

**SISTEM REKOMENDASI BUKU PERPUSTAKAAN
SEKOLAH MENGGUNAKAN METODE
*CONTENT-BASED FILTERING***

**Skripsi
untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi
Teknik Komputer**



**RYKY ARDIANSYAH
1903010002**

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS SAINS, TEKNOLOGI DAN PENDIDIKAN
UNIVERSITAS MUHAMMADIYAH LAMONGAN**

2023

HALAMAN PENGESAHAN

SKRIPSI

Sistem Rekomendasi Buku Perpustakaan Sekolah Menggunakan Metode *Content-Based Filtering*

Oleh:

RYKY ARDIANSYAH
1903010002

Telah diujikan dan dinyatakan lulus ujian skripsi pada tanggal 12 Juni 2023 oleh tim penguji Program Studi Teknik Komputer, Fakultas Sains, Teknologi, dan Pendidikan Universitas Muhammadiyah Lamongan.

Lamongan, 8 Agustus 2023
Mengetahui,

Pembimbing I



Bagus Dwi Saputra, S.Kom., M.Kom
NIDN. 0724099402

Penguji I



Eko Handoyo, S.Kom., M.Kom
NIDN. 0717029104

Pembimbing II



Mufti Ari Bianto, S.Kom., M.Kom
NIDN. 0710069501

**Ketua Program Studi
Teknik Komputer**



Mufti Ari Bianto, S.Kom., M.Kom
NIDN. 0710069501

**Mengetahui,
Dekan Fakultas Sains
Teknologi dan Pendidikan**



Eko Handoyo, S.Kom., M.Kom
NIDN. 0717029104

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar akademik di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebut dalam daftar pustaka.

Lamongan, 8 Agustus 2023



Ryky Ardiansyah

PERNYATAAN PERSETUJUAN

PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademis Universitas Muhammadiyah Lamongan, saya yang bertanda tangan dibawah ini :

Nama : Ryky Ardiansyah
NIM : 1903010002
Program Studi : S1 Teknik Komputer
Fakultas : Sains, Teknologi dan Pendidikan
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Lamongan Hak bebas Royalti Noneksklusif atas karya ilmiah saya yang berjudul :

Sistem Rekomendasi Buku Perpustakaan Sekolah Menggunakan Metode *Content-Based Filtering*

Beserta perangkat yang ada. Dengan Hak bebas Royalti Noneksklusif ini Program Studi Teknik Komputer Fakultas Sains, Teknologi dan Pendidikan Universitas Muhammadiyah Lamongan berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*) merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Dibuat di : Lamongan
Pada Tanggal : 8 Agustus 2023
Yang menyatakan



Ryky Ardiansyah
1903010002

KATA PENGANTAR

Alhamdulillah puji syukur kepada Allah SWT yang telah memberikan kemudahan dalam menyelesaikan skripsi ini. Sholawat dan salam dilimpahkan kepada junjungan kita Nabi Muhammad SAW dan semoga kita semua menerima syafaatnya di hari akhir, Aamiin.

Dalam setiap langkah saya berusaha semaksimal mungkin untuk mewujudkan harapan-harapan yang saya impikan sebagai ungkapan terima kasih, saya ingin mempersembahkan skripsi ini untuk :

1. Bapak Eko Handoyo, S.Kom., M.Kom, sebagai Dekan Fakultas Sains, Teknologi, dan Pendidikan Universitas Muhammadiyah Lamongan yang telah memberikan izin dalam proses penulisan skripsi ini.
2. Bapak Mufti Ari Bianto, S.Kom., M.Kom, sebagai Ketua Jurusan Teknik Komputer Universitas Muhammadiyah Lamongan, telah memberikan persetujuan dan dukungan yang berarti dalam proses penulisan skripsi ini.
3. Bapak Bagus Dwi Saputra, S.Kom., M.Kom dan Bapak Mufti Ari Bianto, S.Kom., M.Kom selaku Dosen Pembimbing 1 dan Dosen Pembimbing 2 yang telah memberikan arahan, bimbingan, dan masukan yang berharga dalam penulisan skripsi ini.
4. Orang Tua yang selalu memberikan dukungan materi, moral, dan doa yang tak putus untuk penulis. Terima kasih untuk semua pengorbanan, perhatian, dan kasih sayang yang diberikan selama ini.
5. Seluruh dosen Program Studi Teknik Komputer yang telah memberikan inspirasi dan juga motivasi kepada penulis.
6. Bapak Eko Cahyono, SE selaku kepala sekolah, seluruh guru dan karyawan SMA Muhammadiyah 8 Sukodadi yang telah mendukung kelancaran penelitian ini.
7. Teman-Teman Angkatan 1 Teknik Komputer Universitas Muhammadiyah Lamongan, yang telah memberikan dukungan, semangat, dan inspirasi selama penulisan skripsi ini. Terima kasih untuk kebersamaan, kegembiraan, dan persahabatan yang telah terjalin selama ini.

DAFTAR ISI

	Halaman
Halaman Judul.....	i
Halaman Pengesahan	ii
Halaman Pernyataan.....	iii
Halaman Pernyataan Publikasi.....	iv
Kata Pengantar	v
Daftar Isi.....	vi
Daftar Gambar.....	viii
Daftar Tabel	ix
Daftar Arti Lambang dan Singkatan	x
Daftar Lampiran	xi
Abstrak	xii
<i>Abstract</i>	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1. Tinjauan Pustaka	5
2.2. Dasar Teori.....	6
2.2.1. Sistem Informasi	6
2.2.2. Sistem Pendukung Keputusan (SPK).....	7
2.2.3. Sistem Rekomendasi	7
2.2.4. Buku	8
2.2.5. <i>Content-Based Filtering</i>	8
2.2.6. <i>Preprocessing</i>	8
2.2.7. Perpustakaan Sekolah.....	12
2.2.8. <i>Hypertext Preprocessor (PHP)</i>	12
2.2.9. <i>Sastrawi</i>	12
2.2.10. <i>HyperText Markup Language (HTML)</i>	13
2.2.11. <i>Cascading Stylesheet (CSS)</i>	13
2.2.12. <i>Entity Relationship Diagram (ERD)</i>	14
2.2.13. <i>Use Case Diagram</i>	15
2.2.14. <i>Term Frequency-Inverse Document Frequency (TF-IDF)</i>	15
2.2.15. <i>Cosine Similarity</i>	17
BAB III METODOLOGI PENELITIAN.....	19
3.1. Bahan dan Alat Penelitian.....	19
3.1.1. Perangkat keras	19
3.1.2. Perangkat lunak	19

3.2. Alur Penelitian	19
3.2.1. Analisa Kebutuhan	20
3.2.2. Pengumpulan Data	21
3.2.3. Teknik Pengumpulan Data	21
3.3. Rancangan <i>Database</i>	21
3.3.1. Perancangan <i>Database</i>	21
3.3.2. <i>Conceptual Data Model (CDM)</i>	22
3.3.3. <i>Physical Data Model (PDM)</i>	23
3.4. Rancangan Sistem	24
3.4.1. <i>Flowchart</i> Sistem Rekomendasi.....	24
3.4.2. <i>Use Case Diagram</i>	25
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	27
4.1. Hasil Penelitian	27
4.1.1. Profil sekolah.....	27
4.1.2. Perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi	28
4.1.3. Data buku Perpustakaan SMA Muhammadiyah 8 Sukodadi	28
4.2. Pembahasan.....	29
4.2.1. Perancangan Sistem.....	29
4.2.2. Perancangan <i>Database</i>	36
4.2.3. Implementasi Antarmuka Sistem	38
4.2.4. Pengujian Sistem	45
4.2.5. Analisa Metode	46
A. <i>Preprocessing</i>	46
B. Pembobotan <i>TF-IDF</i>	51
C. Pembobotan <i>cosine similarity</i>	53
D. Perhitungan skor <i>cosine similarity</i>	53
BAB V PENUTUP.....	55
5.1. Kesimpulan	55
5.2. Saran	55
DAFTAR PUSTAKA	56
LAMPIRAN.....	59
Lampiran 1.	59
Lampiran 2.	60

DAFTAR GAMBAR

	Halaman
Gambar 2. 1. Tahapan <i>Preprocessing</i>	9
Gambar 3. 1. Metode <i>waterfall</i> yang digunakan.....	20
Gambar 3. 2. Rancangan <i>conceptual data model</i>	22
Gambar 3. 3. Rancangan <i>physical data model</i>	23
Gambar 3. 4. <i>Flowchart</i> sistem rekomendasi	24
Gambar 3. 5. Rancangan <i>use case</i>	25
Gambar 4. 1. Foto bangunan sekolah SMA Muhammadiyah 8 Sukodadi.....	27
Gambar 4. 2. <i>Use case diagram</i>	29
Gambar 4. 3. <i>Activity diagram login</i>	31
Gambar 4. 4. <i>Activity diagram</i> kelola buku	32
Gambar 4. 5. <i>Activity diagram</i> peminjaman buku	33
Gambar 4. 6. <i>Activity diagram</i> pengembalian buku.....	34
Gambar 4. 7. <i>Activity diagram</i> siswa	35
Gambar 4. 8. <i>Entity relationship diagram</i> sistem	36
Gambar 4. 9. Tampilan halaman awal	38
Gambar 4. 10. Tampilan halaman daftar akun	39
Gambar 4. 11. Tampilan halaman <i>login</i>	40
Gambar 4. 12. Tampilan halaman pustakawan	41
Gambar 4. 13. Tampilan halaman guru.....	42
Gambar 4. 14. Tampilan halaman siswa	43
Gambar 4. 15. Tampilan detail buku.....	44

DAFTAR TABEL

	Halaman
Tabel 2. 1. Contoh implementasi <i>case folding</i>	9
Tabel 2. 2. Contoh implementasi <i>tokenizing</i>	10
Tabel 2. 3. Contoh implementasi <i>filtering</i>	11
Tabel 2. 4. Contoh implementasi <i>stemming</i>	11
Tabel 2. 4. Contoh implementasi <i>stemming</i> (lanjutan)	12
Tabel 2. 5. Komponen <i>ERD</i>	14
Tabel 2. 6. Komponen <i>use case diagram</i>	15
Tabel 3. 1. Deskripsi rancangan <i>use case</i>	25
Tabel 3. 1. Deskripsi rancangan <i>use case</i> (lanjutan)	26
Tabel 4. 1. Deskripsi <i>use case diagram</i>	29
Tabel 4. 1. Deskripsi <i>use case diagram</i> (lanjutan)	30
Tabel 4. 2. Hasil pengujian sistem	45
Tabel 4. 3. Data sampel buku yang digunakan	46
Tabel 4. 4. Proses <i>case folding</i>	47
Tabel 4. 5. Proses <i>tokenizing</i>	48
Tabel 4. 5. Proses <i>tokenizing</i> (lanjutan)	49
Tabel 4. 6. Proses <i>filtering</i>	49
Tabel 4. 6. Proses <i>filtering</i> (lanjutan)	50
Tabel 4. 7. Proses <i>steeming</i>	50
Tabel 4. 7. Proses <i>steeming</i> (lanjutan)	51
Tabel 4. 8. Pembobotan <i>TF-IDF</i>	51
Tabel 4. 8. Pembobotan <i>TF-IDF</i> (lanjutan)	52
Tabel 4. 9. Pembobotan <i>cosine similarity</i>	53
Tabel 4. 10. Perhitungan skor	54

DAFTAR ARTI LAMBANG DAN SINGKATAN

DAFTAR ARTI LAMBANG

Lambang	Arti Lambang
TF	<i>term frequency</i>
$tf_{i,j}$	kemunculan kata t_i dalam dokumen d_j .
idf_i	<i>inverse document frequency</i>
N	jumlah dokumen yang diambil oleh sistem
d_i	jumlah dokumen dalam koleksi
$Sim(q, d_j)$	seberapa mirip <i>query</i> dengan dokumen

DAFTAR SINGKATAN

Singkatan	Kepanjangan Singkatan
<i>Library</i>	kumpulan program yang digunakan untuk mengembangkan sistem.
<i>Preprocessing</i>	proses pembersihan data mentah menjadi data bersih yang dapat diolah.

DAFTAR LAMPIRAN

Lampiran 1. Surat keterangan melaksanakan penelitian.....	59
Lampiran 2. <i>Source code</i>	60

Sistem Rekomendasi Buku Perpustakaan Sekolah Menggunakan Metode *Content-Based Filtering*

ABSTRAK

Buku adalah sumber pengetahuan dan informasi yang dapat meningkatkan pemahaman siswa tentang berbagai topik. Siswa seringkali kesulitan dalam menemukan buku yang sesuai dengan preferensi yang diinginkan siswa karena kurangnya informasi mengenai berbagai jenis buku. Salah satu cara mengelola informasi ini adalah melalui penggunaan sistem rekomendasi. Sistem rekomendasi telah terbukti efektif dalam mengatasi jumlah informasi yang tersedia dan memberikan rekomendasi buku sesuai dengan keinginan pengguna. Penelitian ini bertujuan untuk membuat dan mengembangkan sistem yang dapat memberikan rekomendasi buku kepada siswa berdasarkan minat mereka dengan menggunakan bahasa pemrograman *PHP*. Data buku yang digunakan dalam penelitian ini terdiri dari 517 data buku yang berasal dari perpustakaan SMA Muhammadiyah 8 Sukodadi. Metode yang digunakan dalam penelitian ini adalah *content-based filtering*. Untuk melakukan pembobotan dan menghitung tingkat kemiripan antar data buku, peneliti menggunakan algoritma *TF-IDF* dan *cosine similarity* untuk mengukur tingkat kemiripan antara vektor A dan vektor B. Berdasarkan hasil pengujian sistem yang telah dibangun, sistem ini mampu memberikan rekomendasi berdasarkan tingkat kemiripan antar buku dengan menghasilkan nilai pembobotan *cosine similarity* sebesar 0,358. Angka tersebut menunjukkan bahwa perhitungan yang dilakukan oleh sistem berhasil dan memberikan rekomendasi yang sesuai dengan perhitungan skor menggunakan metode *cosine similarity*.

Kata kunci : Sistem rekomendasi, *Content based filtering*, *Cosine similarity*, *TF-IDF*.

School Library Book Recommendation System using Content-Based Filtering Method

ABSTRACT

The Book is a source of knowledge and information that can enhance students' understanding of various topics. Students often struggle to find books that match their preferences due to a lack of information about various types of books. One way to manage this information is through the use of a recommendation system. Recommendation systems have proven to be effective in dealing with the vast amount of available information and providing book recommendations based on user preferences. This research aims to create and develop a system that can provide book recommendations to students based on their interests using the PHP programming language. The book data used in this research consists of 517 book entries obtained from Muhammadiyah 8 Sukodadi High School library. The method employed in this research is content-based filtering. To perform weighting and calculate the similarity level between book data, the researcher utilized the TF-IDF algorithm and cosine similarity to measure the similarity between vector A and vector B. Based on the testing results of the constructed system, it is capable of providing recommendations based on the similarity level between books by producing a cosine similarity weighting score of 0.358. This value indicates that the system's calculations are successful in providing recommendations that align with the cosine similarity score calculation method.

Keyword : Recommendation system, Content based filtering, Cosine similarity, TF-IDF.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Membaca buku merupakan salah satu cara yang dapat dilakukan oleh semua orang untuk menambah ilmu pengetahuan dan wawasan baru mengenai banyak hal. Sekarang ini, informasi mengenai berbagai jenis buku telah tersedia dengan luas di internet, sehingga mudah diakses oleh banyak orang. Walaupun informasi mengenai berbagai jenis buku telah beredar di internet, minat baca di Indonesia masih tergolong rendah. Menurut data riset yang dilakukan oleh *UNESCO* menunjukkan bahwa minat baca masyarakat Indonesia sangat rendah. Hanya 0,001% dari total penduduk Indonesia yang rajin membaca yang berarti hanya 1 dari 1000 orang Indonesia yang memiliki kebiasaan membaca (Susanti dan Santi, 2019).

Hasil riset yang dilakukan menunjukkan bahwa minat baca di Indonesia masih tergolong rendah dibandingkan dengan negara lain, termasuk Singapura dan Malaysia (Susanti dan Santi, 2019). Hal ini tentu akan mempengaruhi pemahaman dalam ilmu pengetahuan dan teknologi. Selain itu, efek yang lebih besar dari keengganan untuk membaca pada generasi muda akan merugikan negara. Hal ini dapat menyebabkan Indonesia kehilangan aset penyumbang dari segi sumber daya manusia yang berkualitas dan produktif. Oleh karena itu, perlu ada upaya untuk meningkatkan minat baca di Indonesia agar dapat menghasilkan generasi yang berkualitas dan produktif (Witanto, 2018). Sampai saat ini, terdapat banyak media yang menawarkan layanan bacaan buku yang beragam kepada masyarakat luas, seperti Gramedia, Togamas, Berdikari, dan sejenisnya. Layanan tersebut memungkinkan masyarakat untuk mengakses berbagai jenis buku yang tersedia. Dalam rangka mempermudah pembaca buku dalam menemukan buku yang sesuai dengan preferensi mereka, penerapan sistem rekomendasi dapat dimanfaatkan karena banyaknya informasi buku yang tersebar.

Dengan demikian, pengguna dapat dengan mudah menemukan referensi baru mengenai buku yang sesuai dengan kategori buku yang disukai oleh pembaca sebelumnya. Selain itu, sistem rekomendasi juga memiliki peran yang penting dalam banyak aplikasi *online*. Performa sistem rekomendasi dapat memengaruhi kesuksesan komersial perusahaan dan kepuasan pengguna (Wei dkk., 2017)

Sistem rekomendasi terdiri dari tiga model yang dapat dipakai, yaitu *Content-Based Filtering*, *Collaborative Filtering*, dan *Hybrid Recommender System*. Kategori-kategori tersebut merupakan model yang dapat digunakan dalam sistem rekomendasi untuk memberikan rekomendasi yang tepat sesuai dengan kebutuhan pengguna. Sistem rekomendasi telah banyak digunakan dalam berbagai aspek kehidupan, salah satunya seperti pada penerapan sistem rekomendasi laptop yang menggunakan metode *content-based filtering* (Wijaya dan Alfian, 2018). Namun, metode tersebut belum terpakai pada perpustakaan. Oleh karena itu, sistem rekomendasi pada aplikasi perpustakaan akan dibuat sebagai otomasi perpustakaan yang bertujuan untuk memudahkan siswa dalam mencari data yang ada dengan mengoptimalkan fungsi metode *content-based filtering* di dalamnya. Ini akan membantu siswa dalam menemukan data yang relevan dan sesuai dengan kebutuhan mereka dengan lebih cepat dan mudah. Dengan demikian, sistem rekomendasi pada website perpustakaan akan menjadi sangat berguna bagi siswa dalam mencari dan menemukan informasi yang diperlukan.

Penerapan metode *content-based filtering* menggunakan konten yang terdapat di deskripsi setiap buku untuk mengukur tingkat kemiripannya dengan buku lain. Dengan mengukur kemiripan dari setiap buku, sistem dapat memberikan skor yang dapat menjadi acuan dalam memberikan rekomendasi sesuai dengan deskripsi dari buku tersebut. Melalui sistem rekomendasi ini, diharapkan siswa dapat menambah daftar referensi baru berdasarkan buku-buku yang telah diminati sebelumnya.

Berdasarkan masalah tersebut, penulis melakukan penelitian tentang sistem rekomendasi buku yang ada di perpustakaan SMA Muhammadiyah 8 Sukodadi. Hasil penelitian tersebut dituangkan dalam skripsi yang berjudul "**Sistem Rekomendasi Buku Perpustakaan Sekolah Menggunakan Metode *Content-***

Based Filtering". Penelitian ini diharapkan dapat memberikan solusi untuk meningkatkan efisiensi dan pelayanan di perpustakaan tersebut.

1.2. Rumusan Masalah

Masalah yang dirumuskan berdasarkan latar belakang diatas adalah :

1. Bagaimana merancang sebuah sistem manajemen dan sistem rekomendasi buku pada perpustakaan SMA Muhammadiyah 8 Sukodadi menggunakan metode *content-based filtering*?
2. Bagaimana merancang sebuah sistem yang dapat memudahkan pustakawan dalam mengelola transaksi peminjaman dan pengembalian buku secara efisien?
3. Bagaimana performa metode *content-based filtering* dalam memberikan rekomendasi buku berdasarkan tingkat kemiripan antar item berdasarkan pembobotan *TF-IDF* dan perhitungan *Cosine Similarity*?

1.3. Batasan Masalah

Dalam sistem ini, agar tidak menyimpang dari tujuan yang ingin dicapai maka pembahasan akan dibatasi sebagai berikut :

1. Membahas tentang perancangan sistem manajemen perpustakaan dan sistem rekomendasi buku.
2. Data yang digunakan pada sistem ini berasal dari data buku yang terdapat pada perpustakaan SMA Muhammadiyah 8 Sukodadi.
3. Data buku yang akan digunakan sebanyak 500 buku yang merupakan gabungan dari buku fisik dan buku digital (*e-book*)
4. Bahasa pemrograman yang dipakai dalam perancangan sistem ini adalah *Hypertext Preprocessor (PHP)* dan *JavaScript*.
5. *Database* yang digunakan adalah *MySQL* dan menggunakan *Laragon* sebagai *Local Server*.

1.4. Tujuan Penelitian

Tujuan dibangunnya sistem rekomendasi buku perpustakaan SMA Muhammadiyah 8 Sukodadi ini adalah sebagai berikut:

1. Merancang dan mengembangkan sebuah sistem manajemen dan sistem rekomendasi buku pada perpustakaan SMA Muhammadiyah 8 Sukodadi dengan menggunakan metode *content-based filtering*.
2. Merancang sebuah sistem yang dapat memudahkan pustakawan dalam mengelola transaksi peminjaman dan pengembalian buku.
3. Mengukur tingkat keakuratan sistem dalam memberikan saran buku menggunakan pembobotan *TF-IDF* dan perhitungan *Cosine Similarity* dari parameter yang diambil dari judul buku, penulis, kategori, dan deskripsi buku yang ada pada perpustakaan SMA Muhammadiyah 8 Sukodadi.

1.5. Manfaat Penelitian

Manfaat yang diharapkan penulis dari penelitian yang dilakukan adalah sebagai berikut :

1. Memudahkan siswa SMA Muhammadiyah 8 Sukodadi dalam mencari rekomendasi buku di perpustakaan sesuai dengan yang mereka cari.
2. Memperluas pengetahuan dan sumber referensi tentang buku yang memiliki kesamaan dengan buku yang diminati oleh siswa.
3. Memudahkan tugas pustakawan dalam mengelola transaksi peminjaman dan pengembalian buku. Meningkatkan referensi dan pengetahuan siswa mengenai buku yang memiliki kemiripan dengan buku yang diminati oleh siswa.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Sistem Rekomendasi Laptop Menggunakan *Collaborative Filtering* Dan *Content-Based Filtering* oleh Anderias Eko Wijaya, dkk. pada tahun 2018 ini membahas tentang kebutuhan akan laptop dan kesulitan dalam memilih atau membeli laptop yang sesuai dengan kebutuhan. Sistem rekomendasi dibutuhkan untuk membantu dalam mencari laptop yang sesuai berdasarkan ketertarikan dan kebutuhan. Penelitian ini menggabungkan metode *collaborative filtering* dan *content-based filtering* dengan teknik *mixed hybrid* dan diuji menggunakan metode *blackbox*, menunjukkan bahwa metode *content-based filtering* memiliki waktu eksekusi tercepat dibandingkan dengan metode *collaborative filtering* dan *mixed hybrid*, serta waktu eksekusi dipengaruhi oleh jumlah item.

Pengoptimalan Peran Perpustakaan Sekolah dalam Menumbuhkan Minat Baca Siswa di Sekolah Dasar oleh Adi Alpian dan Hikmatu Ruwaida pada tahun 2022 ini dilakukan untuk mengoptimalkan peran perpustakaan dalam menumbuhkan minat baca siswa di Sekolah Dasar Islam Terpadu Ihsanul Amal. Metode kualitatif dengan jenis *narrative research* digunakan untuk penelitian ini, dengan teknik pengumpulan data menggunakan teknik observasi dan wawancara. Hasil penelitian menunjukkan bahwa meskipun terdapat kendala seperti dana pembelian buku yang terbatas, ruangan yang belum terlalu luas, dan dekorasi perpustakaan yang belum diperbaharui, perpustakaan cukup berhasil dalam upaya menumbuhkan minat baca siswa dengan adanya program-program seperti *read aloud*, baca sahari salambar, dan bedah buku.

Sistem Rekomendasi *Product Emina Cosmetics* dengan Menggunakan Metode *Content-Based Filtering* oleh Fatoni Batari Agung Larasati, dkk pada tahun 2018 ini membuat aplikasi *content-based filtering* yang merekomendasikan produk emina berdasarkan produk yang telah dicari sebelumnya, dengan menggunakan algoritma *cosine similarity*. Produk diurutkan berdasarkan nilai similaritas tertinggi hingga terendah, dengan produk termirip memiliki nilai similaritas sebesar 0,7195.

Sistem Rekomendasi Produk Pena Eksklusif Menggunakan Metode *Content-Based Filtering* dan *TF-IDF* oleh Mariani Widia Putri, dkk pada tahun 2020 ini membangun sistem rekomendasi produk pena eksklusif menggunakan metode *Content-Based Filtering* dan *Term Frequency Inverse Document Frequency (TF-IDF)* untuk meningkatkan *Customer Relationship Management (CRM)* dan *brand awareness* pelanggan. Data yang digunakan terdiri dari 258 kode produk dengan 8 kategori dan 33 kata kunci, dan hasil perhitungan *TF-IDF* menunjukkan tingkat keakuratan sebesar 96,5% untuk memberikan rekomendasi produk terbaik pertama. Sistem rekomendasi ini bertujuan untuk meminimalisir terjadinya gagal transaksi dan menjadi tren dalam meningkatkan kemudahan dan kecepatan pengguna untuk menemukan barang yang diinginkan secara *online*.

Sistem Rekomendasi Film Menggunakan Metode *Hybrid Collaborative Filtering* Dan *Content-Based Filtering* oleh Hilmi Hidayat Arfisko, dkk. Penelitian pada tahun 2022 ini mengembangkan sistem rekomendasi film dengan menggunakan metode *hybrid collaborative filtering* dan *content-based filtering*. Dalam penelitian ini, hasil rekomendasi sistem dibandingkan dengan metode *collaborative filtering*, metode *content-based filtering*, dan kombinasi kedua metode tersebut. Hasil pengujian menunjukkan bahwa metode *hybrid collaborative filtering* dan *content-based filtering* menghasilkan *list* rekomendasi film yang lebih baik daripada metode lainnya yang diuji dalam *dataset* pengujian.

2.2. Dasar Teori

2.2.1. Sistem Informasi

Sistem informasi merupakan suatu sistem yang ada di dalam organisasi dengan fungsi untuk menghubungkan kebutuhan pengolahan transaksi, mendukung operasi, memiliki sifat manajerial, dan melibatkan kegiatan strategi dari organisasi tersebut. Selain itu, sistem ini juga menyediakan laporan-laporan kepada pihak luar yang relevan (Saputra, 2020).

Sistem informasi merupakan elemen yang penting dalam sebuah perusahaan yang harus saling berinteraksi antara komponen-komponennya untuk mencapai tingkat yang baik dalam struktur organisasi, sumber daya manusia, produk, dan rekanan dalam organisasi tersebut. Dalam sistem ini, berbagai komponen

perusahaan bekerja sama dan saling terhubung untuk mencapai tujuan yang diinginkan. Sistem informasi juga berperan dalam mengatur aliran informasi dan mengelola sumber daya yang ada di dalam perusahaan. Sistem ini mengintegrasikan berbagai aspek penting, seperti kegiatan operasional, pengolahan data transaksi, dan strategi organisasi. Melalui sistem informasi, perusahaan dapat meningkatkan efisiensi dan efektivitasnya dalam menjalankan aktivitas sehari-hari (Wahyudi dan Ridho, 2019).

2.2.2. Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan atau *Decision Support System* merupakan sebuah sistem yang bertujuan untuk membantu para pengambil keputusan dalam situasi keputusan yang tidak sepenuhnya terstruktur, tetapi bukan untuk menggantikan peran seorang pengambil keputusan (Faisol dkk., 2014).

Sistem Pendukung Keputusan (SPK) tidak berfungsi sebagai pengambil keputusan, tetapi sebagai sebuah sistem yang membantu para pengambil keputusan dengan memberikan informasi yang relevan dan diperlukan dari data yang telah diproses, sehingga memungkinkan mereka untuk membuat keputusan dengan lebih cepat dan akurat. SPK dirancang untuk membantu pengambil keputusan dalam menyelesaikan masalah yang semi atau tidak terstruktur, dengan fokus pada penyediaan informasi sebagai alternatif untuk membuat keputusan terbaik.

2.2.3. Sistem Rekomendasi

Sistem rekomendasi merupakan suatu aplikasi yang berguna untuk memberikan saran kepada pengguna mengenai item-item yang mungkin akan mereka minati. Aplikasi ini menggunakan data yang tersedia untuk menganalisis apa yang mungkin diminati oleh pengguna dan memberikan rekomendasi yang sesuai dengan keinginan mereka. Sistem rekomendasi sering disebut juga sebagai mesin rekomendasi yang dapat membantu dalam membuat keputusan yang diinginkan oleh pengguna (Rymmai dan Saleema, 2017).

Beberapa faktor yang dapat memengaruhi keputusan sistem dalam memberikan rekomendasi kepada pengguna, seperti perilaku pengguna, deskripsi item, preferensi, dan kebiasaan dari sekelompok pengguna yang memiliki kemiripan dalam menilai suatu item. Sistem rekomendasi telah menjadi alat yang

efektif untuk menyaring informasi dan menekankan pada karakteristik penyaringan informasi, yaitu menyaring produk yang mungkin menarik bagi pengguna. Sistem rekomendasi sangat berguna dalam mengeliminasi kebisingan informasi dan memberikan rekomendasi yang relevan bagi pengguna (Shao dkk., 2021).

2.2.4. Buku

Buku terdiri dari sejumlah lembar kertas yang berisi informasi yang tercetak dan disusun secara sistematis. Buku terdapat jilid yang menjaga agar lembar-lembar kertas tersebut tidak terpisah. Bagian luar buku biasanya dilindungi oleh bahan-bahan seperti kertas tebal, karton, atau bahan lain. Buku merupakan kumpulan informasi yang tercetak dan disusun secara teratur. Buku biasanya memiliki pelindung bagian luar untuk melindungi lembar-lembar kertas di dalamnya (Sitepu, 2012).

2.2.5. *Content-Based Filtering*

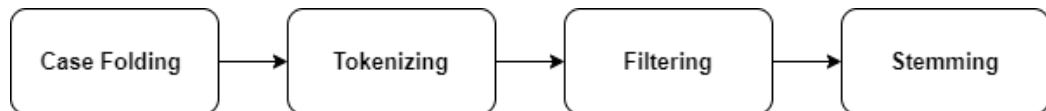
Content-Based Filtering merupakan metode yang digunakan pada sistem rekomendasi. Metode ini mempertimbangkan perilaku pengguna dari masa lalu dan mengidentifikasi pola perilakunya untuk merekomendasikan barang yang sesuai dengan pola tersebut. *Content-Based Filtering* menganalisis preferensi dari perilaku pengguna dimasa lalu untuk membuat model.

Model tersebut akan dibandingkan dengan serangkaian karakteristik atribut dari barang yang akan direkomendasikan. Barang yang cocok dengan model akan menjadi rekomendasi untuk pengguna, barang yang cocok dengan model akan menjadi rekomendasi dengan tingkat kecocokan tertinggi (Reddy dkk., 2019).

2.2.6. *Preprocessing*

Preprocessing merupakan proses mengubah teks menjadi indeks kata dengan maksud menghasilkan kumpulan indeks kata yang dapat mewakili dokumen. Dalam tahap awal klasifikasi, dua hal yang sangat penting adalah *preprocessing* teks dan ekstraksi fitur. Dalam proses ini, setiap teks diubah menjadi vektor dengan mewakili frekuensi kata (*term*) untuk setiap dokumen (Supriyati dan Iqbal, 2018).

Tahap *preprocessing* digunakan untuk memproses data mentah agar bisa diolah pada tahap selanjutnya, langkah awal yang dilakukan adalah *preprocessing*. Beberapa tahapan yang dilakukan untuk *preprocessing* data meliputi *case folding*, *tokenizing*, *filtering* (*stopword removal*), dan *stemming*. Tahapan-tahapan tersebut dapat dilihat pada Gambar 2.1.



Gambar 2. 1. Tahapan *Preprocessing*

A. *Case folding*

Proses *case folding* dilakukan untuk mengubah huruf besar pada kata-kata atau kalimat di dalam dokumen menjadi huruf kecil. Tujuan dari proses ini adalah untuk membuat kesamaan pada setiap kata dalam dokumen data buku. Contoh implementasi tahap ini dapat dilihat pada Tabel 2.1.

Tabel 2. 1. Contoh implementasi *case folding*

Sebelum	Sesudah
Melepasmu adalah keputusan paling dewasa yang aku ambil, di mana aku butuh berfikir sekeras-kerasnya, butuh tenaga sebesar-besarnya, dan terpaksa harus sesakit-sakitnya.	melepasmu adalah keputusan paling dewasa yang aku ambil, di mana aku butuh berfikir sekeras-kerasnya, butuh tenaga sebesar-besarnya, dan terpaksa harus sesakit-sakitnya.
Walaupun sudah susah payah merapikan rumah, apakah kertas-kertas terus saja bertumpuk dan pakaian harus terus Anda jejal-jejalkan di lemari? Kenapa kita tidak bisa menjaga kerapian rumah?	walaupun sudah susah payah merapikan rumah, apakah kertas-kertas terus saja bertumpuk dan pakaian harus terus anda jejal-jejalkan di lemari? kenapa kita tidak bisa menjaga kerapian rumah?
Perkenalkan, Raib, seorang gadis belia berusia lima belas tahun yang tidak biasa. Dia bisa menghilang. Jangan beritahu siapapun, Itu adalah rahasia terbesar yang tak pernah ia ceritakan pada siapapun, termasuk kedua orangtuanya.	perkenalkan, raib, seorang gadis belia berusia lima belas tahun yang tidak biasa. dia bisa menghilang. jangan beritahu siapapun, itu adalah rahasia terbesar yang tak pernah ia ceritakan pada siapapun, termasuk kedua orangtuanya.

B. *Tokenizing*

Tokenizing merupakan proses memisahkan *string input* menjadi token atau kata-kata berdasarkan spasi atau karakter pemisah tertentu. Pada tahap ini, data dari setiap buku akan dibersihkan dari simbol, tanda baca, angka, *backslash*, karakter non-ASCII, *URL*, *whitespace leading* dan *trailing*, serta mengubah *multiple whitespace* menjadi *single whitespace*. Dengan membersihkan data, ukuran dokumen yang akan di-*tokenize* menjadi lebih kecil sehingga prosesnya dapat dilakukan lebih cepat. Implementasi dari dokumen data buku yang telah melalui proses *tokenizing* dapat dilihat pada Tabel 2.2.

Tabel 2. 2. Contoh implementasi *tokenizing*

Sebelum	Sesudah
melepassmu adalah keputusan paling dewasa yang aku ambil, di mana aku butuh berfikir sekeras-kerasnya, butuh tenaga sebesar-besarnya, dan terpaksa harus sesakit-sakitnya.	[melepassmu, adalah, keputusan, paling dewasa, yang, aku, ambil, di, mana, aku, butuh, berfikir, sekeras-kerasnya, butuh, tenaga, sebesar-besarnya, dan, terpaksa, harus, sesakit-sakitnya]
walaupun sudah susah payah merapikan rumah, apakah kertas-kertas terus saja bertumpuk dan pakaian harus terus anda jejal-jejalkan di lemari? kenapa kita tidak bisa menjaga kerapian rumah?	[walaupun, sudah, susah, payah, merapikan, rumah, apakah, kertas-kertas, terus, saja, bertumpuk, dan, pakaian, harus, terus, anda, jejal-jejalkan, di, lemari, kenapa, kita, tidak, bisa, menjaga, kerapian, rumah,]
perkenalkan, raib, seorang gadis belia berusia lima belas tahun yang tidak biasa. dia bisa menghilang. jangan beritahu siapapun, itu adalah rahasia terbesar yang tak pernah ia ceritakan pada siapapun, termasuk kedua orangtuanya.	[perkenalkan, raib, seorang, gadis, belia, berusia, lima, belas, tahun, yang, tidak, biasa, dia, bisa, menghilang, jangan, beritahu, siapapun, itu, adalah, rahasia, terbesar, yang, tak, pernah, ia, ceritakan, pada, siapapun, termasuk, kedua, orangtuanya]

C. *Filtering*

Proses *filtering* dilakukan untuk mengekstraksi kata-kata penting dari dokumen data buku dengan menghilangkan kata-kata yang tidak deskriptif seperti kata penghubung, kata depan, dan kata pengganti seperti "di", "yang", "dan", "ke", dan "dari". Tujuan dari proses *filtering* adalah untuk mengurangi jumlah kata dalam dokumen sehingga proses pengolahan dapat dilakukan dengan lebih akurat menggunakan kata-kata yang penting dalam dokumen. Contoh implementasi dari proses *filtering* dapat dilihat pada Tabel 2.3.

Tabel 2. 3. Contoh implementasi *filtering*

Sebelum	Sesudah
[melepasmu, adalah, keputusan, paling dewasa, yang, aku, ambil, di, mana, aku, butuh, berfikir, sekeras-kerasnya, butuh, tenaga, sebesar-besarnya, dan, terpaksa, harus, sesakit-sakitnya]	[melepasmu, adalah, keputusan, paling, dewasa, aku, ambil, mana, aku, butuh, berfikir, sekeras-kerasnya, butuh, tenaga, sebesar-besarnya, terpaksa, harus, sesakit-sakitnya]
[walaupun, sudah, susah, payah, merapikan, rumah, apakah, kertas-kertas, terus, saja, bertumpuk, dan, pakaian, harus, terus, anda, jejal-jejalkan, di, lemari, kenapa, kita, tidak, bisa, menjaga, kerapian, rumah,]	[walaupun, sudah, susah, payah, merapikan, rumah, apakah, kertas-kertas, terus, saja, bertumpuk, pakaian, harus, terus, anda, jejal-jejalkan, lemari, kenapa, kita, tidak, bisa, menjaga, kerapian, rumah,]
[perkenalkan, raib, seorang, gadis, belia, berusia, lima, belas, tahun, yang, tidak, biasa, dia, bisa, menghilang, jangan, beritahu, siapapun, itu, adalah, rahasia, terbesar, yang, tak, pernah, ia, ceritakan, pada, siapapun, termasuk, kedua, orangtuanya]	[perkenalkan, raib, seorang, gadis, belia, berusia, lima, belas, tahun, tidak, biasa, dia, bisa, menghilang, jangan, beritahu, siapapun, itu, adalah, rahasia, terbesar, tak, pernah, ia, ceritakan, pada, siapapun, termasuk, kedua, orangtuanya]

D. Stemming

Tahap *stemming* dilakukan untuk menemukan kata dasar dari kata yang telah melewati proses *filtering*. *Stemming* berfungsi untuk menghapus kata awalan, akhiran, sisipan, dan kombinasi antara awalan dan akhiran dari sebuah kata. Proses *stemming* dilakukan dengan menggunakan *library sastrawi* yang mampu mengubah kata-kata dengan imbuhan pada dokumen data buku menjadi bentuk dasarnya. Contoh implementasi proses *stemming* dapat dilihat pada Tabel 2.4.

Tabel 2. 4. Contoh implementasi *stemming*

Sebelum	Sesudah
[melepasmu, adalah, keputusan, paling, dewasa, aku, ambil, mana, aku, butuh, berfikir, sekeras-kerasnya, butuh, tenaga, sebesar-besarnya, terpaksa, harus, sesakit-sakitnya]	[lepas, adalah, putusan, paling, dewasa, aku, ambil, mana, aku, butuh, fikir, keras, butuh, tenaga, besar, paksa, harus, sakit]
[walaupun, sudah, susah, payah, merapikan, rumah, apakah, kertas-kertas, terus, saja, bertumpuk, pakaian, harus, terus, anda, jejal-jejalkan, lemari, kenapa, kita, tidak, bisa, menjaga, kerapian, rumah,]	[walau, sudah, susah, payah, rapikan, rumah, apakah, kertas, terus, saja, tumpuk, pakaian, harus, terus, anda, jejal, lemari, kenapa, kita, tidak, bisa, jaga, rapi, rumah,]

Tabel 2. 4. Contoh implementasi *stemming* (lanjutan)

Sebelum	Sesudah
[perkenalkan, raib, seorang, gadis, belia, berusia, lima, belas, tahun, tidak, biasa, dia, bisa, menghilang, jangan, beritahu, siapapun, itu, adalah, rahasia, terbesar, tak, pernah, ia, ceritakan, pada, siapapun, termasuk, kedua, orangtuanya]	[kenal, raib, seorang, gadis, belia, usia, lima, belas, tahun, tidak, biasa, dia, bisa, hilang, jangan, beritahu, siapapun, itu, adalah, rahasia, besar, tak, pernah, ia, cerita, pada, siapapun, termasuk, kedua, orangtua]

2.2.7. Perpustakaan Sekolah

Perpustakaan sekolah merupakan bagian dari lembaga pendidikan sekolah yang bertujuan untuk mendukung tercapainya tujuan pendidikan sekolah tersebut. Perpustakaan sekolah merupakan sumber belajar yang tersedia di sekolah. Perpustakaan sekolah merupakan bagian integral dari sekolah yang bersangkutan. Perpustakaan sekolah merupakan fasilitas yang terdapat di sekolah untuk mendukung kegiatan belajar siswa. Perpustakaan sekolah merupakan tempat dimana siswa dapat mencari sumber belajar yang diperlukan untuk mendukung keberhasilan di sekolah (Hartono, 2016).

2.2.8. *Hypertext Preprocessor (PHP)*

PHP merupakan bahasa pemrograman yang digunakan untuk menerjemahkan kode program menjadi bentuk yang dapat dimengerti oleh komputer. *PHP* merupakan bahasa pemrograman *server-side*, yang berarti bahwa kode *PHP* akan dijalankan di *server* sebelum ditampilkan di halaman web. Bahasa *PHP* sangat berguna untuk menambahkan fitur-fitur dinamis pada halaman web yang dibuat dengan *HTML*. *PHP* dapat digunakan untuk mengolah data, mengelola *database*, dan melakukan berbagai tugas lain yang diperlukan untuk menjalankan sebuah situs web (Supono dan Putratama, 2016).

2.2.9. *Sastrawi*

Algoritma *stemming Sastrawi* merupakan hasil pengembangan dari algoritma *NA* yang merupakan sebuah *library stemmer*. Algoritma *Sastrawi* dapat digunakan dalam bahasa pemrograman *python* dan *PHP* sebagai *library stemmer*. Proses

stemming dengan menggunakan algoritma Sastrawi sangat tergantung pada kamus kata dasar. Keberhasilan proses *stemming* sangat dipengaruhi oleh kualitas kamus yang digunakan. Algoritma Sastrawi bertujuan untuk mengubah kata-kata dalam bentuk infleksi menjadi kata dasar. Dengan menggunakan algoritma ini, pengguna dapat memperoleh kata dasar dari kata-kata yang beragam bentuknya (Rosid dkk., 2020).

2.2.10. *HyperText Markup Language (HTML)*

Hypertext Markup Language (HTML) merupakan sebuah bahasa yang sering digunakan oleh *web developer* untuk membuat tampilan pada aplikasi web. *HTML* merupakan salah satu bahasa yang paling sering digunakan oleh klien dalam mengembangkan aplikasi web dikarenakan bahasa pemrograman ini sangat mudah dipahami dan dipelajari, serta dapat digunakan untuk membuat tampilan yang atraktif dan responsif. Selain itu, *HTML* juga merupakan bahasa standar yang digunakan oleh sebagian besar *browser* internet, sehingga aplikasi web yang dikembangkan dengan menggunakan *HTML* akan bisa diakses oleh banyak orang (Yuli dan Synta, 2018).

2.2.11. *Cascading Stylesheet (CSS)*

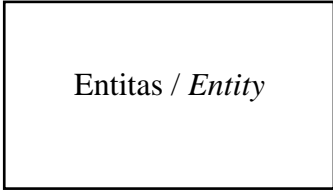


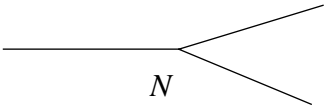
Cascading Style Sheets (CSS) merupakan bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun komponen dalam sebuah website, sehingga website tersebut terlihat lebih terstruktur, rapi, dan seragam. *CSS* memiliki tujuan utama untuk menghias dan mengatur tampilan dan *layout* website agar terlihat lebih elegan dan menarik. *CSS* merupakan teknologi internet yang direkomendasikan oleh *World Wide Web Consortium (W3C)* pada tahun 1996. Dengan menggunakan *CSS*, pengguna dapat dengan mudah mengatur tampilan website sesuai dengan keinginan, sehingga website tersebut terlihat lebih profesional dan menarik bagi pengunjung (Wahyudi, 2017)

2.2.12. Entity Relationship Diagram (ERD)

Entity relationship diagram (ERD) merupakan model yang digunakan untuk menjelaskan hubungan antar data dalam basis data. *ERD* digunakan untuk menggambarkan hubungan antar relasi dalam basis data notasi dan simbol tertentu digunakan dalam *ERD* untuk menggambarkan model struktur data dan hubungan antar data.

ERD merupakan bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan *OODMBS*, perancangan *ERD* tidak perlu dilakukan (Samsudin dkk., 2019). Tabel 2.5. adalah komponen *ERD* yang akan digunakan untuk penelitian ini.


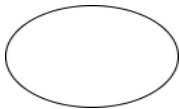

Tabel 2. 5. Komponen *ERD*

Komponen	Deskripsi
	Entitas adalah data inti yang akan disimpan dalam sebuah basis data. Ia merupakan bakal tabel yang digunakan untuk menyimpan data. Entitas adalah benda yang memiliki data yang harus disimpan agar dapat diakses oleh aplikasi komputer. Penanaman entitas biasanya lebih ke arah kata benda dan belum merupakan nama tabel pada basis data.
	Kolom atau bidang data yang perlu disimpan dalam sebuah entitas.
	Relasi merupakan Penghubung antar entitas. biasanya diwali dengan kata kerja.
Asosiasi 	Kardinalitas adalah jembatan antara relasi dan entitas di mana kedua ujungnya memiliki jumlah maksimum pemakaian keterkaitan antara entitas satu dengan yang lain. Contohnya, kardinalitas 1 ke N, atau yang sering disebut sebagai <i>one to many</i> , menghubungkan entitas A dan entitas B.

2.2.13. Use Case Diagram

Use Case merupakan penjelasan tentang fungsi dari sebuah sistem dari sudut pandang pengguna sistem. *Use Case* menentukan apa yang akan dikerjakan oleh sistem dan komponennya. *Use Case* bekerja dengan menggunakan skenario yang merupakan deskripsi dari urutan atau tahapan yang menjelaskan apa yang dilakukan oleh pengguna terhadap sistem atau sebaliknya (Setiyani, 2021). Tabel 2.6. adalah komponen *Use Case* yang akan digunakan untuk penelitian ini.

Tabel 2. 6. Komponen *Use Case Diagram*

Komponen	Deskripsi
Aktor 	Menunjukkan peran dari individu, sistem lain, atau alat saat berkomunikasi dengan use case.
<i>Use Case</i> 	Deskripsi umum dari hubungan dan komunikasi antara sistem dan pihak-pihak yang berinteraksi dengannya (aktor).
Asosiasi 	Deskripsi umum dari hubungan antara aktor dengan <i>use case</i> sebagai perantara.

2.2.14. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF merupakan sebuah aplikasi dari statistik numerik yang menunjukkan seberapa relevan suatu kata kunci terhadap sejumlah dokumen tertentu. Dengan menyediakan kata kunci yang tersedia, dokumen tertentu dapat diidentifikasi atau dikategorikan berdasarkan tingkat relevansinya (Qaiser dan Ali, 2018).

TF-IDF merupakan gabungan dari dua konsep yang berbeda, yaitu *Term Frequency (TF)* dan *Inverse Document Frequency (IDF)*. *Term Frequency (TF)* digunakan untuk mengukur berapa kali sebuah istilah muncul dalam sebuah dokumen. Sebagai contoh, dalam sebuah dokumen "A7" yang berisi 1000 kata dan terdapat kata "Informasi" sebanyak 10 kali.

Panjang total dokumen bisa bervariasi dari sangat kecil hingga besar, sehingga ada kemungkinan bahwa istilah apapun bisa muncul lebih sering dalam dokumen yang lebih besar daripada dokumen yang lebih kecil. Oleh karena itu, untuk mengatasi masalah ini, kemunculan istilah apapun dalam dokumen dibagi dengan total istilah yang ada dalam dokumen tersebut untuk menemukan frekuensi istilah. Jadi dalam hal ini, frekuensi istilah kata "Informasi" dalam dokumen "A7" adalah $TF = 10/1000 = 0.001$ (Qaiser dan Ali, 2018).

Inverse Document Frequency (IDF) digunakan untuk mengukur seberapa penting sebuah kata di dalam sebuah dokumen saat menghitung frekuensi istilah dalam dokumen. Dapat diamati bahwa algoritma memperlakukan semua kata kunci dengan cara yang sama, tidak peduli apakah kata tersebut termasuk *stopwords* seperti "dari". Setiap kata kunci memiliki nilai penting yang berbeda. Sebagai contoh, jika *stopword* "dari" muncul dalam dokumen sebanyak 2000 kali tetapi tidak memiliki arti yang terlalu penting, itulah fungsi dari *IDF*. *Inverse Document Frequency* memberikan bobot yang lebih rendah pada kata-kata yang sering muncul dan memberikan bobot yang lebih tinggi pada kata-kata yang jarang muncul (Qaiser dan Ali, 2018).

Contoh perhitungan *IDF* adalah jika terdapat 10 dokumen dan kata "Teknologi" muncul dalam 5 dokumen tersebut, maka ukuran *Inverse Document Frequency (IDF) = log(10/5) = 0.3010*. Algoritma *TF-IDF* umumnya digunakan sebagai metode untuk mengukur deskripsi suatu item tekstual dalam pendekatan *Content Based Filtering* (Lops dkk., 2019). Persamaan (2.1) digunakan untuk melakukan penghitungan *term frequency* (Oeyliawan dan Gunawan, 2017).

$$tf_{i,j} = f_{i,j} \quad (2.1)$$

TF adalah *term frequency* dan $tf_{i,j}$ adalah seberapa sering sebuah kata muncul dalam sebuah dokumen. $tf_{i,j}$ adalah jumlah kemunculan kata t_i dalam dokumen d_j . Untuk menghitung *TF* dengan menghitung berapa kali kata t_i muncul dalam dokumen d_j . Setelah itu, untuk menghitung *Inverse Document Frequency* dapat dilakukan dengan menggunakan persamaan (2.2)

$$idf_i = \log\left(\frac{N}{df_i}\right) \quad (2.2)$$

idf_i adalah kebalikan frekuensi dokumen (*Inverse Document Frequency*). N adalah total dokumen yang diambil oleh sistem, sedangkan DFI adalah jumlah dokumen dalam koleksi *term* t_i muncul di dalamnya. Perhitungan idf_i digunakan untuk mengetahui jumlah *term* yang dicari (d_{fi}) yang muncul dalam dokumen lain (Oeyliawan dan Gunawan, 2017). Perhitungan bobot *term* yang didasarkan pada perkalian nilai tf dan idf , deskripsi terbaik dari suatu dokumen dapat ditentukan dengan mencari *term* yang banyak muncul dalam dokumen tersebut dan sangat jarang muncul dalam dokumen lain. Persamaan (2.3) digunakan untuk menghitung bobot *Term Frequency-Inverse Document Frequency*.

$$W_{i,j} = tf_{i,j} \log \frac{N}{df_i} \quad (2.3)$$

2.2.15. Cosine Similarity

Metode *Cosine Similarity* adalah sebuah cara untuk mengukur seberapa mirip dua dokumen atau teks. Dalam metode ini, dokumen atau teks dianggap sebagai *vector* (Samuel, dkk., 2018). Secara lebih luas, *cosine similarity* antara dua vektor (atau dua dokumen dalam ruang vektor) adalah ukuran yang mengukur sudut kosinus antara keduanya.

Metrik ini mengukur orientasi, bukan ukuran, dan dapat dilihat sebagai perbandingan antara dokumen dalam ruang yang dinormalisasi, karena tidak hanya besaran setiap kata yang dihitung (*TF-IDF*) dari setiap dokumen yang dipertimbangkan, tetapi juga sudut antar dokumen (Rymmai dan Saleema, 2017). Salah satu cara untuk membangun persamaan *cosine similarity* adalah dengan menggunakan persamaan (2.4)

$$Sim(q, d_j) = \frac{q \times d_j}{|q| \times |d_j|} = \frac{\sum_{i=1}^n W_{i,q} \times W_{i,j}}{\sqrt{\sum_{i=1}^t (W_{i,q})^2} \sqrt{\sum_{i=1}^t (W_{i,j})^2}} \quad (2.4)$$

Dari persamaan (2. 4), $Sim(q, d_j)$ adalah seberapa mirip *query* dengan dokumen, $|q|$ adalah jarak *query*, dan $|d_j|$ adalah jarak dokumen. $W_{i,j}$ adalah bobot dari dokumen ke-i, dan $W_{i,q}$ adalah bobot dari query dokumen ke-i. Similaritas antara *query* dan dokumen bergantung pada hasil perkalian bobot *query* (q) dengan bobot dokumen (d_j) dan berkurang seiring dengan perkalian dari akar kuadrat dari jarak *query* ($|q|$) dengan akar kuadrat dari jarak dokumen ($|d_j|$). Hasil dari perhitungan similaritas akan menghasilkan bobot dokumen yang mendekati nilai 1 (Oeyliawan dan Gunawan, 2017).

BAB III

METODOLOGI PENELITIAN

3.1. Bahan dan Alat Penelitian

Dalam penelitian ini, penulis menggunakan alat yang digunakan untuk melakukan penelitian yang berupa perangkat keras dan perangkat lunak, yaitu:

3.1.1. Perangkat keras

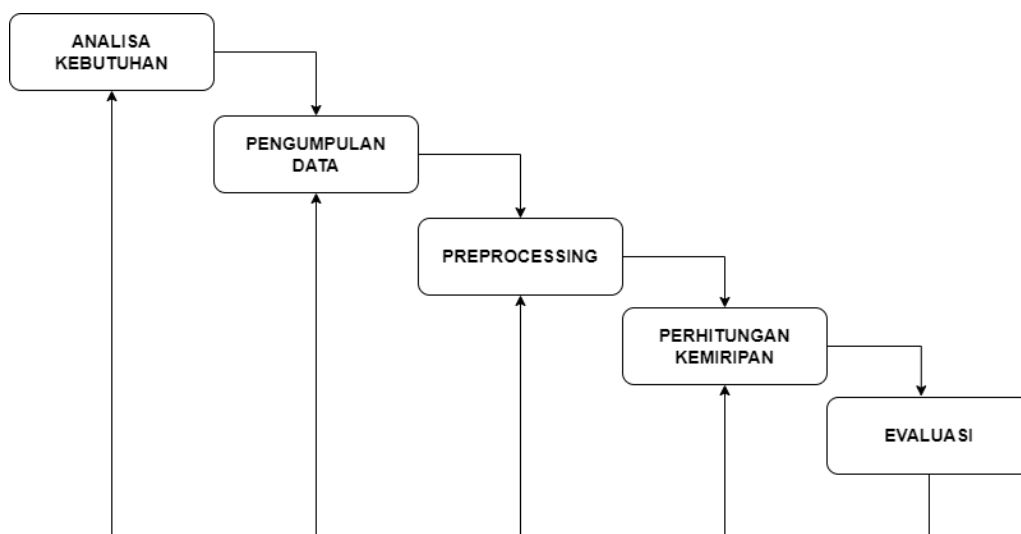
1. Laptop dengan spesifikasi *CPU Intel(R) Core(TM) i5-1035G1, RAM 8,00 GB DDR4* dengan OS *Windows 11 Home 64-bit*.
2. *Smartphone* dengan spesifikasi *CPU Qualcomm SM8350 Snapdragon 888, RAM 8,00 GB* dengan OS *Android 13*.

3.1.2. Perangkat lunak

1. *PHP* dan *JavaScript* sebagai bahasa pemrograman.
2. *MySQL* sebagai *database*
3. *Chrome* sebagai *web browser*
4. *Laragon* sebagai *local server*
5. *Visual Studio Code* sebagai *code editor*

3.2. Alur Penelitian

Untuk mengembangkan sistem ini, penulis menggunakan metode *waterfall*. Metode *waterfall* ini banyak dipakai dalam *Software Engineering*. Metode *waterfall* dianggap sebagai metode yang tepat karena setiap tahap harus diselesaikan secara bertahap dan harus berurutan. Setiap tahap harus menunggu selesainya tahap sebelumnya sebelum melanjutkan ke tahap berikutnya. Alur penelitian ditunjukkan pada Gambar 3.1.



Gambar 3. 1. Metode *waterfall* yang digunakan

3.2.1. Analisa Kebutuhan

Pada langkah ini, penulis akan melakukan analisis kebutuhan yang diperoleh melalui diskusi dengan kepala perpustakaan dan observasi langsung di perpustakaan SMA Muhammadiyah 8 Sukodadi. Tahapan ini bertujuan untuk memperoleh data yang relevan dan mendalam mengenai kebutuhan perpustakaan tersebut. Dalam diskusi tersebut, tujuan penelitian ini akan menjadi topik utama yang dibahas. Tujuan tersebut antara lain:

1. Merancang dan mengembangkan sebuah sistem rekomendasi buku pada perpustakaan SMA Muhammadiyah 8 Sukodadi dengan menggunakan metode *content-based filtering*.
2. Merancang sebuah sistem yang dapat memudahkan admin perpustakaan dalam mengelola transaksi peminjaman dan pengembalian buku.
3. Mengukur tingkat keakuratan sistem dalam memberikan saran buku menggunakan pembobotan *TF-IDF* dan perhitungan *cosine similarity* dari parameter yang diambil dari judul buku, penulis, kategori, dan deskripsi buku yang ada pada perpustakaan SMA Muhammadiyah 8 Sukodadi.

3.2.2. Pengumpulan Data

Data yang akan digunakan untuk sistem rekomendasi buku ini berasal dari perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi. Alasan penggunaan data dari perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi adalah karena perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi adalah satu-satunya perpustakaan yang ada di sekolah SMA Muhammadiyah 8 Sukodadi yang sering digunakan siswa untuk membaca buku dan mencari referensi untuk tugas sekolah. Terdapat 517 buku yang digunakan pada penelitian ini dan buku yang diambil dari berbagai kategori seperti novel, sejarah, fantasi, cerita rakyat, dan lain-lain.

3.2.3. Teknik Pengumpulan Data

Proses pengambilan data ini dibutuhkan beberapa indikator penting pada sistem rekomendasi seperti judul buku, penulis, kategori, dan deskripsi dari buku tersebut. Data buku diambil sebanyak 517 buku dari berbagai macam kategori yang merupakan buku-buku populer dari perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi. Hasil data yang diambil berupa judul, penulis, kategori, deskripsi dari buku tersebut kemudian disimpan ke dalam *database*, sehingga data tersebut dapat diolah dan dimanfaatkan untuk membuat sistem rekomendasi dengan menggunakan bahasa pemrograman *PHP*.

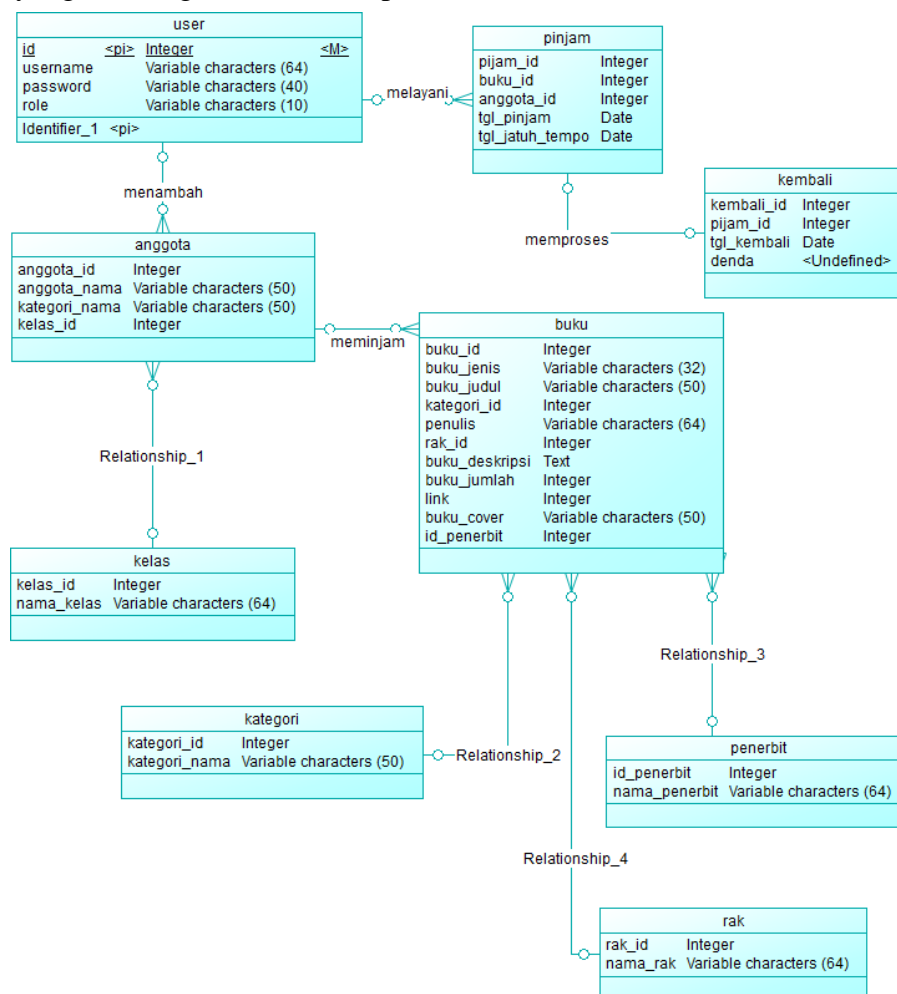
3.3. Rancangan Database

3.3.1. Perancangan Database

Untuk membuat sistem rekomendasi buku ini, langkah pertama yang harus dilakukan adalah membuat perancangan *database*. *Database* tersebut akan dibagi menjadi beberapa tabel yang akan digunakan untuk menyimpan semua informasi yang terkait dengan perpustakaan tersebut. Tabel-tabel tersebut akan dibuat menggunakan *server* basis data bernama *MySQL Server*. Perancangan *database* pada penelitian ini dibuat pada struktur diagram *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)* yang nantinya akan diimplementasikan pada *database MySQL*.

3.3.2. Conceptual Data Model (CDM)

Conceptual Data Model (CDM) adalah sebuah konsep yang berhubungan dengan cara pandang pengguna terhadap data yang disimpan dalam sebuah basis data. CDM dibuat dalam bentuk tabel-tabel tanpa memperhatikan tipe data yang mendeskripsikan hubungan antar tabel, hal ini dilakukan untuk tujuan implementasi ke dalam basis data (Rahim, 2020). Gambar 3.2. adalah rancangan *conceptual data model* yang akan digunakan dalam penelitian ini.

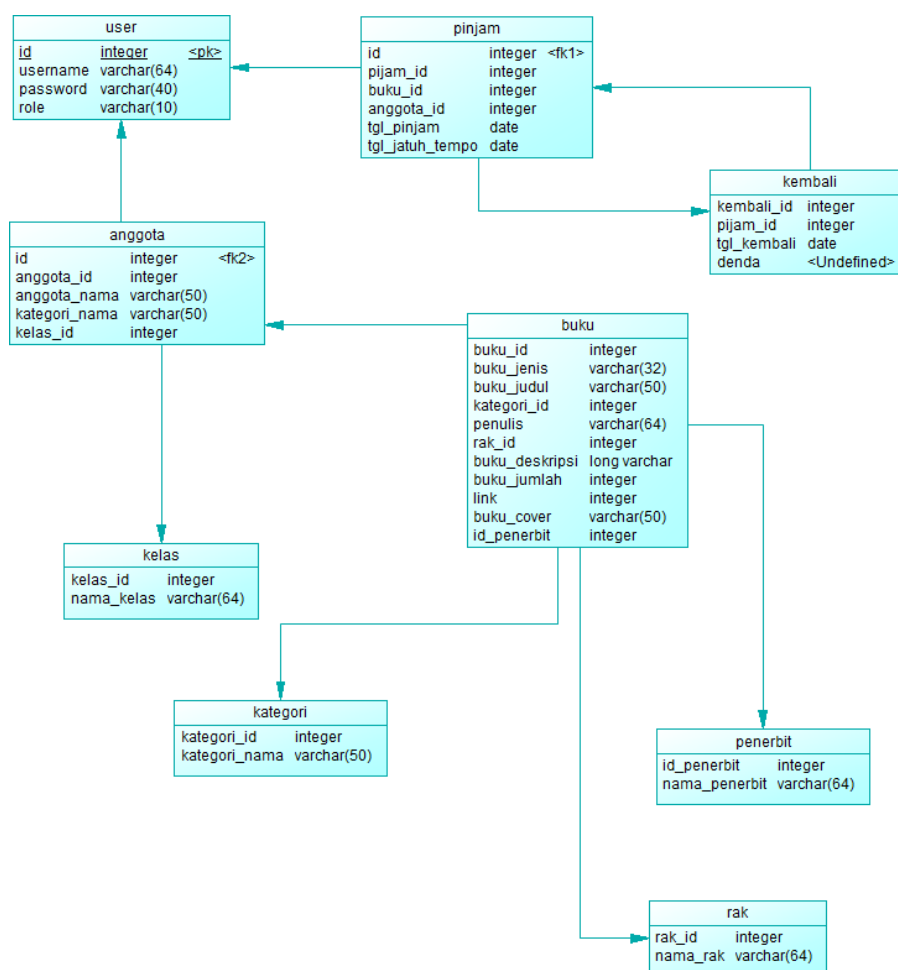


Gambar 3. 2. Rancangan *conceptual data model*

Rancangan *Conceptual Data Model (CDM)* pada Gambar 3.2. menunjukkan rancangan hubungan antar tabel. Nantinya akan dibuat 9 tabel pada *database* bernama buku dan akan memiliki relasi antara satu tabel dengan tabel yang lain.

3.3.3. Physical Data Model (PDM)

Physical data model (PDM) adalah model yang mendefinisikan struktur fisik dari *database*, termasuk tabel, kolom, tipe data, indeks, *constraint*, dan lain-lain. Sebuah *Physical Data Model (PDM)* adalah representasi rinci dari rencana struktur basis data yang dibuat khusus untuk suatu program aplikasi (Diaz dkk., 2016). Gambar 3.3. merupakan rancangan *physical data model* yang akan digunakan pada penelitian kali ini.

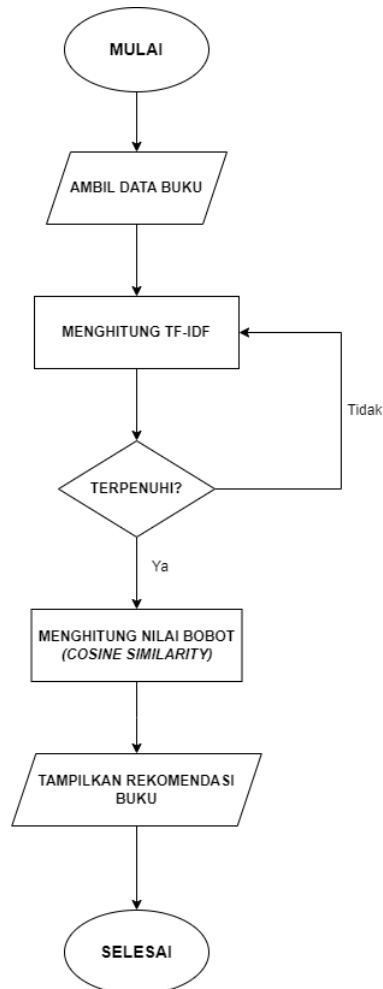


Gambar 3. 3. Rancangan *physical data model*

Pada rancangan *PDM* ini, dapat disimpulkan bahwa model telah berbentuk rancangan *database* yang siap digunakan dalam pengembangan website. Setiap entitas sudah memiliki atribut dengan tipe data yang sesuai untuk perancangan *database*, serta telah diidentifikasi atribut mana yang berperan sebagai *primary key*, *foreign key*, dan *non-primary key*.

3.4. Rancangan Sistem

3.4.1. *Flowchart* Sistem Rekomendasi



Gambar 3. 4. *Flowchart* sistem rekomendasi

Gambar 3.4. adalah rancangan *flowchart* sistem rekomendasi yang akan diimplementasikan pada penelitian ini. Tahapan-tahapan dari proses sistem rekomendasi ini adalah sebagai berikut:

1. Mulai : Memulai proses
2. Ambil data buku : Mengambil data kriteria buku yang ada pada *database*
3. Menghitung *TF-IDF* : Proses perhitungan kriteria buku menggunakan pembobotan *TF-IDF*. jika kriteria terpenuhi, proses akan dilanjutkan. Namun jika tidak terpenuhi, proses perhitungan akan diulang.
4. Menghitung nilai bobot (*cosine similarity*) : Proses perhitungan nilai bobot kriteria menggunakan rumus *cosine similarity*.

5. Tampilkan rekomendasi buku : Menampilkan rekomendasi buku yang sesuai berdasarkan perhitungan *TF-IDF* dan pembobotan *cosine similarity*. Dengan menggunakan perhitungan *TF-IDF* dan pembobotan *cosine similarity*, sistem dapat menampilkan rekomendasi buku yang paling relevan.

6. Selesai : Proses selesai

3.4.2. Use Case Diagram



Gambar 3. 5. Rancangan *use case*

Tabel 3. 1. Deskripsi rancangan *use case*

Aktor	Deskripsi
Pustakawan	<i>user</i> pustakawan dapat menambah <i>user</i> guru, menambahkan data <i>user</i> siswa baru, melihat data siswa, menambah data buku baru, melihat buku, mengubah stok buku, dan dapat mengelola peminjaman dan pengembalian buku di perpustakaan.

Tabel 3. 1. Deskripsi rancangan *use case* (lanjutan)

Aktor	Dekskripsi
Guru	<i>user</i> guru dapat melihat data <i>user</i> siswa, melihat stok buku, mencari judul buku, melihat deskripsi buku, melihat kategori buku, dan melihat rekomendasi buku.
Siswa	<i>user</i> siswa dapat melihat stok buku, mencari judul buku, melihat deskripsi buku, melihat kategori buku, dan melihat rekomendasi buku.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1. Profil sekolah

Sekolah SMA Muhammadiyah 8 Sukodadi berdiri pada tahun 1988. Sekolah SMA Muhammadiyah 8 ini berlokasi di Jl.Raden Wijaya Nomor 19, Kecamatan Sukodadi, Kabupaten Lamongan. Sekolah ini terletak dekat dengan Stasiun Sumlaran dan Pasar Tradisional Sumlaran. Jumlah ruang kelas yang ada di SMA Muhammadiyah 8 Sukodadi adalah 6 yang terdiri dari X-IPA sampai XII IPA dan X-IPS sampai XII-IPS. SMA Muhammadiyah 8 Sukodadi juga memiliki 1 Ruang Kepala Sekolah, 1 Ruang Guru, 1 Ruang Tata Usaha, 1 Perpustakaan Sekolah, 5 WC, dan 1 Lapangan yang digunakan untuk Upacara Bendera.



Gambar 4. 1. Foto bangunan sekolah SMA Muhammadiyah 8 Sukodadi

Sekolah ini berkomitmen untuk mengembangkan potensi siswa melalui program akademik dan karakter yang berkelanjutan, serta menawarkan berbagai program studi dan ekstrakurikuler. Fasilitas yang dimiliki SMA Muhammadiyah 8 Sukodadi seperti perpustakaan, laboratorium komputer, laboratorium sains, dan aula serbaguna membuat lingkungan belajar menjadi kondusif.

Sekolah ini juga memiliki tenaga pendidik yang berkualitas dan berdedikasi untuk membantu siswa mencapai potensi terbaik mereka. SMA Muhammadiyah 8 Sukodadi menjadi salah satu sekolah menengah atas terbaik di Sukodadi, dan menjadi pilihan yang tepat bagi siswa yang ingin mendapatkan pendidikan yang berkualitas dan berlandaskan nilai-nilai keagamaan.

4.1.2. Perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi

Perpustakaan Sekolah SMA Muhammadiyah 8 Sukodadi adalah perpustakaan sekolah yang dikelola langsung oleh Tata Usaha SMA Muhammadiyah 8 Sukodadi. Perpustakaan Sekolah SMA Muhammadiyah 8 Sukodadi memiliki beberapa Rak Buku yang terdiri dari Buku Bacaan, Buku Bacaan Agama, Buku Bacaan Umum, Buku Cerita, Buku Pengetahuan Umum.

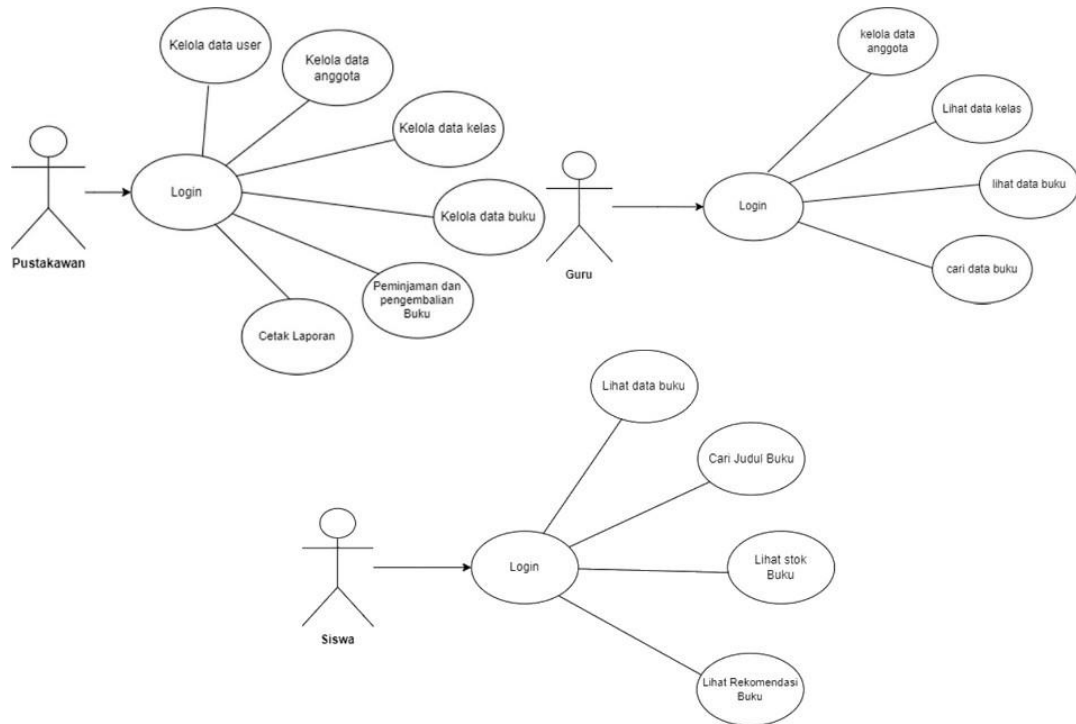
4.1.3. Data buku Perpustakaan SMA Muhammadiyah 8 Sukodadi

Perpustakaan Sekolah SMA Muhammadiyah 8 Sukodadi memiliki jumlah buku sebanyak 517 buku yang terdiri dari beberapa kategori buku seperti Biografi, Buku Bacaan, Buku Cerita, Buku Mata Pelajaran, Cerita Rakyat, Edukasi, Kamus, Novel, Religi, *Sci & Fic*, dan Sejarah. Buku yang terdapat pada Perpustakaan Sekolah SMA Muhammadiyah 8 Sukodadi adalah Buku Fisik.

4.2. Pembahasan

4.2.1. Perancangan Sistem

A. Use Case Diagram



Gambar 4. 2. Use case diagram

Tabel 4. 1. Deskripsi use case diagram

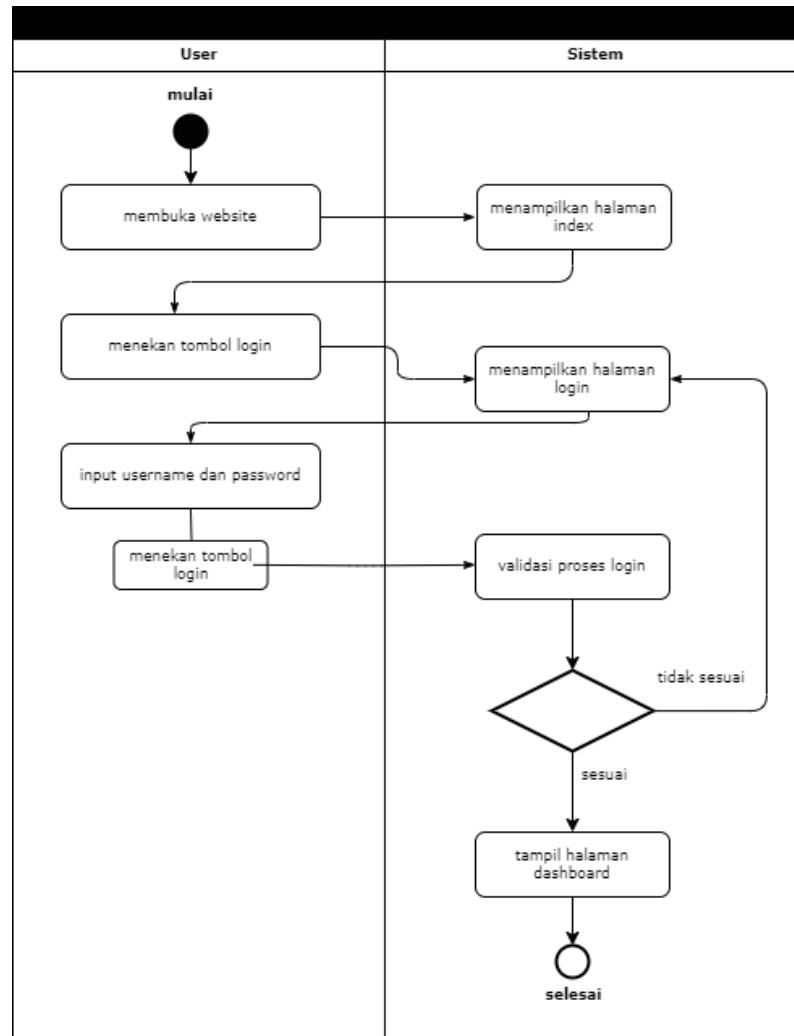
Aktor	Deskripsi
Pustakawan	<ol style="list-style-type: none"> 1. Melakukan <i>login</i> 2. Mengelola data <i>user</i> seperti menambah, melihat, mengedit dan menghapus data <i>user</i>. 3. Mengelola data anggota seperti menambah, melihat, mengubah dan menghapus data anggota. 4. Mengelola data kelas seperti menambah, melihat, mengubah dan menghapus data kelas. 5. Mengelola data buku seperti menambah, melihat, mengubah dan menghapus data buku. 6. Melayani peminjaman dan pemngembalian buku 7. Melakukan cetak laporan data anggota, data buku dan data transaksi.

Tabel 4. 1. Deskripsi *use case diagram* (lanjutan)

Aktor	Deskripsi
Guru	1. Melakukan <i>login</i> 2. Mengelola data anggota seperti menambah, melihat, mengubah dan menghapus data anggota. 3. Melihat data kelas. 4. Melihat data buku. 5. Mencari data buku
Siswa	1. Melakukan <i>login</i> 2. Mencari Judul buku. 3. Melihat stok buku. 4. Mendapatkan rekomendasi buku lain.

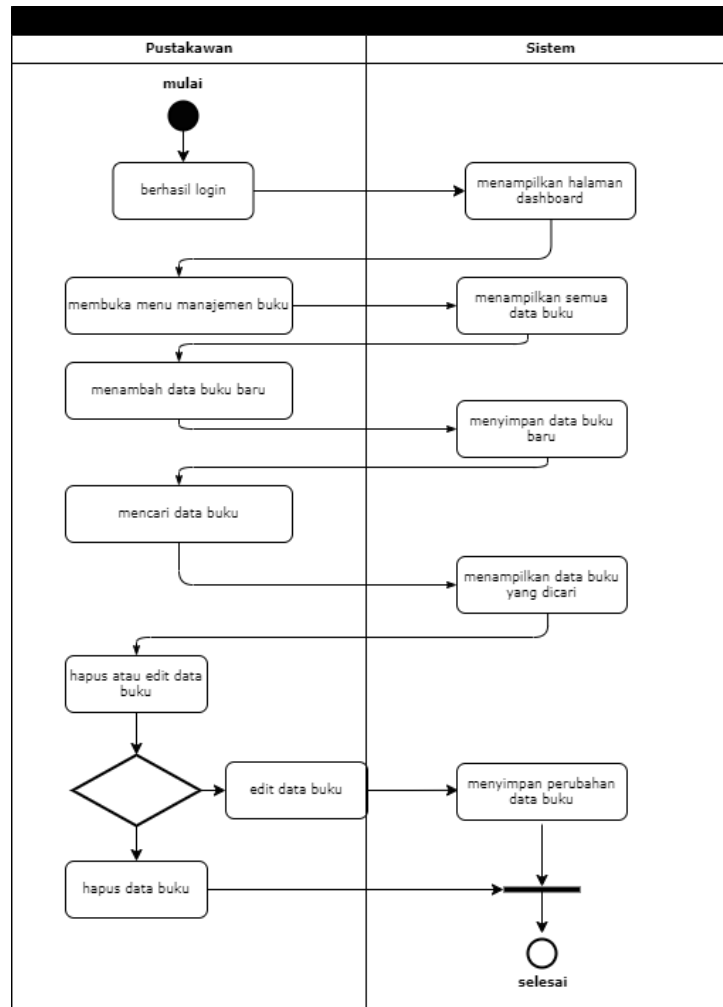
B. *Activity Diagram*

Activity diagram atau diagram aktivitas dalam bahasa Indonesia, berfungsi sebagai representasi visual yang mampu memodelkan proses-proses dalam suatu sistem. Dalam diagram ini, urutan proses sistem disajikan secara vertikal untuk memudahkan pemahaman. *Activity diagram* merupakan perkembangan dari *Use Case* yang menggambarkan alur dari berbagai aktivitas (Prasetya dkk., 2022). *Activity diagram* merepresentasikan grafis dari alur kerja atau rangkaian kegiatan dalam suatu sistem atau menu yang terdapat dalam perangkat lunak. Diagram ini digunakan untuk memodelkan serangkaian aktivitas yang terjadi dalam suatu sistem dan menggambarkan hubungan antara aktivitas tersebut. Fungsi dari *activity diagram* mencakup pengaturan urutan dan pengelompokan tampilan dari sistem atau antarmuka pengguna. Dalam *activity diagram*, setiap aktivitas dianggap memiliki desain antarmuka tampilan yang terkait dengannya. Hal ini memungkinkan pengguna untuk memahami bagaimana sistem berinteraksi dengan pengguna dan bagaimana setiap aktivitas mempengaruhi alur kerja keseluruhan. *Activity diagram* juga membantu dalam menganalisis dan memahami proses bisnis dalam suatu sistem. Dengan menggunakan diagram ini, kita dapat mengidentifikasi langkah-langkah yang diperlukan untuk menyelesaikan suatu tugas atau proses tertentu. Diagram ini juga memungkinkan untuk mengidentifikasi kegiatan yang dapat dilakukan secara paralel atau berurutan. Rancangan *activity diagram* pada penelitian dapat dilihat pada Gambar 4.3.



Gambar 4. 3. Activity diagram login

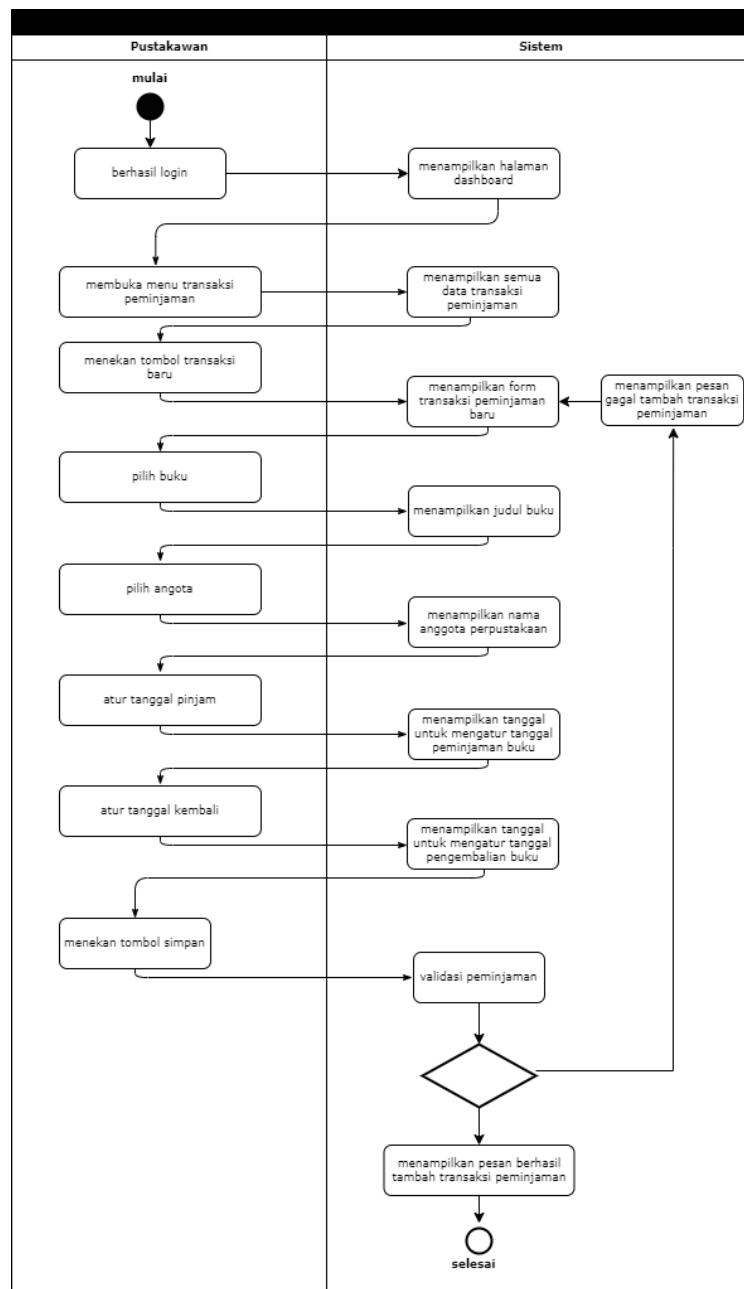
Proses pada Gambar 4.3. dimulai dengan *user* membuka halaman *website* dan sistem akan menampilkan halaman *index*. Selanjutnya *user* menekan tombol *login* setelah itu sistem akan menampilkan halaman *login*. Selanjutnya *user* memasukkan *username* dan *password* dan menekan tombol *login* selanjutnya sistem akan melakukan validasi data *username* dan *password* yang ada pada *database*. Jika *username* dan *password* sesuai, sistem akan menampilkan halaman *dashboard*. Jika *username* dan *password* tidak sesuai, sistem akan mengembalikan tampilan ke halaman *login*.



Gambar 4. 4. Activity diagram kelola buku

Proses pada Gambar 4.4. dimulai dengan *user* pustakawan berhasil melakukan *login* dan sistem akan menampilkan halaman *dashboard*. Setelah itu *user* membuka menu manajemen buku. Selanjutnya sistem akan menampilkan semua data buku yang ada pada *database*. Jika *user* menambah data buku, sistem akan menyimpan data buku ke dalam *database*.

Jika *user* ingin mencari data buku, sistem akan menampilkan data buku yang dicari dan *user* akan diberikan dua opsi yaitu ubah atau hapus data buku. Jika *user* mengubah data buku, sistem akan memperbarui data buku dan menyimpannya ke dalam *database*. Jika *user* menghapus data buku, sistem juga akan menghapus data buku yang terdapat dalam *database*.

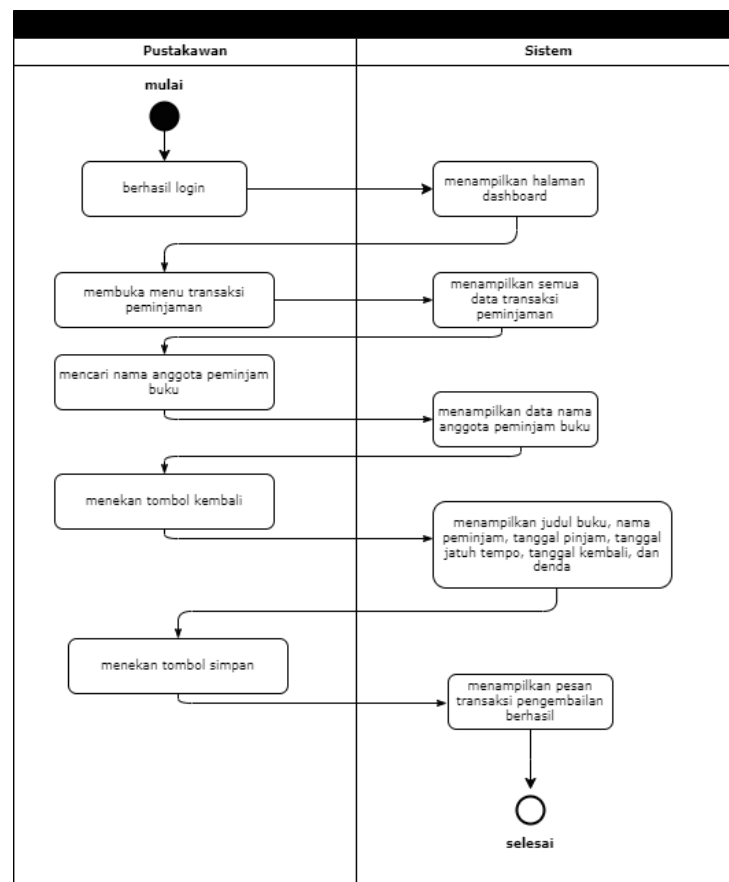


Gambar 4. 5. Activity diagram peminjaman buku

Proses pada Gambar 4.5. dimulai dengan *user* pustakawan berhasil melakukan *login* dan sistem akan menampilkan halaman *dashboard*. Setelah itu *user* membuka menu transaksi dan sistem menampilkan semua data transaksi yang ada. Setelah itu *user* menekan tombol transaksi baru untuk menambahkan transaksi. Sistem akan menampilkan form transaksi peminjaman baru. Setelah itu *user* akan memasukkan judul buku, nama anggota, mengatur tanggal pinjam, dan mengatur

tanggal kembali. *User* akan dibantu oleh sistem untuk menampilkan data judul buku dan nama anggota yang ada di dalam *database* saat memasukkan data.

Selanjutnya *user* menekan tombol simpan dan sistem akan melakukan validasi peminjaman. Jika sesuai sistem akan menampilkan pesan berhasil tambah transaksi peminjaman dan menampilkan semua data transaksi. Jika tidak sesuai, sistem akan menampilkan pesan gagal tambah transaksi peminjaman dan menampilkan form transaksi peminjaman baru.

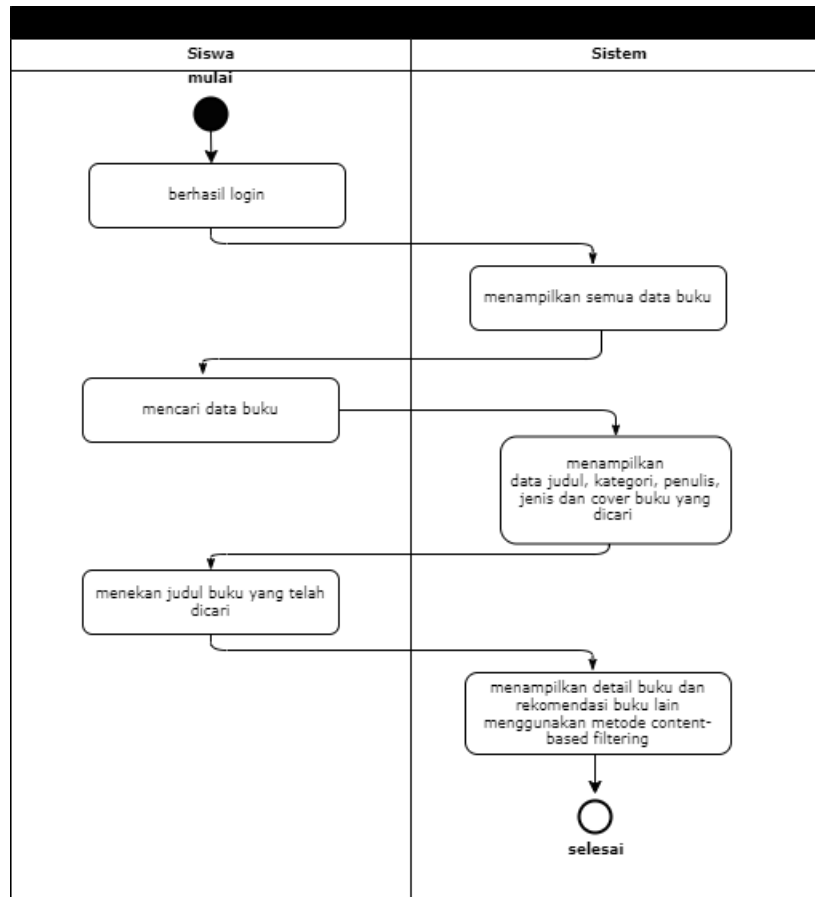


Gambar 4. 6. *Activity diagram* pengembalian buku

Proses pada Gambar 4.6. dimulai dengan *user* pustakawan berhasil melakukan *login* dan sistem akan menampilkan halaman *dashboard*. Setelah itu *user* pustakawan membuka menu transaksi peminjaman dan sistem menampilkan semua data transaksi peminjaman. Kemudian *user* akan mencari nama anggota peminjam buku.

Sistem akan menampilkan data nama anggota peminjam buku dan sistem akan memberikan opsi tiga tombol yaitu kembali, ubah, dan hapus. Setelah itu

menekan tombol kembali dan sistem menampilkan judul buku, nama peminjam, tanggal pinjam, dan denda. Setelah itu *user* menekan tombol simpan dan sistem menampilkan pesan transaksi pengembalian berhasil.



Gambar 4. 7. Activity diagram siswa

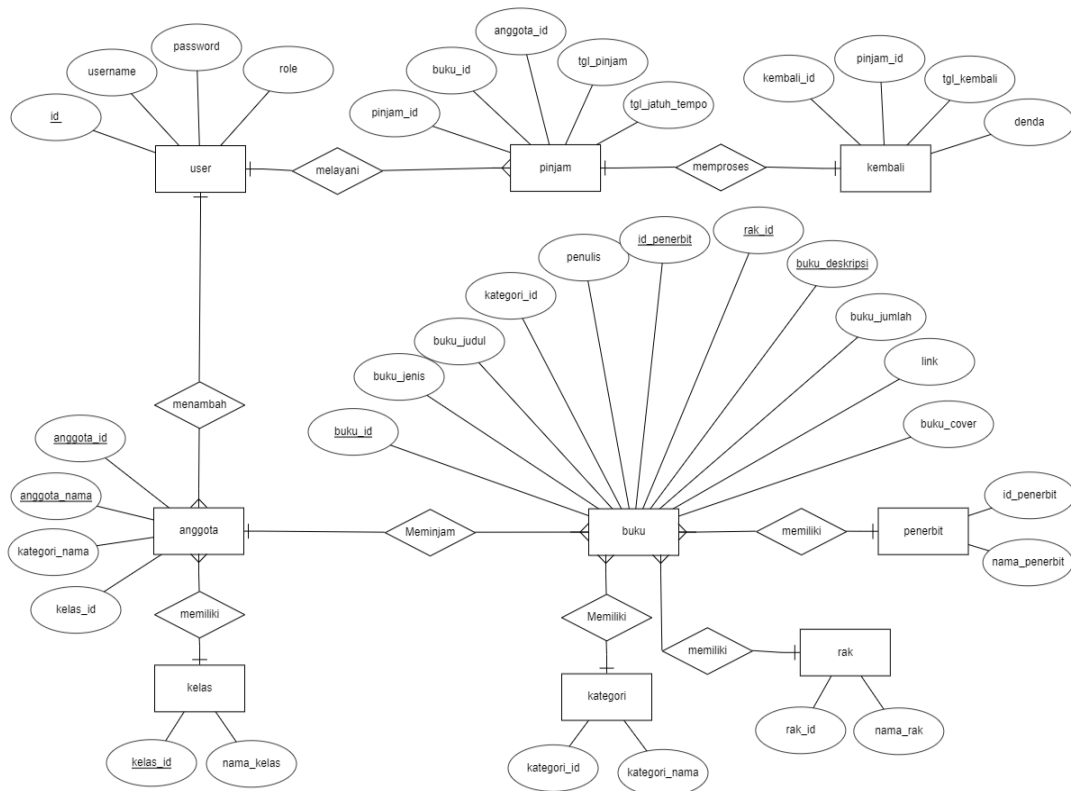
Proses pada Gambar 4.7. dimulai dengan *user* siswa melakukan *login*. Setelah berhasil melakukan *login* sistem akan menampilkan semua data buku yang ada dalam *database*. Selanjutnya *user* mencari data buku dan sistem akan menampilkan data buku seperti judul, kategori, penulis, jenis dan *cover* buku yang dicari. Setelah itu *user* menekan judul buku yang telah dicari untuk melihat detail buku dan sistem akan menampilkan detail buku dan menampilkan rekomendasi buku lain yang sesuai.

4.2.2. Perancangan Database

Untuk membuat sistem rekomendasi buku ini, langkah pertama yang harus dilakukan adalah membuat perancangan *database*. *Database* tersebut akan dibagi menjadi beberapa tabel yang akan digunakan untuk menyimpan semua informasi yang terkait dengan perpustakaan tersebut. Tabel-tabel tersebut akan dibuat dan diimplementasikan menggunakan *server* basis data bernama *MySQL Server*.

A. Entity Relationship Diagram

Entity relationship diagram (ERD) adalah representasi visual dari hubungan antara entitas dalam suatu *database*. Diagram ERD ini memudahkan untuk menggambarkan struktur logis dari basis data dengan jelas dan sederhana (Marudut dkk., 2021). Model *ERD* digunakan untuk memvisualisasikan dan mewakili bagaimana entitas dalam *database* saling terhubung dan berinteraksi satu sama lain. Gambar 4.8. menampilkan rancangan *ERD* yang dapat dilihat untuk memvisualisasikan hubungan tersebut.



Gambar 4. 8. *Entity Relationship Diagram* sistem

Sebagaimana ditunjukkan pada Gambar 4.8, Sistem rekomendasi buku ini terdiri dari beberapa tabel. Tabel-tabel tersebut diatur dengan saling terkait satu sama lain (*relation*). Relasi tabel adalah kaitan antara tabel yang mewakili hubungan antara objek. Berikut ini adalah rancangan relasi antar tabel yang nantinya akan digunakan :

- a. Relasi tabel *user* dengan tabel anggota

Tabel *user* menggunakan relasi *one to many* dengan tabel anggota karena *user* dengan *role superadmin* dan *admin* akan dapat menambahkan data lebih dari satu anggota.

- b. Relasi tabel anggota dengan tabel kelas

Tabel anggota menggunakan relasi *many to one* dengan tabel kelas karena beberapa anggota perpustakaan memiliki satu kelas yang sama.

- c. Relasi tabel anggota dengan tabel buku

Tabel anggota menggunakan relasi *one to many* dengan tabel buku karena satu anggota perpustakaan akan dapat meminjam lebih dari satu buku.

- d. Relasi tabel buku dengan tabel kategori

Tabel buku menggunakan relasi *many to one* dengan tabel kategori karena beberapa buku memiliki hanya satu kategori.

- e. Relasi tabel buku dengan tabel penerbit

Tabel buku menggunakan relasi *many to one* dengan tabel penerbit karena beberapa buku memiliki dapat memiliki hanya satu penerbit.

- f. Relasi tabel buku dengan tabel rak

Tabel buku menggunakan relasi *many to one* dengan tabel rak karena beberapa buku terdapat pada satu lokasi rak.

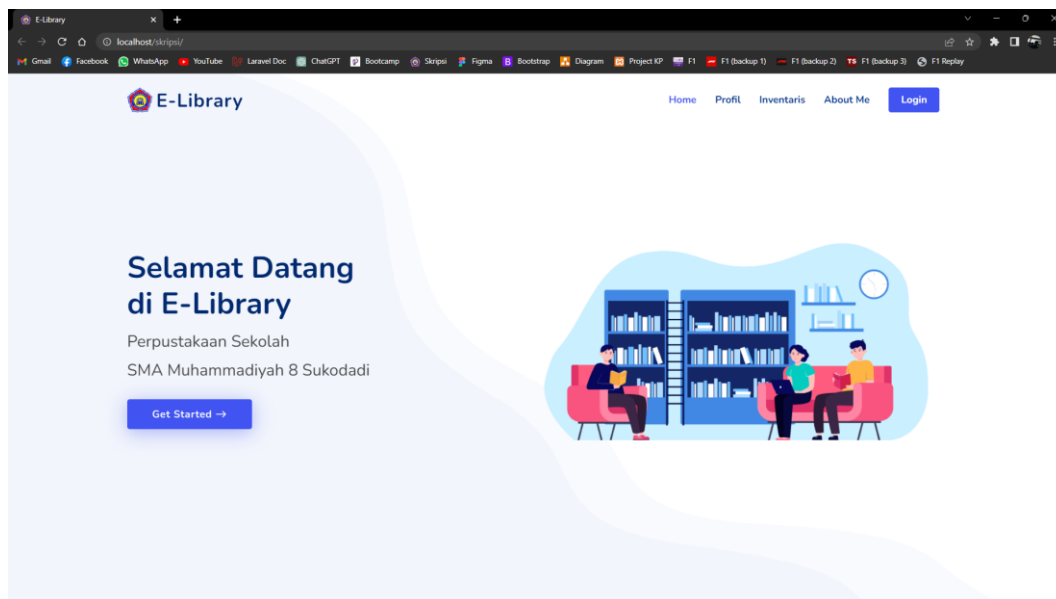
- g. Relasi tabel *user* dengan tabel pinjam

Tabel *user* menggunakan relasi *one to many* dengan tabel pinjam karena satu *user* dengan *role superadmin* akan dapat melayani peminjaman buku untuk banyak anggota.

h. Relasi tabel pinjam dengan tabel kembali

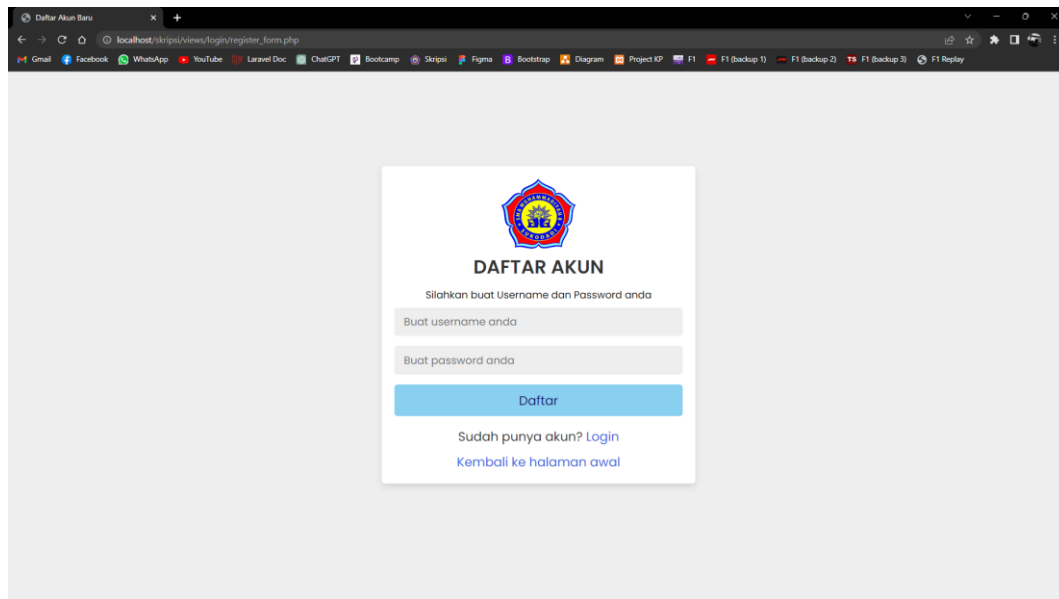
Tabel pinjam menggunakan relasi *one to one* dengan tabel kembali karena satu transaksi kembali akan diproses dengan mengambil data dari satu transaksi pinjam.

4.2.3. Implementasi Antarmuka Sistem



Gambar 4. 9. Tampilan halaman awal

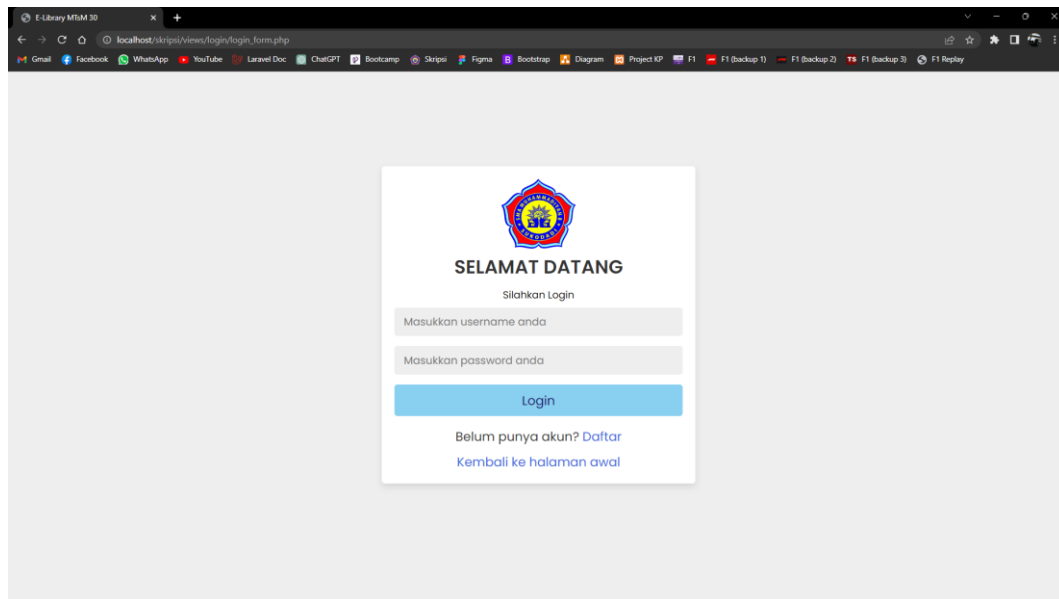
Pada Gambar 4.9. merupakan tampilan antarmuka halaman *index* sistem. Jadi ketika *user* membuka website, tampilan yang akan muncul pertama kali adalah halaman ini. Pada tampilan ini tersedia menu *home* sebagai tampilan awal yang memuat ucapan selamat datang dan nama perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi, profil yang akan menampilkan profil sekolah SMA Muhammadiyah 8 sukodadi dan perpustakaan sekolah SMA Muhammadiyah 8 Sukodadi, inventaris sebagai tampilan informasi yang akan menampilkan total jumlah data buku yang ada pada perpustakaan sekolah dan jumlah anggota perpustakaan yang terdaftar pada sistem, *about me* yang berfungsi sebagai tampilan profil penulis, dan tombol *login* yang berguna bagi *user* untuk melakukan *login* sesuai dengan *role* dengan memasukkan *username* dan *password*.



Gambar 4. 10. Tampilan halaman daftar akun

Pada Gambar 4.10 merupakan tampilan antarmuka yang menampilkan halaman daftar akun untuk *user* siswa. Halaman tersebut dirancang untuk digunakan oleh siswa dalam melakukan *login* ke dalam *website*. Pada halaman tersebut terdapat *textbox* yang digunakan untuk membuat *username*. Selain itu, halaman tersebut juga memiliki *textbox* yang berguna untuk membuat *password*. Terdapat tombol daftar di halaman tersebut yang berfungsi untuk menyimpan data *user* yang telah dibuat ke dalam *database*. Tampilan antarmuka ini memudahkan siswa untuk membuat akun mereka sendiri. Data yang diisi oleh siswa akan disimpan dan digunakan untuk *login* ke dalam *website*.

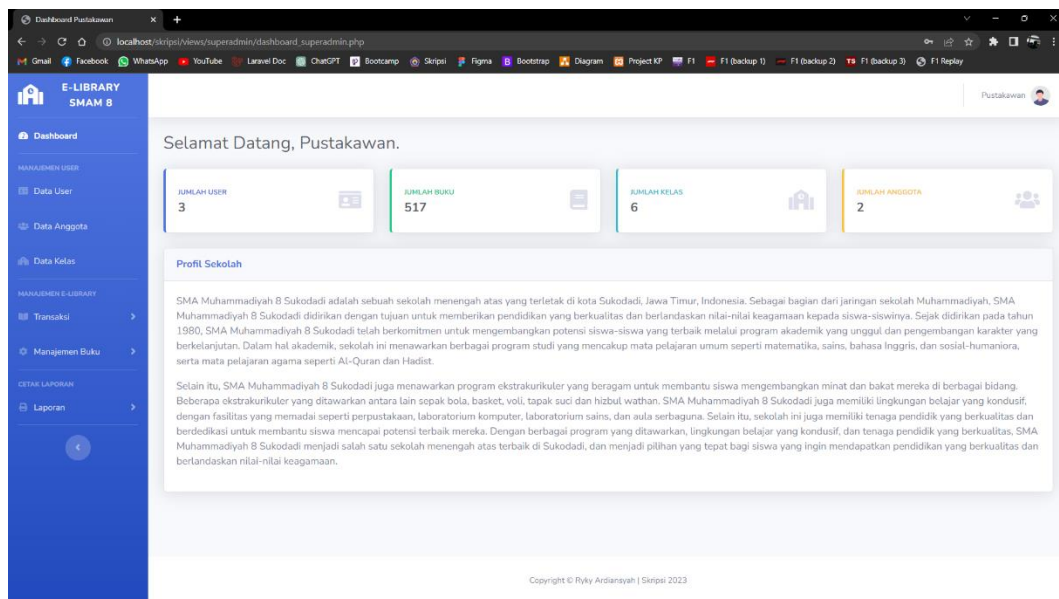
Jika proses menyimpan data *user* berhasil, maka *user* siswa akan di arahkan kembali ke halaman *login* dan menampilkan pesan berhasil menyimpan data. Namun jika penyimpanan data *user* gagal, tampilan akan tetap berada di halaman daftar akun dan akan ditampilkan pesan gagal menyimpan data. Selain itu, terdapat teks “*login*” yang berguna untuk membuka tampilan halaman *login* dan melakukan *login* jika siswa sudah mendaftar dan memiliki akun sebelumnya. Pada bagian bawah tampilan, terdapat teks “kembali ke halaman awal” yang dapat ditekan yang berfungsi untuk kembali ke halaman *index*.



Gambar 4. 11. Tampilan halaman *login*

Pada Gambar 4.11 adalah tampilan antarmuka yang menampilkan halaman *login* untuk semua *user*. Halaman ini memiliki sebuah *textbox* yang digunakan untuk memasukkan *username* yang telah dibuat sebelumnya. Selain itu, terdapat juga sebuah *textbox* yang digunakan untuk memasukkan *password* yang telah dibuat sebelumnya. Untuk mempermudah proses *login*, halaman ini dilengkapi dengan sebuah tombol "*login*" yang berguna untuk melakukan *login*.

Dengan menekan tombol tersebut, *user* dapat masuk ke website dengan menggunakan *username* dan *password* yang telah dimasukkan sesuai dengan *role*. Selain itu, terdapat teks "daftar" yang dapat ditekan. Jika teks tersebut ditekan, sistem akan membawa *user* ke halaman daftar akun yang berguna untuk *user* untuk melakukan pendaftaran akun dengan *role* siswa jika belum memiliki akun. Pada bagian bawah tampilan, terdapat teks "kembali ke halaman awal" yang dapat ditekan yang berfungsi untuk kembali ke halaman *index*. Dengan adanya tampilan ini, pengguna dapat memasukkan informasi pengguna mereka dan masuk ke akun mereka dengan mudah melalui halaman *login* ini. Halaman ini memberikan *user* kemudahan dalam mengakses dan menggunakan layanan yang tersedia di dalam sistem rekomendasi ini.

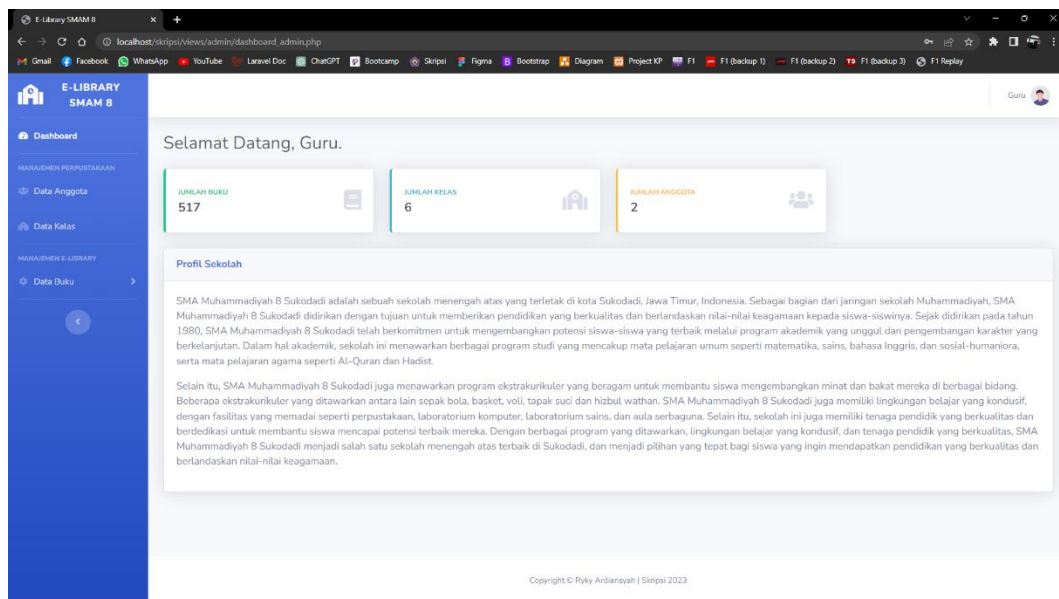


Gambar 4. 12. Tampilan halaman pustakawan

Pada Gambar 4.12. merupakan tampilan antarmuka yang ditujukan untuk *user* pustakawan. Ketika *user* pustakawan berhasil melakukan *login*, sistem akan menampilkan halaman *dashboard* yang memuat jumlah *user*, jumlah buku, jumlah kelas, dan jumlah anggota. Selain itu, pada tampilan awal *dashboard* juga terdapat profil singkat mengenai sekolah SMA Muhammadiyah 8 Sukodadi.

Pada tampilan ini, *user* pustakawan juga dapat mengelola data buku seperti melihat, menambah, mengubah, dan menghapus data buku, penerbit, dan rak dengan menekan tombol menu pada *sidebar* yang terletak di bagian kiri layar. Selain itu, *user* pustakawan juga dapat mengelola data kelas seperti melihat, menambah, mengubah, dan menghapus data kelas.

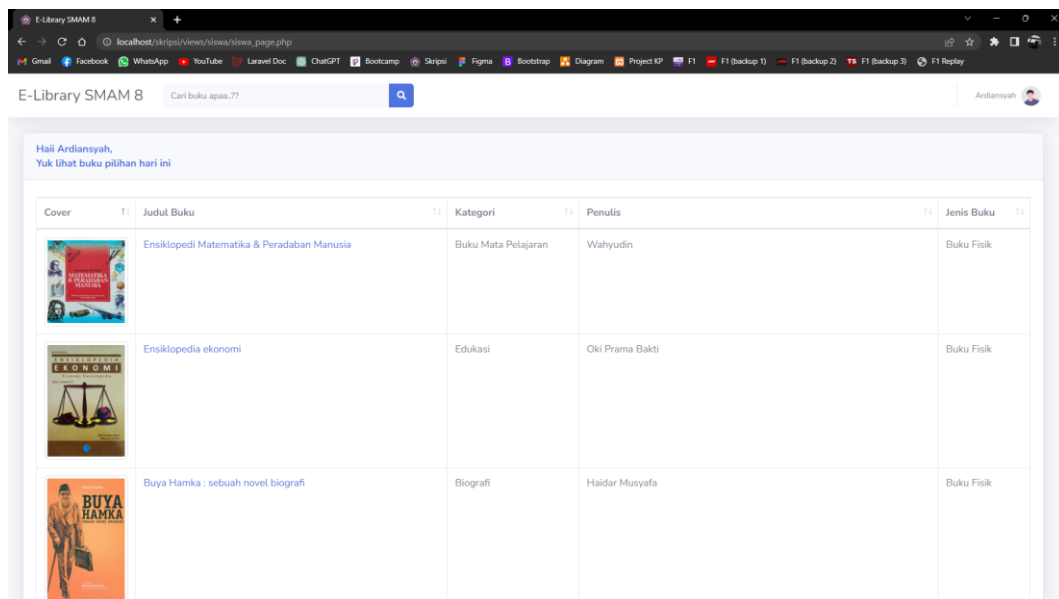
User pustakawan juga dapat menambah *user* siswa melalui menu yang tersedia dengan cara membukanya pada *sidebar* yang terletak disebelah kiri layar. Selain itu, *user* pustakawan dapat mengelola data anggota seperti melihat, menambah, mengubah, dan menghapus data anggota pada menu yang tersedia. *user* pustakawan juga dapat menambah *user* pustakawan baru melalui menu yang tersedia. *user* pustakawan juga dapat mencetak laporan data anggota, data buku dan data transaksi perpustakaan pada menu yang tersedia.



Gambar 4. 13. Tampilan halaman guru

Pada Gambar 4.13 merupakan sebuah tampilan antarmuka yang khusus ditujukan untuk *user* guru. Setelah *user* guru berhasil melakukan *login*, sistem akan menampilkan halaman *dashboard* yang berisi informasi jumlah buku yang tersedia, jumlah kelas yang ada, dan jumlah anggota yang terdaftar dalam sistem. Selain itu, pada tampilan awal *dashboard* juga terdapat profil singkat mengenai sekolah SMA Muhammadiyah 8 Sukodadi. Profil ini memberikan gambaran singkat tentang sekolah tersebut, termasuk sejarah, visi, dan misi sekolah.

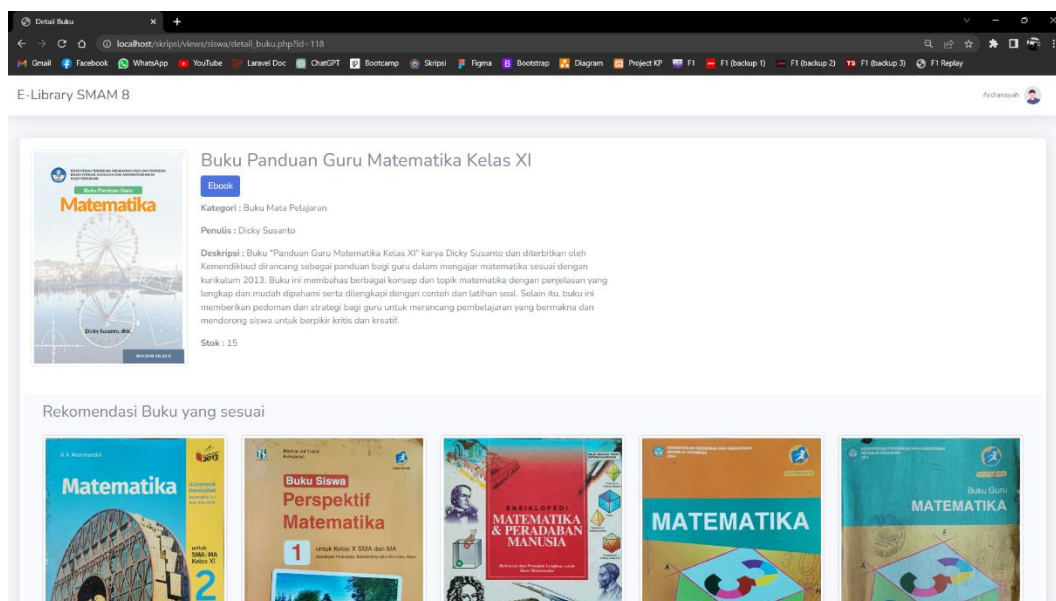
User guru memiliki kemampuan untuk mengelola data anggota, termasuk melihat, menambah, mengubah, dan menghapus data anggota sesuai kebutuhan. Selain itu, *user* guru juga dapat mengakses informasi tentang data kelas, termasuk nama kelas, jumlah anggota, dan informasi penting lainnya. *User* guru juga memiliki kemampuan untuk melihat informasi data buku seperti judul, kategori, penerbit, dan lokasi rak buku. Untuk mengakses halaman-halaman tersebut, *user* guru hanya perlu menekan menu di *sidebar* sebelah kiri layar. Menu tersebut akan membawa *user* guru ke halaman yang diinginkan, baik itu data anggota, data kelas, atau data buku. Dengan fitur-fitur ini, *user* guru memiliki kontrol penuh dalam mengelola data anggota, mengakses informasi kelas, dan melihat data buku yang relevan.



Gambar 4. 14. Tampilan halaman siswa

Pada Gambar 4.14 adalah tampilan halaman depan yang ditujukan untuk *user* siswa. Setelah *user* siswa berhasil melakukan *login*, sistem akan menampilkan halaman *dashboard* yang menampilkan semua data buku yang tersedia dalam *database* sistem. Halaman ini akan menampilkan *cover* buku, judul buku, kategori buku, penulis buku, dan jenis buku. Melalui tampilan ini, *user* siswa dapat melihat daftar buku yang tersedia. Penting untuk dicatat bahwa jika halaman diperbarui, daftar buku juga akan berubah sesuai dengan perubahan yang terjadi dalam tampilan sistem.

Pada bagian tampilan *navbar*, terdapat sebuah *textbox* yang berguna bagi *user* siswa. *Textbox* ini memungkinkan mereka untuk mencari buku dengan cara mengetik judul buku, kategori buku, penulis buku, atau jenis buku yang ingin mereka cari. Selain itu, pada bagian *navbar* juga terdapat tombol *dropdown* yang berguna bagi *user* siswa untuk melakukan *logout* akun mereka. Pada tampilan awal *user* siswa ini, judul buku yang terdapat dalam teks dapat ditekan oleh *user* siswa. Saat teks judul buku ditekan, halaman detail buku akan ditampilkan. Halaman detail buku ini akan memberikan informasi seperti judul buku, jenis buku, penulis buku, deskripsi buku, dan stok buku yang tersedia.



Gambar 4. 15. Tampilan detail buku

Pada gambar 4.15 menunjukkan antarmuka detail buku dalam tampilan *user* siswa. Tampilan ini menyajikan informasi seperti judul buku, jenis buku, kategori buku, penerbit, stok, dan deskripsi buku. Selain itu, dalam tampilan ini, metode *content-based filtering* akan digunakan untuk memberikan rekomendasi judul buku lain yang memiliki kesamaan dengan buku yang sedang dilihat. Perhitungan *cosine similarity* akan digunakan untuk menentukan skor kemiripan antara buku-buku tersebut, dan rekomendasi buku akan ditampilkan di bagian bawah halaman. Melalui antarmuka tersebut, *user* siswa dapat melihat secara rinci informasi tentang buku yang dipilih, termasuk informasi penting seperti judul, jenis, kategori, penerbit, stok, dan deskripsi buku. Tampilan ini memberikan kemudahan bagi *user* siswa untuk memperoleh gambaran lengkap tentang buku yang mereka minati.

Selain itu pada teks jenis buku, terdapat tombol yang jika jenis buku adalah *e-book*, *user* siswa dapat menekan tombol jenis buku tersebut untuk membaca buku secara *online* menggunakan jaringan internet. *User* siswa juga dapat mengunduh file buku dan mencetaknya jika diperlukan. Jika *user* siswa ingin kembali ke halaman awal, *user* siswa cukup menekan judul website pada *navbar*.

4.2.4. Pengujian Sistem

Pengujian pada sistem rekomendasi buku ini digunakan metode pengujian *black box testing*. Tujuan pengujian tersebut adalah untuk memastikan bahwa sistem yang telah dibangun memenuhi setiap fungsi yang telah ditetapkan pada tahap analisis dan perancangan sistem. Pengujian dilakukan dengan mengobservasi hasil eksekusi melalui data uji dan memeriksa fungsi-fungsi yang ada dalam sistem. Pengujian sistem ini mencakup autentifikasi pengguna, pengujian masukan sistem, dan pengujian keluaran sistem. Informasi lengkap mengenai daftar pengujian *black box testing* yang dilakukan dapat ditemukan dalam Tabel 4.2.

Tabel 4. 2. Hasil pengujian sistem

Nama Kasus Uji	Proses	Keterangan
<i>Login</i>	Menampilkan halaman login	Berhasil masuk kedalam sistem sesuai dengan <i>role user</i> .
Menu <i>input</i> data	a. Tambah data <i>user</i> b. Tambah data kelas c. Tambah data anggota d. Tambah data buku e. Tambah data kategori f. Tambah data penerbit g. Tambah data rak	Berhasil menambahkan data dan memasukkan data tersebut kedalam <i>database</i> .
Menu <i>export</i> data	a. Cetak data anggota b. Cetak data buku c. Cetak data transaksi	Berhasil melakukan <i>export</i> data menjadi file dengan ekstensi pdf
Menu rekomendasi	Menampilkan rekomendasi buku pada halaman detail buku	Berhasil menampilkan 5 rekomendasi buku dengan skor tertinggi hingga terendah.

Berdasarkan hasil *black box testing* yang telah dilakukan pada Tabel 4.2, dapat disimpulkan bahwa sistem berhasil menjalankan prosesnya dan menghasilkan keluaran yang sesuai dengan perancangan sistem. Seluruh kebutuhan sistem telah terpenuhi dan tidak ditemukan adanya kesalahan pada fungsi-fungsi yang ada.

4.2.5. Analisa Metode

Dalam penelitian ini, dilakukan evaluasi terhadap metode yang digunakan dengan menggunakan dua sampel judul buku yang nantinya akan dihitung tingkat kemiripannya menggunakan perhitungan pembobotan *TF-IDF* dan *cosine similarity*. proses ini digunakan untuk mengukur tingkat keakuratan rekomendasi buku yang diberikan oleh sistem. Informasi lengkap mengenai data sampel buku yang digunakan ditunjukkan pada Tabel 4.3.

Tabel 4. 3. Data sampel buku yang digunakan

Buku 1	Buku 2
Buku Panduan Guru Matematika Kelas XI Buku Mata Pelajaran Dicky Susanto Buku "Panduan Guru Matematika Kelas XI" karya Dicky Susanto dan diterbitkan oleh Kemendikbud dirancang sebagai panduan bagi guru dalam mengajar matematika sesuai dengan kurikulum 2013. Buku ini membahas berbagai konsep dan topik matematika dengan penjelasan yang lengkap dan mudah dipahami serta dilengkapi dengan contoh dan latihan soal. Selain itu, buku ini memberikan pedoman dan strategi bagi guru untuk merancang pembelajaran yang bermakna dan mendorong siswa untuk berpikir kritis dan kreatif.	Matematika 2 Kelas XI Buku Mata Pelajaran B.K. Noormandiri Buku Matematika Kelompok Peminatan Ilmu-ilmu Alam untuk SMA/MA kelas XI yang ditulis oleh Noormandiri dan diterbitkan oleh Erlangga, memuat materi matematika yang khusus dibutuhkan oleh siswa yang memilih jurusan IPA. Buku ini berisi materi matematika yang dikemas dengan bahasa yang mudah dipahami dan dilengkapi dengan contoh soal yang relevan. Tujuannya adalah untuk membantu siswa memahami konsep-konsep matematika konteks ilmu pengetahuan alam.

A. Preprocessing

Tahapan pertama yaitu *preprocessing*. Disini tahapan seperti *case folding*, *tokenizing*, *filtering (stopword removal)*, dan *stemming* akan dilakukan untuk mengolah kata yang telah diambil pada data sample sebelumnya.

1. Case folding

Proses *case folding* digunakan untuk mengkonversi huruf kapital pada kata-kata atau kalimat dalam dokumen menjadi huruf kecil. Tujuan dari proses ini adalah mencapai keseragaman dalam setiap kata yang terdapat dalam dokumen data buku. proses tahap ini dapat dilihat pada Tabel 4.4.

Tabel 4. 4. Proses *case folding*

Sebelum	Sesudah
<p>Buku Panduan Guru Matematika Kelas XI Buku Mata Pelajaran Dicky Susanto Buku "Panduan Guru Matematika Kelas XI" karya Dicky Susanto dan diterbitkan oleh Kemendikbud dirancang sebagai panduan bagi guru dalam mengajar matematika sesuai dengan kurikulum 2013. Buku ini membahas berbagai konsep dan topik matematika dengan penjelasan yang lengkap dan mudah dipahami serta dilengkapi dengan contoh dan latihan soal. Selain itu, buku ini memberikan pedoman dan strategi bagi guru untuk merancang pembelajaran yang bermakna dan mendorong siswa untuk berpikir kritis dan kreatif.</p>	<p>buku panduan guru matematika kelas xi buku mata pelajaran dicky susanto buku "panduan guru matematika kelas xi" karya dicky susanto dan diterbitkan oleh kemendikbud dirancang sebagai panduan bagi guru dalam mengajar matematika sesuai dengan kurikulum 2013. buku ini membahas berbagai konsep dan topik matematika dengan penjelasan yang lengkap dan mudah dipahami serta dilengkapi dengan contoh dan latihan soal. selain itu, buku ini memberikan pedoman dan strategi bagi guru untuk merancang pembelajaran yang bermakna dan mendorong siswa untuk berpikir kritis dan kreatif.</p>
<p>Matematika 2 Kelas XI Buku Mata Pelajaran B.K. Noormandiri Buku Matematika Kelompok Peminatan Ilmu-ilmu Alam untuk SMA/MA kelas XI yang ditulis oleh Noormandiri dan diterbitkan oleh Erlangga, memuat materi matematika yang khusus dibutuhkan oleh siswa yang memilih jurusan IPA. Buku ini berisi materi matematika yang dikemas dengan bahasa yang mudah dipahami dan dilengkapi dengan contoh soal yang relevan. Tujuannya adalah untuk membantu siswa memahami konsep-konsep matematika konteks ilmu pengetahuan alam.</p>	<p>matematika 2 kelas xi buku mata pelajaran b.k. noormandiri buku matematika kelompok peminatan ilmu-ilmu alam untuk sma/ma kelas xi yang ditulis oleh noormandiri dan diterbitkan oleh erlangga, memuat materi matematika yang khusus dibutuhkan oleh siswa yang memilih jurusan ipa. buku ini berisi materi matematika yang dikemas dengan bahasa yang mudah dipahami dan dilengkapi dengan contoh soal yang relevan. tujuannya adalah untuk membantu siswa memahami konsep-konsep matematika konteks ilmu pengetahuan alam.</p>

2. Tokenizing

Tokenizing adalah proses yang digunakan untuk memisahkan string input menjadi token atau kata-kata berdasarkan spasi atau karakter pemisah yang ditentukan. Pada langkah ini, data yang terkait dengan setiap buku akan diolah untuk menghilangkan simbol, tanda baca, angka, *backslash*, karakter *non-ASCII*, *URL*, *whitespace leading* dan *trailing*, serta mengubah *multiple whitespace* menjadi *single whitespace*. Proses *Tokenizing* dapat dilihat pada Tabel 4.5.

Tabel 4. 5. Proses *tokenizing*

Sebelum	Sesudah
buku panduan guru matematika kelas xi buku mata pelajaran dicky susanto buku "panduan guru matematika kelas xi" karya dicky susanto dan diterbitkan oleh kemendikbud dirancang sebagai panduan bagi guru dalam mengajar matematika sesuai dengan kurikulum 2013. buku ini membahas berbagai konsep dan topik matematika dengan penjelasan yang lengkap dan mudah dipahami serta dilengkapi dengan contoh dan latihan soal. selain itu, buku ini memberikan pedoman dan strategi bagi guru untuk merancang pembelajaran yang bermakna dan mendorong siswa untuk berpikir kritis dan kreatif.	[buku, panduan, guru, matematika, kelas xi, buku, mata, pelajaran, dicky, susanto, buku, panduan, guru, matematika, kelas, xi, karya, dicky, susanto, dan, diterbitkan, oleh, kemendikbud, dirancang, sebagai, panduan, bagi, guru, dalam, mengajar, matematika, sesuai, dengan, kurikulum, 2013, buku, ini, membahas, berbagai, konsep, dan, topik, matematika, dengan, penjelasan, yang, lengkap, dan, mudah, dipahami, serta, dilengkapi, dengan, contoh, dan, latihan, soal, selain, itu, buku, ini, memberikan, pedoman, dan, strategi, bagi, guru, untuk, merancang, pembelajaran, yang, bermakna, dan, mendorong, siswa, untuk, berpikir, kritis, dan, kreatif]
matematika 2 kelas xi buku mata pelajaran b.k. noormandiri buku matematika kelompok peminatan ilmu-ilmu alam untuk sma/ma kelas xi yang ditulis oleh noormandiri dan diterbitkan oleh erlangga, memuat materi matematika yang khusus dibutuhkan oleh siswa yang memilih jurusan ipa. buku ini berisi materi matematika yang	[matematika, 2, kelas, xi, buku, mata, pelajaran, b, k, noormandiri, buku, matematika, kelompok, peminatan, ilmu-ilmu, alam, untuk, sma/ma, kelas, xi, yang, ditulis, oleh, noormandiri, dan, diterbitkan, oleh, erlangga, memuat, materi, matematika, yang, khusus, dibutuhkan, oleh, siswa, yang, memilih, jurusan, ipa. Buku, ini, berisi, materi, matematika, yang,

Tabel 4. 5. Proses *tokenizing* (lanjutan)

Sebelum	Sesudah
dikemas dengan bahasa yang mudah dipahami dan dilengkapi dengan contoh soal yang relevan. tujuannya adalah untuk membantu siswa memahami konsep-konsep matematika konteks ilmu pengetahuan alam.	dikemas, dengan, bahasa, yang, mudah, dipahami, dan, dilengkapi, dengan, contoh, soal, yang, relevan, tujuannya, adalah, untuk, membantu, siswa, memahami, konsep-konsep, matematika, konteks, ilmu, pengetahuan, alam]

3. *Filtering*

Proses *filtering* dilakukan untuk mengidentifikasi kata-kata kunci yang relevan dari dokumen data buku dengan menghapus kata-kata yang tidak memiliki deskripsi, seperti kata penghubung, kata depan, dan kata ganti seperti "di", "yang", "dan", "ke", dan "dari". Proses dari tahapan *filtering* dapat dilihat pada Tabel 4.6.

Tabel 4. 6. Proses *filtering*

Sebelum	Sesudah
[buku, panduan, guru, matematika, kelas xi, buku, mata, pelajaran, dicky, susanto, buku, panduan, guru, matematika, kelas, xi, karya, dicky, susanto, dan, diterbitkan, oleh, kemendikbud, dirancang, sebagai, panduan, bagi, guru, dalam, mengajar, matematika, sesuai, dengan, kurikulum, 2013, buku, ini, membahas, berbagai, konsep, dan, topik, matematika, dengan, penjelasan, yang, lengkap, dan, mudah, dipahami, serta, dilengkapi, dengan, contoh, dan, latihan, soal, selain, itu, buku, ini, memberikan, pedoman, dan, strategi, bagi, guru, untuk, merancang, pembelajaran, yang, bermakna, mendorong, siswa, untuk, berpikir, kritis, dan, kreatif]	[buku, panduan, guru, matematika, kelas xi, buku, mata, pelajaran, dicky, susanto, buku, panduan, guru, matematika, kelas, xi, karya, dicky, susanto, diterbitkan, oleh, kemendikbud, rancang, sebagai, panduan, bagi, guru, dalam, mengajar, matematika, sesuai, dengan, kurikulum, 2013, buku, ini, membahas, berbagai, konsep, topik, matematika, dengan, penjelasan, lengkap, mudah, pahami, serta, lengkapi, dengan, contoh, latihan, soal, selain, itu, buku, ini, memberikan, pedoman, strategi, bagi, guru, untuk, merancang, pembelajaran, bermakna, mendorong, siswa, untuk, berpikir, kritis, kreatif]
[matematika, 2, kelas, xi, buku, mata, pelajaran, b, k, noormandiri, buku, matematika, kelompok, peminatan, ilmu-ilmu, alam, untuk, sma/ma, kelas, xi, yang, ditulis, oleh,	[matematika, 2, kelas, xi, buku, mata, pelajaran, b, k, noormandiri, buku, matematika, kelompok, peminatan, ilmu-ilmu, alam, untuk, sma/ma, kelas, xi, tulis, oleh, noormandiri,

Tabel 4. 6. Proses *filtering* (lanjutan)

Sebelum	Sesudah
noormandiri, dan, diterbitkan, oleh, erlangga, memuat, materi, matematika, yang, khusus, dibutuhkan, oleh, siswa, yang, memilih, jurusan, ipa. Buku, ini, berisi, materi, matematika, yang, dikemas, dengan, bahasa, yang, mudah, dipahami, dan, dilengkapi, dengan, contoh, soal, yang, relevan, tujuannya, adalah, untuk, membantu, siswa, memahami, konsep-konsep, matematika, konteks, ilmu, pengetahuan, alam]	terbitkan, oleh, erlangga, memuat, materi, matematika, khusus, butuhkan, oleh, siswa, memilih, jurusan, ipa. Buku, berisi, materi, matematika, dikemas, dengan, bahasa, mudah, pahami, lengkapi, dengan, contoh, soal, relevan, tujuannya, adalah, untuk, membantu, siswa, memahami, konsep-konsep, matematika, konteks, ilmu, pengetahuan, alam]

4. *Stemming*

Proses stemming bertujuan untuk menemukan kata dasar dari kata-kata yang telah melalui proses penyaringan. *Stemming* berperan dalam menghilangkan awalan, akhiran, sisipan, serta kombinasi awalan dan akhiran dari suatu kata. Proses *stemming* ini dilakukan menggunakan *library* Sastrawi, yang memiliki kemampuan untuk mengubah kata-kata yang mengandung imbuhan dalam dokumen data buku menjadi bentuk dasarnya. Proses *stemming* dapat ditemukan dalam Tabel 4.7.

Tabel 4. 7. Proses *steeming*

Sebelum	Sesudah
[buku, panduan, guru, matematika, kelas xi, buku, mata, pelajaran, dicky, susanto, buku, panduan, guru, matematika, kelas, xi, karya, dicky, susanto, terbitkan, oleh, kemendikbud, rancang, sebagai, panduan, bagi, guru, dalam, mengajar, matematika, sesuai, dengan, kurikulum, 2013, buku, ini, membahas, berbagai, konsep, topik, matematika, dengan, penjelasan, lengkap, mudah, pahami, serta, lengkapi, dengan, contoh, latihan, soal, selain, itu, buku, ini, memberikan, pedoman, strategi, bagi, guru, untuk, merancang, pembelajaran, bermakna, mendorong, siswa, untuk, berpikir, kritis, kreatif]	[buku, panduan, guru, matematika, kelas xi, buku, mata, pelajaran, dicky, susanto, buku, panduan, guru, matematika, kelas, xi, karya, dicky, susanto, terbitkan, kemendikbud, rancang, panduan, guru, mengajar, matematika, sesuai, kurikulum, 2013, buku, membahas, berbagai, konsep, topik, matematika, penjelasan, lengkap, mudah, pahami, serta, lengkapi, dengan, contoh, latihan, soal, buku, memberikan, pedoman, strategi, guru, merancang, pembelajaran, bermakna, dan, mendorong, siswa, berpikir, kritis, kreatif]

Tabel 4. 7. Proses *steeming* (lanjutan)

Sebelum	Sesudah
[matematika, 2, kelas, xi, buku, mata, pelajaran, b, k, noormandiri, buku, matematika, kelompok, peminatan, ilmu-ilmu, alam, untuk, sma/ma, kelas, xi, tulis, oleh, noormandiri, terbitkan, oleh, erlangga, memuat, materi, matematika, khusus, butuhkan, oleh, siswa, memilih, jurusan, ipa. Buku, berisi, materi, matematika, dikemas, dengan, bahasa, mudah, pahami, lengkapi, dengan, contoh, soal, relevan, tujuannya, adalah, untuk, membantu, siswa, memahami, konsep-konsep, matematika, konteks, ilmu, pengetahuan, alam]	[matematika, 2, kelas, xi, buku, mata, pelajaran, noormandiri, buku, matematika, kelompok, peminatan, ilmu-ilmu, alam, sma/ma, kelas, xi, tulis, noormandiri, terbitkan, erlangga, memuat, materi, matematika, khusus, butuhkan, siswa, memilih, jurusan, ipa. Buku, berisi, materi, matematika, dikemas, bahasa, mudah, pahami, lengkapi, contoh, soal, relevan, tujuannya, adalah, membantu, siswa, memahami, konsep-konsep, matematika, konteks, ilmu, pengetahuan, alam]

B. Pembobotan *TF-IDF*

Pada tahap ini, semua kata yang ada pada kriteria akan dihitung dan diberi bobot menggunakan rumus 2.2. Proses perhitungan pembobotan *TF-IDF* dapat dilihat pada Tabel 4.8.

Tabel 4. 8. Pembobotan *TF-IDF*

Kata	Buku 1	Buku 2	df	idf	tf-idf	
					Buku 1	Buku 2
Buku	2	5	7	$\text{Log}(2/7) = 0,3979$	0	0,3979
panduan	4	0	4	$\text{Log}(2/4) = 0,3010$	0,3010	0
guru	4	0	4	$\text{Log}(2/4) = 0,3010$	0,3010	0
matematika	2	5	7	$\text{Log}(2/7) = 0,5440$	0	0,5440
kelas	2	2	4	$\text{Log}(2/4) = 0,3010$	0,1505	0,1505
xi	2	2	4	$\text{Log}(2/4) = 0,3010$	0,1505	0,1505
Mata	1	1	2	$\text{Log}(2/2) = 0$	0	0
Pelajaran	1	1	2	$\text{Log}(2/2) = 0$	0	0
Dicky	2	0	2	$\text{Log}(2/2) = 0$	0	0
Susanto	2	0	2	$\text{Log}(2/2) = 0$	0	0
terbit	1	1	2	$\text{Log}(2/2) = 0$	0	0
oleh	2	3	5	$\text{Log}(2/5) = 0,3979$	0	0,3979
erlangga	0	2	1	$\text{Log}(2/1) = 0,3010$	0	0,3010
kemendikbud	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
rancang	1	2	3	$\text{Log}(2/3) = 0,1760$	0	0,1760

Tabel 4. 8. Pembobotan *TF-IDF* (lanjutan)

Kata	Buku 1	Buku 2	df	idf	tf-idf	
					Buku 1	Buku 2
sebagai	1	1	2	$\text{Log}(2/2) = 0$	0	0
bagi	2	0	2	$\text{Log}(2/2) = 0$	0	0
dalam	1	1	2	$\text{Log}(2/2) = 0$	0	0
mengajar	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
sesuai	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
dengan	3	2	5	$\text{Log}(2/5) = 0,3979$	0	0,3979
kurikulum	1	1	2	$\text{Log}(2/2) = 0$	0	0
2013	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
membahas	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
berbagai	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
konsep	2	1	3	$\text{Log}(2/3) = 0,1760$	0,1760	0
topik	1	1	2	$\text{Log}(2/2) = 0$	0	0
penjelasan	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
lengkap	2	1	3	$\text{Log}(2/3) = 0,1760$	0,1760	0
mudah	1	1	2	$\text{Log}(2/2) = 0$	0	0
paham	1	1	2	$\text{Log}(2/2) = 0$	0	0
serta	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
contoh	1	1	2	$\text{Log}(2/2) = 0$	0	0
latihan	1	0	1	$\text{Log}(2/2) = 0$	0,3010	0
soal	1	1	2	$\text{Log}(2/2) = 0$	0	0
selain	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
beri	1	1	2	$\text{Log}(2/2) = 0$	0	0
pedoman	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
strategi	1	1	1	$\text{Log}(2/2) = 0$	0	0
belajar	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
makna	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
dorong	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
siswa	1	0	2	$\text{Log}(2/1) = 0$	0,3010	0
pikir	1	0	1	$\text{Log}(2/1) = 0,3010$	0,3010	0
kritis	1	1	2	$\text{Log}(2/2) = 0$	0	0
kreatif	1	1	2	$\text{Log}(2/2) = 0$	0	0

C. Pembobotan *cosine similarity*

Pada tahap ini, perhitungan *cosine similarity* digunakan sebagai metode untuk mengukur sejauh mana kedua vektor atau dokumen dalam ruang vektor memiliki kesamaan. Dengan menggunakan metode ini, sistem dapat menilai tingkat kemiripan antara satu data buku dengan data buku lainnya. Kata yang diambil dalam perhitungan *cosine similarity* ini sudah melalui proses *preprocessing*. Proses pembobotan *cosine similarity* dilihat pada tabel 4.9.

Tabel 4. 9. Pembobotan *cosine similarity*

No	Term	T(A)	T(B)
1	buku	2	5
2	matematika	2	5
3	oleh	2	3
4	erlangga	0	2
5	rancang	1	2
6	kurikulum	1	1
7	konsep	2	1
8	topik	1	1
9	strategi	1	1
10	kritis	1	1
11	kreatif	1	1
12	siswa	1	0

D. Perhitungan skor *cosine similarity*

Setelah dilakukan pembobotan *cosine similarity*, didapatkan Vektor A dan Vektor B. Dalam konteks ini, Vektor A dan Vektor B digunakan sebagai representasi untuk mengidentifikasi jumlah nilai yang terkait dengan setiap kata unik dalam dokumen. Selanjutnya, perhitungan akan dilakukan menggunakan metode *cosine similarity*.

a) Vektor A = (2,2,2,0,1,1,2,1,1,1,1,1)

b) Vektor B = (5,5,3,2,2,1,1,1,1,1,1,0)

Dalam mencari hasil perhitungan antara Vektor A dan Vektor B yang terdapat pada Tabel 4.9, dilakukan perhitungan kemiripan. Dalam hal ini, rumus 2.4. digunakan untuk mengetahui persamaan antara Vektor A dan Vektor B. proses perhitungan skor dapat dilihat dalam Tabel 4.10.

Tabel 4. 10. Perhitungan skor

$$\begin{aligned}
 \text{similarity}(A, B) &= \frac{2 + 2 + 2 + 0 + 1 + 1 + 2 + 1 + 1 + 1 + 1 + 1}{\sqrt{2^2 + 2^2 + 2^2 + 0^2 + 1^2 + 1^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2} \times \sqrt{5^2 + 5^2 + 3^2 + 2^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 0^2}} \\
 \text{similarity}(A, B) &= \frac{2 + 2 + 2 + 0 + 1 + 1 + 2 + 1 + 1 + 1 + 1 + 1}{\sqrt{4 + 4 + 4 + 0 + 1 + 1 + 4 + 1 + 1 + 1 + 1 + 1} \times \sqrt{25 + 25 + 9 + 4 + 4 + 1 + 1 + 1 + 1 + 1 + 0}} \\
 \text{similarity}(A, B) &= \frac{15}{\sqrt{23} \times \sqrt{76}} \\
 \text{similarity}(A, B) &= \frac{15}{41,8} \\
 \text{similarity}(A, B) &= \frac{15}{41,8} = 0,358
 \end{aligned}$$

Setelah berhasil mengimplementasikan perhitungan *cosine similarity*, didapatkan hasil yang menunjukkan tingkat kemiripan antara Vektor A dan Vektor B dalam ruang vektor sebesar 0,358. Angka tersebut mengindikasikan bahwa perhitungan yang dilakukan oleh sistem berhasil dan memberikan rekomendasi yang sesuai dengan perhitungan skor menggunakan metode *cosine similarity*.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil yang telah diperoleh dari penelitian dalam pembuatan sistem rekomendasi buku menggunakan metode *content-based filtering*, dapat disimpulkan bahwa:

- A. Sistem rekomendasi buku perpustakaan SMA Muhammadiyah 8 Sukodadi dibangun menggunakan bahasa pemrograman *PHP* dan *Javascript*.
- B. Sistem ini diharapkan akan memberikan kemudahan bagi pustakawan dalam mengelola transaksi buku dan menghasilkan laporan data buku yang dibutuhkan. Keberadaan sistem ini memungkinkan pembuatan dan pencetakan laporan dilakukan tanpa perlu lagi menggabungkan data yang diperlukan.
- C. Sistem rekomendasi buku SMA Muhammadiyah 8 Sukodadi memiliki akurasi yang sesuai setelah dihitung menggunakan perhitungan *cosine similarity*. hasil dari perhitungan menunjukkan persamaan vektor A dan vektor B sebesar 0,358 yang mengindikasikan bahwa sistem dapat memberikan rekomendasi buku sesuai dengan perhitungan skor menggunakan metode *cosine similarity*.

5.2. Saran

Hasil dari penelitian saat ini memberikan beberapa saran yang dapat dilakukan pada penelitian selanjutnya, antara lain:

- A. Peningkatan jumlah data yang digunakan pada sistem rekomendasi buku.
- B. Penggunaan *Framework PHP* seperti *Codeigniter* atau *Laravel* dapat di implementasikan dalam pengembangan sistem rekomendasi buku ini kedepannya.
- C. Menu “pinjam buku” langsung dari halaman siswa dapat di tambahkan dalam pengembangan sistem rekomendasi buku ini.

DAFTAR PUSTAKA

- Ahmad Faisol, M. Aziz Muslim, & Hadi Suyono. (2014). Komparasi *Fuzzy AHP* dengan *AHP* pada Sistem Pendukung Keputusan Investasi Properti. *Jurnal EECCIS*, 8(2).
- Anderias Eko Wijaya, & Deni Alfian. (2018). Sistem Rekomendasi Laptop Menggunakan *Collaborative Filtering* Dan *Content-Based Filtering*. *Jurnal Computech & Bisnis*, 12(1), 11–27.
- Bagus Dwi Saputra. (2020). Sistem Informasi E-Business Pada Studio Foto Alluvio Photography. Dalam *Jurnal Ilmiah NERO* (Vol. 5, Nomor 2).
- Bilin Shao, Xiaojun Li, & Genqing Bian. (2021). *A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph*. Dalam *Expert Systems with Applications* (Vol. 165). Elsevier Ltd. <https://doi.org/10.1016/j.eswa.2020.113764>
- Diaz, C., Diaz, P., Sulistiowati, & Lemantara, J. (2016). Rancang Bangun Aplikasi Penjualan Online Pada Cv. Mitra Techno Sains (Studi Kasus: Cv. Mitra Techno Sains). Dalam *Tahun 2016 JSIKA* (Vol. 5, Nomor 12). www.lunix.vencelstudio.com
- Dini Susanti, & Santi. (2019). Pemanfaatan Taman Bacaan Masyarakat (Tbm) Dalam Meningkatkan Minat Baca Remaja (Studi Kasus di TBM Gunung Ilmu). *JURNAL COMM-EDU*.
- Endang Supriyati, & Mohammad Iqbal. (2018). Pengukuran Similarity Tema Pada Juz 30 Al Qur'an Menggunakan Teks Klasifikasi. *Jurnal SIMETRIS*, 9(1). <https://translate.google.co.id/>
- Endra Robby Yuli, & Aprilita Dwi Synta. (2018). E-Report Berbasis Web Menggunakan Metode Model View Controller Untuk Mengetahui Peningkatan Perkembangan Prestasi Anak Didik.
- Hartono. (2016). *Manajemen Perpustakaan Sekolah: Menuju Perpustakaan Modern dan Profesional*. Ar-Ruzz Media.
- Janan Witanto. (2018). *Minat Baca Yang Sangat Rendah*. Manajemen Kurikulum dan Pembelajaran,.
- Jian Wei, Jianhua He, Kai Chen, Yi Zhou, & Zuoyin Tang. (2017). *Collaborative filtering and deep learning based recommendation system for cold start items*. *Expert Systems with Applications*, 69, 1–39. <https://doi.org/10.1016/j.eswa.2016.09.040>
- Lila Setiyani. (2021). *Implementasi Cybersecurity pada Operasional Organisasi*.

- Marudut, V., Siregar, M., & Siagian, N. F. (2021). Sistem Informasi *Front Office* Untuk Peningkatan Pelayanan Pelanggan Dalam Reservasi Kamar Hotel. *Jurnal TEKINKOM*, 4, 2621–3079. <https://doi.org/10.37600/tekinkom.v4i1.279>
- Mochamad Alfian Rosid, Arif Senja Fitriani, Ika Ratna Indra Astutik, Nasrudin Iqrok Mulloh, & Haris Ahmad Gozali. (2020). *Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi*. *IOP Conference Series: Materials Science and Engineering*, 874(1). <https://doi.org/10.1088/1757-899X/874/1/012017>
- Muhammad Deni Wahyudi, & Muhammad Rasid Ridho. (2019). Sistem Informasi Penjualan Mobil Bekas Berbasis Web Pada Cv Phutu Oil Club Di Kota Batam. *Computer and Science Industrial Engineering*.
- Muhammad Samsudin, Muhdar Abdurahman, & Muksin Hi Abdullah. (2019). Sistem Informasi Pengkreditan Nasabah Pada Koperasi Simpan Pinjam Sejahtera Baru Kota Ternate Berbasis Web. *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*, 2(1). <https://doi.org/10.47324/ilkominfo.v2i1.16>
- Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, & Marijn Koolen. (2019). *Trends in content-based recommendation: Preface to the special issue on Recommender systems based on rich item descriptions*. Dalam *User Modeling and User-Adapted Interaction* (Vol. 29, Nomor 2, hlm. 239–249). Springer Netherlands. <https://doi.org/10.1007/s11257-019-09231-w>
- Prasetya, Sintia, & Putri. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (*Unified Modelling Language*). Dalam *DOI: ...* (Vol. 1, Nomor 1).
- Rahim, A. (2020). Sistem Informasi Pengolahan Data Rumah pada Perumahan Bumi Sempaja City Samarinda. Dalam *JURTI* (Vol. 4, Nomor 1).
- Richard Firdaus Oeyliawan, & Dennis Gunawan. (2017). Aplikasi Rekomendasi Buku pada Katalog Perpustakaan Universitas Multimedia Nusantara Menggunakan Vector Space Model. *ULTIMATICS*, IX(2), 97.
- Rofeca Giri Rymmai, & Saleema JS. (2017). Book Recommendation using Cosine Similarity. *International Journal of Advanced Research in Computer Science*, 8(3). www.ijarcs.info
- Rut Samuel, Ripa Natan, Fitria, & Umami Syafiqoh. (2018). Penerapan Cosine Similarity dan K-Nearest Neighbor (K-NN) pada Klasifikasi dan Pencarian Buku. *Journal of Big Data Analytic and Artificial Intelligence*, 1.

- Shahzad Qaiser, & Ramsha Ali. (2018). *Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents*. *International Journal of Computer Applications*, 181(1), 25–29. <https://doi.org/10.5120/ijca2018917395>
- Sitepu. (2012). *Penulisan Buku Teks Pelajaran*. PT Remaja Rosdakarya.
- SRS Reddy, Sravani Nalluri, Subramanyam Kuniseti, S. Ashok, & B. Venkatesh. (2019). *Content-based movie recommendation system using genre correlation*. *Smart Innovation, Systems and Technologies*, 105, 391–397. https://doi.org/10.1007/978-981-13-1927-3_42
- Supono & Putratama. (2016). *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter: Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter*.
- Tri Wahyudi. (2017). *Rancang Bangun Sistem Informasi Pondok Pesantren (Studi Kasus Darul Abror Watumas)*.

LAMPIRAN

Lampiran 1. Surat keterangan melaksanakan penelitian



SURAT KETERANGAN

Nomor : 008/KET/III.4.AU.308/F/2023

Yang bertanda tangan di bawah ini Kepala SMA Muhammadiyah 8 Sukodadi Lamongan menerangkan dengan sesungguhnya :

Nama : RYKY ARDIANSYAH
Jenis Kelamin : Laki - Laki
Tempat, tanggal lahir : Lamongan, 01 Mei 2001
NIM : 19.03.01.0002
Program Studi : S-1 Teknik Komputer
Fakultas : Sains, Teknik, dan Pendidikan
Perguruan Tinggi : Universitas Muhammadiyah Lamongan

bahwa yang tersebut di atas dengan sebenarnya telah melaksanakan Kegiatan Penelitian di SMA Muhammadiyah 8 Sukodadi guna menyelesaikan penulisan tugas akhir kuliah dengan judul " *Sistem Rekomendasi Buku Perpustakaan Sekolah Menggunakan Metode Content - Based Filtering* ".

Demikian surat keterangan ini dibuat dengan sesungguhnya untuk dapat dipergunakan sebagaimana mestinya.

Sukodadi, 03 Mei 2023



Lampiran 2. Source code

Menentukan kriteria

```

$result = $database->query('SELECT buku.*,
kategori.kategori_nama
FROM buku
JOIN kategori ON buku.kategori_id =
kategori.kategori_id');

$data = [];
foreach ($result as $row) {
    $data[$row->buku_id] = pre_process($row->buku_judul .
' ' . $row->penulis . ' ' . $row->kategori_id . ' ' .
$row->buku_deskripsi);
}

```

- 1) `$result = $database->query('SELECT buku.*, kategori.kategori_nama FROM buku JOIN kategori ON buku.kategori_id = kategori.kategori_id');` merupakan kode yang berfungsi untuk melakukan *query* pada *database* untuk mengambil data buku dan kategori dari tabel buku dan kategori. Join dilakukan dengan menggunakan JOIN pada kolom `kategori_id` yang ada pada kedua tabel untuk memanggil `kategori_nama` pada tabel buku.
- 2) `$data = [];` : Membuat array kosong `$data` yang nantinya akan diisi dengan hasil *preprocessing* dari tiap dokumen.
- 3) `foreach ($result as $row)` : Melakukan perulangan pada setiap baris hasil *query* yang dihasilkan.
- 4) `$data[$row->buku_id] = pre_process($row->buku_judul . ' ' . $row->penulis . ' ' . $row->kategori_id . ' ' . $row->buku_deskripsi);` : Menyimpan hasil kriteria dari dokumen buku pada variabel `$data`. Data kriteria dihasilkan dengan menjalankan fungsi `pre_process()` pada teks yang terdiri dari judul buku, penulis, id kategori yang nantinya akan memanggil `kategori_nama`, dan deskripsi buku. Hasil kriteria dari tiap dokumen buku akan disimpan pada indeks `$row->buku_id` pada *array* `$data`.

Preprocessing

```
function pre_process($str)
{
    $stemmerFactory = new
\Sastrawi\Stemmer\StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();
    $stopWordRemoverFactory = new
\Sastrawi\StopWordRemover\StopWordRemoverFactory();
    $stopword = $stopWordRemoverFactory-
>createStopWordRemover();

    $str = strtolower($str);
    $str = $stemmer->stem($str);
    $str = $stopword->remove($str);

    return $str;
}
```

- 1) Mengubah semua huruf pada teks menjadi huruf kecil (*lowercase*) menggunakan fungsi `strtolower()`.
- 2) Melakukan *stemming* pada teks menggunakan *library* Sastrawi versi bahasa Indonesia. *Stemming* adalah proses menghilangkan imbuhan pada kata untuk mengembalikan kata dasar (*root word*) agar kata yang memiliki makna yang sama dapat dikenali sebagai satu kata yang sama. Fungsi `createStemmer()` digunakan untuk membuat objek *stemmer*, sedangkan `stem($str)` digunakan untuk melakukan *stemming* pada teks.
- 3) Melakukan penghapusan kata-kata yang tidak relevan (*stopword*) pada teks menggunakan *library* Sastrawi. *Stopword* adalah kata-kata yang sering muncul dalam sebuah teks namun tidak memiliki makna yang signifikan seperti kata depan, kata sambung, atau kata ganti. Fungsi `createStopWordRemover()` digunakan untuk membuat objek *remover*, sedangkan `remove($str)` digunakan untuk melakukan penghapusan *stopword* pada teks.

Menghitung nilai DF

```
private $corpus_terms = array();
function df($term)
{
    $d = array();
    $tr = $this->corpus_terms[$term];
    foreach ($tr as $t)
        $d[] = $t[0];

    $dx = array_unique($d);
    return count($dx);
}
```

- 1) `$d = array();` : Membuat *array* kosong `$d` yang nantinya akan diisi dengan nomor dokumen yang mengandung *term* yang dicari.
- 2) `$tr = $this->corpus_terms[$term];` : Mendapatkan *array* yang berisi informasi dokumen dan frekuensi kemunculan *term* pada dokumen tersebut dari korpus yang telah diolah sebelumnya, dengan parameter yang diambil adalah *term* yang sedang dicari.
- 3) `foreach ($tr as $t) $d[] = $t[0];` : Melakukan perulangan pada setiap elemen *array* `$tr` dan menambahkan nomor dokumen ke dalam *array* `$d`.
- 4) `$dx = array_unique($d);` : Menghapus duplikat nomor dokumen dari *array* `$d` dan menyimpan hasilnya pada variabel `$dx`.
- 5) `return count($dx);` : Mengembalikan jumlah elemen pada *array* `$dx`, yang merepresentasikan jumlah dokumen yang mengandung *term* yang dicari. Jumlah ini kemudian digunakan untuk menghitung *Inverse Document Frequency (IDF)* pada suatu *term* dalam rumus *TF-IDF*.

Menghitung nilai IDF

```
private $num_docs = 0;
private $corpus_terms = array();

function idf()
{
    $ndf = [];
    foreach ($this->corpus_terms as $t => $terms) {
        $df = $this->df($t);
        $ddf = $this->num_docs / $this->df($t);
        $idf = round(log10($ddf), 4);
        $ndf[$t][0] = $df;
        $ndf[$t][1] = $idf;
    }
    return $ndf;
}
```

- 1) `private $num_docs = 0;` : Inisialisasi jumlah total dokumen pada korpus dengan nilai 0.
- 2) `private $corpus_terms = array();` : Inisialisasi *array* `$corpus_terms` sebagai wadah untuk menyimpan *term* dan informasi terkait dalam korpus.
- 3) `function idf()` : Fungsi untuk menghitung *IDF*.
- 4) `$ndf = [];` : Inisialisasi *array* `$ndf` sebagai wadah untuk menyimpan informasi mengenai *DF* dan *IDF* dari tiap *term*.
- 5) `foreach ($this->corpus_terms as $t => $terms)` : Melakukan perulangan pada setiap *term* dalam *array* `$corpus_terms` dan menyimpan hasilnya pada variabel `$t`.
- 6) `$df = $this->df($t);` : Menghitung *Document Frequency (DF)* dari *term* yang sedang diproses dengan memanggil fungsi `$this->df()` yang telah didefinisikan sebelumnya. Hasilnya disimpan pada variabel `$df`.
- 7) `$ddf = $this->num_docs / $this->df($t);` : Menghitung nilai *IDF* dengan rumus $\log(\text{total dokumen} / DF)$. Hasilnya disimpan pada variabel `$ddf`.

- 8) `$idf = round(log10($ddf), 4);` : Mengubah nilai *IDF* menjadi logaritma basis 10 dan membulatkannya ke empat angka di belakang koma. Hasilnya disimpan pada variabel `$idf`.
- 9) `$ndf[$t][0] = $df;` : Menyimpan nilai *DF* dari *term* yang sedang diproses pada indeks ke-0 *array* `$ndf` pada indeks yang sesuai.
- 10) `$ndf[$t][1] = $idf;` : Menyimpan nilai *IDF* dari *term* yang sedang diproses pada indeks ke-1 *array* `$ndf` pada indeks yang sesuai.
- 11) `return $ndf;` : Mengembalikan *array* `$ndf` yang berisi informasi *DF* dan *IDF* dari seluruh *term* dalam korpus.

Pembobotan TF-IDF

```
private $doc_weight = array();

function weight(){
    $ndw = [];
    foreach($this->docs as $k=>$d){
        $dterm = explode(" ", $d);
        $dx = array_count_values($dterm);
        foreach($this->idf() as $t => $terms){
            if(empty($dx[$t]))
                $ndw[$k][$t] = 0;
            else $ndw[$k][$t] = $dx[$t] * $terms[1];
        }
    }
    $this->doc_weight = $ndw;

    return $ndw;
}
```

- 1) `$ndw = [];` : Membuat sebuah *array* kosong `$ndw` yang nantinya akan diisi dengan bobot dokumen.
- 2) `foreach($this->docs as $k=>$d){` : Melakukan perulangan pada setiap dokumen pada variabel `$this->docs`. `$k` adalah indeks dari dokumen, sedangkan `$d` adalah teks dokumen yang telah dipreproses.
- 3) `$dterm = explode(" ", $d);` : Memecah teks dokumen menjadi *array* kata-kata menggunakan fungsi `explode()`, dengan separator spasi.

- 4) `$dx = array_count_values($dterm);` : Menghitung frekuensi kemunculan setiap kata pada dokumen menggunakan fungsi `array_count_values()`.
- 5) `foreach($this->idf() as $t => $terms){` : Melakukan perulangan pada setiap term atau kata yang ada pada corpus menggunakan fungsi `idf()` (yang tidak ditampilkan di dalam kode tersebut). `$t` adalah term atau kata, sedangkan `$terms` adalah array yang berisi nilai IDF dari term tersebut.
- 6) `if(empty($dx[$t])) $ndw[$k][$t] = 0;` : Jika term tersebut tidak muncul pada dokumen, maka bobotnya diatur menjadi 0.
- 7) `else $ndw[$k][$t] = $dx[$t] * $terms[1];` : Jika term tersebut muncul pada dokumen, maka bobotnya dihitung menggunakan rumus *TF-IDF* yaitu $TF * IDF$, dimana *TF* adalah frekuensi kemunculan *term* pada dokumen dan *IDF* adalah invers dari frekuensi kemunculan *term* pada seluruh dokumen pada corpus. Nilai *IDF* diambil dari indeks ke-1 dari *array* `$terms` yang dihasilkan oleh fungsi `idf()`.
- 8) `$this->doc_weight = $ndw;` : Menyimpan hasil perhitungan bobot dokumen pada variabel `$this->doc_weight`.
- 9) `return $ndw;` : Mengembalikan hasil perhitungan bobot dokumen.

Pencarian relevansi dokumen

```
private $doc_weight = array();
function search($keyword)
{
    $key = explode(" ", $keyword);
    $score = [];
    $i = 0;
    foreach ($this->doc_weight as $ndw => $w) {
        $score[$ndw] = 0;
        foreach ($w as $wg => $v) {
            foreach ($key as $k) {
                if ($k == $wg)
                    $score[$ndw] += $v;
            }
        }
        $i++;
    }
    arsort($score);
    return $score;
}
```

- 1) `private $doc_weight = array();` : mendefinisikan sebuah properti *private* bernama `$doc_weight` yang berisi sebuah *array* kosong.
- 2) `function search($keyword)`: mendefinisikan sebuah fungsi bernama *search*, yang memiliki satu argumen yaitu `$keyword`.
- 3) `explode(" ", $keyword)` : mengambil string `$keyword` dan memisahkannya menjadi sebuah *array* berisi kata-kata individu dengan menggunakan spasi sebagai pemisah. *Array* hasilnya disimpan pada variabel `$key`.
- 4) `$score = []`; : menginisialisasi sebuah *array* kosong bernama `$score` yang akan digunakan untuk menyimpan hasil pencarian.
- 5) `$i = 0`; : menginisialisasi sebuah variabel *counter* bernama `$i`.
- 6) `foreach ($this->doc_weight as $ndw => $w)` : melakukan perulangan pada properti `$doc_weight` dari objek saat ini, kemudian menetapkan setiap pasangan *key-value* ke variabel `$ndw` dan `$w`.
- 7) `$score[$ndw] = 0`; : menginisialisasi skor untuk dokumen saat ini menjadi 0.

- 8) `foreach ($w as $wg => $v)` : melakukan perulangan pada kata-kata dan bobot untuk dokumen saat ini, kemudian menetapkan setiap pasangan *key-value* ke variabel `$wg` dan `$v`.
- 9) `foreach ($key as $k)` : melakukan perulangan pada kata-kata individu dalam *query* pencarian, kemudian menetapkan setiap kata ke variabel `$k`.
- 10) `if ($k == $wg)` : memeriksa apakah kata saat ini dalam *query* pencarian cocok dengan kata saat ini dalam dokumen. Jika cocok, bobot kata dalam dokumen ditambahkan ke skor untuk dokumen saat ini.
- 11) `$score[$ndw] += $v;` : menambahkan bobot kata saat ini dalam dokumen ke skor untuk dokumen saat ini.
- 12) `$i++;` : mengincrement variabel *counter* `$i`.
- 13) `arsort($score);` : mengurutkan hasil pencarian berdasarkan skor secara menurun.
- 14) `return $score;` : mengembalikan hasil pencarian yang sudah diurutkan dalam bentuk *array* asosiatif, dimana *key*-nya adalah nama dokumen dan *value*-nya adalah skornya.

Pembobotan *cosine similarity*

```

private $corpus_terms = array();
private $doc_weight = array();

public function similarity($d1)
{
    $score = [];
    foreach ($this->doc_weight as $ndw => $w) {
        $score[$ndw] = $this->cosim($d1, $ndw);
    }

    arsort($score);
    return $score;
}

private function cosim($d1, $d2)
{
    $dw = $this->doc_weight;

    # sum square dari 2 doc
    $dw1 = $dw[$d1];
    $dw2 = $dw[$d2];

    $dx = 0;
    $dx1 = 0;
    $dx2 = 0;

    foreach ($this->corpus_terms as $t => $terms) {
        $dx += $dw1[$t] * $dw2[$t];
        $dx1 += $dw1[$t] * $dw1[$t];
        $dx2 += $dw2[$t] * $dw2[$t];
    }

    return round($dx / (sqrt($dx1) * sqrt($dx2)), 4);
}

```

- 1) `private $corpus_terms = array();` : Membuat variabel `$corpus_terms` yang bertipe *array* dengan nilai awal *array* kosong, digunakan untuk menyimpan kumpulan term dalam korpus teks.
- 2) `private $doc_weight = array();` : Membuat variabel `$doc_weight` yang bertipe *array* dengan nilai awal *array* kosong, digunakan untuk menyimpan nilai bobot dari setiap dokumen dalam korpus.

- 3) `public function similarity($d1) :` Membuat fungsi *similarity* yang menerima satu parameter `$d1`, yang digunakan untuk mencari kesamaan antara dokumen `$d1` dengan semua dokumen lain dalam korpus.
- 4) `$score = [];` : Membuat variabel `$score` yang bertipe *array* dengan nilai awal *array* kosong, digunakan untuk menyimpan nilai skor kesamaan antara dokumen `$d1` dengan dokumen lain dalam korpus.
- 5) `foreach ($this->doc_weight as $ndw => $w):` Melakukan pengulangan untuk setiap dokumen dalam `$this->doc_weight` dengan menggunakan variabel `$ndw` sebagai kunci dokumen dan `$w` sebagai nilai bobot dokumen.
- 6) `$score[$ndw] = $this->cosim($d1, $ndw);` : Menghitung skor kesamaan antara dokumen `$d1` dan dokumen dengan kunci `$ndw` menggunakan fungsi `cosim()`, lalu menyimpan nilai skor tersebut ke dalam *array* `$score` dengan kunci `$ndw`.
- 7) `arsort($score);` : Mengurutkan *array* `$score` secara menurun (*descending*) berdasarkan nilai skor kesamaan dokumen.
- 8) `return $score;` : Mengembalikan nilai *array* `$score` yang berisi skor kesamaan dokumen.
- 9) `private function cosim($d1, $d2) :` Membuat fungsi `cosim()` yang menerima dua parameter `$d1` dan `$d2`, digunakan untuk menghitung nilai kesamaan kosinus antara dua dokumen.
- 10) `$dw = $this->doc_weight;` : Menyimpan nilai dari variabel `$this->doc_weight` ke dalam variabel `$dw`.
- 11) `$dw1 = $dw[$d1];` : Menyimpan nilai bobot dokumen `$d1` ke dalam variabel `$dw1`.
- 12) `$dw2 = $dw[$d2];` : Menyimpan nilai bobot dokumen `$d2` ke dalam variabel `$dw2`.

- 13) `$dx = 0;; $dx1 = 0;; dan $dx2 = 0;;` : Menginisialisasi variabel `$dx`, `$dx1`, dan `$dx2` dengan nilai 0.
- 14) `foreach ($this->corpus_terms as $t => $terms) :`
Melakukan pengulangan untuk setiap term dalam `$this->corpus_terms` dengan menggunakan variabel `$t` sebagai kunci *term* dan `$terms` sebagai nilai dari *array term*.
- 15) `$dx += $dw1[$t] * $dw2[$t];` : Menghitung nilai `$dx` dengan mengalikan bobot term dari dokumen `$d1` dengan dokumen `$d2`.
- 16) `$dx1 += $dw1[$t] * $dw1[$t];` : Menghitung nilai `$dx1` dengan mengalikan bobot *term* dari dokumen `$d1` dengan dirinya sendiri.
- 17) `$dx2 += $dw2[$t] * $dw2[$t];` : menghitung kuadrat panjang dokumen ke-2 dengan cara mengalikan setiap nilai dari `$dw2` dan menambahkan hasilnya ke dalam variabel `$dx2`.
- 18) `return round($dx / (sqrt($dx1) * sqrt($dx2)), 4);`
: menghitung nilai *cosine similarity* antara dokumen ke-1 dan dokumen ke-2 dengan membagi hasil dari `$dx` dengan hasil perkalian akar kuadrat dari `$dx1` dan `$dx2`. Nilai yang dikembalikan telah dibulatkan menjadi 4 angka di belakang koma menggunakan fungsi `round()`.

Membuat index untuk terms dokumen

```
private $num_docs = 0;
private $corpus_terms = array();
private $docs = array();

function create_index($d)
{
    $this->docs = $d;
    $this->num_docs = count($d);
    foreach ($d as $k => $dv) {
        $doc_terms = array();
        $doc_terms = explode(" ", $dv);

        $num_terms = count($doc_terms);
        for ($j = 0; $j < $num_terms; $j++) {
            term = strtolower($doc_terms[$j]);
            $this->corpus_terms[$term][] = array($k, $j);
        }
    }
}
```

- 1) `private $num_docs = 0;` : adalah variabel untuk menyimpan jumlah dokumen yang akan diindeks, awalnya diinisialisasi dengan nilai 0.
- 2) `private $corpus_terms = array();` : adalah *array* asosiatif untuk menyimpan daftar *term* dalam korpus dokumen beserta posisi masing-masing term pada setiap dokumen. *Array* ini nantinya akan menjadi indeks yang dibuat.
- 3) `private $docs = array();` : adalah *array* yang menyimpan daftar dokumen yang akan diindeks.
- 4) `function create_index($d)` : adalah fungsi untuk membuat indeks dari dokumen-dokumen yang diberikan sebagai parameter (`$d`). Fungsi ini akan memanggil *array* dokumen `$d` pada setiap iterasinya.
- 5) `$this->docs = $d;` : menyimpan daftar dokumen yang diberikan pada variabel `$docs`.
- 6) `$this->num_docs = count($d);` : menyimpan jumlah dokumen dalam variabel `$num_docs`.
- 7) `foreach ($d as $k => $dv)` : digunakan untuk mengulangi setiap dokumen dalam *array* `$d`. Variabel `$k` adalah index dokumen, sedangkan `$dv` adalah teks dari dokumen itu sendiri.

- 8) `$doc_terms = array();` : membuat *array* kosong untuk menyimpan term dari dokumen.
- 9) `$doc_terms = explode(" ", $dv);` : memecah teks dokumen menjadi *term-term* yang dipisahkan oleh spasi dan disimpan dalam *array* `$doc_terms`.
- 10) `$num_terms = count($doc_terms);` : menghitung jumlah *term* dalam dokumen tersebut.
- 11) `for ($j = 0; $j < $num_terms; $j++)` : Melakukan *looping* pada setiap *term* pada dokumen yang sedang diproses.
- 12) `term = strtolower($doc_terms[$j]);` : Mengonversi setiap *term* pada dokumen yang sedang diproses menjadi *lowercase*, lalu menyimpannya ke dalam variabel `$term`.
- 13) `$this->corpus_terms[$term][] = array($k, $j);` : Menambahkan elemen baru pada variabel `$corpus_terms` dengan indeks `$term` dan value sebuah *array* berisi indeks dokumen dan indeks *term* dalam dokumen yang sedang diproses. Elemen ini merepresentasikan keberadaan suatu *term* pada suatu dokumen.

Menampilkan hasil pembuatan index

```
private $corpus_terms = array();
function show_index()
{
    ksort($this->corpus_terms);
    foreach ($this->corpus_terms as $term =>
$doc_locations) {
        echo "<b>$term:</b>";
        echo "<br />";
        foreach ($doc_locations as $doc_location) {
            echo "{" . $doc_location[0] . ", " .
$doc_location[1] . "} ";
            echo "<br />";
        }
    }
}
```

- 1) `private $corpus_terms = array();` : Mendeklarasikan variabel `$corpus_terms` dengan nilai awal sebuah *array* kosong. Variabel ini digunakan untuk menyimpan setiap *term* dari semua dokumen yang diproses pada saat membuat *inverted index*.
- 2) `function show_index()` : Mendeklarasikan fungsi `show_index` yang digunakan untuk menampilkan hasil pembuatan *inverted index*.
- 3) `ksort($this->corpus_terms);` : Mengurutkan *array* `$this->corpus_terms` secara *ascending* berdasarkan indeks (yaitu *term*) menggunakan fungsi `ksort`.
- 4) `foreach ($this->corpus_terms as $term => $doc_locations)` : Melakukan *looping* pada setiap elemen dari *array* `$this->corpus_terms`, dan menetapkan nilai dari setiap elemen tersebut ke dalam variabel `$doc_locations`, sedangkan indeks elemen tersebut disimpan dalam variabel `$term`.
- 5) `echo "$term:";` : Menampilkan *term* saat ini dalam bentuk teks *bold* (tebal).
- 6) `echo "
";` : Menampilkan baris kosong sebagai pemisah antar *term*.
- 7) `foreach ($doc_locations as $doc_location)` : Melakukan *looping* pada setiap elemen dari *array* `$doc_locations`, dan menetapkan nilai dari setiap elemen tersebut ke dalam variabel `$doc_location`.
- 8) `echo "{" . $doc_location[0] . ", " . $doc_location[1] . "}"` : Menampilkan indeks dokumen dan indeks *term* dalam dokumen yang sedang diproses pada *term* saat ini dalam bentuk kurung kurawal.
- 9) `echo "
";` : Menampilkan baris kosong sebagai pemisah antar indeks dokumen dan *term*.

Menampilkan rekomendasi buku

```

$cbars = new Cbars();
$cbars->create_index($data);
$cbars->idf();
$w = $cbars->weight();
$r = $cbars->similarity($id);
$n = 8;

<?php $i = 0; ?>
<?php foreach ($r as $k => $row) : ?>
    <?php if ($i == $n) break; ?>
    <?php if ($row == 1) continue; ?>
    <?php $h = get_buku_detail($k, $database); ?>
    <li><a href="detail_buku.php?id=<?php echo $h-
>buku_id ?>"><?php echo $h->buku_judul ?></a> (<?php echo
$row ?>)</li>
        <?php $i++ ?>
<?php endforeach ?>

```

- 1) `$cbars = new Cbars();` Membuat objek baru dari kelas *Cbars*, yang digunakan untuk melakukan *indexing*, *term weighting*, dan perhitungan *similarity score*.
- 2) `$cbars->create_index($data);` : memanggil method `create_index` dari objek `$cbars` dengan parameter `$data`, yang berfungsi untuk membuat *index* dari data yang telah diinputkan.
- 3) `$cbars->idf();` : memanggil method `idf` dari objek `$cbars`, yang berfungsi untuk menghitung nilai *idf* (*inverse document frequency*) dari setiap *term* dalam korpus dokumen.
- 4) `$w = $cbars->weight();` : memanggil method `weight` dari objek `$cbars`, yang berfungsi untuk menghitung nilai *tf-idf* (*term frequency-inverse document frequency*) dari setiap *term* dalam setiap dokumen yang telah diinputkan, kemudian menyimpan nilai tersebut pada variabel `$w`.
- 5) `$r = $cbars->similarity($id);` : memanggil *method similarity* dari objek `$cbars` dengan parameter `$id`, yang berfungsi untuk menghitung nilai kemiripan antara dokumen yang memiliki *id* `$id` dengan dokumen lainnya yang telah diinputkan. Kemudian, nilai kemiripan tersebut disimpan pada variabel `$r`.
- 6) `$n = 8;` : menginisialisasi nilai variabel `$n` dengan angka 8.

- 7) `<?php $i = 0; ?>` : menginisialisasi variabel `$i` dengan nilai 0 dalam bentuk sintaksis *PHP*.
- 8) `<?php foreach ($r as $k => $row) : ?>` : memulai perulangan *foreach* dengan `$r` sebagai *array* yang *dilooping*, dimana setiap elemen *array* akan disimpan ke variabel `$row` dengan indeksnya disimpan di variabel `$k`.
- 9) `<?php if ($i == $n) break; ?>` : sebuah kondisi yang digunakan untuk menghentikan perulangan jika variabel `$i` sama dengan nilai variabel `$n`.
- 10) `<?php if ($row == 1) continue; ?>` : sebuah kondisi yang digunakan untuk melanjutkan ke perulangan berikutnya jika nilai dari variabel `$row` sama dengan 1.
- 11) `<?php $h = get_buku_detail($k, $database); ?>` : memanggil fungsi `get_buku_detail` dengan parameter `$k` dan `$database`, yang berfungsi untuk mengambil detail dari buku dengan id `$k` dari *database*. Hasil dari fungsi ini disimpan pada variabel `$h`.
- 12) `<a href="detail_buku.php?id=<?php echo $h->buku_id ?>"><?php echo $h->buku_judul ?> (<?php echo $row ?>)` : menampilkan data buku yang mirip dengan dokumen yang memiliki *id* `$id`. Informasi buku yang ditampilkan meliputi judul buku dan nilai kemiripannya dengan dokumen yang memiliki *id* `$id`.
- 13) `<?php $i++ ?>` : menambahkan nilai variabel `$i` sebesar 1 setiap kali satu perulangan selesai dieksekusi.
- 14) `<?php endforeach ?>` : menutup perulangan *foreach*.