna. Berikut Gambaran implementasi melalui Gambar 4.10

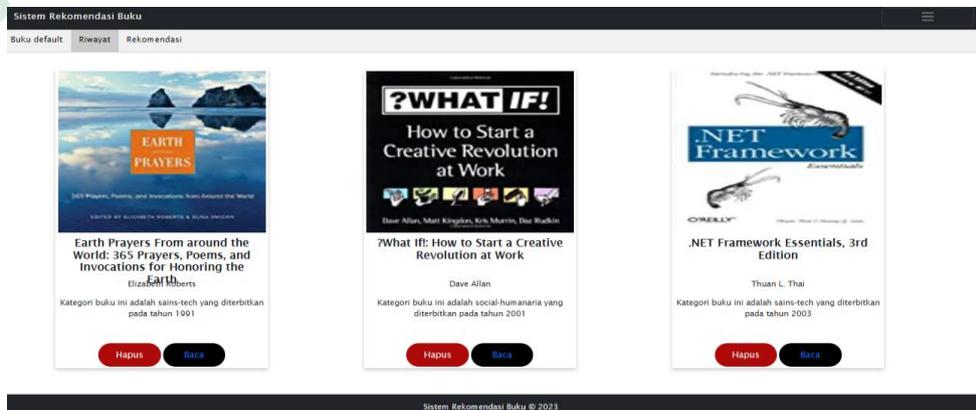


Gambar 4.10 Implementasi halaman Pusat Bacaan

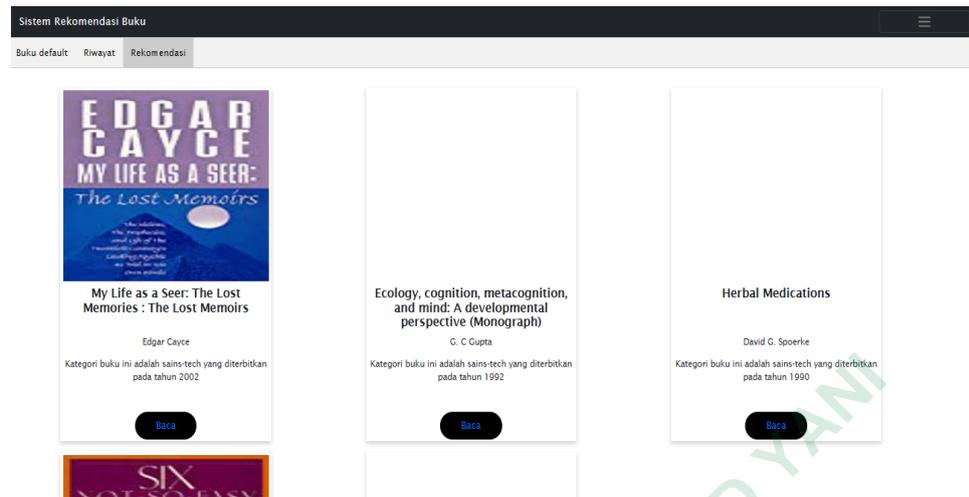
## 7. Halaman Pustaka Rekomendasi



Gambar 4.11 Implementasi Pustaka Rekomendasi buku default



Gambar 4.12 Implementasi Pustaka Rekomendasi riwayat



Gambar 4.13 Implementasi halaman Pustaka Rekomendasi inisiasi

Pada Gambar 4.11 sampai Gambar 4.13 merupakan implementasi antarmuka dari fitur Pustaka rekomendasi. Pada fitur tersebut, ada tiga *tabs* yaitu buku default yang merupakan buku bacaan matakuliah, riwayat yaitu buku yang telah dibaca oleh pengguna, dan rekomendasi yaitu hasil dari inisiasi rekomendasi. Inisiasi rekomendasi didapatkan oleh sistem yang berasal dari buku bacaan matakuliah, apabila pengguna telah membaca buku dan tersimpan di dalam riwayat maka inisiasi rekomendasi diperoleh berdasarkan riwayat bacaan pengguna.

### 4.3 BASIS DATA

Pada penelitian ini, basis data yang dipilih adalah berbasis dokumen tidak yaitu MongoDB dengan menggunakan penyedia layanan MongoDB Atlas (MongoDB, Inc, n.d.). Basis data dinamakan **db-skripsi** yang terdiri dari koleksi *student*, *subject*, *rncp\_titles*, *books*, dan *book\_recommenders* seperti pada Gambar 4.14. Sedangkan pada Gambar 4.15 sampai Gambar 4.19 merupakan skema dan tipe dari setiap koleksi basis data.



Gambar 4.14 Implementasi basis data

Pada Gambar 4.15 di sisi kanan adalah tipe data sedangkan tipe kiri adalah nama *field* dari dokumen, skema *book\_recommenders* digunakan untuk mengumpulkan data buku yang diperoleh dari Google Books yang berjumlah 18,000. Data buku ini ditampilkan oleh fungsi *GetAllBooks* pada GraphQL *query* sedangkan untuk *User Interface* buku ini ditampilkan pada fitur Pusat Bacaan.

<b>_id</b>	643	ObjectId
<b>ISBN</b>	A18	String
<b>status</b>	acti	String
<b>book_author</b>	Lau	String
<b>book_title</b>	The	String
<b>image_url_l</b>	http	String
<b>image_url_m</b>	http	String
<b>image_url_s</b>	http	String
<b>publisher</b>	Safi	String
<b>year_publication</b>	1,99	Int32
<b>category_type</b>	sain	String

Gambar 4.15 Skema koleksi *book\_recommenders*

Pada Gambar 4.16 di sisi kanan adalah tipe data sedangkan tipe kiri adalah nama *field* dari dokumen, skema koleksi *books* digunakan dalam penampilan data terkait buku yang berasal dari matakuliah pelajar. Tampilan ini disebut buku *default* dari fitur Pustaka Rekomendasi.

 <b>_id</b>	640	ObjectId
 <b>google_id</b>	620	Int32
 <b>image</b>	006	String
 <b>url_image</b>	http	String
 <b>title</b>	Mur	String
 <b>recommender_count</b>	17	Int32
 <b>category</b>	Mys	String
 <b>status</b>	acti	String
 <b>category_type</b>	gen	String
 <b>author</b>	Aga	String

Gambar 4.16 Skema koleksi books

Pada Gambar 4.17 di sisi kanan adalah tipe data sedangkan tipe kiri adalah nama *field* dari dokumen, koleksi ini untuk memberi keterangan terkait **\_id** dari koleksi tersebut yang menjadi referensi ke koleksi *students*. Data dari koleksi *rncp\_titles* juga digunakan untuk menampilkan informasi terkait pelajar yang masuk sebagai pengguna pada fitur Informasi Umum

 <b>_id</b>	61f	ObjectId
 <b>short_name</b>	S-M	String
 <b>long_name</b>	Mar	String
 <b>rncp_code</b>	351	String
 <b>rncp_level</b>	7	String
 <b>rncp_level_europe</b>	7	String
 <b>rncp_logo</b>	null	Null
 <b>is_published</b>	true	Bool
 <b>is_certifier_also_pc</b>	false	Bool
 <b>journal_text</b>	<p>	String
 <b>preparation_centers</b>	Arre	Array
 <b>classes</b>	Arre	Array
 <b>status</b>	acti	String
 <b>specializations</b>	Arre	Array
 <b>journal_date</b>	20/0	String
 <b>certifier</b>	596	ObjectId
 <b>admtc_dir_responsible</b>	5a3	ObjectId
 <b>year_of_certification</b>	202	String
 <b>name_of_signatory</b>	Eric	String
 <b>createdAt</b>	1/2	Date

Gambar 4.17 Skema koleksi rncp\_titles

Pada Gambar 4.18 di sisi kanan adalah tipe data sedangkan tipe kiri adalah nama *field* dari dokumen, koleksi ini digunakan sebagai identitas pengguna yaitu pelajar ketika melakukan *Login* atau proses otentikasi dan otorisasi. Data ini bisa diubah dengan cara lupa kata sandi atau update profil pada Profile.

 _id	5a8 ObjectId
 first_name	dev String
 last_name	test String
 email	dev String
 status	acti String
 rncp_title	594 ObjectId
▷  book_interest	Arr Array

Gambar 4.18 Skema koleksi students

Gambar 4.19 di sisi kanan adalah tipe data sedangkan tipe kiri adalah nama *field* dari dokumen, koleksi ini adalah kumpulan data matakuliah oleh pengguna dari pelajar. Data matakuliah akan ditampilkan pada fitur Informasi Umum.

 _id	638e ObjectId
 subject_name	Recc String
 status	activ String
▷  books	Array Array

Gambar 4.19 Skema koleksi subjects

#### 4.4 FITUR-FITUR SISTEM

Sistem Rekomendasi Buku memiliki fitur seperti

1. Akses (*login* dan *logout*).
2. Kotak dialog formulir lupa kata sandi
3. Menampilkan grafik dan Tabel pra-pemrosesan data serta inisiasi rekomendasi (*Dashboard*).
4. Mengubah profil (*Profile*).
5. Menampilkan nama kelas, nama RNCP, tingkatan, dan matakuliah serta informasi terkait sistem (Informasi Umum).
6. Menampilkan seluruh buku dan menambahkan riwayat bacaan (Pusat bacaan).

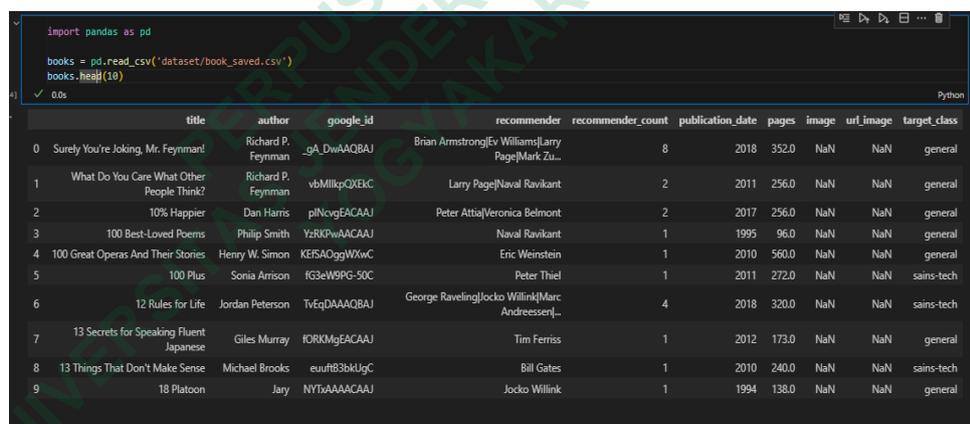
- Menampilkan buku bawaan yang dari matakuliah, riwayat bacaan dan buku rekomendasi serta menambah dan menghapus riwayat buku (Pustaka rekomendasi).

## 4.5 PEMBAHASAN

Pada bagian pembahasan ini menguraikan hasil dari penelitian yang sudah dilaksanakan berdasarkan dari langkah-langkah pada sub-bab 3.2 Jalan Penelitian, seperti.

### 4.5.1 Pengumpulan Data

Seperti yang sudah disebutkan bahwa data dari penelitian ini diperoleh dari platform Kaggle berjumlah 8702 dengan , maka data tersebut diunduh dan disimpan dalam ekstensi (.csv) pada folder yang sudah ditentukan dan dimuat menggunakan modulasi *Pandas* dari bahasa pemrograman Python. Berikut cuplikan kode dan keluaran untuk melihat sampel data buku tersebut dari Gambar 4.20.



```

import pandas as pd

books = pd.read_csv('dataset/book_saved.csv')
books.head(10)

```

	title	author	google_id	recommender	recommender_count	publication_date	pages	image	url_image	target_class
0	Surely You're Joking, Mr. Feynman!	Richard P. Feynman	_gA_DwAAQBAJ	Brian Armstrong Ev Williams Larry Page Mark Zu...	8	2018	352.0	NaN	NaN	general
1	What Do You Care What Other People Think?	Richard P. Feynman	vbMllkpQXEK	Larry Page Naval Ravikant	2	2011	256.0	NaN	NaN	general
2	10% Happier	Dan Harris	plNcvgEACAAJ	Peter Attia Veronica Belmont	2	2017	256.0	NaN	NaN	general
3	100 Best-Loved Poems	Philip Smith	YzRKPwaACAAJ	Naval Ravikant	1	1995	96.0	NaN	NaN	general
4	100 Great Operas And Their Stories	Henry W. Simon	KEFAOggWwxC	Eric Weinstein	1	2010	560.0	NaN	NaN	general
5	100 Plus	Sonia Arrison	fG3eW9PG-30C	Peter Thiel	1	2011	272.0	NaN	NaN	sains-tech
6	12 Rules for Life	Jordan Peterson	TvEqDAAAQBAJ	George Raveling Jocko Willink Marc Andreessen ...	4	2018	320.0	NaN	NaN	sains-tech
7	13 Secrets for Speaking Fluent Japanese	Giles Murray	fORKMgEACAAJ	Tim Ferris	1	2012	173.0	NaN	NaN	general
8	13 Things That Don't Make Sense	Michael Brooks	euufR3bbkJgC	Bill Gates	1	2010	240.0	NaN	NaN	sains-tech
9	18 Platoon	Jary	NYtAAAAAAJ	Jocko Willink	1	1994	138.0	NaN	NaN	general

Gambar 4.20 Cuplikan kode dan keluaran sebelum pra-pemrosesan

### 4.5.2 Pre-processing

Pada tahapan ini, data yang dimuat oleh Pandas disimpan dalam variabel *books*. Ada beberapa langkah di dalam pra-pemrosesan seperti

- Menghilangkan duplikasi data berdasarkan kolom *title*.

Pada bagian ini jumlah data awal yang berjumlah 8702 berkurang menjadi 8648 data. Seperti penguraian pada Gambar 4.21 dan Gambar 4.22 yang merupakan keterangan sebelum dan sesudah menghilangkan duplikasi data.

```

books = pd.read_csv('dataset/book_saved.csv')
books.info()
✓ 0.1s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8702 entries, 0 to 8701

```

Gambar 4.21 Sebelum menghilangkan duplikasi

```

books = books[~books.duplicated(subset=['title'])]
books = books[['title', 'recommender_count', 'target_class']]
✓ 0.0s

books.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8648 entries, 0 to 8701

```

Gambar 4.22 Setelah menghilangkan duplikasi

## 2. Fungsionalitas dalam pra-pemrosesan

Pada bagian ini data yang konsisten dari duplikasi, kemudian dilakukannya pra-pemrosesan seperti kalimat diubah ke dalam bentuk standar menjadi huruf kecil, menghilangkan karakter dan angka, tokenisasi, normalisasi dan menghilangkan kata yang termasuk ke dalam *stopword* atau kata yang diabaikan dan dihilangkan sehingga jumlah data berkurang menjadi 8637 data. Seperti pada Gambar 4.23 dan Gambar 4.24 merupakan cuplikan kode dan Gambar 4.25 yang merupakan keluaran dari hasil fungsionalitas ini. Pada Gambar 4.26 merupakan tambahan informasi terkait jumlah pengelompokkan label setelah pra-pemrosesan.

```

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import string
from nltk.stem import WordNetLemmatizer

# merubah menjadi lowercase
def casefolding(text):
    return text.casefold()

# menghilangkan karakter yang tidak berfungsi dan angka pada teks
def removeCharNumber(text):
    for i in string.punctuation:
        text = text.replace(i).format(i, '')
    if text.isnumeric():
        return text
    else:
        return ''.join(i for i in text if not i.isdigit())

# menghitung stopwords dan menghilangkannya serta tokenisasi teks dalam kata berdasarkan pembatas
def word_token(text:str):
    stopWords = set(stopwords.words('english'))
    word_tokens = word_tokenize(text)
    arrays = [w for w in word_tokens if w not in stopWords]
    return arrays

# normalisasi teks untuk menjaga kosa kata, menghilangkan kata tidak jelas, dan meningkatkan akurasi kata
def lemmatizing(word):
    wn1 = WordNetLemmatizer()
    return ' '.join([wn1.lemmatize(words) for words in word])

# memisahkan kata menjadi bentuk list
def splitter(text):
    return text.split()

```

Gambar 4.23 Cuplikan kode fungsi

Gambar 4.23 merupakan cuplikan kode fungsi yang digunakan selama tahapan pra-pemrosesan seperti perubahan huruf *lowercase*, menghilangkan karakter atau angka yang informasinya sangat rendah, tokenisasi kalimat menjadi kata, lematisasi kalimat untuk menghilangkan kata tidak jelas dan meningkatkan akurasi kata. Setelah itu kalimat diubah menjadi kata yang diurutkan melalui tipe data *list*.

```

containers = []
for title in books['title']:
    title = casefolding(title)
    title = removeCharNumber(title)
    title = word_token(title)
    title = lemmatizing(title)
    title = splitter(title)
    containers.append(title)
books['title_preprocessing'] = containers
books['text_cleaning'] = books['title_preprocessing'].apply(lambda x: ' '.join(x))

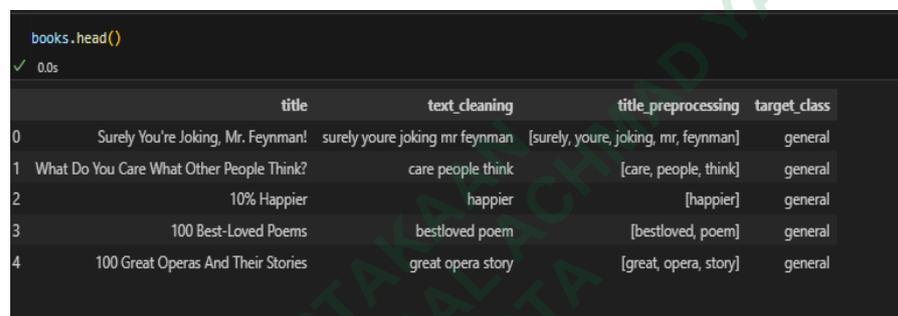
```

Gambar 4.24 Cuplikan kode pemanfaatan fungsi

Gambar 4.24 merupakan cuplikan kode pemanfaatan fungsi dari pra-pemrosesan menggunakan perulangan. Dari gambar tersebut terlihat bahwa perulangan terjadi pada variabel *books* dengan kata kunci dari kolom *title* atau judul

buku dan diinisiasi dengan variabel *title*. Urutan pemanggilan fungsi dan nilai parameter dilakukan sesuai dengan tahapan pra-pemrosesan yang dibutuhkan, kemudian hasil dari pra-pemrosesan disimpan ke dalam variabel *containers*. Tujuannya adalah untuk menambahkan satu kolom baru dengan teknik penambahan data secara objektif yaitu *books['text\_cleaning']*.

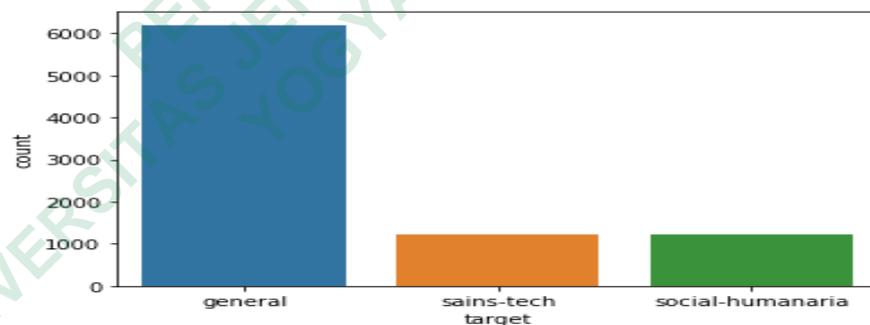
Gambar 4.25 memperlihatkan hasil dari pra-pemrosesan berupa tabel dengan tampilan dari *dataframe* Pandas.



```
books.head()
✓ 0.0s
```

	title	text_cleaning	title_preprocessing	target_class
0	Surely You're Joking, Mr. Feynman!	surely youre joking mr feynman	[surely, youre, joking, mr, feynman]	general
1	What Do You Care What Other People Think?	care people think	[care, people, think]	general
2	10% Happier	happier	[happier]	general
3	100 Best-Loved Poems	bestloved poem	[bestloved, poem]	general
4	100 Great Operas And Their Stories	great opera story	[great, opera, story]	general

Gambar 4.25 Keluaran hasil pra-pemrosesan



Gambar 4.26 Grafik jumlah pengelompokkan label

Gambar 4.26 merupakan tambahan informasi terkait jumlah keseluruhan data berdasarkan label. Data tersebut terdiri dari tiga label yaitu *general*, *sains-tech*, dan *social-humanaria*. Rincian dari data-data itu adalah *sains-tech* berjumlah 1218 data, *social-humanaria* berjumlah 1217 data sedangkan *general* berjumlah 6271 data.

### 3. Transformasi TF-IDF

Pada bagian ini, TF-IDF teks dihitung dalam pembobotan frekuensi kemunculannya. TF-IDF yang digunakan adalah *Unigrams* yang memisahkan kalimat dari hasil pra-pemrosesan sebelumnya menjadi setiap kata, seperti *care people think* menjadi [*care, people, think*]. Berikut cuplikan kode dan keluaran hasil seperti pada Gambar 4.27, Gambar 4.28, dan tambahan informasi pemetaan transformasi teks menggunakan PCA pada Gambar 4.29

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(
    min_df=1, # min count for relevant vocabulary
    strip_accents='unicode', # replace all accented unicode char
    # by their corresponding ASCII char
    analyzer='word', # features made of words
    token_pattern=r'\w{1,}', # tokenize only words of 4+ chars
    ngram_range=(1, 1), # features made of a single tokens
    use_idf=True, # enable inverse-document-frequency reweighting
    smooth_idf=True, # prevents zero division for unseen words
    sublinear_tf=False
)
sparse = tfidf.fit_transform(df.text_cleaning.values.tolist())
print(sparse)
```

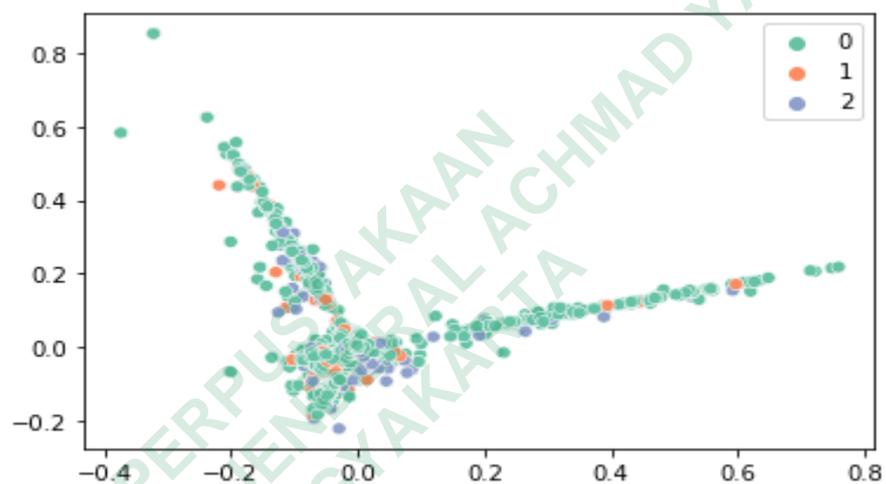
Gambar 4.27 Cuplikan kode transformasi teks

Gambar 4.27 menunjukkan bahwa tahap transformasi data atau pembobotan kata dilakukan menggunakan TF-IDF *unigram*, dengan kosa kata yang berkaitan minimal 1, serta beberapa parameter yang dimanfaatkan seperti yang terlihat pada gambar tersebut.

```
(0, 3542) 0.4174952675479513
(0, 6440) 0.406845433847043
(0, 5153) 0.4973184530513057
(0, 10732) 0.406845433847043
(0, 9424) 0.4973184530513057
(1, 9688) 0.6318529496341289
(1, 7224) 0.5343513489430802
(1, 1437) 0.5614539036478075
(2, 4302) 1.0
(3, 7434) 0.6021847675629504
(3, 922) 0.7983567534098748
(4, 9278) 0.4190470287097017
(4, 6892) 0.7530716730270911
(4, 4129) 0.5072303648380526
(5, 7423) 1.0
(6, 5547) 0.5583399178271121
(6, 8347) 0.8296122806231918
(7, 5092) 0.4630422241041315
(7, 3659) 0.6094633225232715
(7, 9062) 0.5279852440176535
(7, 8570) 0.3679374121983425
```

Gambar 4.28 Keluaran hasil transformasi teks

Gambar 4.28 menampilkan hasil keluaran dari TF-IDF yang sebelumnya sudah dijelaskan pada Gambar 4.27. Data berupa teks tersebut diubah menjadi pembobotan kata dengan dimensi yang berbeda-beda. Karena perbedaan dimensi tersebut, penelitian ini memetakan hasil dari pembobotan kata menggunakan konsep PCA. Tujuannya agar dimensi tersebut menjadi standar tanpa mengurangi sedikit informasi terkait data tersebut seperti pada Gambar 4.29



Gambar 4.29 Pemetaan transformasi kata

#### 4.5.3 Pelatihan data

Sebelum membahas hasil penelitian terkait penyesuaian model SVC, peneliti memberi penjelasan bagaimana *Cosine Similarity* memberikan rekomendasi buku dan menghitung kesamaan dari sudut *kosinus*. Dengan tahap awal, peneliti membaca *dataframe* dari hasil transformasi kata dengan *index* teks judul buku dan kolom nama fitur dari penyesuaian transformasi TF-IDF. Seperti pada Gambar 4.30 dan Gambar 4.31 menguji *Cosine Similarity* dengan fungsionalitas dan menghitung sudut kesamaan dari suatu teks serta memberikan rekomendasi berdasarkan nilai yang berurutan.

```

table = pd.DataFrame(sparse.toarray(), index=documents, columns=tfidf.get_feature_names())
table.head()

```

	0	000	001	01	02	0353	0354	0391	04	0439	...	zoo	zoonosi	zoos	zorba	zoroaster	zoroastrian	zoroastrianism	zoroastro	zozzy	zucked
100 Plus	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12 Rules for Life	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13 Things That Don't Make Sense	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
59 Seconds	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A Brief History of Time	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```

5 rows x 12167 columns

```

```

similarity_scores = cosine_similarity(table)
similarity_scores

```

```

array([[1.0, 0.0, 0.0, ..., 0.0, 0.0,
        0.0, 1.0, 0.0, ..., 0.0, 0.0,
        0.0, 0.0, 1.0, ..., 0.0, 0.0,
        ...,
        0.0, 0.0, 0.0, ..., 1.0, 0.0,
        0.0338396],
       [0.0, 0.0, 0.0, ..., 0.0, 1.0,
        0.01140736],
       [0.0, 0.0, 0.0, ..., 0.0338396, 0.01140736,
        1.0]])

```

Gambar 4.30 *Cosine Similarity* menghitung kesamaan kata dari kalimat

```

def recommend(book):
    array = []
    book_index = np.where(table.index == book)[0][0]
    distances = similarity_scores[book_index]
    book_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])

    for i in book_list:
        obj = {}
        obj['akurasi'] = i[1]
        obj['title'] = table.index[i[0]]
        obj['label'] = df[df['title'] == table.index[i[0]]]['target_class'].values.tolist()[0]
        array.append(obj)
    return array

array = recommend("Chemistry: Concepts & Applications, Student Edition")
array

```

```

7.2s

```

```

{'akurasi': 1.0,
 'title': 'Chemistry: Concepts & Applications, Student Edition',
 'label': 'sains-tech'},
{'akurasi': 0.5188934782218979,
 'title': 'Theories of Development: Concepts and Applications',
 'label': 'sains-tech'},

```

Gambar 4.31 Urutan jarak terdekat dari suatu teks

Setelah menguraikan terkait *Cosine Similarity* seperti Gambar 4.30 dan Gambar 4.31, buku diurutkan berdasarkan hasil *kosinus* tertinggi dan hasil tersebut menjadi dimensi baru untuk tahap klasifikasi SVC. Setelah menganalisa nilai *Cosine Similarity*, tahap berikutnya adalah memisahkan data latih dan data uji dari

hasil pra-pemrosesan menggunakan *train\_test\_split* seperti pada Gambar 4.32 sebelum membangun model SVC

```
x_train, x_test, y_train, y_test = train_test_split(df['text_cleaning'].values.tolist(), labels, test_size=0.25)
```

Gambar 4.32 Pemisahan data latih dan data uji

Kemudian peneliti membangun *pipeline* data di Gambar 4.33 dan Gambar 4.34 dengan dua parameter yaitu TF-IDF dan model SVC. Untuk pembahasan saat ini memberikan penguraian Gambar terkait perbandingan SVC menggunakan parameter  $C=0.05$  dan  $C=1$  dengan fitur yaitu *x\_train* dan label *y\_train*.

```
model = make_pipeline(
    TfidfVectorizer(
        min_df=1, # min count for relevant vocabulary
        strip_accents='unicode', # replace all accented unicode char
        # by their corresponding ASCII char
        analyzer='word', # features made of words
        token_pattern='\w{1,}', # tokenize only words of 4+ chars
        ngram_range=(1, 1), # features made of a single tokens
        use_idf=True, # enable inverse-document-frequency reweighting
        smooth_idf=True, # prevents zero division for unseen words
        sublinear_tf=False
    ),
    SVC(kernel='cosine_similarity', C=1)
)
model.fit(x_train, y_train)

Pipeline(memory=None,
steps=[('tfidfvectorizer',
TfidfVectorizer(analyzer='word', binary=False,
decode_error='strict',
dtype=<class 'numpy.float64'>,
encoding='utf-8', input='content',
lowercase=True, max_df=1.0, max_features=None,
min_df=1, ngram_range=(1, 1), norm='l2',
preprocessor=None, smooth_idf=True,
stop_words=None, strip_accents='unicode',
sublinear_tf=False, ..tern='\w{1,}',
tokenizer=None, use_idf=True,
vocabulary=None)),
('svc',
SVC(C=1, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0, decision_function_shape='ovr', degree=3,
gamma='scale',
kernel=function cosine_similarity at 0x0000019814008950>,
max_iter=1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)],
verbose=False)
```

Gambar 4.33 SVC dengan  $C=1$

```
model_value_min = make_pipeline(
    TfidfVectorizer(
        min_df=1, # min count for relevant vocabulary
        strip_accents='unicode', # replace all accented unicode char
        # by their corresponding ASCII char
        analyzer='word', # features made of words
        token_pattern='\w{1,}', # tokenize only words of 4+ chars
        ngram_range=(1, 1), # features made of a single tokens
        use_idf=True, # enable inverse-document-frequency reweighting
        smooth_idf=True, # prevents zero division for unseen words
        sublinear_tf=False
    ),
    SVC(kernel='cosine_similarity', C=0.05)
)
model_value_min.fit(x_train, y_train)

Pipeline(memory=None,
steps=[('tfidfvectorizer',
TfidfVectorizer(analyzer='word', binary=False,
decode_error='strict',
dtype=<class 'numpy.float64'>,
encoding='utf-8', input='content',
lowercase=True, max_df=1.0, max_features=None,
min_df=1, ngram_range=(1, 1), norm='l2',
preprocessor=None, smooth_idf=True,
stop_words=None, strip_accents='unicode',
sublinear_tf=False, ..ern='\w{1,}',
tokenizer=None, use_idf=True,
vocabulary=None)),
('svc',
SVC(C=0.05, break_ties=False, cache_size=200,
class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='scale',
kernel=function cosine_similarity at 0x0000019814008950>,
max_iter=1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)],
verbose=False)
```

Gambar 4.34 SVC dengan  $C=0.05$

#### 4.5.4 Pengujian data

Setelah pelatihan data dengan membentuk *pipeline* dari TF-IDF dan model. Maka langkah selanjutnya adalah pengujian data dengan menggunakan data uji sebanyak, menghitung akurasi dan mengevaluasi model menggunakan *Confusion Matrix* dengan modulasi *classification\_report* untuk membaca beberapa nilai yaitu *accuracy*, *precision*, *f1-score*, dan *recall*. Diperoleh saat pengujian data, akurasi model yang diberikan adalah 0.8842 atau 88.4% untuk  $C=1$  sedangkan  $C=0.05$  memberikan akurasi 0.722 atau 72.2%. Berikut penguraian akurasi dan evaluasi yang diuraikan melalui Gambar 4.35 dan Gambar 4.36

```

y_pred_test = model.predict(x_test)
print('model accuracy score with cosine similarity: {0} -> {1}'.format(accuracy_score(y_test, y_pred_test), y_pred_test))
✓ 0.2s
model accuracy score with cosine similarity: 0.8842592592592593 -> [0 0 0 ... 0 0 0]

print('classification report for C=1\n', classification_report(y_test, y_pred_test, target_names=['sains-tech', 'social-humanaria', 'general']))
✓ 0.0s
classification report for C=1
      precision    recall  f1-score   support

sains-tech      0.88      0.99      0.93      1549
social-humanaria 0.90      0.52      0.66       300
general         0.94      0.69      0.80       311

 accuracy                   0.88      2160
 macro avg                 0.90      0.74      0.80      2160
 weighted avg              0.89      0.88      0.87      2160

```

Gambar 4.35 Pengujian data dan evaluasi  $C=1$

```

y_pred_test_model_min = model_value_min.predict(x_test)
print('model accuracy score with cosine similarity: {0} -> {1}'.format(accuracy_score(y_test, y_pred_test_model_min), y_pred_test_model_min))
✓ 0.1s
model accuracy score with cosine similarity: 0.7226851851851852 -> [0 0 0 ... 0 0 0]

print('classification report for C=0.05\n', classification_report(y_test, y_pred_test_model_min, target_names=['sains-tech', 'social-humanaria', 'general']))
✓ 0.0s
classification report for C=0.05
      precision    recall  f1-score   support

sains-tech      0.72      1.00      0.84      1549
social-humanaria 0.00      0.00      0.00       300
general         1.00      0.04      0.07       311

 accuracy                   0.72      2160
 macro avg                 0.57      0.35      0.30      2160
 weighted avg              0.66      0.72      0.61      2160

```

Gambar 4.36 Pengujian data dan evaluasi  $C=0.05$

SVC menggunakan nilai parameter  $C$  untuk mengoptimalkan SVM dalam menghindari kesalahan klasifikasi. Untuk pengoptimalan ini, SVC bergantung pada besarnya nilai  $C$  dalam memilih *hyperplane*. Maka untuk memberikan klasifikasi

dalam Sistem Rekomendasi Buku, maka peneliti memilih model dengan parameter  $C=1$ . Pada Gambar 4.37 merupakan hasil dari validasi model menggunakan *cross\_val\_score* yang memberikan hasil evaluasi yang tidak jauh berbeda dengan evaluasi sebelumnya yaitu 0.8804 atau 88.04%.

```

table_valid = pd.read_csv('dataset/book_saved.csv')
scores = cross_val_score(model, table_valid.title, table_valid.target_class)
✓ 8.5s

print("accuracy %0.2f (+/- %0.2f)", scores.mean(), scores.std() * 2)
✓ 0.0s
accuracy %0.2f (+/- %0.2f) 0.880491724269973 0.03610369001895047

```

Gambar 4.37 Hasil validasi model

Pada Gambar 4.38 memberikan hasil klasifikasi suatu teks judul buku. Hal ini bertujuan untuk menguji apakah model dapat memprediksi masukkan pengguna dengan nilai apapun.

```

text = ['Chemistry: Concepts & Applications, Student Edition']
text_pred = model.predict(text)
text_pred
✓ 0.0s

array(['sains-tech'], dtype=object)

```

Gambar 4.38 Menguji model dengan masukkan

Ini membuktikan bahwa *Cosine Similarity* sebagai kernel dari SVC membantu mentransformasikan data ke ruang dimensi lebih tinggi berdasarkan nilai *kosinus* yang paling mendekati nilai satu yaitu *sains-tech* yang merupakan hasil klasifikasi dari prediksi model. Bukti lainnya juga mengarah pada Gambar 4.26 yaitu nilai rekomendasi *Cosine Similarity* tertinggi terdapat pada label *sains-tech* dengan nilai *kosinusnya* adalah 0.707 yang berarti nilai tersebut mendekati angka satu sedangkan sudutnya adalah  $45.00865^\circ$ .

Setelah model yang disesuaikan diuji, maka peneliti melakukan ekspor model menggunakan modulasi *pickle*. Hasil dari ekspor disimpan dalam ekstensi (*.pkl*) dan dapat digunakan kembali saat diimplementasikan ke dalam sebuah layanan API pada Sistem Rekomendasi Buku. Implementasi ekspor model terdapat pada Gambar 4.39

```
import pickle5 as pickle

pickle.dump(model, open('result/model.pkl', 'wb'))
```

Gambar 4.39 Cuplikan kode ekspor model

#### 4.5.5 Pembuatan Sistem Rekomendasi

Pada pembahasan ini menguraikan penjelasan Sistem Rekomendasi Buku dibangun dengan bahasa pemrograman NodeJS dan mengimplementasikan model yang diekspor di pembahasan sebelumnya dalam API yaitu pemanggilan sebuah fungsionalitas API GraphQL menggunakan query *BookRecommended* yang diautorisasi oleh kredensial pelajar yang terdaftar pada basis data Sistem Rekomendasi. Untuk menginisiasi rekomendasi, peneliti menggunakan modulasi Python *Shell* untuk sebagai alat bantu berupa modulasi yang dapat membaca kode program berbahasa Python dari NodeJS. Karena umumnya, NodeJS belum mendukung modulasi untuk membaca file berisi *matrix* dari model *pickle* yang merupakan hasil ekspor, sehingga untuk melakukan pemuatan model dan prediksi dari masukkan pengguna, diperlukan dua bahasa pemrograman.

Berdasarkan Gambar 4.40 adalah *query* dalam memperoleh *response* dari layanan API yang berfungsi untuk menginisiasi atau mengimplementasikan model yang sudah diekspor sebelumnya dan Gambar 4.41, Gambar 4.42, dan Gambar 4.43 merupakan *response* atau tampilan data yang diberikan.

```

1  mutation {
2    BookRecommended{
3      book_default{
4        book_id{
5          title
6          author
7          category_type
8          publication_date
9          url_image
10         google_id
11       }
12     }
13     book_history{
14       book_id{
15         book_title
16         image_url_m
17         book_author
18         category_type
19         year_publication
20         ISBN
21       }
22     }
23     book_recommender{
24       book_title
25       image_url_m
26       image_url_l
27       book_author
28       category_type
29       year_publication
30       ISBN
31       _id
32     }
33   }
34 }

```

Gambar 4.40 Query mutasi dari *BookRecommended*

Gambar 4.40 adalah *query* pemanggilan mutasi dari fungsi *BookRecommended* oleh GraphQL API. Mutasi secara metode dari layanan API adalah *Post*. Secara umum, mutasi digunakan untuk melakukan proses penyimpanan, penghapusan, perubahan dan proses lain yang dianggap penting oleh pengembang. Pada penelitian ini, mutasi *BookRecommended* dimanfaatkan oleh fitur inisiasi rekomendasi yaitu Pustaka Rekomendasi.

```

{
  "data": {
    "BookRecommended": {
      "book_default": [
        {
          "book_id": {
            "title": "A-Z of Embroidery Stitches 2 (A-Z of Needlecraft)",
            "author": "",
            "category_type": "general",
            "publication_date": null,
            "url_image": "http://ecx.images-amazon.com/images/I/61k1Dw%2BcyL.jpg",
            "google_id": "1782211691"
          }
        },
        {
          "book_id": {
            "title": "Faull and Nikpay: The EU Law of Competition",
            "author": "Jonathan Faull",
            "category_type": "social-humanaria",
            "publication_date": null,
            "url_image": "http://ecx.images-amazon.com/images/I/411R%2BaMKu3L.jpg",
            "google_id": "199665095"
          }
        },
        {
          "book_id": {
            "title": "Transformations: 2013 Coelho Calendar (Spanish Edition)",
            "author": "Paulo Coelho",
            "category_type": "general",
            "publication_date": null,
            "url_image": "http://ecx.images-amazon.com/images/I/618hUYwzSyl.jpg",
            "google_id": "307949052"
          }
        }
      ]
    }
  }
},
],

```

Gambar 4.41 *Response data mutasi BookRecomended*

```

,,
"book_history": [
  {
    "book_id": {
      "book_title": "A Change of Climate",
      "image_url_m": "http://images.amazon.com/images/P/0805052054.01.MZZZZZZZ.jpg",
      "book_author": "Hilary Mantel",
      "category_type": "sains-tech",
      "year_publication": 1997,
      "ISBN": "A0805052054"
    }
  },
  {
    "book_id": {
      "book_title": "Earthly Remains: The History and Science of Preserved Human Bodies",
      "image_url_m": "http://images.amazon.com/images/P/0195218523.01.MZZZZZZZ.jpg",
      "book_author": "Michael Parker Pearson",
      "category_type": "sains-tech",
      "year_publication": 2002,
      "ISBN": "A0195218523"
    }
  }
],
],

```

Gambar 4.42 *Response data mutasi BookRecomended*

```

"book_recommender": [
  {
    "book_title": "The Ships of Earth : Homecoming: Volume 3 (Homecoming Saga)",
    "image_url_m": "http://images.amazon.com/images/P/0812532635.01.MZZZZZZZ.jpg",
    "image_url_l": "http://images.amazon.com/images/P/0812532635.01.LZZZZZZZ.jpg",
    "book_author": "Orson Scott Card",
    "category_type": "sains-tech",
    "year_publication": 1995,
    "ISBN": "A0812532635",
    "_id": "643b3bd7999a62c9324bcca0"
  },
  {
    "book_title": "Memories of Summer",
    "image_url_m": "http://images.amazon.com/images/P/0440229219.01.MZZZZZZZ.jpg",
    "image_url_l": "http://images.amazon.com/images/P/0440229219.01.LZZZZZZZ.jpg",
    "book_author": "Ruth White",
    "category_type": "sains-tech",
    "year_publication": 2002,
    "ISBN": "A0440229219",
    "_id": "643b3bd9999a62c9324e5466"
  },
  {
    "book_title": "The Druid Way (Earth Quest)",
    "image_url_m": "http://images.amazon.com/images/P/1852303654.01.MZZZZZZZ.jpg",
    "image_url_l": "http://images.amazon.com/images/P/1852303654.01.LZZZZZZZ.jpg",
    "book_author": "Philip Carr-Gomm",
    "category_type": "sains-tech",
    "year_publication": 1993,
    "ISBN": "A1852303654",
    "_id": "643b3bd8999a62c9324cdf64"
  },
  {
    "book_title": "Herbs: The Magic Healers: A Complete Guide to Physical and Spiritual Well-Being",
    "image_url_m": "http://images.amazon.com/images/P/1570430330.01.MZZZZZZZ.jpg",
    "image_url_l": "http://images.amazon.com/images/P/1570430330.01.LZZZZZZZ.jpg",
    "book_author": "Paul Twitchell",
    "category_type": "sains-tech",
    "year_publication": 1997,
    "ISBN": "A1570430330",
    "_id": "643b3bd9999a62c9324de857"
  },
  {
    "book_title": "Inside Network Marketing: An Expert's View into the Hidden Truths and Exploited Myths of America's Most Misunderstood Industry",
    "image_url_m": "http://images.amazon.com/images/P/0761521763.01.MZZZZZZZ.jpg",
    "image_url_l": "http://images.amazon.com/images/P/0761521763.01.LZZZZZZZ.jpg",
    "book_author": "Leonard W. Clements",
    "category_type": "sains-tech",
    "year_publication": 2000,
  }
]

```

Gambar 4.43 Response data mutasi BookRecomended

Pada Gambar 4.41 dan Gambar 4.43 merupakan *response* dari mutasi fungsi *BookRecomenders*. *Response* ini didefinisikan melalui *typedef* dari skema GraphQL sesuai penamaan fungsi dan variabel nya. Secara umum, *typedef* ini mendefinisikan *response* sesuai kebutuhan pengembang untuk disajikan dan pendefinisian tersebut ditulis ke dalam file *typedefs.js* menggunakan modulasi *gql* seperti pada Tabel 4.1 Skema *response* menjelaskan data yang perlu ditampilkan beserta tipe data tersebut.

Tabel 4.1 Skema *response*

Nama tipe	Field : tipe data
book_interests	category: String title: String book_id: books
bookHistory	title: String book_id:book_recommender

book_recommender	_id: ID image_url_m: String book_author: String book_title: String image_url_l: String publisher: String year_publication: Int category_type: String ISBN: String count_document: String
responseBookRecommended	book_default:[book_interests] book_history: [bookHistory] book_recommender: [book_recommender]

PERPUSTAKAAN  
 UNIVERSITAS JENDERAL AGUM NYANI  
 YOGYAKARTA