

LAPORAN TUGAS PROYEK AKHIR KULIAH

PEMROGRAMAN BERBASIS OBJEK

Aplikasi Perpustakaan Bookcamps



**UIN SUNAN AMPEL
S U R A B A Y A**

Dosen Pengampu:

Bayu Adi Nugroho, Ph.D

Disusun Oleh:

Rizki Cahyani Fitonah

09020622042

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA

FAKULTAS SAINS DAN TEKNOLOGI

SISTEM INFORMASI

2023/2024

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perpustakaan merupakan salah satu elemen kunci dalam proses pembelajaran dan penelitian di perguruan tinggi. Namun, pengelolaan perpustakaan yang efisien dan efektif menjadi tantangan tersendiri, terutama dengan adanya jumlah koleksi yang terus bertambah dan kebutuhan pengguna yang beragam. Dalam konteks ini, pembuatan aplikasi perpustakaan menjadi solusi yang relevan untuk meningkatkan efisiensi dan memberikan pengalaman pengguna yang lebih baik. Aplikasi perpustakaan dapat membantu dalam manajemen koleksi, peminjaman dan pengembalian buku, serta memberikan akses cepat dan mudah terhadap sumber informasi digital.

Pembuatan aplikasi perpustakaan ini tidak hanya mencerminkan upaya untuk menjawab tantangan aktual dalam pengelolaan sumber daya informasi, tetapi juga merupakan bagian integral dari tahap akhir pendidikan. Proyek ini secara khusus diarahkan sebagai tugas akhir dalam kurikulum perkuliahan, dengan penugasan langsung dari dosen pembimbing. Pembuatan aplikasi perpustakaan bukan hanya merupakan proyek teknis semata, tetapi juga merupakan demonstrasi kemampuan mahasiswa dalam menerapkan konsep dan teori yang telah mereka pelajari selama perkuliahan ke dalam suatu solusi praktis dan tangguh.

1.2 Tujuan

Adapun tujuan dari proyek ini adalah untuk memberikan kesempatan kepada kami sebagai mahasiswa untuk menerapkan pengetahuan dan keterampilan yang telah dipelajari selama masa studi.

BAB II

LANDASAN TEORI

2.1 Dasar Teori

Menurut Wikipedia pemrograman berorientasi objek atau *object-oriented programming* yang biasa disingkat OOP merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk *field* atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai *method*. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Berikut cakupan dasar pemrograman berbasis objek:

1. Kelas dan Objek:

- Definisi Kelas: Kelas adalah struktur dasar dalam OOP yang mendefinisikan atribut (variabel) dan metode (fungsi) yang dapat dimiliki oleh suatu objek.
- Objek: Objek adalah instansi konkret dari suatu kelas. Setiap objek memiliki atribut yang unik dan dapat memanggil metode khususnya.

2. Inheritance (Pewarisan):

- Konsep Pewarisan: Inheritance memungkinkan kelas baru ("subclass" atau "child class") mewarisi atribut dan metode dari kelas yang sudah ada ("superclass" atau "parent class").
- Keuntungan: Meningkatkan reusable code dan memfasilitasi pengelolaan hierarki kelas.

3. Polimorfisme:

- Arti Polimorfisme: Polimorfisme memungkinkan suatu metode memiliki nama yang sama tetapi dapat melakukan tindakan yang berbeda tergantung pada objek yang memanggilnya.
- Contoh Penggunaan: Metode overriding dan interface adalah implementasi polimorfisme dalam OOP.

4. Encapsulation (Enkapsulasi):

- Prinsip Enkapsulasi: Enkapsulasi melibatkan pengemasan atribut dan metode ke dalam satu unit tunggal (kelas), dengan mengakses atribut menggunakan metode (getters dan setters).
- Keuntungan: Menyediakan kontrol akses yang lebih baik dan membatasi akses langsung ke atribut.

5. Abstraksi:

- Arti Abstraksi: Abstraksi melibatkan menyembunyikan detail implementasi dan hanya mengekspos fungsionalitas penting suatu objek.

- Implementasi di OOP: Interface dan kelas abstrak digunakan untuk menciptakan tingkat abstraksi.

6. Class Diagram:

- Definisi Class Diagram: Class diagram adalah representasi visual dari struktur kelas, hubungan antar kelas, dan sifat atribut dan metode.
- Manfaat: Memberikan pandangan visual yang jelas tentang hubungan antar objek dalam suatu sistem.

7. Persistence API:

Sebuah antarmuka pemrograman aplikasi (API) yang menyediakan metode dan fungsi untuk melakukan operasi persistence, seperti menyimpan, mengambil, dan menghapus data.

2.2 Software yang Digunakan

Software yang digunakan dalam proyek aplikasi Perpustakaan Bookcamps adalah sebagai berikut:

- Bahasa Pemrograman: Java (JDK 8)
- Aplikasi IDE: Apache NetBeans IDE 15
- Database: PostgreSQL 14
- Report: Jasper Report

2.3 Hardware yang Digunakan

Hardware yang digunakan dalam proyek aplikasi Perpustakaan Bookcamps adalah sebagai berikut:

- RAM: 4 GB
- System Operasi: 64-bit

BAB III

DATABASE

1. Membuat Database dengan nama “Db_perpustakaan”

2. Membuat Tabel:

a) Create table Buku

-- Table: public.buku

-- DROP TABLE IF EXISTS public.buku;

CREATE TABLE IF NOT EXISTS public.buku

```
(
  id_buku character(11) COLLATE pg_catalog."default" NOT NULL,
  judul character varying(50) COLLATE pg_catalog."default",
  subjudul character varying(50) COLLATE pg_catalog."default",
  isbn character(13) COLLATE pg_catalog."default",
  bahasa character varying(15) COLLATE pg_catalog."default",
  penerbit character varying(50) COLLATE pg_catalog."default",
  tahun_terbit character varying(4) COLLATE pg_catalog."default",
  jumlah_halaman integer,
  jumlah_buku integer,
  CONSTRAINT pk_buku PRIMARY KEY (id_buku)
)
```

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.buku

OWNER to postgres;

-- Index: buku_pk

-- DROP INDEX IF EXISTS public.buku_pk;

CREATE UNIQUE INDEX IF NOT EXISTS buku_pk

ON public.buku USING btree

(id_buku COLLATE pg_catalog."default" ASC NULLS LAST)

TABLESPACE pg_default;

b) Create Table Peminjam

-- Table: public.peminjam

-- DROP TABLE IF EXISTS public.peminjam;

CREATE TABLE IF NOT EXISTS public.peminjam

```
(
  id_peminjam character(5) COLLATE pg_catalog."default" NOT NULL,
  nama character varying(30) COLLATE pg_catalog."default",
```

```

nim character varying(15) COLLATE pg_catalog."default",
fakultas character varying(30) COLLATE pg_catalog."default",
program_studi character varying(30) COLLATE pg_catalog."default",
semester integer,
alamat character varying(100) COLLATE pg_catalog."default",
telephone character varying(13) COLLATE pg_catalog."default",
email character varying(30) COLLATE pg_catalog."default",
CONSTRAINT pk_peminjam PRIMARY KEY (id_peminjam)
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS public.peminjam

```

```

    OWNER to postgres;

```

```

-- Index: peminjam_pk

```

```

-- DROP INDEX IF EXISTS public.peminjam_pk;

```

```

CREATE UNIQUE INDEX IF NOT EXISTS peminjam_pk

```

```

    ON public.peminjam USING btree

```

```

    (id_peminjam COLLATE pg_catalog."default" ASC NULLS LAST)

```

```

    TABLESPACE pg_default;

```

c) Create Table Skripsi

```

-- Table: public.skripsi

```

```

-- DROP TABLE IF EXISTS public.skripsi;

```

```

CREATE TABLE IF NOT EXISTS public.skripsi

```

```

(
    id_skripsi character(11) COLLATE pg_catalog."default" NOT NULL,
    judul character varying(50) COLLATE pg_catalog."default",
    bahasa character varying(15) COLLATE pg_catalog."default",
    fakultas character varying(30) COLLATE pg_catalog."default",
    program_studi character varying(30) COLLATE pg_catalog."default",
    tahun character varying(4) COLLATE pg_catalog."default",
    penulis character varying(30) COLLATE pg_catalog."default",
    pembimbing character varying(30) COLLATE pg_catalog."default",
    jumlah_halaman integer,
    jumlah_salinan integer,
    CONSTRAINT pk_skripsi PRIMARY KEY (id_skripsi)
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS public.skripsi

```

```

    OWNER to postgres;

```

d) Create Table Peminjaman

-- Table: public.peminjaman

-- DROP TABLE IF EXISTS public.peminjaman;

```
CREATE TABLE IF NOT EXISTS public.peminjaman
(
    id_petugas character(5) COLLATE pg_catalog."default" NOT NULL,
    id_peminjam character(5) COLLATE pg_catalog."default" NOT NULL,
    no_peminjaman character(10) COLLATE pg_catalog."default" NOT
NULL,
    tanggal_pinjam date,
    tanggal_kembali date,
    keterangan character varying(100) COLLATE pg_catalog."default",
    status character varying(8) COLLATE pg_catalog."default",
    CONSTRAINT pk_peminjaman PRIMARY KEY (id_petugas,
id_peminjam, no_peminjaman),
    CONSTRAINT fk_peminjam_melakukan_peminjam FOREIGN KEY
(id_peminjam)
        REFERENCES public.peminjam (id_peminjam) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT,
    CONSTRAINT fk_peminjam_melakukan_petugas FOREIGN KEY
(id_petugas)
        REFERENCES public.petugas (id_petugas) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.peminjaman
    OWNER to postgres;
```

e) Create Table Pengarang

-- Table: public.pengarang

-- DROP TABLE IF EXISTS public.pengarang;

```
CREATE TABLE IF NOT EXISTS public.pengarang
(
    id_buku character(11) COLLATE pg_catalog."default",
    nama_pengarang character varying(30) COLLATE pg_catalog."default",
    CONSTRAINT fk_pengarang_memiliki_buku FOREIGN KEY (id_buku)
        REFERENCES public.buku (id_buku) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
)
```

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.pengarang
OWNER to postgres;

f) Create Table Petugas

-- Table: public.petugas

-- DROP TABLE IF EXISTS public.petugas;

CREATE TABLE IF NOT EXISTS public.petugas

(
id_petugas character(5) COLLATE pg_catalog."default" NOT NULL,
nama character varying(30) COLLATE pg_catalog."default",
email character varying(30) COLLATE pg_catalog."default",
alamat character varying(100) COLLATE pg_catalog."default",
jenis_kelamin character(1) COLLATE pg_catalog."default",
telephone character varying(13) COLLATE pg_catalog."default",
username character varying(30) COLLATE pg_catalog."default",
password character varying(100) COLLATE pg_catalog."default",
CONSTRAINT pk_petugas PRIMARY KEY (id_petugas)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.petugas
OWNER to postgres;

g) Create Table Detail Buku

-- Table: public.detail_buku

-- DROP TABLE IF EXISTS public.detail_buku;

CREATE TABLE IF NOT EXISTS public.detail_buku

(
id_buku character(11) COLLATE pg_catalog."default" NOT NULL,
id_petugas character(5) COLLATE pg_catalog."default" NOT NULL,
id_peminjam character(5) COLLATE pg_catalog."default" NOT NULL,
no_peminjaman character(10) COLLATE pg_catalog."default" NOT
NULL,
jumlah integer,
CONSTRAINT pk_detail_buku PRIMARY KEY (id_buku, id_petugas,
id_peminjam, no_peminjaman),
CONSTRAINT fk_detail_b_memiliki7_peminjam FOREIGN KEY
(id_peminjam, id_petugas, no_peminjaman)


```

        REFERENCES public.peminjaman (id_peminjam, id_petugas,
no_peminjaman) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT,
    CONSTRAINT fk_detail_b_memiliki8_buku FOREIGN KEY (id_buku)
        REFERENCES public.buku (id_buku) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
    )

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.detail_buku
    OWNER to postgres;

```

h) Create Table Detail Skripsi

```

-- DROP TABLE IF EXISTS public.detail_skripsi;

CREATE TABLE IF NOT EXISTS public.detail_skripsi
(
    id_skripsi character(11) COLLATE pg_catalog."default" NOT NULL,
    id_petugas character(5) COLLATE pg_catalog."default" NOT NULL,
    id_peminjam character(5) COLLATE pg_catalog."default" NOT NULL,
    no_peminjaman character(10) COLLATE pg_catalog."default" NOT
NULL,
    jumlah integer,
    CONSTRAINT pk_detail_skripsi PRIMARY KEY (id_skripsi,
id_petugas, id_peminjam, no_peminjaman),
    CONSTRAINT fk_detail_s_memiliki5_peminjam FOREIGN KEY
(id_peminjam, id_petugas, no_peminjaman)
        REFERENCES public.peminjaman (id_peminjam, id_petugas,
no_peminjaman) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT,
    CONSTRAINT fk_detail_s_memiliki6_skripsi FOREIGN KEY
(id_skripsi)
        REFERENCES public.skripsi (id_skripsi) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.detail_skripsi
    OWNER to postgres;
-- Index: detail_skripsi_pk

```

```

-- DROP INDEX IF EXISTS public.detail_skripsi_pk;

CREATE UNIQUE INDEX IF NOT EXISTS detail_skripsi_pk
    ON public.detail_skripsi USING btree
    (id_skripsi COLLATE pg_catalog."default" ASC NULLS LAST,
    id_petugas COLLATE pg_catalog."default" ASC NULLS LAST,
    id_peminjam COLLATE pg_catalog."default" ASC NULLS LAST,
    no_peminjaman COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;
-- Index: memiliki5_fk

-- DROP INDEX IF EXISTS public.memiliki5_fk;

CREATE INDEX IF NOT EXISTS memiliki5_fk
    ON public.detail_skripsi USING btree
    (id_petugas COLLATE pg_catalog."default" ASC NULLS LAST,
    id_peminjam COLLATE pg_catalog."default" ASC NULLS LAST,
    no_peminjaman COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;
-- Index: memiliki6_fk

-- DROP INDEX IF EXISTS public.memiliki6_fk;

CREATE INDEX IF NOT EXISTS memiliki6_fk
    ON public.detail_skripsi USING btree
    (id_skripsi COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;

```

i) Create Table Kategori Buku

```

-- Table: public.kategori_buku

-- DROP TABLE IF EXISTS public.kategori_buku;

CREATE TABLE IF NOT EXISTS public.kategori_buku
(
    id_buku character(11) COLLATE pg_catalog."default",
    kategori character varying(20) COLLATE pg_catalog."default",
    CONSTRAINT fk_kategori_memiliki1_buku FOREIGN KEY (id_buku)
        REFERENCES public.buku (id_buku) MATCH SIMPLE
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
)
    TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.kategori_buku

```

OWNER to postgres;

j) Create Table Kategori Skripsi

-- Table: public.kategori_skripsi

-- DROP TABLE IF EXISTS public.kategori_skripsi;

```
CREATE TABLE IF NOT EXISTS public.kategori_skripsi
(
    id_skripsi character(11) COLLATE pg_catalog."default",
    kategori character varying(20) COLLATE pg_catalog."default",
    CONSTRAINT fk_kategori_memiliki3_skripsi FOREIGN KEY
(id_skripsi)
    REFERENCES public.skripsi (id_skripsi) MATCH SIMPLE
    ON UPDATE RESTRICT
    ON DELETE RESTRICT
)
```

TABLESPACE pg_default;

```
ALTER TABLE IF EXISTS public.kategori_skripsi
    OWNER to postgres;
```

BAB IV PERANCANGAN APLIKASI

1. Tampilan Dashboard



2. Tampilan Menu Buku



3. Tampilan Input Data Buku

INPUT DATA BUKU

05 Desember 2023 Logout

ID Buku ISBN Tahun Terbit

Judul Buku

Subjudul Buku

Penerbit

Bahasa

Kategori: Ilmu Komput...

Jumlah Halaman: 0

Jumlah Buku: 0

Pengarang

TAMBAH BATAL

ISBN	PENERBIT	KATEGORI
1234567	Grasindo	
4345455	gramedia	
456 982 334	Gramedia	
575 458 456	Gramedia	
9780743273565	Gramedia	A Novel
9780061120084	Gramedia	Non Fiksi
9780451524935	Grasindo	A Novel
9780241950425	Bentang Pustaka	A Novel
9780061120091	Mizan	Fiksi
9780060850524	-	Non Fiksi
9781590302255	Shambhala	Non Fiksi
9780345391803	Harper	Sejarah
9780062316097	Harper	Sejarah
9780544003415	Mariner Books	A Novel
9780064471046	HarperCollins	A Novel
9780307474278	Anchor Books	Biografi
9789793458961	Bentang Pustaka	Novel
9789793850023	Republika	Novel
9789794034873	Pramoedya Ananta Toer	Novel
9786020383174	GagasMedia	Novel
9786020303271	Bentann Pustaka	Novel

4. Tampilan Menu Skripsi

DATA SKRIPSI PERPUSTAKAAN

05 Desember 2023 Logout

Cari

ISBN

CETAK

TAMBAH HAPUS UBAH

ID Skripsi	Judul	Penulis	Tahun
13123	fdfdf	ffss	3242

5. Tampilan Input Data Skripsi

INPUT DATA SKRIPSI

05 Desember 2023 [Logout](#)

Form Input:

ID Skripsi:

Tahun:

Pembimbing:

Judul Skripsi:

Penulis:

Program Studi:

Fakultas:

Bahasa:

Kategori:

Jumlah Halaman:

Jumlah Salinan:

[TAMBAH](#)

Table Data:

Penulis	Tahun
mfss	3242

[CETAK](#)

6. Tampilan Menu Anggota

BOOKCAMPS

[Dashboard](#)

[Anggota](#)

[Buku](#)

[Skripsi](#)

[Peminjaman](#)

[Petugas](#)

DAFTAR ANGGOTA

Cari

ID Anggota

[TAMBAH](#)

[HAPUS](#)

[UBAH](#)

ID Anggota	Nama	NIM	Program Studi	Fakultas	Alamat	Telephone	E-mail	Semester
12345	rizki cahyani	12332324	sistem informasi	saintek	surabaya	081233747239	rcahyani@gmail...	2
23456	Dea Ayu	33445566	Sistem Informasi	Saintek	Kebonsari	0821232372	deaayu@gmail....	2
34567	Fauziah Afrisa	55667788	Biologi	Saintek	Gedangan	0815661243	fauziah88@gm...	3

05 Desember 2023 [Logout](#)

20:55
05/12/2023

7. Tampilan Input Data Anggota

The screenshot shows a web application interface. In the foreground, a modal window titled "INPUT DATA ANGGOTA" is open. It contains several input fields for member information: ID Anggota, NIM, Semester (a dropdown menu currently showing '0'), Nama, Alamat, Program Studi, Fakultas, Telephone, and E-mail. At the bottom of the modal are two buttons: "TAMBAH" (Add) and "BATAL" (Cancel). In the background, a table of member data is visible. The table has columns for ID, Fakultas, Alamat, Telephone, E-mail, and Semester. It contains three rows of data.

ID	Fakultas	Alamat	Telephone	E-mail	Semester
nas	saintek	surabaya	081233747239	rcahyani@gmail...	2
nas	Saintek	Kebonsari	0821232372	deayu@gmail...	2
	Saintek	Gedangan	0815661243	fauziah88@gm...	3

8. Tampilan Menu Peminjaman

The screenshot displays the "DATA PEMINJAMAN PERPUSTAKAAN BOOKCAMPS" page. On the left is a sidebar menu with icons and labels for "Dashboard", "Anggota", "Buku", "Skripsi", "Peminjaman", and "Petugas". The main content area includes a search bar, a "NO PEMINJAMAN" dropdown, and a table of borrowing records. Below the table is a "DETAIL PEMINJAMAN" section with various input fields for borrowing details, including "Tanggal Pinjam", "Tanggal Kembali", "Status" (a dropdown menu showing "Dipinjam"), "Keterangan", "ID Peminjam", "Nama Peminjam", "Fakultas Peminjam", "Prodi Peminjam", "Email Peminjam", and "Telephone Peminjam". At the bottom of the main area is a table with columns "No", "Judul", "Kategori", and "Pengarang", followed by a "SIMPAN" (Save) button. The top right corner shows the date "05 Desember 2023" and a "Logout" button.

NO. PEMINJAMAN	NAMA PEMINJAM	STATUS	TANGGAL PINJAM	TANGGAL KEMBALI
	rizki cahyani	Dipinjam	2023-12-05	0014-06-15

9. Tampilan Input Data Peminjaman

05 Desember 2023 Logout

INPUT PEMINJAMAN

NO :

Tanggal

Pinjam: 2023-12-05 Kembali:

Peminjam

ID:

BUKU

ID Buku: ISBN: Judul: Tahun: Bahasa: Pengarang: Penerbit: Jumlah: 0

SKRIPSI

ID Skripsi: Judul: Tahun: Bahasa: Prodi: Penulis: Pembimbing: Jumlah: 0

ID Buku

ID Buku	Judul	Pengarang	Jumlah

ID Skripsi

ID Skripsi	Judul	Penulis	Jumlah

BATAL SIMPAN INPUT BUKU INPUT SKRIPSI

10. Tampilan Menu Petugas

05 Desember 2023 Logout

DATA PETUGAS

Cari: ID Anggota:

TAMBAH HAPUS UBAH

ID Petugas	Nama	Alamat	Telephone	Email	Jenis Kelamin
5566	Fatiq	Gununga Any...	0842236546	fatiq@gmail...	L

11. Tampilan Input Data Petugas

The screenshot displays a web application interface. On the left, a modal window titled "INPUT DATA PETUGAS" is open, containing a form for adding a new employee. The form includes input fields for ID Petugas, Nama, Alamat, Telephone, Email, Username, Password, and Ulangi Password, along with a dropdown menu for Jenis Kelamin (currently set to "Laki-laki"). At the bottom of the form are two red buttons: "TAMBAH" and "BATAL".

In the background, the main application window is visible. It has a dark header with the date "05 Desember 2023" and a "Logout" button. Below the header is a search bar with a magnifying glass icon and a dropdown arrow. A table is displayed below the search bar, showing employee data:

Alamat	Telephone	Email	Jenis Kelamin
Gununga Any...	0842236546	fatig@gmail....	L

The Windows taskbar at the bottom shows various application icons and the system clock indicating 20:57 on 05/12/2023.

BAB V

KESIMPULAN

Proyek pembuatan aplikasi perpustakaan "BookCamps" bukan hanya menghasilkan produk yang memenuhi tujuan awal, tetapi juga memberikan pengalaman yang berharga dalam pengembangan perangkat lunak dan manajemen proyek. Harapan saya aplikasi ini akan memberikan dampak positif dalam mendukung kegiatan belajar-mengajar dan penelitian di lingkungan perpustakaan. Terima kasih kepada semua pihak yang terlibat dalam kesuksesan proyek ini.

DAFTAR PUSTAKA

Link GitHub: