

Laporan Tugas Mata Kuliah Pemrograman Berbasis Obyek : Final Project UAS PBO



Dosen Pembimbing
Bayu Adhi Nugroho, PH.

Disusun Oleh
Putri Roro Andini
09020622037

Tahun Pembelajaran 2022/2023
UIN Sunan Ampel Surabaya
Fakultas Sains dan Teknologi
Sistem Informasi
2023/2024

DAFTAR ISI

DDL	3
DML	5
Login	9
Home/Dashboard	13
Transaksi	21
Inventory	29

DDL

```
BEGIN;

CREATE TABLE IF NOT EXISTS public.buku
(
    kodebuku character varying(4) COLLATE pg_catalog."default" NOT NULL,
    judulbuku character varying(50) COLLATE pg_catalog."default" NOT NULL,
    subjudulbuku character varying(50) COLLATE pg_catalog."default" DEFAULT '-':character varying,
    pengarang character(30) COLLATE pg_catalog."default" NOT NULL,
    penerbit character varying(50) COLLATE pg_catalog."default" NOT NULL,
    tahunterbit integer,
    halaman integer,
    isbn character varying COLLATE pg_catalog."default",
    kategori character varying(25) COLLATE pg_catalog."default",
    CONSTRAINT "Buku_pkey" PRIMARY KEY (kodebuku)
);

CREATE TABLE IF NOT EXISTS public.peminjamanbuku
(
    kodepinjamb character varying(6) COLLATE pg_catalog."default" NOT NULL,
    kodebuku character varying(4) COLLATE pg_catalog."default" NOT NULL,
    nama character varying(20) COLLATE pg_catalog."default",
    nim character varying(14) COLLATE pg_catalog."default",
    angkatan integer,
    tglpinjam date,
    tglbalik date,
    status character varying COLLATE pg_catalog."default",
    CONSTRAINT peminjamanbuku_pkey PRIMARY KEY (kodepinjamb)
);

CREATE TABLE IF NOT EXISTS public.peminjamanskripsi
(
    kodepinjams character varying(6) COLLATE pg_catalog."default" NOT NULL,
    kodeskripsi character varying(4) COLLATE pg_catalog."default" NOT NULL,
    nama character varying(20) COLLATE pg_catalog."default",
    nim character varying(14) COLLATE pg_catalog."default",
    angkatan integer,
    tglpinjam date,
    tglbalik date,
    status character varying COLLATE pg_catalog."default",
    CONSTRAINT peminjamanskripsi_pkey PRIMARY KEY (kodepinjams)
);

CREATE TABLE IF NOT EXISTS public.petugas
(
    username character varying(20) COLLATE pg_catalog."default" NOT NULL,
    fullname character(30) COLLATE pg_catalog."default",
    password character varying(20) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Petugas_pkey" PRIMARY KEY (username)
);
```

```
CREATE TABLE IF NOT EXISTS public.skripsi
(
    kodeskripsi character varying(4) COLLATE pg_catalog."default" NOT NULL,
    judulskripsi character varying(50) COLLATE pg_catalog."default" NOT NULL,
    penulis character(30) COLLATE pg_catalog."default" NOT NULL,
    tahun integer,
    halaman integer,
    CONSTRAINT "Skripsi_pkey" PRIMARY KEY (kodeskripsi)
);
```

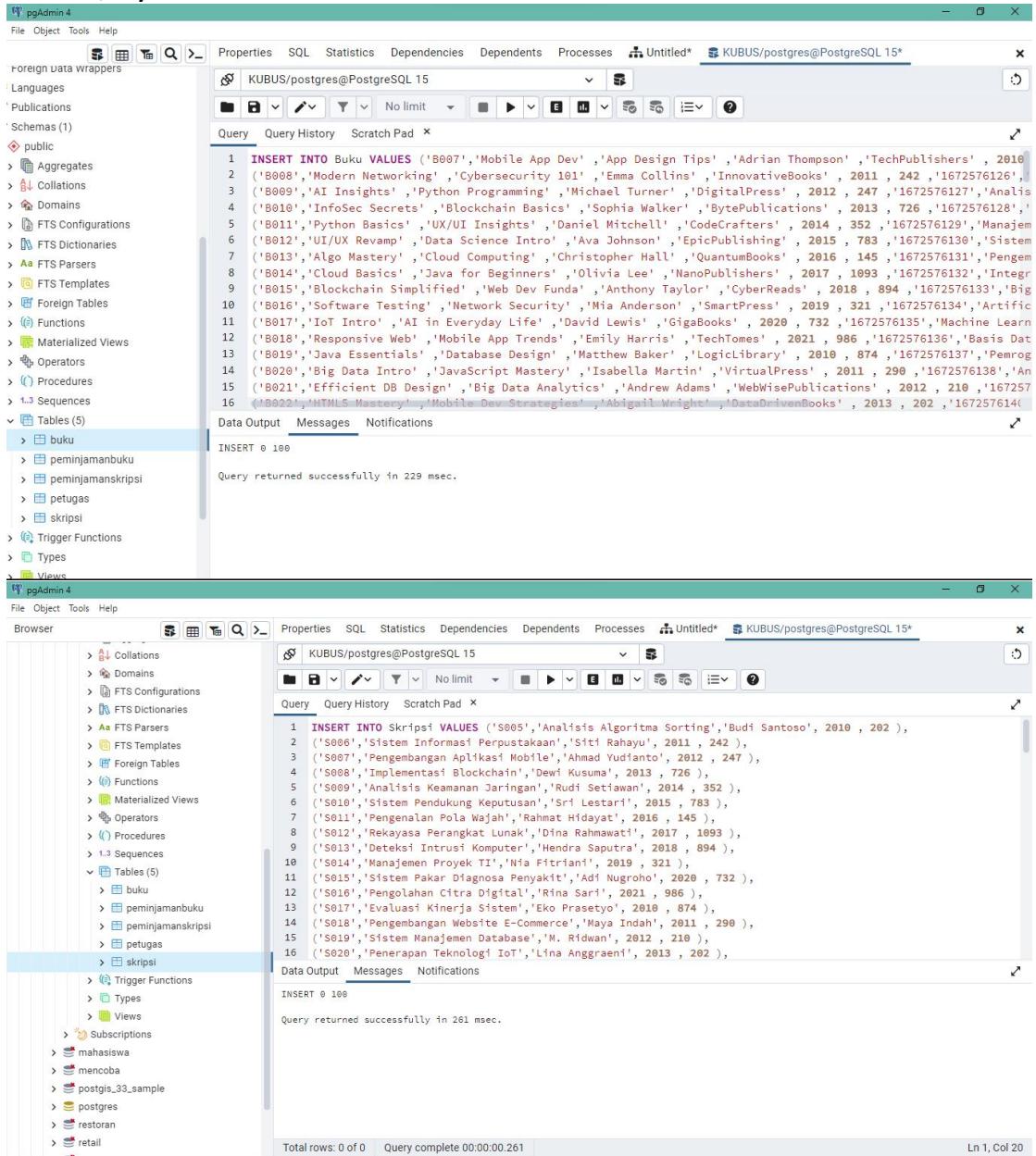
```
ALTER TABLE IF EXISTS public.peminjamanbuku
ADD CONSTRAINT kode_buku FOREIGN KEY (kodebuku)
REFERENCES public.buku (kodebuku) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;
```

```
ALTER TABLE IF EXISTS public.peminjamanskripsi
ADD CONSTRAINT kode_skripsi FOREIGN KEY (kodeskripsi)
REFERENCES public.skripsi (kodeskripsi) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;
```

```
END;
```

DML

INSERT Query:



The screenshot shows two pgAdmin 4 windows side-by-side. Both windows have the title bar "pgAdmin 4" and the tab "KUBUS/postgres@PostgreSQL 15*".

Left Window (buku Table):

```
1 INSERT INTO Buku VALUES ('B007','Mobile App Dev' , 'App Design Tips' , 'Adrian Thompson' , 'TechPublishers' , 2016
2 ('B008','Modern Networking' , 'Cybersecurity 101' , 'Emma Collins' , 'InnovativeBooks' , 2011 , 242 , '1672576126',
3 ('B009','AI Insights' , 'Python Programming' , 'Michael Turner' , 'DigitalPress' , 2012 , 247 , '1672576127','Analisis
4 ('B010','InfoSec Secrets' , 'Blockchain Basics' , 'Sophia Walker' , 'BytePublications' , 2013 , 726 , '1672576128','
5 ('B011','Python Basics' , 'UX/UI Insights' , 'Daniel Mitchell' , 'CodeCrafters' , 2014 , 352 , '1672576129','Manajemen
6 ('B012','UI/UX Revamp' , 'Data Science Intro' , 'Ava Johnson' , 'EpicPublishing' , 2015 , 783 , '1672576130','Sistem
7 ('B013','Algo Mastery' , 'Cloud Computing' , 'Christopher Hall' , 'QuantumBooks' , 2016 , 145 , '1672576131','Pengembangan
8 ('B014','Cloud Basics' , 'Java for Beginners' , 'Olivia Lee' , 'NanоНPublishers' , 2017 , 1093 , '1672576132','Integrasi
9 ('B015','Blockchain Simplified' , 'Web Dev Funda' , 'Anthony Taylor' , 'CyberReads' , 2018 , 894 , '1672576133','Big
10 ('B016','Software Testing' , 'Network Security' , 'Mia Anderson' , 'SmartPress' , 2019 , 321 , '1672576134','Artificial
11 ('B017','IoT Intro' , 'AI in Everyday Life' , 'David Lewis' , 'GigaBooks' , 2020 , 732 , '1672576135','Machine Learning
12 ('B018','Responsive Web' , 'Mobile App Trends' , 'Emily Harris' , 'TechTomes' , 2021 , 986 , '1672576136','Basis Data
13 ('B019','Java Essentials' , 'Database Design' , 'Matthew Baker' , 'LogicLibrary' , 2018 , 874 , '1672576137','Pemrograman
14 ('B020','Big Data Intro' , 'JavaScript Mastery' , 'Isabella Martin' , 'VirtualPress' , 2011 , 290 , '1672576138','Analisis dan Pengembangan
15 ('B021','Efficient DB Design' , 'Big Data Analytics' , 'Andrew Adams' , 'WebWisePublications' , 2012 , 210 , '1672576139','
16 ('B022','HTML5 Mastery' , 'Mobile Dev Strategies' , 'Abigail Wright' , 'DataDrivenBooks' , 2013 , 202 , '1672576140')
```

Right Window (skripsi Table):

```
1 INSERT INTO Skripsi VALUES ('S005','Analisis Algoritma Sorting' , 'Budi Santoso' , 2010 , 202 ),
2 ('S006','Sistem Informasi Perpustakaan' , 'Siti Rahayu' , 2011 , 242 ),
3 ('S007','Pengembangan Aplikasi Mobile' , 'Ahmad Yudianto' , 2012 , 247 ),
4 ('S008','Implementasi Blockchain' , 'Dewi Kusuma' , 2013 , 726 ),
5 ('S009','Analisis Keamanan Jaringan' , 'Rudi Setiawan' , 2014 , 352 ),
6 ('S010','Sistem Pendukung Keputusan' , 'Sri Lestari' , 2015 , 783 ),
7 ('S011','Pengenalan Pola Wajah' , 'Rahmat Hidayat' , 2016 , 145 ),
8 ('S012','Rekayasa Perangkat Lunak' , 'Dina Rahmawati' , 2017 , 1093 ),
9 ('S013','Deteksi Intrusi Komputer' , 'Hendra Saputra' , 2018 , 894 ),
10 ('S014','Manajemen Proyek TI' , 'Nia Fitriani' , 2019 , 321 ),
11 ('S015','Sistem Pakar Diagnosa Penyakit' , 'Adi Nugroho' , 2020 , 732 ),
12 ('S016','Pengolahan Citra Digital' , 'Rina Sari' , 2021 , 986 ),
13 ('S017','Evaluasi Kinerja Sistem' , 'Eko Prasetyo' , 2010 , 874 ),
14 ('S018','Pengembangan Website E-Commerce' , 'Maya Indah' , 2011 , 290 ),
15 ('S019','Sistem Manajemen Database' , 'M. Ridwan' , 2012 , 210 ),
16 ('S020','Penerapan Teknologi IoT' , 'Lina Anggraeni' , 2013 , 202 ),
```

Total rows: 0 of 0 Query complete 00:00:00.261 Ln 1, Col 20

pgAdmin 4

File Object Tools Help

Browser Properties SQL Statistics Dependencies Dependents Processes Untitled* KUBUS/postgres@PostgreSQL 15*

Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (5) buku peminjamanbuku peminjamanskripsi petugas skripsi Trigger Functions Types Views Subscriptions mahasiswa mencoba postgres_33_sample restoran retail

Query History Scratch Pad

```
1 INSERT INTO Peminjamanbuku VALUES ('PB010','B001','Budi Santoso','0908093845', 2010 , '02/10/2023','15/10/2023', 'Kembal1'),  
2 ('PB011','B002','Siti Rahayu','0908093846', 2011 , '03/10/2023','16/10/2023','Kembal1'),  
3 ('PB012','B003','Ahmad Yudianto','0908093847', 2012 , '04/10/2023','17/10/2023','Kembal1'),  
4 ('PB013','B004','Dewi Kusuma','0908093848', 2013 , '05/10/2023','18/10/2023','Kembal1'),  
5 ('PB014','B005','Rudi Setiawan','0908093849', 2014 , '06/10/2023','19/10/2023','Kembal1'),  
6 ('PB015','B006','Sri Lestari','0908093850', 2015 , '07/10/2023','20/10/2023','Kembal1'),  
7 ('PB016','B007','Rahmat Hidayat','0908093851', 2016 , '08/10/2023','21/10/2023','Kembal1'),  
8 ('PB017','B008','Dina Rahmawati','0908093852', 2017 , '09/10/2023','22/10/2023','Kembal1'),  
9 ('PB018','B009','Hendra Saputra','0908093853', 2018 , '10/10/2023','23/10/2023','Kembal1'),  
10 ('PB019','B010','Nia Fitrian','0908093854', 2019 , '11/10/2023','24/10/2023','Kembal1'),  
11 ('PB020','B011','Adi Nugroho','0908093855', 2020 , '12/10/2023','25/10/2023','Kembal1'),  
12 ('PB021','B012','Rina Sarl','0908093856', 2021 , '13/10/2023','26/10/2023','Kembal1'),  
13 ('PB022','B013','Eko Prasetyo','0908093857', 2018 , '14/10/2023','27/10/2023','Kembal1'),  
14 ('PB023','B014','Maya Indah','0908093858', 2011 , '15/10/2023','28/10/2023','Kembal1'),  
15 ('PB024','B015','M. Ridwan','0908093859', 2012 , '16/10/2023','29/10/2023','Kembal1'),  
16 ('PB025','B016','Lina Anggiani','0908093860', 2013 , '17/10/2023','30/10/2023','Kembal1'),
```

Data Output Messages Notifications

INSERT 0 68

Query returned successfully in 347 msec.

Total rows: 0 of 0 Query complete 00:00:00.347 Ln 3, Col 40

pgAdmin 4

File Object Tools Help

Browser Properties SQL Statistics Dependencies Dependents Processes Untitled* KUBUS/postgres@PostgreSQL 15*

Query History Scratch Pad

```

1 INSERT INTO Peminjamanbuku (kodepinjamb, kodebuku, nama, nim, angkatan, tglpinjam, status) VALUES ('PB078','B001'
2 ('PB079','B010','Fahmi Setiawan','0908093914', 2019 , '19/10/2023','Dipinjam'),
3 ('PB080','B011','Rita Susanti','0908093915', 2020 , '20/10/2023','Dipinjam'),
4 ('PB081','B012','Achmad Iqbal','0908093916', 2021 , '21/10/2023','Dipinjam'),
5 ('PB082','B013','Desi Rosmalinda','0908093917', 2018 , '22/10/2023','Dipinjam'),
6 ('PB083','B014','Muhammad Haris','0908093918', 2011 , '23/10/2023','Dipinjam'),
7 ('PB084','B015','Rini Kurniati','0908093919', 2012 , '24/10/2023','Dipinjam'),
8 ('PB085','B016','Agung Prasetya','0908093920', 2013 , '25/10/2023','Dipinjam'),
9 ('PB086','B017','Sari Ramadhan','0908093921', 2014 , '26/10/2023','Dipinjam'),
10 ('PB087','B018','Andre Setiawan','0908093922', 2015 , '27/10/2023','Dipinjam'),
11 ('PB088','B019','Intan Permatasari','0908093923', 2016 , '28/10/2023','Dipinjam'),
12 ('PB089','B020','Ridwan Ramadhan','0908093924', 2017 , '29/10/2023','Dipinjam'),
13 ('PB090','B021','Laila Nurjannah','0908093925', 2018 , '30/10/2023','Dipinjam'),
14 ('PB091','B022','Eko Prasetya','0908093926', 2019 , '31/10/2023','Dipinjam'),
15 ('PB092','B023','Rina Sulistiani','0908093927', 2020 , '01/11/2023','Dipinjam'),
16 ('PB093','B024','Rahmat Hidayat','0908093928', 2021 , '02/11/2023','Dipinjam'),
```

Data Output Messages Notifications

INSERT 0 32

Query returned successfully in 201 msec.

Total rows: 0 of 0 Query complete 00:00:00.201

✓ Query returned successfully in 201 msec. X

Ln 1, Col 50

pgAdmin 4

File Object Tools Help

Browser Properties SQL Statistics Dependencies Dependents Processes Untitled* KUBUS/postgres@PostgreSQL 15*

Query History Scratch Pad

```

1 INSERT INTO Peminjamanskripsi (kodepinjams, kodeskripsi, nama, nim, angkatan, tglpinjam, status) VALUES
2 ('PS076','S033','Dewi Wahyuni','0908093913', 2018 , '18/10/2023','Dipinjam'),
3 ('PS077','S034','Fahmi Setiawan','0908093914', 2019 , '19/10/2023','Dipinjam'),
4 ('PS078','S035','Rita Susanti','0908093915', 2020 , '20/10/2023','Dipinjam'),
5 ('PS079','S036','Achmad Iqbal','0908093916', 2021 , '21/10/2023','Dipinjam'),
6 ('PS080','S061','Desi Rosmalinda','0908093917', 2018 , '22/10/2023','Dipinjam'),
7 ('PS081','S062','Muhammad Haris','0908093918', 2011 , '23/10/2023','Dipinjam'),
8 ('PS082','S063','Rini Kurniati','0908093919', 2012 , '24/10/2023','Dipinjam'),
9 ('PS083','S064','Agung Prasetya','0908093920', 2013 , '25/10/2023','Dipinjam'),
10 ('PS084','S065','Sari Ramadhan','0908093921', 2014 , '26/10/2023','Dipinjam'),
11 ('PS085','S066','Andre Setiawan','0908093922', 2015 , '27/10/2023','Dipinjam'),
12 ('PS086','S067','Intan Permatasari','0908093923', 2016 , '28/10/2023','Dipinjam'),
13 ('PS087','S068','Ridwan Ramadhan','0908093924', 2017 , '29/10/2023','Dipinjam'),
14 ('PS088','S069','Laila Nurjannah','0908093925', 2018 , '30/10/2023','Dipinjam'),
15 ('PS089','S010','Eko Prasetya','0908093926', 2019 , '31/10/2023','Dipinjam'),
16 ('PS090','S011','Rina Sulistiani','0908093927', 2020 , '01/11/2023','Dipinjam'),
```

Data Output Messages Notifications

INSERT 0 32

Query returned successfully in 240 msec.

Total rows: 0 of 0 Query complete 00:00:00.240

Ln 2, Col 1

pgAdmin 4

File Object Tools Help

Browser Properties SQL Statistics Dependencies Dependents Processes Untitled* KUBUS/postgres@PostgreSQL 15*

Query History Scratch Pad

```
1 INSERT INTO Peminjamanskripsi VALUES
2 ('PS008','S001','Budi Santoso','0908093845', 2010 , '02/10/2023', '15/10/2023', 'Kembali'),
3 ('PS009','S002','Siti Rahayu','0908093846', 2011 , '03/10/2023', '16/10/2023', 'Kembali'),
4 ('PS010','S003','Ahmad Yudianto','0908093847', 2012 , '04/10/2023', '17/10/2023', 'Kembali'),
5 ('PS011','S004','Dewi Kusuma','0908093848', 2013 , '05/10/2023', '18/10/2023', 'Kembali'),
6 ('PS012','S005','Rudi Setiawan','0908093849', 2014 , '06/10/2023', '19/10/2023', 'Kembali'),
7 ('PS013','S006','Sri Lestari','0908093850', 2015 , '07/10/2023', '20/10/2023', 'Kembali'),
8 ('PS014','S007','Rahmat Hidayat','0908093851', 2016 , '08/10/2023', '21/10/2023', 'Kembali'),
9 ('PS015','S008','Dina Rahmawati','0908093852', 2017 , '09/10/2023', '22/10/2023', 'Kembali'),
10 ('PS016','S009','Hendra Saputra','0908093853', 2018 , '10/10/2023', '23/10/2023', 'Kembali'),
11 ('PS017','S010','Nita Fitrian'i','0908093854', 2019 , '11/10/2023', '24/10/2023', 'Kembali'),
12 ('PS018','S011','Adi Nugroho','0908093855', 2020 , '12/10/2023', '25/10/2023', 'Kembali'),
13 ('PS019','S012','Rina Sari','0908093856', 2021 , '13/10/2023', '26/10/2023', 'Kembali'),
14 ('PS020','S013','Eko Prasetyo','0908093857', 2020 , '14/10/2023', '27/10/2023', 'Kembali'),
15 ('PS021','S014','Maya Indah','0908093858', 2021 , '15/10/2023', '28/10/2023', 'Kembali'),
16 ('PS022','S015','M. Ridwan','0908093859', 2022 , '16/10/2023', '29/10/2023', 'Kembali'),
```

Data Output Messages Notifications

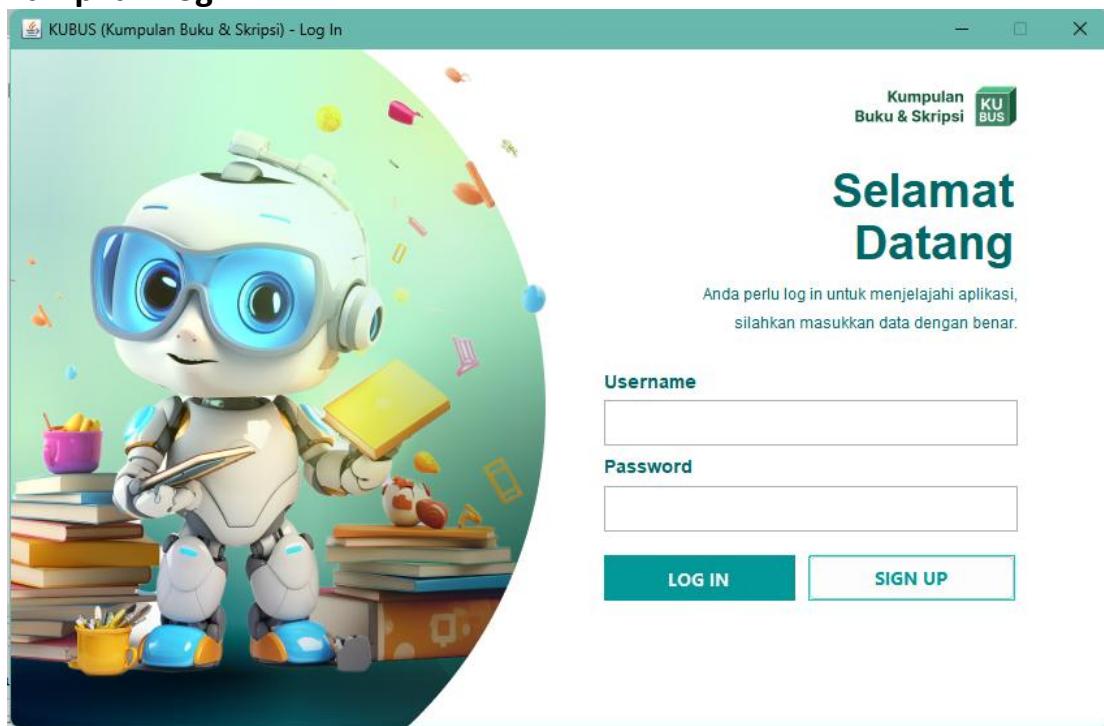
INSERT 0 68

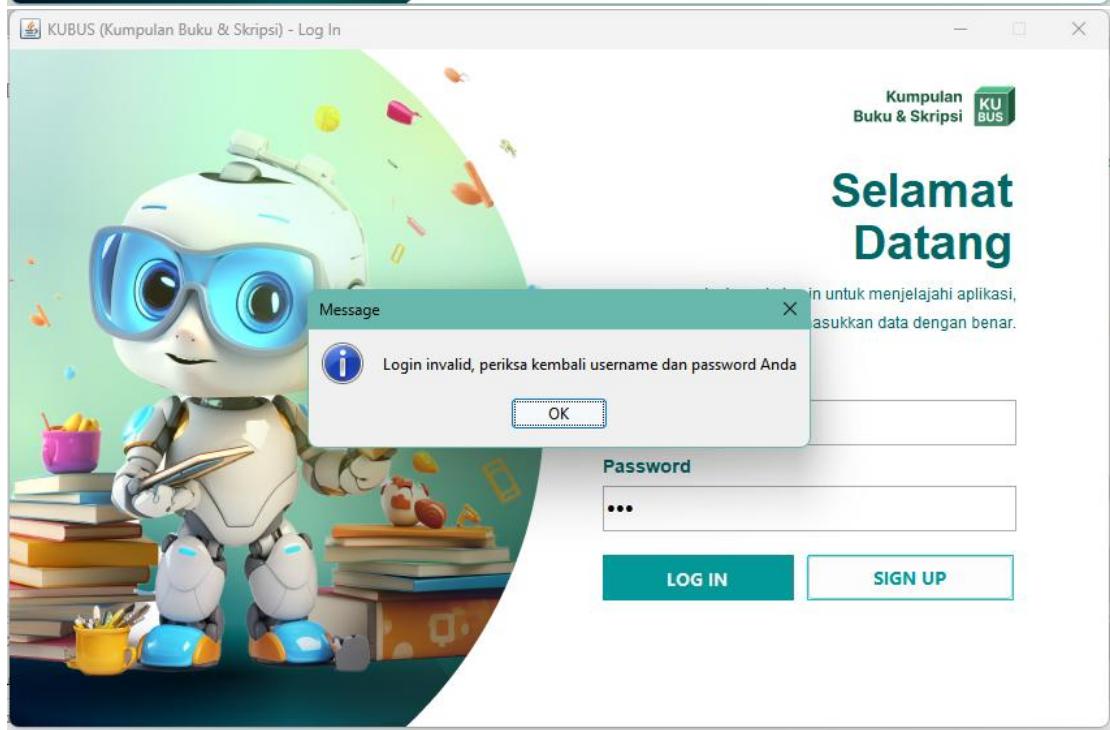
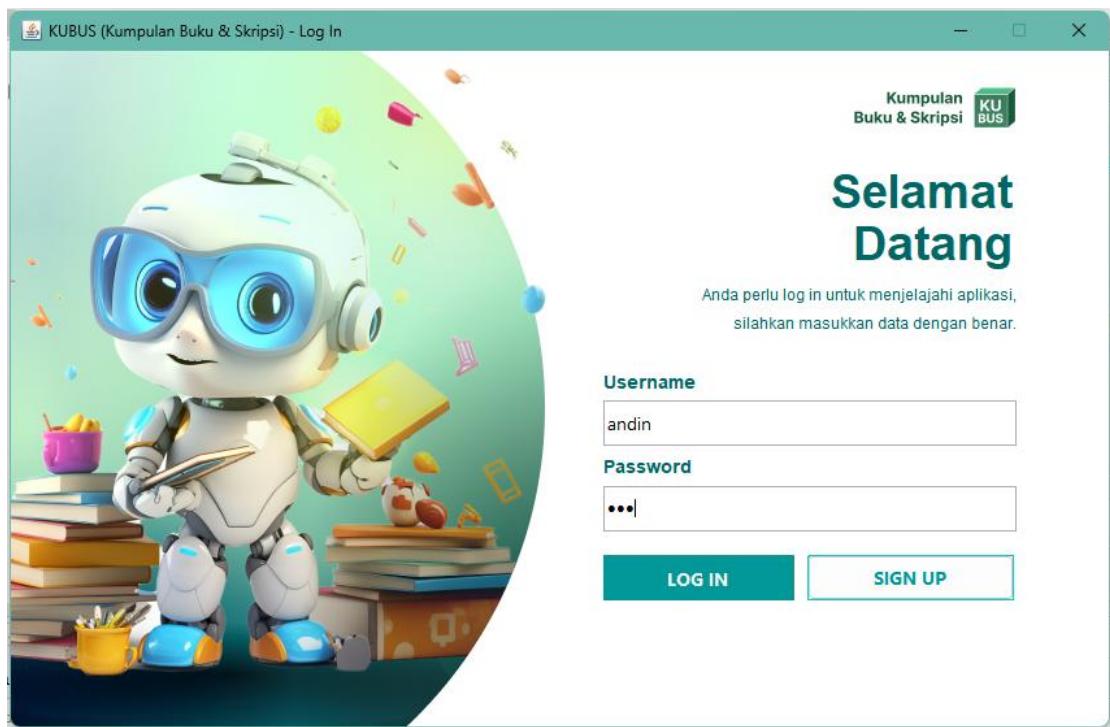
Query returned successfully in 203 msec.

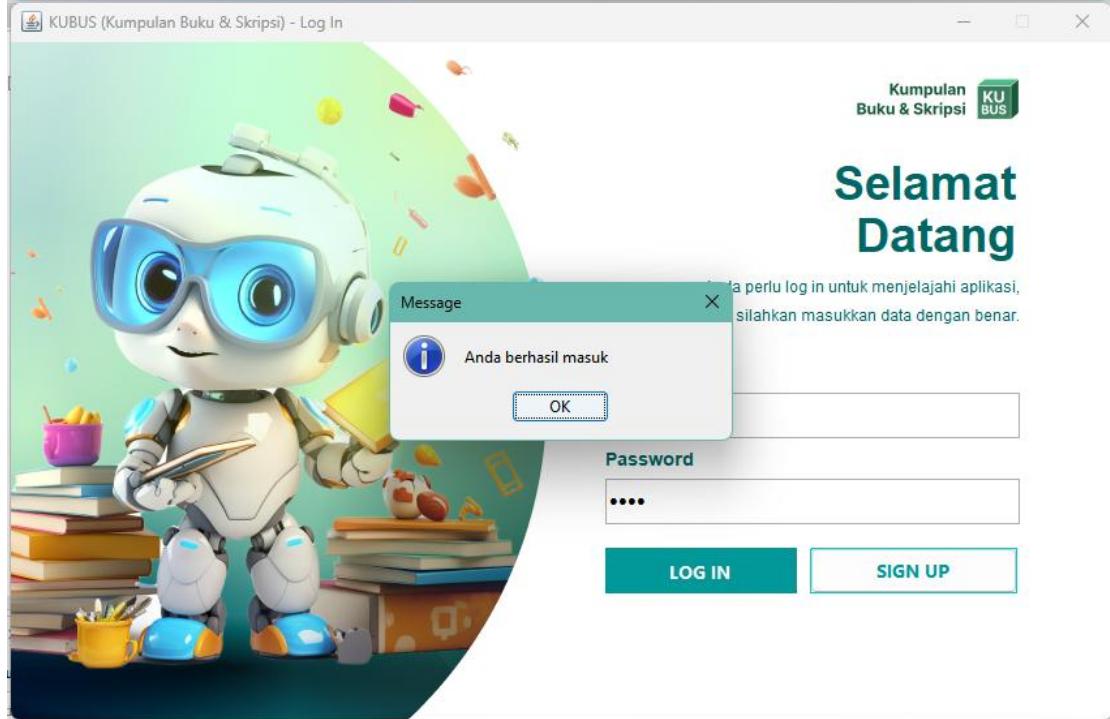
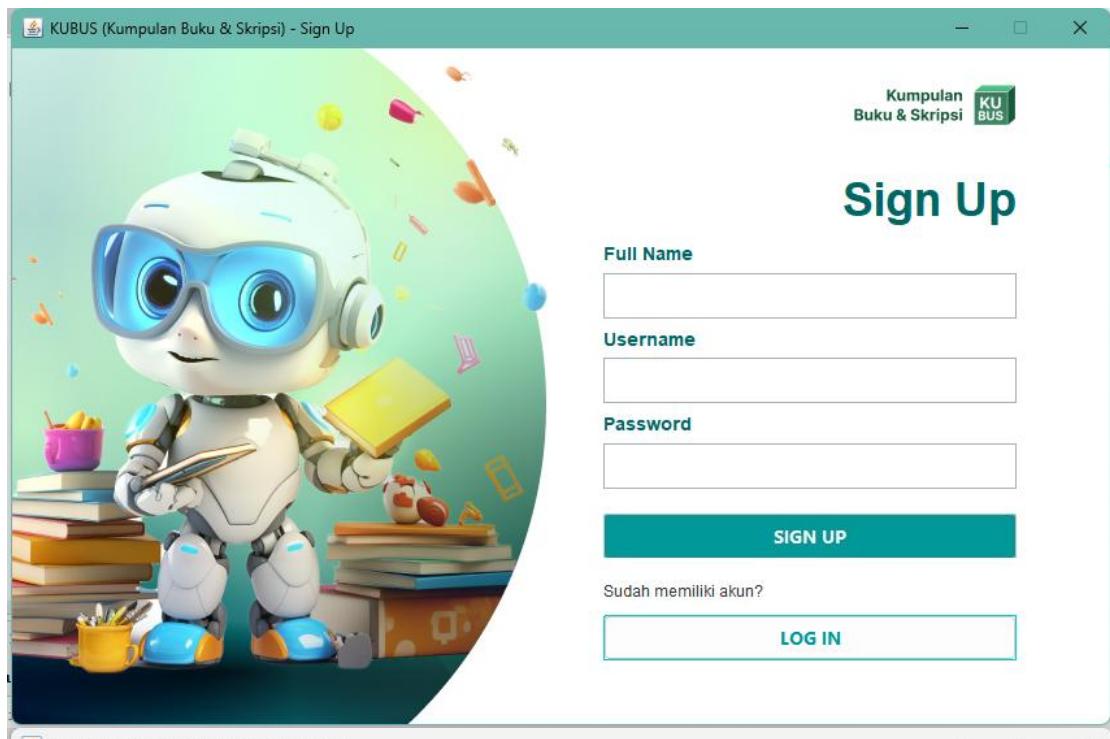
Total rows: 0 of 0 Query complete 00:00:00.203 Ln 1, Col 38

Login

Tampilan Login





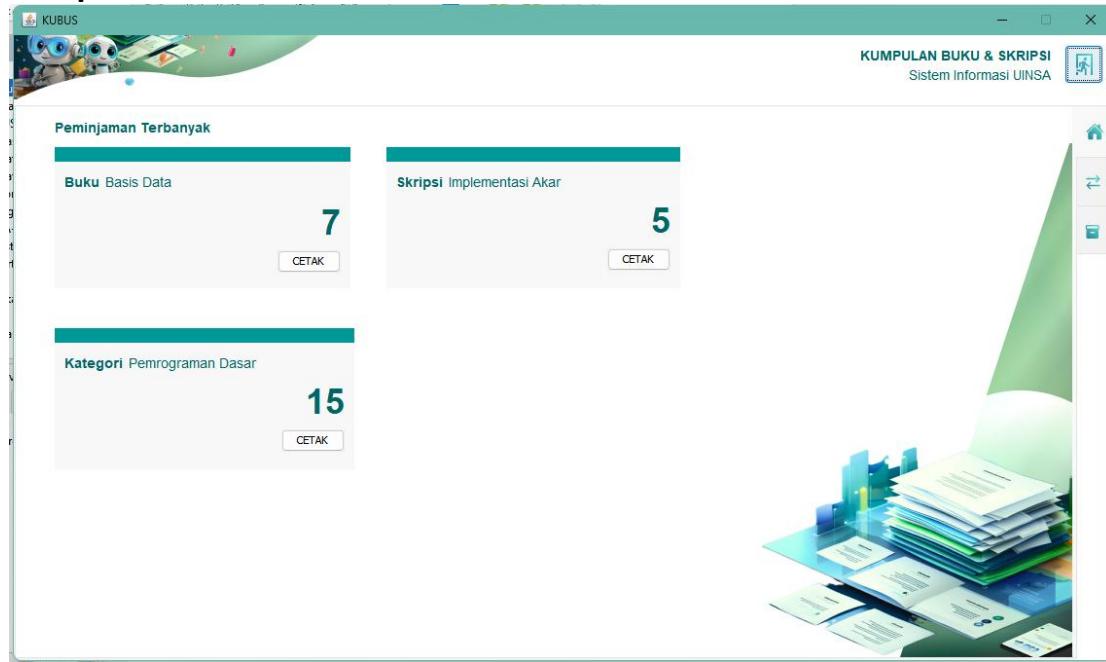


Sourcecode Login:

```
private void jButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String username = jTextFieldUsername.getText();  
    String password = new String(value:jPasswordField.getPassword());  
  
    EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "KumpulanBukuSkripsiPU");  
    EntityManager em = emf.createEntityManager();  
  
    try {  
        em.getTransaction().begin();  
  
        // Assuming you have a User entity with attributes 'username' and 'password'  
        Query q = em.createQuery(string: "SELECT p FROM Petugas p WHERE p.username = :username AND p.password = :password");  
        q.setParameter(string: "username", :username);  
        q.setParameter(string: "password", :password);  
  
        if (!q.getResultList().isEmpty()) {  
            JOptionPane.showMessageDialog(parentComponent: this, message: "Anda berhasil masuk");  
  
            new Dashboard().setVisible(b: true);  
            this.dispose();  
  
            kosongkan_form();  
        } else {  
            JOptionPane.showMessageDialog(parentComponent: this, message: "Login invalid, periksa kembali username dan password Anda");  
  
            kosongkan_form();  
        }  
  
        em.getTransaction().commit();  
    } finally {  
        em.close();  
        emf.close();  
    }  
}
```

Home/Dashboard

Tampilan dashboard



Source code Home/Dashboard:

```
private void totalbukukategori() {
    EntityManagerFactory emf = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU");
    EntityManager em = emf.createEntityManager();

    em.getTransaction().begin();

    Query query = em.createNativeQuery(
        "SELECT COUNT(pb.kodepinjamb) AS jumlah_peminjaman FROM Buku b LEFT JOIN Peminjamanbuku"
        +"(pb ON b.kodebuku = pb.kodebuku GROUP BY b.kategori ORDER BY jumlah_peminjaman DESC LIMIT 1");
    Long totalktg = (Long) query.getSingleResult();
    Integer totalktgint = totalktg.intValue();

    em.getTransaction().commit();

    em.close();
    emf.close();

    lbl_kategoriterbanyak.setText(String.valueOf(obj:totalktgint));
}

private void totalbukuterbanyak() {
    EntityManagerFactory emf = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU");
    EntityManager em = emf.createEntityManager();

    em.getTransaction().begin();

    Query query = em.createNativeQuery("SELECT COUNT(pb.kodepinjamb) AS jumlah_peminjaman FROM Buku b "
        +"LEFT JOIN Peminjamanbuku pb ON b.kodebuku = pb.kodebuku GROUP BY b.kodebuku ORDER BY jumlah_peminjaman DESC LIMIT 1");
    Long totalBuku = (Long) query.getSingleResult();
    Integer totalBukuInt = totalBuku.intValue();
    em.getTransaction().commit();

    em.close();
    emf.close();

    lbl_bukuterbanyak.setText(String.valueOf(obj:totalBukuInt));
}
```

```

    private void totalskripsibanyak() {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName: \"KumpulanBukuSkripsiPU\"");
        EntityManager em = emf.createEntityManager();

        em.getTransaction().begin();

        Query query = em.createNativeQuery("SELECT COUNT(ps.kodepinjams) AS jumlah_peminjaman FROM Skripsi s"
        +" LEFT JOIN Peminjamanskripsi ps ON s.kodeskripsi = ps.kodeskripsi GROUP BY s.kodeskripsi ORDER BY jumlah_peminjaman DESC LIMIT 1");
        Long totalSkripsi = (Long) query.getSingleResult();
        Integer totalSkripsiInt = totalSkripsi.intValue();

        em.getTransaction().commit();

        em.close();
        emf.close();

        lbl_skripsiterbanyak.setText(text: String.valueOf(obj: totalSkripsiInt));
    }

    private void judulkategoribanyak() {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName: \"KumpulanBukuSkripsiPU\"");
        EntityManager em = emf.createEntityManager();

        em.getTransaction().begin();

        Query query = em.createNativeQuery("SELECT b.kategori FROM Buku b LEFT JOIN Peminjamankategori pb "
        +" ON b.kodebuku = pb.kodebuku GROUP BY b.kategori ORDER BY COUNT(pb.kodepinjamb) DESC LIMIT 1");
        String judulktg = (String) query.getSingleResult();

        em.getTransaction().commit();

        em.close();
        emf.close();

        jLabeljudulkategori.setText(text: String.valueOf(obj: judulktg));
    }

    private void judulskripsibanyak() {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName: \"KumpulanBukuSkripsiPU\"");
        EntityManager em = emf.createEntityManager();

        em.getTransaction().begin();

        Query query = em.createNativeQuery("SELECT s.judulskripsi FROM Skripsi s LEFT JOIN Peminjamanskripsi ps"
        +" ON s.kodeskripsi = ps.kodeskripsi GROUP BY s.judulskripsi ORDER BY COUNT(ps.kodepinjams) DESC LIMIT 1");
        String judulskripsi = (String) query.getSingleResult();

        em.getTransaction().commit();

        em.close();
        emf.close();

        jLabeljudulskripsi.setText(text: String.valueOf(obj: judulskripsi));
    }

    private void judulbukubanyak() {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName: \"KumpulanBukuSkripsiPU\"");
        EntityManager em = emf.createEntityManager();

        em.getTransaction().begin();

        Query query = em.createNativeQuery("SELECT b.judulbuku FROM Buku b LEFT JOIN Peminjamankategori pb ON b.kodebuku = pb.kodebuku "
        +" GROUP BY b.judulbuku ORDER BY COUNT(pb.kodepinjamb) DESC LIMIT 1");
        String judulbuku = (String) query.getSingleResult();

        em.getTransaction().commit();

        em.close();
        emf.close();

        jLabeljudulbuku.setText(text: String.valueOf(obj: judulbuku));
    }
}

```

Tampilan cetak ditekan pada bagian peminjaman terbanyak: Buku

JasperViewer

KUBUS

Bu Pa 3US Da Dat Dat For Sig TA sist or t cka s ora ian

Pem Bu Kat

LAPORAN BUKU PEMINJAMAN TERBANYAK PROGRAM STUDI SISTEM
UIN SUNAN AMPEL SURABAYA

Tanggal cetak: Tuesday 05 December 2023

KODE BUKU	JUDUL BUKU	JUMLAH PEMINJAMAN
B002	Basis Data	7
B003	Fun Math	6
B001	Indonesia Hari ini	6
B004	Mengarang Bebas	5
B005	War Of AI	5
B013	Algo Mastery	4
B006	Cloud Computing	4
B008	Modern Networking	4
B009	AI Insights	4
B010	InfoSec Secrets	4
B015	Blockchain Simplified	4
B012	UI/UX Revamp	4
B011	Python Basics	4
B007	Mobile App Dev	4
B014	Cloud Basics	4
B016	Software Testing	4

Page 1 of 2

KumpulanRukuSkrinici (run) #2 running (1 more) 1018-1 INK

sourcecode cetak buku terbanyak:

```
private void jButtonCetakBukubanyakActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String reportPath = "src/Report/reportBukuTerbanyak.jrxml";

        EntityManagerFactory emf = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiP");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();

        CriteriaBuilder cb = em.getCriteriaBuilder();
        CriteriaQuery<Object[]> cq = cb.createQuery(type: Object[].class);
        Root<Peminjamanbuku> peminjamanRoot = cq.from(type: Peminjamanbuku.class);
        Join<Peminjamanbuku, Buku> bukuJoin = peminjamanRoot.join(string: "kodebuku", jt: JoinType.INNER);

        cq.multiselect(
            selectns:bukuJoin.get(string: "kodebuku"),
            selectns:bukuJoin.get(string: "judulbuku"),
            selectns:cb.count(exprsn: peminjamanRoot.get(string: "kodepinjamb")).alias(string: "jumlahpeminjaman")
        );

        cq.groupBy(exprsns: bukuJoin.get(string: "kodebuku"), exprsns: bukuJoin.get(string: "judulbuku"));
        cq.orderBy(order: cb.desc(exprsn: cb.count(exprsn: peminjamanRoot.get(string: "kodepinjamb"))));

        TypedQuery<Object[]> query = em.createQuery(cq);
        List<Object[]> results = query.getResultList();
        em.getTransaction().commit();

        em.close();
        emf.close();

        // Mapping hasil query ke dalam parameter JasperReport
        List<Map<String, Object>> mappedResults = new ArrayList<>();
        for (Object[] result : results) {
            Map<String, Object> map = new HashMap<>();

            // Mapping hasil query ke dalam parameter JasperReport
            List<Map<String, Object>> mappedResults = new ArrayList<>();
            for (Object[] result : results) {
                Map<String, Object> map = new HashMap<>();
                String kodebuku = (String) result[0];
                String judulbuku = (String) result[1];
                Long jumlah_peminjaman = (Long) result[2];

                map.put(key: "kodebuku", value: kodebuku);
                map.put(key: "judulbuku", value: judulbuku);
                map.put(key: "jumlah_peminjaman", value: jumlah_peminjaman);
                mappedResults.add(e: map);
            }
        }

        // Creating a JasperReports data source from the mapped results
        JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: mappedResults);

        // Loading the report design file (*.jrxml)
        JasperReport jasperReport = JasperCompileManager.compileReport(sourceFileName: reportPath);
        JasperPrint print = JasperFillManager.fillReport(jasperReport, parameters: null, dataSource);
        JasperViewer viewer = new JasperViewer(jasperPrint: print, isExitOnClose: false);
        viewer.setVisible(b: true);

    } catch (JRException ex) {
        Logger.getLogger(name: Peminjamanbuku.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

Tampilan cetak ditekan pada bagian peminjaman terbanyak: Skripsi

S Buku1 JasperViewer KUBUS 14 1 100% X

LAPORAN PEMINJAMAN SKRIPSI TERBANYAK PROGRAM STUDI SISTEM
UIN SUNAN AMPEL SURABAYA

Tanggal cetak: Tuesday 05 December 2023

KODE SKRIPSI	JUDUL SKRIPSI	JUMLAH PEMINJAMAN
S002	Koordinat Tabung Menggunakan Paralon	5
S001	Implementasi Akar	5
S003	Mengungkap Identitas Angin Utara	4
S004	Implementasi Aplikasi Odoo Pada Rumah Makan Padang	4
S008	Implementasi Blockchain	3
S009	Analisis Keamanan Jaringan	3
S010	Sistem Pendukung Keputusan	3
S011	Pengenalan Pola Wajah	3
S019	Sistem Manajemen Database	3
S006	Sistem Informasi Perpustakaan	3
S016	Pengolahan Citra Digital	3
S028	Desain Antarmuka Pengguna	3
S022	Optimasi Algoritma	3
S013	Deteksi Intrusi Komputer	3
S023	Keamanan Aplikasi Web	3

Page 1 of 2

Sourcecode skripsi terbanyak:

```
private void jButtonCetakSkripsiBanyakActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String reportPath = "src/Report/reportSkripsiTerbanyak.jrxml";

        EntityManagerFactory emf = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();

        CriteriaBuilder cb = em.getCriteriaBuilder();
        CriteriaQuery<Object[]> cq = cb.createQuery(type: Object[].class);
        Root<Peminjamanskripsi> peminjamanRootl = cq.from(type: Peminjamanskripsi.class);
        Join<Peminjamanskripsi, Skripsi> skripsiJoin = peminjamanRootl.join(string: "kodeskripsi", jt: JoinType.INNER);

        cq.multiselect(
            select: skripsiJoin.get(string: "kodeskripsi"),
            select: skripsiJoin.get(string: "judulskripsi"),
            select: cb.count(expr: peminjamanRootl.get(string: "kodepinjams")).alias(string: "jumlahpeminjaman")
        );

        cq.groupBy(exprs: skripsiJoin.get(string: "kodeskripsi"), exprs: skripsiJoin.get(string: "judulskripsi"));
        cq.orderBy(orders: cb.desc(exprs: cb.count(exprs: peminjamanRootl.get(string: "kodepinjams"))));

        TypedQuery<Object[]> query = em.createQuery(cq);
        List<Object[]> results = query.getResultList();
        em.getTransaction().commit();

        em.close();
        emf.close();

        // Mapping hasil query ke dalam parameter JasperReport
        List<Map<String, Object>> mappedResults = new ArrayList<>();
        for (Object[] result : results) {
            Map<String, Object> map = new HashMap<>();
            String kodeskripsi = (String) result[0];
            String judulskripsi = (String) result[1];
            Long jumlah_peminjaman = (Long) result[2];

            map.put(key: "kodeskripsi", value: kodeskripsi);
            map.put(key: "judulskripsi", value: judulskripsi);
            map.put(key: "jumlah_peminjaman", value: jumlah_peminjaman);
            mappedResults.add(e: map);
        }

        // Creating a JasperReports data source from the mapped results
        JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: mappedResults);

        // Loading the report design file (*.jrxml)
        JasperReport jasperReport = JasperCompileManager.compileReport(sourceFileName: reportPath);
        JasperPrint print = JasperFillManager.fillReport(jasperReport, parameters: null, dataSource);
        JasperViewer viewer = new JasperViewer(jasperPrint: print, isExitOnClose: false);
        viewer.setVisible(b: true);

    } catch (JRException ex) {
        Logger.getLogger(name: Peminjamanbuku.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

Tampilan cetak ditekan pada bagian peminjaman terbanyak: kategori

JasperViewer

KUBUS

LAPORAN PEMINJAMAN KATEGORI TERBANYAK PROGRAM STUDI SISTEM
UIN SUNAN AMPEL SURABAYA

Tanggal cetak: Tuesday 05 December 2023

NAMA KATEGORI	JUMLAH PEMINJAMAN
Keamanan Informasi	14
Analisis Desain Sistem	8
Integrasi Sistem	9
Sistem Operasi	14
Pemrograman Dasar	15
Machine Learning	4
Manajemen Proyek TI	8
Big Data dan Analitika	5
Pengembangan Aplikasi	13
Artificial Intelligence	11
Basis Data	8

Page 1 of 1

Sourcecode cetak kategori terbanyak:

```
private void jButtonCetakKategoriBanyakActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String reportPath = "src/Report/reportKategoriTerbanyak.jrxml";

        EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName: \"KumpulanBukuSkripsiPU\"");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();

        CriteriaBuilder cb = em.getCriteriaBuilder();
        CriteriaQuery<Object[]> cq = cb.createQuery(type: Object[].class);
        Root<Peminjamanbuku> peminjamanRoot2 = cq.from(type: Peminjamanbuku.class);
        Join<Peminjamanbuku, Buku> kategoriJoin = peminjamanRoot2.join(string: "kodebuku", jt: JoinType.INNER);
        //Join<Peminjamanbuku, Buku> kategoriJoin = kategoriJoin.join("kodepinjamb", JoinType.INNER);

        cq.multiselect(
            select: kategoriJoin.get(string: "kategori"),
            select: cb.count(expr: peminjamanRoot2.get(string: "kodepinjamb")).alias(string: "jumlah_peminjaman")
        );

        cq.groupBy(exprs: kategoriJoin.get(string: "kategori"));
        //cq.orderBy(cb.desc(cb.count(peminjamanRoot2.get("kodepinjamb"))));

        TypedQuery<Object[]> query = em.createQuery(cq);
        List<Object[]> results = query.getResultList();
        em.getTransaction().commit();

        em.close();
        emf.close();

        // Mapping hasil query ke dalam parameter JasperReport
        List<Map<String, Object>> mappedResults = new ArrayList<>();
        for (Object[] result : results) {
            Map<String, Object> map = new HashMap<>();
            String kategori = (String) result[0];
            Long jumlahPeminjaman = (Long) result[1];

            map.put(key: "kategori", value: kategori);
            map.put(key: "jumlah_peminjaman", value: jumlahPeminjaman);
            mappedResults.add(e: map);
        }

        // Creating a JasperReports data source from the mapped results
        JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: mappedResults);

        // Loading the report design file (*.jrxml)
        JasperReport jasperReport = JasperCompileManager.compileReport(sourceFileName: reportPath);
        JasperPrint print = JasperFillManager.fillReport(jasperReport, parameters: null, dataSource);
        JasperViewer viewer = new JasperViewer(jasperPrint: print, isExitOnClose: false);
        viewer.setVisible(b: true);

    } catch (JRException ex) {
        Logger.getLogger(name: Peminjamanbuku.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

Transaksi

Peminjaman Buku

Tampilan Peminjaman Buku:

The screenshot shows a Java Swing application window titled 'KUBUS'. The main title bar says 'KUMPULAN BUKU & SKRIPSI Sistem Informasi UINSA'. The window contains a table of borrowing records and several input fields and buttons for searching and filtering.

KODE PINJAM	KODE BUKU	NAMA	NIM	ANGKATAN	TANGGAL PINJAM	TANGGAL KEMBALI	STATUS
PB003	B001	Putri Roro Andini	090233202409	2022	Tue Nov 14 00:00:00...	Tue Nov 21 00:00:00...	Kembali
PB004	B002	Raden Roro	090230283394	2022	Tue Nov 14 00:00:00...	Tue Nov 21 00:00:00...	Kembali
PB005	B002	Raden	090230283394	2022	Wed Nov 01 00:00:00...	Wed Nov 08 00:00:00...	Kembali
PB006	B002	Nia	090245922903	2020	Wed Nov 01 00:00:00...	Sun Nov 12 00:00:00...	Kembali
PB007	B003	Ratna	0920930246048	2021	Thu Nov 16 00:00:00...	Tue Nov 21 00:00:00...	Kembali
PB008	B005	Arum	0902062286876	2022	Tue Nov 21 00:00:00...	Tue Nov 28 00:00:00...	Kembali
PB002	B001	Putri Roro	090206222037	2022	Tue Nov 28 00:00:00...	Sun Dec 24 00:00:00...	Kembali
PB001	B003	Selon	090203040506	2023	Fri Dec 01 00:00:00...	Sat Dec 02 00:00:00...	Kembali
PB009	B004	Limas	0902062286876	2022	Tue Nov 21 00:00:00...	Sat Dec 02 00:00:00...	Kembali
PB010	B001	Budi Santoso	0908093845	2010	Mon Oct 02 00:00:00...	Sun Oct 15 00:00:00...	Kembali
PB011	B002	Siti Rahayu	0908093846	2011	Tue Oct 03 00:00:00...	Mon Oct 16 00:00:00...	Kembali
PB012	B003	Ahmad Yudianto	0908093847	2012	Wed Oct 04 00:00:00...	Tue Oct 17 00:00:00...	Kembali
PB013	B004	Dewi Kusuma	0908093848	2013	Thu Oct 05 00:00:00...	Wed Oct 18 00:00:00...	Kembali
PB014	B005	Rudi Setiawan	0908093849	2014	Fri Oct 06 00:00:00...	Thu Oct 19 00:00:00...	Kembali

Below the table are search/filter fields:

- Kode Peminjaman (dropdown)
- Nama Lengkap (text field)
- Kode Buku (dropdown)
- NIM (text field)
- Tanggal Pinjam (date picker)
- Tanggal Kembali (date picker)
- Tahun Angkatan (text field)
- Status Peminjaman (dropdown)
- Cetak Laporan (button)
- Tanggal Awal (text field)
- Tanggal Akhir (text field)
- CETAK (button)

At the bottom are buttons: TAMBAH, UBAH, and HAPUS.

Source code tambah:

```
private void jButtonTambahPinjamBukuActionPerformed(java.awt.event.ActionEvent evt) {
    String kodepinjamb = jTextFieldKodepinjambuku.getText().trim();
    String namab = jTextFieldnamab.getText();
    String nimb = jTextFieldnimb.getText();
    int angkatanb = Integer.parseInt(jTextFieldangkatanb.getText());
    Date tglpinjamb = jDateChoosertpb.getDate();
    Date tglbalikb = jDateChoosertkb.getDate();
    String status = jComboBoxspb.getSelectedItem().toString();
    try {
        EntityManager entityManager = Persistence.createEntityManagerFactory
            ("persistenceUnitName: \"KumpulanBukuSkripsiPU\"").createEntityManager();
        entityManager.getTransaction().begin();
        String kodebuku = (String) jComboBoxKodebuku.getSelectedItem();
        Buku kodbuk = entityManager.find(type:Buku.class, o:kodebuku);
        Peminjamanbku pb = new Peminjamanbku();
        pb.setKodepinjamb(kodepinjamb);
        pb.setKodebuku(kodbuk);
        pb.setNama(nama:namab);
        pb.setNim(nim:nimb);
        pb.setAngkatan(angkatan:angkatanb);
        pb.setTglpinjam(tglpinjam:tglpinjamb);
        pb.setTglbalik(tglbalik:tglbalikb);
        pb.setStatus(status);
        entityManager.persist(o:pb);
        entityManager.getTransaction().commit();

        JOptionPane.showMessageDialog(parentComponent: null, message: "Data buku berhasil disimpan!");

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(parentComponent: null, "Data buku gagal disimpan karena : " + String.valueOf(obj:ex));
    }
    tampilanPinjamBuku();
}
```

Source code ubah:

```
private void jButtonUbahPinjamBukuActionPerformed(java.awt.event.ActionEvent evt) {
    String kodepinjamb = jTextFieldKodepinjambuku.getText().trim();
    Buku kodbuk = new Buku();
    kodbuk.setKodebuku((String) jComboBoxKodebuku.getSelectedItem());
    String namab = jTextFieldnamab.getText();
    String nimb = jTextFieldnimb.getText();
    int angkatanb = Integer.parseInt(jTextFieldangkatanb.getText());
    Date tglpinjamb = jDateChoosertpb.getDate();
    Date tglbalikb = jDateChoosertkb.getDate();
    String status = jComboBoxspb.getSelectedItem().toString();

    EntityManager em = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU").createEntityManager();
    em.getTransaction().begin();

    try {
        Peminjamanbuku pb = em.find(Peminjamanbuku.class, kodepinjamb); // Mencari buku berdasarkan kode buku

        if (pb != null) {
            pb.setKodepinjamb(kodepinjamb);
            pb.setKodebuku(kodebuku);
            pb.setNama(nama:namab);
            pb.setNim(nim:nimb);
            pb.setAngkatan(angkatan:angkatanb);
            pb.setTglpinjam(tglpinjam:tglpinjamb);
            pb.setTglbalik(tglbalik:tglbalikb);
            pb.setStatus(status);

            em.merge(pb);
            em.getTransaction().commit(); // Commit transaksi
            JOptionPane.showMessageDialog(null, "Data berhasil diubah.");
            tampilanPinjamBuku();
        } else {
            JOptionPane.showMessageDialog(null, "Data dengan kode buku: " + kodepinjamb + " tidak ditemukan.");
        }
    } catch (Exception e) {
        em.getTransaction().rollback(); // Rollback jika terjadi kesalahan
        JOptionPane.showMessageDialog(null, "Terjadi kesalahan saat mengubah data." + String.valueOf(e));
    } finally {
        jTextFieldKodepinjambuku.setText("");
        jComboBoxKodebuku.setSelectedIndex(0);
        jTextFieldnamab.setText("");
        jTextFieldnimb.setText("");
        jTextFieldangkatanb.setText("");
        jDateChoosertpb.setCalendar(cldr:null);
        jDateChoosertkb.setCalendar(cldr:null);
        jComboBoxspb.setSelectedIndex(0);
    }
}
```

Source code hapus:

```
private void jButtonHapusPinjamBukuActionPerformed(java.awt.event.ActionEvent evt) {
    String kodepinjamb = jTextFieldKodepinjambuku.getText().trim();
    EntityManager entityManager = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU").createEntityManager();
    entityManager.getTransaction().begin();
    Peminjamanbuku pb = entityManager.find(Peminjamanbuku.class, kodepinjamb);

    if (pb != null) {
        entityManager.remove(pb);
    }
    entityManager.flush();
    entityManager.getTransaction().commit();
    JOptionPane.showMessageDialog(null, "Data berhasil dihapus.");
    tampilanPinjamBuku();
}
```

Source code cetak:

```
private void jButtonCetakPinjamBukuActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String selection = (String) jComboBoxPinjamSkripsi.getSelectedItem();  
        String reportPath = "src/Report/reportPeminjamanSkripsi.jrxml";  
        Date tanggalAwal = (Date) jDateChooserTAWPB.getDate();  
        Date tanggalAkhir = (Date) jDateChooserTAKPB.getDate();  
  
        String searchTerm = jTextFieldPencarianSkripsi.getText().trim();  
        String queryString = "SELECT ps FROM Peminjamanskripsi ps ";  
  
        if (!searchTerm.isEmpty()) {  
            switch (jComboBoxPinjamSkripsi.getSelectedIndex()) {  
                case 0:  
                    queryString += "WHERE LOWER(ps.kodepinjams) LIKE LOWER(:searchTerm)";  
                    break;  
                case 1:  
                    queryString += "WHERE LOWER(ps.nama) LIKE LOWER(:searchTerm)";  
                    break;  
                case 2:  
                    queryString += "WHERE LOWER(ps.nim) LIKE LOWER(:searchTerm)";  
                    break;  
                case 3:  
                    queryString += "WHERE ps.angkatan = :searchTerm";  
                    break;  
                case 4:  
                    queryString += "WHERE LOWER(ps.status) LIKE LOWER(:searchTerm)";  
                    break;  
                default:  
                    throw new IllegalArgumentException(":Invalid search criteria selected.");  
            }  
        }  
  
        if (tanggalAwal != null && tanggalAkhir != null) {  
            if (!searchTerm.isEmpty()) {  
                queryString += " AND ";  
            } else {  
                queryString += " WHERE ";  
            }  
            queryString += "ps.tglpinjam BETWEEN :tanggalAwal AND :tanggalAkhir";  
        }  
  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU");  
        EntityManager em = emf.createEntityManager();  
        em.getTransaction().begin();  
  
        TypedQuery<Peminjamanskripsi> query = em.createQuery("string:queryString, type:Peminjamanskripsi.class");  
  
        // Set parameter tanggal jika dimasukkan  
        if (tanggalAwal != null && tanggalAkhir != null) {  
            query.setParameter("tanggalAwal", :tanggalAwal);  
            query.setParameter("tanggalAkhir", :tanggalAkhir);  
        }  
  
        if (!searchTerm.isEmpty()) {  
            if (jComboBoxPinjamSkripsi.getSelectedIndex() == 3) {  
                query.setParameter("searchTerm", :Integer.valueOf(:searchTerm));  
            } else {  
                query.setParameter("searchTerm", "%" + searchTerm + "%");  
            }  
        }  
    }  
}
```

```

List<Peminjamanskripsi> results = query.getResultList();
List<Object[]> data = new ArrayList<>();
for (Peminjamanskripsi result : results) {
    Object[] rowData = {
        result.getKodepinjams(),
        result.getKodeskripsi().getKodeskripsi(),
        result.getNama(),
        result.getNim(),
        result.getTglpinjam(),
        result.getTglbalik(),
        result.getStatus()
    };
    data.add(e: rowData);
}

em.getTransaction().commit();
em.close();
emf.close();

Map<String, Object> param = new HashMap<>();
param.put(key: "queryString", value: searchTerm);
param.put(key: "cariBerdasar", value: selection);
param.put(key: "tanggalAwal", value: tanggalAwal);
param.put(key: "tanggalAkhir", value: tanggalAkhir);

// Membuat sumber data untuk JasperReports dari data hasil pencarian
JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: results);

JasperReport jasperReport = JasperCompileManager.compileReport(sourceFileName: reportPath);
JasperPrint print = JasperFillManager.fillReport(jasperReport, parameters: param, dataSource);
JasperViewer viewer = new JasperViewer(jasperPrint: print, isExitOnClose: false);
viewer.setVisible(b: rootPaneCheckingEnabled);

} catch (JRException ex) {
    Logger.getLogger(Dashboard.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
}
}
}

```

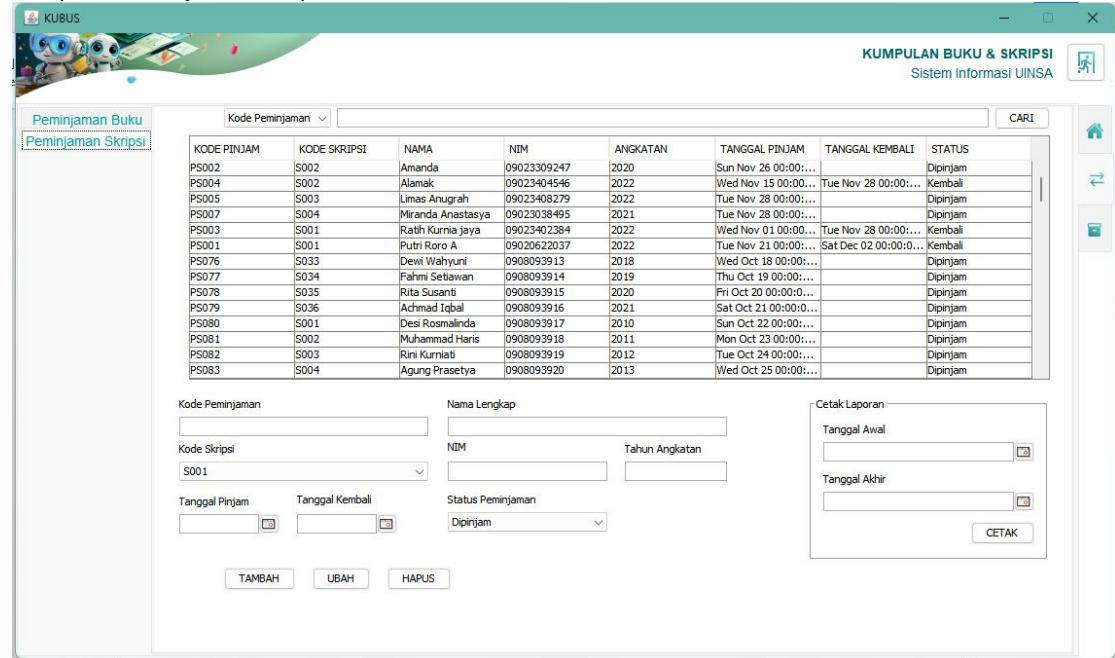
Tampilan mencetak apabila tidak diisi kolom pencarian dan kolom tanggal, maka hasilnya

LAPORAN PEMINJAMAN SKRPSI PROGRAM STUDI SISTEM INFORMASI
UIN SUNAN AMPEL SURABAYA

Hasil cetak pencarian : berdasarkan Kode Peminjaman
Tanggal awal : null
Tanggal akhir : null

KODE	KODE	NAMA LENGKAP	NIM	ANGKATAN	TGL PINJAM	TGL KEMBALI	STATUS
PS002	S002	Amanda	09023009247	2020	11/09/23, 12:00 AM	null	Dipinjam
PS004	S002	Alamak	09023404546	2022	11/15/23, 12:00 AM	11/28/23, 12:00 AM	Kembali
PS005	S003	Limas Anugrah	09023408279	2022	11/28/23, 12:00 AM	null	Dipinjam
PS007	S004	Miranda Anastasya	09023038495	2021	11/28/23, 12:00 AM	null	Dipinjam
PS003	S001	Rathn Kurnia jaya	09023402384	2022	11/21/23, 12:00 AM	11/28/23, 12:00 AM	Kembali
PS001	S001	Puti Roro A.	09020622037	2022	11/21/23, 12:00 AM	12/02/23, 12:00 AM	Kembali
PS078	S013	Dewi Wahyuni	0908093913	2018	10/18/23, 12:00 AM	null	Dipinjam
PS077	S014	Fahmi Sesawan	0908093914	2019	10/19/23, 12:00 AM	null	Dipinjam
PS078	S015	Rita Susanii	0908093915	2020	10/20/23, 12:00 AM	null	Dipinjam
PS079	S016	Achmad Iqbal	0908093916	2021	10/21/23, 12:00 AM	null	Dipinjam
PS080	S001	Desi Rosmalinda	0908093917	2010	10/22/23, 12:00 AM	null	Dipinjam
PS081	S002	Muhammad Haris	0908093918	2011	10/23/23, 12:00 AM	null	Dipinjam
PS082	S003	Riri Kurniasih	0908093919	2012	10/24/23, 12:00 AM	null	Dipinjam
PS083	S004	Agung Prasetya	0908093920	2013	10/25/23, 12:00 AM	null	Dipinjam
PS084	S005	Sari Ramadhani	0908093921	2014	10/26/23, 12:00 AM	null	Dipinjam
PS085	S006	Andre Setiawan	0908093922	2015	10/27/23, 12:00 AM	null	Dipinjam

Tampilan Peminjaman Skripsi



Source code tambah:

```

private void jButtonPinjamSkripsiActionPerformed(java.awt.event.ActionEvent evt) {
    String kodepinjams = jTextFieldKodepinjamskripsi.getText().trim();
    Skripsi kodskrip = new Skripsi();
    kodskrip.setKodeskripsi((String) jComboBoxKodeskripsi.getSelectedItem());
    String namas = jTextFieldnamas.getText();
    String nims = jTextFieldnims.getText();
    int angkatans = Integer.parseInt(jTextFieldangkatans.getText());
    Date tglpinjams = jDateChooserpinjams.getDate();
    Date tglbalik = jDateChoosertkbs.getDate();
    String status = jComboBoxspis.getSelectedItem().toString();
    try {
        EntityManager entityManager = Persistence.createEntityManagerFactory
        ( persistenceUnitName: "KumpulanBukuSkripsiPU").createEntityManager();
        entityManager.getTransaction().begin();
        Peminjamanskripsi pb = new Peminjamanskripsi();
        pb.setKodepinjams(kodepinjams);
        pb.setKodeskripsi( kodeskripsi:kodskrip);
        pb.setNama( namas);
        pb.setNim( nims);
        pb.setAngkatan( angkatans);
        pb.setTglpinjam( tglpinjams: tglpinjams);
        pb.setTglbalik( tglbalik: tglbalik);
        pb.setStatus(status);
        entityManager.persist( ::pb);
        entityManager.getTransaction().commit();
        JOptionPane.showMessageDialog( parentComponent: null, message: "Data buku berhasil disimpan!");
    } catch (Exception ex) {
        JOptionPane.showMessageDialog( parentComponent: null, "Data Skripsi gagal disimpan karena : " + String.valueOf( obj: ex));
    }
    tampilanPinjamSkripsi();
}

```

Source code ubah:

```
private void jButtonUbahPinjamSkripsiActionPerformed(java.awt.event.ActionEvent evt) {  
    String kodepinjams = jTextFieldKodepinjamskripsi.getText().trim();  
    Skripsi skrip = new Skripsi();  
    skrip.setKodeskripsi((String) jComboBoxKodeskripsi.getSelectedItem());  
    String namab = jTextFieldnamas.getText();  
    String nimb = jTextFieldnims.getText();  
    int angkatnb = Integer.parseInt(jTextFieldangkatans.getText());  
    Date tglpinjamb = jDateChoosertps.getDate();  
    Date tglbalikb = jDateChoosertks.getDate();  
    String status = jComboBoxsp.getSelectedItem().toString();  
  
    EntityManager em = Persistence.createEntityManagerFactory("KumpulanBukuSkripsiPU").createEntityManager();  
    em.getTransaction().begin();  
  
    try {  
        Peminjamanskripsi ps = em.find(type:Peminjamanskripsi.class, o:kodepinjams);  
  
        if (ps != null) {  
  
            ps.setKodepinjams(kodepinjams);  
            ps.setKodeskripsi(todeskripsi.skrip);  
            ps.setNama(nama:namab);  
            ps.setNim(nim:nimb);  
            ps.setAngkatan(angkatanb);  
            ps.setTglpinjam(tglpinjam:tglpinjamb);  
            ps.setTglbalik(tglbalik:tglbalikb);  
            ps.setStatus(status);  
  
            em.merge(o:ps);  
            em.getTransaction().commit(); // Commit transaksi  
            JOptionPane.showMessageDialog(parentComponent:null, message:"Data berhasil diubah.");  
            tampilanPinjamSkripsi();  
        } else {  
            JOptionPane.showMessageDialog(parentComponent:null, "Data dengan kode peminjaman: " + kodepinjams + " tidak ditemukan.");  
        }  
    } catch (Exception e) {  
        em.getTransaction().rollback(); // Rollback jika terjadi kesalahan  
        JOptionPane.showMessageDialog(parentComponent:hull, "Terjadi kesalahan saat mengubah data." + String.valueOf(obj:e));  
    } finally {  
        jTextFieldKodepinjamskripsi.setText(o:"");  
        jComboBoxKodeskripsi.setSelectedIndex(anIndex: 0);  
        jTextFieldnamas.setText(o:"");  
        jTextFieldnims.setText(o:"");  
        jTextFieldangkatans.setText(o:"");  
        jDateChoosertps.setCalendar(cndr:null);  
        jDateChoosertks.setCalendar(cndr:null);  
        jComboBoxsp.setSelectedIndex(anIndex: 0);  
    }  
}
```

Source code hapus:

```
private void jButtonHapusPinjamSkripsiActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String kodepinjamskripsi = jTextFieldKodepinjamskripsi.getText().trim();  
    EntityManager entityManager = Persistence.createEntityManagerFactory(  
        (persistenceUnitName: "KumpulanBukuSkripsiPU").createEntityManager();  
    entityManager.getTransaction().begin();  
    Peminjamanskripsi ps = entityManager.find(type:Peminjamanskripsi.class, o:kodepinjamskripsi);  
  
    if (ps != null) {  
        entityManager.remove(o:ps);  
    }  
    entityManager.flush();  
    entityManager.getTransaction().commit();  
    JOptionPane.showMessageDialog(parentComponent:null, message:"Data berhasil dihapus");  
    tampilanPinjamSkripsi();  
}
```

Source code cetak:

```
private void jButtonCetakPinjamSkripsiActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String selection = (String) jComboBoxPinjamSkripsi.getSelectedItem();  
        String reportPath = "src/Report/reportPeminjamanSkripsi.jrxml";  
        Date tanggalAwal = (Date) jDateChooserTAWPB.getDate();  
        Date tanggalAkhir = (Date) jDateChooserTAKPB.getDate();  
  
        String searchTerm = jTextFieldPencarianSkripsi.getText().trim();  
        String queryString = "SELECT ps FROM Peminjamanskripsi ps ";  
  
        if (!searchTerm.isEmpty()) {  
            switch (jComboBoxPinjamSkripsi.getSelectedIndex()) {  
                case 0:  
                    queryString += "WHERE LOWER(ps.kodepinjams) LIKE LOWER(:searchTerm)";  
                    break;  
                case 1:  
                    queryString += "WHERE LOWER(ps.nama) LIKE LOWER(:searchTerm)";  
                    break;  
                case 2:  
                    queryString += "WHERE LOWER(ps.nim) LIKE LOWER(:searchTerm)";  
                    break;  
                case 3:  
                    queryString += "WHERE ps.angkatan = :searchTerm";  
                    break;  
                case 4:  
                    queryString += "WHERE LOWER(ps.status) LIKE LOWER(:searchTerm)";  
                    break;  
                default:  
                    throw new IllegalArgumentException(":Invalid search criteria selected.");  
            }  
        }  
  
        if (tanggalAwal != null && tanggalAkhir != null) {  
            if (!searchTerm.isEmpty()) {  
                queryString += " AND ";  
            } else {  
                queryString += "WHERE ";  
            }  
            queryString += "ps.tglpinjam BETWEEN :tanggalAwal AND :tanggalAkhir";  
        }  
  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "KumpulanBukuSkripsiPU");  
        EntityManager em = emf.createEntityManager();  
        em.getTransaction().begin();  
  
        TypedQuery<Peminjamanskripsi> query = em.createQuery(string:queryString, type:Peminjamanskripsi.class);  
  
        // Set parameter tanggal jika dimasukkan  
        if (tanggalAwal != null && tanggalAkhir != null) {  
            query.setParameter(string:"tanggalAwal", :tanggalAwal);  
            query.setParameter(string:"tanggalAkhir", :tanggalAkhir);  
        }  
  
        if (!searchTerm.isEmpty()) {  
            if (jComboBoxPinjamBuku.getSelectedIndex() == 3) {  
                query.setParameter(string:"searchTerm", :Integer.parseInt(:searchTerm));  
            } else {  
                query.setParameter(string:"searchTerm", :searchTerm);  
            }  
        }  
    }  
}
```

```

List<Peminjamanskripsi> results = query.getResultList();
List<Object[]> data = new ArrayList<>();
for (Peminjamanskripsi result : results) {
    Object[] rowData = {
        result.getKodepinjams(),
        result.getKodeskripsi().getKodeskripsi(),
        result.getNama(),
        result.getNim(),
        result.getTglpinjam(),
        result.getTglbalik(),
        result.getStatus()
    };
    data.add(rowData);
}

em.getTransaction().commit();
em.close();
emf.close();

Map<String, Object> param = new HashMap<>();
param.put("queryString", value:searchTerm);
param.put("cariBerdasar", value:selection);
param.put("tanggallAwal", value:tanggalAwal);
param.put("tanggallAkhir", value:tanggalAkhir);

// Membuat sumber data untuk JasperReports dari data hasil pencarian
JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: results);

JasperReport jasperReport = JasperCompileManager.compileReport(sourceFileName:reportPath);
JasperPrint print = JasperFillManager.fillReport(jasperReport, parameters: param, dataSource);
JasperViewer viewer = new JasperViewer(jasperPrint:print, isExitOnClose:false);
viewer.setVisible(b:rootPaneCheckingEnabled);

} catch (JRException ex) {
    Logger.getLogger(name: Dashboard.class.getName()).log(level:Level.SEVERE, msg:null, thrown:ex);
}

```

Tampilan mencetak apabila tidak diisi kolom pencarian dan kolom tanggal, maka hasilnya

LAPORAN PEMINJAMAN SKRPSI PROGRAM STUDI SISTEM INFORMASI
UIN SUNAN AMPEL SURABAYA

KODE	KODE	NAMA LENGKAP	NIM	ANGKATAN	TGL PINJAM	TGL KEMBALI	STATUS
PS002	S002	Amanda	09023009247	2020	11/26/23, 12:00 AM	null	Dipinjam
PS004	S002	Alamak	09023404546	2022	11/15/23, 12:00 AM	11/28/23, 12:00 AM	Kembali
PS005	S003	Limas Anugrah	09023408279	2022	11/28/23, 12:00 AM	null	Dipinjam
PS007	S004	Miranda Anastasya	09023038495	2021	11/28/23, 12:00 AM	null	Dipinjam
PS003	S001	Rahil Kumira jaya	09023402384	2022	11/10/23, 12:00 AM	11/28/23, 12:00 AM	Kembali
PS001	S001	Putri Roro A	0902622037	2022	11/21/23, 12:00 AM	12/2/23, 12:00 AM	Kembali
PS076	S033	Dewi Wahyuni	0908093913	2018	10/19/23, 12:00 AM	null	Dipinjam
PS077	S034	Fahr Selawan	0908093914	2019	10/19/23, 12:00 AM	null	Dipinjam
PS078	S035	Rita Susanî	0908093915	2020	10/20/23, 12:00 AM	null	Dipinjam
PS079	S036	Achmad Iqta'	0908093916	2021	10/21/23, 12:00 AM	null	Dipinjam
PS080	S001	Desi Rosminda	0908093917	2010	10/22/23, 12:00 AM	null	Dipinjam
PS081	S002	Muhammad Haris	0908093918	2011	10/23/23, 12:00 AM	null	Dipinjam
PS082	S003	Rini Kumisâi	0908093919	2012	10/24/23, 12:00 AM	null	Dipinjam
PS083	S004	Agung Prasetya	0908093920	2013	10/25/23, 12:00 AM	null	Dipinjam
PS084	S005	Sari Ramadhanî	0908093921	2014	10/26/23, 12:00 AM	null	Dipinjam
PS085	S006	Andre Setiawan	0908093922	2015	10/27/23, 12:00 AM	null	Dipinjam

Inventory

Buku

Tampilan untuk buku



The screenshot shows the 'Buku' (Books) section of the KUBUS application. The interface includes a header with the title 'KUMPULAN BUKU & SKRIPSI' and the subtitle 'Sistem Informasi UINSA'. Below the header is a search bar labeled 'CARI'. On the left, there are two tabs: 'Buku' (selected) and 'Skripsi'. The main content area displays a table titled 'Buku' with the following columns: KODE, JUDUL, SUB JUDUL, PENGARANG, PENERBIT, TAHUN, HALAMAN, ISBN, and KATEGORI. The table contains 20 rows of book data. At the bottom right of the table are buttons for 'CETAK' and 'Edit Data Buku'.

KODE	JUDUL	SUB JUDUL	PENGARANG	PENERBIT	TAHUN	HALAMAN	ISBN	KATEGORI
B006	Cloud Computing		Nana	Gramedia	2019	563	119383722787	Integrasi Sistem
B005	War Of AI		Kim Wagon	Gramedia	2022	563	11938372298	Artificial Intelligence
B002	Basis Data	CRUD	Sucipto	Erlangga	2010	700	126713836782	Pemrograman Da...
B003	Fun Math	Linear Programming	Tian	Erlangga	2003	2913	138397418971	Keamanan Infor...
B004	Mengarang Bebas		Antonio	Erlangga	2014	291	119383749484	Pengembangan A...
B001	Indonesia Hari ini	Dampak Iklim	Sujono	Erlangga	2020	123	123421347869	Sistem Operasi
B007	Mobile App Dev	App Design Tips	Adrian Thompson	TechPublishers	2010	202	1672576125	Basis Data
B008	Modern Networking	Cybersecurity 101	Emma Collins	InnovativeBooks	2011	242	1672576126	Pemrograman Da...
B009	AI Insights	Python Programm...	Michael Turner	DigitalPress	2012	247	1672576127	Analisis Desain Si...
B010	InfoSec Secrets	Blockchain Basics	Sophia Walker	BytePublications	2013	726	1672576128	Keamanan Infor...
B011	Python Basics	UX/UI Insights	Daniel Mitchell	CodeCrafters	2014	352	1672576129	Manajemen Proj...
B012	UI/UX Revamp	Data Science Intro	Ava Johnson	EpicPublishing	2015	783	1672576130	Sistem Operasi
B013	Algo Mastery	Cloud Computing	Christopher Hall	QuantumBooks	2016	145	1672576131	Pengembangan A...
B014	Cloud Basics	Java for Beginners	Olivia Lee	NanoPublishers	2017	1093	1672576132	Integrasi Sistem
B015	Blockchain Simplified	Web Dev Funda	Anthony Taylor	CyberReads	2018	894	1672576133	Big Data dan Ana...
B016	Software Testing	Network Security	Mia Anderson	SmartPress	2019	321	1672576134	Artificial Intelligence
B017	IoT Intro	AI in Everyday Life	David Lewis	GigaBooks	2020	732	1672576135	Machine Learning
B018	Responsive Web	Mobile App Trends	Emily Harris	TechTimes	2021	986	1672576136	Basis Data
B019	Java Essentials	Database Design	Matthew Baker	LogicLibrary	2010	874	1672576137	Pemrograman Da...

Source code tampilan buku:

```
private void tampilanBuku() {
    DefaultTableModel model = (DefaultTableModel) jTableBuku.getModel();
    model.setRowCount( rowCount:0 );
    EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "KumpulanBukuSkripsiPU" );
    EntityManager em = emf.createEntityManager();
    em.getTransaction().begin();
    CriteriaBuilder cb = em.getCriteriaBuilder();
    CriteriaQuery<Buku> cq = cb.createQuery( type:Buku.class );
    Root<Buku> stud = cq.from( type:Buku.class );
    cq.select( slectn:stud.get( string:"kodebuku" ) );
    CriteriaQuery<Buku> select = cq.select( slectn:stud );
    TypedQuery<Buku> q = em.createQuery( cq.select );
    List<Buku> list = q.getResultList();

    for (Buku data : list) {
        Object[] baris = new Object[9];
        baris[0] = data.getKodebuku();
        baris[1] = data.getJudulbuku();
        baris[2] = data.getSubjudulbuku();
        baris[3] = data.getPengarang();
        baris[4] = data.getPenerbit();
        baris[5] = data.getTahunterbit();
        baris[6] = data.getHalaman();
        baris[7] = data.getIsbn();
        baris[8] = data.getKategori();
        model.addRow( rowData:baris );
    }

    jTableBuku.setModel( dataModel:model );
    em.getTransaction().commit();
    em.close();
    emf.close();
}
```

Source code cetak dengan pencarian:

```
private void jButtonCetakSearchBukuActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String selection = (String) FilterBuku.getSelectedItem();
        String reportPath = "src/Report/reportBuku.jrxml";
        String searchTerm = jTextFieldCariBuku.getText().trim();

        // Building the JPA query dynamically based on the selected criteria
        String queryString = "SELECT b FROM Buku b ";

        switch (FilterBuku.getSelectedIndex()) {
            case 0:
                queryString += "WHERE LOWER(b.kodebuku) LIKE LOWER(:searchTerm)";
                break;
            case 1:
                queryString += "WHERE LOWER(b.judulbuku) LIKE LOWER(:searchTerm)";
                break;
            case 2:
                queryString += "WHERE LOWER(b.subjudulbuku) LIKE LOWER(:searchTerm)";
                break;
            case 3:
                queryString += "WHERE LOWER(b.pengarang) LIKE LOWER(:searchTerm)";
                break;
            case 4:
                queryString += "WHERE LOWER(b.penerbit) LIKE LOWER (:searchTerm)";
                break;
            case 5:
                queryString += "WHERE (b.tahunterbit) = (:searchTerm)";
                break;
            case 6:
                queryString += "WHERE (b.halaman) = (:searchTerm)";
                break;
            case 7:
                queryString += "WHERE LOWER(b.isbn) LIKE LOWER (:searchTerm)";
                break;
            case 8:
                queryString += "WHERE LOWER(b.kategori) LIKE LOWER (:searchTerm)";
                break;
            default:
                queryString += "WHERE LOWER(b.kodebuku) LIKE LOWER(:searchTerm)";
                break;
        }

        EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "KumpulanBukuSkripsiPU");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();

        CriteriaBuilder cb = em.getCriteriaBuilder();
        CriteriaQuery<Buku> cq = cb.createQuery(type: Buku.class);
        Root<Buku> bok = cq.from(type: Buku.class);
        cq.select(slectn:bok);

        // Check if WHERE clause is not empty
        if (queryString.endsWith(suffix: " WHERE ")) {
            Query query = em.createQuery(string: "SELECT b FROM Buku b");
        }

        TypedQuery<Buku> q = em.createQuery(cq);
        List<Buku> list = q.getResultList();
        Query query = em.createQuery(string: queryString);
        //query.setParameter("searchTerm", "%" + searchTerm + "%");
    }
}
```

```

        if (!searchTerm.isEmpty()) {
            if (FilterBuku.getSelectedIndex() == 5 || FilterBuku.getSelectedIndex() == 6) {
                query.setParameter("searchTerm", Integer.parseInt(searchTerm));
            } else {
                query.setParameter("searchTerm", "%" + searchTerm + "%");
            }
        }

        List<Buku> results = query.getResultList();

        List<Object[]> data = new ArrayList<>();
        for (Buku result : results) {
            Object[] rowData = {
                result.getKodebuku(),
                result.getJudulbuku(),
                result.getSubjudulbuku(),
                result.getPengarang(),
                result.getPenerbit(),
                result.getTahunterbit(),
                result.getHalaman(),
                result.getIsbn(),
                result.getKategori(),};
            data.add(rowData);
        }
        em.getTransaction().commit();
        em.close();
        emf.close();

        Map<String, Object> param = new HashMap<>();
        param.put("queryString", searchTerm);
        param.put("cariBerdasar", selection);

        // Membuat sumber data untuk JasperReports dari data hasil pencarian
        JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: results);

        // Memuat file desain laporan (*.jrxml)
        JasperReport jasperReport = JasperCompileManager.compileReport(sourceFileName: reportPath);
        JasperPrint print = JasperFillManager.fillReport(jasperReport, parameters: param, dataSource);
        JasperViewer viewer = new JasperViewer(jasperPrint: print, isExitOnClose: false);
        viewer.setVisible(true);
    } catch (JRException ex) {
        Logger.getLogger(Dashboard.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}

```

Tampilan edit buku

Data Buku

KODE	JUDUL	SUB JUDUL	PENGARANG	PENERBIT	TAHUN	HALAMAN	ISBN	KATEGORI
B001	Cloud Computing	Nana	Gramaedia	2019	563	119383722787	Integrasi Sistem	
B005	War Of AI	Kim Wagon	Gramaedia	2022	563	119383722958	Artificial Intellig...	
B002	Basis Data	Sucipto	Erlangga	2010	700	126713836782	Pemrograman D...	
B003	Fun Math	Linear Program...	Tian	Erlangga	2003	2913	138397418971	Keamanan Infor...
B004	Mengarang Bebas	Antonio	Erlangga	2014	291	119383749484	Pengembangan ...	
B001	Indonesia Hari ini	Dampak Idim	Sujono	Erlangga	2020	123	123421347869	Sistem Operasi
B007	Mobile App Dev	App Design Tips	Adrian Thompson	TechPubl...	2010	202	1672576125	Basis Data
B008	Modern Network	Cybersecurity 101	Emma Collins	Innovativ...	2011	242	1672576126	Pemrograman D...
B009	AI Insights	Python Program...	Michael Turner	DigitalPress	2012	247	1672576127	Analisis Desain ...
B010	InfoSec Secrets	Blockchain Basics	Sophia Walker	BytePubl...	2013	726	1672576128	Keamanan Infor...
B011	Python Basics	UX/UI Insights	Daniel Mitchell	CodeCra...	2014	352	1672576129	Manajemen Pro...
B012	UI/AJX Revamp	Data Science Intro	Ava Johnson	EpicPubl...	2015	783	1672576130	Sistem Operasi
B013	Algo Mastery	Cloud Computing	Christopher Hall	Quantum...	2016	145	1672576131	Pengembangan ...
B014	Cloud Basics	Java for Beginners	Olivia Lee	NanoPubl...	2017	1093	1672576132	Integrasi Sistem
B015	Blockchain Simpl...	Web Dev Funda	Anthony Taylor	CyberRe...	2018	894	1672576133	Big Data dan An...
B016	Software Testing	Network Security	Mia Anderson	SmarTPress	2019	321	1672576134	Artificial Intellig...
B017	IoT Intro	AI in Everythin...	David Lewis	GigaBooks	2020	732	1672576135	Machine Learning
B018	Responsive Web	Mobile App Trends	Emily Harris	TechTimes	2021	986	1672576136	Basis Data
B019	Java Essentials	Database Design	Matthew Baker	LogiLibr...	2010	874	1672576137	Pemrograman D...
B020	Big Data Intro	JavaScript Mast...	Isabella Martin	VirtualPr...	2011	290	1672576138	Analisis Desain ...
B021	Efficient DB Design	Big Data Analytics	Andrew Adams	WebWise...	2012	210	1672576139	Keamanan Infor...
B022	HTML5 Mastery	Mobile Dev Str...	Abigail Wright	DataDriv...	2013	202	1672576140	Manajemen Pro...
B023	Data Analytics	Software Testing	Ryan Garcia	CloudCr...	2014	242	1672576141	Sistem Operasi
B024	Scala Basics	Responsive Web	Madison Scott	InfoInsig...	2015	247	1672576142	Pengembangan ...
B025	Microservices 101	IOS App Develo...	Nicholas Martinez	AgilePress	2016	726	1672576143	Integrasi Sistem
B026	Web App Security	Android Studio ...	Ella King	GeniusB...	2017	352	1672576144	Big Data dan An...
B027	AI Exploration	Machine Learning	James Miller	PixelPubl...	2018	783	1672576145	Artificial Intellig...
B028	Flutter Basics	Swift Programming	Avery Wilson	FutureW...	2019	145	1672576146	Machine Learning
B029	Legal Tech Aspe...	Game Dev Conc...	John Moore	InnoTexts	2020	1093	1672576147	Basis Data
B030	ML in Business	Tech Project Mgmt	Grace Clark	ByteBrill...	2021	894	1672576148	Pemrograman D...
B031	Git Mastery	Data Analysis Tip	Robert White	CodeCra...	2010	321	1672576149	Analisis Desain ...
B032	C++ Basics	Web Security Tip	Scarlett Turner	Quantum...	2011	732	1672576150	Keamanan Infor...
B033	Software Projec...	Agile Development	William Ward	EpicTech...	2012	986	1672576151	Manajemen Pro...
B034	Facial Recognition	Network Protocols	Sofia Turner	NanoNo...	2013	874	1672576152	Sistem Operasi
B035	Unique UX Design	IoT Applications	Joseph Thompson	CyberSc...	2014	290	1672576153	Pengembangan ...

Source code tambah buku:

```
private void jButtonTambahActionPerformed(java.awt.event.ActionEvent evt) {  
    String kodeBuku = jTextFieldKodeBuku.getText().trim();  
    String judul = jTextFieldJudulBuku.getText();  
    String subjudul = jTextFieldSubjudul.getText();  
    String pengarang = jTextFieldPengarang.getText();  
    String penerbit = jTextFieldPenerbit.getText();  
    int tahun = Integer.parseInt(jTextFieldTahun.getText());  
    int halaman = Integer.parseInt(jTextFieldHalaman.getText());  
    String isbn = jTextFieldISBN.getText();  
    String kategori = jComboBoxKategori.getSelectedItem().toString();  
    try (EntityManager entityManager = Persistence.createEntityManagerFactory(  
        persistenceUnitName: "KumpulanBukuSkripsiPU").createEntityManager()  
    )  
    {  
        entityManager.getTransaction().begin();  
        Buku b = new Buku();  
        b.setKodebuku(kodeBuku);  
        b.setJudulbuku(judul);  
        b.setSubjudulbuku(subjudul);  
        b.setPengarang(pengarang);  
        b.setPenerbit(penerbit);  
        b.setTahunterbit(tahun);  
        b.setHalaman(halaman);  
        b.setIsbn(isbn);  
        b.setKategori(kategori);  
        entityManager.persist(b);  
        entityManager.getTransaction().commit();  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Data buku berhasil disimpan!");  
        kosongkanBuku();  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent: null, "Data buku gagal disimpan karena : " + String.valueOf(obj: ex));  
    }  
  
    tampilanBuku();
```

Source code ubah buku:

```
private void jButtonUbahActionPerformed(java.awt.event.ActionEvent evt) {  
    String kodeBuku = jTextFieldKodeBuku.getText().trim();  
    String judul = jTextFieldJudulBuku.getText();  
    String subjudul = jTextFieldSubjudul.getText();  
    String pengarang = jTextFieldPengarang.getText();  
    String penerbit = jTextFieldPenerbit.getText();  
    int tahun = Integer.parseInt(jTextFieldTahun.getText());  
    int halaman = Integer.parseInt(jTextFieldHalaman.getText());  
    String isbn = jTextFieldISBN.getText();  
    String kategori = (String) jComboBoxKategori.getSelectedItem();  
    EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "KumpulanBukuSkripsiPU");  
    EntityManager em = emf.createEntityManager();  
    em.getTransaction().begin();  
    try (Buku b = em.find(type: Buku.class, e: kodeBuku);  
    if (b != null) {  
        b.setKodebuku(kodeBuku);  
        b.setJudulbuku(judul);  
        b.setSubjudulbuku(subjudul);  
        b.setPengarang(pengarang);  
        b.setPenerbit(penerbit);  
        b.setTahunterbit(tahun);  
        b.setHalaman(halaman);  
        b.setIsbn(isbn);  
        b.setKategori(kategori);  
        em.merge(b);  
        em.getTransaction().commit();  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Data berhasil diubah.");  
        tampilanBuku();  
    } else {  
        JOptionPane.showMessageDialog(parentComponent: null, "Data dengan kode buku: " + kodeBuku + " tidak ditemukan.");  
    } catch (Exception e) {  
        em.getTransaction().rollback(); // Rollback jika terjadi kesalahan  
        JOptionPane.showMessageDialog(parentComponent: null, "Terjadi kesalahan saat mengubah data." + String.valueOf(obj: e));  
    } finally {  
        kosongkanBuku();  
    }  
}
```

Source code hapus buku:

```
private void jButtonHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String kodebuku = jTextFieldKodeBuku.getText().trim();  
    EntityManager entityManager = Persistence.createEntityManagerFactory()  
        (persistenceUnitName: "KumpulanBukuSkripsiPU").createEntityManager();  
    entityManager.getTransaction().begin();  
    Buku b = entityManager.find(type: Buku.class, o: kodebuku);  
  
    if (b != null) {  
        entityManager.remove(o: b);  
    }  
    entityManager.flush();  
    entityManager.getTransaction().commit();  
    JOptionPane.showMessageDialog(parentComponent: null, message: "Data berhasil dihapus");  
    tampilanBuku();  
}
```

Source code cetak buku:

```
private void jButtonCetakActionPerformed(java.awt.event.ActionEvent evt) {  
    String reportPath = "src/Report/reportBuku.jrxml";  
  
    EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "KumpulanBukuSkripsiPU");  
    EntityManager em = emf.createEntityManager();  
    em.getTransaction().begin();  
  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    CriteriaQuery<Buku> cq = cb.createQuery(type: Buku.class);  
    Root<Buku> bukul = cq.from(type: Buku.class);  
    cq.select(slectn: bukul);  
  
    Query query = em.createQuery(string: "SELECT b FROM Buku b");  
    List<Buku> results = query.getResultList();  
  
    JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: results);  
  
    try {  
        JasperReport jr = JasperCompileManager.compileReport(sourceFileName: reportPath);  
        JasperPrint jp = JasperFillManager.fillReport(jasperReport: jr, parameters: null, dataSource);  
        JasperViewer vw = new JasperViewer(jasperPrint: jp, isExitOnClose: false);  
        vw.setVisible(b: true);  
    } catch (JRException ex) {  
        Logger.getLogger(name: DataBuku.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
    }  
  
    em.getTransaction().commit();  
    em.close();  
    emf.close();  
}
```

Tampilan ketika menekan cetak, maka akan tampil keseluruhan data Skripsi:

KODE	JUDUL BUKU	SUBJUDUL BUKU	PENERBIT	TAHUN	HALAMAN	ISBN	KATEGORI
B006	Cloud Computing		Gramedia	2019	563	119383722787	Integrasi Sistem
B005	War Of AI		Gramedia	2022	563	119383722388	Artificial Intelligence
B002	Basis Data	CRUD	Erlangga	2010	700	126713836782	Programmation Dasar
B003	Fun Math	Linear Programming	Erlangga	2003	2913	138397418971	Keamanan Informasi
B004	Mengarang Bervas		Erlangga	2014	291	119383749484	Pengembangan Aplikasi
B001	Indonesia Hari ini	Dampak Iklim	Erlangga	2020	123	123421347889	Sistem Operasi
B007	Mobile App Dev	App Design Tips	TechPublishers	2010	200	1672576125	Basis Data
B008	Modern Networking	Cybersecurity 101	InnovativeBooks	2011	242	1672576126	Programmation Dasar
B009	All Insights	Python Programming	DigitalPress	2012	247	1672576127	Analisis Desain Sistem
B010	InfoSec Secrets	Blockchain Basics	BytePublications	2013	728	1672576128	Keamanan Informasi
B011	Python Basics	UXUI Insights	CodeCrafters	2014	352	1672576129	Manajemen Proyek TI
B012	UIUX Revamp	Data Science Intro	EpicPublishing	2015	783	1672576130	Sistem Operasi
B013	Algo Mastery	Cloud Computing	QuantumBooks	2016	145	1672576131	Pengembangan Aplikasi
B014	Cloud Basics	Java for Beginners	NanoPublishers	2017	1093	1672576132	Integrasi Sistem
B015	Blockchain Simplified	Web Dev Funda	CyberReads	2018	894	1672576133	Big Data dan Analitika
B016	Software Testing	Network Security	SmartPress	2019	321	1672576134	Artificial Intelligence

Skripsi

Tampilan untuk skripsi:

KODE	JUDUL	PENGARANG	TAHUN	HALAMAN
S001	Implementasi Akar	Almas	2014	64
S002	Koordinat Tabung Menggunakan P...	Nia Angguni	2020	54
S003	Mengungkap Identitas Angin Utara	Ababil Ram	2021	78
S004	Implementasi Aplikasi Odoo Pada ...	Ratha Sarimpit	2021	31
S005	Analisis Algoritma Sorting	Budi Santoso	2010	202
S006	Sistem Informasi Perpustakaan	Siti Rahayu	2011	242
S007	Pengembangan Aplikasi Mobile	Ahmad Yudianto	2012	247
S008	Implementasi Blockchain	Dewi Kusuma	2013	726
S009	Analisis Keamanan Jaringan	Rudi Setiawan	2014	352
S010	Sistem Pendukung Keputusan	Sri Lestari	2015	783
S011	Pengenalan Pola Wajah	Rahmat Hidayat	2016	145
S012	Rekayasa Perangkat Lunak	Dina Rahmawati	2017	1093
S013	Deteksi Intrusi Komputer	Hendra Saputra	2018	894
S014	Manajemen Proyek TI	Nia Fibriani	2019	321
S015	Sistem Pakar Diagnosa Penyakit	Adi Nugroho	2020	732
S016	Pengolahan Citra Digital	Rina Sari	2021	986
S017	Evaluasi Kinerja Sistem	Eko Prasetyo	2010	874
S018	Pengembangan Website E-Comme...	Maya Indah	2011	290
S019	Sistem Manajemen Database	M. Ridwan	2012	210

Source code tampilan skripsi:

```

private void tampilanSkripsi() {
    DefaultTableModel model = (DefaultTableModel) jTableSkripsi.getModel();
    model.setRowCount( rowCount:0 );

    EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "KumpulanBukuSkripsiPU" );
    EntityManager em = emf.createEntityManager();
    em.getTransaction().begin();

    CriteriaBuilder cb = em.getCriteriaBuilder();
    CriteriaQuery<Skripsi> cq = cb.createQuery( type: Skripsi.class );
    Root<Skripsi> stud = cq.from( type: Skripsi.class );
    cq.select( slectn: stud.get( string: "kodeskripsi" ) );

    CriteriaQuery<Skripsi> select = cq.select( slectn: stud );
    TypedQuery<Skripsi> q = em.createQuery( cq: select );
    List<Skripsi> list = q.getResultList();

    for ( Skripsi data : list ) {
        Object[] baris = new Object[5];
        baris[0] = data.getKodeskripsi();
        baris[1] = data.getJudulskripsi();
        baris[2] = data.getPenulis();
        baris[3] = data.getTahun();
        baris[4] = data.getHalaman();

        model.addRow( rowData: baris );
    }
}

```

Tampilan edit skripsi

KODE SKRIPSI	JUDUL	PENGARANG	TAHUN	HALAMAN
S001	Implementasi Algoritma Almas	Nia Angguni	2014	64
S002	Koordinat Tabung Mengikuti Lingkaran	Nia Angguni	2020	54
S003	Mengungkap Identitas A... Ababil Rani	Ababil Rani	2021	78
S004	Implementasi Aplikasi Od... Ratna Sarimbit	Ratna Sarimbit	2021	31
S005	Analisis Algoritma Sorting	Budi Santoso	2010	202
S006	Sistem Informasi Perpus...	Siti Rahayu	2011	242
S007	Pengembangan Aplikasi...	Ahmad Yudanto	2012	247
S008	Implementasi Blockchain	Dewi Kusuma	2013	726
S009	Analisis Keamanan Jaringan	Rudi Setiawan	2014	352
S010	Sistem Pendukung Keput...	Sri Lestari	2015	783
S011	Pengenalan Pola Wajah	Rahmat Hidayat	2016	145
S012	Rekayasa Perangkat Lunak	Dina Rahmawati	2017	1093
S013	Deteksi Intrusi Komputer	Hendra Saputra	2018	894
S014	Manajemen Proyek TI	Nia Fitriani	2019	321
S015	Sistem Pakar Diagnosa P...	Adi Nugroho	2020	732
S016	Pengolahan Citra Digital	Rini Sari	2021	986
S017	Evaluasi Kinerja Sistem	Eko Prasetyo	2010	874
S018	Pengembangan Website ...	Maya Indah	2011	290
S019	Sistem Manajemen Datab...	M. Ridwan	2012	210
S020	Penerapan Teknologi IoT	Lina Anggraeni	2013	202
S021	Analisis Big Data	Firman Syahputra	2014	242
S022	Optimasi Algoritma	Nurul Hasanah	2015	247
S023	Keamanan Aplikasi Web	Rizki Amelia	2016	726
S024	Sistem Informasi Geografi	Haryono Wilbowo	2017	352
S025	Pengembangan Game Ed...	Ratna Dewi	2018	783
S026	Manajemen Basis Data T...	Irfan Abdullah	2019	145
S027	Pemodelan dan Simulasi	Ani Widarti	2020	1093
S028	Desain Antarmuka Pengg...	Robby Susanto	2021	894
S029	Pengembangan Aplikasi GIS	Citra Utami	2010	321
S030	Pengenalan Pola Suara	Hendra Kurniawan	2011	732
S031	Sistem Informasi Akademik	Ayu Septiani	2012	986
S032	Analisis Kinerja Jaringan	Didik Prabowo	2013	874
S033	Pengembangan Aplikasi AI	Lia Agustina	2014	290
S034	Rekayasa Keamanan Per...	Andi Saputra	2015	210
S035	Deonolahan Bahasa Alami	Irina Moutasari	2016	202

Source code tambah data skripsi:

```
private void jButtonTambahActionPerformed(java.awt.event.ActionEvent evt) {  
    String kodeskripsi = jTextFieldKodeSkripsi.getText().trim();  
    String judul = jTextFieldJudul.getText();  
    String pengarang = jTextFieldPengarang.getText();  
    int tahun = Integer.parseInt(jTextFieldTahun.getText());  
    int halaman = Integer.parseInt(jTextFieldHalaman.getText());  
  
    try {  
        EntityManager entityManager = Persistence.createEntityManagerFactory(  
            (persistenceUnitName: "KumpulanBukuSkripsiPU").createEntityManager();  
        entityManager.getTransaction().begin();  
  
        Skripsi s = new Skripsi();  
  
        s.setKodeskripsi(kodeskripsi);  
        s.setJudulskripsi(judul);  
        s.setPenulis(penulis: pengarang);  
        s.setTahun(tahun);  
        s.setHalaman(halaman);  
  
        entityManager.persist(s);  
        entityManager.getTransaction().commit();  
  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Data buku berhasil disimpan!");  
        kosongkanSkripsi();  
  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent: null, "Data buku gagal disimpan karena : " + String.valueOf(obj: ex));  
    }  
  
    tampilanSkripsi();  
}
```

Source code ubah data skripsi:

```
private void jButtonUbahActionPerformed(java.awt.event.ActionEvent evt) {  
    String kodeskripsi = jTextFieldKodeSkripsi.getText().trim();  
    String judul = jTextFieldJudul.getText();  
    String pengarang = jTextFieldPengarang.getText();  
    int tahun = Integer.parseInt(jTextFieldTahun.getText());  
    int halaman = Integer.parseInt(jTextFieldHalaman.getText());  
  
    EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "KumpulanBukuSkripsiPU");  
    EntityManager em = emf.createEntityManager();  
    em.getTransaction().begin();  
    try {  
        Skripsi s = em.find(type: Skripsi.class, : kodeskripsi);  
        if (s != null) {  
  
            s.setKodeskripsi(kodeskripsi);  
            s.setJudulskripsi(judul);  
            s.setPenulis(penulis: pengarang);  
            s.setTahun(tahun);  
            s.setHalaman(halaman);  
  
            em.merge(s);  
            em.getTransaction().commit(); // Commit transaksi  
            JOptionPane.showMessageDialog(parentComponent: null, message: "Data berhasil diubah.");  
            tampilanSkripsi();  
        } else {  
            JOptionPane.showMessageDialog(parentComponent: null, "Data dengan kode skripsi: " + kodeskripsi + " tidak ditemukan.");  
        }  
    } catch (Exception e) {  
        em.getTransaction().rollback(); // Rollback jika terjadi kesalahan  
        JOptionPane.showMessageDialog(parentComponent: null, "Terjadi kesalahan saat mengubah data." + String.valueOf(obj: e));  
    } finally {  
        kosongkanSkripsi();  
    }  
}
```

Source code hapus data skripsi:

```
private void jButtonHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String kodeskripsi = jTextFieldKodeSkripsi.getText().trim();  
    EntityManager entityManager = Persistence.createEntityManagerFactory()  
        ( persistenceUnitName: "KumpulanBukuSkripsiPU" ).createEntityManager();  
    entityManager.getTransaction().begin();  
    Skripsi s = entityManager.find( type: Skripsi.class, o: kodeskripsi );  
  
    if ( s != null ) {  
        entityManager.remove( o: s );  
    }  
    entityManager.flush();  
    entityManager.getTransaction().commit();  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Data berhasil dihapus" );  
    tampilanSkripsi();  
}
```

Source code cetak data buku:

```
private void jButtonCetakActionPerformed(java.awt.event.ActionEvent evt) {  
    String reportPath = "src/Report/reportSkripsi.jrxml";  
  
    EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "KumpulanBukuSkripsiPU" );  
    EntityManager em = emf.createEntityManager();  
    em.getTransaction().begin();  
  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    CriteriaQuery<Skripsi> cq = cb.createQuery( type: Skripsi.class );  
    Root<Skripsi> skrip = cq.from( type: Skripsi.class );  
    cq.select( slectn: skrip );  
  
    Query query = em.createQuery( string: "SELECT s FROM Skripsi s" );  
    List<Buku> results = query.getResultList();  
  
    JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource( beanCollection: results );  
  
    try {  
        JasperReport jr = JasperCompileManager.compileReport( sourcefileName: reportPath );  
        JasperPrint jp = JasperFillManager.fillReport( jasperReport: jr, parameters: null, dataSource );  
        JasperViewer vw = new JasperViewer( jasperPrint: jp, isExitOnClose: false );  
        vw.setVisible( b: true );  
    } catch ( JRException ex ) {  
        Logger.getLogger( name: DataBuku.class.getName() ).log( level: Level.SEVERE, msg: null, thrown: ex );  
    }  
  
    em.getTransaction().commit();  
    em.close();  
    emf.close();  
}
```

Tampilan ketika menekan cetak, maka akan tampil keseluruhan data Skripsi:

LAPORAN SKRIPSI PROGRAM STUDI SISTEM INFORMASI
UIN SUNAN AMPEL SURABAYA

Tanggal cetak: Tuesday 05 December 2023

KODE	JUDUL SKRIPSI	PENULIS	TAHUN	HALAMAN
S001	Implementasi Akar	Aimas	2014	64
S002	Koordinat Tabung Menggunakan Paralel	Nia Angguri	2020	54
S003	Mengingkat Identitas Angkut Utara	Ababil Rani	2021	78
S004	Implementasi Aplikasi Odometer Pada Rumah Makan Padang	Rina Sarimpit	2021	31
S005	Analisis Algoritma Sorting	Budi Santoso	2010	202
S006	Sistem Informasi Perpustakaan	Siti Rahayu	2011	242
S007	Pengembangan Aplikasi Mobile	Ahmad Yudianto	2012	247
S008	Implementasi Blockchain	Dewi Kusuma	2013	726
S009	Analisis Keamanan Jaringan	Rudi Setiawan	2014	352
S010	Sistem Pendukung Keputusan	Sri Lestari	2015	783
S011	Pengenalan Pola Wajah	Rahmat Hidayat	2016	145
S012	Rekayasa Perangkat Lunak	Dina Rahmawati	2017	1093
S013	Deteksi Intrusi Komputer	Hendra Saputra	2018	894
S014	Manajemen Proyek TI	Nia Fitri	2019	321
S015	Sistem Pakar Diagnosa Penyakit	Adi Nugroho	2020	732
S016	Pengolahan Citra Digital	Rina Sari	2021	986

Ketika menekan tombol logout, maka akan tampil:

KUMPULAN BUKU & SKRIPSI
Sistem Informasi UINSA

Peminjaman Terbanyak

Buku Basis Data 7 CETAK

Skripsi Implementasi Akar 5 CETAK

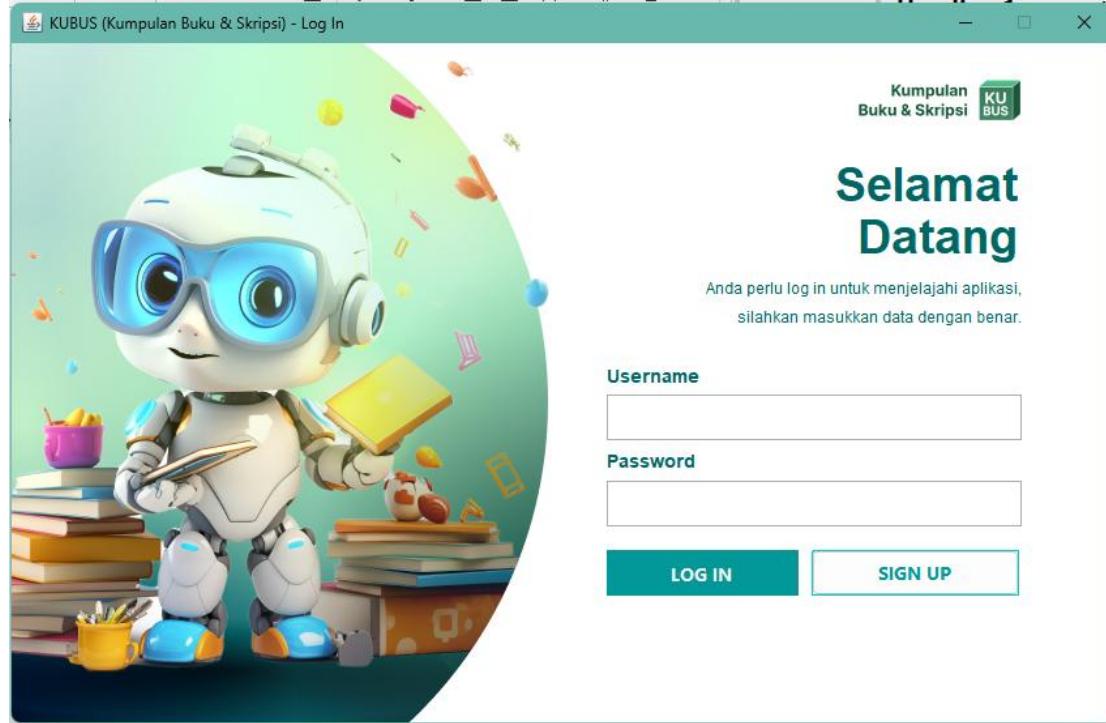
Konfirmasi Logout

Apakah Anda yakin ingin logout?

Yes No

Kategori Pemrograman Dasar 15 CETAK

Ketika menekan opsi “yes”, maka akan kembali pada frame login



Source code log out:

```
private void btn_logoutActionPerformed(java.awt.event.ActionEvent evt) {
    int confirmLogout = JOptionPane.showConfirmDialog(Dashboard.this, "Apakah Anda yakin ingin keluar?", "Konfirmasi Logout", JOptionPane.YES_NO_OPTION);

    if (confirmLogout == JOptionPane.YES_OPTION) {
        FormLogin FL = new FormLogin();
        FL.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        FL.setVisible(true);

        dispose();
    } else {
    }
}
```