

Penerapan Sistem Rekomendasi Registrasi Mata Kuliah Informatika UKDW

Renaldi Kristian Hartono¹, Aditya Wikan Mahastama², Yuan Lukito³

Program Studi Informatika, Universitas Kristen Duta Wacana

Jl. Dr. Wahidin Sudirohusodo 5-25, Yogyakarta

renaldi.kristian@ti.ukdw.ac.id

mahas@staff.ukdw.ac.id

yuanlukito@ti.ukdw.ac.id

Abstract— Every time a new academic year begins, the UKDW Informatics study program will provide courses that students can choose when registering. The courses provided are basically divided into compulsory courses and elective courses. Based on these 2 types of courses, for Informatics students, especially in choosing elective courses, it is often adjusted to each student's area of interest. This area of interest is then adjusted to the courses available at the time of registration, however, Informatics students often experience confusion in choosing courses at the time of registration.

Therefore, this research will design a course recommendation system that is used to help recommend courses to UKDW Informatics Students before registering. There are 2 engines used by the system, namely Python and Laravel. Python is used to process recommendations, while Laravel is only used to display recommendations. The recommendation process carried out by Python will use Collaborative Filtering which has processes including collecting data in the form of course rating results, processing similarities between courses for each area of interest, and calculating an evaluation matrix using MAP. Then the results of the highest similarity will be used as criteria for displaying recommendations. The results of the recommendations will then be checked using Laravel, so that each student can have different course recommendations.

The system was tested with 10 test cases in the form of student scenarios from the classes of 2021, 2020 and 2019. The test case results show the courses that students should take, while the evaluation is related to the results of the courses recommended by the system. The testcase evaluation results were stated to have a success rate of 80% based on 8 out of 10 successful testcases. Furthermore, the system has an accuracy level for subjects of interest of 46.36% which is calculated using the Mean Average Precision (MAP) method by averaging based on the precision of 10 testcases. Precision is calculated by comparing the courses in the field of interest that have been recommended with the courses in the field of interest that have been taken in each test case.

Intisari— Setiap tahun ajaran baru dimulai, prodi Informatika UKDW akan menyediakan mata kuliah yang dapat dipilih mahasiswa saat melakukan registrasi. Mata kuliah yang disediakan ini pada dasarnya dibagi menjadi mata kuliah wajib dan mata kuliah pilihan. Berdasarkan 2 macam mata kuliah ini, bagi Mahasiswa Informatika terutama dalam memilih mata kuliah pilihan seringkali disesuaikan dengan bidang minat dari masing-masing mahasiswa. Bidang minat ini yang kemudian disesuaikan dengan mata kuliah yang ada pada saat registrasi, namun walaupun demikian bagi Mahasiswa Informatika sering mengalami kebingungan dalam memilih mata kuliah pada saat registrasi.

Oleh sebab itu penelitian ini akan merancang sebuah sistem rekomendasi mata kuliah yang digunakan untuk membantu

merekomendasikan mata kuliah kepada Mahasiswa Informatika UKDW sebelum melakukan registrasi. Terdapat 2 mesin yang digunakan sistem dalam memproses rekomendasi yaitu Python dan Laravel. Proses rekomendasi yang dilakukan oleh Python akan menggunakan Item-Based Collaborative Filtering yang memiliki proses diantaranya mengumpulkan data berupa hasil rating mata kuliah, memproses similarity antar mata kuliah untuk tiap bidang minat, serta menghitung matriks evaluasi dengan menggunakan MAP. Kemudian hasil dari similarity yang paling tinggi akan digunakan sebagai kriteria untuk menampilkan rekomendasi menggunakan Laravel.

Sistem dilakukan uji coba dengan 10 testcase berupa skenario mahasiswa dari angkatan 2021, 2020 dan 2019. Hasil testcase menunjukkan mata kuliah yang seharusnya diambil mahasiswa, sedangkan evaluasi terkait dengan hasil mata kuliah yang direkomendasikan sistem. Hasil evaluasi testcase dinyatakan memiliki tingkat keberhasilan 80% berdasarkan 8 dari 10 testcase yang berhasil. Selanjutnya sistem memiliki tingkat keakuratan untuk mata kuliah bidang minat sebesar 46.36% yang dihitung menggunakan metode Mean Average Precision (MAP) dengan melakukan rata-rata berdasarkan presisi dari 10 testcase. Presisi dihitung dalam dengan cara membandingkan mata kuliah bidang minat yang telah direkomendasikan dengan mata kuliah bidang minat yang telah diambil pada tiap testcase.

Kata Kunci— Rekomendasi, Mata Kuliah, Collaborative Filtering, Mahasiswa, Informatika, Antarmuka, PHP, Testcase

I. PENDAHULUAN

Sebuah sistem rekomendasi dapat dikatakan baik apabila bisa berguna membantu pengguna mengatasi kelebihan informasi dengan memberikan rekomendasi khusus kepada pengguna, sehingga dapat memenuhi keinginan dan kebutuhan pengguna [1]. Dalam penelitiannya disebutkan bahwasannya terdapat 3 jenis sistem rekomendasi yaitu Collaborative Filtering, Content-Based Filtering, dan Hybrid. Kemudian ditampilkan untuk menerapkan metode dari jenis sistem rekomendasi yaitu Collaborative Filtering dengan membuat satu sistem rekomendasi film yang tujuannya adalah untuk membantu para pecinta film untuk memilih film yang tepat sesuai dengan kesukaannya. Metode yang digunakan yaitu Item-based Collaborative Filtering dengan memakai dataset dari MovieLens.org berupa 100.000 rating yang diberikan oleh pengguna terhadap film.

Berdasarkan perkembangan industri 4.0 saat ini banyak perusahaan yang beralih ke sistem digital. Tentunya didalam sistem digital terdapat data-data penting yang perlu

dilakukan analisis, sehingga [2] mengatakan Machine Learning adalah solusi dalam melakukan analisis data penting untuk memberikan user experience yang baik untuk pengguna. Item-based Collaborative Filtering merupakan metode dalam penelitiannya untuk membuat rekomendasi produk PT. Sentral Tukang Indonesia dengan memakai algoritma Cosine Similarity untuk menghitung tingkat kemiripan antar produk. Prediksi score memakai rumus Weighted Sum dan dalam menghitung tingkat error memakai Rumus Root Mean Squared Error.

Data yang digunakan adalah data histori penjualan dalam 1 periode (Q1 2021), data pelanggan, dan data produk pada PT. Sentral Tukang Indonesia. Data histori penjualan tersebut akan dibagi menjadi 80% untuk dataset training dan 20% untuk dataset testing. Hasil dari penelitian ini memperlihatkan rekomendasi top 10 produk per pelanggan. Produk yang tampil merupakan produk yang memiliki score tertinggi dari pelanggan tersebut. Dalam penelitian yang dilakukan oleh [3] mengenai sistem rekomendasi penjualan produk online toko mebel menjelaskan bahwa sistem rekomendasi adalah sistem yang dapat memberikan rekomendasi tentang item tertentu yang dapat digunakan untuk membantu pengguna sistem membuat keputusan. Dengan bantuan sistem rekomendasi yang diusulkan, pelanggan menerima rekomendasi produk yang akan dibeli berupa pemilihan material. Oleh sebab itu teknik Collaborative Filtering melakukan proses penilaian atau penyaringan barang berdasarkan pendapat orang lain dengan memberikan informasi kepada konsumen berdasarkan kesamaan karakteristik. Produk dengan nilai similaritas tertinggi lebih diutamakan untuk direkomendasikan kepada pelanggan.

Kemudian terdapat satu penelitian mengenai pembuatan aplikasi Android yang mampu merekomendasikan pariwisata berbasis Android di Kota Bengkulu yang dilakukan oleh [4]. Penelitian ini dilakukan karena adanya keterbatasan pengetahuan orang dengan tempat-tempat di kota tersebut sehingga menjadi kendala tersendiri bagi penumpang atau pendamping yang akan melakukan perjalanan ke daerah tersebut. Maka dari itu diperlukan pemandu yang dapat memandu dan menunjukkan tempat orang tersebut berada saat ini. Hasil dari penelitian ini adalah sistem informasi rekomendasi Pariwisata Bengkulu berbasis Android yang diuji dengan Metode Black Box Test dengan hasil 100% dan mendapatkan nilai user experience sangat baik sebesar 90,752% pada uji usability.

[5] membuat sistem rekomendasi untuk sebuah toko penjualan casing smartphone bernama Flazzstore dengan menggunakan metode Content-Based Collaborative Filtering. Secara prinsip metode ini mencari nilai similarity yang disesuaikan untuk memfilter produk milik toko Flazzstore berdasarkan rating dari tiap item. Kemudian nilai yang dihasilkan dari perhitungan similarity akan digunakan untuk menghitung nilai prediksi tiap produk dengan menggunakan Persamaan Weighted Average of Deviation. Berdasarkan hasil yang diperoleh dari persamaan ini, berikutnya perlu dilakukan operasi perhitungan dengan menghitung MAE / Mean Absolute Error untuk

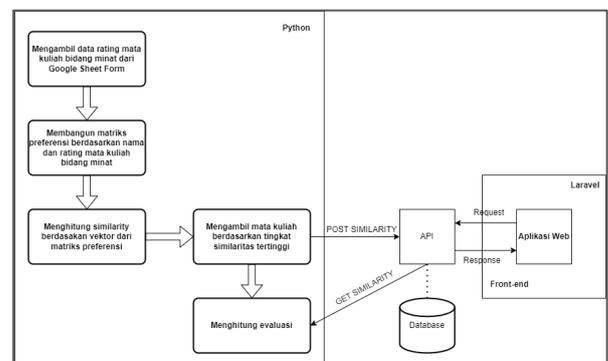
mendapatkan selisih antara nilai rating yang sebenarnya dengan prediksi.

Hasil dari Perhitungan MEE ini akan diurutkan berdasarkan nilai terkecil hingga terbesar. Penelitian ini menghasilkan rata-rata MAE 0,572039 dengan proses eksekusi 6,4 detik. Melihat dari hasil ini dapat disimpulkan waktu yang dibutuhkan cukup lama untuk mencari Nilai MAE. Oleh sebab itu penelitian berikutnya mengkombinasikan pendekatan Metode Content-based Filtering dan Collaborative Filtering atau disebut dengan Item-based Clustering Hybrid Method (ICHM) untuk memperoleh hasil yang lebih baik dan waktu yang lebih singkat.

Berdasarkan sumber dari tinjauan pustaka yang terkait dan menjadi landasan untuk penelitian ini, metode Collaborative Filtering merupakan salah satu metode yang dapat diterapkan untuk membuat sebuah sistem rekomendasi yang sesuai dengan kebutuhan. Secara prinsip dari metode ini akan menyeleksi data yang memiliki persamaan karakteristik dengan data lain serta melakukan perhitungan evaluasi sehingga menjadi sebuah pola untuk sistem dalam merekomendasikan sebuah data.

II. METODOLOGI PENELITIAN

A. Perancangan Arsitektur Sistem



Gambar 1. Arsitektur Sistem

Gambar 1 merupakan cara kerja dari arsitektur yang terdapat pada sistem. Pada dasarnya sistem memiliki 2 mesin yang dijalankan yaitu mesin dari Python dan Laravel. Python akan digunakan untuk memproses tahapan Collaborative Filtering dan Laravel digunakan sebagai aplikasi web yang disajikan untuk user. Untuk Collaborative Filtering akan diproses oleh sistem menggunakan Python, supaya tidak memberatkan aplikasi web yang digunakan user. Karena pada dasarnya aplikasi web hanya diperuntukan untuk menampilkan data dalam bentuk User Interface. Pada saat sistem berada di tahap perhitungan similarity, maka di tahap ini sistem memerlukan sebuah API yang digunakan untuk menyimpan hasil similarity ke database.

Sistem melakukan 1 kali proses similarity untuk semua mata kuliah dari tiap bidang minat melalui Python, kemudian dari proses ini sistem akan menyimpan hasil similarity ke database melalui API. Data similarity yang telah disimpan di database ini akan digunakan pada saat user memilih bidang minat yang telah ditampilkan oleh aplikasi web. Sehingga

sistem dapat memproses lebih cepat karena hanya perlu mengambil data mata kuliah yang sesuai dengan bidang minat yang diinput dari similarity tertinggi.

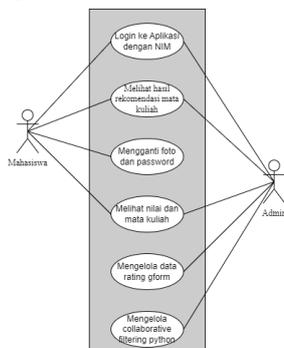
Kemudian tahap selanjutnya setelah memproses similarity adalah tahap menghitung evaluasi dengan menggunakan metode Mean Average Precision (MAP). Untuk melakukan perhitungan dengan metode ini diperlukan data berupa hasil similarity dari proses sebelumnya. Proses yang dilakukan similarity akan menghitung semua kemungkinan mata kuliah dari tiap bidang minat, dan hasil dari similarity ini telah dikirimkan melalui request POST ke API. Oleh sebab itu di tahap ini sistem akan melakukan request GET melalui API untuk mendapatkan data similarity sebelumnya.

TABEL I.
STRUKTUR ENDPOINT API

No	URL	Parameter
1	/get_similarity	-
2	/get_similarity/{bidangminat}	bidangminat (String)
3	/insert_similarity	first_matakuliah (String). second_matakuliah (String) result (int)

Berdasarkan Gambar 1 terdapat penggunaan API yang digunakan untuk menyimpan similarity. Struktur dari endpoint API pada Tabel 1 mengindikasikan ada 3 buah endpoint yaitu 2 endpoint dengan request GET dan 1 endpoint dengan request POST. Untuk endpoint pada nomor pertama digunakan untuk menampilkan keseluruhan hasil similarity, endpoint kedua digunakan untuk menampilkan hasil similarity berdasarkan bidang minat, serta endpoint ketiga digunakan untuk meng-input hasil similarity. Pada dasarnya endpoint dari API ini digunakan untuk menyimpan hasil perhitungan similarity yang diproses dari Python menggunakan request method post. Kemudian hasil request tersebut akan disimpan ke database dan hasilnya dapat ditampilkan dengan Laravel.

B. Use Case Diagram



Gambar 2. Use Case Diagram

Use case diagram pada penelitian ini merepresentasikan interaksi dari mahasiswa dengan sistem yang diantaranya dapat melakukan login dengan username nim dan password nim. Kemudian dapat melihat informasi seperti daftar nilai, daftar mata kuliah yang belum diambil, serta dapat mengganti password dan foto profil, serta dapat melihat hasil rekomendasi dan proses perhitungannya. Selain itu terdapat admin yang mengelola sistem, admin memiliki peran untuk mengelola Collaborative Filtering yang ada pada Python serta mengelola data rating dari Google Form. Pada dasarnya interaksi antara mahasiswa dengan sistem dapat dilakukan oleh admin. Namun khusus untuk fitur ganti password dan foto profil hanya dapat dikelola oleh tiap mahasiswa masing-masing akun mahasiswa.

C. Perancangan Pengujian Sistem

Pada penelitian ini pengujian sistem dilakukan untuk merancang evaluasi yang diperoleh dari beberapa skenario berupa testcase dari mahasiswa. Testcase yang berisi kondisi nilai dari beberapa mahasiswa serta input bidang minat yang dipilih user. Sebagai perancangan untuk pengujian sistem, tiap testcase akan diproses untuk mendapatkan daftar mata kuliah wajib yang seharusnya diambil. Kemudian pada hasil pengujian sistem, akan ditambahkan hasil rekomendasi mata kuliah bidang minat yang dipilih.

Untuk pengujian sistem akan dilakukan dengan 10 testcase seperti pada Tabel 2. Didalam testcase tersebut berisi NIM mahasiswa serta bidang minat yang dipilih. Seperti contoh pada testcase 1 terdapat mahasiswa angkatan 2021 dengan NIM 71210716 yang memilih bidang minat database, dan begitu seterusnya hingga testcase 10. Sehingga pada hasil pengujian sistem akan menampilkan keseluruhan mata kuliah yang direkomendasikan untuk ke-10 testcase yang tersedia.

TABEL II
PERANCANGAN TESTCASE PENGUJIAN SISTEM

Testcase	Mahasiswa	Angkatan	Bidang Minat
1	71210716	2021	Database
2	71190416	2019	Database, Programming
3	71190424	2019	AI
4	71190424	2019	Programming
5	71190417	2019	Network Systems, Desain Interface
6	71200541	2020	Programming, Database, AI
7	71190435	2019	Programming, Desain Interface
8	71190465	2019	Programming
9	71190441	2019	Database
10	71190412	2019	AI

III. HASIL DAN ANALISIS

A. Implementasi Data

Proses pengumpulan data merupakan implementasi awal yang dilakukan sebelum membangun sistem. Data yang dikumpulkan dalam sistem penelitian ini adalah data nilai dan identitas mahasiswa, data detail mata kuliah, serta data untuk cek prasyarat mata kuliah dan data rating. Data seperti nilai dan identitas mahasiswa serta data mata kuliah didapatkan dengan melakukan request data ke Data Warehouse dari Prodi Informatika.

Kemudian untuk pengumpulan data rating didapatkan dengan cara membuat form di G-Form yang dapat diakses di <https://forms.gle/JGTnBRNPSAg7sK796> seperti Gambar 3, kemudian form tersebut dibagikan melalui platform media sosial dengan mahasiswa Informatika sebagai target user untuk mengisi form ini. Selanjutnya hasil dari pengumpulan form ini akan di-share dalam bentuk google sheet untuk dapat diproses dengan Python.

Gambar 3. Tampilan Halaman GForm

B. Implementasi Sistem

Collaborative Filtering merupakan sebuah metode yang diimplementasikan oleh sistem dengan bahasa pemrograman Python menggunakan *engine* dari Google Collab. Metode ini akan memproses secara keseluruhan data rating yang telah didapatkan, data rating ini dikarenakan dibuat menggunakan Google Form maka outputnya tertampil dalam bentuk Google Sheet. Untuk mengelola data dari Google Sheet ke Python maka tahap awal dilakukan import dari library pandas, requests dan StringIO. Kemudian buat sebuah variabel untuk menyimpan url Google Sheet, url ini karena masih dalam bentuk Google Sheet maka harus diubah dalam bentuk CSV supaya dapat diproses oleh pandas.

```
1 import pandas as pd
2 import requests
3 from io import StringIO
4 url =
'https://docs.google.com/spreadsheets/d/1qVhSYC_Fmm7xEDZkIA3zVBmpUJ
JZ6dDEZ-DW6a8h2s/export?format=csv'
5 response = requests.get(url)
6 if response.status_code == 200:
7     data = response.text
8     df_wine = pd.read_csv(StringIO(data), sep=',')
9 else:
10    print("Gagal mengunduh file CSV.")
```

Selanjutnya Python perlu melakukan request dari url yang telah dimasukkan untuk dapat memproses data secara langsung dengan cara mengetikan `requests.get(url)` kemudian simpan dalam variabel `response` dan lakukan pengecekan untuk status request-nya sukses atau tidak, jika sukses maka ubah data menjadi text lalu baca text tersebut dalam bentuk csv dengan menambahkan separator ',' menggunakan fungsi `read_csv()` dari library pandas. Untuk mengecek apakah CSV telah dapat digunakan maka dapat mengetikan perintah `df_wine.head(5)` untuk menampilkan 5 data pertama dari CSV. Data tersebut berisi kolom seperti timestamp, nama panggilan, angkatan, serta kolom-kolom dari mata kuliah pilihan. Kemudian kolom tersebut untuk bagian timestamp, nama panggilan dan angkatan dihapus dengan mengetikan kode program dibawah. Setelah itu buat variabel bernama `getnama_mhs` untuk baca kembali data CSV yang hanya menampilkan kolom nama panggilan saja.

```
1 kolom_yang_dihapus = ["Timestamp", "Nama panggilan",
"Angkatan"]
df_wine.drop(columns=kolom_yang_dihapus, inplace=True)
2 getnama_mhs = pd.read_csv(StringIO(data), sep=',')[['Nama
panggilan']]
3 df = pd.DataFrame(df_wine)
4 # Filter kolom yang memenuhi kondisi untuk menampilkan data
yang minimal memiliki 1 user yang memberikan rating diatas 0
5 data = (df > 0).sum()
6 kolom_yang_memenuhi_kondisi = data[data >= 1].index
7 df_wine = pd.concat([getnama_mhs,
df[kolom_yang_memenuhi_kondisi]], axis=1)
```

Sistem kemudian akan mengatur data kolom data yang telah dihapus tadi untuk dibuat menjadi DataFrame. Ini dilakukan karena salah satu bagian untuk melakukan cleaning data adalah membuang data dari kolom mata kuliah yang hanya memiliki kurang dari 1 user yang memberikan rating diatas 0. Hal ini dilakukan karena pada saat menghitung vektor dari similarity-nya harus ada minimal 1 data yang memiliki rating diatas 0 supaya hasilnya tidak menjadi 0.

Setelah itu hasil data yang telah dilakukan cleaning akan digabungkan dengan data dari kolom nama panggilan yang telah dibuat sebelumnya. Ketika dilakukan print maka akan menghasilkan output seperti Gambar 4 merupakan hasil kolom yang telah dilakukan cleaning data sebelumnya. Output yang dihasilkan dapat memungkinkan untuk bertambah apabila dilakukan penambahan user yang memberikan rating terhadap mata kuliah yang tidak muncul.

Gambar 4. Ouput

```

1 from sklearn.metrics.pairwise import cosine_similarity
2 def calculate_cosine_similarity(data, bidangminat=''):
3     arry_matkul.clear()
4     count = 0
5     selected_columns = df_wine.filter(like='Nama panggilan')
6     for column in data:
7         selected_columns = pd.concat([selected_columns,
df_wine.filter(like=column)], axis=1, sort=False)
8     filter = pd.concat([selected_columns], axis=1)
9     filter = filter.drop(columns=['Nama panggilan'])
10    cosine_sim = cosine_similarity(filter.T)
11    cosine_sim_df = pd.DataFrame(cosine_sim,
index=filter.columns, columns=filter.columns)
12    print(cosine_sim_df)

```

Selanjutnya pada code diatas merupakan fungsi yang digunakan Python untuk menghitung Cosine Similarity. Python akan memodelkan matriks similarity-nya dahulu dengan cara mengambil kolom yang berisi rating mata kuliah dan membuang kolom seperti nama panggilan. Kemudian kolom yang telah dilakukan filter akan dilakukan transpose dengan memproses dalam libaray Cosine Similarity seperti baris 10. Namun pada baris 10 ini hanya menghasilkan output berupa hasil vektor similarity-nya saja, maka selanjutnya perlu diproses untuk dijadikan DataFrame dengan indeks dan kolom yang telah di-filter pada baris 11.

```

calculate_cosine_similarity(['ProgAndro', 'ProgIOS', 'ProgHyb',
'ProgDesk', 'PWL', 'CP', 'TekAnim'], 'Programming')

```

	ProgAndro	ProgIOS	ProgDesk	ProgHyb	PWL	CP
ProgAndro	1.000000	0.158362	0.379049	0.356313	0.779590	0.264262
ProgIOS	0.158362	1.000000	0.348155	0.500000	0.228086	0.310460
ProgDesk	0.379049	0.348155	1.000000	0.348155	0.430134	0.378309
ProgHyb	0.356313	0.500000	0.348155	1.000000	0.152057	0.543305
PWL	0.779590	0.228086	0.430134	0.152057	1.000000	0.330454
CP	0.264262	0.310460	0.378309	0.543305	0.330454	1.000000
TekAnim	0.289529	0.481125	0.335013	0.481125	0.314582	0.328615

Gambar 5. Hasil Similarity Bidang Minat Programming

Kemudian dari fungsi Cosine Similarity yang telah dibuat, maka dapat dipanggil dengan kode program diatas. Parameter yang dimasukkan merupakan mata kuliah dari bidang minat 'Programming', pada parameter pertama merupakan daftar mata kuliah yang ingin diproses *similarity* sedangkan parameter 2 merupakan jenis bidang minat dari mata kuliah yang diproses *similarity*. Berdasarkan input diatas maka apabila dijalankan program akan menghasilkan output berupa serangkaian hasil *similarity* dari masing-masing mata kuliah seperti Gambar 4.

```

calculate_cosine_similarity(['Abas', 'DW', 'BaDaTer', 'KamBD',
'AbasNor'], 'Database')

```

	Abas	AbasNor	DW	BaDaTer	KamBD
Abas	1.000000	0.365299	0.706890	0.618978	0.573685
AbasNor	0.365299	1.000000	0.254651	0.703906	0.628700
DW	0.706890	0.254651	1.000000	0.595367	0.418960
BaDaTer	0.618978	0.703906	0.595367	1.000000	0.558630
KamBD	0.573685	0.628700	0.418960	0.558630	1.000000

Gambar 6. Hasil Similarity Bidang Minat Database

```

calculate_cosine_similarity(['APB', 'ECom', 'EGov', 'Techno', 'MKW',
'KomBis', 'SIG', 'BII', 'BPM'], 'Manajemen Sistem Informasi')

```

	APB	ECom	EGov	Techno	MKW	KomBis	SIG
APB	1.000000	0.584016	0.594726	0.639021	0.573539	0.497499	0.615076
ECom	0.584016	1.000000	0.544971	0.472718	0.339422	0.677637	0.510882
EGov	0.594726	0.544971	1.000000	0.543502	0.418121	0.379957	0.558536
Techno	0.639021	0.472718	0.543502	1.000000	0.371391	0.709501	0.663812
MKW	0.573539	0.339422	0.418121	0.371391	1.000000	0.275371	0.376288
KomBis	0.497499	0.677637	0.379957	0.709501	0.275371	1.000000	0.544000
SIG	0.615076	0.510882	0.558536	0.663812	0.376288	0.544000	1.000000
BII	0.406054	0.587771	0.368026	0.448592	0.191346	0.701452	0.504009
BPM	0.580709	0.339422	0.601049	0.348179	0.312500	0.520624	0.505637

	BII	BPM
APB	0.406054	0.580709
ECom	0.587771	0.339422

Gambar 7. Hasil Similarity Bidang Minat Manajemen Sistem Informasi

```

calculate_cosine_similarity(['KomDat', 'AlGraf', 'CP', 'BPM'],
'Algoritma dan Struktur Data')

```

	KomDat	AlGraf	CP	BPM
KomDat	1.000000	0.410391	0.439057	0.312500
AlGraf	0.410391	1.000000	0.180185	0.365505
CP	0.439057	0.180185	1.000000	0.432197
BPM	0.312500	0.365505	0.432197	1.000000

Gambar 8. Hasil Similarity Bidang Minat Algoritma dan Struktur Data

```

calculate_cosine_similarity(['KomDat', 'Andasta', 'PraKK', 'BPM'],
'Analitika Data')

```

	KomDat	AlGraf	CP	BPM
KomDat	1.000000	0.410391	0.439057	0.312500
AlGraf	0.410391	1.000000	0.180185	0.365505
CP	0.439057	0.180185	1.000000	0.432197
BPM	0.312500	0.365505	0.432197	1.000000

Gambar 9. Hasil Similarity Bidang Minat Analitika Data

```

calculate_cosine_similarity(['DL', 'DigHum', 'SisPak', 'SemWeb',
'ML'], 'Data Science')

```

	DL	DigHum	SisPak	SemWeb	ML
DL	1.000000	0.257863	0.165616	0.355117	0.617876
DigHum	0.257863	1.000000	0.186840	0.400626	0.126738
SisPak	0.165616	0.186840	1.000000	0.643268	0.488393
SemWeb	0.355117	0.400626	0.643268	1.000000	0.436343
ML	0.617876	0.126738	0.488393	0.436343	1.000000

Gambar 10. Hasil Similarity Bidang Minat Data Science

```

calculate_cosine_similarity(['IoT', 'JST', 'KBS', 'NLP', 'PCD',
'PSD', 'GamEng', 'ML'], 'Artificial Intelligence')

```

	IoT	JST	KBS	NLP	PCD	PSD	GamEng	ML
IoT	<u>1.000000</u>	<u>0.476536</u>	0.417500	0.273894	0.169113	0.263181	0.296267	0.597803
JST	<u>0.476536</u>	<u>1.000000</u>	0.474516	0.476103	0.169949	0.244137	0.209393	0.809509
KBS	<u>0.417500</u>	<u>0.474516</u>	<u>1.000000</u>	0.656033	0.165301	0.514496	0.220639	0.456715
NLP	<u>0.273894</u>	<u>0.476103</u>	0.656033	<u>1.000000</u>	0.361477	0.337526	0.144747	0.511117
PCD	<u>0.169113</u>	<u>0.169949</u>	0.165301	0.361477	<u>1.000000</u>	0.321288	0.430571	0.075496
PSD	<u>0.263181</u>	<u>0.244137</u>	0.514496	0.337526	0.321288	<u>1.000000</u>	0.428845	0.234978
GamEng	<u>0.296267</u>	0.209393	0.220639	0.144747	0.430571	0.428845	<u>1.000000</u>	0.167949
ML	<u>0.597803</u>	<u>0.809509</u>	0.456715	0.511117	0.075496	0.234978	0.167949	<u>1.000000</u>

Gambar 11. hasil Similarity Bidang Minat AI

calculate_cosine_similarity(['PeKamjar', 'KamJar', 'IoT', 'ForText'], 'Cybersecurity')

	PeKamjar	KamJar	IoT	ForText
PeKamjar	<u>1.000000</u>	0.416333	0.344279	0.282843
KamJar	<u>0.416333</u>	<u>1.000000</u>	0.305443	0.452911
IoT	<u>0.344279</u>	<u>0.305443</u>	<u>1.000000</u>	0.263181
ForText	<u>0.282843</u>	<u>0.452911</u>	0.263181	<u>1.000000</u>

Gambar 12. Hasil Similarity Bidang Minat Cybersecurity

calculate_cosine_similarity(['DesGa', 'GrafGa', 'GAU', 'GamEng'], 'Game')

	DesGa	GrafGa	GAU	GamEng
DesGa	<u>1.000000</u>	<u>0.637522</u>	0.444603	0.624616
GrafGa	<u>0.637522</u>	<u>1.000000</u>	0.605878	0.784044
GAU	<u>0.444603</u>	<u>0.605878</u>	<u>1.000000</u>	0.577350
GamEng	<u>0.624616</u>	<u>0.784044</u>	0.577350	<u>1.000000</u>

Gambar 13. Hasil Similarity Bidang Minat Game

calculate_cosine_similarity(['PeKamjar', 'KamJar', 'Cloud', 'JarNir', 'OtoJar', 'TekWAN', 'IoT', 'EN'], 'Network Systems')

	PeKamjar	KamJar	Cloud	JarNir	OtoJar	TekWAN	IoT	EN
PeKamjar	<u>1.000000</u>	0.416333	0.454369	0.769667	0.431029	0.350000	0.344279	0.494975
KamJar	<u>0.416333</u>	<u>1.000000</u>	0.327408	0.508632	0.487199	0.320256	0.305443	0.301941
Cloud	<u>0.454369</u>	<u>0.327408</u>	<u>1.000000</u>	0.388570	0.432014	0.397573	0.343512	0.401610
JarNir	<u>0.769667</u>	<u>0.508632</u>	0.388570	<u>1.000000</u>	0.505935	0.488678	0.318295	0.575912
OtoJar	<u>0.431029</u>	<u>0.487199</u>	0.432014	0.505935	<u>1.000000</u>	0.338062	0.353881	0.478091
TekWAN	<u>0.350000</u>	<u>0.320256</u>	0.397573	0.488678	0.338062	<u>1.000000</u>	0.186097	0.707107
IoT	<u>0.344279</u>	<u>0.305443</u>	0.343512	0.318295	0.353881	0.186097	<u>1.000000</u>	0.263181
EN	<u>0.494975</u>	<u>0.301941</u>	0.401610	0.575912	0.478091	0.707107	0.263181	<u>1.000000</u>

Gambar 14. Hasil Similarity Bidang Minat Network Systems

calculate_cosine_similarity(['PDAP', 'DesEk', 'DEA', 'ProgAndro', 'ProgIoS', 'ProgHyb', 'PWL', 'VisDat', 'UXW'], 'Design Interface')

	PDAP	DesEk	DEA	ProgAndro	ProgHyb	PWL	VisDat	UXW
PDAP	<u>1.000000</u>	<u>0.954798</u>	0.838556	0.728583	0.324760	0.647453	0.843662	0.597573
DesEk	<u>0.954798</u>	<u>1.000000</u>	0.823925	0.717067	0.335410	0.643186	0.841472	0.593969
DEA	<u>0.838556</u>	<u>0.823925</u>	<u>1.000000</u>	0.718619	0.251230	0.735375	0.795984	0.461406
ProgAndro	<u>0.728583</u>	0.717067	0.718619	<u>1.000000</u>	0.356313	0.779590	0.602671	0.310563
ProgHyb	<u>0.324760</u>	<u>0.335410</u>	0.251230	0.356313	<u>1.000000</u>	0.152057	0.182089	0.311286
PWL	<u>0.647453</u>	<u>0.643186</u>	0.735375	0.779590	0.152057	<u>1.000000</u>	0.589754	0.321867
VisDat	<u>0.843662</u>	<u>0.841472</u>	0.795984	0.602671	0.182089	0.589754	<u>1.000000</u>	0.603096
UXW	<u>0.597573</u>	<u>0.593969</u>	0.461406	0.310563	0.311286	0.321867	0.603096	<u>1.000000</u>

Gambar 15. Hasil Similarity Bidang Minat Design Interface

Program dan data diatas adalah proses perhitungan keseluruhan dari semua mata kuliah tiap bidang minat.

Selanjutnya hasil dari similarity tersebut perlu diatur threshold atau ambang batas-nya terlebih dahulu supaya digunakan untuk mengatur nilai batas minimum dari mata kuliah yang akan ditampilkan similarity-nya. Sebagai contoh jika nilai threshold yang diatur adalah 0.5 apabila jika mengambil contoh dari hasil similarity di Gambar 4.13 maka mata kuliah seperti TekAnim dan ProgIOS tidak akan direkomendasikan karena nilai similarity-nya yaitu 0.481125 yang dibawah 0.5. Terkait threshold yang digunakan pada penelitian ini mengambil pemilihan threshold dengan nilai 0.1 yang ditentukan dengan mengambil nilai terkecil dari keseluruhan similarity. Pada similarity di Gambar 4.13 diatas apabila nilai terendah dibulatkan maka akan menjadi 0.1, dalam hal ini dibuat 0.1 threshold karena pada dasarnya hasil rekomendasi diambil dengan mengurutkan dari hasil similarity tertinggi.

```

1 for course, similar_courses in most_similar_courses.items():
2     if similar_courses:
3         print(f"Mata kuliah {course} memiliki kemiripan
4             tertinggi dengan:")
5         print(f"- {course} dengan similarity {1.00}")
6         arry_matkul[f"data-{count}"] = [course, course,
7             1.00]
8         url =
9             'https://renaldikristian.my.id/insert_similarity'
10        data_to_send = {
11            'first_matakuliah': course,
12            'second_matakuliah': course,
13            'bidangminat': bidangminat,
14            'result': 1.0,
15        }
16        for similar_course in similar_courses:
17            count+=1
18            similarity = cosine_sim_df[course][similar_course]
19            print(f"- {similar_course} dengan similarity
20                {similarity}")
21            arry_matkul[f"data-{count}"] = [course,
22                similar_course, similarity]
23            nilai_similarity.append(similarity)
24            # Periksa status respons
25            url =
26                'https://renaldikristian.my.id/insert_similarity' # Ganti dengan
27                URL API Anda
28            data_to_send = {
29                'first_matakuliah': course,
30                'second_matakuliah': similar_course,
31                'bidangminat': bidangminat,
32                'result': similarity,
33            }
34            response = requests.post(url, json=data_to_send)
35            print()

```

Maka sudah pasti mata kuliah yang paling direkomendasikan dapat diambil dengan melakukan sorting. Sehingga nilai threshold tidak perlu diambil dari similarity yang tertinggi. Disatu sisi diambil dengan threshold terendah karena bisa saja kemungkinan mata kuliah yang berada di similarity tertinggi banyak yang belum bisa diambil oleh mahasiswa karena prasyarat yang belum memenuhi. Sebagai contoh jika threshold ditentukan 0.7 maka mata kuliah dengan similarity dibawah 0.7 tidak akan masuk sebagai kategori rekomendasi. Namun yang menjadi permasalahan

terdapat kemungkinan bahwa mata kuliah dengan similarity minimal 0.7 sudah banyak diambil dan ada yang belum memenuhi syarat. Maka dari itu diambil threshold terendah untuk mengantisipasi beberapa kemungkinan tersebut.

Proses berikutnya adalah melakukan looping untuk menyimpan data hasil similarity ke database melalui API. Pada program diatas setiap program menampilkan perhitungan dari tiap mata kuliah kedua, maka program akan simpan mata kuliah pertama dan kedua yang dilakukan similarity beserta dengan jenis bidang minat-nya dan hasil similarity nya. Didalam menghitung similarity apabila ada mata kuliah yang sama seperti mata kuliah A dan A maka secara otomatis dihitung 1 karena kedua-nya memiliki vektor yang sama dan berdasarkan rumus cosine yang membagi vektor a dan b maka jika pembagi sama dapat menghasilkan nilai 1. Dan kondisi mata kuliah yang sama ini terjadi di setiap data yang di looping, sehingga pada program di baris 2 hingga 14 diberi kondisi jika terdapat data 'similarity_course' maka insert data melalui API dengan mengirimkan data first matakuliah dan second mata kuliah dengan nilai course, jenis bidang minat serta result yang bernilai 1. Kemudian di luar kondisi tersebut maka looping lagi similarity_course untuk menampilkan tiap mata kuliah yang memiliki kesamaan dengan mata kuliah lainnya, lalu simpan data nya melalui API dengan result masing-masing.

```

1  urls = 'https://renaldikristian.my.id/get_similarity';
2  result = requests.get(urls)
3
4  def average_precision(similarity_data):
5      precision = 0
6      num_relevant = 0
7      for i, data in enumerate(similarity_data):
8          if data['result'] >= 0.1: # Threshold for relevance
9              num_relevant += 1
10             precision += num_relevant / (i + 1)
11         if num_relevant == 0:
12             return 0
13         return precision / num_relevant
14
15     def mean_average_precision(similarity_data):
16         total_ap = 0
17         num_queries = len(similarity_data)
18         for query in similarity_data:
19             ap = average_precision([query])
20             total_ap += ap
21         return total_ap / num_queries
22
23     # Data similarity similarity_data = result.json()['data']
24     map_score = mean_average_precision(similarity_data)
25     print("Mean Average Precision (MAP):", map_score)

```

Kode program diatas merupakan fungsi yang digunakan untuk menghitung Mean Average Precision (MAP) menggunakan library Python. Untuk memproses MAP pada dasarnya membutuhkan data berupa similarity, pada program diatas similarity diambil dengan melakukan request GET ke endpoint /get_similarity. Fungsi average_precision menghitung nilai precision rata-rata dari suatu set data similarity. Precision dihitung dengan menggunakan threshold 0.1 untuk menentukan relevansi. Fungsi

mean_average_precision menghitung Mean Average Precision (MAP) dari suatu set data similarity. Ini mengambil setiap query, menghitung average precision menggunakan fungsi sebelumnya, dan kemudian menghitung rata-rata dari nilai-nilai tersebut. Sehingga apabila kode program diatas dijalankan maka akan menghasilkan output 1.0.

C. Hasil Pengujian Sistem

Berikutnya merupakan hasil dari pengujian sistem untuk menguji hasil evaluasi dari Collaborative Filtering yang telah dirancang. Pada bab 3 dibagian perancangan pengujian sistem telah disediakan beberapa testcase dari mahasiswa yang memiliki angkatan berbeda-beda. Pada testcase pertama adalah mahasiswa angkatan 2021 yang memiliki NIM yaitu 71210716, Gambar 16 merupakan implementasi pada saat user dari testcase 1 telah melakukan login ke aplikasi dan menginput salah satu bidang minat yaitu database. Kemudian dari hasil rekomendasi yang ditampilkan Gambar 17 artinya bahwa mahasiswa terkait akan disarankan untuk mengambil mata kuliah wajib dahulu. Maka dengan demikian hasil yang direkomendasikan sistem telah sesuai dengan mata kuliah yang diharapkan untuk kasus pada mahasiswa di semester awal seperti 1 hingga 3.

Gambar 16. Implementasi Input Bidang Minat Testcase 1

Hasil Rekomendasi Bidang Minat Database					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0133	STRUKTUR DATA	3	0	Wajib
2	TI0141	PRAKTIKUM STRUKTUR DATA	1	0	Wajib
3	TI0153	INFRASTRUKTUR LAN	3	0	Wajib
4	TI0161	PRAKTIKUM INFRASTRUKTUR LAN	1	0	Wajib
5	TI0173	SISTEM BASIS DATA	3	0	Wajib
6	TI0181	PRAKTIKUM SISTEM BASIS DATA	1	0	Wajib
7	TI0193	RISET OPERASI	3	0	Wajib
8	TI0213	SISTEM OPERASI	3	0	Wajib

Gambar 17. Hasil Rekomendasi Testcase 1

Pada testcase 1 ini tidak memerlukan proses dari Item-Based Collaborative Filtering karena mahasiswa pada testcase belum ada yang memenuhi prasyarat dari mata kuliah bidang minat database. Maka dari itu apabila dilakukan perhitungan presisi akan menghasilkan 0 karena tidak ada rekomendasi berupa mata kuliah bidang minat yang dibandingkan dengan mata kuliah bidang minat di KHS.

Hasil Rekomendasi Bidang Minat Database, Programming					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0313	MANAJEMEN PROYEK TEKNOLOGI INFORMASI	3	0	Wajib
2	TI6113	PEMROGRAMAN PERANGKAT BERGERAK ANDROID	3	0.77959020370695	Pilihan Wajib Profil
3	TI6153	PEMROGRAMAN WEB LANJUT	3	0.77959020370695	Pilihan Wajib Profil
4	TI3013	ADMINISTRASI BASIS DATA	3	0.70688979643829	Pilihan Wajib Profil
5	TI3033	DATA WAREHOUSE	3	0.70688979643829	Pilihan Wajib Profil
6	TI3063	ADMINISTRASI BASIS DATA NON RELASIONAL	3	0.7039061521025	Pilihan Wajib Profil
7	TI3043	BASIS DATA TERDISTRIBUSI	3	0.7039061521025	Pilihan Wajib Profil
8	TI3053	KEAMANAN BASIS DATA	3	0.62869974773752	Pilihan Wajib Profil

Gambar 18. Hasil Rekomendasi Testcase 2

Selanjutnya pada testcase 2 merupakan hasil evaluasi berikutnya terhadap rekomendasi yang dihasilkan sistem. Pada testcase 2 yang dirancang sebelumnya di bab 3 pada perancangan pengujian sistem merupakan mahasiswa angkatan 2020 dengan NIM yaitu 71200534. Mahasiswa tersebut menginputkan bidang minat database dan programming, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 18. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah yang terkait bidang minat, dikarenakan mata kuliah wajib yang direkomendasikan hanya 3 sks. Maka sisa sks lain akan dipakai untuk mata kuliah bidang minat.

TABEL III
HASIL PRESISI TESTCASE 2

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI6113	ProgAndro	TI6113	ProgAndro	1
TI6153	PWL	TI6153	PWL	1
TI3013	Abas	TI3013	Abas	1
TI3033	DW	TI3033	DW	1
TI3063	AbasNor	-	-	0
TI3043	BaDaTer	TI3043	BaDaTer	1
TI3053	KamBD	TI3053	KamBD	1

Tabel 3 merupakan hasil presisi dari testcase 2, pada hasil presisi ini akan membandingkan mata kuliah yang direkomendasikan dengan mata kuliah yang ada di KHS. Apabila dalam tabel terdapat kode dan nama mata kuliah KHS yang diberi tanda minus artinya mata kuliah tersebut tidak memiliki kecocokan dengan mata kuliah rekomendasi yang terkait dan diberi label 0, sedangkan untuk mata kuliah hasil rekomendasi yang memiliki kecocokan diberi label 1. Setelah memberikan label, maka selanjutnya dapat dilakukan perhitungan presisi dengan cara menjumlah tiap mata kuliah yang memiliki label 1 yang dibagi dengan total keseluruhan mata kuliah. Ketika diterapkan pada testcase 2 maka hasil presisi yaitu $(1+1+1+1+0+1+1) / 7$ yang akan menghasilkan 0.8571.

Selanjutnya pada testcase 3 merupakan hasil evaluasi berikutnya terhadap rekomendasi yang dihasilkan sistem. Pada testcase 3 yang dirancang sebelumnya di bab 3 pada perancangan pengujian sistem merupakan mahasiswa angkatan 2019 dengan NIM yaitu 71190424. Mahasiswa tersebut menginputkan bidang minat AI, maka hasil

rekomendasi yang ditampilkan sistem akan seperti Gambar 19. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah wajib.

Hasil Rekomendasi Bidang Minat Artificial Intelligence					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0243	PEMROGRAMAN WEB	3	0	Wajib
2	TI0251	PRAKTIKUM PEMROGRAMAN WEB	1	0	Wajib
3	TI0323	RISET TEKNOLOGI INFORMASI	3	0	Wajib
4	TI0333	KULIAH KERJA NYATA	3	0	Wajib
5	TI0353	KERJA PRAKTIK	3	0	Wajib
6	TI0373	REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	3	0	Wajib
7	TI0382	PRAKTIKUM REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	2	0	Wajib
8	TI6053	JARINGAN SYARAF TIRUAN	3	0.80950855930337	Pilihan Wajib Profil
9	TI6043	MACHINE LEARNING	3	0.80950855930337	Pilihan Wajib Profil

Gambar 19. Hasil Rekomendasi Testcase 3

TABEL IV
HASIL PRESISI TESTCASE 3

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI6053	JST	TI6053	JST	1
TI6043	ML	-	-	0

Tabel 4 merupakan hasil presisi dari testcase 3, pada hasil presisi ini akan membandingkan mata kuliah yang direkomendasikan dengan mata kuliah yang ada di KHS. Apabila dalam tabel terdapat kode dan nama mata kuliah KHS yang diberi tanda minus artinya mata kuliah tersebut tidak memiliki kecocokan dengan mata kuliah rekomendasi yang terkait dan diberi label 0, sedangkan untuk mata kuliah hasil rekomendasi yang memiliki kecocokan diberi label 1. Ketika diterapkan pada testcase 3 maka hasil presisi yaitu $(1+0) / 2$ yang akan menghasilkan 0.5.

Hasil Rekomendasi Bidang Minat Programming					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0243	PEMROGRAMAN WEB	3	0	Wajib
2	TI0251	PRAKTIKUM PEMROGRAMAN WEB	1	0	Wajib
3	TI0323	RISET TEKNOLOGI INFORMASI	3	0	Wajib
4	TI0333	KULIAH KERJA NYATA	3	0	Wajib
5	TI0353	KERJA PRAKTIK	3	0	Wajib
6	TI0373	REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	3	0	Wajib
7	TI0382	PRAKTIKUM REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	2	0	Wajib
8	TI9053	COMPETITIVE PROGRAMMING	3	0.54330536799443	Pilihan Bebas Profil
9	TI6143	PEMROGRAMAN DESKTOP	3	0.43013357084993	Pilihan Wajib Profil

Gambar 20. Hasil Rekomendasi Testcase 4

Gambar 20 merupakan sebuah hasil rekomendasi mata kuliah dari mahasiswa testcase 4. Terlihat ada total 18 SKS untuk rekomendasi mata kuliah wajib dan 6 mata kuliah bidang minat. namun berdasarkan 6 SKS dari mata kuliah bidang minat yang disarankan adalah mata kuliah Pemrograman Desktop. Hal ini karena berdasarkan hasil similarity dari Tabel 12 terdapat mata kuliah yang dapat direkomendasikan adalah PWL dan ProgAndro. Apabila mengecek prasyarat dari kedua mata kuliah ini di Buku Panduan Akademik maka akan ditemukan bahwa prasyarat dari mata kuliah PWL harus sudah mengambil Pemrograman Web minimal C.

Namun prasyarat ini belum terpenuhi oleh mahasiswa, sehingga hasil untuk tidak merekomendasikan mata kuliah PWL sudah benar karena tidak muncul di testcase 4. Tetapi untuk mata kuliah ProgAndro seharusnya muncul direkomendasi tetapi pada Gambar 19 tidak muncul. Untuk mata kuliah ProgAndro memiliki prasyarat yaitu telah mengambil mata kuliah PBO dengan nilai minimal C atau telah mengambil mata kuliah RPL-BO dengan nilai minimal C. Namun mahasiswa pada testcase 4 ini telah lulus dari mata kuliah PBO dengan nilai A, sehingga apabila berdasarkan kondisi 'OR' maka mata kuliah ProgAndro seharusnya disarankan. Pada hasil rekomendasi yang ditampilkan pada testcase ini dapat disebut gagal, namun perhitungan presisi tetap akan dilakukan.

TABEL V
HASIL PRESISI TESTCASE 4

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI6053	JST	TI6053	JST	1
TI6043	ML	-	-	0

Untuk menghitung presisi pada mahasiswa testcase 4 belum mengambil CP dan ProgDesk di KHS sehingga jika dihitung presisi akan menghasilkan 0 karena item mata kuliah dari KHS bernilai 0, sehingga jika dibandingkan dengan berapapun jumlah mata kuliah rekomendasi maka hasil presisinya tetap bernilai 0.

Hasil Rekomendasi Bidang Minat Network Systems, Design Interface					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0193	RISET OPERASI	3	0	Wajib
2	TI0243	PEMROGRAMAN WEB	3	0	Wajib
3	TI0251	PRAKTIKUM PEMROGRAMAN WEB	1	0	Wajib
4	TI0373	REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	3	0	Wajib
5	TI0382	PRAKTIKUM REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	2	0	Wajib
6	TI5023	DESAIN EKSPERIMENTAL	3	0.9547979771086	Pilihan Wajib Profil
7	TI5013	POLA DESAIN ANTARMUKA PENGGUNA	3	0.9547979771086	Pilihan Wajib Profil
8	TI9333	VISUALISASI DATA	3	0.84366249914693	Pilihan Bebas Prodi
9	TI5033	DESAIN DAN EVALUASI ANTARMUKA	3	0.83855611871482	Pilihan Wajib Profil

Gambar 21. Hasil Rekomendasi Testcase 5

Pada testcase ke-5 selanjutnya berisi mahasiswa angkatan 2019 dengan NIM 71190417. Mahasiswa tersebut menginputkan bidang minat programing dan network systems dan desain interface, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 21. Mata kuliah yang direkomendasikan sistem memiliki jumlah 12 SKS untuk mata kuliah wajib dan bidang minat.

TABEL VI
HASIL PRESISI TESTCASE 5

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI5023	DesEk	TI5023	DesEk	1
TI5013	PDAP	TI5013	PDAP	1
TI9333	VisDat	TI9333	VisDat	1
TI5033	DEA	TI5033	DEA	1

Hasil presisi yang diproses pada testcase 5 tertera seperti pada Tabel 6. Presisi tersebut menunjukkan bahwa dari 4 mata kuliah yang direkomendasikan, semua telah diambil dari KHS mahasiswa. Sehingga label yang diberikan bernilai 1 untuk semua dari 4 mata kuliah tersebut, maka presisi akan bernilai 1 yang diperoleh dari 4 / 4.

Hasil Rekomendasi Bidang Minat Programming, Database, Artificial Intelligence					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0313	MANAJEMEN PROYEK TEKNOLOGI INFORMASI	3	0	Wajib
2	TI6053	JARINGAN SYARAF TIURUAN	3	0.80950855930337	Pilihan Wajib Profil
3	TI6043	MACHINE LEARNING	3	0.80950855930337	Pilihan Wajib Profil
4	TI6113	PEMROGRAMAN PERANGKAT BERGERAK ANDROID	3	0.77959020370695	Pilihan Wajib Profil
5	TI6153	PEMROGRAMAN WEB LANJUT	3	0.77959020370695	Pilihan Wajib Profil
6	TI3013	ADMINISTRASI BASIS DATA	3	0.70688979643829	Pilihan Wajib Profil
7	TI3033	DATA WAREHOUSE	3	0.70688979643829	Pilihan Wajib Profil
8	TI3063	ADMINISTRASI BASIS DATA NON RELASIONAL	3	0.7039061521025	Pilihan Wajib Profil

Gambar 22. Hasil Rekomendasi Testcase 6

Pada testcase ke-6 selanjutnya berisi mahasiswa angkatan 2020 dengan NIM 71200541. Mahasiswa tersebut menginputkan bidang minat programing, database dan AI, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 22. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah yang terkait bidang minat, dikarenakan mata kuliah wajib yang direkomendasikan hanya 3 sks.

TABEL VII
HASIL PRESISI TESTCASE 6

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI6053	JST	TI6053	JST	1
TI6043	ML	TI6043	ML	1
TI6113	ProgAndro	TI6113	ProgAndro	1
TI6153	PWL	-	-	0
TI3013	Abas	-	-	0
TI3033	DW	-	-	0
TI3063	AbasNor	-	-	0

Presisi yang didapatkan pada Tabel 7 dapat diketahui bahwa ada 3 dari 7 mata kuliah rekomendasi yang telah diambil di KHS mahasiswa. Pada tabel ini ada 3 mata kuliah dari KHS yang sesuai dengan mata kuliah hasil rekomendasi diantaranya JST, ML dan ProgAndro. Sehingga jika total keseluruhan mata kuliah adalah 7, maka hasil presisi-nya adalah 3 / 7 yang menghasilkan 0.426.

Hasil Rekomendasi Bidang Minat Programming, Design Interface					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0161	PRAKTIKUM INFRASTRUKTUR LAN	1	0	Wajib
2	TI0323	RISET TEKNOLOGI INFORMASI	3	0	Wajib
3	TI0353	KERJA PRAKTIK	3	0	Wajib
4	TI5023	DESAIN EKSPERIMENTAL	3	0.9547979771086	Pilihan Wajib Prodi
5	TI5013	POLA DESAIN ANTARMUKA PENGGUNA	3	0.9547979771086	Pilihan Wajib Prodi
6	TI9333	VISUALISASI DATA	3	0.84366249914693	Pilihan Bebas Prodi
7	TI5033	DESAIN DAN EVALUASI ANTARMUKA	3	0.83855611871482	Pilihan Wajib Prodi
8	TI6153	PEMROGRAMAN WEB LANJUT	3	0.77959020370695	Pilihan Wajib Prodi

Gambar 23. Hasil Rekomendasi Testcase 7

Pada testcase ke-7 selanjutnya berisi mahasiswa angkatan 2019 dengan NIM 71190435. Mahasiswa tersebut menginputkan bidang minat programing dan desain interface, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 23. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah yang terkait bidang minat, dikarenakan mata kuliah wajib yang direkomendasikan hanya 3 sks.

TABEL VIII
HASIL PRESISI TESTCASE 7

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI5023	DesEk	TI5023	DesEk	1
TI5013	PDAP	TI5013	PDAP	1
TI9333	VisDat	TI9333	VisDat	0
TI5033	DEA	TI5033	DEA	1
TI6153	PWL	TI6153	PWL	0

Hasil rekomendasi tersebut menunjukkan bahwa hasil rekomendasi yang telah memiliki 7 SKS untuk mata kuliah wajib, maka mata kuliah bidang minat akan ditambahkan sebanyak 15 SKS. Sisa SKS ini digunakan untuk mata kuliah seperti DesEk, PDAP, VisDat, DEA, dan PWL berdasarkan urutan similaritas tertinggi. Untuk mata kuliah dari mahasiswa testcase 7 yang telah diambil di KHS dan sesuai dengan mata kuliah hasil rekomendasi adalah mata kuliah DesEk, PDAP, dan DEA. Dengan demikian maka ketiga mata kuliah tersebut diberi label 1 dan mata kuliah lain diberi label 0. Maka presisi dapat dihitung dengan cara $3 / 5$ yang akan menghasilkan 0.6 sebagai nilai presisi dari testcase 7.

Hasil Rekomendasi Bidang Minat Programming					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0323	RISET TEKNOLOGI INFORMASI	3	0	Wajib
2	TI6153	PEMROGRAMAN WEB LANJUT	3	0.77959020370695	Pilihan Wajib Prodi
3	TI9053	COMPETITIVE PROGRAMMING	3	0.54330536799443	Pilihan Bebas Prodi
4	TI9293	TEKNIK ANIMASI	3	0.48112522432469	Pilihan Bebas Prodi
5	TI6143	PEMROGRAMAN DESKTOP	3	0.43013357084993	Pilihan Wajib Prodi

Gambar 24. Hasil Rekomendasi Testcase 8

Pada testcase ke-8 selanjutnya berisi mahasiswa angkatan 2019 dengan NIM 71190465. Mahasiswa tersebut menginputkan bidang minat programing, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 4.32. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah yang terkait bidang minat,

dikarenakan mata kuliah wajib yang direkomendasikan hanya 3 sks yaitu untuk mata kuliah Riset Teknologi Informasi saja.

TABEL IX
HASIL PRESISI TESTCASE 8

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI6153	PWL	TI6153	PWL	1
TI9053	CP	TI9053	CP	0
TI9293	TekAnim	TI9293	TekAnim	0
TI6243	ProgDesk	TI6143	ProgDesk	0

Pada testcase 8 apabila dilakukan perhitungan presisi akan seperti pada Tabel 9. Untuk mata kuliah dari mahasiswa testcase 8 yang telah diambil di KHS dan sesuai dengan mata kuliah hasil rekomendasi adalah 1 dari 4 mata kuliah yang direkomendasikan yaitu PWL. Dengan demikian maka ketiga mata kuliah PWL diberi label 1 dan mata kuliah lain diberi label 0. Maka presisi dapat dihitung dengan cara $3 / 5$ yang akan menghasilkan 0,25 sebagai nilai presisi dari testcase 8.

Selanjutnya untuk testcase 9 berisi mahasiswa angkatan 2019 dengan NIM 71190412. Mahasiswa tersebut menginputkan bidang minat manajemen sistem informasi dan database, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 25. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah yang terkait bidang minat, dikarenakan mata kuliah wajib yang direkomendasikan hanya mata kuliah Riset Teknologi Informasi dan Kerja Praktek.

Hasil Rekomendasi Bidang Minat Database					
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result	Jenis Mata Kuliah
1	TI0353	KERJA PRAKTIK	3	0	Wajib
2	TI0366	SKRIPSI	6	0	Wajib
3	TI3063	ADMINISTRASI BASIS DATA NON RELASIONAL	3	0.7039061521025	Pilihan Wajib Prodi
4	TI3053	KEAMANAN BASIS DATA	3	0.62869974773752	Pilihan Wajib Prodi

Gambar 25. Hasil Rekomendasi Testcase 9

Apabila dilakukan perhitungan presisi pada testcase 9, pada Tabel 4.23 bahwa untuk mata kuliah dengan bidang minat database yang memiliki presisi antara mata kuliah dari KHS dengan mata kuliah hasil rekomendasi adalah 1 dari 2 mata kuliah yang direkomendasikan. Mata kuliah tersebut adalah KamBD, sehingga nilai presisi pada testcase 9 akan menghasilkan 0.5 yang berasal dari $1 / 2$.

TABEL X
HASIL PRESISI TESTCASE 9

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI3063	AbasNor	TI3063	AbasNor	0
TI3053	KamBD	TI3053	KamBD	1

Selanjutnya untuk testcase 10 berisi mahasiswa angkatan 2019 dengan NIM 71190412. Mahasiswa tersebut

menginputkan bidang minat manajemen sistem informasi, maka hasil rekomendasi yang ditampilkan sistem akan seperti Gambar 4.34. Mata kuliah yang direkomendasikan sistem cenderung lebih banyak mata kuliah yang terkait bidang minat.

Hasil Rekomendasi Bidang Minat Artificial Intelligence				
No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Result
1	TI0141	PRAKTIKUM STRUKTUR DATA	1	0
2	TI0161	PRAKTIKUM INFRASTRUKTUR LAN	1	0
3	TI0243	PEMROGRAMAN WEB	3	0
4	TI0251	PRAKTIKUM PEMROGRAMAN WEB	1	0
5	TI0313	MANAJEMEN PROYEK TEKNOLOGI INFORMASI	3	0
6	TI0382	PRAKTIKUM REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK	2	0
7	TI6053	JARINGAN SYARAF TIRUAN	3	0.80950855930337
8	TI6043	MACHINE LEARNING	3	0.80950855930337
9	TI6063	KNOWLEDGE BASED SYSTEM	3	0.6560334277721
10	TI6073	PEMROSESAN BAHASA NATURAL	3	0.6560334277721

Gambar 26. Hasil Rekomendasi Testcase 10

Berdasarkan hasil rekomendasi yang telah memiliki 11 SKS untuk mata kuliah wajib, maka mata kuliah bidang minat akan ditambahkan sebanyak 12 SKS. Kemudian untuk melakukan perhitungan presisi pada testcase 10 ini dapat diketahui bahwa 2 dari 4 mata kuliah yang direkomendasikan yaitu JST dan ML merupakan mata kuliah yang telah diambil di KHS. Maka hasil presisi akan menjadi 0.5 yang berasal dari 2 / 4.

TABEL XI
HASIL PRESISI TESTCASE 10

Mata kuliah Rekomendasi		Mata kuliah KHS		Status
Kode	Nama	Kode	Nama	
TI6153	PWL	TI6153	PWL	1
TI9053	CP	TI9053	CP	0
TI9293	TekAnim	TI9293	TekAnim	0
TI6243	ProgDesk	TI6143	ProgDesk	0

D. Analisis Hasil Pengujian

Sistem memiliki 10 testcase yang digunakan untuk evaluasi pengujian terhadap rekomendasi yang dihasilkan sistem. Berdasarkan hasil testcase ini akan dilakukan analisis berupa presentase tingkat keberhasilan testcase serta presentase tingkat kesesuaian rekomendasi bidang minat dengan user. Untuk melakukan analisis presentase terhadap kesesuaian rekomendasi bidang minat dengan user akan dilakukan rata-rata presisi dari setiap testcase yang terdapat mata kuliah bidang minat didalamnya.

TABEL XII
HASIL EVALUASI KESELURUHAN 10 TESTCASE

No	Status	KHS	Rekomendasi	Presisi
1	Sukses	0 matkul	0 matkul	0
2	Sukses	6 matkul	7 matkul	0.8571
3	Sukses	1 matkul	2 matkul	0.5
4	Gagal	0 matkul	2 matkul	0
5	Sukses	4 matkul	4 matkul	1
6	Sukses	3 matkul	7 matkul	0.4286
7	Sukses	3 matkul	5 matkul	0.6

8	Gagal	1 matkul	4 matkul	0.25
9	Sukses	1 matkul	2 matkul	0.5
10	Sukses	2 matkul	4 matkul	0.5

Tabel 12 berisi data keseluruhan hasil evaluasi dari 10 testcase yang digunakan pada sub-bab pengujian sistem sebelumnya. Terdapat 2 hasil data evaluasi didalam Tabel 4.26 yaitu data mengenai status testcase yang dapat dinyatakan sukses atau gagal, dan data mengenai hasil presisi dari keseluruhan testcase. Untuk perhitungan mengenai hasil presisi tidak dipengaruhi walaupun hasil rekomendasi dinyatakan gagal, sehingga dalam hal ini tiap testcase tetap memiliki nilai presisi masing-masing.

Selanjutnya berdasarkan hasil dari semua testcase di Tabel 12 diketahui bahwa 8 dari 10 testcase yang digunakan sebagai evaluasi sistem berhasil, sehingga presentase keberhasilan testcase mencapai 80%. Kemudian untuk presentase kesesuaian rekomendasi bidang minat terhadap user memiliki tingkat kesesuaian 46.36%. Hasil ini diperoleh dengan melakukan perhitungan Mean Average Precision (MAP) dengan menjumlah setiap presisi dari masing-masing testcase dan dibagi dengan total jumlah presisi seperti Gambar 27.

$$\text{hasil} = (0 + 0.8571 + 0.5 + 0 + 1 + 0.4286 + 0.6 + 0.25 + 0.5 + 0.5) / 10$$

$$\text{hasil} = 4.6357 / 10 = 0.46357 (46.36\%)$$

Gambar 27. Presentase Kesesuaian Presisi

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, berikut adalah kesimpulan yang dapat diambil diantaranya:

1. Pemilihan treshold *similarity* menggunakan nilai 0.1 dengan tujuan agar hasil rekomendasi yang diberikan sistem tidak terlalu ketat, sehingga prasyarat tidak menjadi hambatan untuk sistem memberikan rekomendasi.

2. Terdapat 8 dari 10 testcase untuk evaluasi sistem yang berhasil, sehingga testcase sistem memiliki tingkat keberhasilan 80% serta memiliki tingkat kesesuaian rekomendasi antara bidang minat dengan user sebesar 46.36% dari 10 testcase.

B. Saran

Berdasarkan kesimpulan diatas, berikut beberapa saran yang diberikan oleh penulis diantaranya:

1. Testcase yang digunakan untuk pengujian sistem dapat ditambahkan supaya tingkat keakuratan sistem yang menampilkan hasil rekomendasi dapat lebih sesuai dan akurat untuk user.

2. Membuat relasi tabel yang baik untuk pengecekan prasyarat agar mengurangi bug pada hasil rekomendasi yang diberikan sistem.

3. Menambahkan data rating yang dimiliki sistem terutama untuk mata kuliah yang tidak banyak diminati mahasiswa supaya mata kuliah tersebut memiliki peluang untuk direkomendasikan.

DAFTAR PUSTAKA

- [1] Jaja, L. V., Susanto, B., & Sasongko, R. L. (2020). Penerapan Metode Item-Based Collaborative Filtering Untuk Sistem Rekomendasi Data MovieLens. *Jurnal Matematika dan Aplikasi*, IX, 78-83. doi:10.35799/dc.9.2.2020.28274
- [2] Theodorus, D., Defit, S., & Gunadi, N. W. (2021). Machine Learning Rekomendasi Produk dalam Penjualan. *Jurnal Informasi dan Teknologi*, III, 202-208. doi:10.37034/jidt.v3i4.151202
- [3] Februariyanti, H., Laksono, D. A., Wibowo, S. J., & Mardi, U. S. (2021). IMPLEMENTASI METODE COLLABORATIVE FILTERING UNTUK SISTEM REKOMENDASI PENJUALAN PADA TOKO MEBEL. *JURNAL KHATULISTIWA INFORMATIKA*, IX, 43-50.
- [4] Setiawan, Y., Nurwanto, A., & Erlansari, A. (2019). Implementasi Item Based Collaborative Filtering Dalam Pemberian Rekomendasi Agenda Wisata Berbasis Android. *Jurnal Pseudocode*, VI, 13-20. doi:https://doi.org/10.33369/pseudocode.6.1.13-20
- [5] Prasetyo, B., Haryanto, H., Astuti, S., Astuti, Z. E., & Rahayu, Y. (2019). Implementasi Metode Item-Based Collaborative Filtering dalam Pemberian Rekomendasi Calon Pembeli Aksesoris Smartphone. *JURNAL EKSPLORA INFORMATIKA*, IX, 17-27. doi:10.30864/eksplora.v9i1.244
- [6] Leal, F., Malheiro, B., & Burguillo, C. J. (2018). Trust and Reputation Modelling for Tourism Recommendations Supported by Crowdsourcing (Vol. 745). Springer, Cham. doi:https://doi.org/10.1007/978-3-319-77703-0_81
- [7] Ricci, F., Rokach, L., & Shapira, B. (2015). Introduction to Recommender Systems Handbook. Springer.
- [8] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender Systems: An Introduction. Cambridge University Press.
- [9] Witten, H. I., Frank, E., Hall, A. M., & Pal, J. C. (2016). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. doi:10.1016/B978-0-12-374856-0.00026-2
- [10] Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.