

# SKRIPSI

## SISTEM REKOMENDASI PROGRAM STUDI UNIVERSITAS KATOLIK PARAHYANGAN MENGGUNAKAN ALGORITMA *COLLABORATIVE FILTERING*



Anugrah Jaya Sakti

NPM: 2016730053

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2020



# UNDERGRADUATE THESIS

## PARAHYANGAN CATHOLIC UNIVERSITY STUDY PROGRAM RECOMMENDATION SYSTEM USING *COLLABORATIVE FILTERING* ALGORITHM



Anugrah Jaya Sakti

NPM: 2016730053

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2020



# LEMBAR PENGESAHAN

SISTEM REKOMENDASI PROGRAM STUDI UNIVERSITAS KATOLIK  
PARAHYANGAN MENGGUNAKAN ALGORITMA *COLLABORATIVE FILTERING*

Anugrah Jaya Sakti

NPM: 2016730053

Bandung, «tanggal» «bulan» 2020

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Husnul Hakim, M.T.

«Pembimbing Pendamping»

Ketua Tim Penguji

Anggota Tim Penguji

Rosa De Lima, M.T.

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

**SISTEM REKOMENDASI PROGRAM STUDI UNIVERSITAS KATOLIK  
PARAHYANGAN MENGGUNAKAN ALGORITMA *COLLABORATIVE FILTERING***

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal «tanggal» «bulan» 2020

Meterai Rp. 6000
---------------------

Anugrah Jaya Sakti  
NPM: 2016730053





## ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

**Kata-kata kunci:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»



## ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

**Keywords:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»



*«kepada siapa anda mempersembahkan skripsi ini...?»*



## KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi dengan judul "Rekomendasi Program Studi di Perguruan Tinggi untuk Siswa SMA". Shalawat dan salam senantiasa tercurah kepada Rasulullah SAW. Penyusunan skripsi ini dimaksudkan untuk memenuhi syarat guna mencapai gelar sarjana di Program studi Teknik Informatika, Fakultas Teknologi Informasi dan Sains, Universitas Parahyangan Bandung.

Penulis menyadari bahwa penulisan ini tidak dapat terselesaikan tanpa dukungan dari berbagai pihak, baik moril maupun materil. Oleh karena itu, penulis ingin mengucapkan ucapan terima kasih kepada :

1. Kedua orang tua, Bapak Ibrahim Warga Purawinata dan (Almh) Ibu Dede Komariah yang telah memberikan dukungan baik moril maupun materil serta doa.
2. Syarif Jordan .
3. Bapak Husnul Hakim selaku Dosen Pembimbing.
4. Ibu Flaviana selaku Wakil Dekan Bidang Kemahasiswaan dan Alumni yang membantu dalam pembuatan surat permohonan kepada Biro Admiristari. Akademik.
5. Seluruh Bapak/Ibu dosen Teknik Informatika yang telah memberikan pengetahuan.
6. Seluruh teman-teman yang telah memeberikan semangat.

Bandung, «bulan» 2020

Penulis





# DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	3
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Sistem Rekomendasi . . . . .	5
2.1.1 Fungsi Sistem Rekomendasi . . . . .	5
2.1.2 Sumber Data dan Pengetahuan . . . . .	6
2.1.3 Teknik Rekomendasi . . . . .	6
2.1.4 <i>Collaborative Filtering</i> . . . . .	7
2.1.5 Aplikasi dan Evaluasi . . . . .	8
2.2 <i>Cluster</i> . . . . .	9
2.2.1 K-Means . . . . .	9
2.3 <i>Library</i> PHP-ML . . . . .	9
2.3.1 Array Dataset . . . . .	9
2.3.2 Random Split . . . . .	9
2.4 Universitas Katolik Parahyangan . . . . .	10
2.4.1 Program Studi . . . . .	10
2.4.2 Syarat Masuk Program Studi . . . . .	15
2.4.3 Karakteristik Program Studi . . . . .	16
<b>3 ANALISIS</b>	<b>19</b>
3.1 Analisis Perangkat Lunak Sejenis . . . . .	19
3.2 <i>Preprocessing</i> Data Mahasiswa . . . . .	22
3.3 Pemilihan Algoritma Sistem Rekomendasi . . . . .	23
3.4 Contoh Perhitungan Pearson Correlation . . . . .	23
3.5 Contoh Perhitungan Metode Evaluasi Sistem Rekomendasi . . . . .	25
3.6 Analisis Kebutuhan Sistem . . . . .	26
3.6.1 Diagram <i>Use Case</i> . . . . .	27
3.6.2 Rancangan Basis Data . . . . .	29

<b>4</b>	<b>PERANCANGAN</b>	<b>31</b>
4.1	Perancangan Fisik Basis Data . . . . .	31
4.1.1	Perancangan Tabel . . . . .	31
4.2	Perancangan Antar Muka . . . . .	32
4.3	Perancangan Algoritma . . . . .	34
4.3.1	Preprocessing Data . . . . .	35
4.3.2	Mahasiswa Controller . . . . .	35
4.3.3	Siswa Controller . . . . .	36
4.3.4	K-Means Controller . . . . .	37
4.3.5	User Based Model Controller . . . . .	44
4.3.6	Pearson Correlation Controller . . . . .	45
4.3.7	Prediction Controller . . . . .	47
4.3.8	Fakultas Controller . . . . .	48
4.3.9	Program Studi Controller . . . . .	48
4.3.10	Pengujian Controller . . . . .	48
4.3.11	Pearson Correlation Pengujian Controller . . . . .	52
4.3.12	Accuracy Controller . . . . .	52
4.4	Perancangan <i>Class Diagram</i> . . . . .	54
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>55</b>
5.1	Lingkungan Implementasi . . . . .	55
5.1.1	Lingkungan Perangkat Keras . . . . .	55
5.2	Lingkungan Perangkat Lunak . . . . .	55
5.3	Implementasi Tabel Basis Data . . . . .	55
5.4	Implementasi Antar Muka . . . . .	57
5.5	Pengujian Fungsional . . . . .	59
5.5.1	Pengujian Fungsional Pemilihan Jurusan SMA . . . . .	59
5.5.2	Pengujian Fungsional Pengisian Nilai . . . . .	59
5.6	Pengujian Eksperimental . . . . .	60
5.6.1	Metode Dasar . . . . .	60
5.6.2	Metode KMeans . . . . .	61
5.6.3	Kesimpulan Hasil Pengujian . . . . .	63
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>65</b>
6.1	Kesimpulan . . . . .	65
6.2	Saran . . . . .	65
	<b>DAFTAR REFERENSI</b>	<b>67</b>
	<b>A KODE PROGRAM</b>	<b>69</b>
	<b>B HASIL EKSPERIMEN</b>	<b>93</b>

## DAFTAR GAMBAR

3.1	7 Dimensi Profil Siswa . . . . .	19
3.2	Tampilan setelah registrasi atau <i>login</i> . . . . .	20
3.3	Modul Pengenalan . . . . .	20
3.4	Modul Potensi Diri . . . . .	21
3.5	Modul Ukur Kemampuan Diri . . . . .	21
3.6	Hasil Rekomendasi . . . . .	22
3.7	Diagram <i>Use Case</i> Sistem Rekomendasi . . . . .	28
3.8	Diagram ERD Sistem Rekomendasi . . . . .	29
4.1	Halaman Index Sistem . . . . .	33
4.2	Halaman Pengisian Nilai IPA . . . . .	33
4.3	Halaman Pengisian Nilai IPS . . . . .	34
4.4	Halaman Hasil Rekomendasi . . . . .	34
4.5	<i>Class Diagram</i> Sistem Rekomendasi . . . . .	54
5.1	Halaman Index Sistem . . . . .	57
5.2	Halaman Index Pengisian Nilai IPA . . . . .	58
5.3	Halaman Index Pengisian Nilai IPS . . . . .	58
5.4	Halaman Hasil Rekomendasi . . . . .	59
B.1	Hasil 1 . . . . .	93
B.2	Hasil 2 . . . . .	93
B.3	Hasil 3 . . . . .	93
B.4	Hasil 4 . . . . .	93



## DAFTAR TABEL

2.1	Tabel syarat program studi . . . . .	16
2.2	Tabel kriteria . . . . .	17
3.1	Contoh data mahasiswa dalam bentuk GPA . . . . .	23
3.2	Contoh data siswa dalam bentuk GPA . . . . .	24
3.3	Nilai kovariasi mahasiswa dan siswa . . . . .	24
3.4	Standar Deviasi Siswa . . . . .	24
3.5	Standar Deviasi Mahasiswa . . . . .	25
3.6	Contoh Perhitungan kesamaan . . . . .	25
3.7	Contoh hasil Prediksi . . . . .	25
3.8	Tabel Data MAE dan RMSE . . . . .	26
3.9	Pendefinisian Aktor . . . . .	27
3.10	Pendefinisian <i>Use Case</i> . . . . .	27
3.11	Skenario Memilih Jurusan SMA . . . . .	27
3.12	Skenario Mengisi Nilai Rapor . . . . .	28
4.1	Perancangan Tabel jurusan_sma . . . . .	31
4.2	Perancangan Tabel fakultas . . . . .	31
4.3	Perancangan Tabel program_studi . . . . .	31
4.4	Perancangan Tabel mahasiswa . . . . .	32
4.5	Perancangan Tabel mata_pelajaran . . . . .	32
4.6	Perancangan Tabel nilai . . . . .	32
5.1	Tabel Pengujian Fungsional Pemilihan SMA . . . . .	59
5.2	Tabel Pengujian Fungsional Pengisian Nilai . . . . .	60
5.3	Hasil Pengujian Jurusan IPA dengan Metode Dasar . . . . .	60
5.4	Hasil Pengujian Jurusan IPS dengan Metode Dasar . . . . .	61
5.5	Hasil Pengujian KMeans Jurusan IPA dengan nilai k 10 . . . . .	61
5.6	Hasil Pengujian KMeans Jurusan IPA dengan nilai k 20 . . . . .	61
5.7	Hasil Pengujian KMeans Jurusan IPA dengan nilai k 30 . . . . .	62
5.8	Hasil Pengujian KMeans Jurusan IPA dengan nilai k 40 . . . . .	62
5.9	Hasil Pengujian KMeans Jurusan IPS dengan nilai k 10 . . . . .	62
5.10	Hasil Pengujian KMeans Jurusan IPS dengan nilai k 20 . . . . .	62
5.11	Hasil Pengujian KMeans Jurusan IPS dengan nilai k 30 . . . . .	63
5.12	Hasil Pengujian KMeans Jurusan IPS dengan nilai k 40 . . . . .	63



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Salah satu tahapan pendidikan setelah lulus dari bangku sekolah menengah atas atau SMA adalah dengan melanjutkan studi ke perguruan tinggi baik perguruan tinggi negeri ataupun swasta. Salah satu hal yang perlu diperhatikan saat akan melanjutkan studi di perguruan tinggi adalah program studi apa yang akan dipilih. Program studi adalah kesatuan rencana belajar sebagai pedoman penyelenggaraan pendidikan akademik dan/atau profesional yang diselenggarakan atas dasar suatu kurikulum serta ditujukan agar mahasiswa dapat menguasai pengetahuan, keterampilan, dan sikap sesuai dengan sasaran kurikulum.

Kesalahan dalam memilih program studi memiliki dampak yang signifikan bagi kehidupan mahasiswa dimasa mendatang. Dampak bisa berupa masalah psikologi, mahasiswa akan merasa terpaksa saat belajar karena mempelajari sesuatu hal yang tidak sesuai minat. Selain masalah psikologi dampak lain yang bisa terjadi berupa masalah pada bidang akademik, prestasi seorang mahasiswa tidak akan maksimal, nilai mata kuliah kurang baik, dan mahasiswa yang salah dalam memilih jurusan memiliki kemungkinan yang lebih tinggi mengalami *Drop Out*.

Ketidakcocokan program studi dengan mahasiswa di Indonesia masih cukup tinggi. Berdasarkan buku Statistik Pendidikan Tinggi pada tahun 2017, terdapat 1.437.425 mahasiswa baru, 6.924.511 mahasiswa terdaftar, dan 1.046.141 mahasiswa lulus. Dengan kata lain ada 391.284 atau 27.22% mahasiswa yang tidak lulus. Jumlah mahasiswa *Drop Out* pada tahun 2017 adalah 195.176 dengan presentasi pada Perguruan Tinggi Negeri (PTN) sebesar 96% dan pada Perguruan Tinggi Swasta (PTS) sebesar 4%. Presentasi jumlah mahasiswa lulus tepat waktu merupakan salah satu faktor yang menentukan kualitas universitas (PP No. 66 Tahun 2010) selain itu, menurut Sudjito (2014): kecocokan program studi merupakan salah satu penentu keberhasilan studi dari seorang mahasiswa.

Maka dari itu diperlukan sebuah sistem rekomendasi yang dapat memberikan rekomendasi item berupa program studi yang sesuai dengan minat siswa SMA. Sistem rekomendasi adalah alat dan teknik perangkat lunak yang menyediakan saran untuk item yang akan digunakan oleh pengguna. Sistem rekomendasi berfokus pada item tertentu dan ditujukan untuk individu atau personal. Beberapa teknik yang biasa digunakan pada sistem rekomendasi, yaitu : *Content-based*, *Collaborative Filtering*, *Demographic*, *Knowledge-based*, *Community-based*, *Hybrid recommender systems*. Teknik yang akan digunakan pada sistem rekomendasi yang akan dibangun adalah *Collaborative Filtering*.

*Collaborative Filtering* merupakan teknik yang merekomendasikan item yang sesuai dengan kebutuhan pengguna berdasarkan *rating* tanpa memerlukan informasi mengenai item ataupun pengguna, contoh informasi yang dimaksud adalah deskripsi mengenai item atau pengguna. Secara sederhana *Collaborative Filtering* menghitung kesamaan atau similaritas antara pengguna aktif dengan beberapa pengguna yang memiliki selera atau minat yang serupa. Untuk menghitung similaritas digunakan metode *Pearson Correlation Coefficient*. *Pearson Correlation Coefficient* bekerja dengan cara menghitung korelasi antara dua variabel dari masing-masing pengguna yang sedang dibandingkan. Semakin tinggi nilai korelasi yang dihasilkan maka mengidentifikasi kedua pengguna memiliki similaritas yang cukup tinggi.

Pada skripsi ini akan dibangun sebuah perangkat lunak sistem rekomendasi yang dapat memberikan rekomendasi item berupa program studi yang sesuai dengan minat siswa SMA. Sistem rekomendasi ini akan menggunakan algoritma *Collaborative Filtering* dengan model *Neighborhood* dengan pendekatan *User-based*. *User-based* memprediksi berdasarkan kesamaan *rating* pengguna dengan item.

*Rating* yang dalam kasus ini adalah indeks prestasi kumulatif (IPK). Berdasarkan Pasal 23 ayat 5 Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor Tahun 2014 tentang Standar Nasional Pendidikan Tinggi yang berbunyi Hasil penilaian capaian pembelajaran lulusan pada akhir program studi dinyatakan dengan indeks prestasi kumulatif (IPK).

## 1.2 Rumusan Masalah

Berikut adalah rumusan masalah dari penulisan skripsi :

1. Bagaimana cara menilai kecocokan seorang calon mahasiswa terhadap suatu program studi ?
2. Bagaimana membangun perangkat lunak untuk memberikan rekomendasi program studi di perguruan tinggi yang cocok untuk calon mahasiswa ?
3. Bagaimana kualitas hasil rekomendasi dari perangkat lunak yang dibangun ?

## 1.3 Tujuan

Tujuan dari penulisan skripsi ini adalah sebagai berikut :

1. Mempelajari cara menilai kecocokan seorang mahasiswa terhadap suatu program studi.
2. Membangun perangkat lunak untuk memberikan rekomendasi program studi di perguruan tinggi yang cocok dengan calon mahasiswa.
3. Menguji hasil rekomendasi dari perangkat lunak yang sudah dibangun.

## 1.4 Batasan Masalah

Mengingat banyaknya perguruan tinggi dan program studi yang ada di Indonesia, maka perlu adanya batasan masalah yang jelas mengenai apa yang dibuat dan diselesaikan dalam penulisan skripsi ini. Berikut merupakan batasan-batasan masalah pada skripsi ini :

1. 15 Program studi Universitas Katolik Parahyangan (UNPAR).
2. Data mahasiswa UNPAR yang masuk melalui jalur Penelusuran Minat dan Kemampuan (PMDK) pada tahun 2013-2018 yang sudah lulus.
3. Hanya menggunakan nilai mata pelajaran Matematika, Bahasa Indonesia, Bahasa Inggris, Pendidikan Kewarganegaraan, Fisika, dan Kimia sebagai atribut.

## 1.5 Metodologi

1. Melakukan studi literatur mengenai sistem rekomendasi.
2. Mempelajari mengenai berbagai program studi dan karakteristiknya.
3. Mempelajari metode yang dapat digunakan untuk menghitung tingkat kecocokan calon mahasiswa dengan program studi.



4. Menganalisis hal-hal yang mempengaruhi kecocokan program studi dengan calon mahasiswa.
5. Mempelajari framework Laravel dan Bootstrap.
6. Membangun perangkat lunak sesuai dengan analisis dan perancangan yang dilakukan.
7. Melakukan pengujian kualitas hasil rekomendasi perangkat lunak yang dibangun.
8. Menulis dokumen skripsi.

## 1.6 Sistematika Pembahasan

1. Bab 1 menjelaskan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan untuk sistem rekomendasi program studi di perguruan tinggi untuk anak SMA.
2. Bab 2 menjelaskan mengenai sistem rekomendasi dengan menggunakan *Collaborative Filtering*, teknik perhitungan kesamaan atau similaritas dengan *Pearson Correlation Coefficient*, teknik *clustering* menggunakan K-Means, penjelasan *Framework Laravel*, dan Program Studi yang berada di Universitas Katolik Parahyangan.
3. Bab 3 menjelaskan analisis yang dilakukan terhadap sistem sudah ada, *preprocessing* data, algoritma yang akan digunakan, contoh perhitungan, dan kebutuhan sistem yang akan dibangun.
4. Bab 4 menjelaskan perancangan fisik basis data, antar muka, algoritma, dan *class diagram* yang akan digunakan sistem.
5. Bab 5 berisikan implementasi dari rancangan yang sudah dilakukan pada bab 4 untuk sistem rekomendasi program studi Universitas Katolik Parahyangan.
6. Bab 6 berisikan kesimpulan yang dapat diambil oleh penulis dan saran untuk memberikan hasil rekomendasi yang lebih baik.



## BAB 2

### LANDASAN TEORI

#### 2.1 Sistem Rekomendasi

Sistem rekomendasi adalah alat dan teknik perangkat lunak yang menyediakan saran untuk item yang akan digunakan oleh pengguna. Saran terkait dengan berbagai proses pengambilan keputusan, seperti barang apa yang akan dibeli, musik apa yang akan didengarkan, atau berita online apa yang akan dibaca.

Sistem rekomendasi biasanya berfokus pada item tertentu seperti buku, musik, dll. Sistem rekomendasi ditujukan untuk individu atau personal yang kurang memiliki pengalaman pribadi. Contoh Sistem rekomendasi buku adalah *website* Amazon.com. Item yang ditawarkan sebagai daftar item peringkat. Sistem rekomendasi mencoba memprediksi produk dengan cara mengumpulkan referensi dari pengguna lainnya.

Pengembangan sistem rekomendasi dimulai dari pengamatan yang sederhana berupa rekomendasi yang diberikan oleh orang lain dalam membuat keputusan rutin sehari-hari bisa berupa buku, musik, film, rekrutmen karyawan, dll.

Sistem rekomendasi menghasilkan rekomendasi menggunakan berbagai jenis pengetahuan dan data tentang pengguna, item yang tersedia, dan transaksi sebelumnya, contohnya berupa e-commerce yang mengatasi masalah kelebihan informasi yang terjadi akibat transaksi pengguna sebelumnya.

##### 2.1.1 Fungsi Sistem Rekomendasi

Fungsi utama sistem rekomendasi adalah menemukan item yang relevan dengan kebutuhan pengguna. Selain untuk menemukan item yang relevan terdapat juga beberapa fungsi sistem rekomendasi, yaitu :

1. Meningkatkan jumlah penjualan barang  
Salah satu fungsi penting untuk sistem rekomendasi yang komersil. Peningkatan jumlah penjualan item ini disebabkan karena penjualan item dilakukan tepat sasaran kepada pembeli yang memang membutuhkan dan menginginkan item tersebut. Merekomendasikan item yang sesuai dengan kebutuhan atau minat pengguna.
2. Menjual barang-barang yang lebih beragam  
Memberikan rekomendasi item yang mungkin sulit ditemukan oleh pengguna jika tanpa menggunakan sistem rekomendasi.
3. Meningkatkan kepuasan pengguna  
Sistem rekomendasi yang dirancang dengan baik memberikan rekomendasi yang sesuai dengan kebutuhan pengguna sehingga pengguna akan merasa senang menggunakan sistem tersebut.
4. Meningkatkan kesetiaan pengguna  
Pengguna akan tetap menggunakan sebuah *website* jika sistem rekomendasi yang hasilkan rekomendasi yang sesuai dengan kebutuhan pengguna.

5. Lebih mengerti apa yang diinginkan pengguna  
Sistem dapat memberikan hasil rekomendasi item yang sesuai dengan kebutuhan pengguna.

### 2.1.2 Sumber Data dan Pengetahuan

Sistem rekomendasi adalah sistem pemrosesan informasi yang secara aktif mengumpulkan berbagai jenis data untuk membangun rekomendasinya. Data utama berupa data item yang disarankan dan pengguna yang akan menerima rekomendasi. Data yang digunakan sistem rekomendasi mencakup pada tipe jenis objek, yaitu :

1. Item  
Item adalah objek yang direkomendasikan, item bisa ditandai oleh kompleksitasnya dan nilai atau kegunaannya. Bisa bernilai positif jika sesuai atau negatif jika tidak sesuai.
2. Pengguna  
Pengguna adalah objek yang menggunakan sistem, memiliki tujuan dan karakteristik beragam. Pengguna juga dapat dijelaskan oleh data pola perilaku (pola penelusuran web, atau pola pencarian perjalanan)
3. Transaksi  
Interaksi yang direkam antara pengguna sistem rekomendasi. Transaksi adalah data seperti log yang menyimpan informasi penting yang dihasilkan selama interaksi manusia-komputer dan berguna untuk algoritma pembuatan rekomendasi yang digunakan sistem. Bentuk dari peringkat yang populer di sistem rekomendasi :
  - Peringkat numerik 1 - 5
  - Peringkat ordinal (sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju)
  - Peringkat biner, buruk (0) dan baik (1)
  - Peringkat unary menunjukkan bahwa pengguna telah mengamati atau membeli barang atau menilai barang secara positif

### 2.1.3 Teknik Rekomendasi

Berikut adalah teknik-teknik yang dapat digunakan pada sistem rekomendasi :

1. *Content-based*  
Sistem merekomendasikan item yang mirip berdasarkan item yang disukai pengguna di masa lalu. Kesamaan dihitung berdasarkan fitur(atribut) yang terkait dengan item. misal , review positif film komedi, maka akan direkomendasikan film di genre yang sama.
2. *Collaborative Filtering*  
Rekomendasi berdasarkan item yang disukai pengguna lain yang memiliki kesamaan. Implementasi paling sederhana, merekomendasikan item yang disukai pengguna lain dengan selera serupa di masa lalu. *Collaborative Filtering* populer dan banyak digunakan pada sistem rekomendasi. *Nearest neighbors* meningkatkan popularitas karena sederhana, efisien, dan kemampuan mereka untuk menghasilkan rekomendasi yang akurat dan menunjukkan ciri personal tertentu.
3. *Demographic*  
Rekomendasi berdasarkan profil demografis pengguna. Asumsinya bahwa rekomendasi yang berbeda harus dihasilkan untuk demografis yang berbeda. Misalnya diarahkan ke web dengan bahasa atau negara pengguna.

4. *Knowledge-based*

Merekomendasikan item berdasarkan pengetahuan domain spesifik tentang fitur (atribut) item tertentu yang memenuhi kebutuhan atau referensi pengguna.

5. *Community-based*

Merekomendasikan item berdasarkan teman-teman pengguna. Bukti menunjukkan bahwa orang cenderung lebih mengandalkan rekomendasi dari teman-teman dari pada rekomendasi dari orang yang belum dikenal.

6. *Hybrid recomender systems*

Kombinasi dari beberapa teknik yang sudah disebutkan sebelumnya. Menggunakan teknik A dan B mencoba untuk menggunakan keunggulan A dan memperbaiki kelemahan B. Contoh, *Collaborative Filtering* memiliki kelemahan terhadap item yang tidak memiliki peringkat (tidak terdapat riwayat) bisa digabungkan dengan metode *Content-based*.

### 2.1.4 Collaborative Filtering

Dalam pengembangan sistem rekomendasi dapat menggunakan teknik *Collaborative Filtering*. *Collaborative Filtering* menghasilkan rekomendasi item yang spesifik untuk pengguna berdasarkan peringkat tanpa memerlukan informasi tambahan mengenai item ataupun pengguna. Gagasan utamanya adalah peringkat pengguna  $u$  untuk item  $i$  cenderung mirip dengan pengguna  $v$ , jika  $u$  dan  $v$  memberikan peringkat item lain dengan nilai yang sama.

Tantangan dalam membangun sistem rekomendasi menggunakan teknik *Collaborative Filtering* adalah sedikitnya jumlah data pengguna sebelumnya yang sudah memberikan peringkat kepada suatu item. Dalam *Collaborative Filtering* terdapat salah satu algoritma yaitu *Neighborhood-based Collaborative Filtering* atau yang dikenal dengan *Memory-base Collaborative Filtering*.

#### *Neighborhood-based Collaborative Filtering*

*Neighborhood-based Collaborative Filtering* atau yang dikenal dengan *Memory-base Collaborative Filtering* adalah algoritma pertama yang dikembangkan untuk teknik *Collaborative Filtering*. Pada algoritma ini *rating user-item* disimpan dalam sistem secara langsung digunakan untuk memprediksi peringkat item baru, dapat dilakukan dengan *user-based model*.

#### *User-based Neighborhood Model*

*User-based* bekerja dengan mengidentifikasi pengguna yang akan diberikan rekomendasi dengan pengguna lain yang memiliki kesamaan. Aktivitas pengguna yang memiliki kesamaan ini akan menjadi dasar dalam memberikan rekomendasi kepada pengguna lain. Aktivitas bisa berupa memberikan *rating* kepada item. Berikut adalah tahapan yang perlu dilakukan pada *User-based Neighborhood Model* :

1. Menghitung nilai rata-rata *rating* yang sudah diberikan oleh pengguna lain.
2. Menghitung kesamaan atau similaritas pengguna menggunakan *Pearson Correlation Coefficient*

2.1 :

$$\text{sim}(i, j) = \text{Pearson}(i, j) = \frac{\sum_{k \in I_i \cap I_j} (r_{i,k} - \mu_i) \cdot (r_{j,k} - \mu_j)}{\sqrt{\sum_{k \in I_i \cap I_j} (r_{i,k} - \mu_i)^2} \cdot \sqrt{\sum_{k \in I_i \cap I_j} (r_{j,k} - \mu_j)^2}} \quad (2.1)$$

Keterangan :

- $\text{sim}(i, j)$  = Kesamaan atau similaritas antara pengguna  $i$  dan pengguna  $j$
- $\sum_{k \in I_i \cap I_j}$  = Himpunan item pengguna  $i$  dan pengguna  $j$  yang saling beririsan

- $r_{i,k}$  = Nilai yang diberikan pengguna i terhadap item k
  - $r_{j,k}$  = Nilai yang diberikan pengguna j terhadap item k
  - $\mu_i$  = Rata-rata nilai yang diberikan pengguna i
  - $\mu_j$  = Rata-rata nilai yang diberikan pengguna j
3. Memilih nilai kesamaan atau similaritas yang bernilai lebih besar dari 0. Nilai kesamaan atau similaritas memiliki rentan nilai -1, 0, dan +1 untuk pengguna yang akan diberikan rekomendasi. Jika hasil perhitungan mendekati -1, berarti pengguna tersebut kurang memiliki kesamaan dengan pengguna yang akan diberikan rekomendasi. Jika hasil perhitungan mendekati 0, berarti pengguna tersebut memiliki kesamaan yang cukup baik dengan pengguna yang akan diberikan rekomendasi. Jika hasil perhitungan mendekati +1, berarti pengguna tersebut memiliki kesamaan yang tinggi dengan pengguna yang akan diberikan rekomendasi.
4. Menghitung nilai prediksi dengan rumus *weighted sum* 2.2 :

$$r_{i,k} = \frac{\sum (Sim(i,j) * r_{j,k})}{\sum Sim(i,j)} \quad (2.2)$$

Keterangan :

- $r_{i,k}$  = Nilai prediksi pengguna i untuk item k
  - $Sim(i,j)$  = kesamaan atau similiaritas pengguna i dan pengguna j
  - $r_{j,k}$  = Penilaian pengguna j terhadap item k
5. Mengurutkan nilai prediksi dari yang terbesar ke terkecil.

### 2.1.5 Aplikasi dan Evaluasi

#### Aplikasi

Faktor pertama yang harus dipertimbangkan adalah domain aplikasi yang akan dibangun karena memiliki efek yang besar pada algoritma yang akan digunakan. Kelas domain paling umum :

1. Entertainment : rekomendasi film dan musik
2. Content : personalisasi berita, dokukumen, dan web page
3. E-commerce : rekomendasi produk untuk di beli
4. Services : rekomendasi servis travel, hotel, dan rumah

#### Evaluasi

Sebuah sistem rekomendasi banyak digunakan untuk memberikan prediksi berupa saran item yang sesuai dengan minat pengguna. Prediksi yang diberikan sistem rekomendasi memiliki nilai keakuratan yang dapat berbeda sesuai dengan kasus yang dihadapi dan juga algoritma yang digunakan. Prediksi yang diberikan harus akurat, oleh karena itu diperlukan evaluasi pada sistem rekomndasi. Evaluasi dapat menggunakan tiga metode yaitu :

1. *Offline*  
Metode *offline* dilakukan dengan cara menjalankan beberapa algoritma pada data yang sama dan membandingkan kinerjanya.

### 2. Online

Metode *online* dilakukan saat perangkat lunak sudah diluncurkan dan melibatkan pengguna nyata.

### 3. Focused user study

Metode *Focused user study* dilakukan saat metode *online* tidak layak dilakukan atau terlalu beresiko.

## 2.2 Cluster

*Clustering* adalah algoritma yang menganalisis objek data tanpa perlu label kelas. *Clustering* bisa digunakan untuk menghasilkan label kelas untuk sebuah kelompok data. Tujuan dari algoritma *cluster* adalah untuk meminimalkan jarak *intra-cluster* sekaligus memaksimalkan jarak *inter-cluster*. Kesamaan ditentukan dengan menggunakan ukuran jarak. Kelompok yang dihasilkan akan memiliki anggota yang memiliki kesamaan yang tinggi satu sama lain didalam kelompok yang sama dan berbeda dengan kelompok lain.

### 2.2.1 K-Means

K-Means adalah algoritma *clustering* yang termasuk kategori *partisial*. Kategori partisial adalah membagi item kedalam *non-overlapping cluster* sehingga setiap item hanya ada pada satu *cluster*. *Cluster* akan dibentuk sebanyak K dan untuk setiap *cluster* memiliki *centroid* awal. *Centroid* untuk setiap *cluster* adalah titik dimana jumlah jarak minimum dari semua item dalam *cluster*. Berikut merupakan tahapan-tahapan dalam pembuatan *cluster* :

1. Memilih secara acak K *centroid*
2. Menghitung jarak tiap objek ke titik *centroid* menggunakan *euclidean distance*
3. Menghitung *centroid* baru dari anggota yang berada didalam *centroid*
4. Lakukan tahap 1-3 hingga *centroid* konvergen

## 2.3 Library PHP-ML

PHP-ML adalah sebuah *library* yang khusus dibuat untuk *Machine Learning* dengan menggunakan bahasa pemrograman PHP. Terdapat lebih dari 20 algoritma yang bisa digunakan. *Library* ini bersifat *open source* yang berlisensi MIT. Versi PHP minimal untuk menggunakan *library* ini adalah PHP 7.1, penginstallan dapat menggunakan Composer.

### 2.3.1 Array Dataset

*Array Dataset* adalah bagian dari fitur *Dataset* yang disediakan oleh PHP-ML. *Array Dataset* adalah kelas yang berfungsi untuk menyimpan data sebagai tipe array dalam PHP. Menerapkan *interface Dataset* yang banyak digunakan di kelas lain. Kelas ini memiliki dua parameter yaitu : *samples* dan *labels*. *Samples* adalah array yang berisikan sample. *Labels* adalah array yang berisikan *label* setiap *sample*.

### 2.3.2 Random Split

*Random Split* adalah bagian dari fitur *Cross Validation* yang disediakan oleh PHP-ML. Kelas *Random Split* adalah salah satu metode paling sederhana dari *Cross Validation*. *Samples* dibagi menjadi dua kelompok yaitu : *train group* dan *test group*. Kelas ini memiliki tiga parameter yaitu : *dataset*, *testSize*, dan *seed*. *Dataset* adalah objek yang mengimplementasikan *interface Dataset*.

*TestSize* adalah bilangan *float* yang menyatakan seberapa banyak anggota pada *test group* dengan nilai dasar 0.3 jika parameter tidak diisi. *Seed* untuk *random generator*.

## 2.4 Universitas Katolik Parahyangan

Perguruan tinggi adalah satuan pendidikan yang menyelenggarakan pendidikan tinggi yang dapat berbentuk akademi, politeknik, sekolah tinggi, institut, atau universitas. Pendidikan tinggi adalah kelanjutan pendidikan menengah yang diselenggarakan untuk menyiapkan peserta didik menjadi anggota masyarakat yang memiliki kemampuan akademik dan/atau profesional yang dapat menerapkan, mengembangkan dan/atau menciptakan ilmu pengetahuan, teknologi dan/atau kesenian.

Universitas Katolik Parahyangan adalah sebuah universitas atau Perguruan tinggi katolik pertama yang didirikan pada 17 Januari 1955. Saat ini terletak di Jalan Ciumbuleuit No.94, Bandung, Jawa Barat, Indonesia. Terdapat tujuh fakultas dengan total program studi yaitu tujuh belas dengan enam belas program studi sarjana dan satu program studi D3.

Terdapat beberapa jalur penerimaan mahasiswa baru yang dilakukan oleh Universitas Katolik Parahyangan. Jalur penerimaan diselenggarakan secara mandiri, berikut jalur penerimaan yang disediakan Universitas Katolik Parahyangan :

1. Penelusuran Minat dan Kemampuan (PMKD) atau jalur prestasi  
PMDK adalah satu jalur penerimaan mahasiswa baru yang dilaksanakan dengan seleksi berdasarkan pada nilai raport SMA di kelas X (Sepuluh) dan XI (Sebelas), tanpa ujian tertulis. Tujuan dari PMDK untuk menjaring siswa-siswa yang berprestasi. PMDK dilakukan hanya satu kali dalam satu tahun penerimaan.
2. Ujian Saringan Masuk (USM)  
USM adalah satu jalur penerimaan mahasiswa baru yang dilaksanakan dengan mengerjakan soal yang disediakan oleh Universitas Katolik Parahyangan. Terdapat dua tempat pelaksanaan untuk USM, pertama dilaksanakan di Universitas Katolik Parahyangan dan kedua dilaksanakan di sekolah-sekolah (*on-site test*). Tujuan dari USM untuk menjaring mahasiswa baru yang memiliki kemampuan akademik untuk mengikuti dan menyelesaikan pendidikan di Universitas Katolik Parahyangan sesuai dengan batas waktu (masa studi) yang ditetapkan.

### 2.4.1 Program Studi

Program studi adalah kesatuan rencana belajar sebagai pedoman penyelenggaraan pendidikan akademik dan/atau profesional yang diselenggarakan atas dasar suatu kurikulum serta ditujukan agar mahasiswa dapat menguasai pengetahuan, keterampilan, dan sikap sesuai dengan sasaran kurikulum. Kurikulum pendidikan tinggi adalah seperangkat rencana dan pengaturan mengenai isi maupun bahan kajian dan pelajaran serta cara penyampaian dan penilaiannya yang digunakan sebagai pedoman penyelenggaraan kegiatan belajar - mengajar di perguruan tinggi.

Terdapat tujuh fakultas yang ada di Universitas Katolik Parahyangan, yaitu :

1. Fakultas Ekonomi
2. Fakultas Hukum
3. Fakultas Ilmu Sosial dan Ilmu Politik
4. Fakultas Teknik
5. Fakultas Falsafah dan Peradaban
6. Fakultas Teknologi Industri
7. Fakultas Teknologi Informasi dan Sains



## **Fakultas Ekonomi**

Terdapat empat program studi pada fakultas Ekonomi, yaitu : Ekonomi Pembangunan, Manajemen, Akuntansi, Manajemen Perusahaan. Manajemen Perusahaan merupakan program studi D3 yang ada di Universitas Katolik Parahyangan. Berikut merupakan penjelasan program studi yang ada pada Fakultas Ekonomi :

### **1. Ekonomi Pembangunan**

Mempelajari persoalan pembangunan ekonomi yang sudah, sedang, dan akan terjadi di negara berkembang. Menganalisis isu perekonomian untuk mencari dan menemukan solusi dari berbagai persoalan ekonomi secara kritis, kreatif, dan inovatif. Program studi Ekonomi Pembangunan mempersiapkan mahasiswanya untuk menjadi perencana bidang pembangunan ekonomi. Ekonomi Pembangunan adalah cabang ilmu ekonomi. Mempelajari pembangunan industri, perbankan, keuangan, dan bisnis. Berkutat dengan analisis berbagai isu perekonomian untuk mendapatkan solusi dari persoalan ekonomi.

Terdapat tiga peminatan pada program studi Ekonomi Pembangunan, yaitu :

- Ekonomi Industri dan Perdagangan
- Ekonomi Kawasan dan Lingkungan
- Ekonomi Moneter dan Keuangan

### **2. Manajemen**

Mempelajari bagaimana mengelola suatu perusahaan atau organisasi. Fokus pada kegiatan mengelola, merencanakan, dan mengatur semua proses dalam perusahaan untuk mencapai tujuan.

Terdapat satu peminatan pada program studi Manajemen, yaitu :

- Manajemen

### **3. Akuntansi**

Mempelajari mengenai keuangan dan ilmu ekonomi, Mahasiswa pada program studi Akuntansi akan memiliki pengetahuan dan penguasaan materi tentang keuangan dan ilmu ekonomi. Mampu mengelola keuangan bisnis.

Terdapat satu peminatan pada program studi Akuntansi, yaitu :

- Akuntansi

## **Fakultas Hukum**

Terdapat satu program studi pada Fakultas Hukum, yaitu : Ilmu Hukum.

### **1. Ilmu Hukum**

Mempelajari tentang hukum baik praktek maupun teori. Hukum mengatur bagaimana manusia bertindak dan bertingkah laku agar tidak merugikan orang lain. Mendalami konsep, teori, dan beberapa kasus hukum yang terjadi.

Terdapat satu peminatan pada program studi Ilmu Hukum, yaitu :

- Ilmu Hukum

### **Fakultas Ilmu Sosial dan Ilmu Politik**

Terdapat tiga program studi pada Fakultas Ilmu Sosial dan Ilmu Politik, yaitu : Ilmu Administrasi Publik, Ilmu Administrasi Bisnis, dan Ilmu Hubungan Internasional.

#### **1. Ilmu Administrasi Publik**

Mempelajari seluk beluk pemerintahan, masyarakat, dan kebijakan publik, sistem pemerintahan, pembuatan kebijakan hingga pengimplementasian dan evaluasi, pelayanan masyarakat, dan segala sesuatu yang berkaitan dengan birokrasi.

Terdapat satu peminatan pada program studi Ilmu Administrasi Publik, yaitu :

- Ilmu Administrasi Publik

#### **2. Ilmu Administrasi Bisnis**

Mempelajari mengenai kegiatan operasional bisnis dan perusahaan, yaitu : pemasaran (marketing), pengelolaan keuangan, pengelolaan personalia (SDM), hingga kegiatan produksi. Mempelajari untuk membuat produk sendiri, bukan membuat, menjual, dan mendapatkan keuntungan, tetapi menciptakan value pada produk yang dipasarkan. Mempelajari urusan klarikal kantor, mengelola sarana dan prasarana kantor, memproses data secara akurat, dan mengelola informasi yang berhubungan dengan pekerjaan kantor. Program studi ini cocok dengan orang yang memiliki ketertarikan dalam bidang pengurusan dokumen.

Terdapat dua peminatan pada program studi Ilmu Administrasi Bisnis, yaitu :

- *General Business*
- *Digital Business*

#### **3. Ilmu Hubungan Internasional**

Mempelajari mengenai interaksi, relasi, dan komunikasi yang terjadi secara internasional. Tidak hanya mempelajari hubungan diplomasi satu negara dengan negara lain, tapi juga konflik, kesejahteraan, ekonomi, dan perdamaian dunia. Beberapa kajian diplomasi dan negosiasi, politik luar negeri, perdagangan luar negeri, politik internasional, ekonomi internasional, hukum internasional, globalisasi, dll. Diasah mengenai isu-isu global, tokoh-tokoh, dan organisasi internasional yang berpengaruh, dan kerjasama internasional.

Terdapat satu peminatan pada program studi Ilmu Administrasi Bisnis, yaitu :

- Ilmu Hubungan Internasional

### **Fakultas Teknik**

Terdapat dua program studi pada Fakultas Teknik, yaitu : Teknik Sipil dan Arsitektur.

#### **1. Teknik Sipil**

Mempelajari proses merancang, membangun, dan merenovasi gedung serta infrastruktur lain, seperti jalan, jembatan, bendungan, dan infrastruktur lainnya. Memahami unsur-unsur bangunan seperti beton, baja, aspal, dan lain-lain. Mempelajari perancangan struktur bangunan yang kuat, layak, dan efisien.

Terdapat satu peminatan pada program studi Teknik Sipil, yaitu :

- Teknik Sipil

## 2. Arsitektur

Mempelajari desain dan rancangan konstruksi bangunan. Lebih menuangkan ide, konsep, dan desain di atas kertas, sedangkan realisasi akan dikerjakan oleh teknik sipil. Harus mempelajari kekuatan bangunan (firmitasi), estetika atau keindahan bangunan (venustas), dan fungsi bangunan (utilitas).

Terdapat satu peminatan pada program studi Arsitektur, yaitu :

- Arsitektur

## Fakultas Falsafah dan Peradaban

Terdapat satu program studi pada Fakultas Falsafah dan Peradaban, yaitu : Ilmu Filsafat.

### 1. Ilmu Filsafat

Filsafat sebagai induk semua ilmu, filsafat lebih mempelajari tentang permasalahan mendasar manusia dan hubungannya dengan realita. Bersifat abstrak dan memerlukan pemahaman yang mendasar. Kajian utamanya yaitu tujuan hidup, esensi manusia, moralitas, dan hati nurani. Mempelajari pemikiran para filsuf. Membantu berpikir secara terstruktur dan mampu memproses informasi secara jernih.

Terdapat dua peminatan pada program studi Ilmu Filsafat, yaitu :

- Filsafat Keilahian
- Filsafat Budaya

## Fakultas Teknologi Industri

Terdapat tiga program studi pada Fakultas Teknologi Industri, yaitu : Teknik Industri, Teknik Kimia, dan Teknik Elektro.

### 1. Teknik Industri

Mempelajari proses industri baik dari sisi manajemen ataupun teknik. Turunan dari teknik mesin. Mempelajari disiplin ilmu lain seperti matematika, fisika, fisiologi, dan manajemen saintifik. Teknik Industri berfokus pada perancangan, peningkatan, dan pemasangan sistem terintegrasi yang membutuhkan manusia, material, peralatan, dan energi. Memiliki tiga bidang dan satu sistem manufaktur (mempelajari peningkatan kualitas, produktivitas, dan efisiensi sistem produk), dua manajemen industri (mempelajari manajemen keuangan, operasional, manajemen inovasi, perencanaan dan pengendalian produksi, dan ekonomi teknik), dan tiga sistem industri dan tekno ekonomi, seperti logistik, statistik, penelitian operasional, dan sistem basis data.

Terdapat satu peminatan pada program studi Teknik Industri, yaitu :

- Teknik Industri

### 2. Teknik Kimia

Cabang ilmu teknik yang mempelajari bagaimana proses dan cara mengubah bahan baku/mentah dan bahan kimia menjadi sebuah produk yang lebih bernilai secara komersial maupun perubahan sifat fisik dan kimia bahan mentah. Dididik untuk merencanakan dan merancang alat-alat proses, mengoperasikan, mengendalikan dan memelihara pabrik/industri, mengkontruksi pendirian suatu pabrik, mengadakan penelitian dan pengembangan proses, serta merencanakan serta mengelola penjualan dan pelayanan.

Terdapat satu peminatan pada program studi Teknik Kimia, yaitu :

- Teknik Kimia

### 3. Teknik Elektro

Mempelajari sifat-sifat elektron yang kita kenal sebagai listrik, mempelajari aplikasi dan pemanfaatan listrik dalam kehidupan sehari-hari, serta teknologi yang terkait. Cakupannya meliputi pembangkit tenaga listrik, sistem jaringan distribusi, pemanfaatan oleh pengguna akhir.

Terdapat satu peminatan pada program studi Teknik Elektro, yaitu :

- Mekatronika

## Fakultas Teknologi Informasi dan Sains

Terdapat tiga program studi pada Fakultas Teknologi Informasi dan Sains, yaitu : Matematika, Fisika, dan Teknik Informatika.

### 1. Matematika

Mempelajari matematika murni seperti aljabar, geometri, dan analisis matematika; statistika; komputasi; aktuaria; dan riset operasi.

Terdapat dua peminatan pada program studi Matematika, yaitu :

- Aktuaria
- Matematika Terapan

### 2. Fisika

Mempelajari gejala alam yang tidak hidup atau materi dalam lingkup ruang dan waktu, mempelajari perilaku dan sifat materi dalam bidang yang sangat beragam (partikel submikroskopis - perilaku materi alam semesta sebagai satu kesatuan kosmos). Ilmu fisika sangat mendukung perkembangan teknologi, yaitu industri, komunikasi, rekayasa, kimia, dan kedokteran.

Terdapat satu peminatan pada program studi Fisika, yaitu :

- Fisika

### 3. Teknik Informatika

Mempelajari dan menerapkan prinsip-prinsip ilmu komputer dan analisa matematis untuk desain, pengembangan, pengujian, evaluasi perangkat lunak, sistem operasi, dan kerja komputer. Menghasilkan ide kreatif, merealisasikan ide, mendiferensiasikan berbagai macam fungsi, dan menciptakan struktur instruksi yang sangat detail dalam bahasa pemrograman untuk mengajarkan komputer apa yang harus dilakukan.

Terdapat dua peminatan pada program studi Teknik Informatika, yaitu :

- *Data Science*
- *Computer Science*

### 2.4.2 Syarat Masuk Program Studi

Berikut merupakan syarat untuk program studi yang ada di Universitas Katolik Parahyangan :

Program Studi	Syarat Jurusan	USM	PMDK	Syarat Khusus
Ekonomi Pembangunan	IPA IPS	Matematika Bahasa Inggris	Matematika Bahasa Indonesia Bahasa Inggris	
Manajemen	IPA IPS	Matematika Bahasa Inggris	Matematika Bahasa Inggris	
Akuntansi	IPA IPS	Matematika Bahasa Inggris	Matematika Bahasa Inggris	
Ilmu Hukum	IPA IPS Bahasa	Matematika Bahasa Inggris	Matematika Bahasa Inggris Pendidikan Kewarganegaraan	
Ilmu Administrasi Publik	IPA IPS Bahasa SMK	Matematika Bahasa Inggris	Matematika Bahasa Inggris	
Ilmu Administrasi Bisnis	IPA IPS Bahasa SMK	Matematika Bahasa Inggris	Matematika Bahasa Inggris	
Ilmu Hubungan Internasional	IPA IPS Bahasa	Matematika Bahasa Inggris	Matematika Bahasa Inggris Uraian Bahasa Inggris	
Teknik Sipil	IPA	Matematika Bahasa Inggris Fisika	Matematika Bahasa Inggris Fisika	
Arsitektur	IPA	Matematika Bahasa Inggris Gambar	Matematika Bahasa Inggris Gambar	
Ilmu Filsafat	IPA IPS Bahasa SMK	Matematika Bahasa Inggris Wawancara	Bahasa Inggris Bahasa Indonesia	
Teknik Industri	IPA	Matematika Matematika	Bahasa Inggris Bahasa Inggris	
Teknik Kimia	IPA	Matematika Bahasa Inggris Fisika	Matematika Bahasa Inggris Fisika Kimia	Tidak buta warna
Teknik Elektro	IPA	Matematika Bahasa Inggris Fisika	Matematika Bahasa Inggris Fisika	Tidak buta warna
Matematika	IPA	Matematika Bahasa Inggris	Matematika Bahasa Inggris	
Fisika	IPA	Matematika	Matematika	

		Bahasa Inggris Fisika	Bahasa Inggris	
Teknik Informatika	IPA	Matematika Bahasa Inggris	Matematika Bahasa Inggris	

Tabel 2.1: Tabel syarat program studi

### 2.4.3 Karakteristik Program Studi

Berikut merupakan kriteria untuk calon mahasiswa sesuai dengan program studi :

Program Studi	Karakteristik
Ekonomi Pembangunan	Tertarik dengan Ilmu Ekonomi Tertarik dengan perhitungan Berpikir kritis Senang menganalisis Mampu memecahkan masalah
Manajemen	Keterampilan komunikasi Senang menganalisis Senang memecahkan masalah
Akuntansi	Tertarik dengan akuntansi Memiliki kemampuan berhitung yang kuat dan teliti Senang menganalisis
Ilmu Hukum	Tertarik dengan hukum Teliti dan berpikir kritis Keterampilan komunikasi Mampuan menganalisis
Ilmu Administrasi Publik	Terstruktur Senang menganalisis Senang memecahkan masalah
Ilmu Administrasi Bisnis	Memiliki minat yang tinggi untuk usaha Kemampuan komunikasi Kemampuan berhitung Terstruktur
Ilmu Hubungan Internasional	Tertarik dengan interaksi internasional Kemampuan berbahasa Inggris Berwawasan luas Kemampuan komunikasi
Teknik Sipil	Senang berhitung Terstruktur
Arsitektur	Tertarik dengan desain dan rancangan bangunan Tertarik dengan menggambar dan seni Tertarik dengan humaniora, sains, dan teknologi
Ilmu Filsafat	Tipe pemikir Berwawasan luas Berpikir Rasional Berpikir kritis
Teknik Industri	Senang berhitung Terstruktur
Teknik Kimia	Tertarik dengan Kimia Senang berhitung

	Terstruktur Tidak buta warna
Teknik Elektro	Tidak buta warna Senang berhitung Terstruktur Teliti
Matematika	Tertarik dengan Matematika Senang memecahkan masalah Terstruktur Teliti
Fisika	Senang berhitung Senang menganalisis Mampu memecahkan masalah Teliti
Teknik Informatika	Tertarik dengan teknologi Senang menganalisis Senang memecahkan masalah Senang berhitung

Tabel 2.2: Tabel kriteria



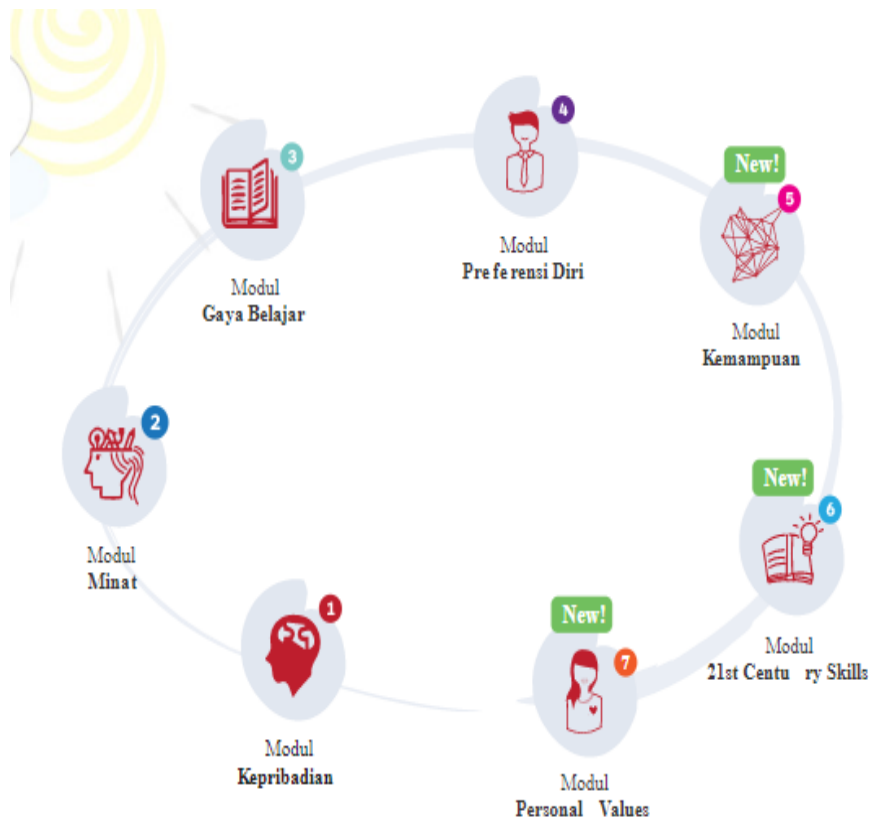


## BAB 3

### ANALISIS

#### 3.1 Analisis Perangkat Lunak Sejenis

Salah satu *website* yang dapat memberikan rekomendasi program studi adalah <https://rencanamu.id>. Sistem tersebut dikembangkan menggunakan riset ilmiah, Rencanamu mengukur 7 dimensi profil siswa sebagai landasan dalam rekomendasi, perencanaan kuliah dan karier yang terintegrasi, berkesinambungan dan menyeluruh. Gambar 3.1 menunjukkan 7 dimensi profil siswa.

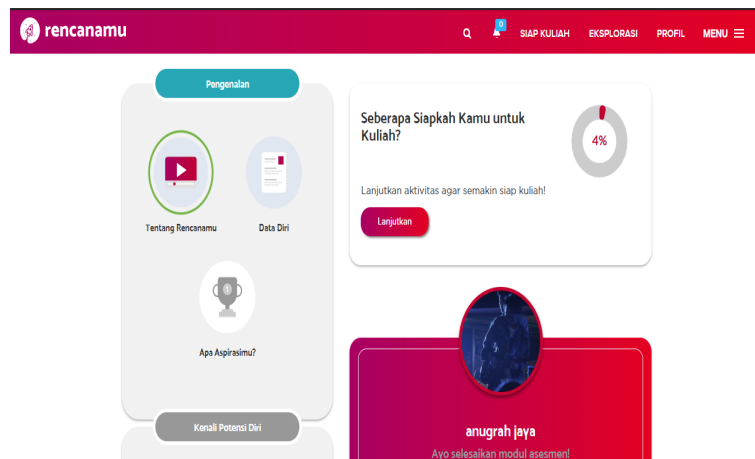


Gambar 3.1: 7 Dimensi Profil Siswa

Pada sistem ini, telah dilakukan beberapa analisis dan hasilnya sebagai berikut :

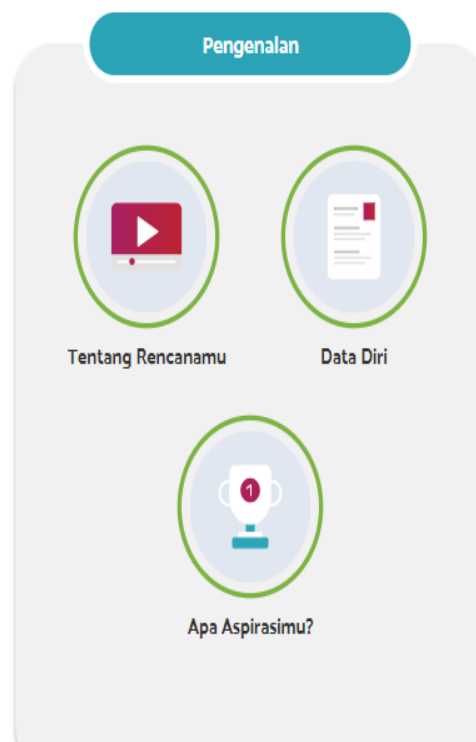
1. Website <https://rencanamu.id> adalah sebuah *platform* persiapan kuliah dan karier *online* berbasis data didukung oleh teknologi *People Science* untuk membantu siswa dalam merancang dan mempersiapkan masa depan mereka.
2. Perlu melakukan registrasi atau *login* kedalam sistem.

3. Gambar 3.2 merupakan tampilan awal setelah registrasi atau *login*.

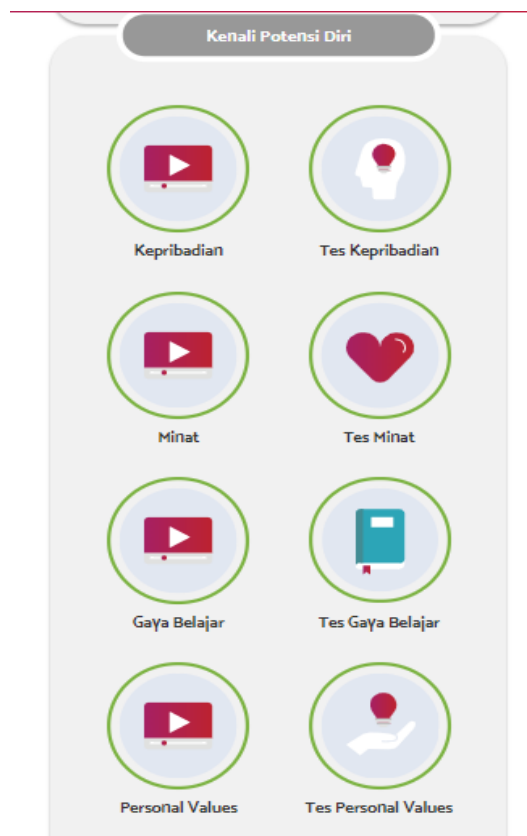


Gambar 3.2: Tampilan setelah registrasi atau *login*

4. Gambar 3.3, Gambar 3.4, dan Gambar 3.5 adalah beberapa modul yang harus dikerjakan.



Gambar 3.3: Modul Pengenalan

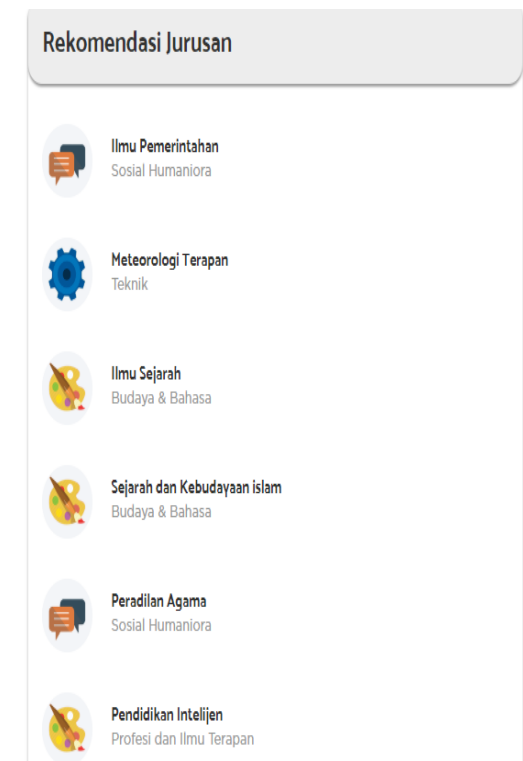


Gambar 3.4: Modul Potensi Diri



Gambar 3.5: Modul Ukur Kemampuan Diri

5. Gambar 3.6 adalah contoh hasil rekomendasi yang diberikan sistem berdasarkan modul yang sudah dikerjakan.



Gambar 3.6: Hasil Rekomendasi

Website <https://rencanamu.id> memiliki kesamaan dengan sistem yang dibangun yaitu memberikan rekomendasi program studi untuk anak SMA. Perbedaannya pada website <https://rencanamu.id> tidak menampilkan prediksi IPK dan harus mengisi beberapa modul untuk mendapatkan rekomendasi program studi.

### 3.2 *Preprocessing* Data Mahasiswa

Data mahasiswa yang digunakan adalah data mahasiswa Universitas Katolik Parahyangan dengan jalur penerimaan Penelurusan dan Kemampuan atau PMDK pada tahun 2013-2018. Pada data yang digunakan terdapat beberapa atribut yang tidak dapat digunakan seperti No.PMB, kota asal sekolah, dan provinsi asal sekolah. Atribut yang tidak dapat digunakan akan dihapus dan data akan dipisahkan menjadi dua *file* mahasiswa dan nilai untuk setiap program studi yang ada. *preprocessing* dilakukan menggunakan Python. Berikut langkah-langkah dalam *preprocessing* :

1. Membaca file .csv yang berisikan data mahasiswa pada fakultas tertentu.
2. Membuat dataframe untuk menampung data mahasiswa dan nilai.
3. Menginisialisasikan batas *looping*, id\_user, id\_nilai, dan asal jurusan.
4. Menambahkan data mahasiswa berupa NPM, id\_prodi, asal jurusan, dan IPK pada dataframe mahasiswa.
5. Mengubah *range* nilai menjadi GPA (*Grade Point Average*) dan menghitung nilai rata-rata untuk setiap nilai mata pelajaran.
6. Menambahkan data GPA, rata-rata nilai, dan id\_user pada dataframe nilai.

7. Menyimpan dataframe mahasiswa dan nilai menjadi .csv.

Hasil *file* .csv nantinya akan *diimport* pada basis data yang akan digunakan pada sistem.

### 3.3 Pemilihan Algoritma Sistem Rekomendasi

Berdasarkan teori 2.1.3 yang menjelaskan mengenai teknik-teknik yang dapat digunakan untuk membangun sistem rekomendasi, teknik *collaborative filtering* adalah teknik yang dapat digunakan untuk memberikan rekomendasi berupa program studi kepada calon mahasiswa berdasarkan kesamaan dengan pengguna lain. Berikut merupakan beberapa hal mengapa memilih teknik *collaborative filtering* :

1. *Collaborative filtering* menghasilkan rekomendasi item yang spesifik untuk pengguna berdasarkan peringkat tanpa memerlukan informasi tambahan mengenai item ataupun pengguna.

Teknik *collaborative filtering* memiliki beberapa kekurangan diantaranya :

1. Rekomendasi yang diberikan mengambil data yang cukup banyak dari basis data sehingga membutuhkan memori yang besar.
2. Tidak bisa memberikan rekomendasi untuk item yang tidak pernah diberikan *rating* oleh pengguna.
3. Pengguna harus memberikan *rating* untuk beberapa atribut agar bisa diberikan rekomendasi.

Pada sistem yang dibangun, akan memberikan rekomendasi berdasarkan nilai raport beberapa mata pelajaran siswa pada kelas 10 dan 11 yang digunakan untuk PMDK. Mata pelajaran yang digunakan adalah Matematika, Bahasa Indonesia, Bahasa Inggris, Fisika, Kimia, dan Pendidikan Kewarganegaraan. Rekomendasi program studi berdasarkan asal jurusan saat SMA, misalnya siswa IPA akan diberikan rekomendasi program studi IPA.

### 3.4 Contoh Perhitungan Pearson Correlation

Berdasarkan 2.1.4 terdapat langkah-langkah perhitungan dari *user-based*. Tabel 3.1 merupakan contoh data mahasiswa.

MP/Semester	101	102	111	112	AVG
Matematika	2.6	2.9	2.95	2.75	2.8
Bahasa Indonesia	0	0	0	0	0
Bahasa Inggris	2.95	3	2.85	2.95	2.9375
PKN	0	0	0	0	0

Tabel 3.1: Contoh data mahasiswa dalam bentuk GPA

Berikut merupakan contoh langkah-langkah perhitungan :

1. Menghitung nilai rata-rata *rating*.

MP/Semester	101	102	111	112	Rumus	AVG
Matematika	2.9	3.4	3.4	2.9	$\frac{2.9+3.4+3.4+2.9}{4}$	3.15
Bahasa Indonesia	2.95	2.9	3.9	3.4	$\frac{2.95+2.9+3.9+3.4}{4}$	3.2875
Bahasa Inggris	3.3	3.35	3.25	2.9	$\frac{3.3+3.35+3.25+2.9}{4}$	3.2
PKN	3.4	2.9	3.35	2.35	$\frac{3.4+2.9+3.35+2.35}{4}$	3

Tabel 3.2: Contoh data siswa dalam bentuk GPA

## 2. Menghitung kesamaan atau similaritas.

No	Rumus	Matematika	Rumus	Bahasa Inggris
1	$(2.9 - 3.15)*$ $(2.6 - 2.8)$	0.5	$(3.3 - 3.2)*$ $(2.95 - 2.9375)$	0.00125
2	$(3.4 - 3.15)*$ $(2.9 - 2.8)$	0.025	$(3.35 - 3.2)*$ $(3 - 2.9375)$	0.009375
3	$(3.4 - 3.15)*$ $(2.95 - 2.8)$	0.0375	$(3.25 - 3.2)*$ $(2.85 - 2.9375)$	-0.004375
4	$(2.9 - 3.15)*$ $(2.75 - 2.8)$	0.0125	$(2.9 - 3.2)*$ $(2.95 - 2.9375)$	-0.00375
Sigma	$0.5 + 0.025 +$ $0.0375 + 0.0125$	0.125	$0.00125 + 0.009375 +$ $-0.004375 + -0.00375$	0.0025
Hasil			$0.125 + 0.0025$	0.1274

Tabel 3.3: Nilai kovariasi mahasiswa dan siswa

No	Rumus	Matematika	Rumus	Bahasa Inggris
1	$(2.9 - 3.15)^2$	0.0625	$(3.3 - 3.2)^2$	0.01
2	$(3.4 - 3.15)^2$	0.0625	$(3.35 - 3.2)^2$	0.0225
3	$(3.4 - 3.15)^2$	0.0625	$(3.25 - 3.2)^2$	0.0025
4	$(2.9 - 3.15)^2$	0.0625	$(2.9 - 3.2)^2$	0.09
Sigma	$0.0625+0.0625+$ $0.0625+0.0625$	0.25	$0.01+0.0225+$ $0.0025+0.09$	0.125
Hasil			$\sqrt{0.25 + 0.125}$	0.612372436

Tabel 3.4: Standar Deviasi Siswa

No	Rumus	Matematika	Rumus	Bahasa Inggris
1	$(2.6 - 2.8)^2$	0.04	$(2.95 - 2.9375)^2$	0.00015625
2	$(2.9 - 2.8)^2$	0.01	$(3 - 2.9375)^2$	0.00390625
3	$(2.95 - 2.8)^2$	0.0225	$(2.85 - 2.9375)^2$	0.00765625
4	$(2.75 - 2.8)^2$	0.0225	$(2.95 - 2.9375)^2$	0.00015625
Sigma	0.04+0.01+ 0.0225+0.0225	0.075	0.0.000156250.00390625+ 0.00765625+0.00015625	0.011875
Hasil			$\sqrt{0.075 + 0.011875}$	0.294745653

Tabel 3.5: Standar Deviasi Mahasiswa

No	Rumus	Kesamaan	IPK
1	$\frac{0.1275}{0.612372436 \cdot 0.294745653}$	0.706394228	3.11
2	$\frac{0.0125}{0.612372436 \cdot 0.2343242}$	0.08711185	2.9
3	$\frac{0.2}{0.612372436 \cdot 0.543242}$	0.601202838	3
4	$\frac{0.125}{0.612372436 \cdot 0.432343}$	0.472134729	3.2
5	$\frac{0.05}{0.612372436 \cdot 0.242345}$	0.336914969	3.4

Tabel 3.6: Contoh Perhitungan kesamaan

- Memilih nilai kesamaan atau similaritas yang bernilai lebih dari 0.  
Berdasarkan langkah nomor 2, maka nilai kesamaan pada tabel 3.6 semuanya dapat digunakan untuk prediksi.
- Menghitung nilai prediksi.

No	Kesamaan	Rumus	Kesamaan*IPK
1	0.706394228	$0.706394228 \cdot 3.11$	2.196886049
2	0.08711185	$0.08711185 \cdot 2.9$	0.252624364
3	0.601202838	$0.601202838 \cdot 3$	1.803608514
4	0.472134729	$0.472134729 \cdot 3.2$	1.510831133
5	0.336914969	$0.336914969 \cdot 3.4$	1.145510893
Sigma	2.203758614	-	6.909460954
Hasil		$\frac{6.909460954}{2.203758614}$	3.13530752

Tabel 3.7: Contoh hasil Prediksi

### 3.5 Contoh Perhitungan Metode Evaluasi Sistem Rekomendasi

Berdasarkan penjelasan mengenai sistem rekomendasi yang dibahas pada bab 2 bagian 2.1.5. Salah satu bagian terpenting adalah evaluasi. Evaluasi pada sistem rekomendasi dilakukan untuk

mendapatkan akurasi dari hasil prediksi yang diberikan. Akurasi merupakan salah satu aspek yang sering dijadikan acuan untuk rekomendasi yang digunakan. Dalam melakukan pengujian akurasi bisa menggunakan metode *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE). Berikut merupakan contoh penerapan kedua metode yang akan disajikan dalam tabel 3.8 :

No	Item	Pengguna	Penilaian Asli $r_{u,i}$	Prediksi Sistem $\hat{r}_{u,i}$	$r_{u,i} - \hat{r}_{u,i}$	$ r_{u,i} - \hat{r}_{u,i} $	$(r_{u,i} - \hat{r}_{u,i})^2$
1	110	1	3.4	3.5	-0.1	0.1	0.01
2	120	30	3.3	3.1	0.2	0.2	0.04
3	130	56	3	2.8	0.2	0.2	0.04
4	200	65	2.9	3.1	-0.2	0.2	0.04
5	310	76	3.1	2.8	0.3	0.3	0.09
6	320	87	3.2	3	0.2	0.2	0.04
7	330	99	2.8	3	-0.2	0.2	0.04
8	410	102	3.4	3.5	-0.1	0.1	0.01
9	420	167	3.1	3.5	-0.4	0.4	0.16
10	510	189	2.8	3.1	-0.3	0.3	0.09
11	610	298	3.1	2.9	0.2	0.2	0.04
12	620	344	3.4	2.9	0.5	0.5	0.25
13	630	365	3.1	3	0.1	0.1	0.01
14	710	465	2.9	3	-0.1	0.1	0.01
15	720	477	3.4	3.5	-0.1	0.1	0.01
16	730	480	3.4	3.6	-0.2	0.2	0.04
Jumlah						3.4	0.92

Tabel 3.8: Tabel Data MAE dan RMSE

Berdasarkan data pada tabel 3.8 jika dihitung dengan persamaan MAE yaitu :

$$MAE = \frac{1}{n} * \Sigma |r_{u,i} - \hat{r}_{u,i}| \quad (3.1)$$

maka akan didapatkan hasil pengimplementasian dari rumus MAE sebagai berikut :

$$MAE = \frac{1}{16} * 3.4 = 0.2125 \quad (3.2)$$

Berdasarkan data pada tabel 3.8 jika dihitung dengan persamaan RMSE yaitu :

$$RMSE = \sqrt{\frac{1}{n} * \Sigma (r_{u,i} - \hat{r}_{u,i})^2} \quad (3.3)$$

maka akan didapatkan hasil pengimplementasian dari rumus RMSE sebagai berikut :

$$RMSE = \sqrt{\frac{1}{16} * 0.91} = 0.0575 \quad (3.4)$$

### 3.6 Analisis Kebutuhan Sistem

Pada sistem yang akan dibangun, memiliki kebutuhan-kebutuhan perangkat lunak seperti : *User Case* dan Rancangan Basis Data.



### 3.6.1 Diagram *Use Case*

Pada sistem yang akan dibangun terdapat satu aktor yaitu Siswa/i. Siswa/i ini adalah calon mahasiswa kelas XI yang merupakan target dari sistem yang akan dibangun. Terdapat 4 langkah yang harus dilakukan, yaitu :

#### 1. Pendefinisian Aktor

No	Aktor	Deskripsi
1	Siswa/i	Siswa/i adalah orang yang akan diberikan rekomendasi program studi yang ada di Universitas Parahyangan.

Tabel 3.9: Pendefinisian Aktor

#### 2. Pendefinisian *Use Case*

No	<i>Use Case</i>	Deskripsi
1	Memilih Jurusan SMA	Merupakan proses untuk memilih jurusan saat SMA.
2	Mengisi Nilai Rapor	Merupakan proses untuk mengisi nilai beberapa nilai mata pelajaran sesuai dengan jurusan saat SMA.

Tabel 3.10: Pendefinisian *Use Case*

#### 3. Pembuatan *Use Case* Skenario

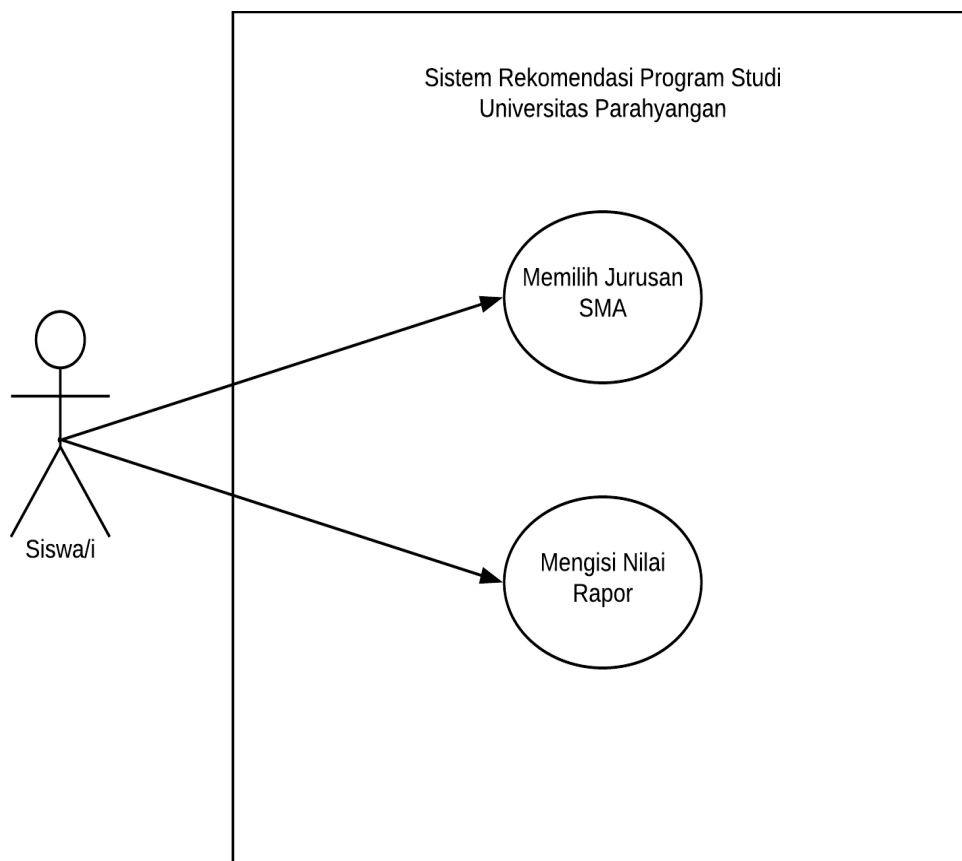
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih jurusan saat SMA.	
	2. Mengarahkan kepada form sesuai jurusan SMA.

Tabel 3.11: Skenario Memilih Jurusan SMA

Skenario Normal	
1. Mengisi nilai sesuai nilai rapor.	
	2. Memeriksa valid tidaknya data yang dimasukkan.
	3. Memeriksa range nilai.
4. Klik tombol submit.	
	5. Mengarahkan kepada halaman hasil rekomendasi.
Skenario Alternatif	
1. Mengisi nilai yang tidak valid.	
	2. Memeriksa valid tidaknya data yang dimasukkan.
	3. Memberikan pesan data tidak valid.
4. Mengisi nilai sesuai nilai rapor yang valid.	
	5. Memeriksa <i>range</i> nilai.

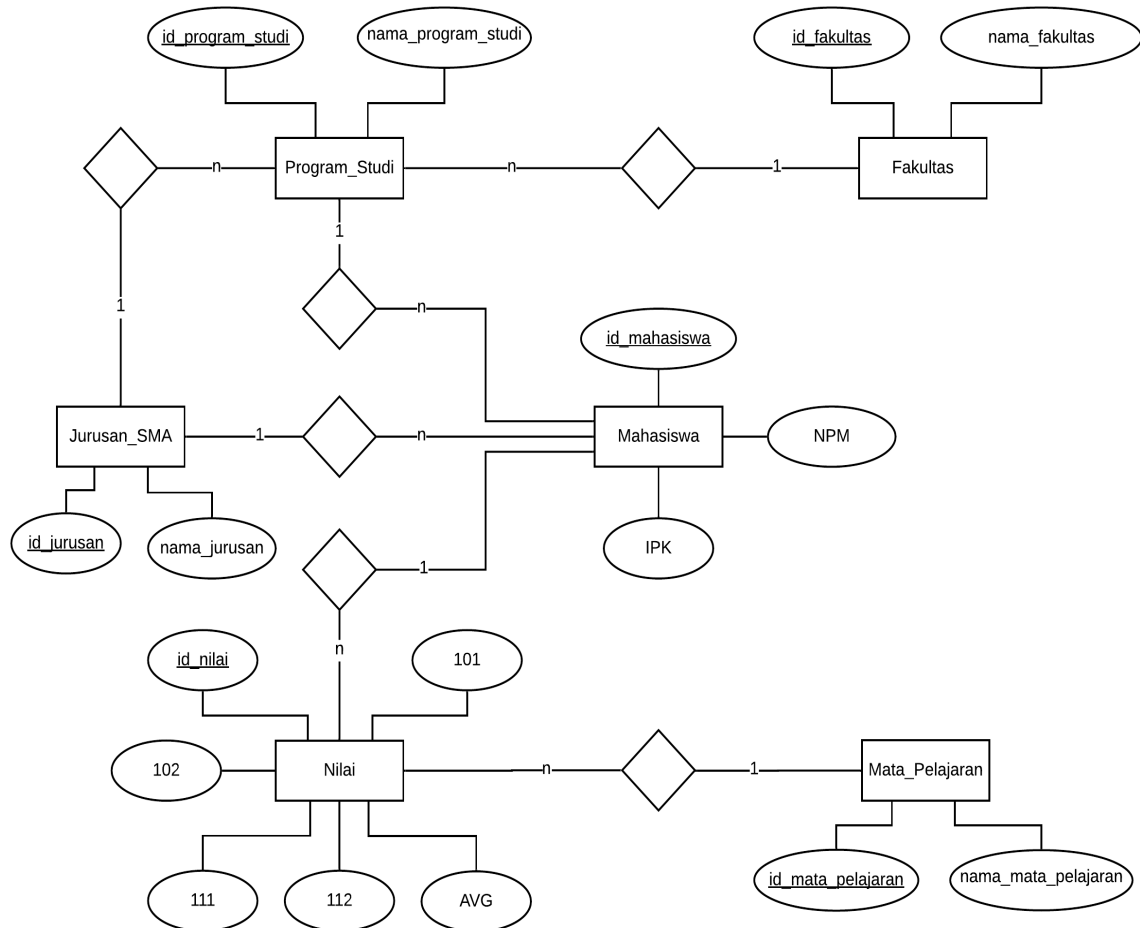
	6. Memberikan pesan <i>range</i> tidak sesuai.
7. Mengisi nilai sesuai <i>range</i> .	
	8. Memeriksa <i>range</i> nilai.
9. Klik tombol submit.	
	10. Mengarahkan kepada <i>page</i> hasil rekomendasi.

Tabel 3.12: Skenario Mengisi Nilai Rapor

4. Menggambarkan Diagram *Use Case*Gambar 3.7: Diagram *Use Case* Sistem Rekomendasi

### 3.6.2 Rancangan Basis Data

#### Diagram ERD



Gambar 3.8: Diagram ERD Sistem Rekomendasi

Berikut merupakan entitas dan atribut gambar 3.8 yang akan digunakan pada sistem yang akan dibangun :

1. Jurusan\_SMA memiliki atribut id\_jurusan dan nama jurusan.
2. Fakultas memiliki atribut id\_fakultas dan nama\_fakultas.
3. Program\_Studi memiliki atribut id\_program\_studi dan nama\_program\_studi.
4. Mahasiswa memiliki atribut id\_mahasiswa, NPM, dan IPK.
5. Mata\_Pelajaran memiliki atribut id\_mata\_pelajaran dan nama\_mata\_pelajaran.
6. Nilai memiliki atribut id\_nilai, 101, 102, 111, 112, dan AVG.



## BAB 4

### PERANCANGAN

#### 4.1 Perancangan Fisik Basis Data

##### 4.1.1 Perancangan Tabel

Tabel data yang akan digunakan untuk sistem rekomendasi program studi Universitas Katolik Parahyangan sesuai diagram ERD pada gambar 3.8 akan dirancang sesuai pada tabel berikut :

###### 1. Jurusan SMA

No	Nama Atribut	Tipe Data	Keterangan
1	id_jurusan	INT	NOT NULL, Primary Key
2	nama_jurusan	VARCHAR(25)	NOT NULL

Tabel 4.1: Perancangan Tabel jurusan\_sma

###### 2. Fakultas

No	Nama Atribut	Tipe Data	Keterangan
1	id_fakultas	INT	NOT NULL, Primary Key
2	nama_fakultas	VARCHAR(50)	NOT NULL

Tabel 4.2: Perancangan Tabel fakultas

###### 3. Program Studi

No	Nama Atribut	Tipe Data	Keterangan
1	id_program_studi	INT	NOT NULL, Primary Key
2	nama_program_studi	VARCHAR(50)	NOT NULL
3	id_fakultas	INT	NOT NULL, Foreign Key dari fakultas
4	id_jurusan	INT	NOT NULL, Foreign Key dari jurusan_sma

Tabel 4.3: Perancangan Tabel program\_studi

###### 4. Mahasiswa

No	Nama Atribut	Tipe Data	Keterangan
1	id_mahasiswa	INT	NOT NULL, Primary Key
2	NPM	VARCHAR(10)	NOT NULL
3	IPK	DOUBLE	NOT NULL
4	id_jurusan	INT	NOT NULL, Foreign Key dari jurusan
5	id_program_studi	INT	NOT NULL, Foreign Key dari program_studi

Tabel 4.4: Perancangan Tabel mahasiswa

## 5. Mata Pelajaran

No	Nama Atribut	Tipe Data	Keterangan
1	id_mata_pelajaran	INT	NOT NULL, Primary Key
2	nama_mata_pelajaran	VARCHAR(20)	NOT NULL

Tabel 4.5: Perancangan Tabel mata\_pelajaran

## 6. Nilai

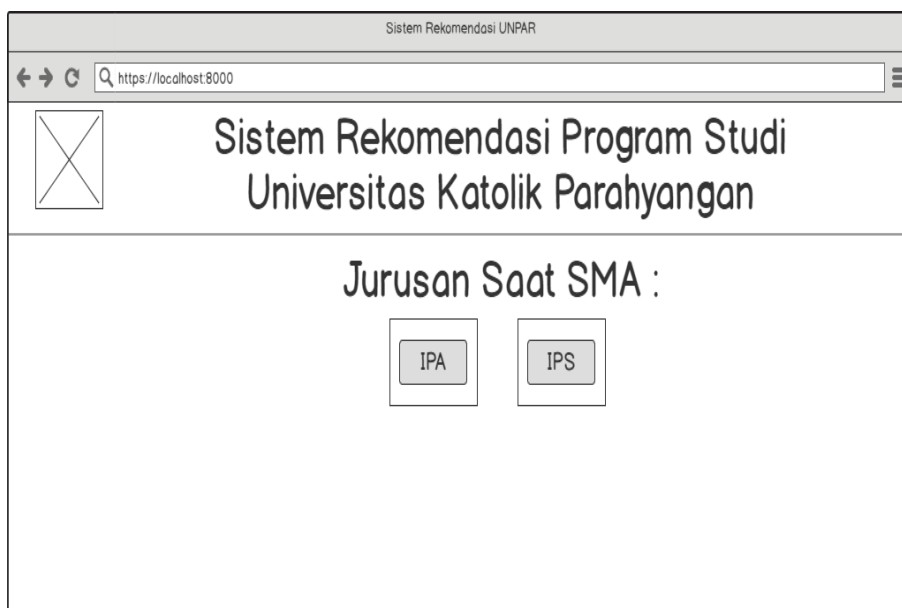
No	Nama Atribut	Tipe Data	Keterangan
1	id_nilai	INT	NOT NULL, Primary Key
2	id_mata_pelajaran	INT	Foreign Key dari mata_pelajaran
3	id_mahasiswa	INT	NOT NULL, Foreign Key dari mahasiswa
4	101	DOUBLE	Nilai kelas 10 semester 1
5	102	DOUBLE	Nilai kelas 10 semester 2
6	111	DOUBLE	Nilai kelas 11 semester 1
7	112	DOUBLE	Nilai kelas 11 semester 2
8	AVG	DOUBLE	Rata-rata nilai

Tabel 4.6: Perancangan Tabel nilai

## 4.2 Perancangan Antar Muka

Pada subbab ini akan berisikan perancangan antar muka untuk sistem rekomendasi, berikut merupakan hasil perancangan :

1. Halaman index saat siswa/i mengakses sistem



Sistem Rekomendasi UNPAR

https://localhost:8000

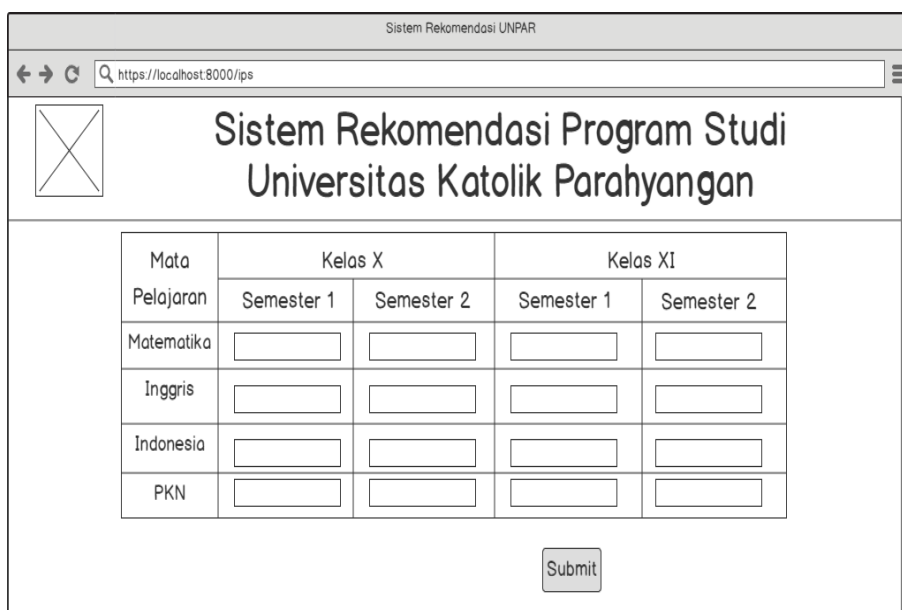
Sistem Rekomendasi Program Studi  
Universitas Katolik Parahyangan

Jurusan Saat SMA :

IPA IPS

Gambar 4.1: Halaman Index Sistem

## 2. Halaman pengisian nilai siswa/i IPA



Sistem Rekomendasi UNPAR

https://localhost:8000/ips

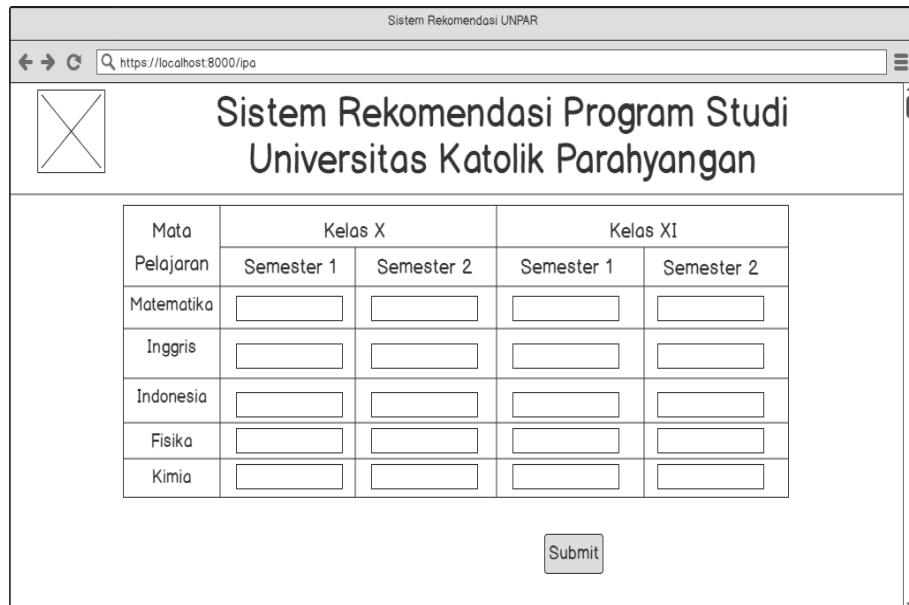
Sistem Rekomendasi Program Studi  
Universitas Katolik Parahyangan

Mata Pelajaran	Kelas X		Kelas XI	
	Semester 1	Semester 2	Semester 1	Semester 2
Matematika	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Inggris	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Indonesia	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
PKN	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Submit

Gambar 4.2: Halaman Pengisian Nilai IPA

## 3. Halaman pengisian nilai siswa/i IPS



Mata Pelajaran	Kelas X		Kelas XI	
	Semester 1	Semester 2	Semester 1	Semester 2
Matematika	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Inggris	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Indonesia	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Fisika	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Kimia	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Gambar 4.3: Halaman Pengisian Nilai IPS

#### 4. Halaman hasil rekomendasi



No	Fakultas	Program Studi	IPK
1	Teknologi Informasi dan Sains	Matematika	3.6
2	Teknologi Informasi dan Sains	Teknik Informatika	3.4
3	Fakultas Teknik	Teknik Sipil	3.3
4	Fakultas Teknik	Arsitektur	3.2
5	Fakultas Teknik Industri	Teknik Industri	3

Gambar 4.4: Halaman Hasil Rekomendasi

### 4.3 Perancangan Algoritma

Pada subbab ini akan berisikan perancangan algoritma yang digunakan pada sistem. Sistem menggunakan beberapa algoritma seperti K-Means untuk membuat kelompok mahasiswa yang memiliki karakteristik yang sama dengan calon mahasiswa, Pearson Correlation Coefficient untuk menghitung kesamaan atau similaritas, dan User-base Collaborative Filtering untuk menghitung prediksi nilai IPK.



### 4.3.1 Preprocessing Data

Pada fungsi *generateCSV* terdapat tiga *parameter* *mhs*, *nilai*, dan *count*. Parameter *mhs* dan *nilai* merupakan *dataframe* untuk menyimpan data mahasiswa dan nilai setelah diproses, sedangkan *count* digunakan untuk menyatakan berapa banyak mata pelajaran yang digunakan pada program studi di setiap fakultas. Data yang digunakan merupakan data yang berisikan mahasiswa dan nilai. Fungsi ini akan digunakan untuk data mahasiswa pada satu fakultas. Berikut merupakan *pseudocode* fungsi *generateCSV* :

---

**Algorithm 1** Generate CSV
 

---

```

1: procedure GENERATECSV(mhs, nilai, count)
2:   size  $\leftarrow$  banyaknyabarispadafiledata
3:   batas  $\leftarrow$  size / (4 * count)            $\triangleright$  4*count karena tiap satu mata pelajaran 4 semester
4:   ipa  $\leftarrow$  1                                $\triangleright$  kode jurusan ipa
5:   ips  $\leftarrow$  2                                $\triangleright$  kode jurusan ips
6:   idUser  $\leftarrow$  idU sersaatini
7:   idNilai  $\leftarrow$  idNialisaatini
8:   for idx into batas do
9:     idx  $\leftarrow$  idx * (4 * count)
10:    npm  $\leftarrow$  NPMdaridata
11:    idProdi  $\leftarrow$  idprodidaridata
12:    isJurusan  $\leftarrow$  ipaatauips                $\triangleright$  tergantung jurusan asal
13:    for i = 7 into 18 do
14:      ipk  $\leftarrow$  ambilnilaiIPKdaridata        $\triangleright$  ambil ipk yang terakhir
15:    end for
16:    insert idUser, npm, ipk, idJurusan, dan idProdi kedalam dataframe mhs
17:    for i = 0 into count do
18:      row  $\leftarrow$  idx + i
19:      idMP  $\leftarrow$  idmatapelajarandaridata
20:      m101  $\leftarrow$  (nilai/20) - 1              $\triangleright$  nilai semester 1 kelas X
21:      m102  $\leftarrow$  (nilai/20) - 1              $\triangleright$  nilai semester 2 kelas X
22:      m111  $\leftarrow$  (nilai/20) - 1              $\triangleright$  nilai semester 1 kelas XI
23:      m112  $\leftarrow$  (nilai/20) - 1              $\triangleright$  nilai semester 2 kelas XI
24:      avg  $\leftarrow$  (m101 + m102 + m111 + m112)/4
25:      insert idNilai, idMP, idUser, m101, m102, m111, m112, dan avg
26:      idNilai  $\leftarrow$  idNilai + 1
27:    end for
28:    idUser  $\leftarrow$  idUser + 1
29:  end for
30:  return mhs, nilai
31: end procedure
32: save mhs menjadi .csv
33: save nilai menjadi .csv

```

---

### 4.3.2 Mahasiswa Controller

*Mahasiswa controller* berfungsi untuk mendapatkan data mahasiswa dari model *MahasiswaModel*, berikut merupakan *pseudocode* fungsi yang terdapat pada mahasiswa controller :

1. Fungsi index

**Algorithm 2** Index

---

```

1: procedure INDEK(jurusanSMA)
2:   idJurusan  $\leftarrow$  1 ▷ IdJurusan IPA
3:   if jurusanSMA == 'IPS' then
4:     idJurusan  $\leftarrow$  2
5:   end if
6:   dataMahasiswa  $\leftarrow$  DATAMAHASISWA(idJurusan)
7:   return dataMahasiswa
8: end procedure

```

---

## 2. Fungsi data mahasiswa

**Algorithm 3** Data Mahasiswa

---

```

1: procedure DATAMAHASISWA(idJurusan)
2:   query  $\leftarrow$  SELECT * FROM mahasiswa INNER JOIN nilai ON mahasiswa.id_mahasiswa =
     nilai.id_mahasiswa WHERE id_jurusan = idJurusan
3:   return query
4: end procedure

```

---

**4.3.3 Siswa Controller**

*Siswa controller* berfungsi untuk memproses data yang diisi oleh siswa pada gambar 4.2. Pada siswa controller terdapat atribut berupa *array* yang berisikan kode mata pelajaran dan id mata pelajaran, berikut merupakan *pseudocode* fungsi yang terdapat pada siswa controller :

## 1. Fungsi index

Fungsi ini digunakan untuk menerima nilai yang diisi oleh siswa, mendapatkan data mahasiswa sesuai jurusan siswa, memilih *cluster* mahasiswa yang paling mirip dengan data siswa, memproses data untuk dihitung kesamaan dan prediksi, dan menampilkan pada halaman 4.4.

**Algorithm 4** Index

---

```

1: procedure INDEX(request)
2:   data  $\leftarrow$  request.INPUT()
3:   DATASISWA(data)
4:   mahasiswa  $\leftarrow$  MAHASISWACONTROLLER()
5:   mhs  $\leftarrow$  mahasiswa.INDEX(siswa["btn"]).TOARRAY()
6:   kmeans  $\leftarrow$  KMEANSCONTROLLER(k, mhs) ▷ k adalah jumlah kelompok yang ingin
     dibentuk
7:   cluster  $\leftarrow$  kmeans.HITUNGJARAKSISWA(siswa)
8:   mhs  $\leftarrow$  kmeans.GETCLUSTER(cluster)
9:   userBasedModel  $\leftarrow$  USERBASEDMODELCONTROLLER(mhs, siswa)
10:  result  $\leftarrow$  userBasedModel.GETRESULT()
11:  return view('/result', ['result'  $\leftarrow$  result])
12: end procedure

```

---

## 2. Fungsi data siswa

Fungsi ini digunakan untuk memproses data yang diinput oleh siswa agar format yang disimpan pada *array* sama seperti data mahasiswa.

**Algorithm 5** Data Siswa

---

```

1: procedure DATASISWA(data)
2:    $i \leftarrow 1$ 
3:   result  $\leftarrow$  ARRAY()
4:   result['nilai']  $\leftarrow$  ARRAY()
5:   for each key  $\Rightarrow$  value  $\in$  data do
6:     if key == '_token' then
7:       result[key]  $\leftarrow$  value
8:     else
9:       if  $i == 1$  then
10:         $k \leftarrow SUBSTR(key, 0, 3)$ 
11:        temp  $\leftarrow$  ARRAY()
12:        ARRAY_PUSH(temp, ((int)value/20)-1)           ▷ Conver kedalam GPA
13:         $i \leftarrow i + 1$ 
14:      else
15:        ARRAY_PUSH(temp, ((int)value/20)-1)           ▷ Conver kedalam GPA
16:         $i \leftarrow i + 1$ 
17:      if  $i == 5$  then
18:        avg  $\leftarrow$  ARRAY_SUM(temp)/COUNT(temp)
19:        ARRAY_PUSH(temp, avg)
20:        temp  $\leftarrow$  REPLACEKEY(temp, 5, 'id_mata_pelajaran')
21:        ARRAY_PUSH(result['nilai'], temp)
22:         $i \leftarrow 1$ 
23:      end if
24:    end if
25:  end if
26: end for
27: if !EMPTY(data['btnIPA']) then
28:   result['btn']  $\leftarrow$  'IPA'
29: else if !EMPTY(data['btnIPS']) then
30:   result['btn']  $\leftarrow$  'IPS'
31: end if
32: return result
33: end procedure

```

---

## 3. Fungsi replace key

**Algorithm 6** Replace Key

---

```

1: procedure REPLACEKEY(temp, oldKey, newKey)
2:   temp[newKey]  $\leftarrow$  temp[oldKey]
3:   UNSET(temp[oldKey])
4:   return temp
5: end procedure

```

---

**4.3.4 K-Means Controller**

*K-Means controller* berfungsi untuk membuat *cluster* pada data mahasiswa, agar pada saat menghitung kesamaan atau kemiripan menghilangkan mahasiswa yang memiliki nilai kemiripan negatif. Pada k-means controller terdapat beberapa atribut yang digunakan, yaitu :

1. *currCentroid*, untuk menyimpan *centroid* saat ini.

2. *preCentroid*, untuk menyimpan *centroid* sebelumnya.
3. *k*, jumlah *cluster* yang akan dibuat.
4. *cluster*, *array* untuk menyimpan anggota tiap *cluster*.
5. *mahasiswa*, untuk menyimpan data mahasiswa yang akan dibuat *cluster*.
6. *J0*, inisialisasi jarak total dari objek ke *centroid*-nya, berisi nilai 100 pada saat awal.
7. *J1*, inisialisasi jarak total dari objek ke *centroid*-nya, untuk *centroid* saat ini.

Berikut merupakan *pseudocode* fungsi yang terdapat pada *k-means contrller* :

1. Fungsi *contract*

---

**Algorithm 7** Contract KMeans
 

---

```

1: procedure CONTRACT(k, dataMahasiswa)
2:   k ← k
3:   mahasiswa ← dataMahasiswa
4:   INISIALISASICLUSTER()
5:   currCentroid ← ARRAY() ▷ centroid saat ini
6:   J0 ← 100 ▷ J0 = inisialisasi jarak total dari objek ke centroid-nya
7:   PILIHCENTROID()
8:   HITUNGJARAKMHS()
9:   status ← TRUE
10:  idx ← 0
11:  while status do
12:    HITUNGCENTROIDBARU()
13:    idx ← idx + 1
14:    status ← CEKBATAS()
15:    HITUNGJARAKMHS()
16:  end while
17: end procedure

```

---

2. Fungsi inisialisasi *cluster*  
 Inisialsi untuk atribut *cluster* ke 1 sampai *k*.

---

**Algorithm 8** Inisialisasi Cluster
 

---

```

1: procedure INISIALISASICLUSTER
2:   cluster ← ARRAY()
3:   for i = 1 into k do
4:     cluster[i] ← ARRAY()
5:   end for
6: end procedure

```

---

3. Fungsi pilih *centroid*  
 Fungsi untuk memilih *centroid* awal pada saat *k-means controller* di jalankan.

**Algorithm 9** Pilih Centroid

---

```

1: procedure PILIHCENTROID
2:    $i \leftarrow 0$ 
3:   while  $i < k$  do
4:      $key \leftarrow RAND(0, 1739)$  ▷ Random sebanyak jumlah mahasiswa
5:     if check key in mahasiswa == TRUE then
6:       if check key in mahasiswa == FALSE then
7:         ARRAY_PUSH(currCentroid, mahasiswa[key])
8:          $i \leftarrow i + 1$ 
9:       end if
10:    end if
11:  end while
12: end procedure

```

---

## 4. Fungsi hitung jarak mahasiswa

Menhitung jarak mahasiswa dengan *centroid* dan menambahkan mahasiswa sesuai dengan jarak terpendek dengan *cluster*.

**Algorithm 10** Hitung Jarak Mahasiswa

---

```

1: procedure HITUNGJARAKMHS
2:    $J1 \leftarrow 0$ 
3:   for each  $valueMhs \in mahasiswa$  do
4:      $temoCluster \leftarrow ARRAY()$  ▷ Penampung cluster sementara
5:      $nilaiMhs \leftarrow valueMhs['nilai']$ 
6:     for each  $valueNilaiMhs \in nilaiMhs$  do
7:        $arrayJarak \leftarrow ARRAY()$ 
8:       for each  $valueNilaiCen \in currCentroid$  do
9:         if  $(valueNilaiMhs['id_mata_pelajaran'] == 1 \text{ AND } valueNilaiCen['id_mata_pelajaran'] ==$ 
10:           $1) \text{ OR } (valueNilaiMhs['id_mata_pelajaran'] == 3 \text{ AND } valueNilaiCen['id_mata_pelajaran'] ==$ 
11:           $3) \text{ then}$ 
12:              $jarak \leftarrow EUCLIDIANCEDISTANCE(valueNilaiMhs, valueNilaiCen)$ 
13:           else if  $valueNilaiMhs['id_mata_pelajaran'] < valueNilaiCen['id_mata_pelajaran']$ 
14:             then
15:               break
16:             end if
17:           end for
18:            $ARRAY\_PUSH(arrJarak, jarak)$ 
19:         end for
20:         if  $tempCluster \text{ is empty}$  then
21:            $ARRAY\_PUSH(tempCluster, arrJarak)$ 
22:         else
23:           for  $i = 1$  into  $k$  do
24:              $tempCluster[0][i] \leftarrow tempCluster[0][i] + arrJarak[i]$ 
25:              $tempCluster[0][i] \leftarrow SQRT(tempCluster[0][i])$ 
26:           end for
27:         end if
28:          $c \leftarrow currentmahasiswacuster$ 
29:          $J1 \leftarrow J1 + tempCluster[0][c]$ 
30:          $ARRAY\_PUSH(tempCluster[0], c, valueMhs['id\_mahasiswa'])$ 
31:          $tempCluster[0]['id\_mahasiswa'] \leftarrow tempCluster[0][k + 1]$ 
32:          $UNSET(tempCluster[0][k + 1])$ 
33:          $ARRAY\_PUSH(cluster[c], valueMhs)$ 
34:       end for
35:     end for
36: end procedure

```

---

## 5. Fungsi hitung jarak siswa

Menghitung jarak siswa dengan *centroid* dan mengembalikan nomor *cluster* yang paling mirip dengan siswa.

**Algorithm 11** Hitung Jarak Siswa

---

```

1: procedure HITUNGJARAKSISWA(siswa)
2:   nilaiSiswa  $\leftarrow$  siswa['nilai']
3:   tempCluster  $\leftarrow$  ARRAY()
4:   for each valueNilaiSiswa  $\in$  nilaiSiswa do
5:     arrJarak  $\leftarrow$  ARRAY()
6:     for each valueCen  $\in$  currCentroid do
7:       jarak  $\leftarrow$  0
8:       nilaiCen  $\leftarrow$  valueCen['nilai']
9:       for each valueNilaiCen  $\in$  nilaiCen do
10:        if (valueNilaiMhs['id_mata_pelajaran'] == 1 AND valueNilaiCen['id_mata_pelajaran'] ==
1) OR (valueNilaiMhs['id_mata_pelajaran'] == 3 AND valueNilaiCen['id_mata_pelajaran'] ==
3) then
11:          jarak  $\leftarrow$  EUCLIDIANCEDISTANCE(valueNilaiSiswa, valueNilaiCen)
12:          else if valueNilaiMhs['id_mata_pelajaran'] < valueNilaiCen['id_mata_pelajaran']
then
13:            break
14:          end if
15:        end for
16:        ARRAY_PUSH(arrJarak, jarak)
17:      end for
18:      if tempCluster is empty then
19:        ARRAY_PUSH(tempCluster, arrJarak)
20:      else
21:        for i = 1 into k do
22:          tempCluster[0][i]  $\leftarrow$  tempCluster[0][i] + arrJarak[i]
23:          tempCluster[0][i]  $\leftarrow$  SQRT(tempCluster[0][i])
24:        end for
25:      end if
26:    end for
27:    res  $\leftarrow$  currentsiswacuster
28:    return res
29: end procedure

```

---

6. Fungsi *euclidiance distance*

Fungsi untuk menghitung jarak mahasiswa atau siswa dengan *centroid* menggunakan metode *euclidiance distance*.

**Algorithm 12** Euclidiance Distance

---

```

1: procedure EUCLIDIANCEDISTANCE(mhs, centroid)
2:   result  $\leftarrow$  0
3:   for i = 1 into 4 do
4:     result  $\leftarrow$  result + POW(mhs[i] - centroid[i], 2)
5:   end for
6:   result  $\leftarrow$  result + POW(mhs['AVG'] - centroid['AVG'], 2)
7:   return result
8: end procedure

```

---

7. Fungsi hitung *centroid* baru

Fungsi untuk menghitung centroid baru dari *cluster* yang sudah dibuat.

**Algorithm 13** Hitung Centroid Baru

---

```

1: procedure HITUNGCENTROIDBARU
2:    $prevCentroid \leftarrow currCentroid$ 
3:   RESETCENTROID()
4:   for each  $keyCen \Rightarrow valueCen \in currCentroid$  do
5:      $nilaiCen \leftarrow valueCen['nilai']$ 
6:     for each  $keyNilaiCen \Rightarrow valueNilaiCen \in nilaiCen$  do
7:        $anggota \leftarrow cluster[keyCen]$ 
8:       if numbers of anggota  $\neq 0$  then
9:         for each  $keyAnggota \Rightarrow valueAnggota \in anggota$  do
10:           $nilaiAnggota = valueAnggota['nilai']$ 
11:          for each  $keyNilaiAnggota \Rightarrow valueNilaiAnggota \in nilaiAnggota$  do
12:            if  $(valueNilaiMhs['id_mata_pelajaran'] == 1 \text{ AND } valueNilaiCen['id_mata_pelajaran'] == 1) \text{ OR } (valueNilaiMhs['id_mata_pelajaran'] == 3 \text{ AND } valueNilaiCen['id_mata_pelajaran'] == 3)$  then
13:              for  $i = 1$  into 4 do
14:                 $nilaiLama \leftarrow currCentroid[keyCen]['nilai'][keyNilaiCen][i]$ 
15:                 $nilaiBaru \leftarrow anggota[keyAnggota]['nilai'][keyNilaiAnggota][i]$ 
16:                UPDATENILAI(keyCen, keyNilaiCen, nilaiLama, nilaiBaru, i)
17:              end for
18:               $nilaiLama \leftarrow currCentroid[keyCen]['nilai'][keyNilaiCen]['AVG']$ 
19:               $nilaiBaru \leftarrow anggota[keyAnggota]['nilai'][keyNilaiAnggota]['AVG']$ 
20:              UPDATENILAI(keyCen, keyNilaiCen, nilaiLama, nilaiBaru, 'AVG')
21:            end if
22:          end for
23:        end for
24:      else
25:        RANDOMNILAIBARU(keyCen, KeyNilaiCen)
26:      end if
27:    end for
28:  end for
29:  HITUNGRATA2()
30: end procedure

```

---

8. Fungsi reset *centroid***Algorithm 14** Reset Centroid

---

```

1: procedure RESETCENTROID
2:   for each  $keyCen \Rightarrow valueCen \in currCentroid$  do
3:      $nilaiCen \leftarrow valueCen['nilai']$ 
4:     for each  $keyNilai \in nilaiCen$  do
5:       for  $i = 1$  into 4 do
6:          $currCentroid[keyCen]['nilai'][keyNilaiCen][i] \leftarrow 0$ 
7:       end for
8:        $currCentroid[keyCen]['nilai'][keyNilaiCen]['AVG'] \leftarrow 0$ 
9:     end for
10:  end for
11: end procedure

```

---



9. Fungsi *update* nilai  
Fungsi untuk *update* nilai *centroid* baru.

---

**Algorithm 15** Update Nilai
 

---

```

1: procedure UPDATENILAI(keyCen, keyNilaiCen, nilaiLama, nilaiBaru, i)
2:   nilai  $\leftarrow$  nilaiLama + nilaiBaru
3:   currCentroid[keyCen]['nilai'][keyNilaiCen][i]  $\leftarrow$  nilai
4: end procedure

```

---

10. Fungsi hitung rata-rata  
Fungsi untuk menghitung rata-rata nilai yang sudah di-*update*.

---

**Algorithm 16** Hitung Rata2
 

---

```

1: procedure HITUNGRATA2
2:   for each keyCen => valueCen  $\in$  currCentroid do
3:     nilaiCen  $\leftarrow$  valueCen['nilai']
4:     anggota  $\leftarrow$  cluster[keyCen]
5:     count  $\leftarrow$  numbersofanggota
6:     if counter  $\neq$  0 then
7:       for each keyNilaiCen  $\in$  nilaiCen do
8:         for i = 1 into 4 do
9:           currCentroid[keyCen]['nilai'][keyNilaiCen][i]  $\leftarrow$ 
             currCentroid[keyCen]['nilai'][keyNilaiCen][i]/count
10:        end for currCentroid[keyCen]['nilai'][keyNilaiCen]['AVG']  $\leftarrow$ 
             currCentroid[keyCen]['nilai'][keyNilaiCen]['AVG']/count
11:      end for
12:    end if
13:  end for
14: end procedure

```

---

11. Fungsi *random* nilai baru

Fungsi untuk *random* nilai baru, jika tidak terdapat anggota pada suatu *cluster*.

---

**Algorithm 17** Random NilaiBaru
 

---

```

1: procedure RANDOMNILAIBARU(keyCen, keyNilaiCen)
2:   for i = 1 into 4 do
3:     currCentroid[keyCen]['nilai'][keyNilaiCen][i]  $\leftarrow$  RAND(1, 3) + RAND(1, 10)/10
4:   end for
5:   currCentroid[keyCen]['nilai'][keyNilaiCen]['AVG']  $\leftarrow$  RAND(1, 3) + RAND(1, 10)/10
6: end procedure

```

---

12. Fungsi cek batas

Fungsi untuk mengecek apakah batas  $< 0.01$ , jika iya berarti anggota pada suatu *cluster* sudah konvergen.

**Algorithm 18** Cek Batas

---

```

1: procedure CEKBATAS
2:    $batas \leftarrow ABS(J0 - J1)$ 
3:   if batas < 0.1 then
4:     return FALSE
5:   end if
6:   return TRUE
7: end procedure

```

---

13. Fungsi *get cluster*

Fungsi untuk mendapatkan *cluster* yang sesuai dengan *cluster* yang sudah dibentuk.

**Algorithm 19** Get Cluster

---

```

1: procedure GETCLUSTER(idx)
2:   return cluster[idx]
3: end procedure

```

---

**4.3.5 User Based Model Controller**

*User Based Model Controller* berfungsi untuk mengidentifikasi pengguna yang akan diberikan rekomendasi dengan pengguna lain. Terdapat tiga atribut pada *user base model controller* yaitu :

1. *pearsonCorrelation*, untuk menampung *PearsonController*.
2. *prediction*, untuk menampung *PredictionController*.
3. *result*, untuk menampung hasil prediksi.

Berikut merupakan *pseudocode* fungsi yang terdapat pada *user based model controller* :

1. Fungsi *contract*

Terdapat tiga parameter, yaitu : *mahasiswa*, *siswa*, dan *mode* dengan nilai awal 0. Parameter *mahasiswa* merupakan *array* yang berisikan data mahasiswa, *siswa* merupakan *array* yang berisikan nilai siswa, dan *mode* merupakan mode yang digunakan pada saat *user based model controller* dijalankan, jika mode bernilai 0 berarti dalam mode penggunaan siswa dan 1 jika dalam pengujian.

**Algorithm 20** Contract

---

```

1: procedure CONTRACT(mahasiswa, siswa, mode = 0)
2:    $prediction \leftarrow PREDICTIONCONTROLLER()$ 
3:   if mode == 0 then
4:      $pearsonCorrelation \leftarrow PEARSONCORRELATIONCONTROLLER()$ 
5:      $pearson \leftarrow CALCULATESIMILARITY(mahasiswa, siswa)$ 
6:      $result \leftarrow CALCULATEPREDICT(pearson)$ 
7:   else if mode == 1 then
8:      $pearsonCorrelation \leftarrow PEARSONCORRELATIONPENGUJIANCONTROLLER()$ 
9:   end if
10: end procedure

```

---

2. Fungsi *calculate similarity***Algorithm 21** Calculate Similarity

---

```

1: procedure CALCULATESIMILARITY(mahasiswa, siswa)
2:   return pearsonCorrelation  $\leftarrow$  CALCULATEPEARSON(mahasiswa, siswa)
3: end procedure

```

---

3. Fungsi *calculate prediction***Algorithm 22** Calculate Predict

---

```

1: procedure CALCULATEPREDICT(pearson)
2:   return prediction  $\leftarrow$  CALCULATEPREDICT(pearson)
3: end procedure

```

---

4. Fungsi *get result***Algorithm 23** User-based Collaborative Filtering

---

```

1: procedure GETRESULT
2:   return result
3: end procedure

```

---

**4.3.6 Pearson Correlation Controller**

*Pearson correlation controller* berfungsi untuk menghitung kesamaan pengguna dengan pengguna lain, berikut merupakan *pseudocode* fungsi yang terdapat pada *pearson correlation controller* :

1. Fungsi *calcaulate pearson***Algorithm 24** Calculate Pearson

---

```

1: procedure CALCULATEPEARSON(mahasiswa, siswa)
2:   res  $\leftarrow$  ARRAY()
3:   for each mhs  $\in$  mahasiswa do
4:     covariance  $\leftarrow$  CALCULATECOVARIANCE(mhs, siswa)
5:     sd  $\leftarrow$  CALCULATESTANDARDEVIATION(mhs, siswa)
6:     sdMhs  $\leftarrow$  sd[0]
7:     sdSiswa  $\leftarrow$  sd[1]
8:     idProdi  $\leftarrow$  mhs['id_program_studi']
9:     IPK  $\leftarrow$  mhs['IPK']
10:    sim  $\leftarrow$  COVARIANCE / (sdMhs * sdSiswa)
11:    if sim > 0 then
12:      res[mhs['id_mahasiswa']]  $\leftarrow$  ARRAY()
13:      ARRAY_PUSH(res[mhs['id_mahasiswa']], sim, idProdi, IPK)
14:    end if
15:  end for
16:  return res
17: end procedure

```

---

2. Fungsi *calculate covaraiance***Algorithm 25** Calculate Covariance

---

```

1: procedure CALCULATECOVARIANCE(mhs, siswa)
2:   res  $\leftarrow$  0
3:   nilaiMhs  $\leftarrow$  mhs['nilai']
4:   nilaiSiswa  $\leftarrow$  siswa['nilai']
5:   for each nSiswa  $\in$  nilaiSiswa do
6:     idMP  $\leftarrow$  nSiswa['id_mata_pelajaran']
7:     for each nMhs  $\in$  nilaiMhs do
8:       if idMP == nMhs['id_mata_pelajaran'] then
9:         for i = 1 to 4 do
10:          res  $\leftarrow$  res + (nMhs[i] - nMhs['AVG']) * (nSiswa[i] - nSiswa['AVG'])
11:        end for
12:      else if idMP < nMhs['id_mata_pelajaran'] then
13:        break
14:      end if
15:    end for
16:  end for
17:  return res
18: end procedure

```

---

3. Fungsi *calculate standar deviation***Algorithm 26** Calculate Standar Deviation

---

```

1: procedure CALCULATESTANDARDEVIATION(mhs, siswa)
2:   res  $\leftarrow$  0
3:   sdMhs  $\leftarrow$  0
4:   sdSiswa  $\leftarrow$  0
5:   nilaiMhs  $\leftarrow$  mhs['nilai']
6:   nilaiSiswa  $\leftarrow$  siswa['nilai']
7:   for each nSiswa  $\in$  nilaiSiswa do
8:     idMP  $\leftarrow$  nSiswa['id_mata_pelajaran']
9:     for each nMhs  $\in$  nilaiMhs do
10:      if idMP == nMhs['id_mata_pelajaran'] then
11:        for i = 1 to 4 do
12:          sdMhs  $\leftarrow$  sdMhs + POW(nMhs[i] - nMhs['AVG'], 2)
13:          sdSiswa  $\leftarrow$  sdSiswa + POW(nSiswa[i] - nSiswa['AVG'], 2)
14:        end for
15:      else if idMP < nMhs['id_mata_pelajaran'] then
16:        break
17:      end if
18:    end for
19:  end for
20:  ARRAY_PUSH(res, SQRT(sdMhs), SQRT(sdSiswa))
21:  return res
22: end procedure

```

---

### 4.3.7 Prediction Controller

*Prediction controller* berfungsi untuk menghitung prediksi dari hasil perhitungan kesamaan atau similaritas menggunakan *pearson correlation*. Terdapat dua atribut dalam *prediction controller*, yaitu :

1. *programStudi*, untuk menyimpan *ProgramStudiController*.
2. *fakultas*, untuk menyimpan *FakultasController*.

Berikut merupakan *pseudocode* fungsi yang terdapat pada *prediction controller* :

1. Fungsi *construct*

---

#### Algorithm 27 Construct

---

```

1: procedure CONSTRUCT
2:   programStudi  $\leftarrow$  PROGRAMSTUDICONTROLLER()
3:   fakultas  $\leftarrow$  FAKULTASCONTROLLER()
4: end procedure

```

---

2. Fungsi *calculate predict*

---

#### Algorithm 28 Calculate Predict

---

```

1: procedure CALCULATEPREDICT(pearson)
2:   res  $\leftarrow$  ARRAY()
3:   a  $\leftarrow$  0
4:   b  $\leftarrow$  0
5:   for each id_mhs  $\Rightarrow$  value  $\in$  pearson do
6:     a  $\leftarrow$  a + value[0] * value[2]
7:     b  $\leftarrow$  b + value[0]
8:     next  $\leftarrow$  NEXT(pearson)
9:     if next  $\neq$  NULL then
10:      if value[1]  $\neq$  next[1] then
11:        res  $\leftarrow$  INSERTDATA(res, a, b, value[1])
12:        a  $\leftarrow$  0
13:        b  $\leftarrow$  0
14:      end if
15:      else if next  $\neq$  NULL then
16:        res  $\leftarrow$  INSERTDATA(res, a, b, value[1])
17:      end if
18:    end for
19:    score  $\leftarrow$  ARRAY_COLUMN(res, 0)
20:    ARRAY_MULTISORT(score, SORT_DESC, res)
21:    return res
22: end procedure

```

---

3. Fungsi *insert data*

**Algorithm 29** Insert Data

---

```

1: procedure INSERTDATA(res, a, b, idProdi)
2:   pred  $\leftarrow a/b$ 
3:   namaFakultas  $\leftarrow fakultas.GETNAMAFAKULTAS(idProdi)$ 
4:   namaProdi  $\leftarrow programStudi.GETNAMAPROGRAMSTUDI(idProdi)$ 
5:   res[idProdi]  $\leftarrow ARRAY()$ 
6:   ARRAY_PUSH(res[idProdi], pred, namaFakultas, namaProdi)
7:   return res
8: end procedure

```

---

**4.3.8 Fakultas Controller**

*Fakultas controller* berfungsi untuk mendapatkan data fakultas dari model *FakultasModel*, berikut merupakan *pseudocode* fungsi yang terdapat pada *fakultas controller* :

1. Fungsi *get* nama fakultas

**Algorithm 30** Get Nama Fakultas

---

```

1: procedure GETNAMAFAKULTAS(idProgramStudi)
2:   idFakultas  $\leftarrow FLOOR(idProgramStudi)$ 
3:   namaFakultas  $\leftarrow \text{SELECT nama\_fakultas FROM fakultas WHERE idFakultas} = \text{idProgramStudi}$ 
4:   return namaFakultas
5: end procedure

```

---

**4.3.9 Program Studi Controller**

*Program studi controller* berfungsi untuk mendapatkan data program studi dari model *ProgramStudiModel*, berikut merupakan *pseudocode* fungsi yang terdapat pada *program studi controller* :

1. Fungsi *get* nama program studi

**Algorithm 31** getNamaProgramStudi

---

```

1: procedure GETNAMAPROGRAMSTUDI(idProgramStudi)
2:   namaProdi  $\leftarrow \text{SELECT nama\_program\_studi FROM program\_studi WHERE id\_program\_studi} = \text{idProgramStudi}$ 
3:   return namaProdi
4: end procedure

```

---

**4.3.10 Pengujian Controller**

*Pengujian controller* digunakan untuk melakukan pengujian terhadap sistem rekomendasi program studi Universitas Katolik Parahyangan. Pengujian dilakukan dengan menghitung akurasi dan waktu eksekusi program. Terdapat beberapa atribut pada *pengujian controller*, yaitu :

1. *train*, untuk data *train* dari hasil pembagian data.
2. *test*, untuk data *test* dari hasil pembagian data.
3. *userBasedModel*, untuk menyimpan *UserBaseModelController*

4. *error1*, untuk menyimpan selisih nilai yang akan digunakan untuk *mae*.
5. *error2*, untuk menyimpan selisih nilai yang akan digunakan untuk *rmse*.
6. *accuracy*, untuk menyimpan *AccuracyController*.
7. *metode*, untuk menyimpan metode pengujian yang digunakan.

Berikut merupakan *pseudocode* fungsi yang terdapat pada *pengujian controller* :

1. Fungsi *construct*

---

**Algorithm 32** Construct

---

```

1: procedure CONSTRUCT(request)
2:   btn  $\leftarrow$  request.INPUT()
3:   idJurusan  $\leftarrow$  SUBSTR(btn['btn'], 0, 3)
4:   metode = SUBSTR(btn['btn'], 4, STRLEN(btn['btn']))
5:   mhs  $\leftarrow$  MAHASISWACONTROLLER()
6:   data  $\leftarrow$  mhs.INDEX(idJurusan).TOARRAY()
7:   arrLabel  $\leftarrow$  ARRAY() ▷ Penampung label data
8:   for each m  $\in$  data do
9:     ARRAY_PUSH(arrLabel, m['id_program_studi'])
10:  end for
11:  dataSet  $\leftarrow$  ARRAYDATASET(data, arrLabel)
12:  dataset  $\leftarrow$  RANDOMSPLIT(dataset, n) ▷ n = numbers of size (0.1 - 0.9)
13:  train  $\leftarrow$  dataset.GETTRAINSAMPLES
14:  test  $\leftarrow$  dataset.GETTESTSAMPLES
15:  accuracy  $\leftarrow$  ACCURACYCONTROLLER()
16:  userBasedModel  $\leftarrow$  USERBASEDMODELCONTROLLER(NULL, NULL, 1)
17: end procedure

```

---

2. Fungsi *index*

Fungsi untuk menentukan metode yang digunakan. Terdapat dua metode, yaitu : pengujian *basic* atau dasar dan pengujian menggunakan metode *kmeans*.

---

**Algorithm 33** Index

---

```

1: procedure INDEX
2:   if metode == 'Basic' then
3:     return PENGUJIANBASIC()
4:   else
5:     return PENGUJIANKMEANS(bts, n)
6:   end if
7: end procedure

```

---

3. Fungsi pengujian *basic*

**Algorithm 34** Pengujian Basic

---

```

1: procedure PENGUJIANBASIC
2:   result  $\leftarrow$  ARRAY()
3:   error1  $\leftarrow$  ARRAY()
4:   error2  $\leftarrow$  ARRAY()
5:   for each t  $\in$  test do
6:     start  $\leftarrow$  counttime
7:     if t['NPM'] NOT IN result then
8:       temp  $\leftarrow$  ARRAY()
9:       pearson  $\leftarrow$  userBasedModel.CALCULATESIMILARITY(train, t)
10:      predict  $\leftarrow$  userBasedModel.CALCULATEPREDICT(pearson)
11:      if predict  $\neq$  NULL then
12:        diff1  $\leftarrow$  ABS(t['IPK'] - predict[0][0])
13:        ARRAY_PUSH(error1, diff1)
14:        diff2  $\leftarrow$  POW(t['IPK'] - predict[0][0], 2)
15:        ARRAY_PUSH(error2, diff2)
16:        ARRAY_PUSH(temp, t['NPM'], predict[0][2], t['IPK'], predict[0][0], diff1, diff2)
17:        ARRAY_PUSH(result, temp)
18:      end if
19:    end if
20:  end for
21:  mae  $\leftarrow$  accuracy.CALCULATEMAE(error1)
22:  rmse  $\leftarrow$  accuracy.CALCULATERMSE(error2)
23:  end  $\leftarrow$  counttime
24:  times  $\leftarrow$  end - start
25:  return view('/pengujian', ['status'=>TRUE, 'result'=>result, 'mae'=>mae, 'rmse'=>rmse,
    'times'=>times, 'metode'=>metode])
26: end procedure

```

---

4. Fungsi pengujian *kmeans*

Pengujian menggunakan metode *kmeans* terdapat dua parameter, yaitu : *bts* dan *n*. Parameter *bts* berfungsi untuk batas k yang akan dicoba dan *n* berfungsi untuk mengulang sebanyak n kali proses *clustering*.



**Algorithm 35** Pengujian Kmeans

---

```

1: procedure PENGUJIANKMEANS(bts, n)
2:   result  $\leftarrow$  ARRAY()
3:   resultPred  $\leftarrow$  ARRAY()
4:   for k = 2 into bts do
5:     start  $\leftarrow$  counttime
6:     tempMae  $\leftarrow$  0
7:     tempRmse  $\leftarrow$  0
8:     for i = 1 into n do
9:       kmeans  $\leftarrow$  KMEANSCOONTROLLER(k, train)
10:      error1  $\leftarrow$  ARRAY()
11:      error2  $\leftarrow$  ARRAY()
12:      for each t  $\in$  test do
13:        if t['NPM'] NOT IN resultPred then
14:          temp  $\leftarrow$  ARRAY()
15:          cluster  $\leftarrow$  kmeans.HITUNGJARAKSISWA(t)
16:          dataTrain  $\leftarrow$  kmeans.GETCLUSTER(cluster)
17:          pearson  $\leftarrow$  userBasedModel.CALCULATESIMILARITY(dataTrain, t)
18:          predict  $\leftarrow$  userBasedModel.CALCULATEPREDICT(pearson)
19:          if predict  $\neq$  NULL then
20:            diff1  $\leftarrow$  ABS(t['IPK'] - predict[0][0])
21:            ARRAY_PUSH(error1, diff1)
22:            diff2  $\leftarrow$  POW(t['IPK'] - predict[0][0], 2)
23:            ARRAY_PUSH(error2, diff2)
24:            ARRAY_PUSH(temp, t['NPM'], predict[0][2], t['IPK'], predict[0][0], diff1,
diff2)
25:            ARRAY_PUSH(resultPred, temp)
26:          end if
27:        end if
28:      end for
29:      mae  $\leftarrow$  accuracy.CALCULATEMAE(error1)
30:      rmse  $\leftarrow$  accuracy.CALCULATERMSE(error2)
31:      tempMae  $\leftarrow$  tempMae + mae
32:      tempRmse  $\leftarrow$  tempRmse + rmse
33:    end for
34:    end  $\leftarrow$  counttime
35:    times  $\leftarrow$  end - start
36:    result[k]  $\leftarrow$  ARRAY()
37:    ARRAY_PUSH(result[k], tempMae/n, tempRmse/n, times)
38:  end for
39:  minMae  $\leftarrow$  MIN(ARRAY_COLUMN(result, 0))
40:  minRmse  $\leftarrow$  MIN(ARRAY_COLUMN(result, 1))
41:  minTime  $\leftarrow$  MIN(ARRAY_COLUMN(result, 2))
42:  return view('/penguajian', ['status'  $\leftarrow$  TRUE, 'resultPred'  $\leftarrow$  resultPred, 'metode'  $\leftarrow$ 
metode, 'result'  $\leftarrow$  result, 'minMae'  $\leftarrow$  minMae, 'minRmse'  $\Rightarrow$  minRmse, 'minTime'  $\leftarrow$ 
minTime])
43: end procedure

```

---

#### 4.3.11 Pearson Correlation Pengujian Controller

*Pearson Correlation pengujina controller* digunakan pada saat pengujian, yang membedakan dengan *pearson correlation* hanya pada fungsi *calculate pearson*, dimana hanya menghitung prediksi sesuai dengan program studi mahasiswa yang dicek. Berikut merupakan *pseudocode* fungsi yang terdapat pada *pearson correlation pengujian controller* :

1. Fungsi *calculaye pearson*

---

**Algorithm 36** Calculate Pearson Pengujian

---

```

1: procedure CALCULATEPEARSON(mahasiswa, siswa)
2:   res  $\leftarrow$  ARRAY()
3:   for each mhs  $\in$  mahasiswa do
4:     if mhs[id_program_studi] == siswa[id_program_studi] then
5:       covariance  $\leftarrow$  CALCULATECOVARIANCE(mhs, siswa)
6:       sd  $\leftarrow$  CALCULATESTANDARDEVIATION(mhs, siswa)
7:       sdMhs  $\leftarrow$  sd[0]
8:       sdSiswa  $\leftarrow$  sd[1]
9:       idProdi  $\leftarrow$  mhs[id_program_studi]
10:      IPK  $\leftarrow$  mhs[IPK]
11:      sim  $\leftarrow$  CONVARIANCE/(sdMhs * sdSiswa)
12:      if sim > 0 then
13:        res[mhs[id_mahasiswa]]  $\leftarrow$  ARRAY()
14:        ARRAY_PUSH(res[mhs[id_mahasiswa]], sim, idProdi, IPK)
15:      end if
16:    end if
17:  end for
18:  return res
19: end procedure

```

---

2. Fungsi *calculate covariance* sama seperti 25.
3. Fungsi *calculate standar deviation* sama seperti 26.

#### 4.3.12 Accuracy Controller

*Accuracy Controller* digunakan untuk menghitung akurasi dari sistem rekomendasi program studi Universitas Katolik Parahyangan, berikut merupakan *pseudocode* fungsi yang terdapat pada *accuracy controller* :

1. Fungsi *calculate mae*

---

**Algorithm 37** Calculate MAE

---

```

1: procedure CALCULATEMAE(arr)
2:   return ARRAY_SUM(arr)/COUNT(arr)
3: end procedure

```

---

2. Fungsi *calculate rmse*

---

**Algorithm 38** Calculate RMSE

---

```
1: procedure CALCULATEMSE(arr)
2:   return SQRT(ARRAY_SUM(arr)/COUNT(arr))
3: end procedure
```

---

[illegible]

Gambar 4.5: *Class Diagram* Sistem Rekomendasi

## BAB 5

### IMPLEMENTASI DAN PENGUJIAN

#### 5.1 Lingkungan Implementasi

Pada subbab ini akan dipaparkan perangkat keras dan perangkat lunak yang digunakan dalam membangun sistem rekomendasi program studi Universitas Katolik Parahyangan.

##### 5.1.1 Lingkungan Perangkat Keras

1. Processor : Intel Core i5-7200
2. Memory : 12 GB
3. Harddisk : 1 T
4. VGA : NVIDIA GeForce 940MX

#### 5.2 Lingkungan Perangkat Lunak

1. Web Server : Apache 2.4.41
2. Tools : XAMPP 3.2.4 dan Visual Studio Code 1.44
3. Bahasa Pemrograman : PHP 7.4.1
4. Database management system : MySQL
5. Operating System : Windows 10

#### 5.3 Implementasi Tabel Basis Data

Dibawah ini merupakan implementasi tabel basis data yang digunakan pada sistem rekomendasi program studi Universitas Katolik Parahyangan.

1. Tabel Jurusan SMA

Tabel jurusan SMA digunakan untuk menyimpan seluruh data jurusan SMA yang digunakan pada sistem rekomendasi.

Listing 5.1: Implementasi tabel jurusan SMA

```
1 CREATE TABLE 'jurusan_sma' (  
2     'id_jurusan' int(10) UNSIGNED NOT NULL,  
3     'nama_jurusan' varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL  
4 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## 2. Tabel Fakultas

Tabel fakultas digunakan untuk menyimpan seluruh fakultas yang ada di Universitas Katolik Parahyangan.

Listing 5.2: Implementasi tabel

```
1 CREATE TABLE 'fakultas' (
2     'id_fakultas' int(10) UNSIGNED NOT NULL,
3     'nama_fakultas' varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL
4 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## 3. Program Studi

Tabel program studi digunakan untuk menyimpan seluruh program studi yang ada di Universitas Katolik Parahyangan.

Listing 5.3: Implementasi tabel

```
1 CREATE TABLE 'program_studi' (
2     'id_program_studi' int(10) UNSIGNED NOT NULL,
3     'nama_program_studi' varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
4     'id_fakultas' int(10) UNSIGNED NOT NULL,
5     'id_jurusan' int(10) UNSIGNED NOT NULL
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## 4. Mahasiswa

Tabel mahasiswa digunakan untuk menyimpan nilai seluruh mahasiswa yang sudah lulus dari Universitas Katolik Parahyangan.

Listing 5.4: Implementasi tabel

```
1 CREATE TABLE 'mahasiswa' (
2     'id_mahasiswa' int(10) UNSIGNED NOT NULL,
3     'NPM' varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
4     'IPK' double(3,2) NOT NULL,
5     'id_jurusan' int(10) UNSIGNED NOT NULL,
6     'id_program_studi' int(10) UNSIGNED NOT NULL
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## 5. Mata Pelajaran

Tabel mata pelajaran digunakan untuk menyimpan mata pelajaran yang digunakan pada PMDK di Universitas Katolik Parahyangan.

Listing 5.5: Implementasi tabel

```
1 CREATE TABLE 'mata_pelajaran' (
2     'id_mata_pelajaran' int(10) UNSIGNED NOT NULL,
3     'nama_mata_pelajaran' varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL
4 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## 6. Nilai

Tabel nilai digunakan untuk menyimpan nilai mahasiswa pada kelas X dan XI untuk semester 1 dan 2 pada saat SMA.

Listing 5.6: Implementasi tabel

```
1 CREATE TABLE 'nilai' (
2     'id_nilai' int(10) UNSIGNED NOT NULL,
```

```
3      'id_mata_pelajaran' int(10) UNSIGNED NOT NULL,  
4      'id_mahasiswa' int(10) UNSIGNED NOT NULL,  
5      '101' double(5,2) NOT NULL,  
6      '102' double(5,2) NOT NULL,  
7      '111' double(5,2) NOT NULL,  
8      '112' double(5,2) NOT NULL,  
9      'AVG' double(5,2) NOT NULL  
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## 5.4 Impelemtasi Antar Muka

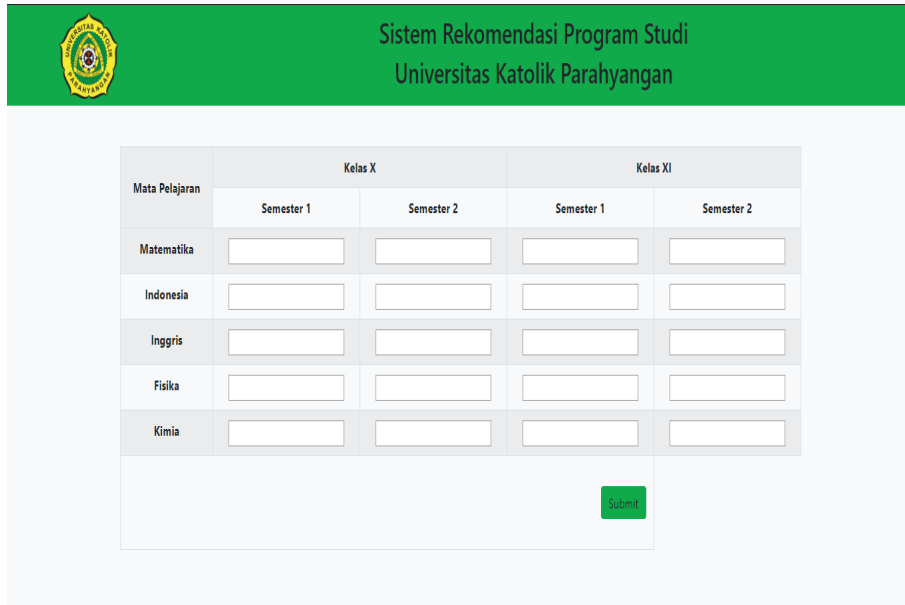
Pada subab ini akan ditampilkan antar muka yang digunakan pada sistem rekomendasi program studi Universitas Katolik Parahyangan.

1. Halaman index saat siswa/i mengakses sistem



Gambar 5.1: Halaman Index Sistem

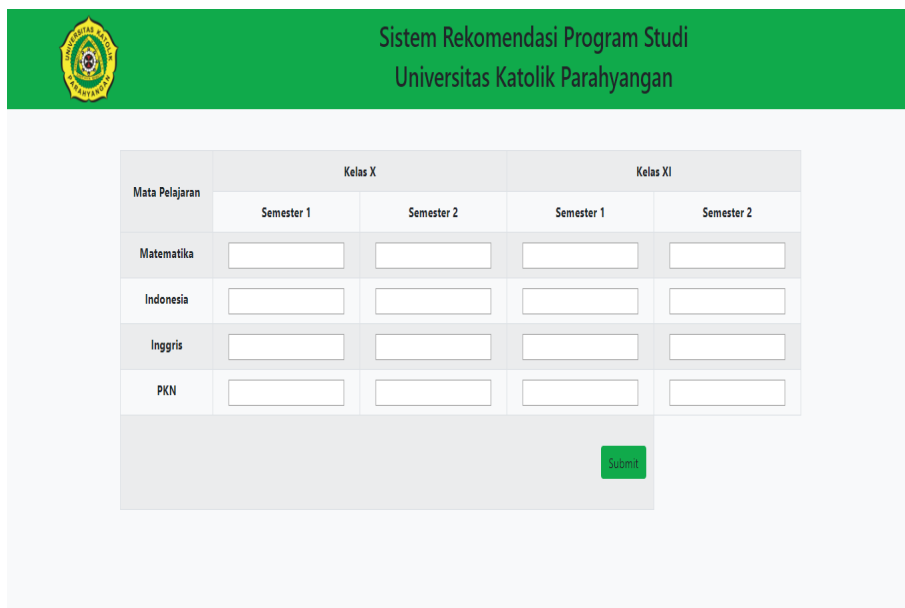
2. Halaman pengisian nilai siswa/i IPA



Mata Pelajaran	Kelas X		Kelas XI	
	Semester 1	Semester 2	Semester 1	Semester 2
Matematika	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Indonesia	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Inggris	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Fisika	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Kimia	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Gambar 5.2: Halaman Index Pengisian Nilai IPA

## 3. Halaman pengisian nilai siswa/i IPS



Mata Pelajaran	Kelas X		Kelas XI	
	Semester 1	Semester 2	Semester 1	Semester 2
Matematika	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Indonesia	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Inggris	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
PKN	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Gambar 5.3: Halaman Index Pengisian Nilai IPS

## 4. Halaman hasil rekomendasi





No	Fakultas	Program Studi	IPK
----	----------	---------------	-----

Gambar 5.4: Halaman Hasil Rekomendasi

## 5.5 Pengujian Fungsional

Pengujian ini dilakukan untuk menguji fitur-fitur yang ada pada sistem rekomendasi program studi Universitas Katolik Parahyangan agar dapat berjalan dengan baik.

### 5.5.1 Pengujian Fungsional Pemilihan Jurusan SMA

Pengujian ini dilakukan pada fitur pemilihan jurusan saat SMA oleh siswa/i yang menjadi target sistem.

No	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian	Status
1	Memilih jurusan saat SMA	Sistem mengarahkan kepada form sesuai jurusan SMA	Sistem mengarahkan kepada form sesuai jurusan SMA	Sesuai

Tabel 5.1: Tabel Pengujian Fungsional Pemilihan SMA

### 5.5.2 Pengujian Fungsional Pengisian Nilai

Pengujian ini dilakukan pada fitur pengisian nilai mata pelajaran sesuai dengan jurusan saat SMA pada kelas X dan XI.

No	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian	Status
1	Mengisi nilai sesuai nilai rapor	Memeriksa valid tidaknya data yang dimasukkan dan memeriksa <i>range</i> nilai	Memeriksa valid tidaknya data yang dimasukkan dan memeriksa <i>range</i> nilai	Sesuai
2	Klik tombol <i>submit</i>	Mengarahkan kepada halaman hasil rekomendasi	Mengarahkan kepada halaman hasil rekomendasi	Sesuai
3	Mengisi nilai yang tidak valid	Memberikan pesan data tidak valid	Memberikan pesan data tidak valid	Sesuai

Tabel 5.2: Tabel Pengujian Fungsional Pengisian Nilai

## 5.6 Pengujian Eksperimental

Pada subbab ini, akan dilakukan pengujian sistem rekomendasi program studi Universitas Katolik Parahyangan. Pengujian dilakukan menggunakan *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan eksekusi waktu program. Data yang digunakan pada pengujian adalah seluruh data mahasiswa yang dibagi menjadi dua yaitu *train set* sebesar 70% dan *test set* sebesar 30%. Pengujian dilakukan dengan dua metode, yaitu :

### 5.6.1 Metode Dasar

Metode dasar ini dilakukan dengan cara menguji secara langsung *test set* kedalam sistem dengan menggunakan *train set*. Berikut merupakan hasil pengujian dengan metode dasar :

#### 1. IPA

No	MAE	RMSE	Time
1	0.2754545455	0.3495110995	0.002562046051
2	0.2718614719	0.330867939	0.002246141434
3	0.2943290043	0.3680562022	0.00180888176
4	0.2781659389	0.3533915527	0.002074956894
5	0.2694805195	0.3351777974	0.00253200531
6	0.2818181818	0.3456570379	0.00125002861
7	0.2858008658	0.363914504	0.001752138138
8	0.2703896104	0.3409561295	0.003805160522
9	0.2937662338	0.3675235953	0.00479888916
10	0.2926839827	0.3660287643	0.002127885818

Tabel 5.3: Hasil Pengujian Jurusan IPA dengan Metode Dasar

#### 2. IPS

No	MAE	RMSE	Time
1	0.2944	0.3655757104	0.003097057343
2	0.29092	0.3719166573	0.002082824707
3	0.28344	0.3553927405	0.00294303894
4	0.286064257	0.3595384145	0.001393079758
5	0.3212851406	0.3974486908	0.001703023911
6	0.29276	0.3556430795	0.002462148666

7	0.28544	0.3532149487	0.001994848251
8	0.2704	0.3468486702	0.001998186111
9	0.27304	0.3416021077	0.00242805481
10	0.28952	0.3625961941	0.001628875732

Tabel 5.4: Hasil Pengujian Jurusan IPS dengan Metode Dasar

### 5.6.2 Metode KMeans

Metode KMeans dilakukan dengan membuat kelompok dari *train set* dan menentukan *test set* masuk kedalam kelompok mana, untuk mengurangi perhitungan kemiripan atau similaritas dengan pengguna yang memiliki kemiripan kecil atau tidak memiliki kemiripan sama sekali. Pada pengujian ini dilakukan dengan nilai maksimum 10, 20, 30, dan 40 dengan pengulangan dalam pembentukan kelompok sebanyak 30 kali. Berikut merupakan hasil pengujian yang dilakukan dengan menggunakan k-means dengan memilih nilai minimum dari  $k = 2$  sampai  $k = n$  ( $n = 10, 20, 30$ , dan  $40$ ) :

#### 1. IPA

(a)  $k = 10$

No	k	MAE	k	RMSE	k	Time
1	10	0.227403435	10	0.299831433	9	39.32930207
2	10	0.2319802782	10	0.3163842676	8	44.21230698
3	9	0.2378234118	9	0.3031256224	8	97.19740295
4	10	0.2258213385	10	0.3049147694	10	84.33423114
5	10	0.2424204718	10	0.3144518957	6	98.56704521
6	10	0.2313912291	10	0.3061390768	7	92.61572003
7	10	0.2218367518	8	0.2863855107	7	91.99784708
8	10	0.2385816174	10	0.317593614	9	90.1020081
9	9	0.235875589	9	0.3079230922	10	43.22178102
10	10	0.2311139896	7	0.3090021998	10	42.38479996

Tabel 5.5: Hasil Pengujian KMeans Jurusan IPA dengan nilai  $k = 10$ 

Berdasarkan tabel 5.5 MAE memiliki nilai rata-rata 0.2324248112 dan RMSE memiliki nilai rata-rata 0.3065751482. Nilai  $k$  yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah 10.

(b)  $k = 20$

No	k	MAE	k	RMSE	k	Time
----	---	-----	---	------	---	------

Tabel 5.6: Hasil Pengujian KMeans Jurusan IPA dengan nilai  $k = 20$ 

Berdasarkan tabel 5.6 MAE memiliki nilai rata-rata dan RMSE memiliki nilai rata-rata. Nilai  $k$  yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah .

(c)  $k = 30$

No	k	MAE	k	RMSE	k	Time
----	---	-----	---	------	---	------

Tabel 5.7: Hasil Pengujian KMeans Jurusan IPA dengan nilai k 30

Berdasarkan tabel 5.7 MAE memiliki nilai rata-rata dan RMSE memiliki nilai rata-rata. Nilai k yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah .

(d) k = 40

No	k	MAE	k	RMSE	k	Time
----	---	-----	---	------	---	------

Tabel 5.8: Hasil Pengujian KMeans Jurusan IPA dengan nilai k 40

Berdasarkan tabel 5.8 MAE memiliki nilai rata-rata dan RMSE memiliki nilai rata-rata. Nilai k yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah .

## 2. IPS

(a) k = 10

No	k	MAE	k	RMSE	k	Time
1	10	0.2471143679	10	0.3158455625	10	54.29916787
2	10	0.2620947798	7	0.3363475028	10	52.25956702
3	10	0.244470971	9	0.3165210982	10	49.05109715
4	10	0.2518617969	10	0.3158170943	9	46.56690907
5	10	0.2553253166	10	0.3150100854	6	48.13360286
6	10	0.2849076998	10	0.3541113867	10	109.8035731
7	10	0.2757233651	10	0.3471834944	5	133.003581
8	10	0.2850747681	9	0.3568411162	10	102.0692439
9	10	0.2529911624	10	0.3205667096	7	110.884131
10	9	0.258458903	8	0.3287219636	7	111.2748449
10	10	0.2618023131	10	0.3306966014	10	81.73457179

Tabel 5.9: Hasil Pengujian KMeans Jurusan IPS dengan nilai k 10

Berdasarkan tabel 5.9 MAE memiliki nilai rata-rata 0.2618023131 dan RMSE memiliki nilai rata-rata 0.3306966014. Nilai k yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah 10.

(b) k = 20

No	k	MAE	k	RMSE	k	Time
----	---	-----	---	------	---	------

Tabel 5.10: Hasil Pengujian KMeans Jurusan IPS dengan nilai k 20

Berdasarkan tabel 5.10 MAE memiliki nilai rata-rata dan RMSE memiliki nilai rata-rata. Nilai k yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah .

(c)  $k = 30$

No	k	MAE	k	RMSE	k	Time
----	---	-----	---	------	---	------

Tabel 5.11: Hasil Pengujian KMeans Jurusan IPS dengan nilai k 30

Berdasarkan tabel 5.11 MAE memiliki nilai rata-rata dan RMSE memiliki nilai rata-rata. Nilai k yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah .

(d)  $k = 40$

No	k	MAE	k	RMSE	k	Time
----	---	-----	---	------	---	------

Tabel 5.12: Hasil Pengujian KMeans Jurusan IPS dengan nilai k 40

Berdasarkan tabel 5.12 MAE memiliki nilai rata-rata dan RMSE memiliki nilai rata-rata. Nilai k yang sering muncul pada pengujian ini untuk nilai minimum MAE dan RMSE adalah .

### 5.6.3 Kesimpulan Hasil Pengujian

#### 1. Metode Dasar

(a) IPA

Berdasarkan tabel 5.3 pengujian ipa dasar nilai MAE memiliki nilai rata-rata 0.2813750354 dan RMSE memiliki nilai rata-rata 0.3521084622.

(b) IPS

Berdasarkan tabel 5.4 pengujian ips dasar nilai MAE memiliki nilai rata-rata 0.2887269398 dan RMSE memiliki nilai rata-rata 0.3609777214.

#### 2. Metode K-Means

(a) IPA

(b) IPS



## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berikut ini merupakan kesimpulan yang didapatkan berdasarkan penelitian yang sudah dilakukan :

1. Sistem rekomendasi program studi Universitas Katolik Parahyangan sudah dapat dikembangkan dan memberikan rekomendasi kepada pengguna khususnya siswa/i SMA pada kelas 11 yang ingin melanjutkan pendidikan di Universitas Katolik Parahyangan.
2. Sistem rekomendasi program studi Universitas Katolik Parahyangan yang dikembangkan dengan teknik *user-based collaborative filtering* memberikan rekomendasi berdasarkan *rating* yang diberikan pengguna lain yang telah lulus dari Universitas Katolik Parahyangan.
3. Hasil pengujian menggunakan metode *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan eksekusi waktu program pada metode dasar dan metode Kmeans dengan nilai MAE berada di 0.2, nilai RMSE berada di 0.3, dan waktu eksekusi program lebih lama pada metode KMeans, dikarenakan diperlukan membuat kelompok pada data *train set*.

#### 6.2 Saran

Berdasarkan hasil penelitian yang dilakukan, penulis dapat memberikan beberapa saran sebagai berikut :

1. Sistem rekomendasi program studi Universitas Katolik Parahyangan menggunakan data mahasiswa lulus pada jalur PMDK, untuk itu perlu ditambahkan data mahasiswa lulus pada jalur USM, dengan harapan semakin banyak jumlah pengguna yang memiliki kesanamaan dengan siswa/i SMA yang menggunakan sistem.
2. Sistem rekomendasi program studi Universitas Katolik Parahyangan menggunakan data mahasiswa yang berasal dari Biro Teknologi Informasi (BTI) Universitas Katolik Parahyangan. Penulis berharap agar format penyimpanan nilai baik untuk jalur penerimaan PMDK dan UMS sama, dengan harapan mudah untuk digunakan pada sistem yang sudah dibangun.
3. Teknik pengelompokkan yang digunakan adalah KMeans dimana saat pengujian membutuhkan waktu yang lebih lama, metode ini memberikan masalah pada waktu eksekusi program. Berdasarkan masalah ini, penulis berharap sistem dapat menggunakan metode lain untuk mengoptimalkan waktu eksekusi program.





## DAFTAR REFERENSI



# LAMPIRAN A

## KODE PROGRAM

Listing A.1: filter\_data.py

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7 #mtk 1
8 #ind 2
9 #ing 3
10 #fis 4
11 #gbr 5
12 #pkn 6
13 #kim 7
14
15 import pandas as pd
16 #import math as m
17
18 #ftis13 -> mtk if 2
19 #ftis2 -> fis 3
20 data = pd.read_csv("ftis13.csv");
21
22
23 mhs = pd.DataFrame(columns=["id_user", "NPM", "IPK", "id_jurusan", "id_program_studi"])
24 nilai = pd.DataFrame(columns=["id_nilai", "id_mata_pelajaran", "id_user", "101", "102", "111", "112", "AVG"])
25
26 def generateCSV(mhs, nilai, count):
27     size= data.shape[0]
28     batas = int(size/(4*count))
29
30     id_user=1529 #pake counter
31     id_nilai=3609 #pake counter
32
33     ipa = 1
34     ips = 2
35
36
37     for idx in range(batas):
38         idx = idx*(4*count)
39
40
41         #input untuk table mahasiswa
42         npm = str(data.iloc[idx][0])
43         id_prodi = data.iloc[idx][1]
44         #id_fakultas = id_prodi/10
45         #id_fakultas = m.floor(id_fakultas/10)
46         id_jurusan = ipa
47         for i in range(7,18):
48             ipk = data.iloc[idx][i]
49             if(ipk!=0 and data.iloc[idx][i+1]==0):
50                 break
51
52         mhs = mhs.append({"id_user":id_user, "NPM":npm, "IPK":ipk, "id_jurusan":id_jurusan,
53             "id_program_studi":id_prodi}, ignore_index=True)
54
55         #input untuk tabel nilai
56
57         for i in range(0,count):
58             row = idx+i
59             id_mp = data.iloc[row][3]
60             m_101 = (data.iloc[row+0*count][5]/20)-1
61             m_102 = (data.iloc[row+1*count][5]/20)-1
62             m_111 = (data.iloc[row+2*count][5]/20)-1
63             m_112 = (data.iloc[row+3*count][5]/20)-1
64             #print(row, "\n")
65
66             avg = (m_101+m_102+m_111+m_112)/4
67
68             #masukin ke nilai
69             nilai = nilai.append({"id_nilai":id_nilai, "id_mata_pelajaran":id_mp, "id_user":id_user,
70                 "101":m_101, "102":m_102, "111":m_111, "112":m_112, "AVG":avg},
71                 ignore_index=True)
72
73             id_nilai+=1
74             id_user+=1
75         return mhs, nilai
```

```

76 rest = generateCSV(mhs, nilai, 2)
77
78 mhs = rest[0]
79 mhs.to_csv(r"C:\Users\anugrahjaya1\Downloads\Data\7_FTIS\new_ftis_mhs13.csv", index=None, header=True)
80
81 nilai= rest[1]
82 nilai.to_csv(r"C:\Users\anugrahjaya1\Downloads\Data\7_FTIS\new_ftis_nilai13.csv", index=None, header=True)
83

```

Listing A.2: MahasiswaController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Mahasiswa;
6 use Illuminate\Http\Request;
7
8 class MahasiswaController extends Controller
9 {
10     public function index($jurusanSMA)
11     {
12         $idJurusan = 1; //IPA
13         if ($jurusanSMA == "IPS") {
14             $idJurusan = 2;
15         }
16
17         $dataMahasiswa = $this->dataMahasiswa($idJurusan);
18
19         return $dataMahasiswa;
20     }
21
22     private function dataMahasiswa($idJurusan)
23     {
24         $query = Mahasiswa::with('Nilai')->where(['id_jurusan'=> $idJurusan])->get();
25         // cuma ambil id_user, NPM, id_mata_pelajaran, nilai, avg, id_program_studi
26
27         return $query;
28     }
29 }

```

Listing A.3: SiswaController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use App\Http\Controllers\UserBasedModelController;
8 use App\Http\Controllers\MahasiswaController;
9 use App\Http\Controllers\KMeansController;
10
11 class SiswaController extends Controller
12 {
13     private $mataPelajaran = array(
14         "mtk" => 1,
15         "ind" => 2,
16         "ing" => 3,
17         "fsk" => 4,
18         "gbr" => 5,
19         "pkn" => 6,
20         "kma" => 7,
21     );
22
23     function index(Request $request)
24     {
25         // untuk penampung input dari form
26         $data = $request->input();
27         // untuk menampung input yang sudah diolah, agar mudah digunakan
28         $siswa = $this->dataSiswa($data);
29
30         // inisialisasi controller mahasiswa
31         $mahasiswa = new MahasiswaController();
32         // data mahasiswa
33         $mhs = $mahasiswa->index($siswa["btn"])->toArray();
34         // inisialisasi kmeans
35         $kmeans = new KMeansController(10, $mhs);
36
37         // hitung jarak siswa dengan centroid
38         // mengembalikan siswa masuk dalam cluster mana
39         $cluster = $kmeans->hitungJarakSiswa($siswa);
40
41         // mengubah data mhs dari seluruh mhs
42         // menjadi anggota satu cluster dengan siswa
43         $mhs = $kmeans->getCluster($cluster);
44
45         // inisialisasi userBasedModel
46         $userBasedModel = new UserBasedModelController($mhs, $siswa);
47
48         $result = $userBasedModel->getResult();
49
50         return view('/result', ['result' => $result]);
51     }
52
53     // mengubah data siswa dari form
54     // menjadi array dengan format mengikuti array mahasiswa

```

```

55 private function dataSiswa($data)
56 {
57     $i = 1;
58     $result = array();
59     $result["nilai"] = array();
60     foreach ($data as $key => $value) {
61         if ($key == "_token") {
62             $result[$key] = $value;
63         } else {
64             if ($i == 1) {
65                 // key untuk mata pelajaran
66                 $k = substr($key, 0, 3);
67                 // temporary array
68                 $temp = [];
69                 // masukan data (nilai) ke temp
70                 array_push($temp, ((int) $value / 20) - 1);
71                 $i++;
72             } else {
73                 // masukan data nilai ke temp
74                 array_push($temp, ((int) $value / 20) - 1);
75                 $i++;
76             }
77             if ($i == 5) {
78                 // avg nilai
79
80                 $avg = array_sum($temp) / count($temp);
81                 array_push($temp, $avg);
82
83                 // replace index ke-4 dengan AVG
84                 $temp = $this->replaceKey($temp, 4, "AVG");
85
86                 array_push($temp, $this->mataPelajaran[$k]);
87
88                 $temp = $this->replaceKey($temp, 5, "id_mata_pelajaran");
89
90                 // masukan data ke result
91                 // $result[$this->mata_pelajaran[$k]] = $temp;
92                 array_push($result["nilai"], $temp);
93                 // print($k." ");
94                 $i = 1;
95             }
96         }
97     }
98 }
99
100 if (!empty($data["btnIPA"])) {
101     $result["btn"] = "IPA";
102 } else if (!empty($data["btnIPS"])) {
103     $result["btn"] = "IPS";
104 }
105 return $result;
106 }
107
108 private function replaceKey($temp, $oldKey, $newKey)
109 {
110     $temp[$newKey] = $temp[$oldKey];
111     unset($temp[$oldKey]);
112
113     return $temp;
114 }
115 }

```

Listing A.4: KMeansController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class KMeansController extends Controller
8 {
9     private $currCentroid;
10    private $prevCentroid;
11    private $k;
12    public $cluster;
13    private $mahasiswa;
14    private $J0, $J1;
15
16    function __construct($k, $dataMahasiswa)
17    {
18        $this->k = $k;
19        $this->mahasiswa = $dataMahasiswa;
20        $this->inisialisasiCluster();
21        $this->currCentroid = array();
22
23        // J0 = inisialisasi jarak total dari objek ke centroid-nya
24        $this->J0 = 100;
25
26        $this->pilihCentroid();
27
28        $this->hitungJarakMhs();
29
30        $status = true;
31        $idx = 0;
32        while ($status) {
33            $this->hitungCentroidBaru();
34            $idx++;

```

```

35         $status = $this->cekBatas();
36         $this->hitungJarakMhs();
37     }
38 }
39
40
41 private function inisialisasiCluster()
42 {
43     $this->cluster = array();
44     // inisialisasi untuk res anggota cluster
45     for ($i = 0; $i < $this->k; $i++) {
46         $this->cluster[$i] = array();
47     }
48 }
49
50 // fungsi untuk memilih secara acak mahasiswa mana
51 // yang akan dijadikan centroid
52 private function pilihCentroid()
53 {
54     $i = 0;
55     while ($i < $this->k) {
56         $key = rand(0, 1739);
57         if (array_key_exists($key, $this->mahasiswa)) {
58             // cek apakah ada key(idx) pada currCentroid
59             if (!array_key_exists($key, $this->currCentroid)) {
60                 array_push($this->currCentroid, $this->mahasiswa[$key]);
61                 $i++;
62             }
63         }
64     }
65 }
66
67 // fungsi untuk menghitung jarak untuk untuk
68 // mata pelajaran mtk (1) dan ing (3)
69 private function hitungJarakMhs()
70 {
71     $this->jl = 0;
72     // looping sebanyak mahasiswa
73     foreach ($this->mahasiswa as $keyMhs => $valueMhs) {
74         // array sementara untuk menentukan masuk cluster mana
75         $tempCluster = array();
76         // array yang berisikan nilai satu mahasiswa
77         $nilaiMhs = $valueMhs['nilai'];
78         // looping sebanyak nilai mahasiswa
79         foreach ($nilaiMhs as $keyNilaiMhs => $valueNilaiMhs) {
80             // array untuk menampung jarak
81             $arrJarak = array();
82             // looping sebanyak centroid
83             foreach ($this->currCentroid as $keyCen => $valuCen) {
84                 // penampung jarak
85                 $jarak = 0;
86                 // array yang berisikan nilai pada centroid
87                 $nilaiCen = $valuCen['nilai'];
88                 // looping sebanyak nilai centroid
89                 foreach ($nilaiCen as $keyNilaiCen => $valueNilaiCen) {
90                     // cek apakah pada mata pelajaran yang sama atau tidak
91                     if (
92                         ($valueNilaiMhs['id_mata_pelajaran'] == 1 && $valueNilaiCen['id_mata_pelajaran'] == 1)
93                         ||
94                         ($valueNilaiMhs['id_mata_pelajaran'] == 3 && $valueNilaiCen['id_mata_pelajaran'] == 3)
95                     ) {
96                         // hitung jarak dengan euclidian distance
97                         $jarak = $this->euclidianDistance($valueNilaiMhs, $valueNilaiCen);
98                     } else if ($valueNilaiMhs['id_mata_pelajaran'] < $valueNilaiCen['id_mata_pelajaran']) {
99                         break;
100                     }
101                 }
102                 // memasukkan jarak antara mhs(nilai) dengan centroid(nilai)
103                 // index 0 nilai dengan mata pelajaran mtk (1)
104                 // index 1 nilai dengan mata pelajaran ing (3)
105                 array_push($arrJarak, $jarak);
106             }
107             // cek apakah tempCluster kosong
108             if (empty($tempCluster)) {
109                 array_push($tempCluster, $arrJarak);
110             } else {
111                 // menghitung jarak sebenarnya
112                 // dari dua nilai
113                 for ($i = 0; $i < $this->k; $i++) {
114                     $tempCluster[0][$i] += $arrJarak[$i];
115                     $tempCluster[0][$i] = sqrt($tempCluster[0][$i]);
116                 }
117             }
118             // menentukan mhs masuk pada cluster mana
119             $c = array_keys($tempCluster[0], min($tempCluster[0]))[0];
120             $this->jl += $tempCluster[0][$c];
121             array_push($tempCluster[0], $c, $valueMhs['id_mahasiswa']);
122
123             // mengubah key index array
124             $tempCluster[0]['id_mahasiswa'] = $tempCluster[0][$this->k + 1];
125             unset($tempCluster[0][$this->k + 1]);
126
127             // memasukkan mhs ke array hasil
128             array_push($this->cluster[$c], $valueMhs);
129         }
130     }
131 }
132
133 }

```

```

134 |
135 | public function hitungJarakSiswa($siswa)
136 | {
137 |     $nilaiSiswa = $siswa['nilai'];
138 |     $tempCluster = array();
139 |     // looping untuk nilai siswa
140 |     foreach ($nilaiSiswa as $keyNilaiSiswa => $valueNilaiSiswa) {
141 |         $arrJarak = array();
142 |         // looping sebanyak centroid
143 |         foreach ($this->currCentroid as $keyCen => $valuCen) {
144 |             // penampung jarak
145 |             $jarak = 0;
146 |             // array yang berisikan nilai pada centroid
147 |             $nilaiCen = $valuCen['nilai'];
148 |             // looping sebanyak nilai centroid
149 |             foreach ($nilaiCen as $keyNilaiCen => $valueNilaiCen) {
150 |                 // cek apakah pada mata pelajaran yang sama atau tidak
151 |                 if (
152 |                     ($valueNilaiSiswa['id_mata_pelajaran'] == 1 && $valueNilaiCen['id_mata_pelajaran'] == 1)
153 |                     ||
154 |                     ($valueNilaiSiswa['id_mata_pelajaran'] == 3 && $valueNilaiCen['id_mata_pelajaran'] == 3)
155 |                 ) {
156 |                     // hitung jarak dengan euclidian distance
157 |                     $jarak = $this->euclidiandistance($valueNilaiSiswa, $valueNilaiCen);
158 |                     } else if ($valueNilaiSiswa['id_mata_pelajaran'] < $valueNilaiCen['id_mata_pelajaran']) {
159 |                         break;
160 |                     }
161 |                 }
162 |                 // memasukkan jarak antara mhs(nilai) dengan centroid(nilai)
163 |                 // index 0 nilai dengan mata pelajaran mtk (1)
164 |                 // index 1 nilai dengan mata pelajaran ing (3)
165 |                 array_push($arrJarak, $jarak);
166 |             }
167 |             // cek apakah tempCluster kosong
168 |             if (empty($tempCluster)) {
169 |                 array_push($tempCluster, $arrJarak);
170 |             } else {
171 |                 // menghitung jarak sebenarnya
172 |                 // dari dua nilai
173 |                 for ($i = 0; $i < $this->k; $i++) {
174 |                     $tempCluster[0][$i] += $arrJarak[$i];
175 |                     $tempCluster[0][$i] = sqrt($tempCluster[0][$i]);
176 |                 }
177 |             }
178 |         }
179 |         // menentukan mhs masuk pada cluster mana
180 |         $res = array_keys($tempCluster[0], min($tempCluster[0]))[0];
181 |
182 |         return $res;
183 |     }
184 |
185 |     // parameter berisikan array of nilai satu mata pelajaran
186 |     private function euclidiandistance($mhs, $centroid)
187 |     {
188 |         // asumsi itung yang beririsan aja
189 |         $result = 0;
190 |         for ($i = 0; $i < 4; $i++) {
191 |             $result += pow($mhs[$i] - $centroid[$i], 2);
192 |         }
193 |         $result += pow($mhs['AVG'] - $centroid['AVG'], 2);
194 |         return $result;
195 |     }
196 |
197 |     private function hitungCentroidBaru()
198 |     {
199 |         // mengisi centroid sebelumnya dengan centroid saat ini
200 |         $this->prevCentroid = $this->currCentroid;
201 |
202 |         // reset nilai curr centroid
203 |         $this->resetCentroid();
204 |
205 |         // looping sebanyak centroid
206 |         foreach ($this->currCentroid as $keyCen => $valuCen) {
207 |             // array yang berisikan nilai pada centroid
208 |             $nilaiCen = $valuCen['nilai'];
209 |             // looping sebanyak nilai centroid
210 |             foreach ($nilaiCen as $keyNilaiCen => $valueNilaiCen) {
211 |                 // penampung untuk anggota pada cluster tertentu (sesuai index/key)
212 |                 $anggota = $this->cluster[$keyCen];
213 |                 if (count($anggota) != 0) {
214 |                     foreach ($anggota as $keyAnggota => $valueAnggota) {
215 |                         // penampung untuk nilai anggota
216 |                         $nilaiAnggota = $valueAnggota['nilai'];
217 |                         foreach ($nilaiAnggota as $keyNilaiAnggota => $valueNilaiAnggota) {
218 |                             if (
219 |                                 ($valueNilaiCen['id_mata_pelajaran'] == 1 && $valueNilaiAnggota['id_mata_pelajaran'] == 1)
220 |                                 ||
221 |                                 ($valueNilaiCen['id_mata_pelajaran'] == 3 && $valueNilaiAnggota['id_mata_pelajaran'] == 3)
222 |                             ) {
223 |                                 // update nilai 101, 102, 111, 112
224 |                                 for ($i = 0; $i < 4; $i++) {
225 |                                     $nilaiLama = $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen][$i];
226 |                                     $nilaiBaru = $anggota[$keyAnggota]['nilai'][$keyNilaiAnggota][$i];
227 |
228 |                                     $this->updateNilai($keyCen, $keyNilaiCen, $nilaiLama, $nilaiBaru, $i);
229 |                                 }
230 |
231 |                                 // update nilai avg
232 |                                 $nilaiLama = $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen]['AVG'];

```

```

233         $nilaiBaru = $anggota[$keyAnggota]['nilai'][$keyNilaiAnggota]['AVG'];
234         $this->updateNilai($keyCen, $keyNilaiCen, $nilaiLama, $nilaiBaru, 'AVG');
235     }
236 }
237 }
238 } else {
239     // random nilai baru
240     $this->randomNilaiBaru($keyCen, $keyNilaiCen);
241 }
242 }
243 }
244
245 $this->hitungRata2();
246 }
247
248 private function resetCentroid()
249 {
250     foreach ($this->currCentroid as $keyCen => $valuCen) {
251         // array yang berisikan nilai pada centroid
252         $nilaiCen = $valuCen['nilai'];
253         // looping sebanyak nilai centroid
254         foreach ($nilaiCen as $keyNilaiCen => $valueNilaiCen) {
255             for ($i = 0; $i < 4; $i++) {
256                 $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen][$i] = 0;
257             }
258             // update nilai avg
259             $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen]['AVG'] = 0;
260         }
261     }
262 }
263
264 private function updateNilai($keyCen, $keyNilaiCen, $nilaiLama, $nilaiBaru, $i)
265 {
266     $nilai = $nilaiLama + $nilaiBaru;
267     $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen][$i] = $nilai;
268 }
269
270 private function hitungRata2()
271 {
272     foreach ($this->currCentroid as $keyCen => $valuCen) {
273         // array yang berisikan nilai pada centroid
274         $nilaiCen = $valuCen['nilai'];
275         // looping sebanyak nilai centroid
276         $anggota = $this->cluster[$keyCen];
277         $count = count($anggota);
278         if ($count != 0) {
279             foreach ($nilaiCen as $keyNilaiCen => $valueNilaiCen) {
280                 for ($i = 0; $i < 4; $i++) {
281                     $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen][$i] = $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen][$i] / $count;
282                 }
283                 // update nilai avg
284                 $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen]['AVG'] = $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen]['AVG'] / $count;
285             }
286         }
287     }
288 }
289
290 private function randomNilaiBaru($keyCen, $keyNilaiCen)
291 {
292     for ($i = 0; $i < 4; $i++) {
293         $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen][$i] = rand(1, 3) + rand(1, 10) / 10;
294     }
295     // update nilai avg
296     $this->currCentroid[$keyCen]['nilai'][$keyNilaiCen]['AVG'] = rand(1, 3) + rand(1, 10) / 10;
297 }
298
299 private function cekBatas()
300 {
301     $batas = abs($this->J0 - $this->J1);
302
303     if ($batas < 0.1) {
304         return false;
305     }
306
307     $this->J0 = $this->J1;
308
309     return true;
310 }
311
312 public function getCluster($idx)
313 {
314     return $this->cluster[$idx];
315 }
316 }
317 }

```

Listing A.5: UserBasedModelController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class UserBasedModelController extends Controller
8 {

```



```

9 | private $pearsonCorrelation;
10 | private $prediction;
11 | private $result;
12 |
13 | // mode 0 untuk pengguna
14 | // mode 1 untuk pengujian
15 | public function __construct($mahasiswa, $siswa, $mode = 0)
16 | {
17 |     // inisialisasi prediction controller
18 |     $this->prediction = new PredictionController();
19 |
20 |     if ($mode == 0) {
21 |         // inisialisasi pearson correlation controller
22 |         $this->pearsonCorrelation = new PearsonCorrelationController();
23 |         // menghitung kesamaan atau similaritas
24 |         $pearson = $this->calculateSimilarity($mahasiswa, $siswa);
25 |
26 |         // menghitung prediksi IPK untuk siswa
27 |         $this->result = $this->calculatePredict($pearson);
28 |     } else if ($mode == 1) {
29 |         $this->pearsonCorrelation = new PearsonCorrelationPengujianController();
30 |     }
31 | }
32 |
33 | public function calculateSimilarity($mahasiswa, $siswa)
34 | {
35 |     return $this->pearsonCorrelation->calculatePearson($mahasiswa, $siswa);
36 | }
37 |
38 | public function calculatePredict($pearson)
39 | {
40 |     return $this->prediction->calculatePredict($pearson);
41 | }
42 |
43 | public function getResult()
44 | {
45 |     return $this->result;
46 | }
47 | }

```

Listing A.6: PearsonCorrelationController.php

```

1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 |
7 | class PearsonCorrelationController extends Controller
8 | {
9 |
10 |     // menghitung kemiripan dengan perason
11 |     // $mahasiswa -> seluruh mahasiswa sesuai dengan jurusan SMA
12 |     // $siswa ->
13 |     public function calculatePearson($mahasiswa, $siswa)
14 |     {
15 |         $res = array();
16 |         foreach ($mahasiswa as $mhs) {
17 |             $covariance = $this->calculateCovariance($mhs, $siswa);
18 |             $sd = $this->calculateStandarDeviation($mhs, $siswa);
19 |             $sdMhs = $sd[0]; // standar deviasi untuk mahasiswa
20 |             $sdSiswa = $sd[1]; // standar deviasi untuk siswa
21 |
22 |             $idProdi = $mhs['id_program_studi'];
23 |             $IPK = $mhs['IPK'];
24 |
25 |             $sim = $covariance / ($sdMhs * $sdSiswa);
26 |             // atur threshold
27 |             if ($sim > 0) {
28 |                 // inisialisasi array agar tidak null
29 |                 $res[$mhs['id_mahasiswa']] = array();
30 |                 array_push($res[$mhs['id_mahasiswa']], $sim, $idProdi, $IPK);
31 |             }
32 |         }
33 |         return $res;
34 |     }
35 |
36 |     // untuk menghitung kovariansi satu mahasiswa
37 |     // dengan satu siswa
38 |     private function calculateCovariance($mhs, $siswa)
39 |     {
40 |         $res = 0;
41 |         // nilai 1 mhs
42 |         $nilaiMhs = $mhs['nilai'];
43 |         $nilaiSiswa = $siswa['nilai'];
44 |         // looping sebanyak nilai
45 |         foreach ($nilaiSiswa as $nSiswa) {
46 |             $idMP = $nSiswa['id_mata_pelajaran'];
47 |             // looping sebanyak nilai mahasiswa pada index
48 |             foreach ($nilaiMhs as $nMhs) {
49 |                 // hanya menghitung mata pelajaran yang beraturan
50 |                 if ($idMP == $nMhs['id_mata_pelajaran']) {
51 |                     for ($i = 0; $i < 4; $i++) {
52 |                         // mahasiswa * siswa
53 |                         $res += ($nMhs[$i] - $nMhs["AVG"]) * ($nSiswa[$i] - $nSiswa["AVG"]);
54 |                     }
55 |                 } else if ($idMP < $nMhs['id_mata_pelajaran']) {
56 |                     break;

```

```

57     }
58   }
59 }
60 return $res;
61 }
62
63 // mengitung standar deviasi untuk satu mahasiswa
64 // dan satu siswa
65 private function calculateStandarDeviation($mhs, $siswa)
66 {
67     $res = array();
68
69     $sdMhs = 0;
70     $sdSiswa = 0;
71     // nilai 1 mhs
72     $nilaiMhs = $mhs['nilai'];
73     $nilaiSiswa = $siswa['nilai'];
74     // looping sebanyak nilai
75     foreach ($nilaiSiswa as $nSiswa) {
76         $idMP = $nSiswa['id_mata_pelajaran'];
77         // looping sebanyak nilai mahasiswa pada index
78         foreach ($nilaiMhs as $nMhs) {
79             // hanya menghitung mata pelajaran yang beririsan
80             if ($idMP == $nMhs['id_mata_pelajaran']) {
81                 for ($i = 0; $i < 4; $i++) {
82                     $sdMhs += pow($nMhs[$i] - $nMhs['AVG'], 2);
83                     $sdSiswa += pow($nSiswa[$i] - $nSiswa['AVG'], 2);
84                 }
85             } else if ($idMP < $nMhs['id_mata_pelajaran']) {
86                 break;
87             }
88         }
89     }
90     array_push($res, sqrt($sdMhs), sqrt($sdSiswa));
91
92     return $res;
93 }
94 }

```

Listing A.7: PredictionController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class PredictionController extends Controller
8 {
9     private $programStudi;
10    private $fakultas;
11
12    public function __construct()
13    {
14        $this->programStudi = new ProgramStudiController();
15        $this->fakultas = new FakultasController();
16    }
17
18    public function calculatePredict($pearson)
19    {
20        $res = array();
21
22        // a = Sigma(sim * IPK)
23        $a = 0;
24        // b = Sigma(sim)
25        $b = 0;
26        // pred = a/b
27        // id_mhs -> sim, id_prodi, IPK, avgMhs, avgSiswa
28        foreach ($pearson as $id_mhs => $value) {
29            $a += $value[0] * $value[2];
30            $b += $value[0];
31
32            $next = next($pearson);
33
34            if ($next != null) {
35                // program studi mhs sekarang berbeda dengan mhs selanjutnya
36                if ($value[1] != $next[1]) {
37                    $res = $this->insertData($res, $a, $b, $value[1]);
38
39                    $a = 0;
40                    $b = 0;
41                }
42            } else if ($next == null) {
43                $res = $this->insertData($res, $a, $b, $value[1]);
44            }
45        }
46
47        // // penampung untuk nilai prediksi IPK
48        $score = array_column($res, 0);
49        // sort berdasarkan nilai prediksi ipk terbesar
50        array_multisort($score, SORT_DESC, $res);
51
52        return $res;
53    }
54
55    private function insertData($res, $a, $b, $idProdi)
56    {
57        $pred = $a / $b;

```

```

58     $namaFakultas = $this->fakultas->getNamaFakultas($idProdi);
59     $namaProdi = $this->programStudi->getNamaProgramStudi($idProdi);
60     $res[$idProdi] = array();
61     // dibalik
62     array_push($res[$idProdi], $pred, $namaFakultas, $namaProdi);
63
64     return $res;
65 }
66 }

```

Listing A.8: FakultasController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Fakultas;
7
8 class FakultasController extends Controller
9 {
10     public function getNamaFakultas($idProgramStudi){
11         $idFakultas = floor($idProgramStudi/100);
12         $fakultas = Fakultas::find($idFakultas);
13         $namaFakultas = $fakultas->nama_fakultas;
14         return $namaFakultas;
15     }
16 }

```

Listing A.9: ProgramStudiController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use App\Program_Studi;
8
9 class ProgramStudiController extends Controller
10 {
11     public function getNamaProgramStudi($idProgramStudi)
12     {
13         $prodi = Program_Studi::find($idProgramStudi);
14         $namaProdi = $prodi->nama_program_studi;
15         return $namaProdi;
16     }
17 }

```

Listing A.10: PengujianController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Http\Controllers\MahasiswaController;
7 use Phpml\CrossValidation\RandomSplit;
8 use App\Http\Controllers\KMeansController;
9 use App\Http\Controllers\PearsonCorrelationPengujianController;
10
11
12 use Phpml\Dataset\ArrayDataset;
13
14 class PengujianController extends Controller
15 {
16     private $train, $test;
17     private $userBasedModel;
18     private $error1, $error2;
19     private $accuracy;
20     private $metode;
21
22     function __construct(Request $request)
23     {
24         $btn = $request->input();
25         $idJurusan = substr($btn['btn'], 0, 3);
26         $this->metode = substr($btn['btn'], 4, strlen($btn['btn']));
27
28         $mhs = new MahasiswaController();
29         $data = $mhs->index($idJurusan)->toArray();
30
31         // untuk label setiap data
32         $arrLabel = array();
33
34         // array labelnya bisa pake id_program_studi
35         foreach ($data as $m) {
36             array_push($arrLabel, $m["id_program_studi"]);
37         }
38         // array sample dan label
39         $dataset = new ArrayDataset($data, $arrLabel);
40
41         $dataset = new RandomSplit($dataset, 0.3);
42
43         $this->train = $dataset->getTrainSamples();
44

```

```

45     $this->test = $dataset->getTestSamples();
46
47     $this->accuracy = new AccuracyController();
48
49     $this->userBasedModel = new UserBasedModelController(null, null, 1);
50 }
51
52 public function index()
53 {
54     if ($this->metode == 'Basic') {
55         return $this->pengujianBasic();
56     } else {
57         // k, looping dataset sebanyak n
58         return $this->pengujianKmeans(30, 30);
59     }
60 }
61
62 // bts untuk jumlah k di kmeans
63 // n jumlah pengulangan kmeans
64 private function pengujianKmeans($bts, $n)
65 {
66     $result = array();
67
68     $resultPred = array();
69
70     // looping dari 2-10 (untuk nilai k)
71     for ($k = 2; $k <= $bts; $k++) {
72         $start = microtime(true);
73
74         $tempMae = 0;
75         $tempRmse = 0;
76         // untuk pengujian sebanyak 30 centroid yang berbeda
77         for ($i = 0; $i < $n; $i++) {
78             $kmeans = new KMeansController($k, $this->train);
79
80             $this->error1 = array();
81             $this->error2 = array();
82             // test = siswa
83             foreach ($this->test as $t) {
84                 // biar tidak ada duplikat
85                 if (!array_key_exists($t["NPM"], $resultPred)) {
86                     $temp = array();
87
88                     // hitung jarak siswa dengan centroid
89                     // mengembalikan siswa masuk dalam cluster mana
90                     $cluster = $kmeans->hitungJarakSiswa($t);
91
92                     // mengubah data mhs dari seluruh mhs
93                     // menjadi anggota satu cluster dengan siswa
94                     $dataTrain = $kmeans->getCluster($cluster);
95
96                     $pearson = $this->userBasedModel->calculateSimilarity($dataTrain, $t);
97
98                     $predict = $this->userBasedModel->calculatePredict($pearson);
99
100                    if ($predict != null) {
101                        // Hitung selisih untuk mean absolute error
102                        $diff1 = abs($t["IPK"] - number_format($predict[0][0], 2));
103                        // Memasukkan diff1 kepada arr
104                        array_push($this->error1, $diff1);
105
106                        // Hitung selisih untuk root mean square error
107                        $diff2 = pow($t["IPK"] - number_format($predict[0][0], 2), 2);
108                        // Memasukkan diff1 kepada arr
109                        array_push($this->error2, $diff2);
110
111                        // isinya npm, nama programstudi, IPK, Prediksi, diff
112                        array_push(
113                            $temp,
114                            $t["NPM"],
115                            $predict[0][2],
116                            $t["IPK"],
117                            number_format($predict[0][0], 2),
118                            $diff1,
119                            $diff2
120                        );
121                        // Memasukkan array temp pada array result
122                        array_push($resultPred, $temp);
123                    }
124                }
125            }
126
127            $mae = $this->accuracy->calculateMAE($this->error1);
128            $rmse = $this->accuracy->calculateRMSE($this->error2);
129
130            $tempMae += $mae;
131            $tempRmse += $rmse;
132        }
133        $send = microtime(true);
134        $times = $send - $start;
135
136        $result[$k] = array();
137
138        // masukin semua ke penampung
139        array_push($result[$k], $tempMae / $n, $tempRmse / $n, $times);
140    }
141    $minMae = min(array_column($result, 0));
142    $minRmse = min(array_column($result, 1));
143    $minTime = min(array_column($result, 2));

```

```

144     return view('/pengujian', [
145         'status' => TRUE, 'resultPred' => $resultPred,
146         'metode' => $this->metode, 'result' => $result,
147         'minMae' => $minMae, 'minRmse' => $minRmse, 'minTime' => $minTime
148     ]);
149 }
150
151 private function pengujianBasic()
152 {
153     $result = array();
154
155     $this->error1 = array();
156     $this->error2 = array();
157
158     foreach ($this->test as $t) {
159         $start = microtime(true);
160         // biar tidak ada duplikat
161         if (!array_key_exists($t["NPM"], $result)) {
162             $temp = array();
163
164             $pearson = $this->userBasedModel->calculateSimilarity($this->train, $t);
165
166             $predict = $this->userBasedModel->calculatePredict($pearson);
167
168             if ($predict != null) {
169                 // Hitung selisih untuk mean absolute error
170                 $diff1 = abs($t["IPK"] - number_format($predict[0][0], 2));
171                 // Memasukkan diff1 kepada arr
172                 array_push($this->error1, $diff1);
173
174                 // Hitung selisih untuk root mean square error
175                 $diff2 = pow($t["IPK"] - number_format($predict[0][0], 2), 2);
176                 // Memasukkan diff1 kepada arr
177                 array_push($this->error2, $diff2);
178
179                 // isinya npm, nama programstudi, IPK, Prediksi, diff
180                 array_push(
181                     $temp,
182                     $t["NPM"],
183                     $predict[0][2],
184                     $t["IPK"],
185                     number_format($predict[0][0], 2),
186                     $diff1,
187                     $diff2
188                 );
189                 // Memasukkan array temp pada array result
190                 array_push($result, $temp);
191             }
192         }
193     }
194
195     $mae = $this->accuracy->calculateMAE($this->error1);
196     $rmse = $this->accuracy->calculateRMSE($this->error2);
197     $end = microtime(true);
198     $times = $end - $start;
199     return view('/pengujian', [
200         'status' => TRUE, 'result' => $result,
201         'mae' => $mae, 'rmse' => $rmse,
202         'times' => $times, 'metode' => $this->metode
203     ]);
204 }
205 }
206 }

```

Listing A.11: PearsonCorrelationPengujianController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class PearsonCorrelationPengujianController extends Controller
8 {
9     private $sdSiswa;
10
11     function __construct()
12     {
13         $this->sdSiswa = array();
14     }
15
16     // untuk menghitung kovariansi satu mahasiswa
17     // dengan satu siswa
18     private function calculateCovariance($mhs, $siswa)
19     {
20         $res = 0;
21         // nilai 1 mhs
22         $nilaiMhs = $mhs['nilai'];
23         $nilaiSiswa = $siswa['nilai'];
24         // looping sebanyak nilai
25         foreach ($nilaiSiswa as $nSiswa) {
26             $idMP = $nSiswa['id_mata_pelajaran'];
27             foreach ($nilaiMhs as $nMhs) {
28                 if ($idMP == $nMhs['id_mata_pelajaran']) {
29                     for ($i = 0; $i < 4; $i++) {
30                         //mahasiswa * siswa
31                         $res += ($nMhs[$i] - $nMhs["AVG"]) * ($nSiswa[$i] - $nSiswa["AVG"]);
32                     }
33                 }
34             }
35         }
36     }
37 }

```

```

33         } else if ($idMP < $nMhs['id_mata_pelajaran']) {
34             break;
35         }
36     }
37 }
38 return $res;
39 }
40
41 // menghitung standar deviasi untuk satu mahasiswa
42 // dan satu siswa
43 private function calculateStandarDeviation($mhs, $siswa)
44 {
45     $res = array();
46
47     $sdMhs = 0;
48     $sdSiswa = 0;
49     // nilai 1 mhs
50     $nilaiMhs = $mhs['nilai'];
51     $nilaiSiswa = $siswa['nilai'];
52     // looping sebanyak nilai
53     foreach ($nilaiSiswa as $nSiswa) {
54         $idMP = $nSiswa['id_mata_pelajaran'];
55         foreach ($nilaiMhs as $nMhs) {
56             if ($idMP == $nMhs['id_mata_pelajaran']) {
57                 for ($i = 0; $i < 4; $i++) {
58                     $sdMhs += pow($nMhs[$i] - $nMhs['AVG'], 2);
59                     $sdSiswa += pow($nSiswa[$i] - $nSiswa['AVG'], 2);
60                 }
61             } else if ($idMP < $nMhs['id_mata_pelajaran']) {
62                 break;
63             }
64         }
65     }
66     array_push($res, sqrt($sdMhs), sqrt($sdSiswa));
67
68     return $res;
69 }
70
71 // menghitung kemiripan dengan perason
72 // $mahasiswa -> seluruh mahasiswa sesuai dengan jurusan SMA
73 // $siswa ->
74 public function calculatePearson($mahasiswa, $siswa)
75 {
76     $res = array();
77     foreach ($mahasiswa as $mhs) {
78         if ($mhs["id_program_studi"] == $siswa["id_program_studi"]) {
79             $covariance = $this->calculateCovariance($mhs, $siswa);
80             $sd = $this->calculateStandarDeviation($mhs, $siswa);
81             $sdMhs = $sd[0]; // standar deviasi untuk mahasiswa
82             $sdSiswa = $sd[1]; // standar deviasi untuk siswa
83
84             $idProdi = $mhs['id_program_studi'];
85             $IPK = $mhs['IPK'];
86
87             $sim = $covariance / ($sdMhs * $sdSiswa);
88
89             // atur threshold
90             if ($sim > 0) {
91                 // inisialisai array agar tidak null
92                 $res[$mhs['id_mahasiswa']] = array();
93                 array_push($res[$mhs['id_mahasiswa']], $sim, $idProdi, $IPK);
94             }
95         }
96     }
97     return $res;
98 }
99 }

```

Listing A.12: AccuracyController.php

```

1 <?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5
6 class AccuracyController extends Controller
7 {
8     public function calculateMAE($arr)
9     {
10         return array_sum($arr) / count($arr);
11     }
12
13     public function calculateRMSE($arr)
14     {
15         return sqrt(array_sum($arr) / count($arr));
16     }
17 }
18 }

```

Listing A.13: Fakultas.php

```

1 <?php
2 namespace App;
3
4 use Illuminate\Database\Eloquent\Model;
5
6

```

```

7 | class Fakultas extends Model
8 | {
9 |     protected $table = "fakultas";
10 |
11 |     protected $primaryKey = "id_fakultas";
12 |
13 |     protected $fillable=[
14 |         'id_fakultas',
15 |         'nama_fakultas'
16 |     ];
17 |
18 |     public function programStudi(){
19 |         return $this->hasMany('App\Program_Studi', 'id_fakultas', 'id_fakultas');
20 |     }
21 | }

```

Listing A.14: Jurusan\_SMA.php

```

1 | <?php
2 |
3 | namespace App;
4 |
5 | use Illuminate\Database\Eloquent\Model;
6 |
7 | class Jurusan_SMA extends Model
8 | {
9 |     protected $table = "jurusan_sma";
10 |     protected $primaryKey = "id_jurusan";
11 |
12 |     protected $fillable = [
13 |         'id_jurusan',
14 |         'nama_jurusan'
15 |     ];
16 |
17 |     public function mahasiswa()
18 |     {
19 |         return $this->hasMany('App\Mahasiswa', 'id_jurusan', 'id_jurusan');
20 |     }
21 |
22 |     public function jurusanSMA()
23 |     {
24 |         return $this->hasMany('App\Program_Studi', 'id_jurusan', 'id_jurusan');
25 |     }
26 | }

```

Listing A.15: Mahasiswa.php

```

1 | <?php
2 |
3 | namespace App;
4 |
5 | use Illuminate\Database\Eloquent\Model;
6 |
7 | class Mahasiswa extends Model
8 | {
9 |     protected $table = "mahasiswa";
10 |     protected $primaryKey = "id_mahasiswa";
11 |
12 |     protected $fillable = [
13 |         'id_mahasiswa',
14 |         'NPM',
15 |         'IPK',
16 |         'id_jurusan',
17 |         'id_program_studi'
18 |     ];
19 |
20 |     public function jurusanSMA()
21 |     {
22 |         return $this->belongsTo('App\Jurusan_SMA', 'id_jurusan', 'id_jurusan');
23 |     }
24 |
25 |     public function programStudi(){
26 |         return $this->belongsTo('App\Program_Studi', 'id_program_studi', 'id_program_studi');
27 |     }
28 |
29 |     public function nilai(){
30 |         return $this->hasMany('App\Nilai', 'id_mahasiswa', 'id_mahasiswa');
31 |     }
32 | }

```

Listing A.16: Mata\_Pelajaran.php

```

1 | <?php
2 |
3 | namespace App;
4 |
5 | use Illuminate\Database\Eloquent\Model;
6 |
7 | class Mata_Pelajaran extends Model
8 | {
9 |     protected $table = "mata_pelajaran";
10 |     protected $primaryKey = "id_mata_pelajaran";
11 |
12 |     protected $fillable = [
13 |         'id_mata_pelajaran',

```

```

14         'nama_mata_pelajaran'
15     ];
16
17     public function nilai(){
18         return $this->hasMany('App\Nilai', 'id_mata_pelajaran', 'id_mata_pelajaran');
19     }
20 }

```

Listing A.17: Nilai.php

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Nilai extends Model
8 {
9     protected $table = "nilai";
10    protected $primaryKey = "id_nilai";
11
12    protected $fillable=[
13        'id_nilai',
14        'id_mata_pelajaran',
15        'id_mahasiswa',
16        '101',
17        '102',
18        '111',
19        '112',
20        'AVG'
21    ];
22
23    public function mahasiswa(){
24        return $this->belongsTo('App\Nilai', 'id_mahasiswa', 'id_mahasiswa');
25    }
26
27    public function mataPelajaran(){
28        return $this->belongsTo('App\Mata_Pelajaran', 'id_mata_pelajaran', 'id_mata_pelajaran');
29    }
30 }

```

Listing A.18: Program\_Studi.php

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Program_Studi extends Model
8 {
9     protected $table = "program_studi";
10    protected $primaryKey = "id_program_studi";
11
12    protected $fillable=[
13        'id_program_studi',
14        'nama_program_studi',
15        'id_fakultas',
16        'id_jurusan'
17    ];
18
19    public function mahasiswa(){
20        return $this->hasMany('App\Mahasiswa', 'id_program_studi', 'id_program_studi');
21    }
22
23    public function fakultas(){
24        return $this->belongsTo('App\Fakultas', 'id_fakultas', 'id_fakultas');
25    }
26
27    public function jurusanSMA(){
28        return $this->belongsTo('App\Jurusan_SMA', 'id_jurusan', 'id_jurusan');
29    }
30 }

```

Listing A.19: header.blade.php

```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5     <!-- Required meta tags -->
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, _initial-scale=1, _shrink-to-fit=no">
8
9     <!-- Bootstrap CSS -->
10    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
        Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
11
12    <title>@yield('title')</title>
13
14    <style>
15        table,
16        th,
17        td {
18            /* border: solid black; */

```





```

33         <button class="btn_btn-info" onclick="window.location='/ips'">
34             IPS
35         </button>
36     </h5>
37 </div>
38 </td>
39 </tr>
40 </table>
41
42 <br>
43 <br>
44 </div>
45 @endsection

```

Listing A.21: ipa.blade.php

```

1  @extends('layout.header')
2
3  @section('title', 'Sistem_Rekomendasi_UNPAR')
4
5  @section('container')
6
7      <br>
8      <br>
9
10     <h1 align="center">Silakan mengisi nilai sesuai dengan nilai rapor</h1>
11     <br>
12
13     <div class="card_bg-light_border-0_">
14         <form action="/result" method="post">
15             @csrf
16             <table align="center" class="table_table-striped_table-bordered_text-center_w-75">
17                 <tr>
18                     <th rowspan="2" style="vertical-align:_middle;">Mata Pelajaran</th>
19                     <th colspan="2">Kelas X</th>
20                     <th colspan="2">Kelas XI</th>
21                 </tr>
22                 <tr>
23                     <th>Semester 1</th>
24                     <th>Semester 2</th>
25                     <th>Semester 1</th>
26                     <th>Semester 2</th>
27                 </tr>
28                 <tr>
29                     <th>Matematika</th>
30                     <td>
31                         <input type="number" name="mtk101" min="0" max="100" step="any" required><br>
32                     </td>
33                     <td>
34                         <input type="number" name="mtk102" min="0" max="100" step="any" required><br>
35                     </td>
36                     <td>
37                         <input type="number" name="mtk111" min="0" max="100" step="any" required><br>
38                     </td>
39                     <td>
40                         <input type="number" name="mtk112" min="0" max="100" step="any" required><br>
41                     </td>
42                 </tr>
43                 <tr>
44                     <th>Indonesia</th>
45                     <td>
46                         <input type="number" name="ind101" min="0" max="100" step="any" required><br>
47                     </td>
48                     <td>
49                         <input type="number" name="ind102" min="0" max="100" step="any" required><br>
50                     </td>
51                     <td>
52                         <input type="number" name="ind111" min="0" max="100" step="any" required><br>
53                     </td>
54                     <td>
55                         <input type="number" name="ind112" min="0" max="100" step="any" required><br>
56                     </td>
57                 </tr>
58                 <tr>
59                     <th>Inggris</th>
60                     <td>
61                         <input type="number" name="ing101" min="0" max="100" step="any" required><br>
62                     </td>
63                     <td>
64                         <input type="number" name="ing102" min="0" max="100" step="any" required><br>
65                     </td>
66                     <td>
67                         <input type="number" name="ing111" min="0" max="100" step="any" required><br>
68                     </td>
69                     <td>
70                         <input type="number" name="ing112" min="0" max="100" step="any" required><br>
71                     </td>
72                 </tr>
73                 <tr>
74                     <th>Fisika</th>
75                     <td>
76                         <input type="number" name="fsk101" min="0" max="100" step="any" required><br>
77                     </td>
78                     <td>
79                         <input type="number" name="fsk102" min="0" max="100" step="any" required><br>
80                     </td>
81                     <td>
82                         <input type="number" name="fsk111" min="0" max="100" step="any" required><br>

```

```

83         </td>
84     <td>
85         <input type="number" name="fsk112" min="0" max="100" step="any" required><br>
86     </td>
87 </tr>
88 <tr>
89     <th>Kimia</th>
90     <td>
91         <input type="number" name="kma101" min="0" max="100" step="any" required><br>
92     </td>
93     <td>
94         <input type="number" name="kma102" min="0" max="100" step="any" required><br>
95     </td>
96     <td>
97         <input type="number" name="kma111" min="0" max="100" step="any" required><br>
98     </td>
99     <td>
100         <input type="number" name="kma112" min="0" max="100" step="any" required><br>
101     </td>
102 </tr>
103 </table>
104 <div class="bottom_split_text-right">
105     <input type="submit" value="Submit" name="btnIPA" class="btn_bg-success">
106 </div>
107 </form>
108 </div>
109
110 <br>
111 <br>
112 <br>
113
114 @endsection

```

Listing A.22: ips.blade.php

```

1 @extends('layout.header')
2
3 @section('title', 'Sistem_Rekomendasi_UNPAR')
4
5 @section('container')
6
7 <br>
8 <br>
9
10 <h1 align="center">Silakan mengisi nilai sesuai dengan nilai rapor</h1>
11 <br>
12
13 <div class="card_bg-light_border-0">
14     <form action="/result" method="post">
15         @csrf
16         <table align="center" class="table_table-striped_table-bordered_text-center_w-75">
17             <tr>
18                 <th rowspan="2" style="vertical-align:_middle;">Mata Pelajaran</th>
19                 <th colspan="2">Kelas X</th>
20                 <th colspan="2">Kelas XI</th>
21             </tr>
22             <tr>
23                 <th>Semester 1</th>
24                 <th>Semester 2</th>
25                 <th>Semester 1</th>
26                 <th>Semester 2</th>
27             </tr>
28             <tr>
29                 <th>Matematika</th>
30                 <td>
31                     <input type="number" name="mtk101" min="0" max="100" step="any" required><br>
32                 </td>
33                 <td>
34                     <input type="number" name="mtk102" min="0" max="100" step="any" required><br>
35                 </td>
36                 <td>
37                     <input type="number" name="mtk111" min="0" max="100" step="any" required><br>
38                 </td>
39                 <td>
40                     <input type="number" name="mtk112" min="0" max="100" step="any" required><br>
41                 </td>
42             </tr>
43             <tr>
44                 <th>Indonesia</th>
45                 <td>
46                     <input type="number" name="ind101" min="0" max="100" step="any" required><br>
47                 </td>
48                 <td>
49                     <input type="number" name="ind102" min="0" max="100" step="any" required><br>
50                 </td>
51                 <td>
52                     <input type="number" name="ind111" min="0" max="100" step="any" required><br>
53                 </td>
54                 <td>
55                     <input type="number" name="ind112" min="0" max="100" step="any" required><br>
56                 </td>
57             </tr>
58             <tr>
59                 <th>Inggris</th>
60                 <td>
61                     <input type="number" name="ing101" min="0" max="100" step="any" required><br>
62                 </td>
63                 <td>

```

```

64         <input type="number" name="ing102" min="0" max="100" step="any" required><br>
65     </td>
66 <td>
67     <input type="number" name="ing111" min="0" max="100" step="any" required><br>
68 </td>
69 <td>
70     <input type="number" name="ing112" min="0" max="100" step="any" required><br>
71 </td>
72 </tr>
73 <tr>
74     <th>PKN</th>
75     <td>
76         <input type="number" name="pkn101" min="0" max="100" step="any" required><br>
77     </td>
78     <td>
79         <input type="number" name="pkn102" min="0" max="100" step="any" required><br>
80     </td>
81     <td>
82         <input type="number" name="pkn111" min="0" max="100" step="any" required><br>
83     </td>
84     <td>
85         <input type="number" name="pkn112" min="0" max="100" step="any" required><br>
86     </td>
87 </tr>
88 </table>
89 <div class="bottom_split_text-right">
90     <input type="submit" value="Submit" name="btnIPS" class="btn_bg-success">
91 </div>
92 </form>
93 </div>
94
95
96 <br>
97 <br>
98 <br>
99
100 @endsection

```

Listing A.23: result.blade.php

```

1  @extends('layout.header')
2
3  @section('title', 'Sistem_Rekomendasi_UNPAR')
4
5  @section('container')
6  <!-- bg-light untuk container -->
7  <br>
8  <br>
9
10 <h1 align="center">Berikut merupakan hasil perhitungan prediksi IPK</h1>
11 <br>
12
13 <div style="width: 75%; margin: auto;" class="container">
14     <table align="center" class="table_table-striped_table-bordered">
15         <tr>
16             <th style="width: 5%">No</th>
17             <th>Fakultas</th>
18             <th>Program Studi</th>
19             <th>IPK</th>
20         </tr>
21         <?php
22             $i = 1;
23             // print_r($predict);
24             foreach ($result as $id_prodi => $value) {
25                 echo "<tr>";
26                 echo "<td>" . $i . "</td>";
27                 echo "<td>" . $value[1] . "</td>";
28                 echo "<td>" . $value[2] . "</td>";
29                 echo "<td>" . number_format($value[0], 2) . "</td>";
30                 echo "</tr>";
31                 $i++;
32             }
33         ?>
34     </table>
35 </div>
36
37
38 <br>
39 <br>
40 <br>
41
42 @endsection

```

Listing A.24: pengujian.blade.php

```

1  @extends('layout.header')
2
3  @section('title', 'Sistem_Rekomendasi_UNPAR')
4
5  @section('container')
6  <!-- bg-light untuk container -->
7  <br>
8  <br>
9
10 <div id="accordion" class="bg-light">
11     <br>
12     <form action="" method="post">

```

```

13 | @csrf
14 | <table align="center">
15 |   <tr>
16 |     <td colspan="2">
17 |       <h2 class="text-center"> Jurusan Saat SMA dan Metode Pengujian: </h2>
18 |     </td>
19 |   </tr>
20 |   <tr>
21 |     <td>
22 |       <div class="card">
23 |         <div class="card-header" id="headingOne">
24 |           <h5 class="text-center">
25 |             <input type="submit" value="IPA_Basic" name="btn" class="btn btn-info">
26 |           </h5>
27 |         </div>
28 |       </td>
29 |     <td>
30 |       <div class="card-header" id="headingTwo">
31 |         <h5 class="text-center">
32 |           <input type="submit" value="IPS_Basic" name="btn" class="btn btn-info">
33 |         </h5>
34 |       </div>
35 |     </td>
36 |   </tr>
37 |   <tr>
38 |     <td>
39 |       <div class="card">
40 |         <div class="card-header" id="headingOne">
41 |           <h5 class="text-center">
42 |             <input type="submit" value="IPA_Kmeans" name="btn" class="btn btn-info">
43 |           </h5>
44 |         </div>
45 |       </td>
46 |     <td>
47 |       <div class="card-header" id="headingTwo">
48 |         <h5 class="text-center">
49 |           <input type="submit" value="IPS_Kmeans" name="btn" class="btn btn-info">
50 |         </h5>
51 |       </div>
52 |     </td>
53 |   </tr>
54 | </table>
55 | </form>
56 | </div>
57 |
58 | <br>
59 | <br>
60 |
61 | <div style="width: 75%; margin: auto;" class="container">
62 |   <table align="center" class="table table-striped table-bordered">
63 |     <?php
64 |     $status ?? '';
65 |     if ($status ?? '') {
66 |       if ($metode == 'Basic') {
67 |         // UNTUK AKURASI
68 |         echo "<tr>";
69 |         echo "<th style=width: 5%>No</th>";
70 |         echo "<th>_MAE_</th>";
71 |         echo "<th>_RMSE_</th>";
72 |         echo "<th>_Time_(Mic_Sec)_</th>";
73 |         echo "</tr>";
74 |
75 |         echo "<tr>";
76 |         echo "<td>" . 1 . "</td>";
77 |         echo "<td>" . $mae . "</td>";
78 |         echo "<td>" . $rmse . "</td>";
79 |         echo "<td>" . $times . "</td>";
80 |         echo "</tr>";
81 |
82 |         // UNTUK BAGIAN DETAIL
83 |         // print("MEAN ABSOLUTE ERROR = " . $mae);
84 |         // echo "<br>";
85 |         // echo "<br>";
86 |
87 |         // print("ROOT MEAN SQUARE ERROR = " . $rmse);
88 |         // echo "<br>";
89 |         // echo "<br>";
90 |
91 |         // print("EXECUTION TIME = " . $times);
92 |         // echo "<br>";
93 |         // echo "<br>";
94 |
95 |         // echo "<tr>";
96 |         // echo "<th style=width: 5%>No</th>";
97 |         // echo "<th>NPM</th>";
98 |         // echo "<th>Program Studi</th>";
99 |         // echo "<th>IPK</th>";
100 |         // echo "<th>Prediksi IPK</th>";
101 |         // echo "<th>Error MAE</th>";
102 |         // echo "<th>Error RMSE</th>";
103 |         // echo "</tr>";
104 |
105 |         // $i = 1;
106 |         // foreach ($result as $res) {
107 |         //   echo "<tr>";
108 |         //   echo "<td>" . $i . "</td>";
109 |         //   echo "<td>" . $res[0] . "</td>";
110 |         //   echo "<td>" . $res[1] . "</td>";
111 |         //   echo "<td>" . $res[2] . "</td>";
112 |         //   echo "<td>" . $res[3] . "</td>";
113 |         //   echo "<td>" . $res[4] . "</td>";
114 |         //   echo "<td>" . $res[5] . "</td>";
115 |         //   echo "<td>" . $res[6] . "</td>";
116 |         //   echo "<td>" . $res[7] . "</td>";
117 |         //   echo "<td>" . $res[8] . "</td>";
118 |         //   echo "<td>" . $res[9] . "</td>";
119 |         //   echo "</tr>";
120 |         //   $i++;
121 |         // }
122 |       }
123 |     }
124 |   </table>
125 | </div>

```

```

112         //      echo "<td>" . $res[1] . "</td>";
113         //      echo "<td>" . $res[2] . "</td>";
114         //      echo "<td>" . $res[3] . "</td>";
115         //      echo "<td>" . $res[4] . "</td>";
116         //      echo "<td>" . $res[5] . "</td>";
117         //      echo "</tr>";
118         //      $i++;
119         //    }
120     } else if ($metode == 'Kmeans') {
121         echo ("*");
122         echo ("Kolom_dengan_warna_hijau_berarti_nilai_minimum_dari_kolom");
123         echo "<tr>";
124         echo "<th style=width: 5%>K</th>";
125         echo "<th>_MAE_</th>";
126         echo "<th>_RMSE_</th>";
127         echo "<th>_Time_(Mic_Sec)_</th>";
128         echo "</tr>";
129
130         foreach ($result as $key => $value) {
131             echo "<tr>";
132             echo "<td>" . $key . "</td>";
133
134             if ($value[0] == $minMae) {
135                 echo "<td_bgcolor='#00FF00'>" . $value[0] . "</td>";
136             } else {
137                 echo "<td>" . $value[0] . "</td>";
138             }
139
140             if ($value[1] == $minRmse) {
141                 echo "<td_bgcolor='#00FF00'>" . $value[1] . "</td>";
142             } else {
143                 echo "<td>" . $value[1] . "</td>";
144             }
145
146             if ($value[2] == $minTime) {
147                 echo "<td_bgcolor='#00FF00'>" . $value[2] . "</td>";
148             } else {
149                 echo "<td>" . $value[2] . "</td>";
150             }
151             echo "</tr>";
152         }
153     }
154 }
155 }
156 ?>
157 </table>
158 </div>
159
160 <br>
161 <br>
162 @endsection

```

Listing A.25: 2020\_02\_07\_141228\_create\_jurusan\_sma\_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateJurusanSMATable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('jurusan_sma', function (Blueprint $table) {
17             $table->increments('id_jurusan');
18             $table->string('nama_jurusan',25);
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27     public function down()
28     {
29         Schema::dropIfExists('jurusan_sma');
30     }
31 }

```

Listing A.26: 2020\_02\_07\_141432\_create\_mata\_pelajaran\_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateMataPelajaranTable extends Migration
8 {
9     /**

```

```

10 |     * Run the migrations.
11 |     *
12 |     * @return void
13 |     */
14 | public function up()
15 | {
16 |     Schema::create('mata_pelajaran', function (Blueprint $table) {
17 |         $table->increments('id_mata_pelajaran');
18 |         $table->string('nama_mata_pelajaran', 20);
19 |     });
20 | }
21 |
22 | /**
23 |  * Reverse the migrations.
24 |  *
25 |  * @return void
26 |  */
27 | public function down()
28 | {
29 |     Schema::dropIfExists('mata_pelajaran');
30 | }
31 | }

```

Listing A.27: 2020\_02\_07\_142008\_create\_fakultas\_table.php

```

1 | <?php
2 |
3 | use Illuminate\Database\Migrations\Migration;
4 | use Illuminate\Database\Schema\Blueprint;
5 | use Illuminate\Support\Facades\Schema;
6 |
7 | class CreateFakultasTable extends Migration
8 | {
9 |     /**
10 |      * Run the migrations.
11 |      *
12 |      * @return void
13 |      */
14 | public function up()
15 | {
16 |     Schema::create('fakultas', function (Blueprint $table) {
17 |         $table->increments('id_fakultas');
18 |         $table->string('nama_fakultas', 50);
19 |     });
20 | }
21 |
22 | /**
23 |  * Reverse the migrations.
24 |  *
25 |  * @return void
26 |  */
27 | public function down()
28 | {
29 |     Schema::dropIfExists('fakultas');
30 | }
31 | }

```

Listing A.28: 2020\_02\_07\_143407\_create\_program\_studi\_table.php

```

1 | <?php
2 |
3 | use Illuminate\Database\Migrations\Migration;
4 | use Illuminate\Database\Schema\Blueprint;
5 | use Illuminate\Support\Facades\Schema;
6 |
7 | class CreateProgramStudiTable extends Migration
8 | {
9 |     /**
10 |      * Run the migrations.
11 |      *
12 |      * @return void
13 |      */
14 | public function up()
15 | {
16 |     Schema::create('program_studi', function (Blueprint $table) {
17 |         $table->increments('id_program_studi');
18 |         $table->string('nama_program_studi', 50);
19 |
20 |         $table->integer('id_fakultas')->unsigned();
21 |         $table->foreign('id_fakultas')->references('id_fakultas')->on('fakultas');
22 |
23 |         $table->integer('id_jurusan')->unsigned();
24 |         $table->foreign('id_jurusan')->references('id_jurusan')->on('jurusan_sma');
25 |     });
26 | }
27 |
28 | /**
29 |  * Reverse the migrations.
30 |  *
31 |  * @return void
32 |  */
33 | public function down()
34 | {
35 |     Schema::dropIfExists('program_studi');
36 | }
37 | }

```

Listing A.29: 2020\_02\_07\_145514\_create\_mahasiswa\_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateMahasiswaTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('mahasiswa', function (Blueprint $table) {
17             $table->increments('id_mahasiswa');
18             $table->string('NPM', 10);
19             $table->double('IPK', 3, 2);
20             $table->integer('id_jurusan')->unsigned();
21             $table->foreign('id_jurusan')->references('id_jurusan')->on('jurusan_sma');
22             $table->integer('id_program_studi')->unsigned();
23             $table->foreign('id_program_studi')->references('id_program_studi')->on('program_studi');
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('mahasiswa');
35     }
36 }

```

Listing A.30: 2020\_02\_07\_145515\_create\_nilai\_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateNilaiTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('nilai', function (Blueprint $table) {
17             $table->increments('id_nilai');
18             $table->integer('id_mata_pelajaran')->unsigned();
19             $table->foreign('id_mata_pelajaran')->references('id_mata_pelajaran')->on('mata_pelajaran');
20             $table->integer('id_mahasiswa')->unsigned();
21             $table->foreign('id_mahasiswa')->references('id_mahasiswa')->on('mahasiswa');
22             $table->double('101', 5, 2);
23             $table->double('102', 5, 2);
24             $table->double('111', 5, 2);
25             $table->double('112', 5, 2);
26             $table->double('AVG', 5, 2);
27         });
28     }
29
30     /**
31      * Reverse the migrations.
32      *
33      * @return void
34      */
35     public function down()
36     {
37         Schema::dropIfExists('nilai');
38     }
39 }

```

Listing A.31: DatabaseSeeder.php

```

1 <?php
2
3 use Illuminate\Database\Seeder;
4
5 class DatabaseSeeder extends Seeder
6 {
7     /**
8      * Seed the application's database.
9      *
10     * @return void
11     */
12     public function run()
13     {
14         $this->call(

```



```

15 |         [
16 |             Jurusan_SMA::class,
17 |             Fakultas::class,
18 |             Program_Studi::class,
19 |             Mata_Pelajaran::class
20 |         ]
21 |     );
22 | }
23 | }

```

Listing A.32: Fakultas.php

```

1 | <?php
2 |
3 | use Illuminate\Database\Seeder;
4 | use Illuminate\Support\Facades\DB;
5 |
6 | class Fakultas extends Seeder
7 | {
8 |     /**
9 |      * Run the database seeds.
10 |      *
11 |      * @return void
12 |      */
13 |
14 |     private $fakultas = array(
15 |         "Fakultas_Ekonomi", "Fakultas_Hukum", "Fakultas_Ilmu_Sosial_dan_Ilmu_Politik",
16 |         "Fakultas_Teknik", "Fakultas_Falsafah_dan_Peradaban", "Fakultas_Teknik_Industri",
17 |         "Fakultas_Teknologi_Informasi_dan_Sains"
18 |     );
19 |
20 |     public function run(){
21 |         for ($id = 1; $id <= 7; $id++) {
22 |             DB::table("fakultas")->insert([
23 |                 "id_fakultas" => $id,
24 |                 "nama_fakultas" => $this->fakultas[$id-1]
25 |             ]);
26 |         }
27 |     }
28 | }

```

Listing A.33: Jurusan\_SMA.php

```

1 | <?php
2 |
3 | use Illuminate\Database\Seeder;
4 | use Illuminate\Support\Facades\DB;
5 |
6 | class Jurusan_SMA extends Seeder
7 | {
8 |     /**
9 |      * Run the database seeds.
10 |      *
11 |      * @return void
12 |      */
13 |
14 |     private $jurusan = array("Ilmu_Pengatahuan_Alam", "Ilmu_Pengatahuan_Sosial");
15 |
16 |     public function run(){
17 |         for ($id = 1; $id <= 2; $id++) {
18 |             DB::table("jurusan_sma")->insert([
19 |                 "id_jurusan" => $id,
20 |                 "nama_jurusan" => $this->jurusan[$id-1]
21 |             ]);
22 |         }
23 |     }
24 | }

```

Listing A.34: Jurusan\_SMA.php

```

1 | <?php
2 |
3 | use Illuminate\Database\Seeder;
4 | use Illuminate\Support\Facades\DB;
5 |
6 | class Jurusan_SMA extends Seeder
7 | {
8 |     /**
9 |      * Run the database seeds.
10 |      *
11 |      * @return void
12 |      */
13 |
14 |     private $jurusan = array("Ilmu_Pengatahuan_Alam", "Ilmu_Pengatahuan_Sosial");
15 |
16 |     public function run(){
17 |         for ($id = 1; $id <= 2; $id++) {
18 |             DB::table("jurusan_sma")->insert([
19 |                 "id_jurusan" => $id,
20 |                 "nama_jurusan" => $this->jurusan[$id-1]
21 |             ]);
22 |         }
23 |     }
24 | }

```

Listing A.35: Jurusan\_SMA.php

```
1 <?php
2
3 use Illuminate\Database\Seeder;
4 use Illuminate\Support\Facades\DB;
5
6 class Jurusan_SMA extends Seeder
7 {
8     /**
9      * Run the database seeds.
10     *
11     * @return void
12     */
13
14     private $jurusan = array("Ilmu_Pengatahuan_Alam", "Ilmu_Pengatahuan_Sosial");
15     public function run(){
16         for ($id = 1; $id <= 2; $id++) {
17             DB::table("jurusan_sma")->insert([
18                 "id_jurusan" => $id,
19                 "nama_jurusan" => $this->jurusan[$id-1]
20             ]);
21         }
22     }
23 }
```

## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4