

MODEL PEMBELAJARAN DAN LAPORAN AKHIR
PROJECT-BASED LEARNING
MATA KULIAH ALGORITMA DAN PEMROGRAMAN LANJUT
RANCANGAN KODE *SCRIPT* PEMROGRAMAN BERBASIS OOP PADA
STUDI KASUS: GRAPH TRAVERSAL
KELAS C



“IMPLEMENTASI ALGORITMA BFS (BREADTH-FIRST SEARCH)
DALAM PENCARIAN BUKU UNTUK SISTEM PERPUSTAKAAN
SMP NEGERI 1 MERAKURAK”

DISUSUN OLEH KELOMPOK “VI” :

- | | |
|------------------------|--------------------------|
| 1. REZA PUTRI ANGGA | (22083010006) - KETUA |
| 2. HANIN ANINDYA PUTRI | (22083010086) - ANGGOTA |
| 3. VIRA AMALIA ZAHRANI | (22083010098) - ANGGOTA |
| 4. KHOLIDATUS EMILIA | (22083010103) - ANGGOTA |

DOSEN PENGAMPU:

TRESNA MAULANA FAHRUDIN, S.S.T., MT (199305012022031007)
KARTIKA MAULIDA HINDRAYANI, S.KOM, M.KOM (199209092022032009)

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAWA TIMUR
2023

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan semua Rahmat-Nya, sehingga penulis dapat menyelesaikan final project “Project-Based Learning” mata kuliah Algoritma dan Pemrograman Lanjut dengan studi kasus “Implementasi Algoritma BFS (Breadth-First Search) Dalam Pencarian Buku Untuk Sistem Perpustakaan SMP Negeri 1 Merakurak”. Terima kasih penulis ucapkan kepada Bapak Tresna Maulana Fahrudin, S.S.T., MT dan Ibu Kartika Maulida Hindrayani, S.KOM, M.KOM selaku dosen mata kuliah Algoritma dan Pemrograman Lanjut, dan teman-teman yang telah membantu penulis baik secara moral maupun materi sehingga tugas akhir “Project-Based Learning” ini dapat terselesaikan dengan tepat waktu.

Penyusunan laporan tugas akhir ini bertujuan untuk memenuhi penugasan mata kuliah “Algoritma dan Pemrograman Lanjut” di Universitas Pembangunan Nasional ‘Veteran’ Jawa Timur.

Penulis menyadari, bahwa dalam penyusunan laporan tugas akhir ini masih jauh dari kata sempurna baik dari segi penyusunan, bahasa, maupun penulisannya. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang dapat membangun laporan tugas akhir ini untuk menjadi acuan agar penulis bisa menjadi lebih baik di masa mendatang.

Semoga laporan tugas akhir ini dapat menambah wawasan bagi para pembaca dan dapat bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan.

Surabaya, 25 Mei 2023

Penulis

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
DAFTAR GAMBAR.....	5
DAFTAR TABEL.....	6
1. BAB I : PENDAHULUAN	7
1.1 Latar Belakang	7
1.2 Permasalahan	8
1.3 Tujuan	8
1.4 Manfaat	8
2. BAB II : TINJAUAN PUSTAKA	9
2.1 Teori Penunjang	9
2.1.1 Sistem Informasi	9
2.1.2 Perpustakaan	9
2.1.3 Pengelolaan Data Perpustakaan	9
2.1.4 Graph Transversal	9
2.1.5 Algoritma BFS (Breadth-First Search)	9
2.1.6 Graphical User Interface (GUI)	10
2.1.7 Tkinter	10
2.2 Penelitian Terkait	10
2.2.1 Sistem Peminjaman Dan Pengembalian Buku Perpustakaan Keliling Berbasis Web Menggunakan Algoritma String Matching Pada Suku Dinas Perpustakaan Dan Kearsipan Kota Administrasi Jakarta Utara (2021)	10
2.2.2 Algoritma Pencarian Untuk Studi Kasus Pencarian Buku (2022)	10
2.2.3 Sistem Rute Terpendek Pencarian Buku Di Perpustakaan Menggunakan Algoritme Dijkstra (2021).....	10
3. BAB III : METODOLOGI PENELITIAN.....	11
3.1 Class Diagram	11
3.1.1 Anggota Perpustakaan	11
3.1.2 Petugas Perpustakaan	12
3.1.3 Buku	13
3.2 Flowchart Desain Sistem Tampilan	13
4. BAB IV : HASIL DAN PEMBAHASAN.....	16
4.1 Kode Script Perancangan Sistem	16
4.1.1 Kode Script Backend (Struktur Dasar)	16

4.1.1.1	Kode Script Structs Yang Berisi OOP (Object-Oriented Programming) ...	17
4.1.1.1.1	Kode Script Class Pengguna	17
4.1.1.1.2	Kode Script Class Buku	18
4.1.1.2	Kode Script Controller Yang Berisi Struktur Dasar Buku Dan Pengguna.	18
4.1.1.2.1	Kode Script Pengelolaan Pengguna	18
4.1.1.2.2	Kode Script Pengelolaan Buku	19
4.1.1.2.3	Kode Script BFS (Breadth-First Search) Dalam Pencarian Buku.....	19
4.1.2	Kode Script Generator Untuk Menghasilkan Dataset Pengguna	20
4.1.3	Kode Script Pages Untuk Menampilkan Tampilan Di GUI	21
4.1.3.1	Kode Script Halaman Pencarian	21
4.1.3.2	Kode Script Halaman Edit Buku	21
4.1.3.3	Kode Script Halaman Tambah Buku	22
4.1.3.4	Kode Script Halaman Hapus Buku	22
4.1.4	Kode Script App Untuk Tampilan Sistem Utama.....	22
4.2	Penampilan GUI (Graphical User Interface) Untuk Pencarian Buku Dalam Sistem Perpustakaan	23
4.2.1	Halaman Login.....	23
4.2.2	Halaman Pencarian Buku.....	24
4.2.3	Halaman Edit Lokasi Buku	25
4.2.4	Halaman Tambah Stok Buku	26
4.2.5	Halaman Hapus Buku	27
4.3	Hasil Pencarian Buku Menggunakan Algoritma BFS (Breadth-First Search) Dalam Bentuk Tabel.....	28
5.	BAB V : KESIMPULAN	29
6.	DAFTAR PUSTAKA	30
7.	LAMPIRAN	31

DAFTAR GAMBAR

Gambar 3. 1 Class Diagram Proyek.....	11
Gambar 3. 2 Flowchart Desain Sistem Tampilan	14
Gambar 4. 1 Kode Script Pengguna.....	17
Gambar 4. 2 Kode Script Pegawai	17
Gambar 4. 3 Kode Script Anggota.....	17
Gambar 4. 4 Kode Script Buku	18
Gambar 4. 5 Kode Script Pengelolaan Pengguna	18
Gambar 4. 6 Kode Script Pengelolaan Buku	19
Gambar 4. 7 Kode Script BFS (Breadth-First Search)	19
Gambar 4. 8 Visualisasi BFS (Breadth-First Search)	20
Gambar 4. 9 Kode Script Generator	20
Gambar 4. 10 Kode Script Halaman Pencarian	21
Gambar 4. 11 Kode Script Halaman Edit Buku	21
Gambar 4. 12 Kode Script Halaman Tambah Buku	22
Gambar 4. 13 Kode Script Halaman Hapus Buku	22
Gambar 4. 14 Kode Script Tampilan Aplikasi Utama	22
Gambar 4. 15 Halaman Login.....	23
Gambar 4. 16 Halaman Pencarian	24
Gambar 4. 17 Penggunaan Halaman Pencarian	24
Gambar 4. 18 Halaman Edit Lokasi Buku	25
Gambar 4. 19 Penggunaan Halaman Edit Lokasi Buku	25
Gambar 4. 20 Halaman Stok Buku	26
Gambar 4. 21 Penggunaan Halaman Tambah Stok Buku.....	26
Gambar 4. 22 Halaman Hapus Buku	27
Gambar 4. 23 Halaman Hapus Buku	27

DAFTAR TABEL

Tabel 4.3. 1 Hasil Halaman Pencarian	28
Tabel 4.3. 2 Hasil Halaman Edit Buku.....	28
Tabel 4.3. 3 Hasil Halaman Tambah Buku.....	28
Tabel 4.3. 4 Hasil Halaman Hapus Buku.....	28

1. BAB I : PENDAHULUAN

1.1 Latar Belakang

Di era digitalisasi saat ini, sistem informasi menjadi salah satu aspek dominan dalam kehidupan. Dengan berkembangnya informasi yang bervariasi dan beragam di berbagai hal, di perlukan adanya sistem informasi sebagai suatu perkembangan teknologi yang dapat menyatakan informasi dan menyajikan hal yang di butuhkan manusia. Sistem informasi ini dapat di terapkan di berbagai aspek kehidupan, termasuk dalam aspek di dunia Pendidikan, salah satunya dalam sarana perpustakaan.

Perpustakaan merupakan salah satu lembaga penting dalam pendidikan yang menyediakan fasilitas ilmu pengetahuan dan informasi sebagai penyedia buku dan sumber pembelajaran lainnya. Bertujuan untuk memenuhi informasi, membantu proses belajar mengajar, mendukung kegiatan penelitian, dan pengembangan ilmu pengetahuan. Namun, seiring meningkatnya jumlah buku, mengakibatkan adanya tata kelola perpustakaan yang rumit dalam melakukan pencarian buku di suatu perpustakaan.

Hingga saat ini, masih banyak perpustakaan yang menerapkan sistem pencarian secara manual untuk mencari judul buku yang di inginkan oleh pengguna, perpustakaan di SMP Negeri 1 Merakurak Tuban atau yang sering di kenal dengan SPENSAMER contohnya. Anggota (siswa) harus mencari buku secara langsung dengan mengunjungi rak - rak yang terdapat di perpustakaan untuk memastikan bahwa buku yang mereka inginkan masih tersedia, tanpa adanya sistem perpustakaan yang memfasilitasi untuk melakukan pencarian buku yang tersedia. Hal ini mengakibatkan tidak adanya efisiensi waktu dan ketidakpastian ketersediaan buku.

Untuk mengatasi permasalahan tersebut, di perlukan adanya suatu sistem perpustakaan yang efektif dan efisien untuk memudahkan anggota (siswa) dan pengguna (petugas perpustakaan) dalam melakukan pencarian ketersediaan buku yang di inginkan dan mempermudah melakukan pembaruan mengenai buku yang tersedia.

Dengan melakukan penerapan atau implementasi dari algoritma BFS (Breadth-First Search) untuk proses pencarian buku yang tersedia dapat mempermudah melakukan pengecekan mengenai ketersediaan buku yang ada. Selain itu, dapat di buat pula sebuah sistem di perpustakaan, yang berisi mengenai pencarian buku dan pembaruan mengenai ketersediaan buku yang tersedia. Penerapan algoritma ini bertujuan untuk memudahkan anggota (siswa) untuk mengetahui informasi mengenai ketersediaan pencarian buku yang di inginkan dan memudahkan petugas untuk melakukan pencatatan pembaruan ketersediaan buku.

Dengan adanya sistem perpustakaan untuk melakukan pencarian buku ini khususnya dalam pemenuhan kebutuhan peminjaman, di harapkan adanya kemudahan yang bisa di dapatkan oleh siswa dan petugas, selain itu di harapkan adanya tata kelola perpustakaan yang lebih efisien, transparan, dan mudah di akses oleh siswa dan petugas melalui device yang telah di sediakan. Anggota (siswa)

dapat dengan mudah melakukan pencarian buku yang di inginkan dan petugas perpustakaan dapat melakukan pembaruan mengenai ketersediaan buku yang ada secara mudah, efektif, efisien, dan akurat melalui sistem yang ada.

1.2 Permasalahan

Berdasarkan uraian latar belakang di atas, dapat di rumuskan permasalahan yakni :

1. Bagaimana cara mengatasi permasalahan mengenai adanya tata kelola perpustakaan yang manual dalam melakukan pencarian dan pembaruan buku di SMP Negeri 1 Merakurak?
2. Bagaimana implementasi algoritma BFS (Breadth-First Search) dalam proses melakukan pencarian buku yang tersedia di perpustakaan SMP Negeri 1 Merakurak?

1.3 Tujuan

Adapun tujuan dari pembahasan proyek ini, yakni :

1. Untuk membuat dan mengembangkan sistem pencarian buku yang efektif dan efisien di perpustakaan SMP Negeri 1 Merakurak.
2. Untuk mencari jalur atau urutan yang sesuai dalam pencarian buku di perpustakaan menggunakan algoritma BFS (Breadth-First Search).

1.4 Manfaat

Adapun manfaat dari pembahasan proyek ini, yakni :

1. Untuk penulis, pembahasan proyek ini bertujuan untuk menambah wawasan dan pengetahuan mengenai pembuatan sistem peminjaman untuk pencarian buku menggunakan algoritma BFS (Breadth-First Search) di perpustakaan SMP Negeri 1 Merakurak.
2. Untuk SMP Negeri 1 Merakurak, pembahasan proyek ini bertujuan untuk mempermudah pengelolaan pada pencarian buku untuk sistem perpustakaan dan mempermudah melakukan pembaruan mengenai ketersediaan buku.
3. Untuk pembaca, pembahasan proyek ini bertujuan untuk memberikan gambaran umum mengenai sistem perpustakaan untuk pencarian dan pembaruan ketersediaan buku yang efektif dan efisien di perpustakaan SMP Negeri 1 Merakurak.

2. BAB II : TINJAUAN PUSTAKA

Pembahasan mengenai teori penunjang dan penelitian terkait mengenai implementasi algoritma BFS (Breadth-First Search) untuk sistem perpustakaan, akan di jelaskan secara rinci sebagai berikut.

2.1 Teori Penunjang

2.1.1 Sistem Informasi

Sistem Informasi adalah sekumpulan komponen yang saling terikat dan berinteraksi untuk mengumpulkan, mengelola, memproses, dan menyimpan informasi yang bermanfaat dalam proses pengambilan keputusan, pengendalian, dan pemberian informasi kepada pihak – pihak tertentu yang membutuhkan.

Terdiri atas komponen – komponen yang di sebut blok bangunan, yaitu komponen input, model, output, hardware, dan software. Semua komponen tersebut saling berinteraksi untuk membentuk suatu kesatuan dalam mencapai sasaran.

2.1.2 Perpustakaan

Perpustakaan adalah salah satu sarana yang di butuhkan dalam dunia perkembangan Pendidikan. Pendidikan tidak akan terlaksana dengan baik tanpa adanya sumber sarana belajar yang di perlukan dalam proses kegiatan belajar mengajar. Perpustakaan memiliki banyak peran, di antaranya yakni penyimpanan, pengumpulan, pengolahan, dan penyebaran buku dan informasi yang luas.

2.1.3 Pengelolaan Data Perpustakaan

Pengelolaan Data Perpustakaan adalah suatu proses pengumpulan, pengolahan, dan pengelolaan data yang berkaitan dengan perpustakaan. Berisi mengenai sistem informasi yang di gunakan untuk mengelola data mengenai buku di perpustakaan yang dapat di lakukan dengan menggunakan komputerisasi yang optimal, efektif, dan efisien.

2.1.4 Graph Traversal

Graph Traversal adalah proses mengunjungi setiap simpul secara sistematis. Memiliki dua jenis graph transversal, yakni BFS (Breadth-First Search) dan DFS (Depth-First Search). Graph traversal memiliki beberapa tujuan, yakni mencari simpul atau set simpul tertentu, mencari jalur terpendek antara dua simpul, dan mengidentifikasi kelompok atau suatu kumpulan dalam simpul.

2.1.5 Algoritma BFS (Breadth-First Search)

Algoritma BFS (Breadth-First Search) adalah algoritma pencarian melebar, yaitu dengan mengunjungi simpul-simpul yang bertetangga dengan simpul awal terlebih dahulu. Kemudian, simpul-simpul yang bertetangga dengan simpul-simpul tersebut dan seterusnya. BFS (Breadth-First Search) menghasilkan jalur terpendek untuk menemukan suatu tujuan dalam sekumpulan grafik. Menggunakan struktur data antrian untuk menyimpan simpul yang telah di kunjungi.

2.1.6 Graphical User Interface (GUI)

Graphical User Interface (GUI) adalah suatu tampilan antar muka yang memungkinkan pengguna berinteraksi dengan perangkat. Graphical User Interface (GUI) di pergunakan dalam berbagai aplikasi perangkat lunak, termasuk sistem informasi. Memiliki kelebihan dalam kemudahan penggunaan dan penyederhanaan tugas yang kompleks.

2.1.7 Tkinter

Tkinter adalah sebuah modul atau pustaka dalam Graphical User Interface (GUI) yang merupakan fungsi *built-in* atau bawaan python yang di pergunakan untuk membuat tampilan aplikasi. Memiliki beberapa widget dasar, yakni button, canvas, entry, message, dan text.

2.2 Penelitian Terkait

Terdapat beberapa penelitian terkait mengenai proyek implementasi algoritma BFS (Breadth-First Search) dalam pencarian buku untuk sistem perpustakaan SMP Negeri 1 Merakurak. Di antaranya, yakni :

2.2.1 Sistem Peminjaman Dan Pengembalian Buku Perpustakaan Keliling Berbasis Web Menggunakan Algoritma String Matching Pada Suku Dinas Perpustakaan Dan Kearsipan Kota Administrasi Jakarta Utara (2021)

Penelitian ini membahas mengenai perpustakaan keliling pada suku dinas perpustakaan dan kearsipan kota administrasi jakarta utara dengan menggunakan metode string matching. Pendekatan ini berfokus pada sistem peminjaman dan pengembalian buku perpustakaan untuk mengoptimalkan pelayanan di perpustakaan dengan menggunakan algoritma string matching. Di mana penelitian ini memiliki kemiripan berkaitan dengan proyek yang di bahas namun menggunakan metode algoritma yang berbeda.

2.2.2 Algoritma Pencarian Untuk Studi Kasus Pencarian Buku (2022)

Penelitian ini membahas mengenai pencarian buku dengan pembuatan aplikasi agar dapat mempercepat proses pencarian buku dengan menggunakan algoritma searching metode sequential searching. Pendekatan ini berfokus pengolahan data buku dan melakukan pencarian buku secara efisien.

Di mana penelitian ini memiliki kemiripan berkaitan dengan proyek yang di usulkan namun menggunakan metode algoritma yang berbeda.

2.2.3 Sistem Rute Terpendek Pencarian Buku Di Perpustakaan Menggunakan Algoritme Dijkstra (2021)

Penelitian ini membahas mengenai permasalahan dalam pencarian buku menggunakan rute terpendek di perpustakaan dengan sistem yang tertanam pada keranjang buku, pengunjung hanya perlu menentukan lokasi diri dan menentukan kategori buku yang ingin di cari menggunakan tombol navigasi pada sistem.

Di mana penelitian ini memiliki kemiripan berkaitan dengan proyek yang di usulkan namun menggunakan metode algoritma yang berbeda dan output pada aplikasi yang berbeda.

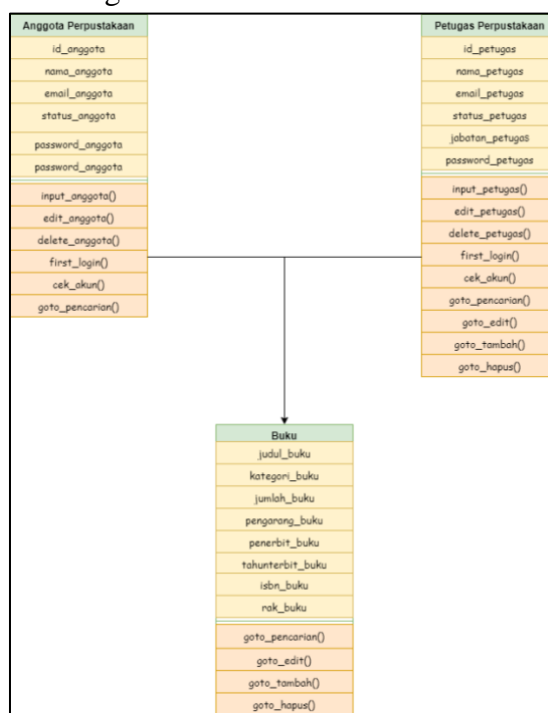
3. BAB III : METODOLOGI PENELITIAN

Sistem perpustakaan untuk pencarian buku di SMP Negeri 1 Merakurak yang akan di buat adalah tampilan antarmuka GUI tkinter yang dapat di akses oleh pengguna yang terdiri atas pegawai dan petugas perpustakaan menggunakan device yang tersedia di perpustakaan dengan melakukan proses komputerisasi. Dengan alur program pengguna di minta untuk memasukkan email dan password di halaman login dan akan di cek status pengguna, jika pengguna tercatat sebagai anggota maka hanya bisa mengakses halaman pencarian buku dan jika pengguna tercatat sebagai petugas maka hanya bisa mengakses halaman pencarian buku, edit buku, tambah stok buku, dan penghapusan buku.

Untuk pengimplementasian dalam class diagram dan flowchart sistem tampilan dapat di representasikan sebagai berikut.

3.1 Class Diagram

Pada class diagram ini di rancang untuk menjelaskan struktur mengenai sistem perpustakaan yang akan di buat. Terdapat 3 bagian / class/ entity utama, yakni pengguna yang di bedakan menjadi anggota perpustakaan dan petugas perpustakaan, dan buku. Di mana pada masing-masing tersebut memiliki atribut dan metode yang berbeda di dalam class diagram ini. Oleh karena itu, class diagram dapat di representasikan sebagai berikut.



Gambar 3. 1 Class Diagram Proyek

Di mana penjelasan pada masing-masing bagian dapat di jelaskan lebih lanjut sebagai berikut.

3.1.1 Anggota Perpustakaan

Pada bagian anggota perpustakaan memiliki beberapa atribut, yakni :

1. Id_anggota yang berisi mengenai identifikasi unik yang membedakan antara satu anggota perpustakaan dengan anggota perpustakaan yang lain.
2. Nama_anggota yang berisi mengenai nama lengkap anggota perpustakaan.
3. Email_anggota yang berisi mengenai alamat email anggota perpustakaan.
4. Status_anggota yang berisi mengenai status anggota perpustakaan.
5. Password_anggota yang berisi mengenai password anggota perpustakaan.

Di mana atribut-atribut tersebut akan di ringkas atau di lakukan pemrosesan lebih lanjut untuk menghasilkan data dalam file dataset pengguna yang memiliki beberapa atribut, yakni id_anggota, nama_anggota, email_anggota, status_anggota, dan password_anggota. Anggota juga memiliki beberapa metode, yakni :

1. Input_anggota yang di pergunakan untuk memasukkan identitas anggota.
2. Edit_anggota yang di pergunakan untuk mengedit identitas anggota.
3. Delete_anggota yang di pergunakan untuk menghapus identitas anggota.
4. First_login yang di pergunakan untuk memasukkan informasi mengenai identitas anggota untuk keperluan login.
5. Cek_akun yang di pergunakan untuk melakukan pengecekan apakah akun tersebut memiliki identitas yang sama dengan pengguna.
6. Goto_pencarian yang di pergunakan untuk mengakses halaman pencarian buku.

3.1.2 Petugas Perpustakaan

Pada bagian petugas perpustakaan memiliki beberapa atribut, yakni :

1. Id_anggota yang berisi mengenai identifikasi unik yang membedakan antara satu petugas perpustakaan dengan petugas perpustakaan yang lain.
2. Nama_anggota yang berisi mengenai nama lengkap petugas perpustakaan.
3. Email_anggota yang berisi mengenai alamat email petugas perpustakaan.
4. Status_anggota yang berisi mengenai status petugas perpustakaan.
5. Password_anggota yang berisi mengenai password petugas perpustakaan.

Di mana atribut-atribut tersebut akan di ringkas atau di lakukan pemrosesan lebih lanjut untuk menghasilkan data dalam file dataset pengguna yang memiliki beberapa atribut, yakni id_petugas, nama_petugas, email_petugas, status_petugas, dan password_petugas. Petugas juga memiliki beberapa metode, yakni :

1. Input_petugas yang di pergunakan untuk memasukkan identitas petugas.
2. Edit_petugas yang di pergunakan untuk mengedit identitas petugas.
3. Delete_petugas yang di pergunakan untuk menghapus identitas petugas.
4. First_login yang di pergunakan untuk memasukkan informasi mengenai identitas petugas untuk keperluan login.
5. Cek_akun yang di pergunakan untuk melakukan pengecekan apakah akun tersebut memiliki identitas yang sama dengan pengguna.
6. Goto_pencarian yang di pergunakan untuk mengakses halaman dan melakukan pencarian buku.
7. Goto_edit yang di pergunakan untuk mengakses halaman dan melakukan edit buku.
8. Goto_tambah yang di pergunakan untuk mengakses halaman dan menambah stok kesediaan buku.
9. Goto_hapus yang di pergunakan untuk mengakses halaman menghapus buku.

3.1.3 Buku

Pada bagian buku yang terdapat di perpustakaan memiliki beberapa atribut, yakni :

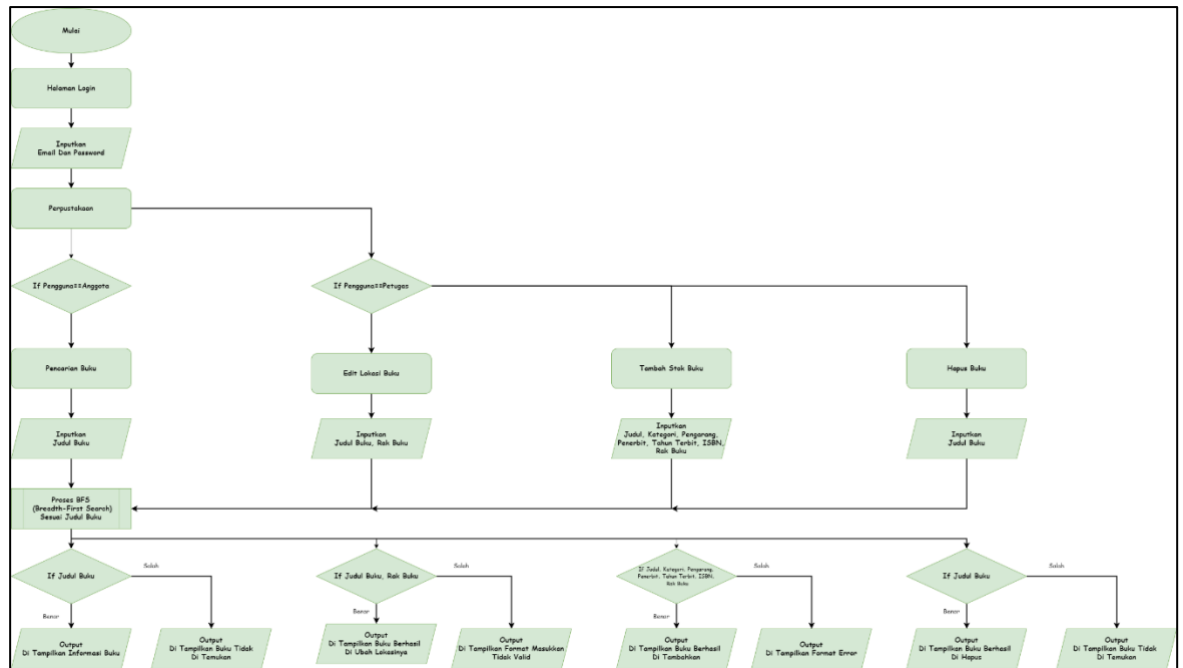
1. Judul_buku yang berisi mengenai judul buku.
2. Kategori buku yang berisi mengenai kategori buku.
3. Jumlah_buku yang berisi mengenai ketersediaan jumlah buku.
4. Pengarang_buku yang berisi mengenai pengarang atau penulis buku.
5. Penerbit_buku yang berisi mengenai penerbit buku.
6. Isbn_buku yang berisi mengenai isbn buku yang bersifat unik untuk membedakan buku satu dengan lainnya.
7. Rak_buku yang berisi mengenai informasi lokasi buku untuk memudahkan pencarian buku.

Di mana buku juga memiliki beberapa metode, yakni :

1. Goto_pencarian yang di pergunakan untuk mengakses halaman dan melakukan pencarian buku.
2. Goto_edit yang di pergunakan untuk mengakses halaman dan melakukan edit buku.
3. Goto_tambah yang di pergunakan untuk mengakses halaman dan menambah stok kesediaan buku.
4. Goto_hapus yang di pergunakan untuk mengakses halaman menghapus buku.

3.2 Flowchart Desain Sistem Tampilan

Untuk flowchart desain sistem tampilan dalam proyek ini dapat di representasikan sebagai berikut.



Gambar 3. 2 Flowchart Desain Sistem Tampilan

Dalam desain sistem tampilan implementasi BFS (Breadth-First Search) dalam pencarian buku untuk sistem perpustakaan SMP Negeri 1 Merakurak dapat di jelaskan sebagai berikut. Desain sistem tampilan di buat mulai dari input-proses-output. Di mana awalnya pengguna akan masuk ke halaman login dan di perintahkan untuk memasukkan email dan password. Jika berhasil melakukan login, maka pengguna akan di bawa ke halaman perpustakaan.

Di mana terdapat dua kondisi jika pengguna tercatat sebagai anggota maka, pengguna tersebut hanya bisa mengakses bagian halaman pencarian. Selanjutnya, pengguna di minta untuk memasukkan judul buku yang di inginkan. Akan di lakukan proses BFS (Breadth-First Search) untuk melakukan pencarian buku sesuai judul. Terdapat 2 output, jika buku di temukan di cetak informasi mengenai buku dan jika tidak maka di cetak informasi bahwa buku tidak di temukan.

Dan jika pengguna tercatat sebagai petugas maka, pengguna tersebut bisa mengakses halaman pencarian buku dengan mekanisme pemrosesan sama seperti pengguna yang tercatat sebagai anggota, halaman edit lokasi buku di mana pengguna tersebut di minta untuk memasukkan judul buku dan rak buku yang di inginkan. Akan di lakukan proses BFS (Breadth-First Search) untuk melakukan pencarian buku sesuai judul untuk memindahkan buku tersebut ke rak yang di inginkan. Terdapat 2 output, jika buku berhasil di pindahkan di cetak informasi buku berhasil di ubah lokasinya dan jika tidak di cetak informasi format masukan tidak valid.

Halaman tambah buku di mana pengguna tersebut di minta untuk memasukkan informasi buku mengenai judul, kategori, pengarang, penerbit, tahun terbit, isbn, dan rak buku. Terdapat 2 output, jika buku berhasil di tambahkan di cetak informasi buku berhasil di tambahkan dan jika tidak di cetak informasi format error. Dan yang terakhir, terdapat halaman hapus buku di mana pengguna tersebut di minta untuk

memasukkan judul buku yang ingin di hapus. Akan di lakukan proses BFS (Breadth-First Search) untuk melakukan penghapusan buku sesuai judul. Terdapat 2 output, jika buku berhasil di hapus di cetak informasi buku berhasil di hapus dan jika tidak di cetak informasi buku tidak di temukan.

Dengan adanya rancangan desain sistem ini di harapkan mampu memudahkan pengguna yang terdiri atas anggota dan petugas dalam menjalankan sistem perpustakaan SMP Negeri 1 Merakurak dalam pencarian buku sesuai dengan ketentuan-ketentuan yang di tetapkan.

4. BAB IV : HASIL DAN PEMBAHASAN

Untuk mengimplementasikan algoritma BFS (Breadth-First Search) dalam pencarian buku untuk sistem perpustakaan di SMP Negeri 1 Merakurak, di perlukan beberapa tahapan untuk melakukan pembuatan perancangan sistem, yang di mulai dengan pembuatan kode script mengenai pembuatan folder backend (isi utama kode script) yang berisi mengenai structs untuk merepresentasikan pengguna, pegawai, anggota, dan buku yang menggunakan konsep OOP (Object-Oriented Programming) dan controller untuk pengelolaan buku dan pengelolaan pengguna.

Terdapat folder dataset yang berisi mengenai data-data buku yang tersedia di perpustakaan dan dataset pengguna yang berisi nama-nama anggota dan petugas di perpustakaan. Terdapat folder generator yang berisi mengenai langkah-langkah pembuatan data pengguna yang di buat secara acak. Terdapat folder img yang berisi mengenai foto atau gambar yang akan di tampilkan pada halaman login aplikasi. Terdapat pages yang berisi mengenai tampilan aplikasi yang akan di tampilkan, yakni tampilan pencarian buku, tambah stok buku, edit lokasi buku, dan hapus buku.

Di mana perlu di ketahui bahwa, jika terdaftar sebagai anggota maka hanya dapat menjalankan tampilan pencarian buku dan jika terdaftar sebagai petugas maka dapat melakukan pencarian buku, tambah stok buku, edit lokasi buku, dan hapus buku. Terdapat file app yang berisi mengenai implementasi tampilan aplikasi pencarian buku untuk sistem perpustakaan GUI yang menggunakan tkinter.

Terdapat pula beberapa library bawaan yang di pergunakan dalam proses ini, yakni library openpyxl untuk membaca file excell, library collections untuk proses bfs (breadth-first search), library tkinter dan customtkinter untuk menghasilkan tampilan, dan library random untuk membuat data random.

Untuk penjelasan kode script setiap langkahnya dapat di jelaskan secara detail sebagai berikut.

4.1 Kode Script Perancangan Sistem

Pada bagian ini akan di jelaskan mengenai perancangan sistem awal.

4.1.1 Kode Script Backend (Struktur Dasar)

Pada bagian ini akan di jelaskan mengenai struktur dasar pemrograman yang terdiri dari dua bagian, yakni structs yang berisi mengenai pengguna yang terdapat anggota dan pegawai (petugas) dan buku, serta controller yang berisi mengenai struktur dasar buku dan pengguna yang telah di buat di structs.

4.1.1.1 Kode Script Structs Yang Berisi OOP (Object-Oriented Programming)

4.1.1.1.1 Kode Script Class Pengguna

```
class Pengguna:
    def __init__(self, id_num, nama, email, status, password):
        self.id_num = id_num
        self.nama = nama
        self.email = email
        self.status = status
        self.password = password

    #menampilkan informasi pengguna dari class pengguna
    def display_info(self):
        print(f"ID: {self.id_num}")
        print(f>Nama: {self.nama}")
        print(f>Email: {self.email}")
        print(f>Status: {self.status}")
        print(f>Password: {self.password}"]
```

Gambar 4. 1 Kode Script Pengguna

```
from Backend.Structs.Pengguna import Pengguna

class Pegawai(Pengguna):
    def __init__(self, id_num, nama, email, status, password, jabatan):
        super().__init__(id_num, nama, email, status, password)
        self.jabatan = jabatan

    #menampilkan informasi dari class pegawai (petugas) yang merupakan turunan dari class pengguna
    def display_info(self):
        super().display_info()
        print(f"Jabatan: {self.jabatan}")
```

Gambar 4. 2 Kode Script Pegawai

```
from Backend.Structs.Pengguna import Pengguna

class Anggota(Pengguna):
    def __init__(self, id_num, nama, email, status, password, status_keanggotaan):
        super().__init__(id_num, nama, email, status, password)
        self.status_keanggotaan = status_keanggotaan

    #menampilkan informasi dari class anggota yang merupakan turunan dari class pengguna
    def display_info(self):
        super().display_info()
        print(f>Status Keanggotaan: {self.status_keanggotaan}"]
```

Gambar 4. 3 Kode Script Anggota

Pada kode script di atas, di lakukan proses pembuatan pengguna yang terdiri atas pegawai (petugas) dan anggota. Di mana kode script tersebut menggunakan konsep OOP (Object-Oriented Programming). Terdapat class pengguna, pegawai, dan anggota di mana class-class tersebut memiliki atribut dan metode yang sama dengan objeknya.

Terdapat inheritance (pewarisan) dari class pengguna kepada class pegawai dengan adanya tambahan atribut jabatan dan class anggota dengan adanya atribut status keanggotaan. Dengan polimorfisme yang di perlihatkan pada adanya fungsi display_info() di setiap class.

4.1.1.1.2 Kode Script Class Buku

```
class Buku:
    def __init__(self, judul, kategori, jumlah_buku, pengarang, penerbit, tahun_terbit, isbn, rak_buku):
        self.judul = judul
        self.kategori = kategori
        self.jumlah_buku = jumlah_buku
        self.pengarang = pengarang
        self.penerbit = penerbit
        self.tahun_terbit = tahun_terbit
        self.isbn = isbn
        self.rak_buku = rak_buku

    #menampilkan dan mengembalikan data buku dalam bentuk daftar
    def get_data(self):
        return [self.judul, self.kategori, self.jumlah_buku, self.pengarang, self.penerbit, self.tahun_terbit, self.isbn, self.rak_buku]
```

Gambar 4. 4 Kode Script Buku

Pada kode script di atas, di lakukan proses pembuatan class buku yang menggunakan konsep OOP (Object-Oriented Programming) yang memiliki atribut, yakni judul, kategori, jumlah, pengarang, penerbit, tahun terbit, isbn, dan rak. Dengan menggunakan enkapsulasi agar tidak dapat di akses secara langsung, maka terdapat metode untuk mengambil dan mencetak informasi mengenai buku, yakni get_data, debug, dan info.

Di mana nantinya, objek-objek yang telah di buat pada bagaian ini akan di lanjutkan dan import di bagian folder controller.

4.1.1.2 Kode Script Controller Yang Berisi Struktur Dasar Buku Dan Pengguna

4.1.1.2.1 Kode Script Pengelolaan Pengguna

```
import openpyxl
from Backend.Structs.Pengguna import Pengguna
from Backend.Structs.Anggota import Anggota
from Backend.Structs.Pegawai import Pegawai

class PengelolaanPengguna:
    def __init__(self, file, sheet):
        self.file = file
        self.sheet = sheet
        self.header = None
        self.users = []
        self.read()

    #membaca data buku dan menginisialisasi user dan header
    def read(self):
```

Gambar 4. 5 Kode Script Pengelolaan Pengguna

Pada kode script di atas, di lakukan pembuatan class pengelolaan pengguna yang berisi anggota, dan pegawai untuk melakukan pengelolaan pengguna dalam sistem aplikasi. Di mana akan di lakukan pencocokan antara username (email pengguna) dan password yang di hasilkan dari data excel pada file generator. Jika email dan password yang di masukkan sama maka, dapat di lakukan pengecekan apakah pengguna tersebut termasuk anggota atau pegawai (petugas).

4.1.1.2.2 Kode Script Pengelolaan Buku

```
import openpyxl
from Backend.Structs.Buku import Buku
from collections import deque

class PengelolaanBuku:
    def __init__(self, file, sheet):
        self.file = file
        self.sheet = sheet
        self.signature = []
        self.books = []
        self.read()

    #membaca data buku
    def read(self):
        sheet = openpyxl.load_workbook(self.file)[self.sheet]
        inc = 0
```

Gambar 4. 6 Kode Script Pengelolaan Buku

Pada kode script di atas, dilakukan pembuatan class pengelolaan buku untuk melakukan pengelolaan buku dalam aplikasi. Di mana dilakukan pengimporan class buku dari structs dan terdapat fungsi untuk melakukan pembacaan, penambahan, pembaruan, dan penghapusan data buku. Di mana data buku di sini diperoleh dari data excel dataset buku. Terdapat pula implementasi algoritma BFS (Breadth-First Search) yang digunakan untuk melakukan pencarian buku dan akan dijelaskan sebagai berikut.

4.1.1.2.3 Kode Script BFS (Breadth-First Search) Dalam Pencarian Buku

```
def search(self, search_attribute, search_key):
    queue = deque()

    for book in self.books:
        if getattr(book, search_attribute.lower()) == search_key:
            queue.append(book)

    #Lakukan pencarian dengan queue
    while queue:
        book = queue.popleft()

        #mengembalikan nilai jika key nya cocok
        if getattr(book, search_attribute.lower()) == search_key:
            return book

    if hasattr(book, "books"):
        for nested_book in book.books:
            if getattr(nested_book, search_attribute.lower()) == search_key:
                queue.append(nested_book)

    return None
```

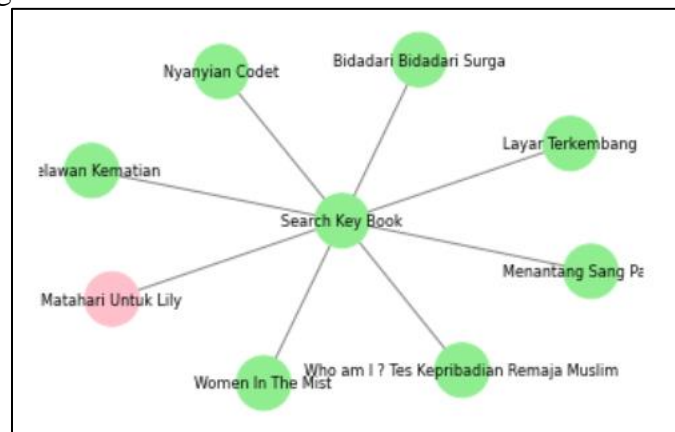
Gambar 4. 7 Kode Script BFS (Breadth-First Search)

Pada kode script di atas, dilakukan proses penerapan algoritma BFS (Breadth-First Search) dalam pencarian buku sesuai dengan judul buku. Dengan melakukan pembuatan antrian awal terlebih dahulu dan melakukan iterasi dalam daftar self.books untuk diperiksa. Jika nilai atribut pencarian sesuai dengan nilai kunci yang diberikan maka buku akan dimasukkan ke dalam proses antrian.

Selanjutnya, dilakukan pencarian melalui antrian tersebut dengan mengambil buku pertama dalam antrian dengan metode popleft() dan disimpan dalam book. Jika nilai atribut pencarian cocok dengan nilai kunci maka buku tersebut ditemukan. Jika buku memiliki atribut, seperti sub buku seperti

adanya series-series buku, maka akan di lakukan iterasi pada setiap sub bab buku. Dan jika, buku tidak di temukan maka akan di tampilkan nilai none yang menunjukkan buku tidak di temukan.

Dapat di simpulkan penggunaan algoritma BFS (Breadth-First Search) dalam pencarian di lakukan dengan mengunjungi buku yang sejajar terlebih dahulu, sebelum mengunjungi dan memeriksa sub-bab buku yang lebih dalam. Pencarian di lakukan secara bertahap melalui setiap level struktur data. Dengan implementasi algoritma BFS (Breadth-First Search) untuk pencarian buku dapat di visualisasikan sebagai berikut.



Gambar 4. 8 Visualisasi BFS (Breadth-First Search)

Pada visualiasasi di atas di gambarkan bahwa search key book menjadi pusat, di mana ini akan menjalankan perintah bahwa proses bfs (breadth-first search) di lakukan dengan pencocokan judul yang di inputkan.

4.1.2 Kode Script Generator Untuk Menghasilkan Dataset Pengguna

Pada bagian ini akan di jelaskan mengenai pembuatan dataset pengguna yang terdiri atas anggota dan pegawai.

```
import openpyxl
import random
from openpyxl.utils import get_column_letter

#menghasilkan nama random
def generate_random_name():
    first_names = ["Adi", "Budi", "Citra", "Dian", "Eka", "Fitri", "Gita", "Hadi", "Indra", "Joko"]
    last_names = ["Wijaya", "Santoso", "Kusuma", "Setiawan", "Pratiwi", "Purnama", "Utami", "Susanto", "Wahyudi", "Widodo"]
    first_name = random.choice(first_names)
    last_name = random.choice(last_names)
    return f"{first_name} {last_name}"
```

Gambar 4. 9 Kode Script Generator

Pada kode script di atas, di lakukan proses pembuatan data pengguna yang terdiri atas petugas dan anggota. Dengan atribut nama lengkap, email, status, dan password secara acak. Lalu, di hasilkan file excel bernama data dengan format xlsx sebanyak 10 data yang nantinya akan di rename menjadi DatasetPengguna dan di simpan ke dalam folder dataset.

Data-data ini nantinya akan terhubung secara langsung ke dalam aplikasi untuk keperluan pengisian halaman login. Dalam hal pembuatan data pengguna ini di buat secara acak karena, tidak bisa mendapatkan dataset mengenai data siswa dan petugas perpustakaan.

4.1.3 Kode Script Pages Untuk Menampilkan Tampilan Di GUI

Pada bagian ini akan di jelaskan mengenai langkah-langkah pembuatan tampilan GUI (Graphical-User Interface).

4.1.3.1 Kode Script Halaman Pencarian

```
import customtkinter as ctk
import tkinter.ttk as ttk
from Backend.Controller.PengelolaanBuku import PengelolaanBuku

class Pencarian:
    def __init__(self, master, isAnggota):
        self.master = master
        if(isAnggota):
            self.state = "disabled"
        else:
            self.state = "normal"
        self.pengelolaan_buku = PengelolaanBuku("Dataset/DatasetBukuPerpustakaan.xlsx", "Dataset_Buku_Perpustakaan")

    #menampilkan tampilan di GUI
    def show(self):
        self.tabview = ctk.CTkTabview(master=self.master, height=10)
        self.tabview.pack(padx=20, pady=20)

        perpustakaan_tab = self.tabview.add("Perpustakaan")
        self.tabview.set("Perpustakaan")
```

Gambar 4. 10 Kode Script Halaman Pencarian

Pada kode script di atas, di lakukan pembuatan halaman pencarian buku di GUI yang dapat di akses anggota dan petugas. Di mana terdapat perintah untuk memasukkan judul buku yang ingin di cari, kemudian akan di lakukan proses pencarian buku dengan menggunakan BFS (Breadth-First Search) dengan jika buku di temukan maka akan di tampilkan informasi mengenai buku dan jika buku tidak di temukan maka di tampilkan pesan bahwa buku tidak di temukan.

4.1.3.2 Kode Script Halaman Edit Buku

```
import customtkinter as ctk
import tkinter.ttk as ttk
from Backend.Controller.PengelolaanBuku import PengelolaanBuku

class Edit:
    def __init__(self, master):
        self.master = master
        self.pengelolaan_buku = PengelolaanBuku("Dataset/DatasetBukuPerpustakaan.xlsx", "Dataset_Buku_Perpustakaan")

    #menampilkan tampilan di GUI
    def show(self):
        self.tabview = ctk.CTkTabview(master=self.master, height=10)
        self.tabview.pack(padx=20, pady=20)

        perpustakaan_tab = self.tabview.add("Perpustakaan")
```

Gambar 4. 11 Kode Script Halaman Edit Buku

Pada kode script di atas, di lakukan pembuatan halaman edit lokasi buku di GUI yang hanya dapat di akses oleh petugas. Di mana terdapat perintah memasukkan judul buku dan rak buku baru yang di inginkan. Jika edit lokasi buku berhasil di lakukan di tampilkan pesan bahwa buku tersebut berhasil di ubah lokasinya ke dalam rak tertentu dan jika tidak maka di tampilkan pesan bahwa format masukkan tidak valid (salah).

4.1.3.3 Kode Script Halaman Tambah Buku

```
import customtkinter as ctk
import tkinter.ttk as ttk
from Pages.Edit import Edit
from Backend.Controller.PengelolaanBuku import PengelolaanBuku
from Backend.Structs.Buku import Buku

class Tambah:
    def __init__(self, master):
        self.master = master
        self.pengelolaan_buku = PengelolaanBuku("Dataset/DatasetBukuPerpustakaan.xlsx", "Dataset_Buku_Perpustakaan")

    #menampilkan tampilan di GUI
    def show(self):
        self.tabview = ctk.CTkTabview(master=self.master, height=10)
        self.tabview.pack(padx=20, pady=20)
```

Gambar 4. 12 Kode Script Halaman Tambah Buku

Pada kode script di atas, di lakukan pembuatan halaman tambah buku di GUI yang hanya dapat di akses oleh petugas. Di mana terdapat perintah untuk memasukkan atribut buku, yakni judul, kategori, jumlah, pengarang, penerbit, tahun terbit, isbn, dan rak buku. Jika buku berhasil di tambahkan maka di tampilkan pesan buku berhasil di tambah dan jika tidak maka di tampilkan input format error pastikan semua kolom di isi.

4.1.3.4 Kode Script Halaman Hapus Buku

```
import customtkinter as ctk
import tkinter.ttk as ttk
from Pages.Edit import Edit
from Pages.Tambah import Tambah
from Backend.Controller.PengelolaanBuku import PengelolaanBuku

class Hapus:
    def __init__(self, master):
        self.master = master
        self.pengelolaan_buku = PengelolaanBuku("Dataset/DatasetBukuPerpustakaan.xlsx", "Dataset_Buku_Perpustakaan")

    #menampilkan tampilan di GUI
    def show(self):
        self.tabview = ctk.CTkTabview(master=self.master, height=10)
        self.tabview.pack(padx=20, pady=20)

        perpustakaan_tab = self.tabview.add("Perpustakaan")
        self.tabview.set("Perpustakaan")
```

Gambar 4. 13 Kode Script Halaman Hapus Buku

Pada kode script di atas, di lakukan pembuatan halaman hapus buku di GUI yang hanya dapat di akses oleh petugas. Di mana terdapat perintah untuk memasukkan judul buku yang ingin di hapus. Jika buku berhasil di hapus maka di tampilkan pesan buku berhasil di hapus dan jika tidak maka di tampilkan buku tidak di temukan.

4.1.4 Kode Script App Untuk Tampilan Sistem Utama

```
import customtkinter as ctk
from PIL import Image
from PIL import ImageTk
import os
from Backend.Controller.PengelolaanPengguna import PengelolaanPengguna
from Pages.Pencarian import Pencarian
from Pages.Edit import Edit
from Pages.Hapus import Hapus
from Pages.Tambah import Tambah

class App(ctk.CTk):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
```

Gambar 4. 14 Kode Script Tampilan Aplikasi Utama

Pada kode script di atas, di lakukan pembuatan implementasi sistem antar muka. Di lakukan beberapa impor yakni, customtkinter, image, pengelolaan pengguna, pencarian, edit, hapus, tambah. Class app merupakan turunan dari customtkinter, di pergunakan untuk mengatur mode tampilan dan tema warna. Dengan melakukan pemanggilan pengelolaan pengguna untuk menampilkan dataset pengguna.

Terdapat metode first login yang di pergunakan untuk memeriksa kecocokan email dengan password jika benar maka akan di alihkan ke halaman pencarian, dengan jika pengguna tersebut termasuk anggota hanya bisa melakukan pencarian buku dan jika pengguna tersebut termasuk petugas maka bisa melakukan pencarian buku, edit lokasi buku, penambahan buku, dan penghapusan buku yang terdapat di folder pages. Kemudian kode tersebut akan di jalankan dan menampilkan tampilan GUI yang sesuai dengan yang di inginkan.

4.2 Penampilan GUI (Graphical User Interface) Untuk Pencarian Buku Dalam Sistem Perpustakaan

Pada bagian ini akan di jelaskan mengenai tampilan pada masing-masing halaman.

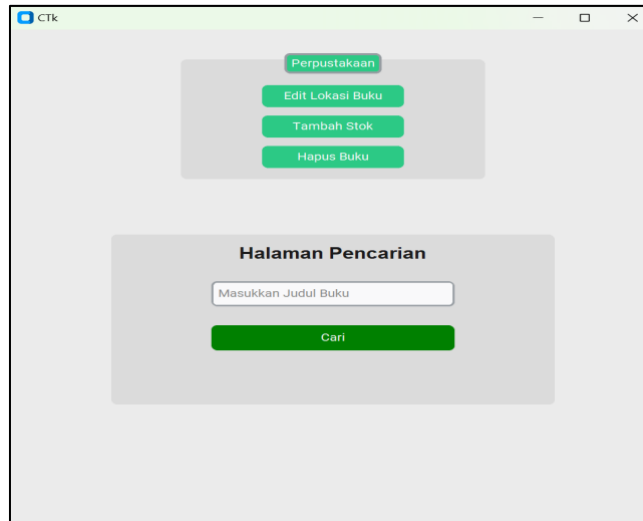
4.2.1 Halaman Login



Gambar 4. 15 Halaman Login

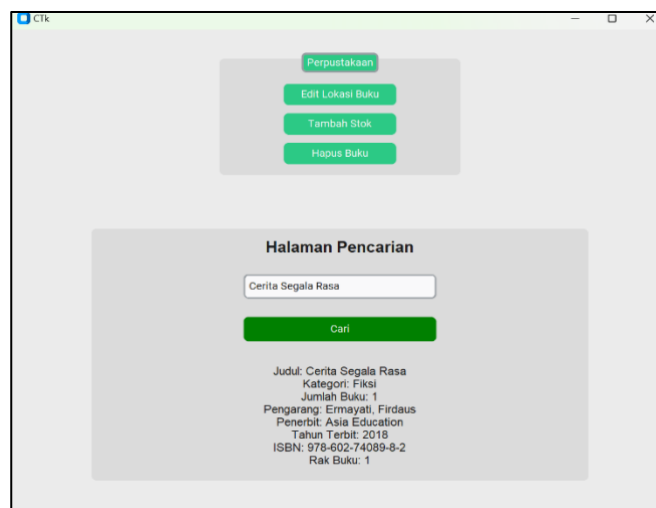
Pada gambar di atas, di tampilkan penampilan halaman login yang telah di buat pada kode script bagian App.py. Sistem ini di jalankan menggunakan library customtkinter dengan penampilan adanya halaman login yang berisi email dan password dari dataset pengguna yang terdapat dalam folder dataset. Jika email dan password yang di masukkan sesuai, maka akan berhasil masuk ke halaman pencarian.

4.2.2 Halaman Pencarian Buku



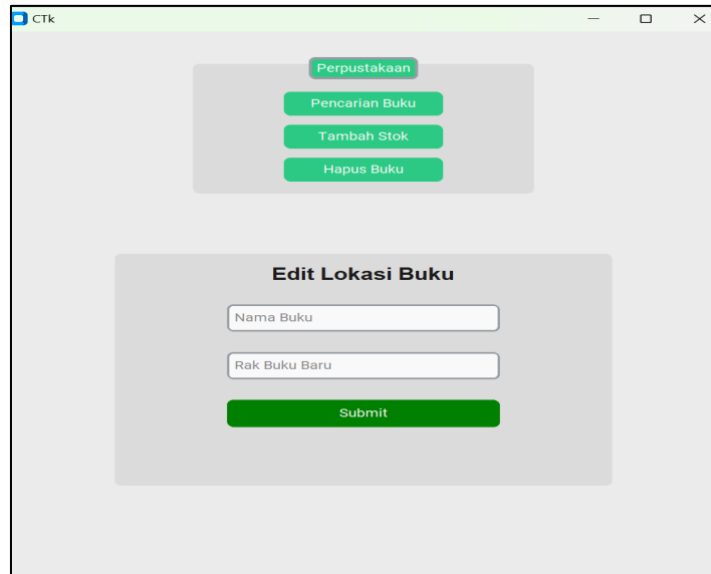
Gambar 4. 16 Halaman Pencarian

Pada gambar di atas, di tampilkan penampilan halaman pencarian yang telah di buat pada folder pages file pencarian yang bisa di akses oleh anggota dan petugas. Lalu, akan di tampilkan perintah untuk memasukkan judul buku yang di inginkan, ketika judul buku yang di inginkan tersebut terdapat dalam dataset buku perpustakaan. Maka, akan di tampilkan informasi dari buku, yakni judul, kategori, jumlah, pengarang, penerbit, tahun terbit, isbn, dan rak buku. Dan jika tidak, maka akan di tampilkan bahwa buku tersebut tidak di temukan. Untuk penggunaan halaman ini dapat di lakukan sebagai berikut.



Gambar 4. 17 Penggunaan Halaman Pencarian

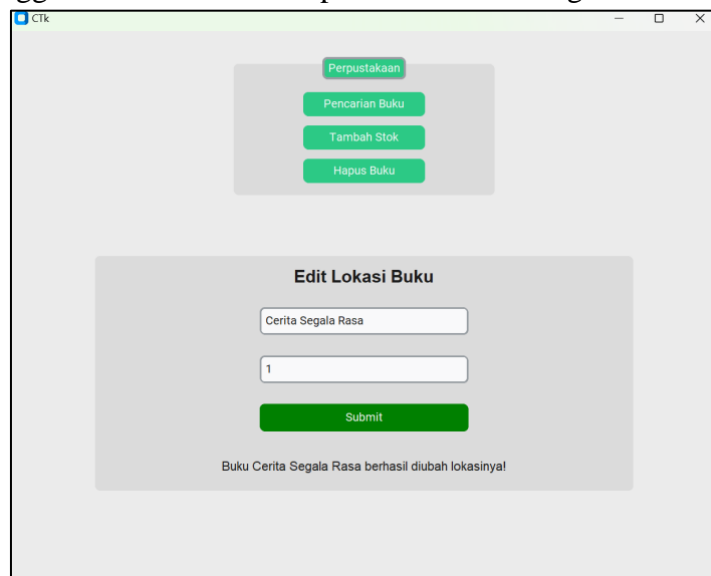
4.2.3 Halaman Edit Lokasi Buku



The screenshot shows a Tkinter window titled 'CTk'. Inside, there's a sidebar with buttons: 'Perpustakaan', 'Pencarian Buku', 'Tambah Stok', and 'Hapus Buku'. The main area features a form titled 'Edit Lokasi Buku' with two input fields: 'Nama Buku' and 'Rak Buku Baru', followed by a green 'Submit' button.

Gambar 4. 18 Halaman Edit Lokasi Buku

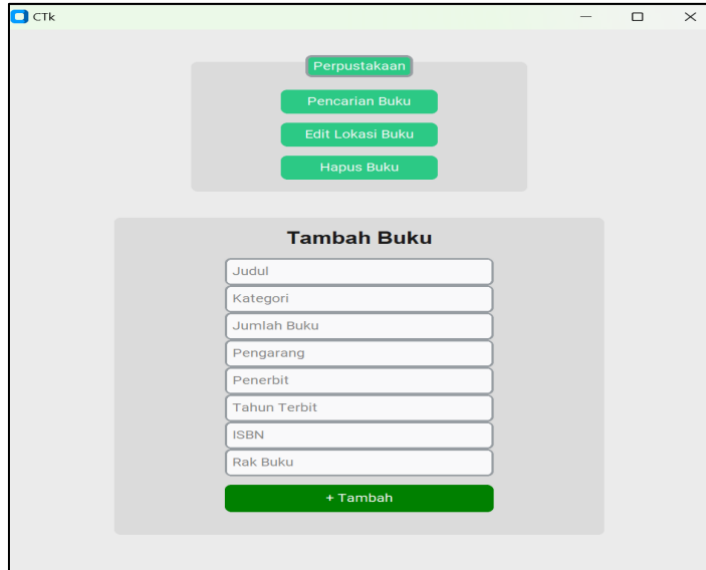
Pada gambar di atas, di tampilkan penampilan halaman edit lokasi buku yang telah di buat di folder pages file edit yang bisa di akses oleh petugas. Lalu akan di tampilkan perintah untuk memasukkan judul buku dan rak buku yang di inginkan untuk mengedit lokasi buku tersebut, ketika judul buku tersebut terdapat dalam dataset buku perpustakaan. Maka, akan di tampilkan informasi bahwa buku tersebut berhasil di ubah lokasinya. Dan jika tidak, maka di tampilkan bahwa format masukan tidak valid (salah). Untuk penggunaan halaman ini dapat di lakukan sebagai berikut.



This screenshot shows the same 'Edit Lokasi Buku' form, but now it contains the text 'Cerita Segala Rasa' in the 'Nama Buku' field and '1' in the 'Rak Buku Baru' field. The green 'Submit' button is still present. Below the form, a message reads: 'Buku Cerita Segala Rasa berhasil diubah lokasinya!'. The sidebar buttons remain the same.

Gambar 4. 19 Penggunaan Halaman Edit Lokasi Buku

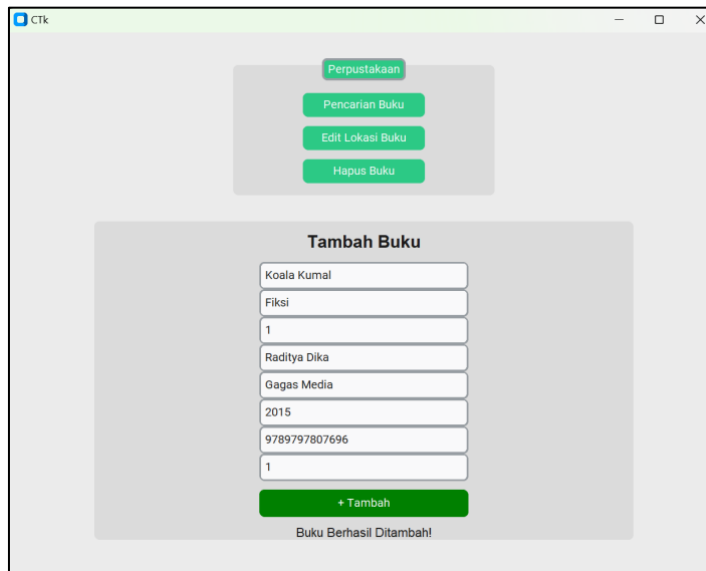
4.2.4 Halaman Tambah Stok Buku



The screenshot shows a web application window titled 'CTk'. It features a sidebar with a 'Perpustakaan' (Library) button and a main menu with 'Pencarian Buku' (Book Search), 'Edit Lokasi Buku' (Edit Book Location), and 'Hapus Buku' (Delete Book). The main content area is titled 'Tambah Buku' (Add Book) and contains a form with the following fields: Judul (Title), Kategori (Category), Jumlah Buku (Number of Books), Pengarang (Author), Penerbit (Publisher), Tahun Terbit (Year Published), ISBN, and Rak Buku (Shelf). A green '+ Tambah' (Add) button is at the bottom of the form.

Gambar 4. 20 Halaman Stok Buku

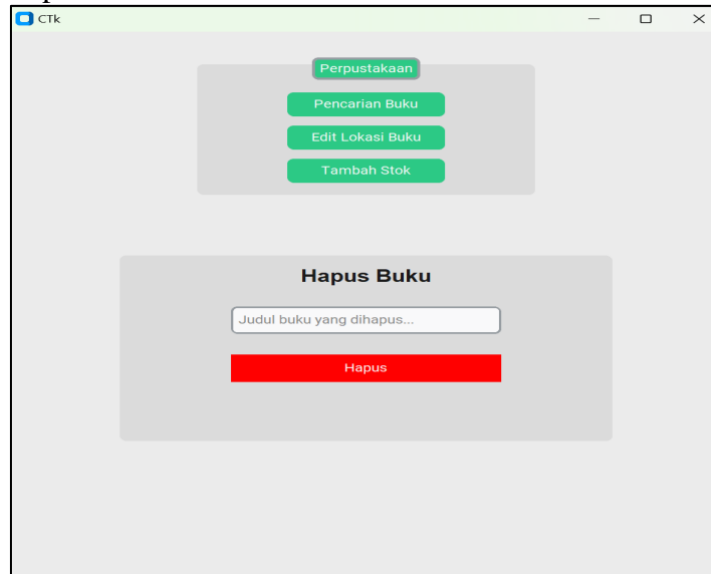
Pada gambar di atas, di lakukan penampilan halaman tambah stok buku yang telah di buat di folder pages file tambah yang bisa di akses oleh petugas. Lalu akan di tampilkan perintah untuk memasukkan informasi mengenai buku, yakni judul, kategori, jumlah, pengarang, penerbit, tahun terbit, isbn, dan rak buku. Jika tambah stok buku berhasil di lakukan maka akan di tampilkan informasi bahwa buku tersebut berhasil di tambahkan. Dan jika tidak maka akan di tampilkan informasi bahwa informasi format error!. Untuk penggunaan halaman ini dapat di lakukan sebagai berikut.



The screenshot shows the same 'Tambah Buku' form as in the previous image, but with sample data entered into the fields: Judul: Koala Kumal, Kategori: Fiksi, Jumlah Buku: 1, Pengarang: Raditya Dika, Penerbit: Gagas Media, Tahun Terbit: 2015, ISBN: 9789797807696, and Rak Buku: 1. The green '+ Tambah' button is still present. Below the form, a message 'Buku Berhasil Ditambah!' (Book Successfully Added!) is displayed.

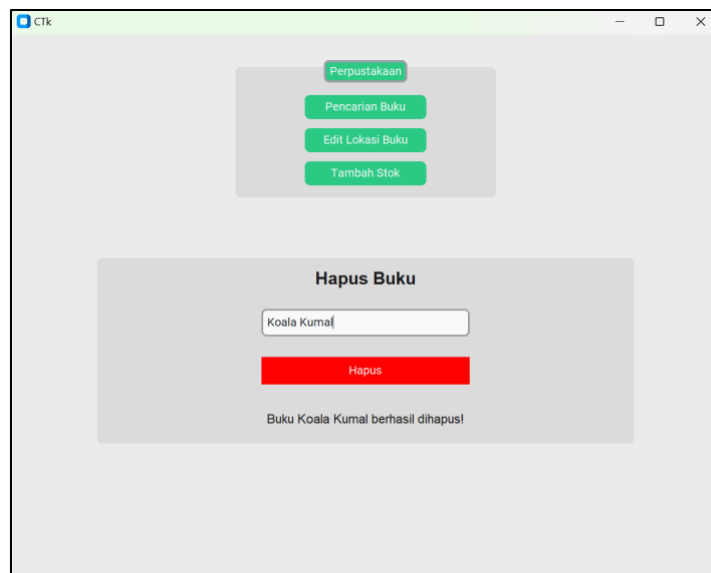
Gambar 4. 21 Penggunaan Halaman Tambah Stok Buku

4.2.5 Halaman Hapus Buku



Gambar 4. 22 Halaman Hapus Buku

Pada gambar di atas, di lakukan penampilan halaman hapus buku yang telah di buat pada folder pager file hapus yang bisa di akses oleh petugas. Lalu, akan di tampilkan perintah untuk memasukkan judul buku yang ingin di hapus, ketika judul buku tersebut terdapat dalam dataset buku perpustakaan maka akan di tampilkan informasi bahwa buku tersebut berhasil di hapus. Dan jika tidak, maka akan di tampilkan informasi bahwa buku dengan judul tersebut tidak di temukan. Untuk penggunaan halaman ini dapat di lakukan sebagai berikut.



Gambar 4. 23 Halaman Hapus Buku

4.3 Hasil Pencarian Buku Menggunakan Algoritma BFS (Breadth-First Search) Dalam Bentuk Tabel

Pada bagian ini di jelaskan pencarian pada masing-masing halaman dalam bentuk tabel.

Tabel 4.3. 1 Hasil Halaman Pencarian

Percobaan Ke	Judul Buku	Status Pencarian
1	Matahari Untuk Lily	Buku Di Temukan Dan Di Tampilkan Informasi Buku
2	Manusia Ikan	Buku Tidak Di Temukan (karena telah di lakukan penghapusan di percobaan ke-1)

Tabel 4.3. 2 Hasil Halaman Edit Buku

Percobaan Ke	Judul Buku	Rak Buku	Status Pengeditan
1	Manusia Ikan	3	Buku Berhasil Di Ubah
2	Fisika Universitas	3	Buku Berhasil Di Ubah

Tabel 4.3. 3 Hasil Halaman Tambah Buku

Percobaan Ke	Identitas Buku	Status Penambahan
1	Sebuah Seni Untuk Bersikap Bodo Amat	Buku Berhasil Di Tambahkan
2	Satwa Terancam Bahaya	Buku Berhasil Di Tambahkan

Tabel 4.3. 4 Hasil Halaman Hapus Buku

Percobaan Ke	Identitas Buku	Status Penghapusan
1	Manusia Ikan	Buku Berhasil Di Hapus
2	Satwa Terancam Bahaya	Buku Berhasil Di Hapus

5. BAB V : KESIMPULAN

Implementasi algoritma BFS (Breadth-First Search) dalam pencarian buku untuk sistem peminjaman perpustakaan SMP Negeri 1 Merakurak diharapkan dapat memudahkan pencarian buku yang tersedia. Dengan menerapkan algoritma BFS, pencarian buku dapat dilakukan dengan proses yang lebih cepat dan dapat menemukan jalur atau urutan yang sesuai dalam pencarian. Dengan melakukan pencarian sesuai judul buku, dan di cek apakah buku tersebut memiliki series atau tidak.

Untuk itu, diperlukan beberapa tahapan untuk melakukan pembuatan perancangan sistem. Tahapan tersebut dimulai dengan pembuatan kode script mengenai pembuatan folder backend yang berisi mengenai structs untuk merepresentasikan pengguna, pegawai, anggota, dan buku yang menggunakan konsep OOP (Object-Oriented Programming) dan controller untuk pengelolaan buku dan pengelolaan pengguna. Selanjutnya, setelah mendapatkan output berupa sistem perpustakaan untuk mengelola perpustakaan, dengan cara kerja sistem yakni, jika pengguna terdaftar sebagai anggota, mereka hanya dapat mengakses tampilan pencarian buku. Sedangkan jika pengguna terdaftar sebagai petugas, mereka dapat melakukan pencarian buku, menambah stok buku, mengedit lokasi buku, dan menghapus buku.

Dengan adanya penerapan sistem perpustakaan yang masih menggunakan metode manual mengakibatkan tidak adanya efisiensi waktu dan ketidakpastian ketersediaan buku. Oleh karena itu, dengan menerapkan algoritma BFS (Breadth-First Search) dalam sistem perpustakaan, diharapkan dapat memudahkan pencarian buku dan meningkatkan efisiensi waktu dalam pengelolaan perpustakaan. Dalam penerapan ini memiliki beberapa kelebihan dan kekurangan, di antaranya penerapan algoritma BFS (Breadth-First Search) memungkinkan pencarian buku dilakukan lebih cepat dan efisien, sistem perpustakaan memudahkan pengelolaan buku dan pengguna, termasuk fitur tambah stok, edit lokasi, dan hapus buku. Namun, terdapat pula kekurangan dengan penerapan ini, yakni adanya batasan aksesibilitas fitur dapat mempengaruhi pengalaman pengguna yang terbatas pada peran tertentu (misalnya, anggota hanya dapat mengakses tampilan pencarian buku).

6. DAFTAR PUSTAKA

Arcita, T. SISTEM INFORMASI PENGELOLAAN DATA BUKU PERPUSTAKAAN.

Sukmawan, R., Gultom, U., Haroen, R., & Yasin, V. (2021). Sistem Peminjaman Dan Pengembalian Buku Perpustakaan Keliling Berbasis Web Menggunakan Algoritma String Matching Pada Suku Dinas Perpustakaan Dan Kearsipan Kota Administrasi Jakarta Utara. JISICOM (Journal of Information System, Informatics and Computing), 5(1), 144-173.

Mubarok, M. S., Syauqy, D., & Fitriyah, H. (2021). Sistem Rute Terpendek Pencarian Buku Di Perpustakaan Menggunakan Algoritme Dijkstra. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 5(6), 2125-2131.

FAUZAN, M. A. (2022). STRATEGI ALGORITMA ALGORITMA PENCARIAN IF4005.

Tiningsih, E. W. T. (2011). SISTEM INFORMASI PENGOLAHAN DATA PERPUSTAKAAN PADA SMK NEGERI 1 DEMAK: Vol. 6 No. 2 Maret 2011. Tatal, 6(2), 89-102.

Pangestu, A. D., & Utami, L. A. (2022). Sistem Informasi Perpustakaan Berbasis Web pada SDN Cawang 12 Pagi. IJIS-Indonesian Journal On Information System, 7(1), 25-34.

7. LAMPIRAN

- Lampiran Link YouTube : <https://youtu.be/DYnKVfsSwvE>
- Lampiran Link YouTube Kode Script Dan GUI : <https://youtu.be/lb4gFiM5Ox4>
- Lampiran Kode Script Dan Pemrograman Aplikasi Di Gdrive :
https://drive.google.com/drive/folders/1FidlqEa_NqukLAhtC01vUADbuVruC5zU?usp=sharing (SistemPerpustakaanSMPN1)