**Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR**

SKRIPSI

Oleh:

SOYID WAHYU DARMAWAN NIM. 1741720151



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

**2021**

**Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

SOYID WAHYU DARMAWAN NIM. 1741720141



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**JUNI 2021**

# **HALAMAN PENGESAHAN**

**Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR**

**Disusun oleh:**

**SOYID WAHYU DARMAWAN NIM. 1741720151**

**Proposal Skripsi ini telah diuji pada …, .. …. 2021**

**Disetujui oleh:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Pembimbing Utama | : | Ridwan Rismanto, S.ST., M.Kom.  NIP. 19860318 201212 1 001 | ........................... |
|  |  |  |  |  |
| 2. | Pembimbing Pendamping | : | Habibie Ed Dien, S.Kom., MT.  NIP. 19920412 201903 1 013 | ........................... |
|  |  |  |  |  |
| 3. | Penguji Utama | : | ……………………………….  NIP. …………………………. | ........................... |
| 4. | Penguji Pendamping | : | ……………………………...  NIP. ……………………….. | ........................... |

Mengetahui,

Ketua Jurusan  
Teknologi Informasi

Rudy Ariyanto, S.T., M.CS  
NIP. 19711110 199903 1 002

Ketua Program Studi  
Teknik Informatika

Imam Fahrur Rozi, ST., MT.  
NIP. 198406102008121004

# **PERNYATAAN**

Dengan ini saya menyatakan bahwa Laporan Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya/Sarjana Terapan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, .. ….. 2021

Soyid Wahyu Darmawan

# **ABSTRAK**

**Wahyu D., Soyid.** “Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR”. **Pendamping: (1)** **Ridwan Rismanto, S.ST., M.Kom. (2) Habibie Ed Dien, S.Kom., MT.**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2021.**

Salah satu layanan di perpustakaan digital rumah sakit adalah sistem informasi untuk menemukan ketersediaan koleksi dari buku elektronikyang dimaksud. Tetapi tidak banyak sistem menyediakan mekanisme yang dapat merekomendasikan karyawan rumah sakit mengenai bukuelektronik yang mereka inginkan untuk mereka jadikan literatur, tetapi juga buku elektronik lainnya yang mungkin perlukan. beberapa judul buku elektronik dengan kandungan isi yang berbeda-beda. Perbedaan dapat dilihat dari beberapa kriteria seperti kelayakan isi, kebahasaan, penyajian, dan kegrafikaan. Sehingga dibuatlah sistem rekomendasi untuk merekomendasikan *E-book* sebagai bahan pertimbangan sesuai yang diinginkan karyawan rumah sakit itu sendiri. Penggunaan metode VIKOR diterapkan karena termasuk dari *Multi Criteria Decision Making* untuk mengambil keputusan bersifat diskrit berdasarkan beberapa kriteria sebagai hasil terbaik dari rekomendasi sistem. Pembuatan sistem menggunakan bahasa pemrograman Python framework Flask sebagai backend API, Javascript framework ReactJs sebagai frontend, dan database MySql.

**Kata Kunci :** Perpustakaan Digital, Sistem Informasi, Rumah Sakit, E-book, VIKOR, Flask, ReactJS, MySql

# ***ABSTRACT***

**Wahyu D., Soyid.** *“E-book recommendations at the Digital Warehouse of the General Hospital of the Christian University of Indonesia using the VIKOR Method”*. ***Counseling Lecturer*: (1)** **Ridwan Rismanto, S.ST., M.Kom. (2) Habibie Ed Dien, S.Kom., MT.**

***Thesis, Informatics Management Study Program, Department of Information Technology, State Polytechnic of Malang, 2021.***

*One of the services in the hospital's digital library is an information system to find the availability of a collection of the electronic books in question. However, not many systems provide a mechanism by which hospital employees can recommend the e-books they want them to turn into literature, but also any other e-books they may need. several titles of electronic books with different contents. The differences can be seen from several criteria such as content feasibility, language, presentation, and graphics. So that a recommendation system was created to recommend the E-book as a consideration according to what the hospital employees wanted. The use of the VIKOR method is applied because it is included in the Multi Criteria Decision Making to make discrete decisions based on several criteria as the best results from system recommendations. The system development uses the Python Flask framework programming language as the API backend, the Javascript framework ReactJs as the frontend, and the MySql database.*

***Keywords:*** *Digital Library, Information System, Hospital, E-book, VIKOR,* Flask, ReactJS, MySql

# **KATA PENGANTAR**

Puji Syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan akhir dengan judul “Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR”. Skripsi ini disusun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Allah SWT yang telah memberikan rahmat dan hidayat sehingga mampu menyelesaikan tugas akhir ini sehingga selesai tepat waktu.
2. Orang tua saya yang telah memberikan doa dan dukungan, baik secara moril maupun materil.
3. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
4. Bapak Imam Fahrur Rozi, S.T., M.T., selaku ketua program studi Teknik Informatika.
5. Bapak Ridwan Rismanto, S.ST., M.Kom. selaku dosen pembimbing utama.
6. Bapak Habibie Ed Dien, S.Kom., MT. selaku dosen pembimbing pendamping.
7. Teman – teman yang telah memberikan bantuan dan dukungan dalam membantu menyelesaikan laporan skripsi.
8. Seluruh dosen, staf, dan karyawan Politeknik Negeri Malang yang telah membantu penulis dalam menyelesaikan laporan skripsi ini.
9. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan

kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, … …. 2021

Soyid Wahyu Darmawan

# **DAFTAR ISI**

[HALAMAN PENGESAHAN i](#_Toc73477424)

[PERNYATAAN ii](#_Toc73477425)

[ABSTRAK iii](#_Toc73477426)

[*ABSTRACT* iv](#_Toc73477427)

[KATA PENGANTAR v](#_Toc73477428)

[DAFTAR ISI vii](#_Toc73477429)

[DAFTAR TABEL ix](#_Toc73477430)

[DAFTAR GAMBAR xi](#_Toc73477431)

[DAFTAR LAMPIRAN xiii](#_Toc73477432)

[BAB I. PENDAHULUAN 1](#_Toc73477433)

[1.1. Latar Belakang 1](#_Toc73477434)

[1.2. Rumusan Masalah 3](#_Toc73477435)

[1.3. Tujuan 3](#_Toc73477436)

[1.4. Batasan Masalah 4](#_Toc73477437)

[1.5. Sistematika Penulisan 5](#_Toc73477438)

[BAB II. LANDASAN TEORI 7](#_Toc73477439)

[2.1. Penelitian Terdahulu 7](#_Toc73477440)

[2.2. Sistem Pendukung Keputusan 9](#_Toc73477442)

[2.3. Metode VlseKriterijumska Optimizacija Kompromisno Resenje (VIKOR) 10](#_Toc73477443)

[2.4. Python 11](#_Toc73477444)

[2.5. JAVASCRIPT 12](#_Toc73477445)

[2.6. MySQL 12](#_Toc73477446)

[BAB III. METODOLOGI PENELITIAN 13](#_Toc73477447)

[3.1. Data 13](#_Toc73477449)

[3.2. Metode Pengembangan Perangkat Lunak 13](#_Toc73477450)

[3.3. Metode Pengumpulan Data 15](#_Toc73477452)

[3.4. Metode Pengolahan Data 15](#_Toc73477453)

[3.5. Metode Pengujian 16](#_Toc73477455)

[BAB IV. ANALISIS DAN PERANCANGAN 17](#_Toc73477456)

[4.1. Analisis 17](#_Toc73477457)

[4.2. Perancangan 37](#_Toc73477492)

[BAB V. IMPLEMENTASI DAN PENGUJIAN 52](#_Toc73477532)

[5.1. Implementasi Basis Data 52](#_Toc73477533)

[5.2. Implementasi Pengkodean Sistem 56](#_Toc73477545)

[5.3. Implementasi tampilan 57](#_Toc73477546)

[5.4. Pengujian Fungsionalitas sistem 62](#_Toc73477555)

[BAB VI. HASIL DAN PEMBAHASAN 63](#_Toc73477556)

[6.1. Hasil Pengujian Metode VIKOR 63](#_Toc73477557)

[6.2. Hasil Pengujian Akurasi 69](#_Toc73477568)

[BAB VII. KESIMPULAN DAN SARAN 71](#_Toc73477571)

[7.1. Kesimpulan 71](#_Toc73477572)

[7.2. Saran 71](#_Toc73477573)

[DAFTAR PUSTAKA 72](#_Toc73477574)

[Lampiran 75](#_Toc73477575)

# **DAFTAR TABEL**

[Tabel 2.1.1. Tabel Penelitian terdahulu 7](#_Toc73477441)

[Tabel 4.1.2.1. Tabel Analisis Pengguna 17](#_Toc73477460)

[Tabel 4.1.3.1. Tabel Kebutuhan Fungsional 18](#_Toc73477462)

[Tabel 4.1.4.1. Tabel Analisis Kebutuhan Perangkat Lunak 19](#_Toc73477464)

[Tabel 4.1.4.2. Tabel Analisis Kebutuhan Perangkat Keras 19](#_Toc73477465)

[Tabel 4.1.5.1. Tabel Deskripsi Aktor 21](#_Toc73477468)

[Tabel 4.1.6.2. Tabel Skenario – mengelola data nakes 22](#_Toc73477471)

[Tabel 4.1.6.3. Tabel Skenario – mengelola data jenis buku 22](#_Toc73477472)

[Tabel 4.1.6.4. Tabel Skenario – mengelola data penulis 23](#_Toc73477473)

[Tabel 4.1.6.5. Tabel Skenario – mengelola data penerbit 23](#_Toc73477474)

[Tabel 4.1.6.6. Tabel Skenario – mengelola data buku 24](#_Toc73477475)

[Tabel 4.1.6.7. Tabel Skenario – mengelola data review 24](#_Toc73477476)

[Tabel 4.1.6.8. Tabel Skenario – mengelola data admin 24](#_Toc73477477)

[Tabel 4.1.6.9. Tabel Skenario – mengelola data nilai buku 25](#_Toc73477478)

[Tabel 4.1.6.10. Tabel Skenario – melihat data buku 25](#_Toc73477479)

[Tabel 4.1.6.11. Tabel Skenario – mengakses isi buku 26](#_Toc73477480)

[Tabel 4.1.7.1. Tabel Spesifikasi use case 27](#_Toc73477482)

[Tabel 4.2.1.1. Tabel Admin 37](#_Toc73477495)

[Tabel 4.2.1.2. Tabel User 37](#_Toc73477496)

[Tabel 4.2.1.3. Tabel Ulasan 38](#_Toc73477497)

[Tabel 4.2.1.4. Tabel Jenis Buku 38](#_Toc73477498)

[Tabel 4.2.1.5. Tabel Tenaga Kesehatan 38](#_Toc73477499)

[Tabel 4.2.1.6. Tabel Penerbit 38](#_Toc73477500)

[Tabel 4.2.1.7. Tabel Penulis 38](#_Toc73477501)

[Tabel 4.2.1.8. Tabel Buku 38](#_Toc73477502)

[Tabel 4.2.1.9. Tabel Kriteria 39](#_Toc73477503)

[Tabel 4.2.1.10. Tabel Nilai Buku 39](#_Toc73477504)

[Tabel 6.1.1. Tabel Matriks Keputusan User Pertama 63](#_Toc73477558)

[Tabel 6.1.2. Tabel Matriks Keputusan User Kedua 63](#_Toc73477559)

[Tabel 6.1.3. Tabel Matriks Keputusan User Ketiga 64](#_Toc73477560)

[Tabel 6.1.4. Tabel Matriks Keputusan Rata-Rata 64](#_Toc73477561)

[Tabel 6.1.5. Tabel Penentuan Nilai Alternatif Masing-Masing Kriteria 65](#_Toc73477562)

[Tabel 6.1.6. Tabel Matriks Nornalisasi 65](#_Toc73477563)

[Tabel 6.1.7. Tabel Matriks Normalisasi Terbobot 66](#_Toc73477564)

[Tabel 6.1.8. Tabel Penentuan Nilai SI tiap Kriteria 66](#_Toc73477565)

[Tabel 6.1.9. Tabel Penentuan Nilai Max Ri tiap Kriteria 67](#_Toc73477566)

[Tabel 6.1.10. Tabel Perankingan Metode VIKOR 68](#_Toc73477567)

[Tabel 6.2.1. Tabel Pengujian Akurasi 69](#_Toc73477569)

[Tabel 6.2.2. Tabel Perhitungan Pengujian Akurasi 70](#_Toc73477570)

# **DAFTAR GAMBAR**

[Gambar 3.1. Gambar Bagan Metodologi Penelitian 13](#_Toc73477448)

[Gambar 3.2.1. Gambar Metode RAD 14](#_Toc73477451)

[Gambar 3.4.1. Gambar Data Buku 16](#_Toc73477454)

[Gambar 4.1.5.1. Gambar Diagram Use case 20](#_Toc73477467)

[Gambar 4.1.8.1. Gambar Activity Diagram Login Admin 29](#_Toc73477484)

[Gambar 4.1.8.2. Gambar Activity Diagram Mengelola Data Nakes 30](#_Toc73477485)

[Gambar 4.1.8.3. Gambar Activity Diagram Mengelola Data Buku 31](#_Toc73477486)

[Gambar 4.1.8.4. Gambar Activity Diagram Mengelola Nilai Buku 32](#_Toc73477487)

[Gambar 4.1.8.5. Gambar Activity Diagram Login User 33](#_Toc73477488)

[Gambar 4.1.8.6. Gambar Activity Diagram Melihat Data Buku 34](#_Toc73477489)

[Gambar 4.1.8.7. Gambar Activity Diagram Melihat Detail Buku 35](#_Toc73477490)

[Gambar 4.1.8.8. Gambar Activity Diagram Mengisi Nilai Buku 36](#_Toc73477491)

[Gambar 4.2.1.1. Gambar Perancangan Basis Data 37](#_Toc73477494)

[Gambar 4.2.2.1.1. Gambar tampilan login 40](#_Toc73477507)

[Gambar 4.2.2.1.2. Gambar tampilan dashboard 40](#_Toc73477508)

[Gambar 4.2.2.1.3. Gambar tampilan data user 41](#_Toc73477509)

[Gambar 4.2.2.1.4. Gambar tampilan data admin 41](#_Toc73477510)

[Gambar 4.2.2.1.5. Gambar tampilan data ulasan 42](#_Toc73477511)

[Gambar 4.2.2.1.6. Gambar tampilan data nakes 42](#_Toc73477512)

[Gambar 4.2.2.1.7. Gambar tampilan data jenis buku 43](#_Toc73477513)

[Gambar 4.2.2.1.8. Gambar tampilan data penerbit 43](#_Toc73477514)

[Gambar 4.2.2.1.9. Gambar tampilan data penulis 44](#_Toc73477515)

[Gambar 4.2.2.1.10. Gambar tampilan data buku 44](#_Toc73477516)

[Gambar 4.2.2.1.11. Gambar tampilan data kriteria 45](#_Toc73477517)

[Gambar 4.2.2.1.12. Gambar tampilan data nilai buku 45](#_Toc73477518)

[Gambar 4.2.2.1.13. Gambar tampilan data Rekomendasi 46](#_Toc73477519)

[Gambar 4.2.2.2.1. Gambar tampilan register 46](#_Toc73477521)

[Gambar 4.2.2.2.2. Gambar tampilan login 47](#_Toc73477522)

[Gambar 4.2.2.2.3. Gambar tampilan lupa password 47](#_Toc73477523)

[Gambar 4.2.2.2.4. Gambar tampilan beranda 48](#_Toc73477524)

[Gambar 4.2.2.2.5. Gambar tampilan data buku 48](#_Toc73477525)

[Gambar 4.2.2.2.6. Gambar tampilan data buku kedokteran 49](#_Toc73477526)

[Gambar 4.2.2.2.7. Gambar tampilan data buku keperawatan 49](#_Toc73477527)

[Gambar 4.2.2.2.8. Gambar tampilan data buku farmasi 50](#_Toc73477528)

[Gambar 4.2.2.2.9. Gambar tampilan data buku rekomendasi 50](#_Toc73477529)

[Gambar 4.2.2.2.10. Gambar tampilan detail buku 51](#_Toc73477530)

[Gambar 4.2.2.2.11. Gambar tampilan ulasan buku 51](#_Toc73477531)

[Gambar 5.1.1. Gambar Desain Basis Data 52](#_Toc73477534)

[Gambar 5.1.2. Gambar tabel admin 52](#_Toc73477535)

[Gambar 5.1.3. Gambar tabel buku 53](#_Toc73477536)

[Gambar 5.1.4. Gambar tabel jenis buku 53](#_Toc73477537)

[Gambar 5.1.5. Gambar tabel kriteria 53](#_Toc73477538)

[Gambar 5.1.6. Gambar tabel tenaga kesehatan 54](#_Toc73477539)

[Gambar 5.1.7. Gambar tabel nilai buku 54](#_Toc73477540)

[Gambar 5.1.8. Gambar tabel penerbit 54](#_Toc73477541)

[Gambar 5.1.9. Gambar tabel penulis 55](#_Toc73477542)

[Gambar 5.1.10. Gambar tabel ulasan 55](#_Toc73477543)

[Gambar 5.1.11. Gambar tabel user 55](#_Toc73477544)

[Gambar 5.3.1.1. Gambar Tampilan Dashboard 58](#_Toc73477548)

[Gambar 5.3.1.2. Gambar Tampilan Kriteria Buku 58](#_Toc73477549)

[Gambar 5.3.1.3. Gambar Tampilan Rekomendasi Buku 59](#_Toc73477550)

[Gambar 5.3.2.1. Gambar Tampilan Buku 60](#_Toc73477552)

[Gambar 5.3.2.2. Gambar Rekomendasi Buku 60](#_Toc73477553)

[Gambar 5.3.2.3. Gambar Tampilan Detail Buku 61](#_Toc73477554)

# **DAFTAR LAMPIRAN**

Lampiran 1 Data Buku

Lampiran 2 Menu Utama Aplikasi

Lampiran 3 Keluaran Sistem

Lampiran 4 Kuesioner Uji Pengguna

Lampiran 5 *Listing Program*

# **BAB I. PENDAHULUAN**

## **Latar Belakang**

Tantangan baru teknologi informasi khususnya penyedia informasi adalah bagaimana menyalurkan informasi secara cepat, tepat dan global. Dengan pesatnya perkembangan teknologi informasi memberikan peluang untuk mengelola informasi dalam format digital (Radjatadoe et al., 2017). Digital library merupakan kumpulan kumpulan koleksi digital berupa konten dan jasa (fungsi) terkait dengan penyimpanan, penemuan, pencarian dan pelestarian ilmu pengetahuan (Wicaksana et al., 2017).

Digital Repository seringkali dikaitkan dengan upaya menghimpun karya-karya intelektual dan materi ilmiah dalam format digital yang dapat diakses secara online (Qurotianti, 2019). Peran lain dengan Adanya digital repository ini memungkinkan pengguna untuk mengakses seluruh informasi secara online dalam format digital sehingga masyarakat semakin dimudahkan dengan kemudahan akses dan kemapanan tempat (Artikel, 2017).

Studi Kasus dari penelitian ini adalah di Rumah Sakit Umum Universitas Kristen Indonesia. Berada di bagian P4 (Pendidikan, Perencanaan, Pengembangan, dan Penelitian) ini sebagai divisi yang mendalami urusan internal dari Rumah Sakit UKI. Demi mewujudkan Rumah Sakit yang mempunyai keunggulan dalam pelayanan dan pendidikan di bidang kesehatan (*Informasi Lengkap Dan Buat Janji Di RSU Universitas Kristen Indonesia (UKI). Biaya, Tindakan Medis, Daftar Dokter, Selengkapnya.*, n.d.).

Berdasarkan penjelasan tersebut, penulis memutuskan untuk memilih topik dari penelitian adalah dengan menggunakan Sistem Pendukung Keputusan (SPK). dengan menerapkan Sistem Pendukung Keputusan (SPK), suatu sistem informasi berbasis komputer yang mengkombinasikan model dan data untuk menyediakan dukungan kepada pengambil keputusan dalam memecahkan masalah terstruktur disebut Sistem Pendukung Keputusan (Pohan & Sinaga, 2020).

Alasan adanya Rekomendasi E-book ini supaya user mengetahui literatur untuk dibaca dengan mudah berdasarkan kriteria penilaian E-book tersebut. Gambaran sistem ini adalah saat user menginputkan nilai kriteria dan nilai tersebut tersimpan pada database sehingga untuk memunculkan hasil penilaian tersebut, ditampilkan sebuah combo box dengan nama “rekomendasi” sehingga user mengetahui Rekomendasi E-book berdasarkan penilaian tersebut. Kriteria penilaian sebuah E-book adalah melihat dari segi kriteria Kelayakan isi, Kebahasaan, Penyajian, dan Kegrafikaan (Arsanti, 2018; Meningkatkan et al., 2020).

Metode yang tepat untuk penelitian ini adalah dengan menggunakan metode VIKOR (VlseKriterijumska Optimizacija Kompromisno Resenje). metode perankingan dengan menggunakan indeks peringkat multikriteria berdasarkan ukuran tertentu dari kedekatan dengan solusi yang ideal (Arisandi & Pribadi, 2020). Lalu metode VIKOR adalah metode untuk optimasi kriteria majemuk dalam suatu sistem yang kompleks dengan Konsep yang menentukan peringkat dari sampel yang ada dengan melihat hasil dari nilai regrets measure (R) dari setiap sampel (Sharma, 2017).

Alasan saya memilih Metode VIKOR karena Metode VIKOR sangat berguna pada situasi dimana pengambil keputusan tidak memiliki kemampuan untuk menentukan pilihan pada saat desain sebuah sistem dimulai (Tumanggor et al., 2018). Sehingga metode VIKOR sangat berpengaruh pada hasil penentuan rekomendasi E-book yang diteliti.

Berdasarkan hal tersebut, penulis memutuskan untuk membuat suatu Sistem Pendukung Keputusan untuk merekomendasikan E-book pada Digital Repository Rumah Sakit Umum UKI dengan menggunakan metode VIKOR. Dengan adanya sistem ini, diharapkan dapat membantu dan mempermudah pihak Rumah Sakit Umum UKI dalam merekomendasi E-book pada Digital Repository dengan baik.

## **Rumusan Masalah**

Berdasarkan Latar Belakang di atas, Rumusan Masalah yang dapat diambil adalah:

1. Bagaimana membuat sistem rekomendasi yang menghasilkan E-book dengan nilai tertinggi berdasarkan kriteria dari E-book tersebut?
2. Bagaimana membangun sistem rekomendasi E-book pada Digital Repository berdasarkan kriteria dengan menggunakan VIKOR?
3. Bagaimana tingkat akurasi VIKOR untuk diimplementasikan ke dalam sistem rekomendasi E-book berdasarkan kriteria dari E-book tersebut?

## **Tujuan**

Tujuan dari penilitian dengan judul Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR adalah sebagai berikut:

1. Untuk membuat sistem yang menghasilkan rekomendasi E-book yang sesuai berdasarkan kriteria penilaian buku. Hasil penentuan tersebut digunakan menjadi salah satu tolak ukur dalam menentukan Rekomendasi E-book.
2. Untuk memberikan rekomendasi E-book yang sesuai dengan kriteria penilaian buku dengan menggunakan Sistem Rekomendasi yang diimplementasikan dengan metode VIKOR
3. Dapat mengetahui tingkat akurasi dalam implementasi metode VIKOR kedalam sistem rekomendasi E-book berdasarkan kriteria penilaian buku (Kelayakan isi, Kebahasaan, Penyajian, Dan Kegrafikaan)

## **Batasan Masalah**

Agar skripsi yang berjudul Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR dapat berjalan sesuai rencana dan tujuan awal, maka penulis menetapkan batasan-batasan masalah yaitu:

1. Aplikasi yang dibuat berbasis website dengan menggunakan Bahasa pemrograman JAVASCRIPT dan framework ReactJS
2. Perhitungan metode hanya dapat dilakukan oleh satu user sehingga tidak adanya suatu GDSS (Group Decision Support System) atau Sistem Pendukung Keputusan secara berkelompok
3. Penentuan Rekomendasi E-book pada Digital Repository berdasarkan kriteria penilaian menggunakan metode VIKOR.
4. Keluaran dari sistem berupa rekomendasi E-book yang ditampilkan di laman Website Digital Repository.
5. Jumlah minimal buku yang di digitalisasi berjumlah 50 buku
6. Jumlah ukuran E-book yang dapat di upload ke sistem Digital Repository berukuran maksimal 350 MB *(Mega Byte)*

## **Sistematika Penulisan**

Sistematika Penulisan skripsi dengan judul “Rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia dengan menggunakan Metode VIKOR” dengan susunan sebagai berikut :

**BAB I : PENDAHULUAN**

Pendahuluan berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan. Bagian pendahuluan akan menjadi dasar penelitian yang dilakukan.

**BAB II : LANDASAN TEORI**

Berisikan teori-teori yang relevan yang melengkapi latar belakang. Sekaligus memberi *review* tentang pustaka yang telah dibaca selama masa pencarian solusi terhadap masalah yang diangkat dalam tugas akhir. Pustaka yang dijadikan rujukan atau referensi merupakan publikasi ilmiah. Publikasi ilmiah dapat berbentuk jurnal, prosiding, modul ajar, diktat dan lain-lain.

**BAB III : METODOLOGI PENELITIAN**

Terdiri dari langkah-langkah yang akan membimbing penulis memilih metode, teknik, prosedur apa yang tepat, dan *tools* apa yang akan digunakan sehingga setiap tahapan dapat dilakukan dengan tepat, termasuk desain dan perancangan sistem yang akan dibuat.

**BAB IV : ANALISIS DAN PERANCANGAN**

Pada bagian ini diuraikan dengan jelas sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan nonfungsional. Rancangan sistem meliputi rancangan model sistem, rancangan arsitektur sistem, rancangan proses, rancangan prosedural, rancangan data dan rancangan antarmuka pengguna

**BAB V : IMPLEMENTASI DAN PENGUJIAN**

Pada bagian ini, implementasi sistem dipaparkan secara detil sesuai rancangan dan komponen (*tools*) bahasa pemrograman yang dipakai. Implementasi rancangan proses dapat disertai dengan potongan kode pada proses yang dimaksud. Pengujian merupakan proses untuk menentukan apakah hasil dari tugas akhir sudah sesuai dengan kebutuhan sistem dan berjalan sesuai lingkungan yang diinginkan. Pengujian dapat berupa pengujian fungsional, penerimaan pengguna, ataupun performa sistem. Pada pengujian dipaparkan secara detil mengenai metode pengujian, tujuan pengujian, proses pengujian serta analisis hasil pengujian.

**BAB VI : HASIL DAN PEMBAHASAN**

Pembahasan merupakan argumentasi rasional dari penulis yang disusun secara sistematis berdasarkan fakta ilmiah yang diperoleh dari hasil pengujian. Pembahasan hasil pengujian dapat disajikan dalam bentuk uraian teoritik, baik secara kualitatif maupun kuantitatif. Secara ilmiah, hasil pengujian yang diperoleh dalam penelitian dapat berupa temuan baru atau perbaikan, penegasan atau penolakan interpretasi suatu fenomena ilmiah yang diteliti pada penelitian sebelumnya. Penyajian hasil penelitian dapat diperjelas dengan menggunakan tabel, kurva, grafik, foto atau bentuk lain dapat digunakan sesuai keperluan secara lengkap dan jelas.

**BAB VII : KESIMPULAN DAN SARAN**

Berisi uraian singkat dan jelas tentang hasil tugas akhir yang diperoleh sesuai dengan tujuan penelitian. Apabila diperlukan, saran dapat digunakan untuk menyampaikan hal-hal yang dapat diperbaiki, dikembangkan atau dijadikan penelitian lebih lanjut.

# **BAB II. LANDASAN TEORI**

Tinjauan pustaka merupakan bagian yang akan membahas tentang penyelesaian masalah yang akan memberikan jalan keluar atau hasil. Dalam hal ini akan dikemukakan beberapa teori yang akan berkaitan dengan permasalahan yang akan diangkat.

## **Penelitian Terdahulu**

Penelitian terdahulu ini menjadi salah satu acuan penulis untuk melakukan penelitian sehingga penulis dapat memperluas teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama dengan judul penelitian penulis terutama di bagian metode penelitian. Namun, penulis mengangkat beberapa penelitian untuk dijadikan referensi dalam menambah kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

### Tabel 2.1.1. Tabel Penelitian terdahulu

|  |  |  |
| --- | --- | --- |
| **Nama Peneliti** | **Judul Penelitian** | **Hasil Penelitian** |
| Farmy Setiawan Radjatadoe, Dony M. Sihotang, Meiton Boru  2017 | Pembuatan aplikasi digital library di jurusan ilmu  Komputer universitas nusa cendana berbasis web | Dalam menjalankan aplikasi berjalan dengan baik karena hasil pengujian sesuai dengan hasil yang diharapkan dan dari aspek fungsionalitas persentase tertinggi pada penilaian tanggapan sangat baik |
| Fikri Firgiawan, Wina Witanti dan Gunawan Abdillah  2018 | Sistem pendukung keputusan rekomendasi pemilihan buku komputer di amazon.com menggunakan metode technique for order preference by similarity to ideal solution (topis) | Dalam penelitian menjelaskan kalau pembuatan Aplikasi Sistem Rekomendasi dengan menggunakan Sistem Pendukung Keputusan dapat memberikan hasil yang baik |
| Boyma Simamora  2017 | Rancang bangun sistem rekomendasi televisi led dengan metode vikor berbasis web | Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa rancang bangun sistem rekomendasi pembelian televisi LED dengan metode VIKOR dengan kriteria sudah dapat diverifikasi dengan membandingkan hasil perhitungan sistem dengan perhitungan manual |
| Betrik Juliana Hutapea, Mentari Ananda Hasmi , Abdul Karim, Suginam  2018 | Sistem pendukung keputusan penentuan jenis kulit terbaik untuk pembuatan sepatu dengan menggunakan metode vikor | hasil penelitian adalah dengan membahas sistem pendukung yang dapat membantu untuk menentukan jenis kulit terbaik dengan menggunkan metode VIKOR. |

Dari beberapa judul penelitian yang telah dipaparkan, terpilih penelitian yang dilakukan oleh Boyma Simamora yang berjudul “Rancang bangun sistem rekomendasi televisi led dengan metode vikor berbasis web” sebagai referensi utama penulis dalam melakukan penelitian. Hal ini dikarenakan adanya pembahasan yang serupa dari segi ide dan konsep dari penelitian yang dilakukan. Dan Jurnal yang berjudul “Sistem Pendukung Keputusan Rekomendasi Pemilihan Buku Komputer Di Amazon.Com Menggunakan Metode Technique For Order Preference By Similarity To Ideal Solution (Topsis)” menjelaskan bahwa dalam mencari rekomendasi dapat digunakan dengan menggunakan Sistem Pendukung Keputusan (Firgiawan et al., 2018).

## **Sistem Pendukung Keputusan**

Sistem pendukung keputusan adalah bagian dari sistem informasi berbasis komputer (termasuk sistem pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. SPK merupakan penggabungan sumber- sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan (Hutapea  Mentari Ananda; Karim, Abdul; Suginam, Suginam, 2018).

Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Hamria Hamria et al., 2020). Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur (Safii & Zulhamsyah, 2018).

## **Metode VlseKriterijumska Optimizacija Kompromisno Resenje (VIKOR)**

VIKOR berasal dari kata VlseKriterijumska Optimizacija Kompromisno Resenje yang merupakan metode analisis pengambilan keputusan dengan multi atribut decision making yang dikembangkan oleh Serafim Opricovic untuk memecahkan permasalahan keputusan dengan kriteria yang saling bertentangan dan dari unit yang berbeda, dengan asumsi bahwa kompromi dapat diterima sebagai resolusi dari konflik yang ada (Bako & Lubis, 2018).

Metode vikor memiliki kelebihan dalam mengkompromi alternatif yang ada, serta dapat menyelesaikan pengambilan keputusan bersifat diskret pada kriteria yang bertentangan dan non commensurable, yaitu perbedaan unit antar kriteria (Sutrikanti et al., 2018).

Langkah-langkah yang digunakan dalam Metode VIKOR adalah sebagai berikut (Simamora, 2017):

1. Dengan menentukan tabel pengamatan dari database dengan mencari nilai data terbaik (*fi\**) dan terburuk (*fi-*) atau dengan istilah *Cost* dan *Benefit*
2. Menentukan bobot kriteria yang diperoleh dari pengguna sesuai dengan kebutuhan atau kriteria yang diinginkan
3. Menghitung Normalisasi matriks

(Rumus 2.3.1)

Keterangan :

Rij = nilai normalisasi sampe *i* pada kriteria *j*

fij = nilai data sampel *i* pada kriteria *j*

fi\* = nilai terbaik dalam satu kriteria

fi- = nilai terjelek dalam satu kriteria

1. Menghitung Normalisasi bobot (Wj x Rij)

Melakukan perkalian antara nilai data yang telah dinormalisasi dengan nilai bobot kriteria yang telah ditentukan

1. Menghitung nilai *Utility Measure (S)* dan *Regret Measure (R)*

(Rumus 2.3.2)

1. Menghitung indeks VIKOR

(Rumus 2.3.3)

Keterangan :

S\* = nilai S terkecil

S- = nilai S terbesar

R\* = nilai R terkecil

R- = nilai R terbesar

1. Perankingan alternatif

Setelah Qj dihitung, maka pengurutan perankingan ditentukan dari nilai yang paling rendah dengan solusi kompromi sebagai solusi ideal dari perankingan Qj dengan nilai terendah. Karena nilai Sj merupakan solusi yang diukur dari titik terjauh solusi ideal, sedangkan nilai Rj merupakan solusi yang diukur dari titik terdekat solusi ideal.

## **Python**

Python adalah bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi dan Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. (*Python (Bahasa Pemrograman) - Wikipedia Bahasa Indonesia, Ensiklopedia Bebas*, n.d.). Sintaks Python yang sederhana dan mudah dipelajari menekankan pada keterbacaan dan karena itu mengurangi biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode (*What Is Python? Executive Summary | Python.Org*, n.d.).

Flask merupakan microframework Python yang digunakan untuk membangun sebuah web service. Pada web service, Flask bertindak sebagai penengah antara website dengan basis data, flask juga bertindak untuk mengambil data dari basis data yang kemudian dijadikan umpan balik. (Parlika et al., 2020)

## **JAVASCRIPT**

JavaScript yaitu salah satu bahasa pemrograman yang paling populer digunakan dalam kurun waktu dua puluh tahun ini. Bahkan JavaScript juga dikenal juga sebagai salah satu dari tiga bahasa pemrograman yang utama bagi web developer (*Apa Itu JavaScript ? Pemahaman Dasar Bagi Pemula*, n.d.). Sebagai bahasa multi-paradigma, JavaScript mendukung gaya pemrograman yang digerakkan oleh peristiwa, fungsional, dan imperative (*JavaScript - Wikipedia*, n.d.).

Secara fungsional, JavaScript digunakan untuk menyediakan script pada objek yang dibenamkan (Ubaya et al., 1907). ReactJs merupakan kerangka kerja open source yang menggunakan library javascript untuk membuat user interface dan React biasa digunakan untuk menangani pengembangan pada aplikasi single-page dan aplikasi mobile. ReactJS memiliki keunggulan dimana kerangka kerja ini memberikan kecepatan, simplicity, dan sclability (Nursaid et al., 2020).

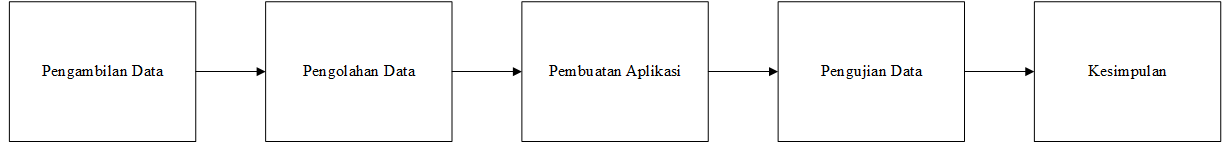
## **MySQL**

MySQL adalah DBMS yang open source dengan dua bentuk lisensi, yaitu Free Software (perangkat lunak bebas) dan Shareware (perangkat lunak berpemilik yang penggunaannya terbatas) (*Pengertian MySQL, Fungsi, Dan Cara Kerjanya (Lengkap)*, n.d.).

MySQL lebih sering digunakan dengan program lain untuk mengimplementasikan aplikasi yang membutuhkan kemampuan database relasional (*MySQL - Wikipedia*, n.d.). MySQL merupakan perangkat lunak manajemen basis data yang berlisensi open source. MySQL berkinerja yang sangat cepat, reliable serta mudah digunakan untuk melakukan organisir basis data (Aprida & Febriliyan, 2013).

# **BAB III. METODOLOGI PENELITIAN**

Proses pengambilan data dilakukan dengan cara melakukan mengambil data yang sudah dimiliki oleh pihak P4 (Pendidikan, Perencanaan, Pengembangan, dan Penelitian) Rumah Sakit Umum Universitas Kristen Indonesia untuk mendapatkan data yang dibutuhkan rekomendasi *E-book*, di dalam aplikasi digital repository, meliputi pengolahan data untuk menghitung perankingan rekomendasi *E-book*. Kesimpulan didapatkan sesuai dengan hasil dan kendala pada proses penelitian.



### Gambar 3.1. Gambar Bagan Metodologi Penelitian

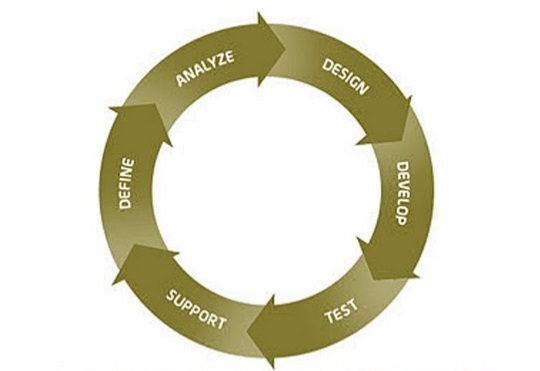
## **Data**

Data yang diolah merupakan data *E-book* yang didapatkan dari pihak P4 (Pendidikan, Perencanaan, Pengembangan, dan Penelitian) Rumah Sakit Umum Universitas Kristen Indonesia selama 5 bulan dimulai pada bulan Januari 2021 sampai dengan Mei 2021.

## **Metode Pengembangan Perangkat Lunak**

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini menerapkan SDLC (*System Development Life Cycle*) dengan model RAD (*Rapid Application Development*). Model RAD adalah metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan feedback berulang-ulang. Metode ini merupakan semacam cikal bakal *agile project management,* karena bisa mengikuti pace bisnis yang terus berkembang dan juga kebutuhan pasar yang terus meningkat. Pengembangan software pada umumnya seperti waterfall model membutuhkan perencanaan yang terbilang cukup kaku. Klien atau pelanggan seakan ‘dipaksa’ untuk menyetujui banyak hal di awal, tetapi mereka tidak bisa melihat proses pembuatannya.

Keuntungan utama menjalankan rapid application development adalah jangka waktu pengembangan lebih cepat. Hal ini dikarenakan feedback dari pelanggan cepat didapatkan dan semua perubahan yang dilakukan akan sesuai hasil tersebut. Akan tetapi, salah satu kekurangan RAD adalah kamu membutuhkan tim berisikan developer yang benar-benar memiliki skill tinggi dan juga metode ini hanya bisa digunakan untuk proyek yang bisa termodulasi.



### Gambar 3.2.1. Gambar Metode RAD

Metode RAD memiliki tahapan-tahapan sebagai berikut :

1. Menentukan project requirement

berisikan apa saja yang harus dicapai dalam sebuah proyek dan strategi untuk menghadapi permasalahan yang mungkin akan muncul. Persyaratan ini juga mencakup timeline dan budget yang ada. Orang-orang yang membuat project requirements adalah para developer, klien, dan juga software user.

1. Membuat prototype

mengembangkan prototipe secara cepat, dengan fitur dan fungsi yang dibutuhkan. Setelah itu, prototipe tersebut akan diberikan kepada klien untuk mengetahui apa saja yang mereka suka dan apa yang tidak. Pada tahap ini, hasil yang diberikan sama sekali belum sempurna, hanya menunjukkan fitur dan fungsi yang akan ada saja.

1. *Rapid construction* dan pengumpulan *feedback*

mencakup fitur, fungsi, visual, dan juga interface dari program yang sedang dikembangkan. Setelah itu, prototipe akan dikembangkan lagi sampai klien memberikan persetujuan untuk finalisasi produk.

1. Implementasi

Fitur, fungsi, visual, dan *interface* akan diulas kembali oleh klien. Pada tahap ini, uji coba akan dilakukan jika memang dibutuhkan. Uji cobanya mencakup kestabilan, *usability testing*, dan pengujian lainnya untuk memastikan semua hal sudah terkontrol.

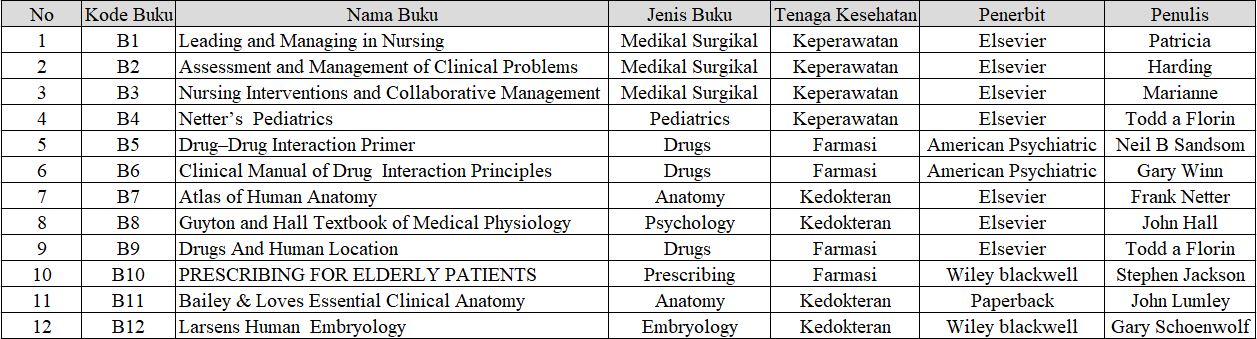
## **Metode Pengumpulan Data**

Metode Pengumpulan data dalam penyusunan penulisan skripsi ini adalah :

1. Studi Literatur dengan tujuan memahami algoritma VIKOR dengan mengumpulkan beberapa jurnal dan E-book melalui website.
2. Pengumpulan data dimulai dengan pengumpulan buku elektronik (*e-book*) dalam bentuk digital (.pdf) yang sudah dimiliki pihak P4 (Pendidikan, Perencanaan, Pengembangan, dan Penelitian) sebagai data dalam proses dalam rekomendasi E-book pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia.

## **Metode Pengolahan Data**

Tahap ini merupakan tahap pengumpulan data yang digunakan untuk proses pengolahan data yang diperoleh dari *e-book* yang dimiliki pihak P4 (Pendidikan, Perencanaan, Pengembangan, dan Penelitian) yang akan diolah pada sistem. Data buku ini digunakan sebagai tolak ukur dalam pembuatan aplikasi rekomendasi *E-book* pada Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia



### Gambar 3.4.1. Gambar Data Buku

## **Metode Pengujian**

Pengujian sistem merupakan pengujian tingkat kualitas sistem dari cara kerja dan penggunaan sistem yang dibuat. Pengujian sistem ini meliputi pengujian fungsionalitas, pengujian akurasi, dan usability testing.

1. Pengujian fungsionalitas dilakukan dengan menggunakan metode pengujian black box. Pengujian ini menggunakan data uji berupa data input dari user pada sistem. Hasil yang diperoleh dari pengujian black box adalah berjalannya fungsi pada sistem sesuai dengan yang diharapkan.

2. Pengujian akurasi digunakan untuk mengetahui tingkat keakuratan perhitungan algoritma VIKOR dengan melakukan pencocokan hasil data Excel.

3. Usability testing digunakan untuk melakukan pengujian pada tingkat manfaat aplikasi sistem bagi user dalam menentukan Rekomendasi E-book pada Digital Repository.

# **BAB IV. ANALISIS DAN PERANCANGAN**

## **4.1. Analisis**

### 4.1.1. Deskripsi Sistem

Sistem akan ada dua pengguna yaitu admin dan user atau karyawan Rumah Sakit Umum Universitas Kristen Indonesia. Dimana admin mengelola data admin, user, nakes, jenis buku, penerbit, penulis, buku, kriteria, nilai buku, dan perhitungan metode VIKOR. User atau karyawan dapat melihat E-book, melihat E-book berdasarkan rekomendasi, melihat detail E-book, membaca E-book, dan menilai E-book berdasarkan kriteria-kriteria.

### 4.1.2. Analisis Pengguna

Berikut hasil analisis pengguna yang telah diidentifikasi :

### Tabel 4.1.2.1. Tabel Analisis Pengguna

|  |  |  |
| --- | --- | --- |
| No | Pengguna | Deskripsi |
| 1 | Admin | Mengelola pengolahan data meliputi data admin, data user, data nakes, data jenis buku, data penerbit, data penulis, data buku, data kriteria, data nilai buku, dan data perhitungan metode VIKOR |
| 2 | Karyawan Rumah Sakit UKI | Mengolah dengan melihat data buku, data buku berdasarkan rekomendasi, data buku berdasarkan tenaga kesehatan, data buku berdasarkan tahun terbit, data buku berdasarkan urutan buku, melihat detail buku, dan menilai buku |

### 4.1.3. Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang membahas mengenai proses-proses yang berkaitan dengan sistem. Dalam sistem yang dibuat terdapat dua pengguna yaitu :

1. Admin

Pada sistem ini, admin melakukan pengolahan data meliputi data admin, data user, data nakes, data jenis buku, data penerbit, data penulis, data buku, data kriteria, data nilai buku, dan data perhitungan metode VIKOR

1. Karyawan

Pada sistem ini, karyawan melakukan Mengolah dengan melihat data buku, data buku berdasarkan rekomendasi, data buku berdasarkan tenaga kesehatan, data buku berdasarkan tahun terbit, data buku berdasarkan urutan buku, melihat detail buku, dan menilai buku

### Tabel 4.1.3.1. Tabel Kebutuhan Fungsional

|  |  |
| --- | --- |
| Data yang digunakan | * Data nilai buku * Data kriteria * Data buku |
| Admin | Menginputkan data yang digunakan meliputi data admin, data user, data nakes, data jenis buku, data penerbit, data penulis, data buku, data kriteria, data nilai buku, dan data perhitungan metode VIKOR |
| Karyawan | Menginputkan data nilai buku berdasarkan kriteria penilaian buku |

### 4.1.4. Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang fokus pada pelaku yang terdapat pada sistem. Berikut merupakan kebutuhan non-fungsional :

1. Keamanan

Keamanan pada sistem ini telah dilengkapi dengan adanya penggunaan password ketika admin dan pihak pelaksana melakukan proses login dengan penggunaan JWT (Json Web Token)

1. Spesifikasi untuk Developer

Spesifikasi yang dibutuhkan untuk developer dalam membangun sistem aplikasi ini akan dijelaskan pada tabel dibawah ini :

1. Analisis Kebutuhan Perangkat Lunak

Berikut adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada tabel 4.1.4.1.

### Tabel 4.1.4.1. Tabel Analisis Kebutuhan Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat** | **Perangkat lunak** |
| 1 | *Web browser* | Brave Browser |
| 2 | *Web server* | Apache |
| 3 | *DBMS* | MySQL |
| 4 | *Text Edior dan IDE* | Jetbrains Pycharm dan Jetbrains IntellijIDEA |
| 5 | *Framework* | Flask dan ReactJS |

1. Analisis Kebutuhan Perangkat Keras

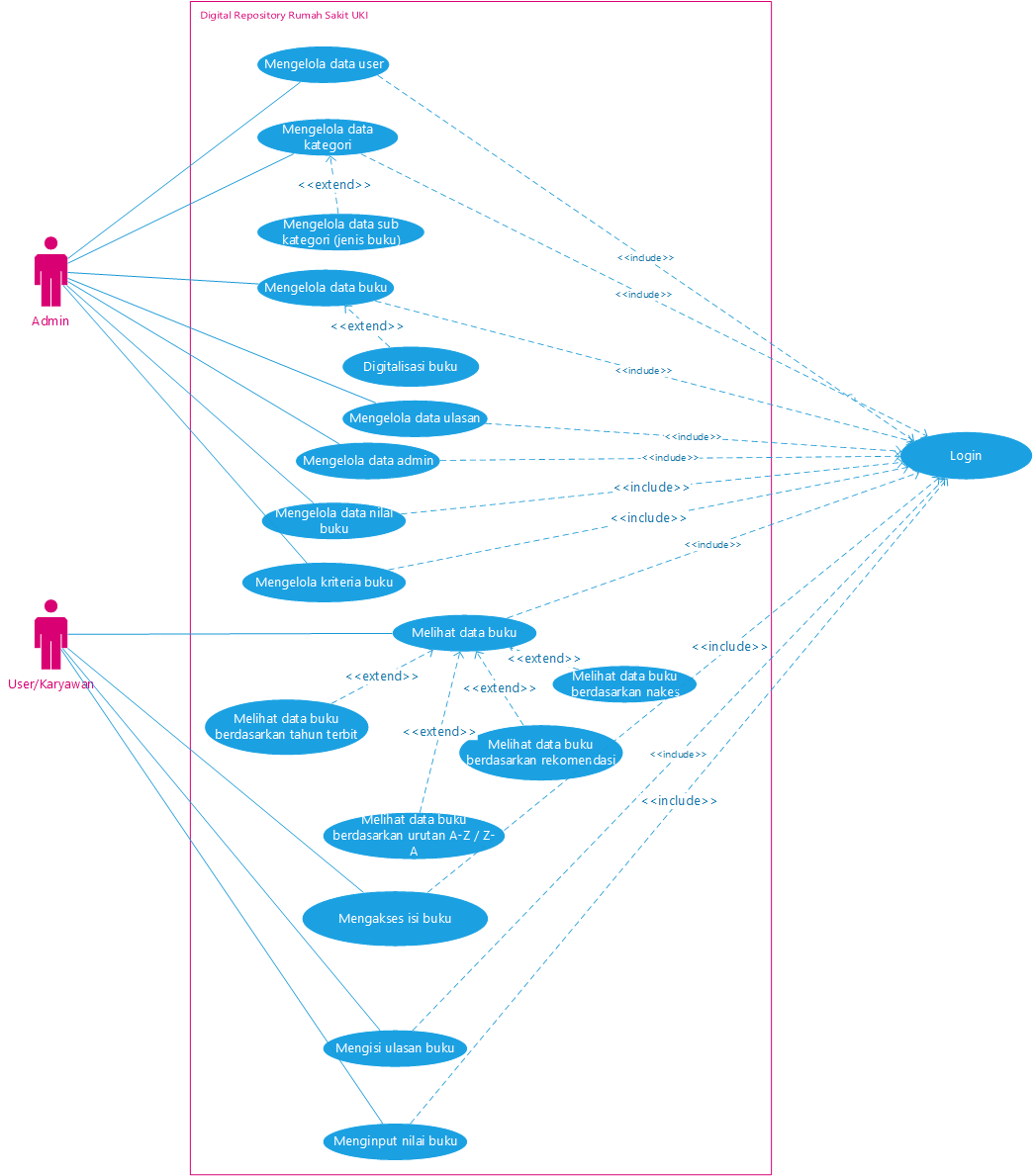
Berikut ini adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada Tabel 4.1.4.2.

### Tabel 4.1.4.2. Tabel Analisis Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| No | Perangkat | Perangkat Keras |
| 1 | *Processor* | Intel Core I7 – 9750H |
| 2 | *Memory* | 24 GB |
| 3 | *SSD* | 512 GB |
| 4 | *Harddisk* | 1 TB |
| 5 | Perangkat Input | Mouse, Harddisk External, Flashdisk, dan mousepad |

### 4.1.5. Diagram *Use case*

Diagram *usecase* merupakan interaksi yang menggambarkan hubungan antara aktor dengan kegiatan yang terdapat di dalam sistem. Berikut tampilan diagram usecase pada gambar 4.1.5.1 :



### Gambar 4.1.5.1. Gambar Diagram Use case

Berikut penjelasan aktor dari use case diagram pada tabel 4.1.5.1.

### Tabel 4.1.5.1. Tabel Deskripsi Aktor

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| Admin | Admin adalah orang yang bertanggung jawab atas segala urusan yang berkaitan tentang mengelola data-data sistem yaitu admin, user, nakes, jenis buku, penulis, penerbit, buku, nilai buku, kriteria, dan perhitungan metode VIKOR |
| Karyawan Rumah Sakit UKI | Karyawan merupakan orang yang memiliki keterkaitan tentang pekerjaan yang berada di tempat tersebut (Rumah Sakit UKI). Karyawan melihat data buku dan melihat data buku berdasarkan rekomendasi. Selain itu, karyawan juga dapat menilai hasil rekomendasi data buku. |

### 4.1.6. Skenario *Use case*

Berikut merupakan skenario yang akan dilakukan untuk setiap *use case*:

### Tabel 4.1.6.1. Tabel Skenario – mengelola data user

|  |  |
| --- | --- |
| Mengelola data user | |
| Tujuan | Menampilkan, mengubah, dan menghapus data user |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data user 2. Sistem menampilkan data user |
| Skenario alternatif | 1. Jika admin ingin mengubah data user karyawan dengan menekan tombol “edit” 2. Jika admin ingin menghapus data user karyawan dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data user sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.2. Tabel Skenario – mengelola data nakes

|  |  |
| --- | --- |
| Mengelola data nakes | |
| Tujuan | Menampilkan, menambah, mengubah, dan menghapus data nakes |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data nakes 2. Admin mengisi data nakes 3. Sistem menampilkan data nakes |
| Skenario alternatif | 1. Jika admin ingin menambah data nakes dengan mengisi form nakes dan menekan tombol “add” 2. Jika admin ingin mengubah data nakes dengan menekan tombol “edit” 3. Jika admin ingin menghapus data nakes dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data nakes sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.3. Tabel Skenario – mengelola data jenis buku

|  |  |
| --- | --- |
| Mengelola data jenis buku | |
| Tujuan | Menampilkan, menambah, mengubah, dan menghapus data jenis buku |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data jenis buku 2. Admin mengisi data jenis buku 3. Sistem menampilkan data jenis buku |
| Skenario alternatif | 1. Jika admin ingin menambah data jenis buku dengan mengisi form jenis buku dan menekan tombol “add” 2. Jika admin ingin mengubah data jenis buku dengan menekan tombol “edit” 3. Jika admin ingin menghapus data jenis buku dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data jenis buku sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.4. Tabel Skenario – mengelola data penulis

|  |  |
| --- | --- |
| Mengelola data penulis | |
| Tujuan | Menampilkan, menambah, mengubah, dan menghapus data penulis |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data penulis 2. Admin mengisi data penulis 3. Sistem menampilkan data penulis |
| Skenario alternatif | 1. Jika admin ingin menambah data penulis dengan mengisi form penulis dan menekan tombol “add” 2. Jika admin ingin mengubah data penulis dengan menekan tombol “edit” 3. Jika admin ingin menghapus data penulis dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data penulis sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.5. Tabel Skenario – mengelola data penerbit

|  |  |
| --- | --- |
| Mengelola data penerbit | |
| Tujuan | Menampilkan, menambah, mengubah, dan menghapus data penerbit |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data penerbit 2. Admin mengisi data penerbit 3. Sistem menampilkan data penerbit |
| Skenario alternatif | 1. Jika admin ingin menambah data penerbit dengan mengisi form penerbit dan menekan tombol “add” 2. Jika admin ingin mengubah data penerbit dengan menekan tombol “edit” 3. Jika admin ingin menghapus data penerbit dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data penerbit sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.6. Tabel Skenario – mengelola data buku

|  |  |
| --- | --- |
| Mengelola data buku | |
| Tujuan | Menampilkan, menambah, mengubah, dan menghapus data buku |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data buku 2. Admin mengisi data buku 3. Sistem menampilkan data buku |
| Skenario alternatif | 1. Jika admin ingin menambah data buku dengan mengisi form buku dan menekan tombol “add” 2. Jika admin ingin mengubah data buku dengan menekan tombol “edit” 3. Jika admin ingin menghapus data buku dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data buku sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.7. Tabel Skenario – mengelola data review

|  |  |
| --- | --- |
| Mengelola data nakes | |
| Tujuan | Menampilkan dan menghapus data user |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data review 2. Sistem menampilkan data review |
| Skenario alternatif | 1. Jika admin ingin menghapus data review dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data review sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.8. Tabel Skenario – mengelola data admin

|  |  |
| --- | --- |
| Mengelola data admin | |
| Tujuan | Menampilkan data admin |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data nakes 2. Sistem menampilkan data nakes |
| Kondisi akhir | Sistem menampilkan data admin yang tersimpan dalam *database* |

### Tabel 4.1.6.9. Tabel Skenario – mengelola data nilai buku

|  |  |
| --- | --- |
| Mengelola data nilai buku | |
| Tujuan | Menampilkan, menambah, mengubah, dan menghapus data user |
| Aktor | Admin |
| Kondisi awal | 1. Login sebagai admin |
| Skenario utama | 1. Admin memilih data nilai buku 2. Admin mengisi data nilai buku 3. Sistem menampilkan data nilai buku |
| Skenario alternatif | 1. Jika admin ingin menambah data nilai buku dengan mengisi form nilai buku dan menekan tombol “add” 2. Jika admin ingin mengubah data nilai buku dengan menekan tombol “edit” 3. Jika admin ingin menghapus data nilai buku dengan menekan tombol “delete” |
| Kondisi akhir | Sistem menampilkan data nilai buku sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.10. Tabel Skenario – melihat data buku

|  |  |
| --- | --- |
| Melihat data buku | |
| Tujuan | Menampilkan data buku |
| Aktor | User |
| Kondisi awal | 1. Login sebagai user |
| Skenario utama | 1. User memilih data buku 2. Sistem menampilkan data nakes |
| Skenario alternatif | 1. Jika user ingin melihat menu buku dengan menekan tombol “data buku” 2. Jika user ingin melihat menu rekomendasi buku dengan menekan tombol “data rekomendasi buku” 3. Jika user ingin melihat menu buku berdasarkan nakes dengan menekan tombol “data buku tenaga kesehatan” 4. Jika user ingin melihat menu buku berdasarkan tahun terbit dengan menekan tombol “data buku berdasarkan tahun terbit” 5. Jika user ingin melihat menu buku berdasarkan urutan buku dengan menekan tombol “data urutan buku” |
| Kondisi akhir | Sistem menampilkan data buku sesuai yang tersimpan dalam *database* |

### Tabel 4.1.6.11. Tabel Skenario – mengakses isi buku

|  |  |
| --- | --- |
| Mengakses isi buku | |
| Tujuan | Menampilkan, mengakses, mengunduh, dan menilai buku |
| Aktor | User |
| Kondisi awal | 1. Login sebagai user |
| Skenario utama | 1. User memilih detail buku setelah menekan buku yang ingin dibaca 2. User mengisi nilai buku berdasarkan kriteria 3. Sistem menampilkan data detail buku |
| Skenario alternatif | 1. Jika user ingin menilai buku dengan mengisi form penilaian buku berdasarkan kriteria berbentuk combo box 2. Jika admin ingin mengakses buku dengan menampilkan detail buku 3. Jika admin ingin mengunduh buku dengan menekan tombol “download” |
| Kondisi akhir | Sistem menampilkan data buku berdasarkan id sesuai yang tersimpan dalam *database* |

Tabel 4.1.6.12. Tabel Skenario – mengisi ulasan

|  |  |
| --- | --- |
| Mengisi ulasan | |
| Tujuan | Menambah data ulasan |
| Aktor | User |
| Kondisi awal | 1. Login sebagai user |
| Skenario utama | 1. User mengisi data ulasan |
| Skenario alternatif | 1. Jika admin ingin menambah data ulasan dengan mengisi form ulasan dan menekan tombol “add” |
| Kondisi akhir | Sistem menampilkan data ulasan sesuai yang tersimpan dalam *database* |

### 4.1.7. Spesifikasi use case

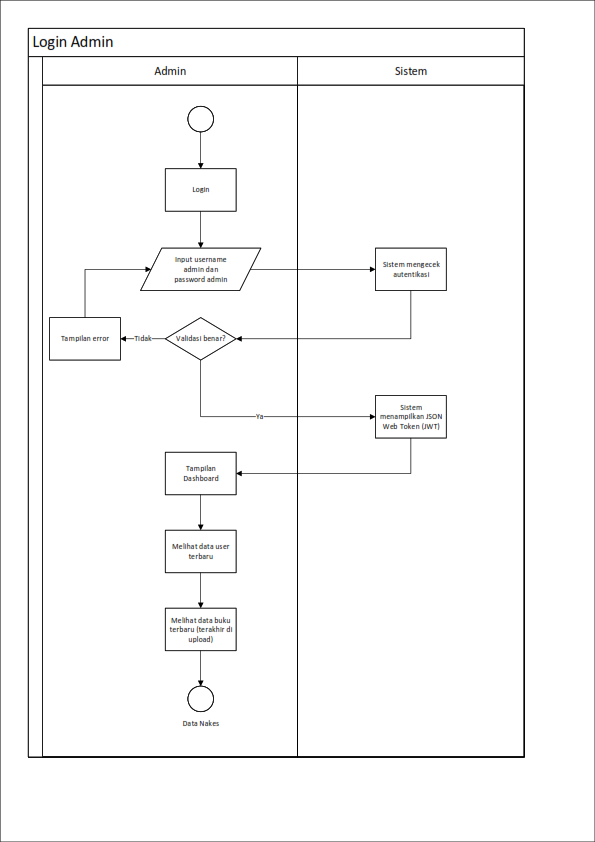
Deskripsi dari *use case* diagram yang menjelaskan bagaimana aktor dan *use case* terlibat dalam sistem.

### Tabel 4.1.7.1. Tabel Spesifikasi use case

|  |  |  |  |
| --- | --- | --- | --- |
| No | Aktor | *Use Case* | Keterangan |
| 1 | Admin | Mengelola data user | Admin dapat melihat dan mengubah data user |
| 2 | Admin | Mengelola data nakes | Admin dapat melihat, menambah, mengubah, dan menghapus data nakes |
| 3 | Admin | Mengelola data jenis buku | Admin dapat melihat, menambah, mengubah, dan menghapus data jenis buku |
| 4 | Admin | Mengelola data penulis | Admin dapat melihat, menambah, mengubah, dan menghapus data penulis |
| 5 | Admin | Mengelola data penerbit | Admin dapat melihat, menambah, mengubah, menghapus data penerbit |
| 6 | Admin | Mengelola data buku | Admin dapat melihat, menambah, mengubah, dan menghapus data buku |
| 7 | Admin | Mengelola data nilai buku | Admin dapat melihat, menambah, mengubah, dan menghapus data nilai buku |
| 8 | Admin | Mengelola data review | Admin dapat melihat dan menghapus data review |
| 9 | Admin | Mengelola data admin | Admin dapat melihat dan menghapus data admin |
| 10 | User | Melihat data buku | User dapat melihat data buku |
| 11 | User | Mengakses isi buku | User dapat mengakses isi buku berdasarkan id buku tersebut |
| 12 | User | Mengisi nilai buku | User dapat mengisi nilai buku berdasarkan kriteria |
| 13 | User | Mengisi ulasan buku | User dapat mengisi ulasan buku |

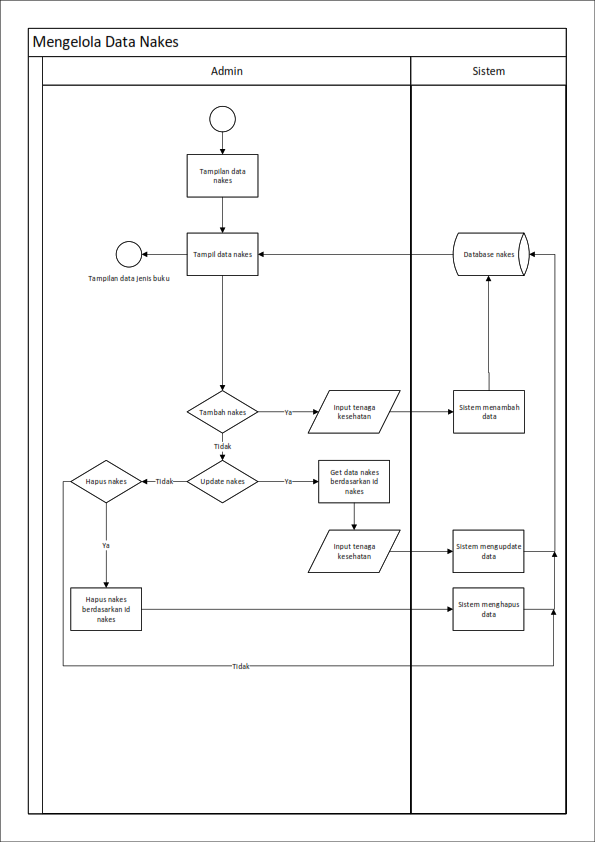
### 4.1.8. Activity Diagram

Activity Diagram digunakan untuk menjelaskan aktivitas apa saja yang dapat dilakukan dalam aplikasi. Kebutuhan yang telah diidentifikasi sebelumnya, dirancang beberapa activity diagram untuk menjelaskan alur proses dari kegiatan antara aktor dengan sistem, berikut proses yang akan berjalan yaitu:



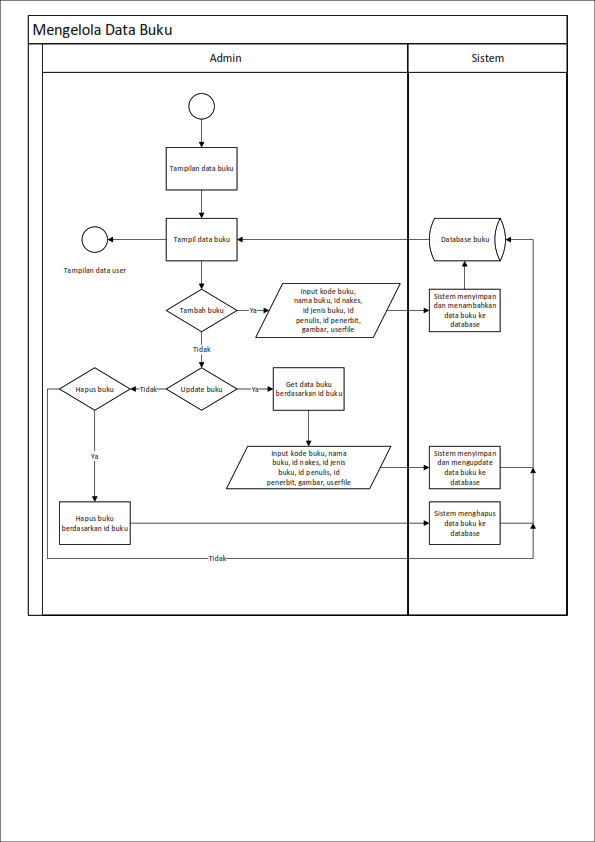
### Gambar 4.1.8.1. Gambar Activity Diagram Login Admin

Gambar 4.1.8.1 merupakan activity diagram login yaitu admin memasukan username dan password untuk melakukan login sistem. Sistem akan melakukan validasi. Jika sukses, akan di arahkan ke dalam aplikasi



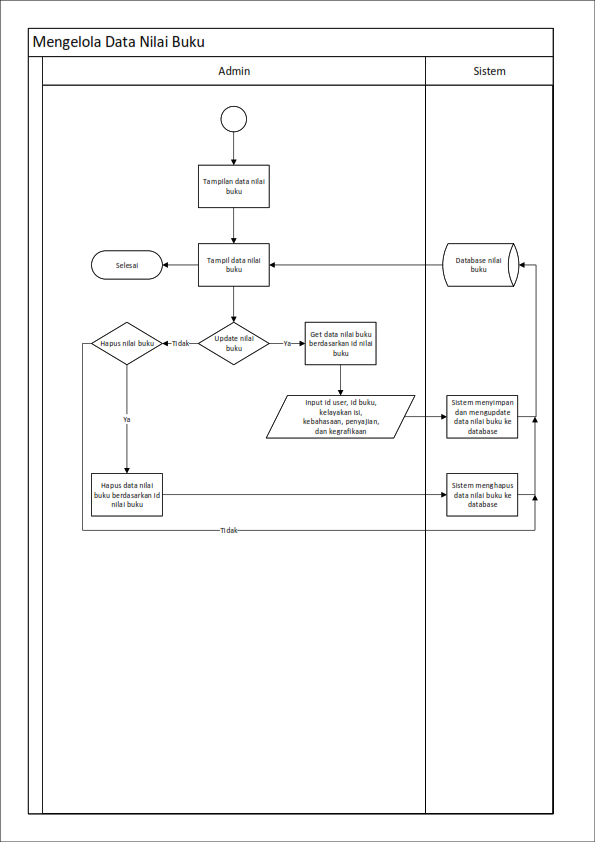
### Gambar 4.1.8.2. Gambar Activity Diagram Mengelola Data Nakes

Gambar 4.1.8.2. merupakan Activity Diagram untuk mengelola data nakes yang dimana admin dapat menambahkan, mengubah, dan menghapus data nakes yang tersimpan pada database



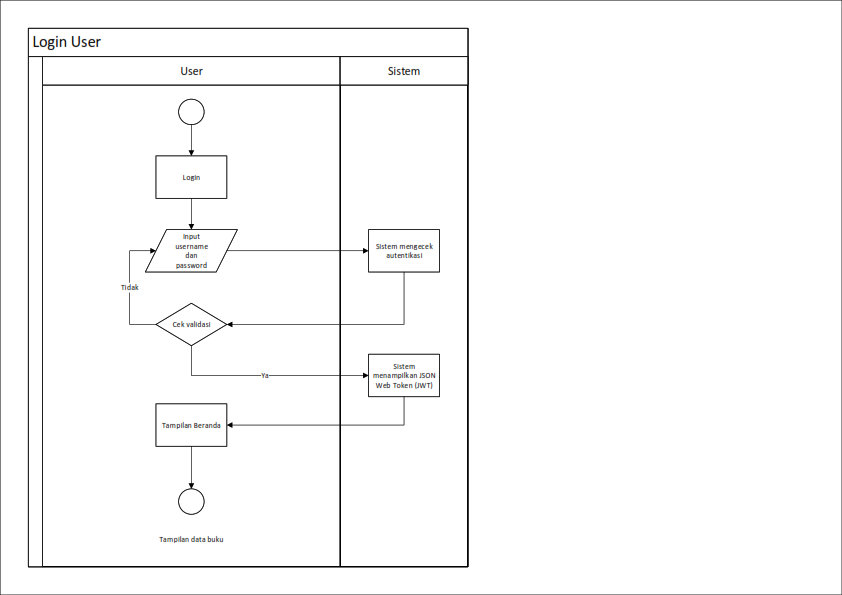
### Gambar 4.1.8.3. Gambar Activity Diagram Mengelola Data Buku

Gambar 4.1.8.3 merupakan activity diagram dalam mengelola data buku yang dimana admin dapat menambah, mengunggah gambar, mengunggah file, mengubah, dan menghapus data buku yang tersimpan pada database



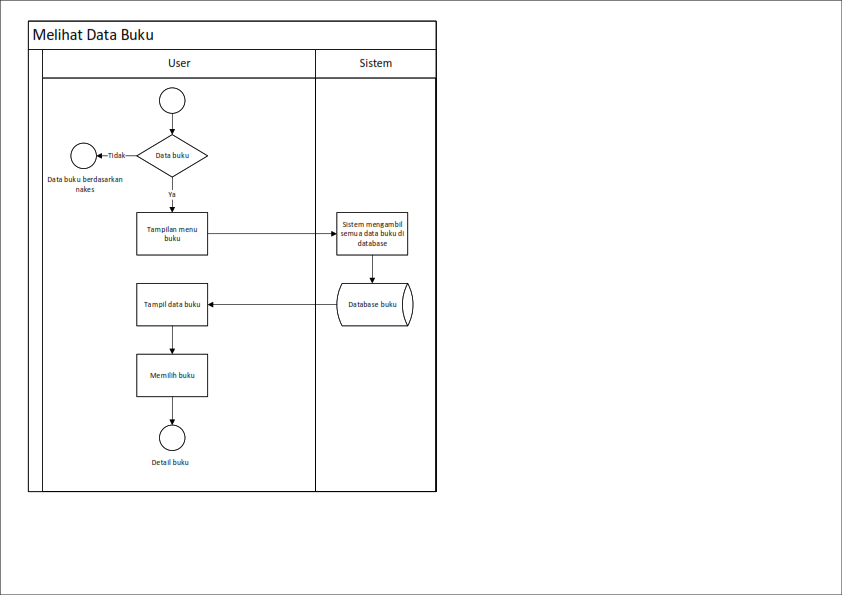
### Gambar 4.1.8.4. Gambar Activity Diagram Mengelola Nilai Buku

Gambar 4.1.8.4 merupakan activity diagram dalam mengelola nilai buku, admin dapat menambah data nilai pada kriteria (kelayakan isi, kebahasaan, penyajian, dan kegrafikaan), mengubah, dan menghapus data nilai buku yang tersimpan pada database.



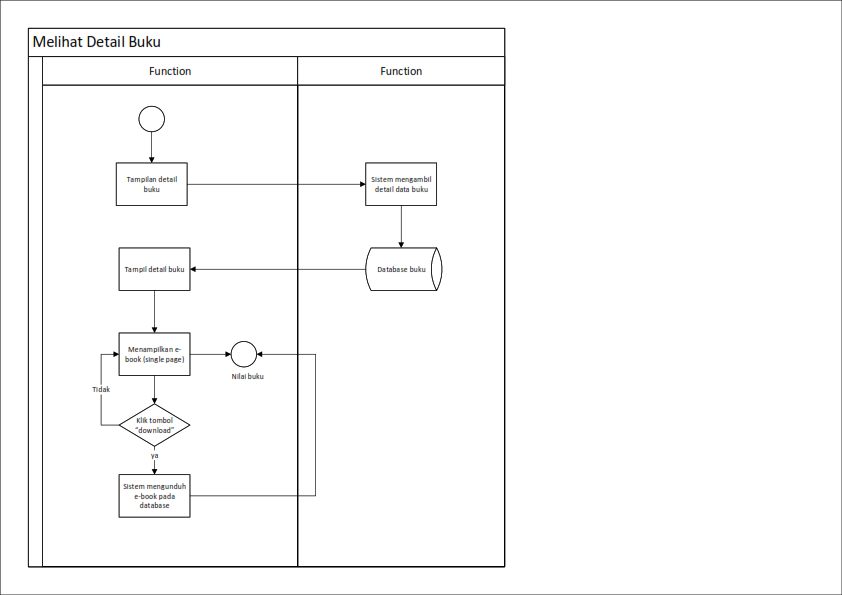
### Gambar 4.1.8.5. Gambar Activity Diagram Login User

Gambar 4.1.8.5 merupakan activity diagram login yaitu user memasukan username dan password untuk melakukan login sistem. Sistem akan melakukan validasi. Jika sukses, akan di arahkan ke dalam aplikasi



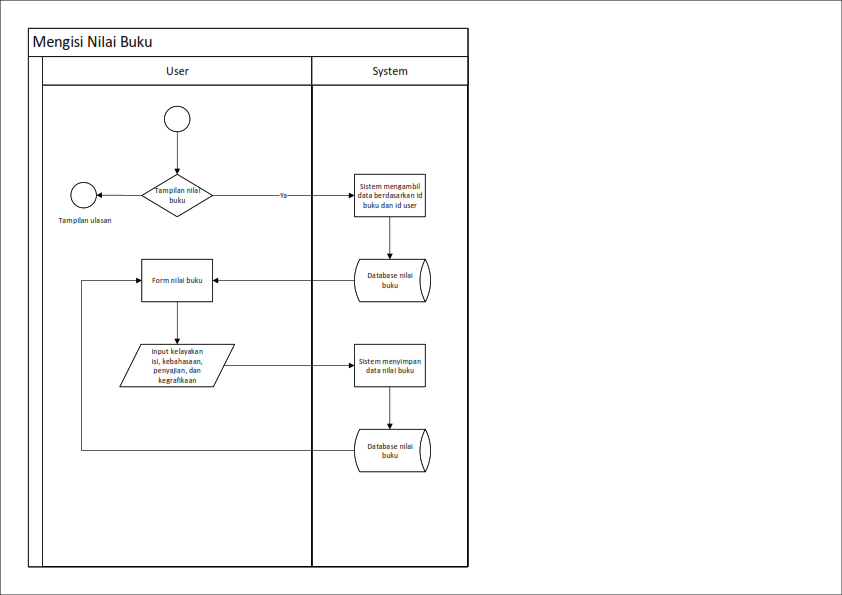
### Gambar 4.1.8.6. Gambar Activity Diagram Melihat Data Buku

Gambar 4.1.8.6 merupakan activity diagram melihat data buku yaitu user memilih judul buku yang diinginkan. Jika user sudah memilih, sistem akan menampilkan detail buku yang terdapat pada gambar 4.1.8.7.



### Gambar 4.1.8.7. Gambar Activity Diagram Melihat Detail Buku

Gambar 4.1.8.7 merupakan activity diagram melihat detail buku yaitu user melihat detail buku tersebut berdasarkan data yang ada pada database. User dapat mengunduh buku dengan menekan tombol “download”.



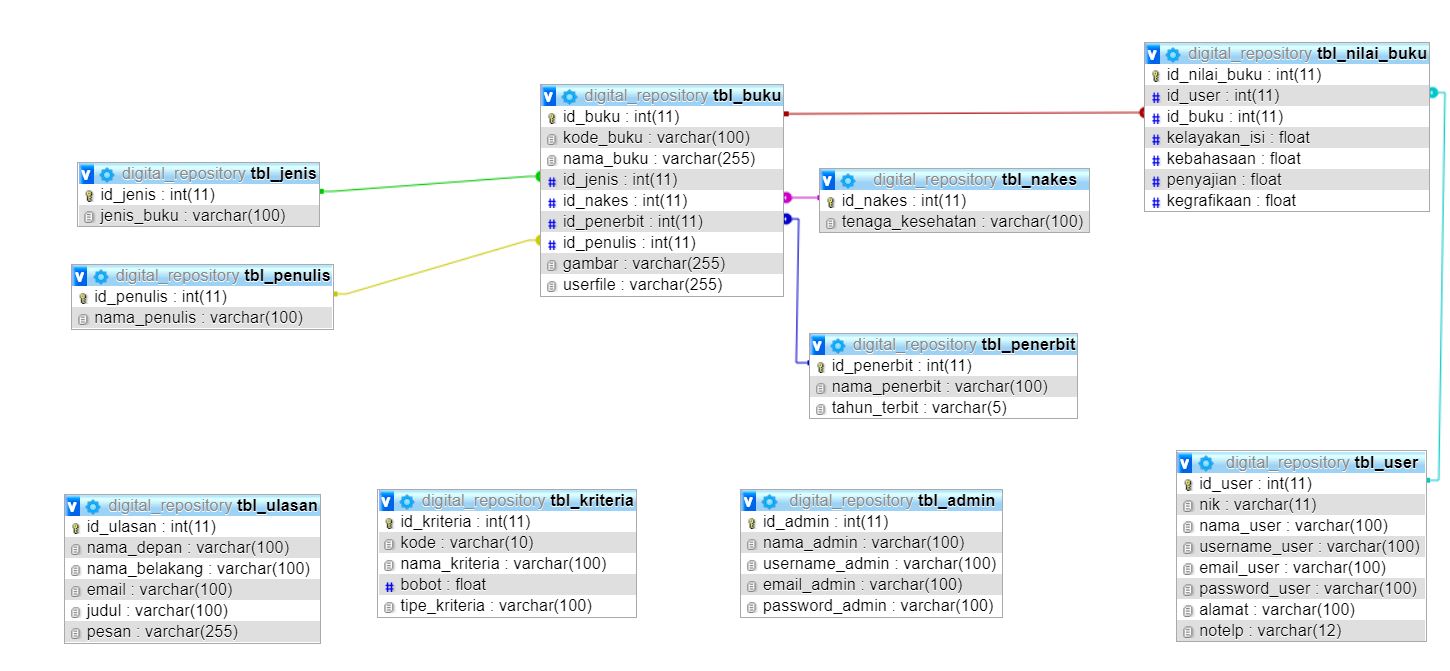
### Gambar 4.1.8.8. Gambar Activity Diagram Mengisi Nilai Buku

Gambar 4.1.8.8 merupakan activity diagram mengisi nilai buku yaitu user mengisi nilai buku berdasarkan kriteria (kelayakan isi, kebahasaan, penyajian, dan kegrafikaan). Nilai tersebut akan di hitung dengan metode VIKOR dan hasil ranking metode VIKOR tersebut akan ditampilkan di menu rekomendasi buku.

## **4.2. Perancangan**

### 4.2.1. Perancangan Basis Data

Berikut merupakan perancangan basis data melalui desain yang dapat dilihat pada gambar 4.2.1.1.



### Gambar 4.2.1.1. Gambar Perancangan Basis Data

### Tabel 4.2.1.1. Tabel Admin

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_admin | Int(11) | Not null, primary key |
| Nama\_admin | Varchar(100) | Not null |
| Username\_admin | Varchar(100) | Not null |
| Email\_admin | Varchar(100) | Not null |
| Password\_admin | Varchar(100) | Not null |

### Tabel 4.2.1.2. Tabel User

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_user | Int(11) | Not null, primary key |
| Nik | Varchar(11) | Not null |
| Nama\_user | Varchar(100) | Not null |
| Username\_user | Varchar(100) | Not null |
| Email\_user | Varchar(100) | Not null |
| Password\_user | Varchar(100) | Not null |
| Alamat | Varchar(100) | Not null |
| notelp | Varchar(12) | Not null |

### Tabel 4.2.1.3. Tabel Ulasan

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_ulasan | Int(11) | Not null, primary key |
| Nama\_depan | Varchar(100) | Not null |
| Nama\_belakang | Varchar(100) | Not null |
| email | Varchar(100) | Not null |
| Judul | Varchar(100) | Not null |
| pesan | Varchar(255) | Not null |

### Tabel 4.2.1.4. Tabel Jenis Buku

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_jenis\_buku | Int(11) | Not null, primary key |
| Jenis\_buku | Varchar(100) | Not null |

### Tabel 4.2.1.5. Tabel Tenaga Kesehatan

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_nakes | Int(11) | Not null, primary key |
| Tenaga\_kesehatan | Varchar(100) | Not null |

### Tabel 4.2.1.6. Tabel Penerbit

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_penerbit | Int(11) | Not null, primary key |
| Nama\_penerbit | Varchar(100) | Not null |
| Tahun\_terbit | Varchar(100) | Not null |

### Tabel 4.2.1.7. Tabel Penulis

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_penulis | Int(11) | Not null, primary key |
| Nama\_penulis | Varchar(100) | Not null |

### Tabel 4.2.1.8. Tabel Buku

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_buku | Int(11) | Not null, primary key |
| Kode\_buku | Varchar(100) | Not null |
| Nama\_buku | Varchar(255) | Not null |
| Id\_jenis | Int(11) | Not null, foreign key |
| Id\_nakes | Int(11) | Not null, foreign key |
| Id\_penerbit | Int(11) | Not null, foreign key |
| Id\_penulis | Int(11) | Not null, foreign key |
| Gambar | Varchar(255) | Not null |
| userfile | Varchar(255) | Not null |

### Tabel 4.2.1.9. Tabel Kriteria

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_kriteria | Int(11) | Not null, primary key |
| Kode | Varchar(100) | Not null |
| Nama\_kriteria | Varchar(100) | Not null |
| bobot | Float | Not null |
| Tipe kriteria | Varchar(100) | Not null |

### Tabel 4.2.1.10. Tabel Nilai Buku

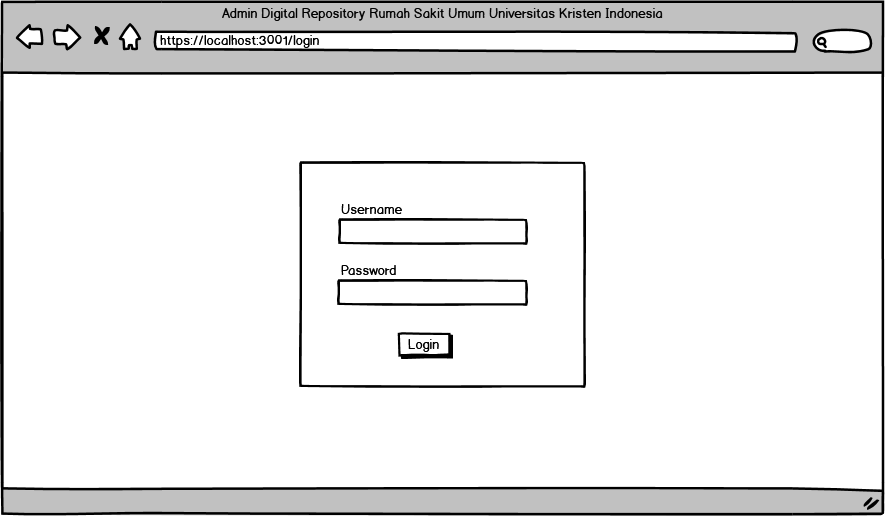
|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_nilai\_buku | Int(11) | Not null, primary key |
| Id\_user | Int(11) | Not null, foreign key |
| Id\_buku | Int(11) | Not null, foreign key |
| Kelayakan\_isi | Float | Not null |
| Kebahasaan | Float | Not null |
| Penyajian | Float | Not null |
| kegrafikaan | Float | Not null |

### 4.2.2. Perancangan Antar Muka

Perancangan *user interface* dijelaskan dalam bentuk tampilan mockup meliputi halaman admin dan user (karyawan)

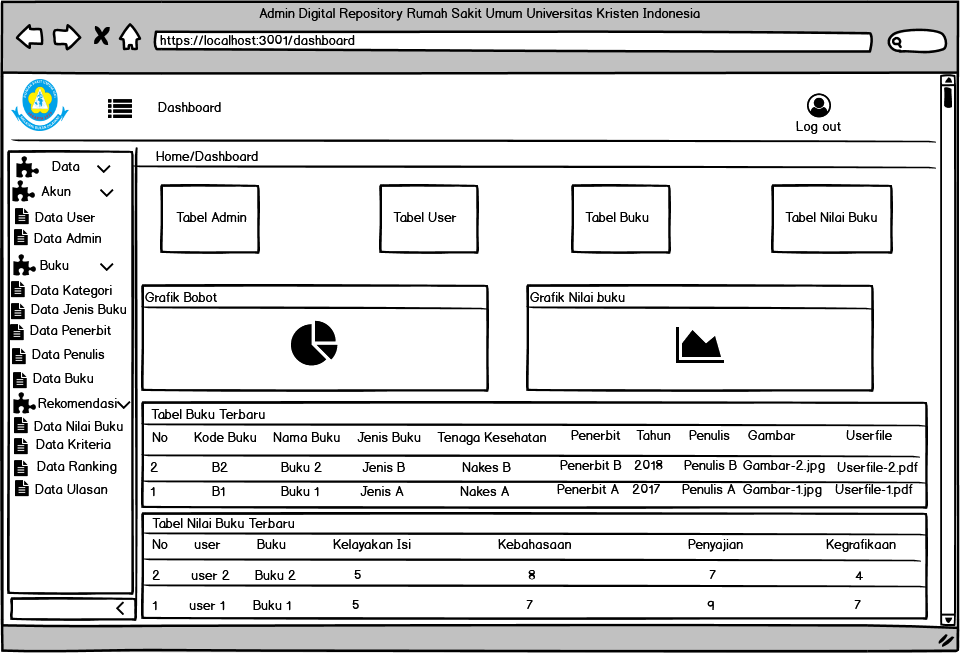
### 4.2.2.1. Halaman Admin

a. tampilan login

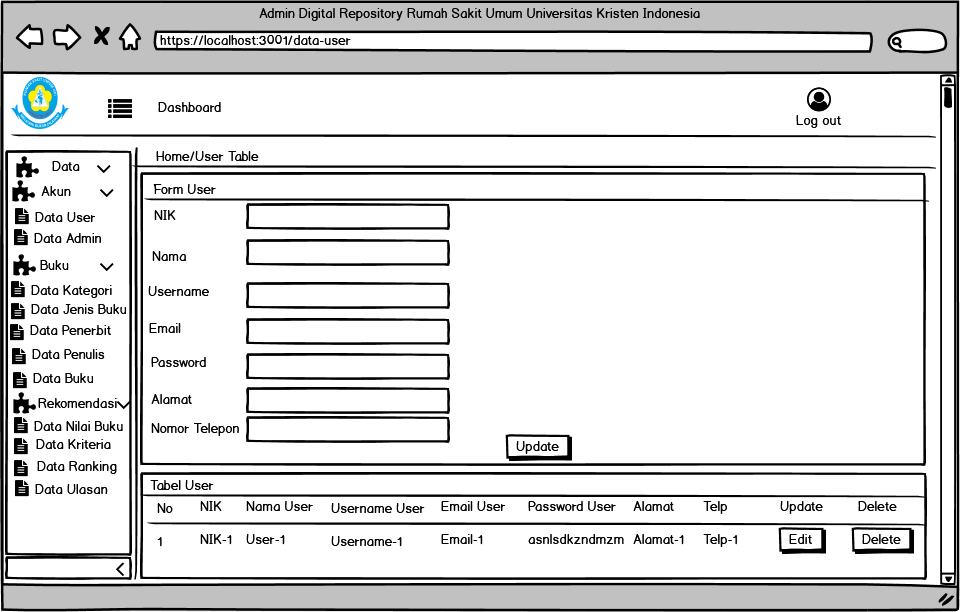


### Gambar 4.2.2.1.1. Gambar tampilan login

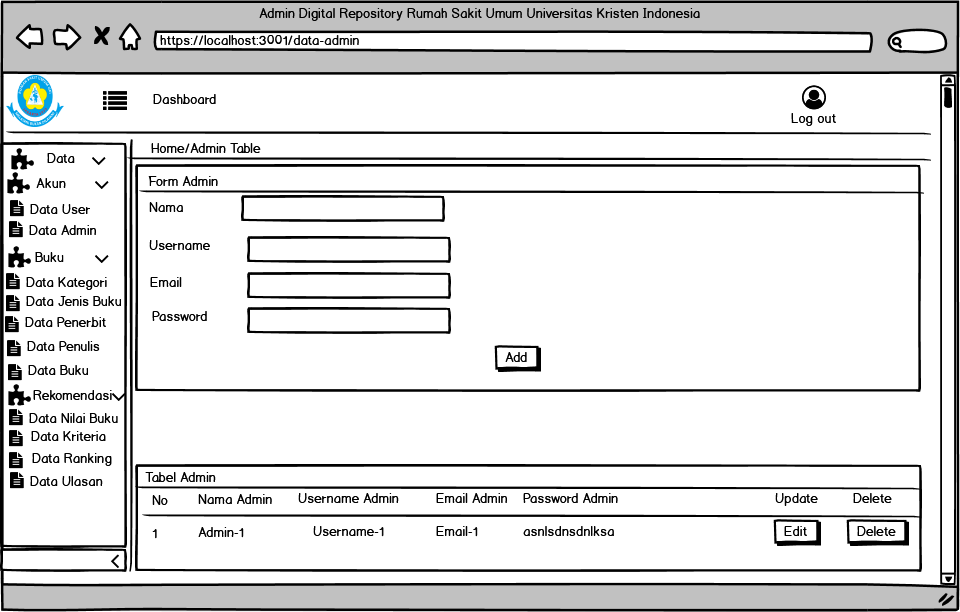
1. tampilan dashboard



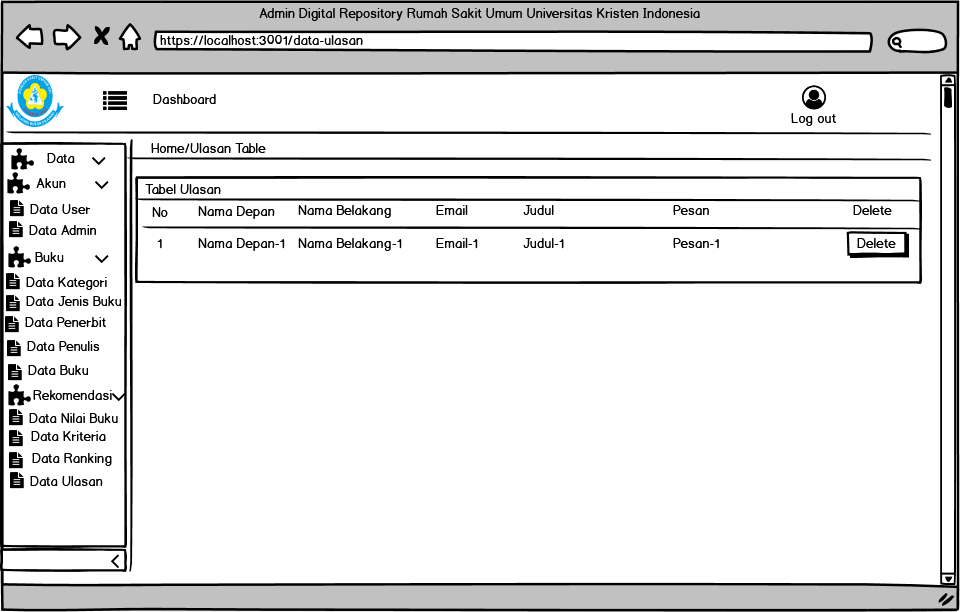
### Gambar 4.2.2.1.2. Gambar tampilan dashboard

1. tampilan data user

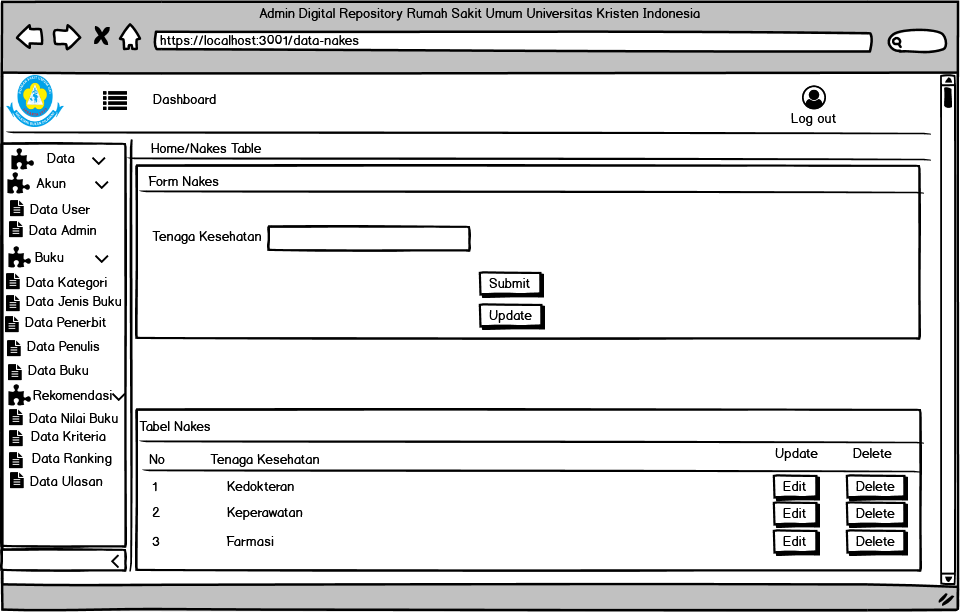
### Gambar 4.2.2.1.3. Gambar tampilan data user

1. tampilan data admin

### Gambar 4.2.2.1.4. Gambar tampilan data admin

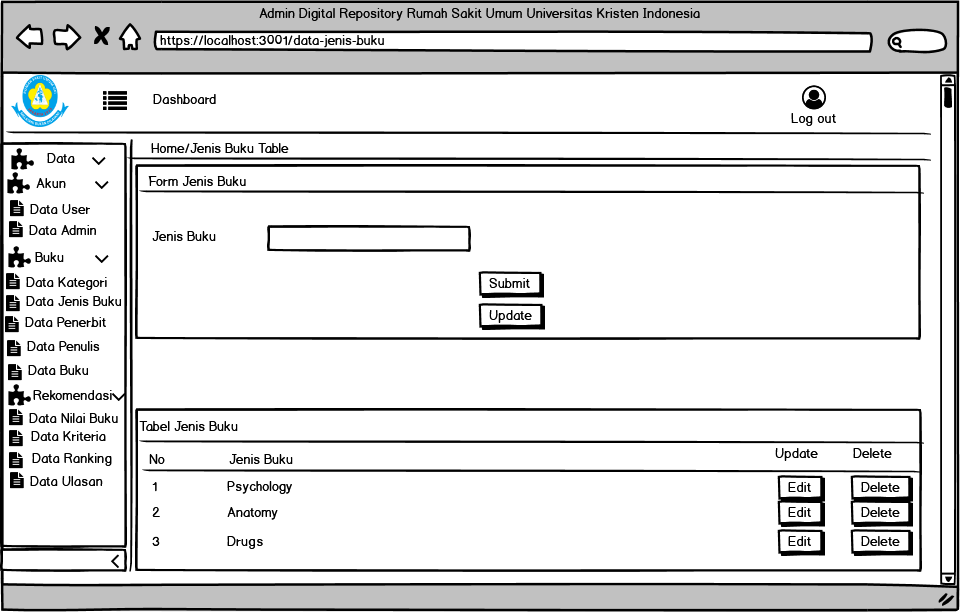
1. tampilan data ulasan

### Gambar 4.2.2.1.5. Gambar tampilan data ulasan

1. tampilan data nakes  
     
   

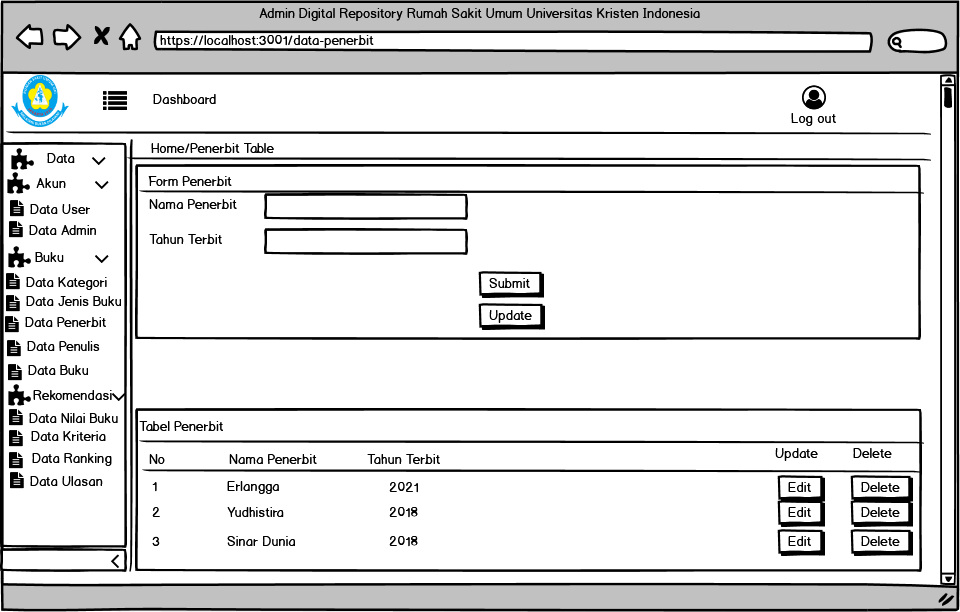
### Gambar 4.2.2.1.6. Gambar tampilan data nakes

1. tampilan data jenis buku



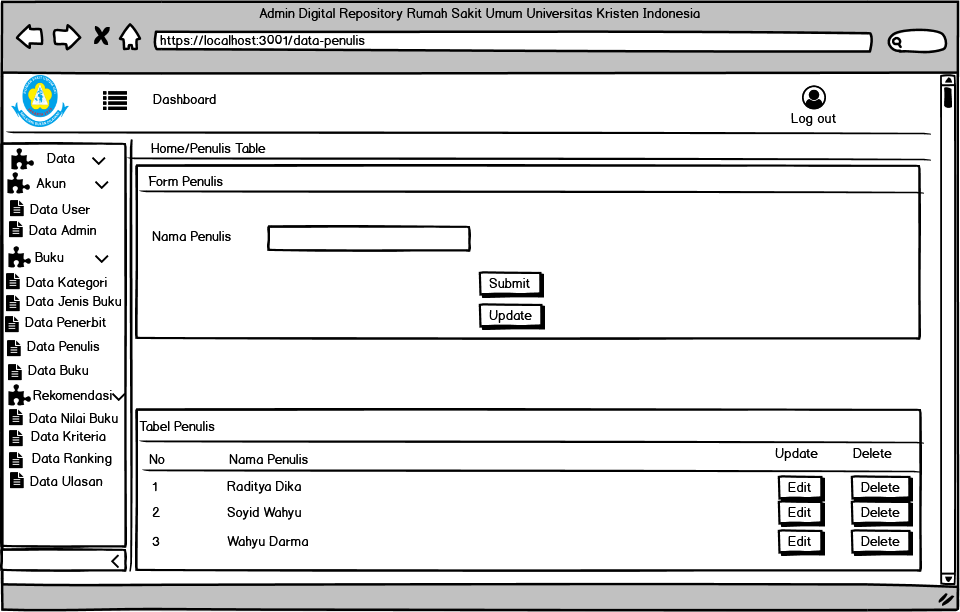
### Gambar 4.2.2.1.7. Gambar tampilan data jenis buku

1. tampilan data penerbit



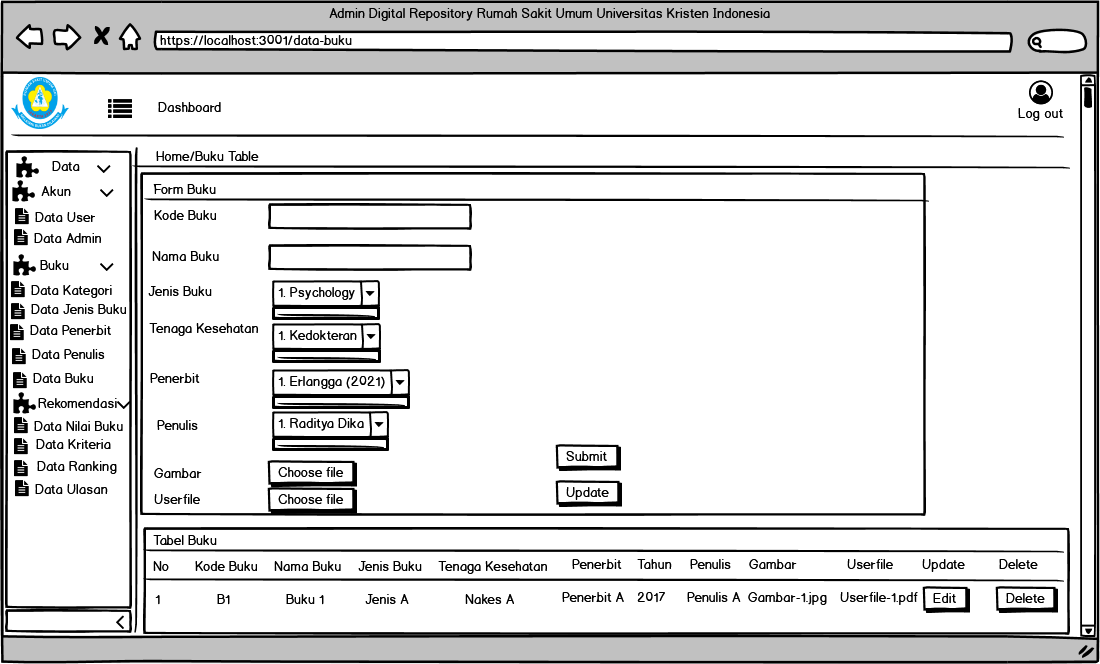
### Gambar 4.2.2.1.8. Gambar tampilan data penerbit

1. tampilan data penulis



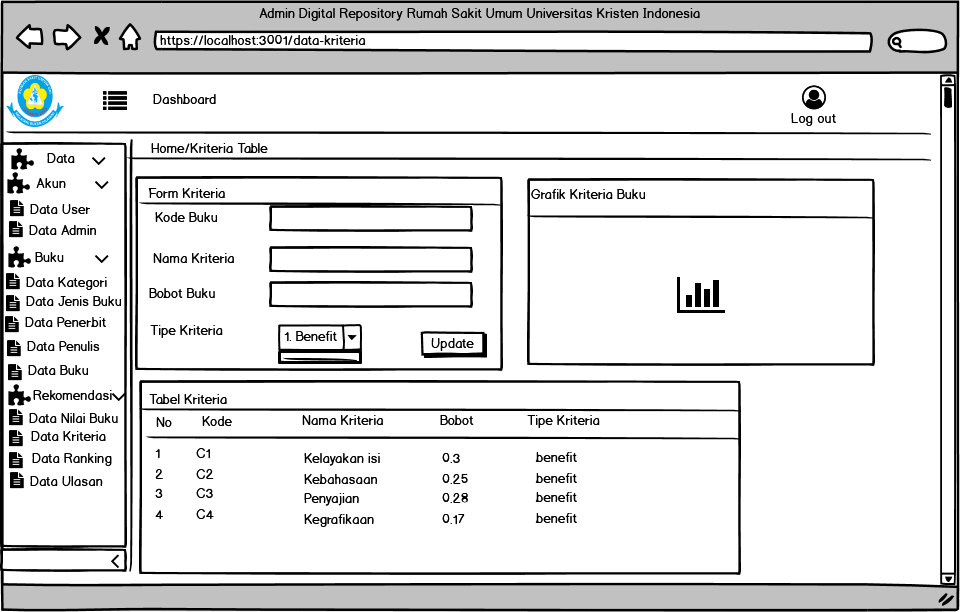
### Gambar 4.2.2.1.9. Gambar tampilan data penulis

1. tampilan data buku



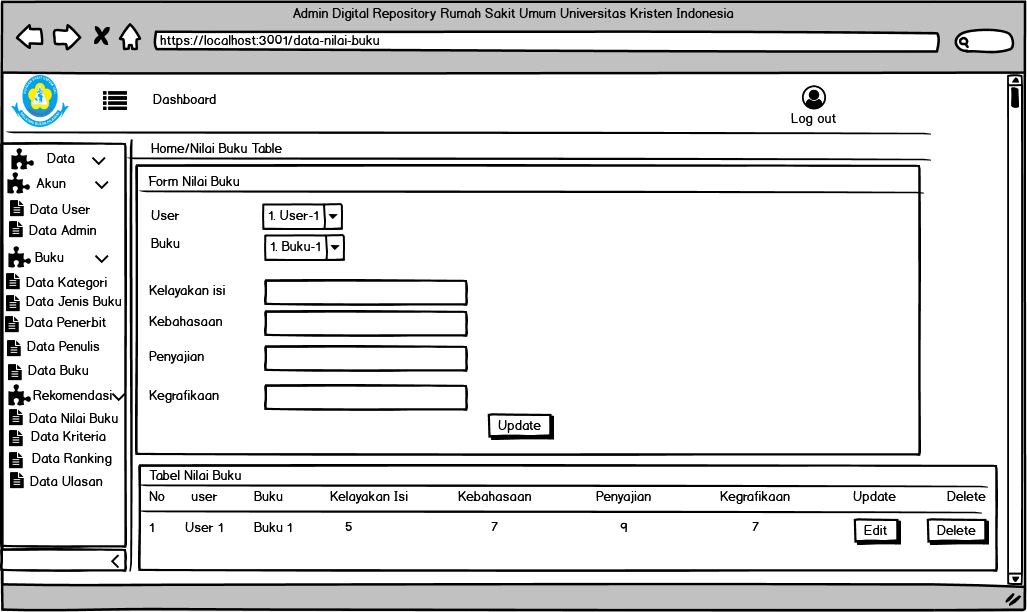
### Gambar 4.2.2.1.10. Gambar tampilan data buku

1. tampilan data kriteria



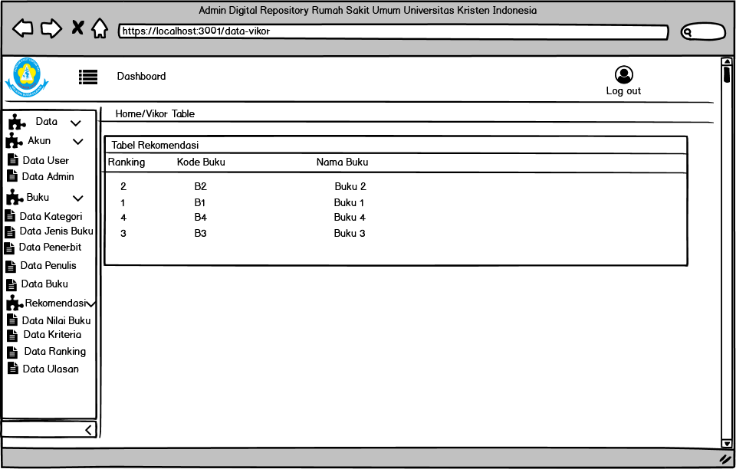
### Gambar 4.2.2.1.11. Gambar tampilan data kriteria

1. tampilan data nilai buku



### Gambar 4.2.2.1.12. Gambar tampilan data nilai buku

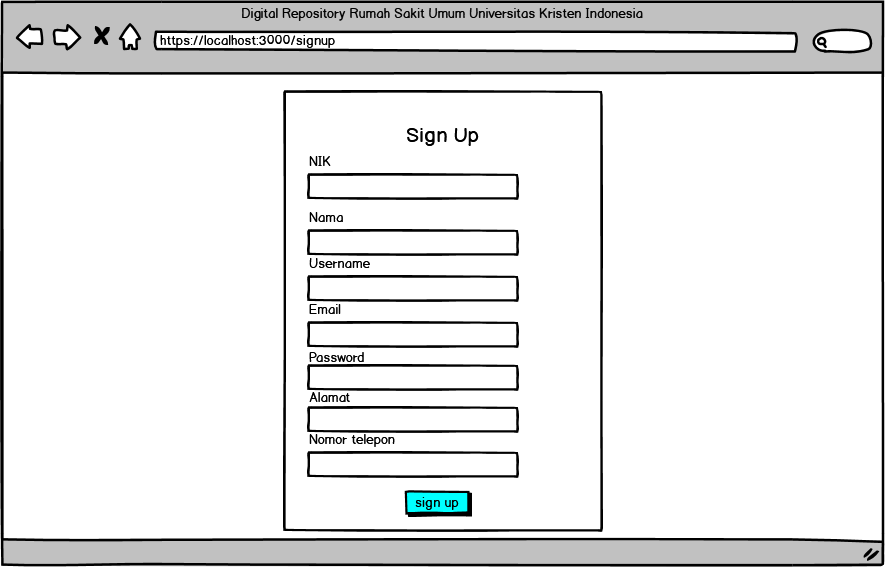
1. tampilan data Rekomendasi



### Gambar 4.2.2.1.13. Gambar tampilan data Rekomendasi

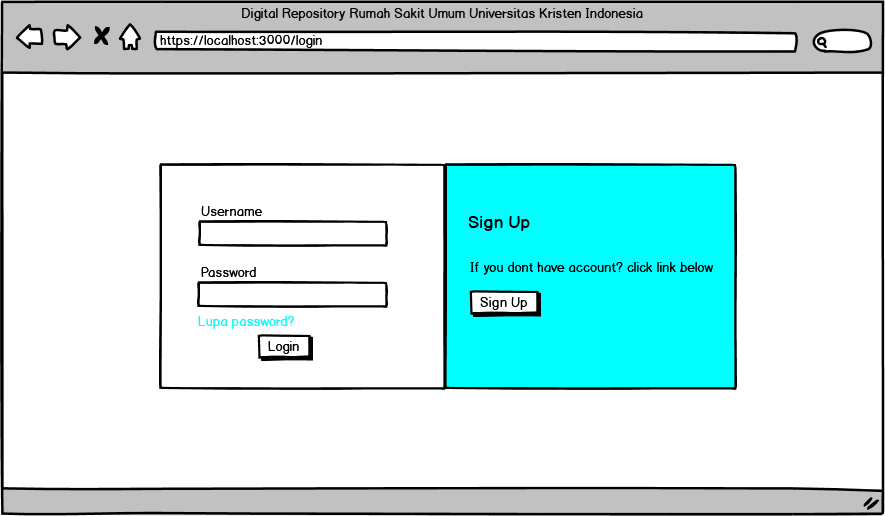
### 4.2.2.2. Halaman user (karyawan)

* Tampilan Register



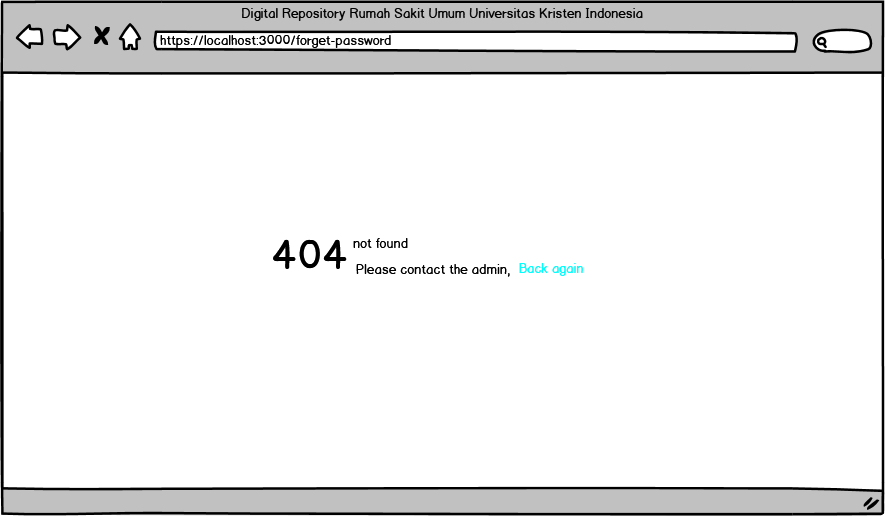
### Gambar 4.2.2.2.1. Gambar tampilan register

* Tampilan Login



### Gambar 4.2.2.2.2. Gambar tampilan login

* Tampilan lupa password



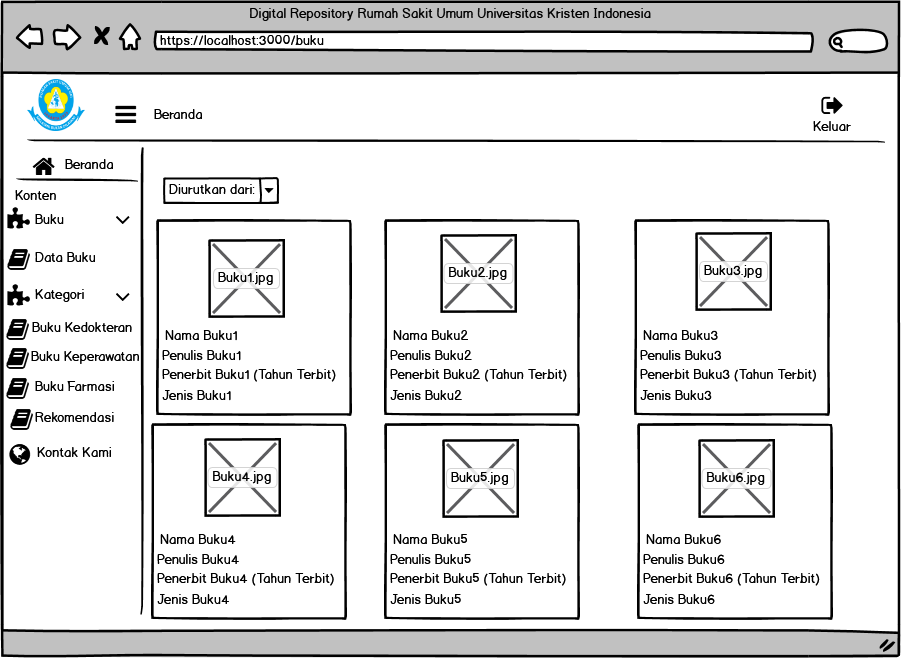
### Gambar 4.2.2.2.3. Gambar tampilan lupa password

* Tampilan Beranda



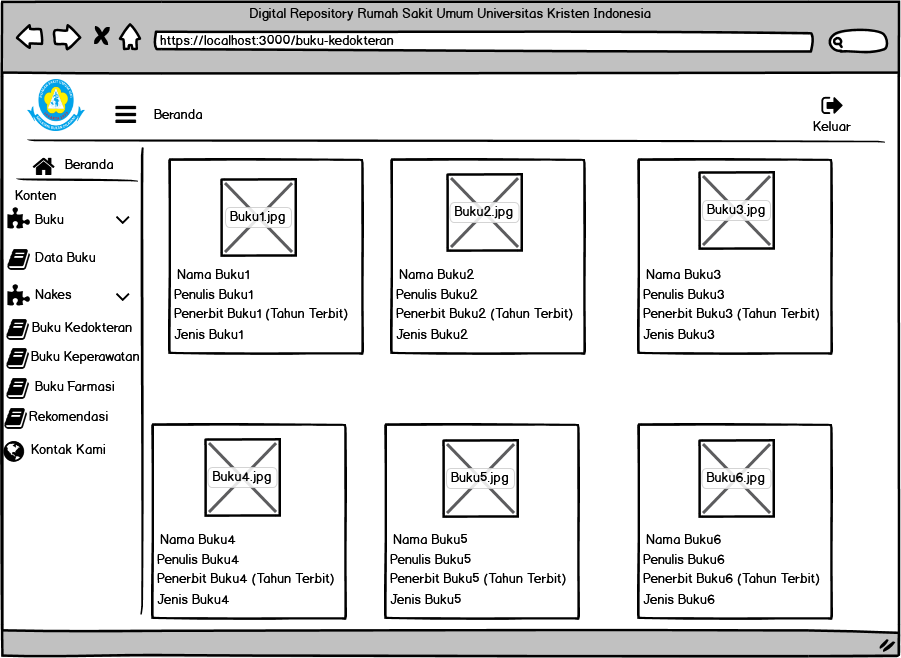
### Gambar 4.2.2.2.4. Gambar tampilan beranda

* Tampilan data buku



### Gambar 4.2.2.2.5. Gambar tampilan data buku

* Tampilan data buku kedokteran



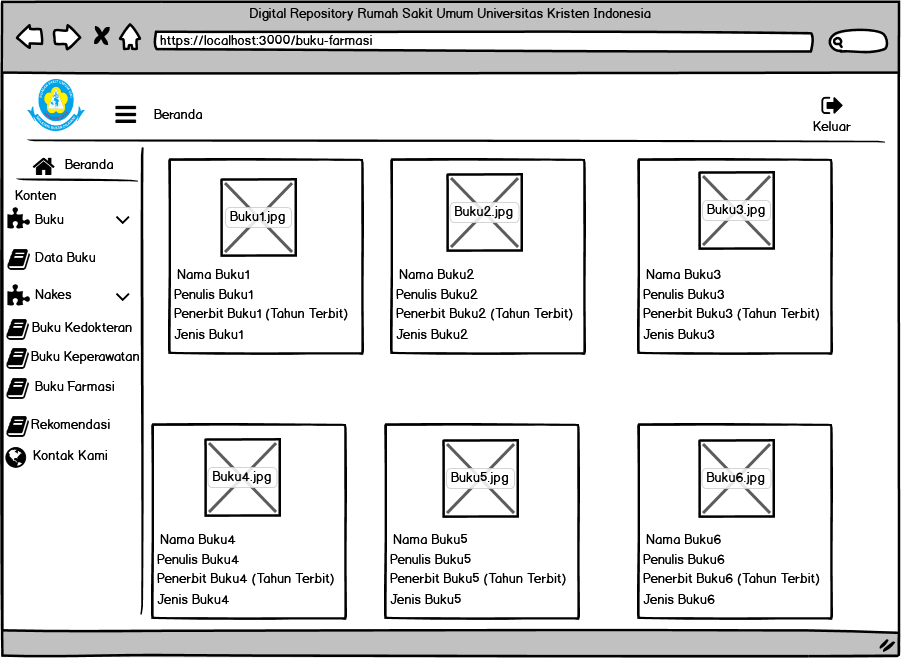
### Gambar 4.2.2.2.6. Gambar tampilan data buku kedokteran

* Tampilan data buku keperawatan



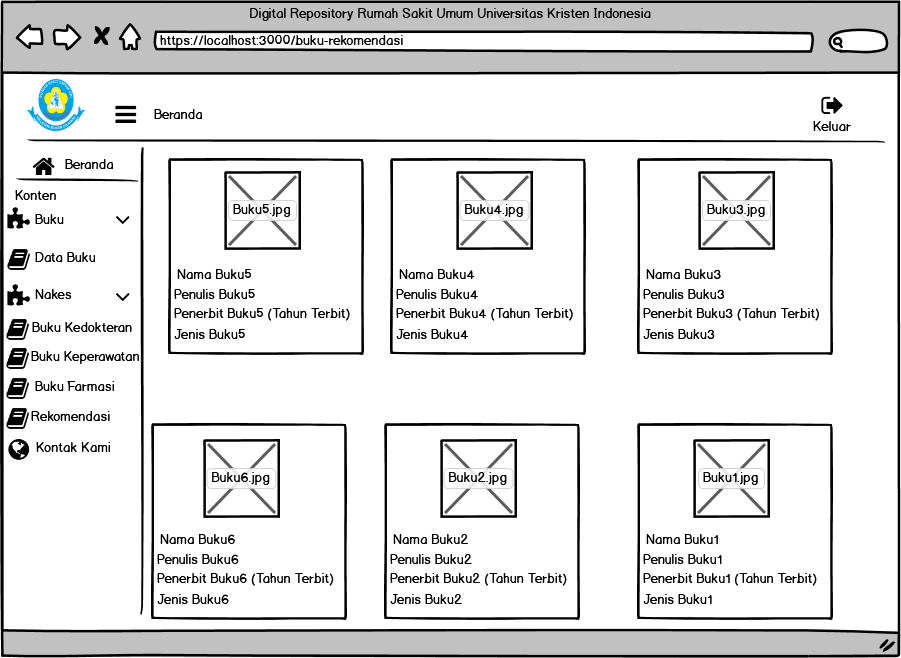
### Gambar 4.2.2.2.7. Gambar tampilan data buku keperawatan

* Tampilan data buku farmasi



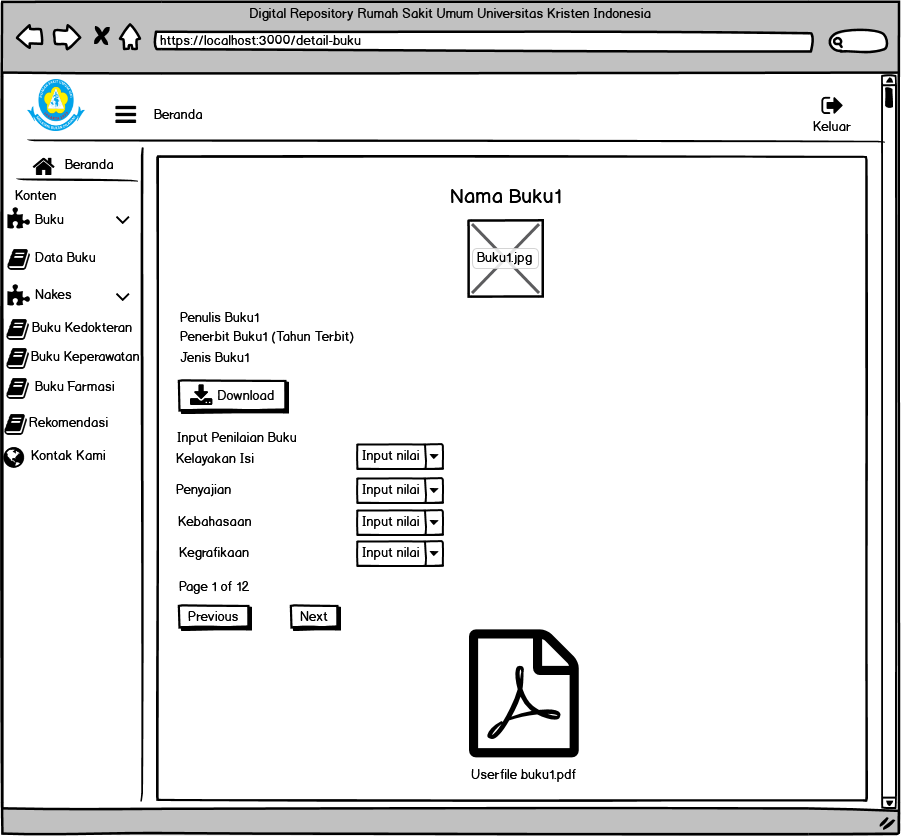
### Gambar 4.2.2.2.8. Gambar tampilan data buku farmasi

* Tampilan data buku rekomendasi



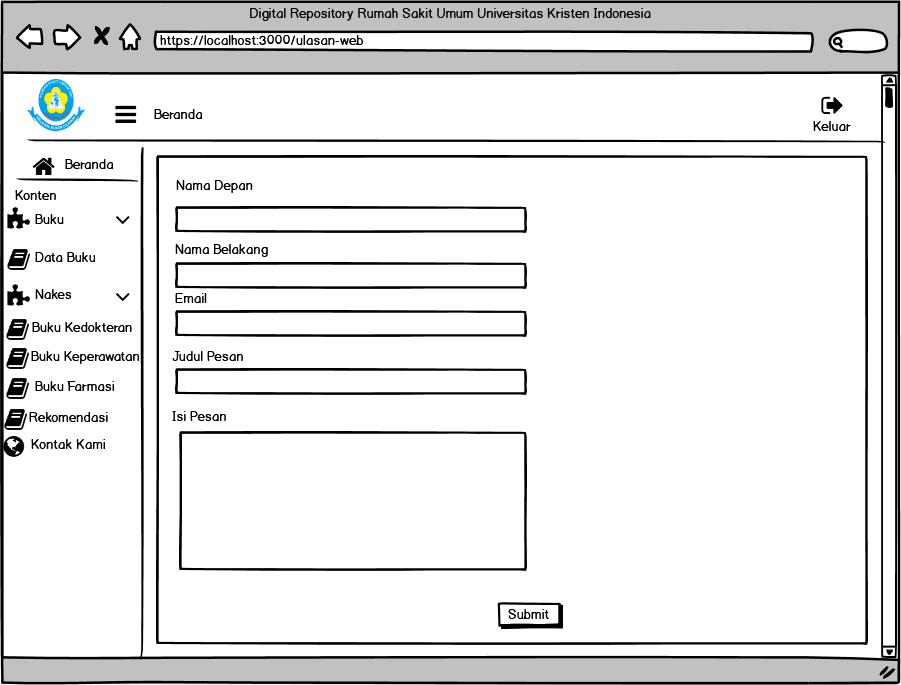
### Gambar 4.2.2.2.9. Gambar tampilan data buku rekomendasi

* Tampilan detail buku



### Gambar 4.2.2.2.10. Gambar tampilan detail buku

* Tampilan ulasan buku

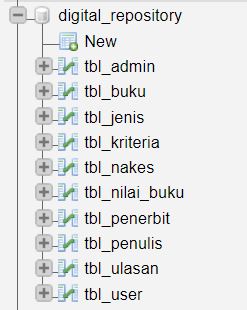


### Gambar 4.2.2.2.11. Gambar tampilan ulasan buku

# **BAB V. IMPLEMENTASI DAN PENGUJIAN**

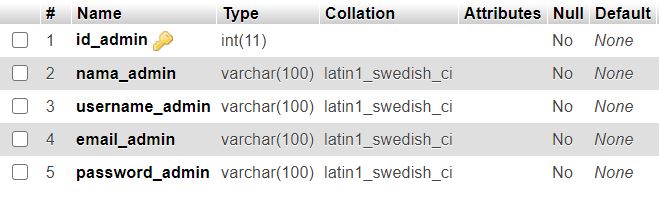
## **Implementasi Basis Data**

Berikut ini merupakan penjelasan mengenai implementasi *database* yang telah dilakukan pada pembuatan sistem aplikasi ini :



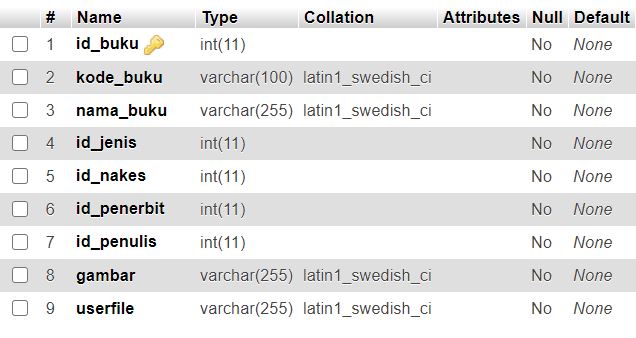
### Gambar 5.1.1. Gambar Desain Basis Data

Pada Gambar 5.1.1 merupakan implementasi *database* yang digunakan untuk menyimpan data dan digunakan dalam pembuatan aplikasi. Tabel yang digunakan meliputi tbl\_admin, tbl\_buku, tbl\_jenis, tbl\_kriteria, tbl\_tbl\_nakes, tbl\_nilai\_buku, tbl\_penerbit, tbl\_penulis, tbl\_ulasan, tbl\_user. Berikut merupakan penjelasan setiap tabelnya:



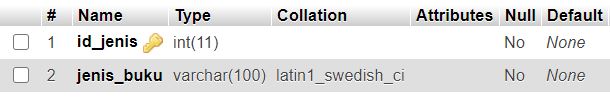
### Gambar 5.1.2. Gambar tabel admin

Gambar 5.1.2 merupakan tabel dalam menyimpan data admin. Tabel ini berisikan id\_admin, nama admin, username\_admin, email\_admin, password\_admin.



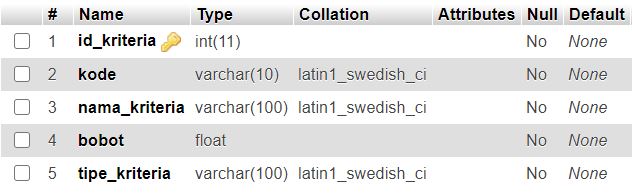
### Gambar 5.1.3. Gambar tabel buku

Gambar 5.1.3 merupakan tabel menyimpan data buku, tabel ini berisi id\_buku, kode\_buku, nama\_buku, id\_jenis, id\_nakes, id\_penerbit, id\_penulis, gambar, dan userfile.



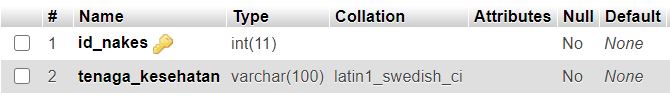
### Gambar 5.1.4. Gambar tabel jenis buku

Gambar 5.1.4 merupakan tabel untuk menyimpan data jenis buku. Tabel ini berisi id\_jenis dan jenis buku.



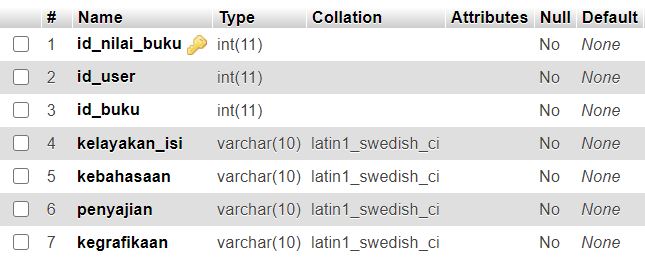
### Gambar 5.1.5. Gambar tabel kriteria

Gambar 5.1.5 merupakan tabel untuk menyimpan kriteria. Tabel ini berisi id\_kriteria, kode, nama\_kriteria, bobot, tipe\_kriteria.



### Gambar 5.1.6. Gambar tabel tenaga kesehatan

Gambar 5.1.6 merupakan tabel untuk menyimpan data tenaga kesehatan. Tabel ini berisi id\_nakes dan tenaga\_kesehatan.



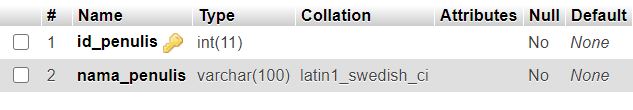
### Gambar 5.1.7. Gambar tabel nilai buku

Gambar 5.1.7 merupakan tabel untuk menyimpan data nilai buku yang dimana data nilai buku ini akan dihitung dengan menggunakan metode VIKOR dan hasilnya akan ditampilkan di menu rekomendasi buku. Tabel ini berisi id\_nilai\_buku, id\_user, id\_buku, kelayakan\_isi, kebahasaan, penyajian, dan kegrafikaan.



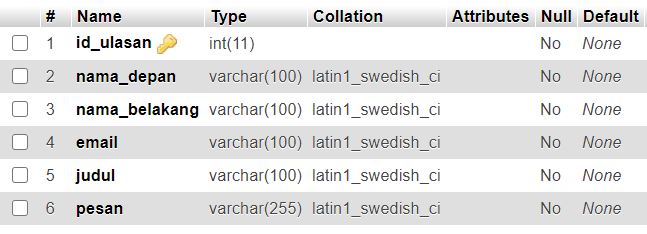
### Gambar 5.1.8. Gambar tabel penerbit

Gambar 5.1.8 merupakan tabel untuk menyimpan data penerbit, tabel ini berisi id\_penerbit, nama\_penerbit, dan tahun\_terbit.



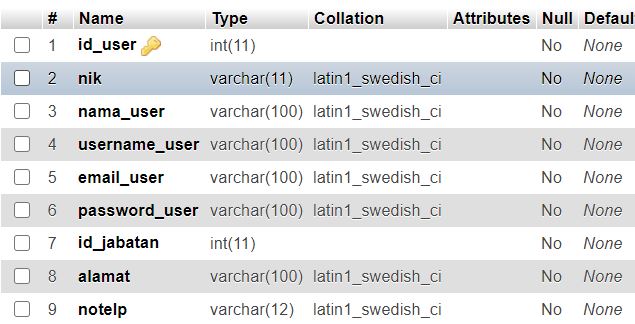
### Gambar 5.1.9. Gambar tabel penulis

Gambar 5.1.9 merupakan tabel untuk menyimpan data penulis, tabel ini berisi id\_penulis dan nama\_penulis.



### Gambar 5.1.10. Gambar tabel ulasan

Gambar 5.1.10 merupakan tabel untuk menyimpan data ulasan, tabel ini berisi id\_ulasan, nama\_depan, nama\_belakang, email, judul, dan pesan.



### Gambar 5.1.11. Gambar tabel user

Gambar 5.1.11 merupakan tabel untuk menyimpan data user, tabel ini berisi id\_user, nik, nama\_user, username\_user, email\_user, password\_user, alamat, dan notelp.

## **Implementasi Pengkodean Sistem**

Aplikasi ini dibuat dan dibangun dengan menggunakan Bahasa pemrograman Python dan Javascript dengan menggunakan *framework Flask dan Javascript.* Implementasi yang dilakukan sesuai dengan analisis yang telah dilakukan sebelumnya. Beberapa proses dilakukan berikut :

1. Membuat API untuk melakukan pengolahan data semua tabel pada database dengan menggunakan flask sebagai backend dan web server
2. Membuat perhitungan metode VIKOR dan diintegrasikan ke API supaya terhubung dengan database
3. Membuat tampilan website dengan menggunakan ReactJS sebagai frontend

**Backend**

1. Deklarasi Database : Deklarasi Database merupakan proses untuk menghubungkan backend dengan database sehingga pengolahan data menjadi mudah Proses menghubungkan database yaitu: *(terlampir pada lampiran 5)*
2. Mengolah user : Mengolah user merupakan proses login dan register user yang terdapat pada database. Proses mengolah user dibagi menjadi beberapa potongan yaitu: *(terlampir pada lampiran 5)*
3. Mengolah buku : Mengolah buku merupakan proses melihat, menambah, mengubah, dan menghapus data user yang terdapat pada database. Proses mengolah user dibagi menjadi beberapa potongan yaitu: *(terlampir pada lampiran 5)*
4. Perhitungan metode vikor : Pada proses ini untuk menghitung nilai buku dengan menggunakan metode VIKOR *(terlampir pada lampiran 5)*
5. Mengolah rekomendasi buku : Pada proses ini adalah untuk menghasilkan ranking dari metode VIKOR untuk ditampilkan pada aplikasi Digital Repository Rumah Sakit Umum Universitas Kristen Indonesia *(terlampir pada lampiran 5)*

**Frontend**

1. Fungsi mengolah kriteria buku: Pada proses ini untuk mengolah data berupa proses melihat, menambah, mengubah, dan menghapus data kriteria buku menggunakan library *axios* pada ReactJS *(terlampir pada lampiran 5)*
2. Fungsi mengolah buku: Pada proses ini untuk mengolah data berupa proses melihat, menambah, mengubah, dan menghapus data buku menggunakan library *axios* pada ReactJS *(terlampir pada lampiran 5)*
3. Tampilan buku (User): Pada proses ini untuk menampilkan data buku pada website user sehingga user dapat memilih buku mana yang ingin user baca dan user nilai *(terlampir pada lampiran 5)*
4. Tampilan rekomendasi buku (user): Pada proses ini untuk menampilkan data rekomendasi buku sehingga user mengetahui buku yang menjadi rekomendasi untuk user dapat baca *(terlampir pada lampiran 5)*

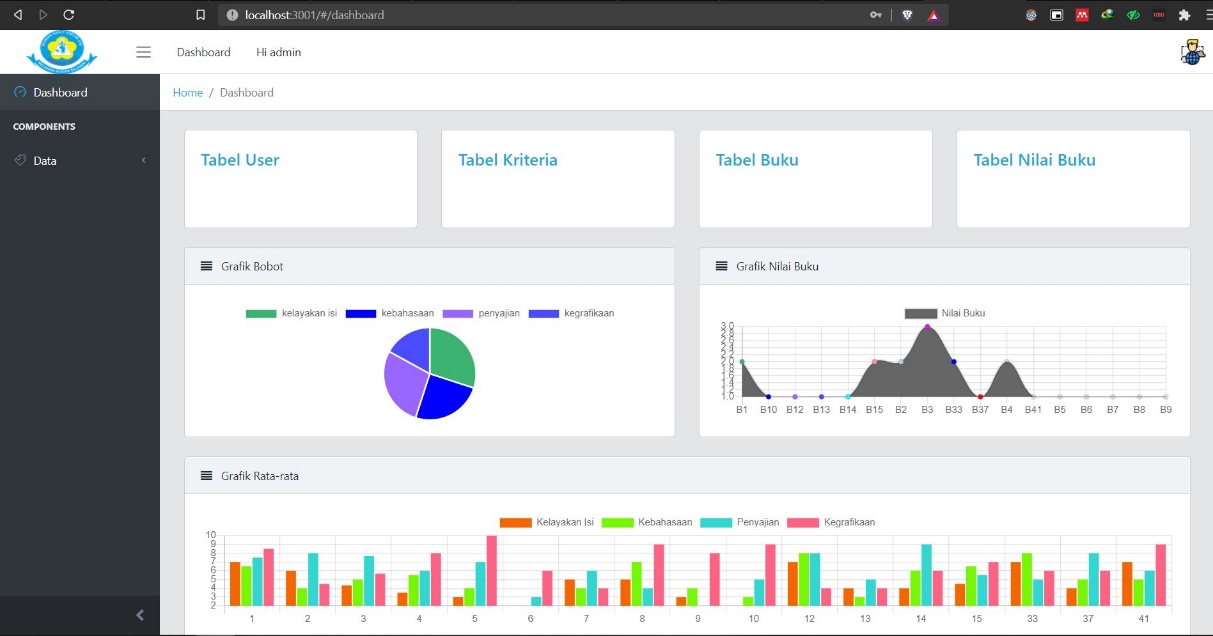
## **Implementasi tampilan**

Implementasi tampilan dibuat berdasarkan perancangan antar muka yang telah dibuat sebelumnya.

### **Tampilan Admin**

1. Tampilan Dashboard

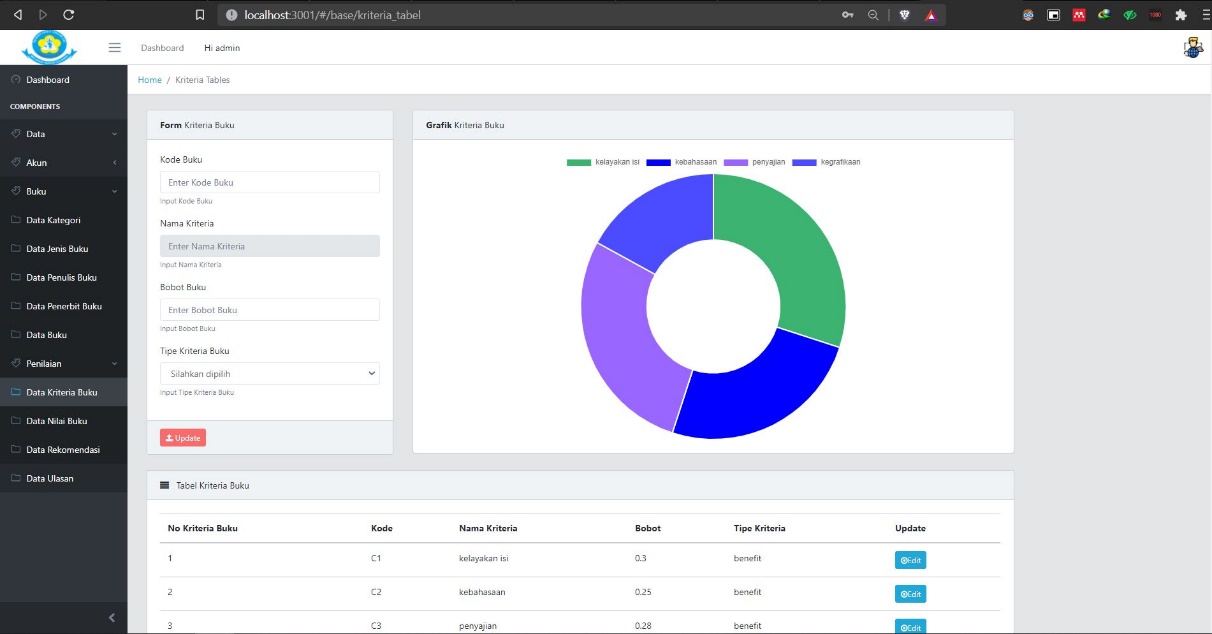
Tampilan Dashboard menampilkan data buku terbaru, grafik data buku, grafik, data kriteria, grafik data penilaian buku berdasarkan kriteria penilaian buku.



### Gambar 5.3.1.1. Gambar Tampilan Dashboard

1. Tampilan Kriteria Buku

Tampilan Kriteria Buku menampilkan kriteria data buku dan bobot kriteria, admin dapat mengubah bobot kriteria dan tipe kriteria yang dimana akan mempengaruhi pada perhitungan metode VIKOR



### Gambar 5.3.1.2. Gambar Tampilan Kriteria Buku

1. Tampilan rekomendasi buku

Tampilan rekomendasi buku menampilkan hasil ranking dari metode VIKOR dan admin dapat memonitoring hasil perangkingan dengan menggunakan metode VIKOR di website

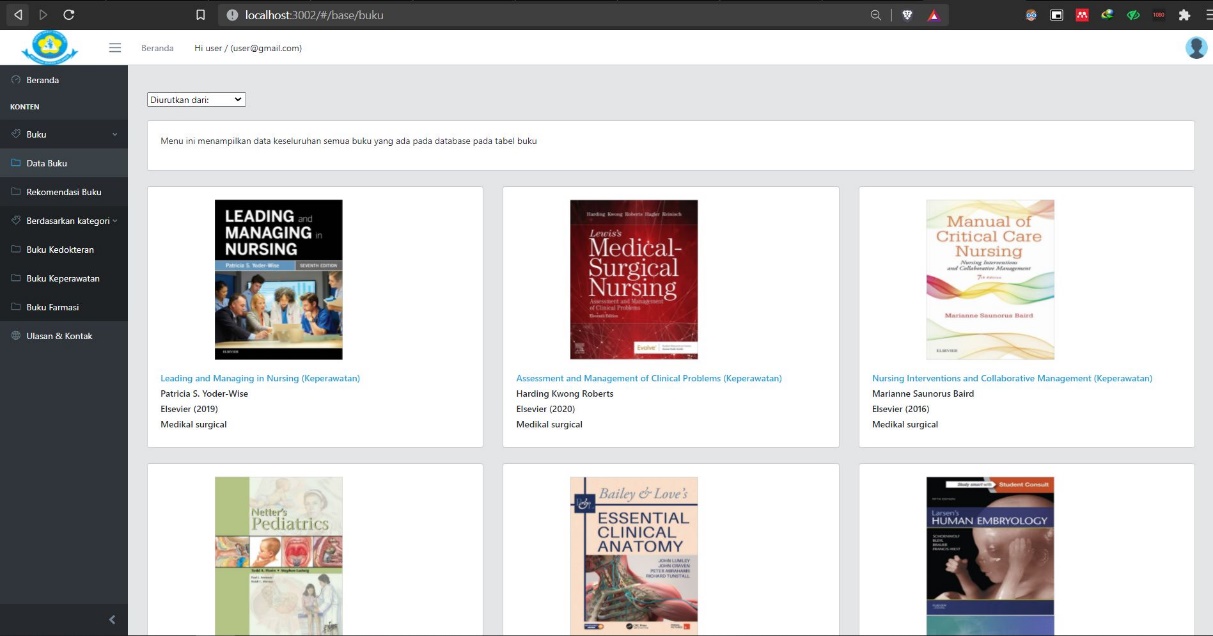


### Gambar 5.3.1.3. Gambar Tampilan Rekomendasi Buku

### **Tampilan User**

1. Tampilan buku

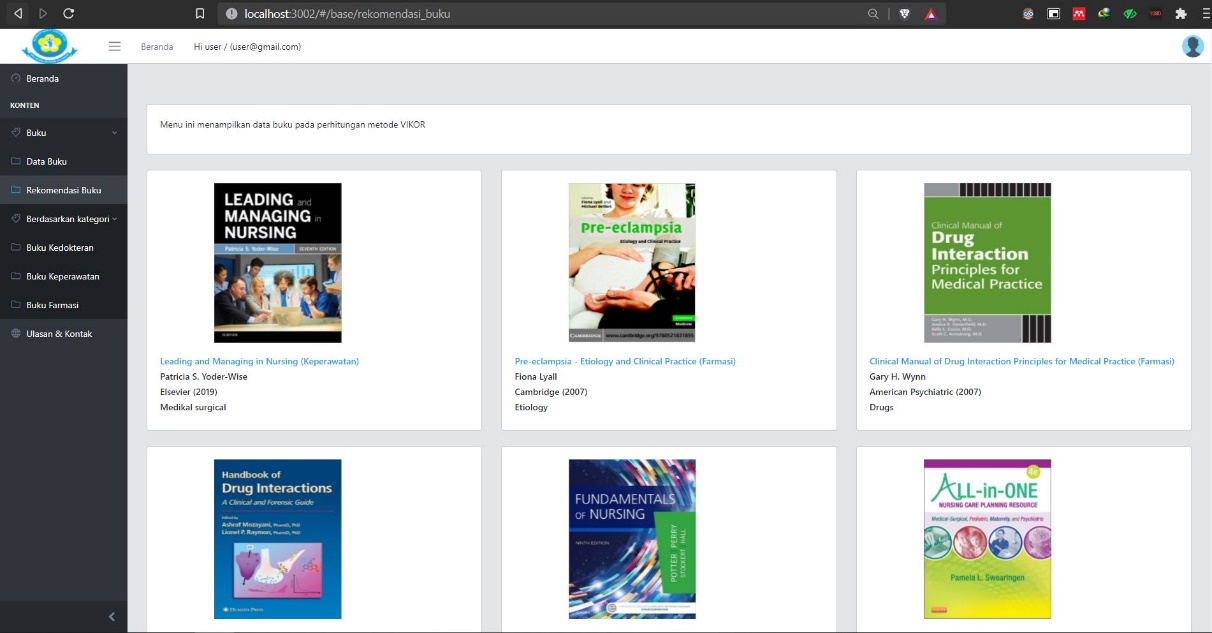
Tampilan Buku ini menampilkan semua buku elektronik berada pada tabel buku di database *Digital\_Repository* dan pengguna dapat memilih buku mana yang ingin dibaca.



### Gambar 5.3.2.1. Gambar Tampilan Buku

1. Tampilan Rekomendasi Buku

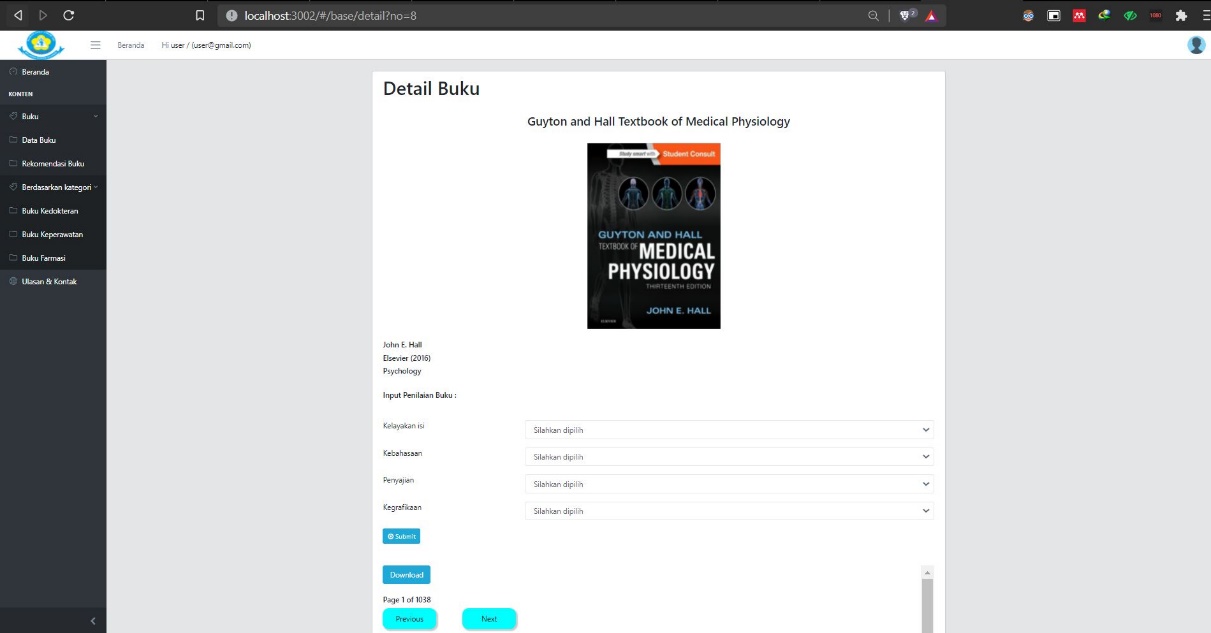
Tampilan Rekomendasi Buku menampilkan hasil rekomendasi berdasarkan ranking dengan menggunakan metode VIKOR sehingga pengguna dapat mengetahui buku mana yang direkomendasikan untuk dibaca.



### Gambar 5.3.2.2. Gambar Rekomendasi Buku

1. Tampilan Detail Buku

Tampilan Detail Buku menampilkan isi buku (judul, penulis, penerbit, tahun terbit, dan isi pdf) dan terdapat form penilaian berdasarkan kriteria penilaian buku yang dimana penilaian dijadikan perhitungan dengan metode VIKOR.



### Gambar 5.3.2.3. Gambar Tampilan Detail Buku

## **Pengujian Fungsionalitas sistem**

Pengujian fungsionalitas sistem dilakukan dengan cara menjalankan tiap fitur dalam aplikasi dan melihat kesesuaian hasil yang terjadi dengan hasil yang diharapkan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *Test Case* | Hasil yang diharapkan | Hasil yang didapatkan | keterangan |
| 1 | Mengolah user | Admin dapat mengolah data user seperti mengedit dan menghapus data user | Admin dapat melihat, edit, dan hapus user | Berhasil |
| 2 | Mengelola data kategori | Admin mengolah kategori (tenaga kesehatan) dan jenis buku dalam bentuk CRUD | Admin dapat melihat, tambah, edit, dan hapus kategori (kedokteran, keperawatan, dan farmasi) dan jenis buku | Berhasil |
| 3 | Mengelola data buku | Admin dapat menginput buku dan dapat mengunggah gambar dan file pdf ke sistem | Admin dapat melihat, tambah, edit, dan hapus data buku | Berhasil |
| 4 | Mengelola data ulasan | Admin dapat memonitoring ulasan yang diberikan user mengenai website digital repository | Admin dapat melihat dan hapus data ulasan | Berhasil |
| 5 | Mengelola data kriteria buku | Admin dapat mengolah data kriteria (bobot dan tipe kriteria) menjadi tolak ukur dalam perhitungan metode VIKOR | Admin dapat melihat, edit, dan hapus data kriteria buku | Berhasil |
| 6 | Mengelola data nilai buku | Admin dapat menginput penilaian berdasarkan kriteria penilaian | Admin dapat melihat, tambah, edit, dan hapus data nilai buku | Berhasil |
| 7 | Melihat data buku | User melihat daftar buku-buku yang ingin dibaca | User dapat melihat data buku (data buku berdasarkan tahun terbit, kategori, rekomendasi, dan urutan judul buku) | Berhasil |
| 8 | Mengakses isi buku | User dapat melihat isi buku dan isi pdf dari buku tersebut | User dapat melihat isi buku (judul, penulis, penerbit, dan tahun terbit), melihat isi pdf dan mengunduh buku | Berhasil |
| 9 | Mengisi ulasan buku | User dapat mengisi komentar mengenai website digital repository | User dapat input ulasan website | Berhasil |
| 10 | Menginput nilai buku | User dapat menginput penilaian buku untuk dijadikan tolak ukur dari rekomendasi dengan metode VIKOR | User dapat input nilai buku berdasarkan kriteria penilaian buku (kelayakan isi, kebahasaan, penyajian, dan kegrafikaan) | Berhasil |

# **BAB VI. HASIL DAN PEMBAHASAN**

Pembahasan ini mengenai argumentasi rasional dari penulis yang disusun secara sistematis berdasarkan fakta ilmiah yang diperoleh dari hasil pengujian. Pembahasan hasil pengujian dapat disajikan dalam bentuk uraian teoritik, baik secara kualitatif maupun kuantitatif

## **Hasil Pengujian Metode VIKOR**

Pada Pengujian Metode VIKOR terdapat tiga pengguna yang menguji aplikasi ini. Masing-masing matriks keputusan dari ketiga user tersebut yaitu:

### Tabel 6.1.1. Tabel Matriks Keputusan User Pertama

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matriks Keputusan User Pertama | | | | |
| Tabel Penilaian VIKOR | | | | |
| Alternatif | C1 | C2 | C3 | C4 |
| B1 | 6 | 5 | 7 | 8 |
| B2 | 9 | 6 | 8 | 5 |
| B3 | 4 | 4 | 8 | 6 |
| B4 | 5 | 7 | 6 | 6 |
| B7 | 5 | 4 | 6 | 4 |
| B9 | 3 | 4 | 2 | 8 |
| B15 | 3 | 4 | 7 | 10 |
| B33 | 5 | 7 | 2 | 3 |

### Tabel 6.1.2. Tabel Matriks Keputusan User Kedua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matriks Keputusan User Kedua | | | | |
| Tabel Penilaian VIKOR | | | | |
| Alternatif | C1 | C2 | C3 | C4 |
| B1 | 8 | 8 | 8 | 9 |
| B3 | 5 | 6 | 7 | 4 |
| B4 | 2 | 4 | 6 | 10 |
| B5 | 3 | 4 | 7 | 10 |
| B8 | 5 | 7 | 4 | 9 |
| B10 | 2 | 3 | 5 | 9 |
| B12 | 7 | 8 | 8 | 4 |
| B15 | 6 | 9 | 4 | 4 |

### Tabel 6.1.3. Tabel Matriks Keputusan User Ketiga

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matriks Keputusan User Ketiga | | | | |
| Tabel Penilaian VIKOR | | | | |
| Alternatif | C1 | C2 | C3 | C4 |
| B2 | 3 | 2 | 8 | 4 |
| B3 | 4 | 5 | 8 | 7 |
| B6 | 2 | 2 | 3 | 6 |
| B13 | 4 | 3 | 5 | 4 |
| B14 | 4 | 6 | 9 | 6 |
| B33 | 9 | 9 | 8 | 9 |
| B37 | 4 | 5 | 8 | 6 |
| B41 | 7 | 5 | 6 | 9 |

Langkah selanjutnya adalah menghitung rata-rata jika terdapat buku yang dinilai lebih dari satu user, matriks keputusan rata-rata yaitu:

### Tabel 6.1.4. Tabel Matriks Keputusan Rata-Rata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matriks Keputusan Rata-Rata | | | | |
| Tabel Penilaian VIKOR | | | | |
| Alternatif | C1 | C2 | C3 | C4 |
| B1 | 7 | 6,5 | 7,5 | 8,5 |
| B2 | 6 | 4 | 8 | 4,5 |
| B3 | 4,33 | 5 | 7,67 | 5,67 |
| B4 | 3,5 | 5,5 | 6 | 8 |
| B5 | 3 | 4 | 7 | 10 |
| B6 | 2 | 2 | 3 | 6 |
| B7 | 5 | 4 | 6 | 4 |
| B8 | 5 | 7 | 4 | 9 |
| B9 | 3 | 4 | 2 | 8 |
| B10 | 2 | 3 | 5 | 9 |
| B12 | 7 | 8 | 8 | 4 |
| B13 | 4 | 3 | 5 | 4 |
| B14 | 4 | 6 | 9 | 6 |
| B15 | 4,5 | 6,5 | 5,5 | 7 |
| B33 | 7 | 8 | 5 | 6 |
| B37 | 4 | 5 | 8 | 6 |
| B41 | 7 | 5 | 6 | 9 |

Langkah kedua adalah dengan menghitung nilai *Max* dan *Min* nilai alternatif dari masing-masing kriteria yaitu:

### Tabel 6.1.5. Tabel Penentuan Nilai Alternatif Masing-Masing Kriteria

|  |  |  |  |
| --- | --- | --- | --- |
| Penentuan Nilai Alternatif Masing-Masing Kriteria | | | |
| Kode | Nama Kriteria | Max | Min |
| C1 | Kelayakan isi | 7 | 2 |
| C2 | Kebahasaan | 8 | 2 |
| C3 | Penyajian | 9 | 2 |
| C4 | Kegrafikaan | 10 | 4 |

Langkah ketiga adalah dengan menghitung nilai normalisasi dari masing-masing kriteria bernilai *benefit*, matriks normalisasi yaitu:

### Tabel 6.1.6. Tabel Matriks Nornalisasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matriks Normalisasi | | | | |
| Alternatif | C1 | C2 | C3 | C4 |
| B1 | 0 | 0,25 | 0,21429 | 0,25 |
| B2 | 0,2 | 0,66667 | 0,14286 | 0,91667 |
| B3 | 0,534 | 0,5 | 0,19 | 0,72167 |
| B4 | 0,7 | 0,41667 | 0,42857 | 0,33333 |
| B5 | 0,8 | 0,66667 | 0,28571 | 0 |
| B6 | 1 | 1 | 0,85714 | 0,66667 |
| B7 | 0,4 | 0,66667 | 0,42857 | 1 |
| B8 | 0,4 | 0,16667 | 0,71429 | 0,16667 |
| B9 | 0,8 | 0,66667 | 1 | 0,33333 |
| B10 | 1 | 0,83333 | 0,57143 | 0,16667 |
| B12 | 0 | 0 | 0,14286 | 1 |
| B13 | 0,6 | 0,83333 | 0,57143 | 1 |
| B14 | 0,6 | 0,33333 | 0 | 0,66667 |
| B15 | 0,5 | 0,25 | 0,5 | 0,5 |
| B33 | 0 | 0 | 0,57143 | 0,66667 |
| B37 | 0,6 | 0,5 | 0,14286 | 0,66667 |
| B41 | 0 | 0,5 | 0,42857 | 0,16667 |

Dikarenakan masing-masing kriteria penilaian ini memiliki nilai bobot, langkah selanjutnya dengan menghitung nilai matriks normalisasi terbobot. Matriks normalisasi terbobot yaitu:

### Tabel 6.1.7. Tabel Matriks Normalisasi Terbobot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matriks Normalisasi Terbobot | | | | |
| Alternatif | C1 | C2 | C3 | C4 |
| B1 | 0 | 0,0625 | 0,06 | 0,0425 |
| B2 | 0,06 | 0,16667 | 0,04 | 0,15583 |
| B3 | 0,1602 | 0,125 | 0,0532 | 0,12268 |
| B4 | 0,21 | 0,10417 | 0,12 | 0,05667 |
| B5 | 0,24 | 0,16667 | 0,08 | 0 |
| B6 | 0,3 | 0,25 | 0,24 | 0,11333 |
| B7 | 0,12 | 0,16667 | 0,12 | 0,17 |
| B8 | 0,12 | 0,04167 | 0,2 | 0,02833 |
| B9 | 0,24 | 0,16667 | 0,28 | 0,05667 |
| B10 | 0,3 | 0,20833 | 0,16 | 0,02833 |
| B12 | 0 | 0 | 0,04 | 0,17 |
| B13 | 0,18 | 0,20833 | 0,16 | 0,17 |
| B14 | 0,18 | 0,08333 | 0 | 0,11333 |
| B15 | 0,15 | 0,0625 | 0,14 | 0,085 |
| B33 | 0 | 0 | 0,16 | 0,11333 |
| B37 | 0,18 | 0,125 | 0,04 | 0,11333 |
| B41 | 0 | 0,125 | 0,12 | 0,02833 |

Langkah selanjutnya adalah dengan menghitung nilai *Utility Measure (S)* dan *Regret Measure(R).* Tabel Perhitungan nilai *S* dan *R* yaitu:

### Tabel 6.1.8. Tabel Penentuan Nilai SI tiap Kriteria

|  |  |
| --- | --- |
| Penentuan Nilai SI tiap Kriteria | |
| Alternatif | Nilai |
| B1 | 0,165 |
| B2 | 0,4225 |
| B3 | 0,461083333 |
| B4 | 0,490833333 |
| B5 | 0,486666667 |
| B6 | 0,903333333 |
| B7 | 0,576666667 |
| B8 | 0,39 |
| B9 | 0,743333333 |
| B10 | 0,696666667 |
| B12 | 0,21 |
| B13 | 0,718333333 |
| B14 | 0,376666667 |
| B15 | 0,4375 |
| B33 | 0,273333333 |
| B37 | 0,458333333 |
| B41 | 0,273333333 |
| S+ | 0,903333333 |
| S- | 0,165 |

### Tabel 6.1.9. Tabel Penentuan Nilai Max Ri tiap Kriteria

|  |  |
| --- | --- |
| Penentuan Nilai Max Ri tiap kriteria | |
| Alternatif | Nilai |
| B1 | 0,0625 |
| B2 | 0,166666667 |
| B3 | 0,1602 |
| B4 | 0,21 |
| B5 | 0,24 |
| B6 | 0,3 |
| B7 | 0,17 |
| B8 | 0,2 |
| B9 | 0,28 |
| B10 | 0,3 |
| B12 | 0,17 |
| B13 | 0,208333333 |
| B14 | 0,18 |
| B15 | 0,15 |
| B33 | 0,16 |
| B37 | 0,18 |
| B41 | 0,125 |
| R+ | 0,3 |
| R- | 0,0625 |

Langkah terakhir adalah perhitungan ranking metode VIKOR yaitu:

* Q\_B1 = (0,5 *x* ((0,165 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,0625 – 0,0625) / (0,3 – 0,0625))) = 0
* Q\_B2 = (0,5 *x* ((0,4225 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,1667 – 0,0625) / (0,3 – 0,0625))) = 0,394
* Q\_B3 = (0,5 *x* ((0,461 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,1602 – 0,0625) / (0,3 – 0,0625))) = 0,406
* Q\_B4 = (0,5 *x* ((0,491 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,21 – 0,0625) / (0,3 – 0,0625))) = 0 ,531
* Q\_B5 = (0,5 *x* ((0,4867 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,24 – 0,0625) / (0,3 – 0,0625))) = 0,592
* Q\_B6 = (0,5 *x* ((0,903 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,3 – 0,0625) / (0,3 – 0,0625))) = 1
* Q\_B7 = (0,5 *x* ((0,5767– 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,17 – 0,0625) / (0,3 – 0,0625))) = 0,505
* Q\_B8 = (0,5 *x* ((0,39 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,2 – 0,0625) / (0,3 – 0,0625))) = 0,442
* Q\_B9 = (0,5 *x* ((0,743 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,28 – 0,0625) / (0,3 – 0,0625))) = 0,85
* Q\_B10 = (0,5 *x* ((0,6967 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,3 – 0,0625) / (0,3 – 0,0625))) = 0,86
* Q\_B12 = (0,5 *x* ((0,21 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,17 – 0,0625) / (0,3 – 0,0625))) = 0,26
* Q\_B13 = (0,5 *x* ((0,718 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,2083 – 0,0625) / (0,3 – 0,0625))) = 0,682
* Q\_B14 = (0,5 *x* ((0,3767 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,18 – 0,0625) / (0,3 – 0,0625))) = 0,39
* Q\_B15 = (0,5 *x* ((0,4375 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,15 – 0,0625) / (0,3 – 0,0625))) = 0,368
* Q\_B33 = (0,5 *x* ((0,273 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,16 – 0,0625) / (0,3 – 0,0625))) = 0,278
* Q\_B37 = (0,5 *x* ((0,4583 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,18 – 0,0625) / (0,3 – 0,0625))) = 0,446
* Q\_B41 = (0,5 *x* ((0,273 – 0,165) / (0,903 – 0,165))) + (1 – 0,5 ) *x* ((0,125 – 0,0625) / (0,3 – 0,0625))) = 0,204

Pengurutan perankingan ditentukan dari nilai yang paling rendah dengan solusi kompromi sebagai solusi ideal dilihat dari perankingan Qj dengan nilai terendah. Karena nilai Sj merupakan solusi yang diukur dari titik terjauh solusi ideal, sedangkan nilai Rj merupakan nilai solusi yang diukur dari titik terdekat solusi ideal

### Tabel 6.1.10. Tabel Perankingan Metode VIKOR

|  |  |  |
| --- | --- | --- |
| Menentukan Nilai Terakhir | | |
| Alternatif | Nilai | Ranking |
| B1 | 0 | 1 |
| B2 | 0,393677478 | 7 |
| B3 | 0,406192111 | 8 |
| B4 | 0,531180943 | 12 |
| B5 | 0,591517168 | 13 |
| B6 | 1 | 17 |
| B7 | 0,505096828 | 11 |
| B8 | 0,441843887 | 9 |
| B9 | 0,849542592 | 15 |
| B10 | 0,860045147 | 16 |
| B12 | 0,25678983 | 3 |
| B13 | 0,681735377 | 14 |
| B14 | 0,390709279 | 6 |
| B15 | 0,368747772 | 5 |
| B33 | 0,278626589 | 4 |
| B37 | 0,446014019 | 10 |
| B41 | 0,204942379 | 2 |

## **Hasil Pengujian Akurasi**

Akurasi didapat dengan membandingkan 17 dari 50 buku yang dipilih dan dinilai oleh user dan dipilih dari sistem dengan hasil uji manual (menggunakan microsoft excel). Hasil pengujian akurasi ditunjukan pada tabel:

### Tabel 6.2.1. Tabel Pengujian Akurasi

|  |  |  |
| --- | --- | --- |
| Kode Buku | Uji Manual | Uji Sistem |
| B1 | 0 | 0 |
| B2 | 0,393677478 | 0.3936774781196784 |
| B3 | 0,406192111 | 0.4061921112035168 |
| B4 | 0,531180943 | 0.5311809433289771 |
| B5 | 0,591517168 | 0.5915171676369253 |
| B6 | 1 | 1 |
| B7 | 0,505096828 | 0.505096827848402 |
| B8 | 0,441843887 | 0.4418438873707972 |
| B9 | 0,849542592 | 0.8495425923725793 |
| B10 | 0,860045147 | 0.8600451467268623 |
| B12 | 0,25678983 | 0.2567898301057384 |
| B13 | 0,681735377 | 0.6817353768167598 |
| B14 | 0,390709279 | 0.39070927884044193 |
| B15 | 0,368747772 | 0.3687477723654509 |
| B33 | 0,278626589 | 0.27862658904597837 |
| B37 | 0,446014019 | 0.44601401924676254 |
| B41 | 0,204942379 | 0.20494237851966257 |

Tahapan Perhitungan Uji Akurasi

Nilai Error = nilai uji sistem – nilai uji manual

Nilai Error % = (nilai error / nilai uji sistem) *x* 100%

Nilai Akurasi = 100 – nilai Error %

Hasil Perhitungan akurasi dapat dilihat pada tabel:

### Tabel 6.2.2. Tabel Perhitungan Pengujian Akurasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Buku | Uji Manual | Uji Sistem | Error | Error % |
| B1 | 0 | 0 | 0 | 0 |
| B2 | 0,393677478 | 0.3936774781196784 | 0 | 0 |
| B3 | 0,406192111 | 0.4061921112035168 | 0 | 0 |
| B4 | 0,531180943 | 0.5311809433289771 | 0 | 0 |
| B5 | 0,591517168 | 0.5915171676369253 | 0 | 0 |
| B6 | 1 | 1 | 0 | 0 |
| B7 | 0,505096828 | 0.505096827848402 | 0 | 0 |
| B8 | 0,441843887 | 0.4418438873707972 | 0 | 0 |
| B9 | 0,849542592 | 0.8495425923725793 | 0 | 0 |
| B10 | 0,860045147 | 0.8600451467268623 | 0 | 0 |
| B12 | 0,25678983 | 0.2567898301057384 | 0 | 0 |
| B13 | 0,681735377 | 0.6817353768167598 | 0 | 0 |
| B14 | 0,390709279 | 0.39070927884044193 | 0 | 0 |
| B15 | 0,368747772 | 0.3687477723654509 | 0 | 0 |
| B33 | 0,278626589 | 0.27862658904597837 | 0 | 0 |
| B37 | 0,446014019 | 0.44601401924676254 | 0 | 0 |
| B41 | 0,204942379 | 0.20494237851966257 | 0 | 0 |
| Total Error | | | | 0 |
| Rata-Rata Error | | | | 0 |
| Nilai Akurasi | | | | 100 % |

Berdasarkan tabel 6.2.2. dapat diamati bahwa perhitungan manual dengan hasil perhitungan sistem memiliki persamaan dengan nilai kesesuaian sebesar 100 %. Akurasi pada perhitungan ini dinyatakan sangat baik dan sangat akurat karena nilai hasil akurasi ≥ 95%

# **BAB VII. KESIMPULAN DAN SARAN**

## **Kesimpulan**

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut :

1. Sistem dapat merekomendasikan E-book sesuai dengan kriteria yang dihasilkan dari sistem dengan menggunakan metode *VIKOR* menentukan rekomendasi E-book sesuai dengan kriteria Penialaian buku (kelayakan isi, kebahasaan, penyajian, dan kegrafikaan).
2. Sistem dapat merekomendasikan E-book dengan menggunakan metode VIKOR dan dari hasil perhitungan manual (microsoft excel) dengan perhitungan sistem memiliki akurasi 100% persentase kesesuaian yang berarti tingkat akurasi dinyatakan sangat baik karena lebih dari 95% akurasi.
3. Rekomendasi E-book sesuai dengan kriteria penilaian buku (kelayakan isi, penyajian, kebahasaan, dan kegrafikan) dinyatakan telah berhasil dilakukan dengan menggunakan Python Framework Flask sebagai Backend, Javascript Framework ReactJS sebagai Frontend, dan Mysql sebagai database

## **Saran**

Saran yang dapat diberikan dari hasil penelitian untuk pengembangan sistem ini kedepan sebagai berikut :

1. Perbaikan tampilan pengguna dan tampilan admin yang lebih menarik.
2. Dapat dikembangkan dengan metode lain seperti metode *Borda* sebagai *Group Decision Support System* sehingga penilaian untuk merekomendasikan E-book dapat dilakukan oleh lebih dari satu pengguna tidak hanya dengan menghitung rata-rata.

# **DAFTAR PUSTAKA**

*Apa Itu JavaScript ? Pemahaman Dasar Bagi Pemula*. (n.d.). Retrieved November 25, 2020, from https://idwebhost.com/blog/apa-itu-javascript/

Aprida, C. D., & Febriliyan, S. (2013). Pembuatan Sistem Informasi Beasiswa Internal Direktorat Jenderal Perbendaharaan Menggunakan PHP dan MySQL. *Jurnal Teknik Pomits*, *2*(2), 350–354. http://ejurnal.its.ac.id/index.php/teknik/article/view/4822

Arisandi, A., & Pribadi, E. S. (2020). Analisa Metode VIKOR pada Rekomendasi Alat Musik Keyboard Electone Terbaik. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, *1*(1), 31–36. https://doi.org/10.30645/kesatria.v1i1.14

Arsanti, M. (2018). Pengembangan Bahan Ajar Mata Kuliah Penulisan Kreatif Bermuatan Nilai-Nilai Pendidikan Karakter Religius Bagi Mahasiswa Prodi Pbsi, Fkip, Unissula. *KREDO : Jurnal Ilmiah Bahasa Dan Sastra*, *1*(2), 71–90. https://doi.org/10.24176/kredo.v1i2.2107

Artikel, I. (2017). *Manajemen keuangan publik*.

Bako, S., & Lubis, N. (2018). *Sistem Pendukung Keputusan Pemilihan Kamar Hotel Terbaik Di Kota Medan Dengan Menggunakan Metode Vikor*. *5*(5), 525–531.

Firgiawan, F., wina, W., & Abdillah, G. (2018). *Sistem Pendukung Keputusan Rekomendasi Pemilihan Buku Komputer Di Amazon.Com Menggunakan Metode Technique for Order Preference By Similarity To Ideal Solution (Topis)*. *2006*, 255–260.

Hamria Hamria, H., Azwar, A., & Arja, K. (2020). Sistem Pendukung Keputusan Penilaian Kinerja Jasa Pramubakti Menggunakan Metode Moora. *Jurnal Ilmiah Informatika*, *8*(01), 25. https://doi.org/10.33884/jif.v8i1.1824

Hutapea Mentari Ananda; Karim, Abdul; Suginam, Suginam, B. J. H. (2018). Sistem Pendukung Keputusan Penentuan Jenis Kulit Terbaik Untuk Pembuatan Sepatu Dengan Menggunakan Metode VIKOR. *JURIKOM (Jurnal Riset Komputer)*, *5*(Vol 5, No 1 (2018): Februari 2018), 6–12. http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom/article/view/562

*Informasi lengkap dan buat janji di RSU Universitas Kristen Indonesia (UKI). Biaya, tindakan medis, daftar dokter, selengkapnya.* (n.d.). Retrieved November 24, 2020, from https://www.halodoc.com/rumah-sakit/nama/rsu-universitas-kristen-indonesia-uki

*JavaScript - Wikipedia*. (n.d.). Retrieved November 25, 2020, from https://en.wikipedia.org/wiki/JavaScript

Meningkatkan, P. Y. U., Konsep, P., Pada, P., Sd, S., Darul, I., & Semarang, K. (2020). *Pendidikan dasar*. *7*(1), 105–120.

*MySQL - Wikipedia*. (n.d.). Retrieved November 25, 2020, from https://en.wikipedia.org/wiki/MySQL

Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, *4*(1), 46–55. http://j-ptiik.ub.ac.id

Parlika, R., Ilham Pradika, S., Muhammad Hakim, A., & Rachman N.M., K. (2020). Bot Whatsapp Sebagai Pemberi Data Statistik Covid-19 Menggunakan Php, Flask, Dan Mysql. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, *1*(2 SE-Articles), 282–293. http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/101

*Pengertian MySQL, Fungsi, dan Cara Kerjanya (Lengkap)*. (n.d.). Retrieved November 25, 2020, from https://www.niagahoster.co.id/blog/mysql-adalah/#Pengertian\_MySQL

Pohan, H., & Sinaga, D. E. (2020). Penerapan Metode Moora Dalam Menentukan Parfume Terbaik Berdasarkan Kepribadian. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, *1*(2), 59–63. https://doi.org/10.30645/kesatria.v1i2.21

*Python (bahasa pemrograman) - Wikipedia bahasa Indonesia, ensiklopedia bebas*. (n.d.). Retrieved November 25, 2020, from https://id.wikipedia.org/wiki/Python\_(bahasa\_pemrograman)

Qurotianti, A. (2019). Optimalisasi Pemanfaatan Akses Digital Repository Dalam Menunjang Tri Dharma Perguruan Tinggi (Studi Kasus Di Perpustakaan Universitas Muhammadiyah Yogyakarta). *Pustabiblia: Journal of Library and Information Science*, *3*(2), 113–125. https://doi.org/10.18326/pustabiblia.v3i2.113-125

Radjatadoe, F. S., Sihotang, D. M., Boru, M., Komputer, J. I., & Cendana, U. N. (2017). *PEMBUATAN APLIKASI DIGITAL LIBRARY DI JURUSAN ILMU menyalurkan informasi secara cepat , tepat dan global . Dengan pesatnya perkembangan teknologi didapat oleh pemakai komputer yang terhubung ke internet . Website pada awalnya adalah ruang informasi dalam* . *5*(2), 34–41.

Safii, M., & Zulhamsyah, A. (2018). Sistem Pendukung Keputusan Pemilihan Mekanik Sepeda Motor Yamaha Alfascorfii Dengan Metode Multi Objective Optimization On The Basis Of Ratio Analysis (MOORA). *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *2*(2), 162. https://doi.org/10.30645/j-sakti.v2i2.79

Sharma, M. (2017). Multi attribute decision making techniques. *International Journal of Research in Management, Science & Technology*, *1*(1), 49–51. http://www.ijrmst.org/vol1\_no1.php

Simamora, B. (2017). Rancang Bangun Sistem Rekomendasi Televisi LED Dengan Metode Vikor Berbasis Web. *Jurnal ULTIMATICS*, *9*(1), 42–49. https://doi.org/10.31937/ti.v9i1.563

Sutrikanti, N., Situmorang, H., Fachrurrazi, Nurdiyanto, H., & Mesran, M. (2018). Implementasi Pendukung Keputusan Dalam Pemilihan Calon Peserta Cerdas Cermat Tingkat SMA Menerapkan Metode VIKOR. *JURIKOM (Jurnal Riset Komputer)*, *5*(2407-389X), 109–113.

Tumanggor, H., Haloho, M., Ramadhani, P., & Darma Nasution, S. (2018). Penerapan Metode VIKOR Dalam Penentuan Penerima Dana Bantuan Rumah Tidak Layak Huni. *Jurikom*, *5*(1), 71–78.

Ubaya, H., Siswati, S. D., Afriansyah, R., Studi, P., Komputer, T., Komputer, D., Komputer, F. I., Sriwijaya, U., Studi, P., Perangkat, R., Manufaktur, P., & Belitung, B. (1907). *Perancangan Monitoring Sensor berbasis Javascript dan Plotly*. 43–49.

*What is Python? Executive Summary | Python.org*. (n.d.). Retrieved November 25, 2020, from https://www.python.org/doc/essays/blurb/

Wicaksana, Rudy, & Lukito. (2017). Mengukur Kesuksesan Layanan Digital Library Universitas Gadjah Mada (Ugm). *Prosiding SNATIF*, *4*(2), 189–196.

# **Lampiran**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Buku Kedokteran | | | | | |
| kode | nama buku | jenis buku | penerbit | tahun terbit | penulis |
| B5 | Bailey & Loves Essential Clinical Anatomy | anatomy | paperback | 2018 | John S.P. Lumley |
| B6 | Larsens Human Embryology | embryology | wiley blackwell | 2009 | Gary C. Schoenwolf |
| B7 | Atlas of Human Anatomy | anatomy | elsevier | 2018 | Frank H. Netter |
| B8 | Guyton and Hall Textbook of Medical Physiology | psychology | elsevier | 2016 | John E. Hall |
| B17 | Netters Concise Radiologic Anatomy | anatomy | elsevier | 2014 | Joel A. Vilensky |
| B18 | Lecture Notes\_ Radiology, 3rd Edition | radiology | wiley blackwell | 2010 | Pradip R Patel |
| B19 | Diagnostic Radiology V.2 | radiology | elsevier | 2015 | Andreas Adam |
| B20 | Diagnostic the Dog and Cat | anatomy | saunders company | 2000 | kevin kealy |
| B21 | PRIMER OF DIAGNOSTIC IMAGING, 4th Edition | diagnostic | elsevier | 2007 | Ralph Weissleder |
| B22 | Atlas of Anatomy Head, Neck and Neuroanatomy | anatomy | elsevier | 2017 | Friedrich Paulsen |
| B23 | Yoga Anatomy | anatomy | breath trust | 2007 | Leslie Kominoff |
| B50 | Review of Medical Physiology | psychology | McGraw Hill | 2019 | Kim. E. Berret |

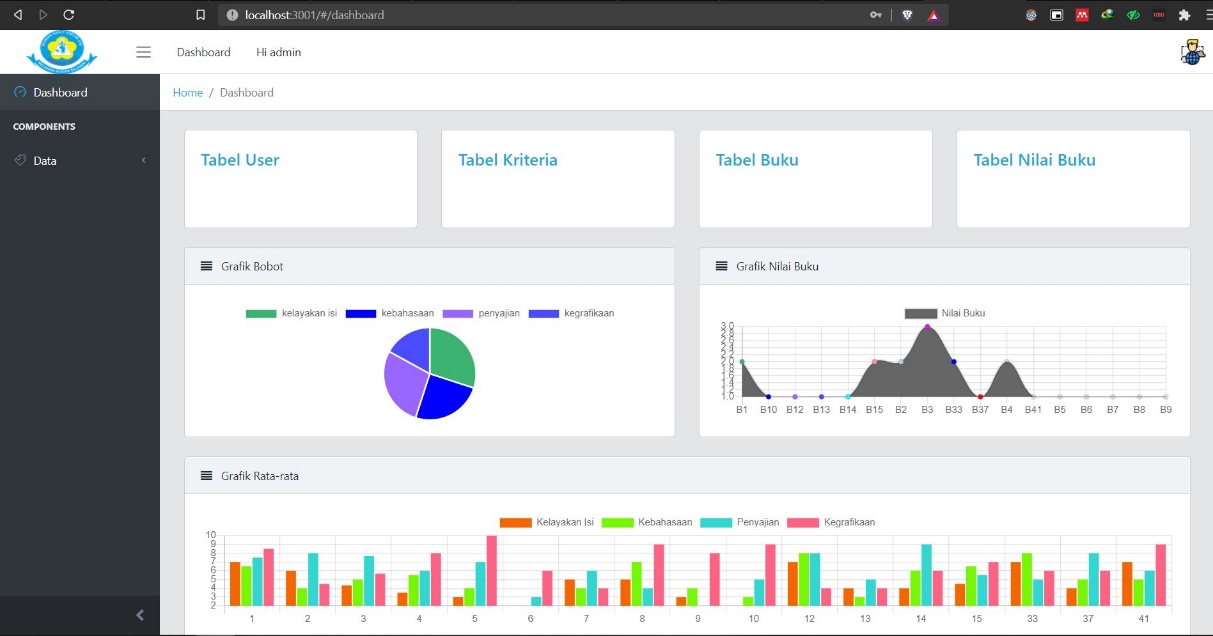
Lampiran 1: Data Buku

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Buku Keperawatan | | | | | |
| kode | nama buku | jenis buku | penerbit | tahun terbit | penulis |
| B1 | Leading and Managing in Nursing | medikal surgikal | elsevier | 2019 | Patricia S. Yoder-Wise |
| B2 | Assessment and Management of Clinical Problems | medikal surgikal | elsevier | 2020 | Harding Kwong Roberts |
| B3 | Nursing Interventions and Collaborative Management | medikal surgikal | elsevier | 2016 | Marianne Saunorus Baird |
| B4 | Netter’s Pediatrics | pediatric | elsevier | 2011 | Todd A. Florin |
| B13 | USMLE Step 2 CK Lecture Notes 2019 Pediatrics | pediatric | kaplan medical | 2019 | William G |
| B14 | All-in-One Nursing Care Planning Resource Medical-Surgical | medikal surgikal | elsevier | 2016 | pamela L. Swearigan |
| B15 | Fundamentals of Nursing | medikal surgikal | elsevier | 2017 | Patricia A. Potter |
| B16 | Mosby’s Pharmacology Memory Notecards Visual, Mnemonic, and Memory Aids for Nurses | pharmacology | elsevier | 2019 | JoAnn Zerwekh |

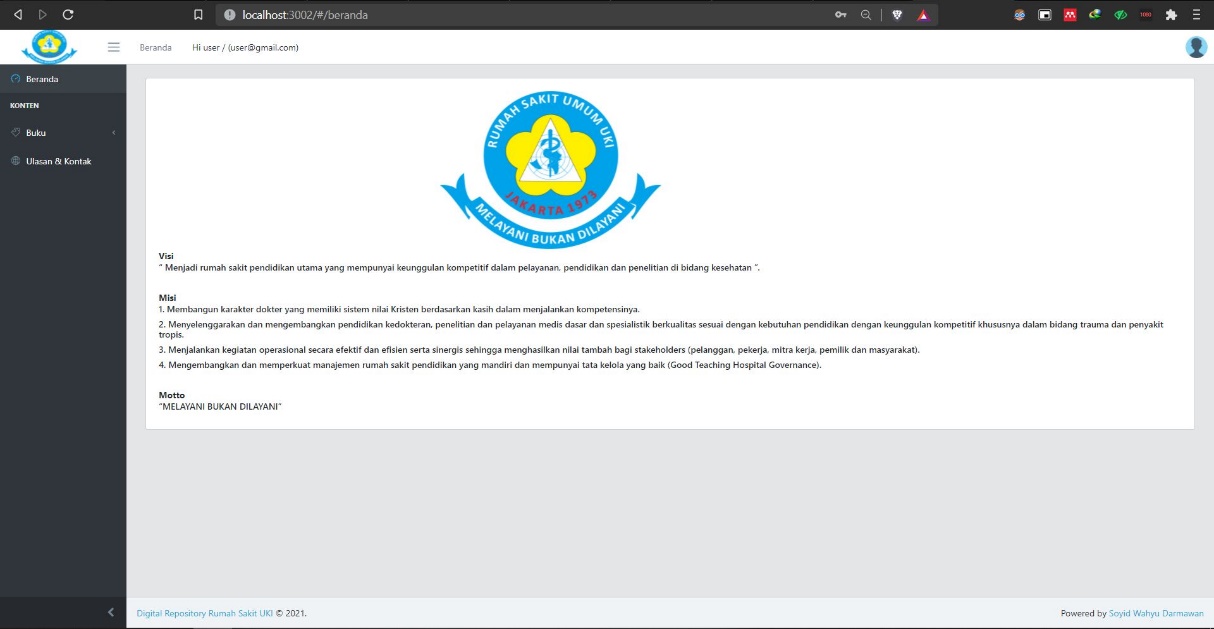
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Buku Farmasi | | | | | |
| kode | nama buku | jenis buku | penerbit | tahun terbit | penulis |
| B9 | Drugs And Human Location | drugs | elsevier | 2019 | Todd A. Florin |
| B10 | PRESCRIBING FOR ELDERLY PATIENTS | drugs | wiley-blackwell | 2009 | Stephen Jackson |
| B11 | Drug–Drug Interaction Primer | prescribing | american psychiatric | 2007 | Neil B. Sandsom |
| B12 | Clinical Manual of Drug Interaction Principles | drugs | american psychiatric | 2007 | Gary H. Wynn |
| B24 | A Manual of Adverse Drug Interactions | drugs | elsevier | 2017 | J.P.Griffin |
| B25 | Adherence to Pediatric Medical Regimens | medical regimen | paperback | 2014 | Michael Rapoff |
| B26 | British National Formulary | formulary | BMJ Group | 2009 | Media GMBH |
| B27 | British National Formulary for Children | formulary | BMJ Group | 2009 | Media GMBH |
| B28 | Community Pharmacy Handbook | drugs | pharmaceutical press | 2008 | Jon Waterfield |
| B29 | Drug interactions in infectious disease 2nd ed | drugs | Humana Press | 2018 | Stephen Pesticelli |
| B30 | Drug-Drug Interactions 2nd ed | drugs | Informa Healthcare | 2008 | David Rodriguez |
| B31 | Drug-related problems in the elderly | drugs | Springer | 2009 | Patrik Midlov |
| B32 | Drugs and Drugs 2nd Ed | drugs | Springer | 2006 | Hamilton |
| B33 | Handbook of Drug Interactions - A Clinical and Forensic Guide | drugs | Humana Press | 2004 | Ashraf Mozayani |
| B34 | MedFacts - Pocket Guide of Drug Interaction | drugs | Humana Press | 2004 | George Bailie |
| B35 | Neonatal Formulary 4 - Drugs in Pregnancy and the First Year of Life - A Pharmacopoeia | formulary | BMJ Books | 2003 | Catherine Hall |
| B36 | Neonatal Formulary 5 - Drug Use in Pregnancy and the First Year of Life | formulary | wiley-blackwell | 2007 | Vicky Pittman |
| B37 | Paediatric Handbook 8th Ed | prescribing | wiley-blackwell | 2009 | Kate Thompson |
| B38 | Pharmacotherapies for the Treatment of Opioid Dependence | pharmacotheraphy | Informa Healthcare | 2009 | Richard Matick |
| B39 | Pharmacotherapy of Diabetes - New Developments | pharmacotheraphy | Springer | 2007 | Erik Mogensen |
| B40 | Pharmacotherapy of Obesity | pharmacotheraphy | Birkhauser | 2008 | Michael Parnham |
| B41 | Pre-eclampsia - Etiology and Clinical Practice | etiology | Cambridge | 2007 | Fiona Lyall |
| B42 | Prescribing in Diabetes | prescribing | Cambridge | 2008 | Jill Hill |
| B43 | Prescribing in Pregnancy 4th Ed | prescribing | wiley-blackwell | 2008 | Peter Rubin |
| B44 | Stockley's Drug Interactions 8th Ed | drugs | pharmaceutical press | 2008 | Karen Baxter |
| B45 | Stockley's Drug Interactions 2009 Pocket Companion | drugs | pharmaceutical press | 2009 | Karen Baxter |
| B46 | Veterinary Drug Handbook (Desk Edition) | drugs | Lowa University | 1999 | Donald Plumb |
| B47 | Davis’s Drug Guide for Nurses | drugs | Davis Company | 2019 | Hazard Vallerand |
| B48 | Principles of Pharmacology The Pathophysiologic Basis of Drug Therapy | pharmacotheraphy | Wolters Kluwer | 2017 | David Golan |
| B49 | The Addicted Brain Why We Abuse Drugs, Alcohol, and Nicotine (FT Press Science) | drugs | elsevier | 2011 | Michael Kuhar |

Lampiran 2 Menu Utama Aplikasi

Admin :

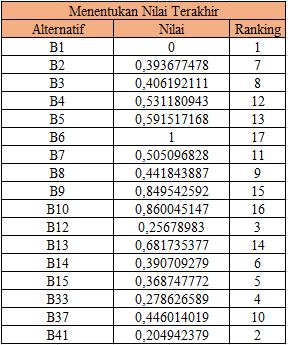


User :

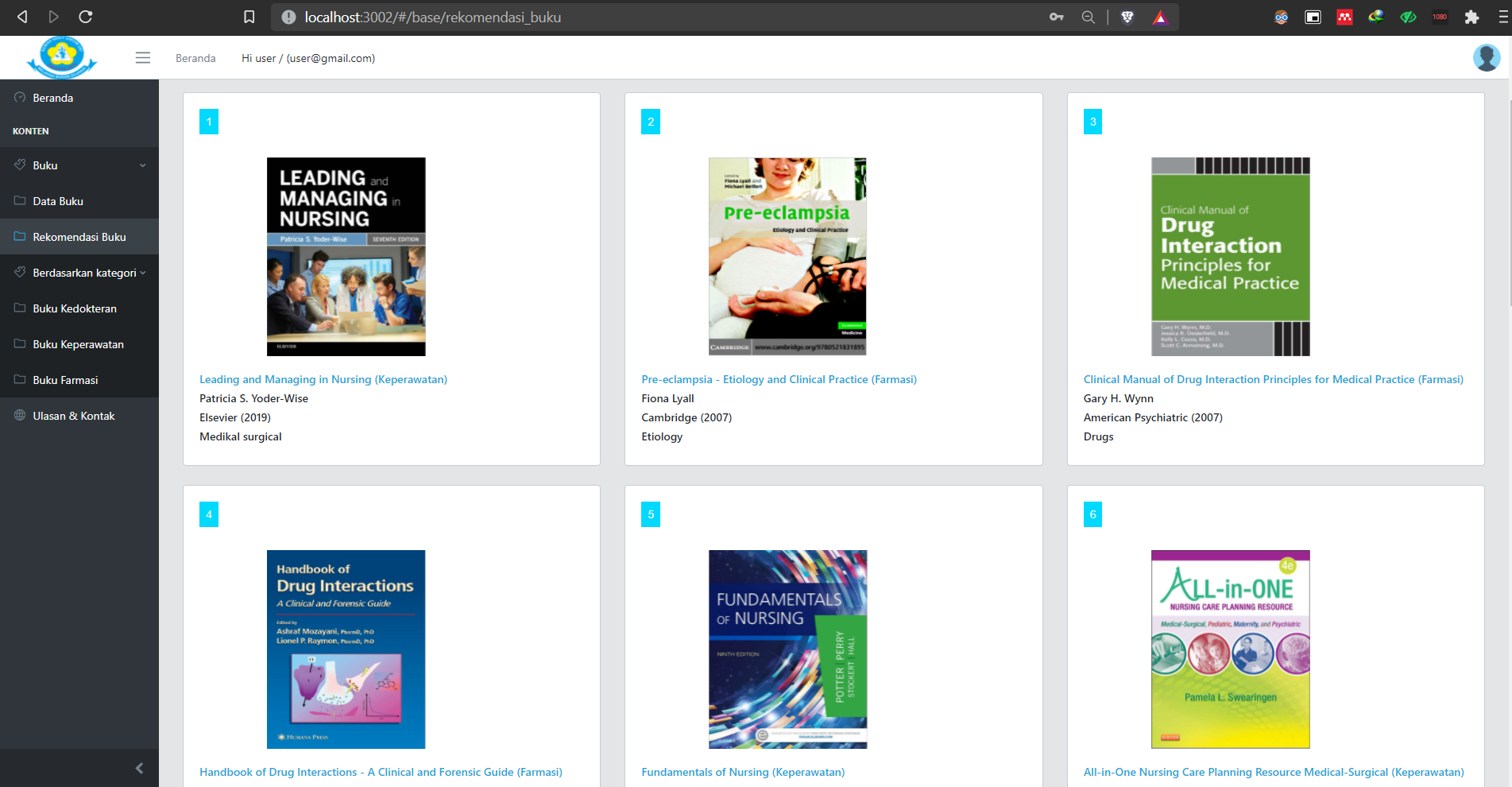


Lampiran 3 Keluaran Sistem

Keluaran sistem pada perhitungan manual menggunakan Microsoft Excel :

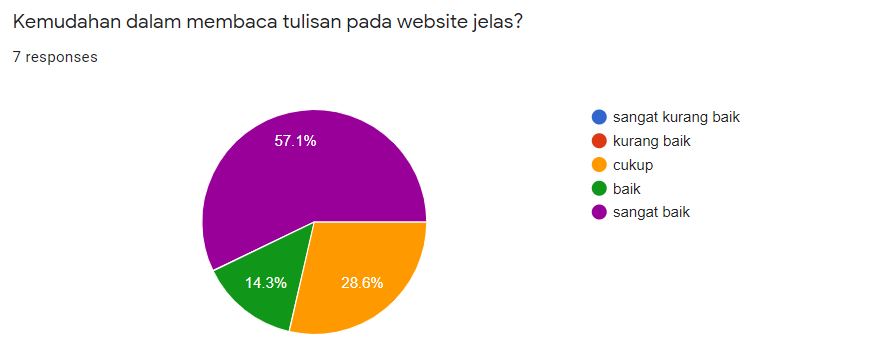


Keluaran sistem pada website user :

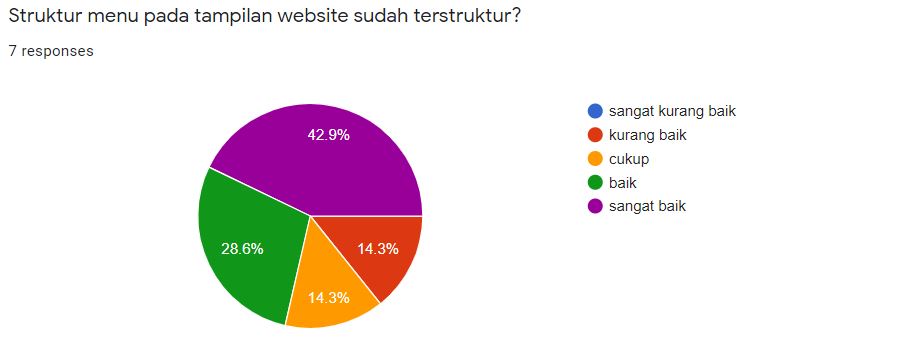


Lampiran 4 Kuesioner Uji Pengguna

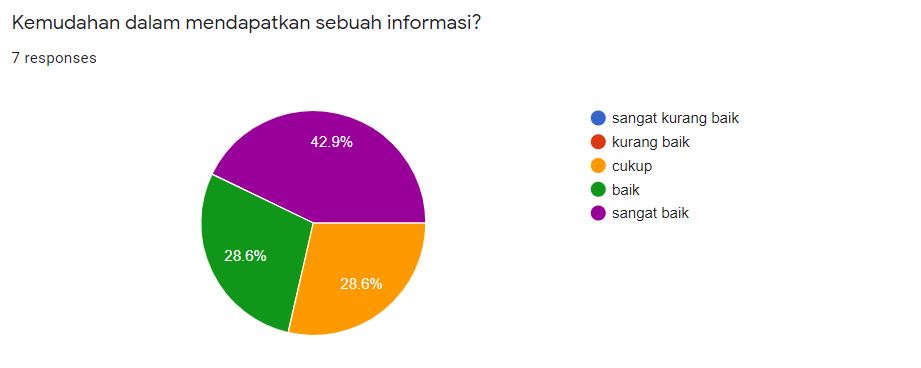














Lampiran 5 *Listing Program*

**Backend**

App.py

from flask import Flask, render\_template

from flask\_mysqldb import MySQL

from flask\_bcrypt import Bcrypt

from flask\_jwt\_extended import JWTManager

from flask\_cors import CORS

db = MySQL()

bcrypt = Bcrypt()

jwt = JWTManager()

# app = Flask()

app = Flask(\_\_name\_\_)

def create\_app():

cors = CORS(app, resources=r'/\*')

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = ''

app.config['MYSQL\_DB'] = 'digital\_repository'

app.config['MYSQL\_CURSORCLASS'] = 'DictCursor'

app.config['MAX\_CONTENT\_LENGTH'] = 350 \* 1024 \* 1024

app.config["JWT\_SECRET\_KEY"] = 'secret'

db.init\_app(app)

bcrypt.init\_app(app)

jwt.init\_app(app)

return app

User.py

from flask import Flask, render\_template, Blueprint

from flask import jsonify, request

from flask\_jwt\_extended import create\_access\_token

from ..run import db, jwt, bcrypt

user = Blueprint('user', \_\_name\_\_)

CREATE / REGISTER

@user.route('/user/register', methods=['POST'])

def add\_user():

cur = db.connection.cursor()

nik = request.get\_json()['nik']

nama\_user = request.get\_json()['nama\_user']

username\_user = request.get\_json()['username\_user']

email\_user = request.get\_json()['email\_user']

password\_user = bcrypt.generate\_password\_hash(request.get\_json()['password\_user']).decode('utf-8')

alamat = request.get\_json()['alamat']

notelp = request.get\_json()['notelp']

cur.execute("INSERT INTO tbl\_user (id\_user,nik, nama\_user, username\_user,email\_user,password\_user, alamat, notelp) "

" SELECT MAX(id\_user)+1, "

"'" + str(nik) + "', "

"'" + str(nama\_user) + "', "

"'" + str(username\_user) + "', "

"'" + str(email\_user) + "', "

"'" + str(password\_user) + "', "

"'" + str(alamat) + "', "

"'" + str(notelp) + "' " +

"FROM tbl\_user")

db.connection.commit()

result = \

{

'nik': nik,

'nama\_user': nama\_user,

'username\_user': username\_user,

'email\_user': email\_user,

'password\_user': password\_user,

'alamat': alamat,

'notelp': notelp

}

return jsonify({"result": result})

# LOGIN

@user.route('/user/login', methods=['POST'])

def login():

cur = db.connection.cursor()

username\_user = request.get\_json()['username\_user']

password\_user = request.get\_json()['password\_user']

result = ""

cur.execute("SELECT \* FROM tbl\_user where username\_user = '" + str(username\_user) + "'")

rv = cur.fetchone()

if bcrypt.check\_password\_hash(rv['password\_user'], password\_user):

access\_token = create\_access\_token(

identity={'id\_user': rv['id\_user'],

'nik': rv['nik'],

'nama\_user': rv['nama\_user'],

'username\_user': rv['username\_user'],

'email\_user': rv['email\_user'],

'alamat': rv['alamat'],

'notelp': rv['notelp']})

result = jsonify({"result": access\_token})

else:

result = jsonify({"error": "Invalid username and password"})

return result

Buku.py

1. Proses tambah

Proses menambah buku digunakan untuk menambahkan kedalam database. Proses tersebut melakukan perintah *query* insert

# CREATE

@buku.route('/buku', methods=['POST'])

def add\_buku():

if request.method == 'POST':

if app.config['MAX\_CONTENT\_LENGTH']:

cur = db.connection.cursor()

kode\_buku = request.form['kode\_buku']

nama\_buku = request.form['nama\_buku']

id\_jenis = request.form['id\_jenis']

id\_nakes = request.form['id\_nakes']

id\_penerbit = request.form['id\_penerbit']

id\_penulis = request.form['id\_penulis']

gambar = save\_images(request.files['gambar'])

userfile = save\_userfile(request.files['userfile'])

cur.execute(

"INSERT INTO tbl\_buku (id\_buku,kode\_buku,nama\_buku,id\_jenis,id\_nakes,id\_penerbit, id\_penulis, gambar, userfile) "

" SELECT MAX(id\_buku)+1, "

"'" + str(kode\_buku) + "', "

"'" + str(nama\_buku) + "', "

"'" + str(id\_jenis) + "', "

"'" + str(id\_nakes) + "', "

"'" + str(id\_penerbit) + "', "

"'" + str(id\_penulis) + "', "

"'" + str(gambar) + "', "

"'" + str(userfile) + "' " +

"FROM tbl\_buku")

db.connection.commit()

result = \

{

'kode\_buku': kode\_buku,

'nama\_buku' : nama\_buku,

'id\_jenis' : id\_jenis,

'id\_nakes': id\_nakes,

'id\_penerbit' : id\_penerbit,

'id\_penulis' : id\_penulis,

'gambar' : gambar,

'userfile' : userfile

}

return jsonify({"result" : result})

1. Proses ubah

# UPDATE

@buku.route('/buku/<id\_buku>', methods=['PUT'])

def update\_buku(id\_buku):

if request.method == 'PUT':

if app.config['MAX\_CONTENT\_LENGTH']:

cur = db.connection.cursor()

kode\_buku = request.form['kode\_buku']

nama\_buku = request.form['nama\_buku']

id\_jenis = request.form['id\_jenis']

id\_nakes = request.form['id\_nakes']

id\_penerbit = request.form['id\_penerbit']

id\_penulis = request.form['id\_penulis']

# gambar

gambar = save\_images(request.files['gambar'])

# file

userfile = save\_userfile(request.files['userfile'])

cur.execute("UPDATE tbl\_buku SET kode\_buku = '" + str(kode\_buku) + "',"

+ "nama\_buku ='" + str(nama\_buku) + "',"

+ "id\_jenis ='" + str(id\_jenis) + "',"

+ "id\_nakes ='" + str(id\_nakes) + "',"

+ "id\_penerbit ='" + str(id\_penerbit) + "'," + "id\_penulis ='" + str(id\_penulis) + "',"

+ "gambar ='" + str(gambar) + "',"

+ "userfile ='" + str(userfile)

+ "'WHERE id\_buku=" + id\_buku)

db.connection.commit()

result = \

{

'id\_buku' : id\_buku,

'kode\_buku': kode\_buku,

'nama\_buku' : nama\_buku,

'id\_jenis' : id\_jenis,

'id\_nakes': id\_nakes,

'id\_penerbit' : id\_penerbit,

'id\_penulis' : id\_penulis,

'gambar' : gambar,

'userfile' : userfile

}

return jsonify({"result": result})

+ "id\_penulis ='" + str(id\_penulis) + "',"

+ "gambar ='" + str(gambar) + "',"

+ "userfile ='" + str(userfile)

+ "'WHERE id\_buku=" + id\_buku)

db.connection.commit()

result = \

{

'id\_buku' : id\_buku,

'kode\_buku': kode\_buku,

'nama\_buku' : nama\_buku,

'id\_jenis' : id\_jenis,

'id\_nakes': id\_nakes,

'id\_penerbit' : id\_penerbit,

'id\_penulis' : id\_penulis,

'gambar' : gambar,

'userfile' : userfile

}

return jsonify({"result": result})

1. Proses hapus

# DELETE

@buku.route('/buku/<id\_buku>', methods=['DELETE'])

def delete\_buku(id\_buku):

cur = db.connection.cursor()

response = cur.execute("DELETE FROM tbl\_buku where id\_buku=" + id\_buku)

db.connection.commit()

if response> 0:

result = {'message' : 'record deleted'}

else:

result = {'message' : 'no record found'}

return jsonify({"result": result})

HitungVikor.py

def hitung\_vikor(self, alternatif, kriteria, tipe\_kriteria, bobot\_kriteria, matriks\_keputusan):

nilai\_maksimum = []

nilai\_minimum = []

for i in range(len(kriteria)):

nilai\_maksimum.append(0)

nilai\_minimum.append(0)

for j in range(len(alternatif)):

if ((j == 0) or (nilai\_maksimum[i] < matriks\_keputusan[j][i])):

nilai\_maksimum[i] = matriks\_keputusan[j][i]

if ((j == 0) or (nilai\_minimum[i] > matriks\_keputusan[j][i])):

nilai\_minimum[i] = matriks\_keputusan[j][i]

matriks\_normalisasi = []

for i in range(len(alternatif)):

matriks\_normalisasi.append([])

for j in range(len(kriteria)):

matriks\_normalisasi[i].append(0)

if tipe\_kriteria[j] == "cost":

matriks\_normalisasi[i][j] = (matriks\_keputusan[i][j] - nilai\_minimum[j]) / (nilai\_maksimum[j] - nilai\_minimum[j])

elif tipe\_kriteria[j] == "benefit": # tipe\_kriteria[j] == "benefit":

matriks\_normalisasi[i][j] = (nilai\_maksimum[j] - matriks\_keputusan[i][j]) / (nilai\_maksimum[j] - nilai\_minimum[j])

(matriks\_keputusan[i][j] - nilai\_minimum[j]) / (nilai\_maksimum[j] - nilai\_minimum[j])

elif tipe\_kriteria[j] == "benefit": # tipe\_kriteria[j] == "benefit":

matriks\_normalisasi[i][j] = (nilai\_maksimum[j] - matriks\_keputusan[i][j]) / (nilai\_maksimum[j] - nilai\_minimum[j])

matriks\_normalisasi\_terbobot = []

for i in range(len(alternatif)):

matriks\_normalisasi\_terbobot.append([])

for j in range(len(kriteria)):

matriks\_normalisasi\_terbobot[i].append(0)

matriks\_normalisasi\_terbobot[i][j] = matriks\_normalisasi[i][j] \* bobot\_kriteria[j]

Si = []

Ri = []

for i in range(len(alternatif)):

Si.append(0)

Ri.append(0)

for j in range(len(kriteria)):

Si[i] = Si[i] + matriks\_normalisasi\_terbobot[i][j]

if ((j == 0) or (Ri[i] < matriks\_normalisasi\_terbobot[i][j])):

Ri[i] = matriks\_normalisasi\_terbobot[i][j]

Splus = max(Si)

Smin = min(Si)

Rplus = max(Ri)

Rmin = min(Ri)

V = 0.5

hasil\_akhir = []

for i in range(len(alternatif)):

hasil\_akhir.append(0)

hasil\_akhir[i] = (V \* ((Si[i] - Smin) / (Splus - Smin))) + ((1 - V) \* ((Ri[i] - Rmin) / (Rplus - Rmin)))

alternatif\_ranking = []

hasil\_akhir\_ranking = []

for i in range(len(alternatif)):

alternatif\_ranking.append(alternatif[i])

hasil\_akhir\_ranking.append(hasil\_akhir[i])

for i in range(len(alternatif)):

for j in range(len(alternatif)):

if j > i:

if (hasil\_akhir\_ranking[i] > hasil\_akhir\_ranking[j]):

tmp\_alternatif = alternatif\_ranking[i]

tmp\_hasil\_akhir = hasil\_akhir\_ranking[i]

alternatif\_ranking[i] = alternatif\_ranking[j]

hasil\_akhir\_ranking[i] = hasil\_akhir\_ranking[j]

alternatif\_ranking[j] = tmp\_alternatif

hasil\_akhir\_ranking[j] = tmp\_hasil\_akhir

return alternatif\_ranking

CobaVikorDB.py

@vikor.route('/buku\_vikor', methods=['GET'])

def get\_buku\_vikor\_data():

try:

ObjekVikor = CobaHitungVIKOR()

alternatif = ObjekVikor.data\_IdBuku()

kriteria = ObjekVikor.data\_Kriteria()

tipe\_kriteria = ObjekVikor.data\_TipeKriteria()

bobot\_kriteria = ObjekVikor.data\_Bobot()

matriks\_keputusan = ObjekVikor.data\_matriks\_keputusan()

arrayData = ObjekVikor.hitung\_vikor(alternatif, kriteria, tipe\_kriteria, bobot\_kriteria, matriks\_keputusan)

data = [str(int) for int in arrayData]

res = ",".join(data)

cur = db.connection.cursor()

cur.execute(

"SELECT id\_buku, kode\_buku, nama\_buku, jenis\_buku, tenaga\_kesehatan, nama\_penerbit, tahun\_terbit, nama\_penulis, gambar, userfile "

"FROM tbl\_buku as b "

"INNER JOIN tbl\_jenis as j "

"on b.id\_jenis = j.id\_jenis "

"INNER JOIN tbl\_nakes as n "

"on b.id\_nakes = n.id\_nakes "

"INNER JOIN tbl\_penerbit as p1 "

"on b.id\_penerbit = p1.id\_penerbit "

"INNER JOIN tbl\_penulis as p2 "

"on b.id\_penulis = p2.id\_penulis WHERE id\_buku in (" + res + ") ORDER BY FIELD(id\_buku," + res + ")")

db.connection.commit()

rv = cur.fetchall()

return jsonify(rv)

# return jsonify({'message': 'data pada '+ arrayData +' ada'})

except:

cur = db.connection.cursor()

cur.execute(

"SELECT id\_buku, kode\_buku, nama\_buku, jenis\_buku, tenaga\_kesehatan, nama\_penerbit, tahun\_terbit, nama\_penulis, gambar, userfile "

"FROM tbl\_buku as b "

"INNER JOIN tbl\_jenis as j "

"on b.id\_jenis = j.id\_jenis "

"INNER JOIN tbl\_nakes as n "

"on b.id\_nakes = n.id\_nakes "

"INNER JOIN tbl\_penerbit as p1 "

"on b.id\_penerbit = p1.id\_penerbit "

"INNER JOIN tbl\_penulis as p2 "

"on b.id\_penulis = p2.id\_penulis ")

db.connection.commit()

rv = cur.fetchall()

return jsonify(rv)

db.connection.commit()

rv = cur.fetchall()

return jsonify(rv)

**Frontend**

KriteriaFunction.js

import axios from 'axios'

export const getKriteria = () => {

return axios

.get('/kriteria', {

headers: { "Content-type": "application/json" }

})

.then(res => {

var tbl\_kriteria = []

Object.keys(res.data).forEach((key) => {

var val = res.data[key]

tbl\_kriteria.push([

val.id\_kriteria,

val.kode,

val.nama\_kriteria,

val.bobot,

val.tipe\_kriteria])

})

return tbl\_kriteria

})

}

export const addKriteria = newKriteria => {

return axios

.post(

'/kriteria', {

kode: newKriteria.kode,

nama\_kriteria: newKriteria.nama\_kriteria,

bobot: newKriteria.bobot,

tipe\_kriteria: newKriteria.tipe\_kriteria,

}, {

headers: { "Content-type": "application/json" }

})

.then((res) => {

console.log(res)

})

}

export const deleteKriteria = id\_kriteria => {

axios

.delete(

`/kriteria/${id\_kriteria}`, {

headers: { "Content-type": "application/json" }

})

.then((res) => {

console.log(res)

})

.catch((res) => {

console.log(res)

})

}

export const updateKriteria = (kode, nama\_kriteria, bobot, tipe\_kriteria, id\_kriteria) => {

return axios

.put(

`/kriteria/${id\_kriteria}`, {

kode: kode,

nama\_kriteria: nama\_kriteria,

bobot: bobot,

tipe\_kriteria: tipe\_kriteria,

id\_kriteria: id\_kriteria

}, {

headers: { "Content-type": "application/json" }

})

.then((res) => {

console.log(res)

})

}

BukuFunction.js

export const addBuku = newBuku => {

return axios

.post(

'/buku', {

kode\_buku: newBuku.kode\_buku,

nama\_buku: newBuku.nama\_buku,

id\_jenis: newBuku.id\_jenis,

id\_nakes: newBuku.id\_nakes,

id\_penerbit: newBuku.id\_penerbit,

id\_penulis: newBuku.id\_penulis,

gambar: newBuku.gambar,

userfile: newBuku.userfile

}, {

headers: {

"Content-Type": "multipart/form-data",

"Accept": "application/json",

"type": "formData"

}

})

.then((res) => {

console.log(res)

})

}

export const deleteBuku = id\_buku => {

axios

.delete(

`/buku/${id\_buku}`, {

headers: { "Content-type": "application/json" }

})

.then((res) => {

console.log(res)

})

.catch((res) => {

console.log(res)

})

}

export const updateBuku = (kode\_buku, nama\_buku, id\_jenis, id\_nakes, id\_penerbit, id\_penulis, gambar, userfile, id\_buku ) => {

return axios

.put(

`/buku/${id\_buku}`, {

kode\_buku: kode\_buku,

nama\_buku: nama\_buku,

id\_jenis: id\_jenis,

id\_nakes: id\_nakes,

id\_penerbit: id\_penerbit,

id\_penulis: id\_penulis,

gambar: gambar,

userfile: userfile,

id\_buku: id\_buku

}, {

headers: { "Content-type": "application/json" }

})

.then((res) => {

console.log(res)

})

}

Buku.js

import React, {Component} from "react";

import { Badge, Card, CardBody, CardHeader, Col, Pagination, PaginationItem, PaginationLink, Row, Table } from 'reactstrap';

import {Link, withRouter} from 'react-router-dom';

const buku = {

marginLeft: 85,

marginRight: 25,

textAlign:'center',

width: 200,

height: 250

}

const Book = (props) =>{

return(

<Col xs="12" xl="4">

<Card>

<CardBody>

<img className="center" src={process.env.PUBLIC\_URL + '/Images/' + props.gambar } style={buku} alt="buku\_menu"></img>

<br/>

<br/>

<h6><Link to={props.LinkDetail}>{props.nama\_buku} ({props.tenaga\_kesehatan})</Link></h6>

<h6>{props.nama\_penulis}</h6>

<h6>{props.nama\_penerbit} ({props.tahun\_terbit})</h6>

<h6>{props.jenis\_buku}</h6>

</CardBody>

</Card>

</Col>

)

}

export default Book

BukuTable.js

import React, { Component } from 'react';

import { Badge, Card, CardBody, CardHeader, Col, Pagination, PaginationItem, PaginationLink, Row, Table } from 'reactstrap';

import Book from "./Buku";

class BukuTables extends Component {

state = {

ListBuku:[]

}

ambilDataDariServerAPI = () =>{

fetch('http://127.0.0.1:5000/buku')

.then(response => response.json())

.then(jsonHasilAmbilDariAPI => {

this.setState({

ListBuku: jsonHasilAmbilDariAPI

})

})

}

ambilDataTahunTerbaru = () => {

fetch('http://127.0.0.1:5000/tahun\_ascending')

.then(response => response.json())

.then(jsonHasilAmbilDariAPI => {

this.setState({

ListBuku: jsonHasilAmbilDariAPI

})

})

}

ambilDataTahunTerlama = () => {

fetch('http://127.0.0.1:5000/tahun\_descending')

.then(response => response.json())

.then(jsonHasilAmbilDariAPI => {

this.setState({

ListBuku: jsonHasilAmbilDariAPI

})

})

}

ambilDataBukuAscending = () => {

fetch('http://127.0.0.1:5000/buku\_ascending')

.then(response => response.json())

.then(jsonHasilAmbilDariAPI => {

this.setState({

ListBuku: jsonHasilAmbilDariAPI

})

})

}

ambilDataBukuDescending = () => {

fetch('http://127.0.0.1:5000/buku\_descending')

.then(response => response.json())

.then(jsonHasilAmbilDariAPI => {

this.setState({

ListBuku: jsonHasilAmbilDariAPI

})

})

}

handleChange(filter){

if(filter == "tahun\_ascending"){

this.ambilDataTahunTerbaru()

}

else if(filter == "tahun\_descending"){

this.ambilDataTahunTerlama()

}

else if(filter == "judul\_ascending"){

this.ambilDataBukuAscending()

}

else if(filter == "judul\_descending"){

this.ambilDataBukuDescending()

}

else{

this.ambilDataDariServerAPI()

}

}

componentDidMount() {

// this.getBukuMenu()

this.ambilDataDariServerAPI()

}

render() {

return (

<div className="animated fadeIn">

<br/>

<from>

<select onChange={(val) => this.handleChange(val.target.value)}>

<option value="0">Order By:</option>

<option value="Buku">Buku</option>

<option value="tahun\_ascending">Tahun Terbit Terlama</option>

<option value="tahun\_descending">Tahun Terbit Terbaru</option>

<option value="judul\_ascending">Judul Buku A - Z</option>

<option value="judul\_descending">Judul Buku Z - A</option>

</select>

</from>

<br/>

<br/>

<Card>

<CardBody>

<p>Menu ini menampilkan data keseluruhan semua buku yang ada pada database pada tabel buku</p>

</CardBody>

</Card>

<Row>

{

this.state.ListBuku.map(buku => {

return <Book key={buku.id\_buku}

id={buku.id\_buku}

gambar = {buku.gambar}

nama\_buku={buku.nama\_buku}

nama\_penulis={buku.nama\_penulis}

nama\_penerbit={buku.nama\_penerbit}

tahun\_terbit={buku.tahun\_terbit}

jenis\_buku={buku.jenis\_buku}

tenaga\_kesehatan = {buku.tenaga\_kesehatan}

LinkDetail={"detail?no="+buku.id\_buku}

/>

})

}

</Row>

</div>

);

}

}

export default BukuTables;

handleChange(filter){

if(filter == "tahun\_ascending"){

this.ambilDataTahunTerbaru()

}

else if(filter == "tahun\_descending"){

this.ambilDataTahunTerlama()

}

else if(filter == "judul\_ascending"){

this.ambilDataBukuAscending()

}

else if(filter == "judul\_descending"){

this.ambilDataBukuDescending()

}

else{

this.ambilDataDariServerAPI()

}

}

componentDidMount() {

// this.getBukuMenu()

this.ambilDataDariServerAPI()

}

render() {

return (

<div className="animated fadeIn">

<br/>

<from>

<select onChange={(val) => this.handleChange(val.target.value)}>

<option value="0">Order By:</option>

<option value="Buku">Buku</option>

<option value="tahun\_ascending">Tahun Terbit Terlama</option>

<option value="tahun\_descending">Tahun Terbit Terbaru</option>

<option value="judul\_ascending">Judul Buku A - Z</option>

<option value="judul\_descending">Judul Buku Z - A</option>

</select>

</from>

<br/>

<br/>

<Card>

<CardBody>

<p>Menu ini menampilkan data keseluruhan semua buku yang ada pada database pada tabel buku</p>

</CardBody>

</Card>

<Row>

{

this.state.ListBuku.map(buku => {

return <Book key={buku.id\_buku}

id={buku.id\_buku}

gambar = {buku.gambar}

nama\_buku={buku.nama\_buku}

nama\_penulis={buku.nama\_penulis}

nama\_penerbit={buku.nama\_penerbit}

tahun\_terbit={buku.tahun\_terbit}

jenis\_buku={buku.jenis\_buku}

tenaga\_kesehatan = {buku.tenaga\_kesehatan}

LinkDetail={"detail?no="+buku.id\_buku}

/>

})

}

</Row>

</div>

);

}

}

export default BukuTables;

Rekomendasi.js

import React, {Component} from "react";

import { Badge, Card, CardBody, CardHeader, Col, Pagination, PaginationItem, PaginationLink, Row, Table } from 'reactstrap';

import {Link, withRouter} from 'react-router-dom';

const Book = (props) =>{

return(

<Col xs="12" xl="4">

<Card>

<CardBody>

<img className="center" src={process.env.PUBLIC\_URL + '/Images/' + props.gambar } style={buku} alt="buku\_menu"></img>

<br/>

<br/>

<h6><Link to={props.LinkDetail}>{props.nama\_buku} ({props.tenaga\_kesehatan})</Link></h6>

<h6>{props.nama\_penulis}</h6>

<h6>{props.nama\_penerbit} ({props.tahun\_terbit})</h6>

<h6>{props.jenis\_buku}</h6>

</CardBody>

</Card>

</Col>

)

}

export default Book