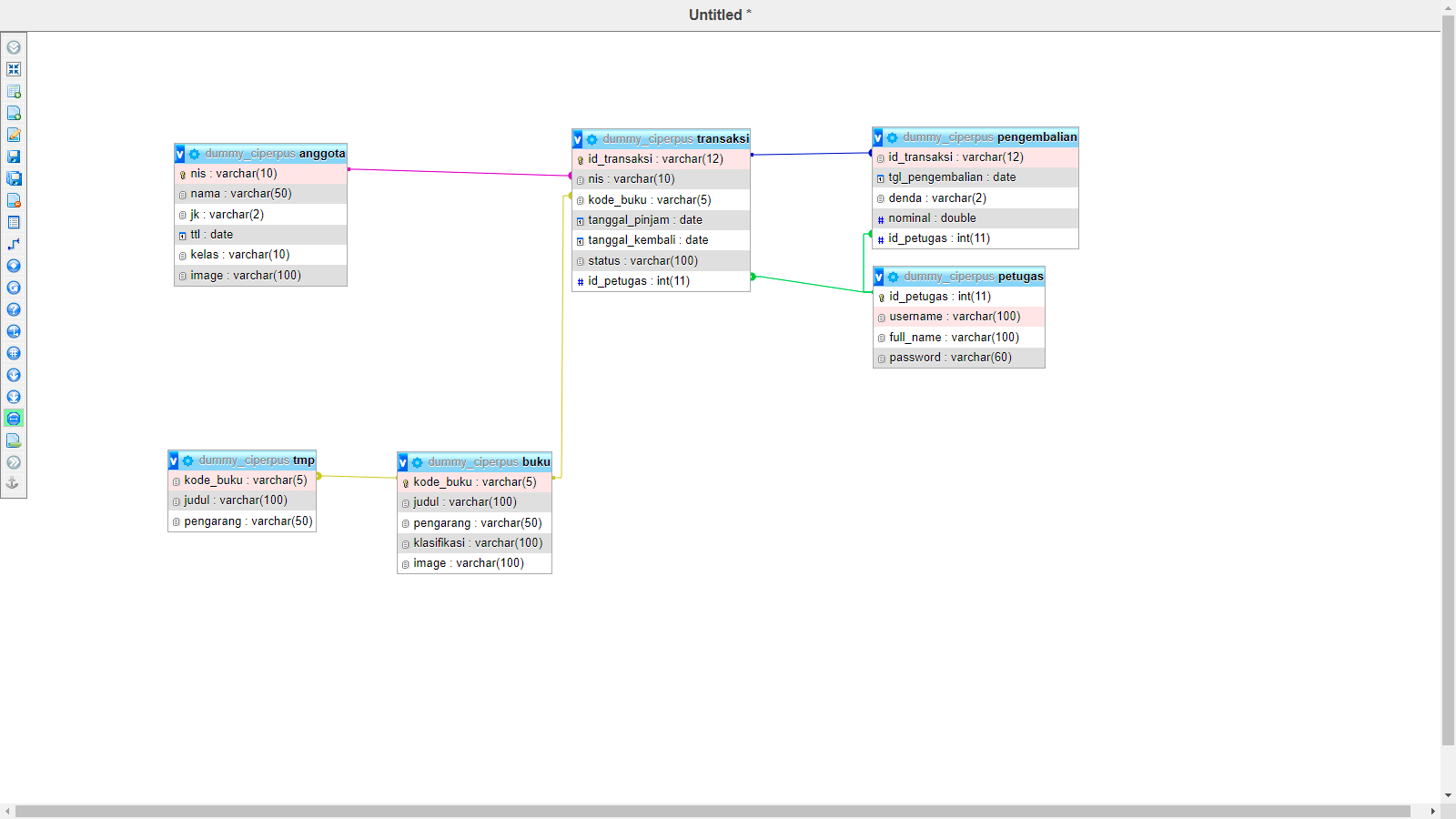
# Portofolio SQL DBA Sistem Manajemen Perpustakaan

Projek database untuk sistem manajemen perpustakaan ini menggunakan database mysql dan memiliki 6 tabel yaitu ada tabel anggota, buku, pengembalian, petugas, tmp dan transaksi. Yang dimana tiap tabel tabel tersebut mempunyai fungsi nya masing masing untuk menunjang kebutuhan fitur yang akan tersedia pada website sistem manajemen perpustakaan ini.

* Fitur yang tersedia di websitenya
* Login dan Logout
* Daftar anggota
* Daftar buku
* Daftar user
* Peminjaman buku
* Pengembalian buku
* Laporan peminjaman buku
* Laporan pengembalian buku

## ERD (Entity Relationship Diagram) Database yang digunakan

Entity-Relationship Diagram (ERD) dalam database adalah alat visual yang digunakan untuk menggambarkan struktur data dalam database. ERD adalah cara yang efektif untuk memodelkan entitas (objek atau konsep), atribut (sifat dari entitas), dan hubungan antar entitas.



## DDL (Data Defenition Language)

Adalah kumpulan perintah yang digunakan untuk mendefinisikan dan mengubah struktur database dan objek-objek yang ada di dalamnya. DDL tidak berurusan dengan data itu sendiri, tetapi dengan cara data tersebut disusun dan bagaimana data tersebut disimpan, dibawah ini merupakan query untuk membuat tabel dan atribut penghubung ke tabel lain.

1. Tabel “anggota”

Yang dimana pada tabel anggota ini berperan untuk menyimpan data dari anggota siswa yang mau meminjam buku dari perpustakaan.

Query :

**create** **table** anggota (

nis **varchar** (10) **not** **null**,

nama **varchar** (50) **not** **null**,

jk **varchar** (2) **not** **null**,

ttl **date** **not** **null**,

kelas **varchar** (10) **not** **null**,

image **varchar** (100) **not** **null**

);

1. Tabel “buku”

Pada tabel buku ini berisi semua informasi tentang buku yang berada di perpustakaan dimulai dari kode\_buku, judul buku sampe gambar buku.

Query :

**create** **table** buku(

kode\_buku **varchar** (5) **not** **null**,

judul **varchar** (100) **not** **null**,

pengarang **varchar** (50) **not** **null**,

klasifikasi **varchar** (100) **not** **null**,

image **varchar** (100) **not** **null**

);

1. Tabel “pengembalian”

Di tabel pengembalian ini berfungsi untuk menyimpan data dari tiap anggota yang sudah mengembalikan buku yand di pinjam dari perpustakaan.

Query :

**create** **table** pengembalian (

id\_transaksi **varchar** (12) **not** **null**,

tgl\_pengembalian **date** **not** **null**,

denda **varchar** (2) **not** **null**,

nominal **double** **not** **null**,

id\_petugas **int** (11) **not** **null**

);

1. Tabel “petugas”

Tabel petugas ini berisi informasi dari petugas perpustakaan yang mengelola website sistem manajemen perpustakaan.

Query :

**create** **table** petugas (

id\_petugas **int** (11) **not** **null**,

username **varchar** (100) **not** **null**,

full\_name **varchar** (100) **not** **null**,

**password** **varchar** (60) **not** **null**);

1. Tabel “tmp”

Pada tabel ini akan fokus untuk menyimpan semua data buku yang masih di pinjam dan ketika buku tersebut sudah dikembalikan maka data dari tmp akan menghilang.

Query :

**create** **table** tmp (

kode\_buku **varchar** (5) **not** **null**,

judul **varchar** (100) **not** **null**,

pengarang **varchar** (50) **not** **null**

);

1. Tabel “Transaksi”

Di tabel ini akan mencatat semua aktifitas peminjaman dan pengembalian buku di sistem manajemen perpustakaan ini.

Query :

**create** **table** transaksi (

id\_transaksi **varchar**(12) **not** **null**,

nis **varchar** (10) **not** **null**,

kode\_buku **varchar** (5) **not** **null**,

tanggal\_pinjam **date** **not** **null**,

tanggal\_kembali **date** **not** **null**,

status **varchar** (100) **not** **null**,

id\_petugas **int** (11) **not** **null**

);

Query berikut berfungsi untuk menghubungkan tiap tabel dan sekaligus sebagai peng index an kolom kolom.

1. Index tabel ”anggota”

Pada query dibawah ini berfungsi untuk menambahkan primary key pada kolom nis.

Query :

**alter** **table** anggota

**add** **primary** **key** (nis);

1. Index tabel “buku”

Query index dibawah ini berfungsi untuk menambahkan primary key pada kolom kode\_buku.

Query :

**alter** **table** buku

**add** **primary** **key** (kode\_buku);

1. Index tabel “petugas”

Di query ini menambahkan struktur otomatis membuat daftar angka yang nantinya kolom id\_petugas akan terisi sendiri mengikuti data yang dimasukan dan menambahkan primary key pada kolom id\_petugas.

Query :

**alter** **table** petugas

**modify** **column** id\_petugas **int** (11) **auto\_increment**,

**add** **primary** **key** (id\_petugas);

1. Index tabel “transaksi”

Pada query ini akan menambahkan primary key pada id\_transaksi dan menghubungkan kolom kode\_buku, id\_petugas dan nis ke tabel tujuan.

Query :

**alter** **table** transaksi

**add** **primary** **key** (id\_transaksi),

**add** **constraint** fk\_kode\_buku\_transaksi **foreign** **key** (kode\_buku) **references** buku (kode\_buku),

**add** **constraint** fk\_id\_petugas\_transaksi **foreign** **key** (id\_petugas) **references** petugas (id\_petugas);

**add** **constraint** fk\_nis\_transaksi **foreign** **key** (nis) **references** anggota (nis);

1. Index tabel “tmp”

Query index dibawah ini akan menghubungkan kolom kode\_ buku ke tabel buku.

Query :

**alter** **table** tmp

**add** **constraint** fk\_kode\_buku\_tmp **foreign** **key** (kode\_buku) **references** buku (kode\_buku);

1. Index tabel “pengembalian”

Di query ini berfungsi untuk menghubungkan kolom id\_transaksi dan id\_petugas ke tabel tujuan masing masing.

Query :

**alter** **table** pengembalian

**add** **constraint** fk\_id\_transaksi\_pengembalian **foreign** **key** (id\_transaksi) **references** transaksi (id\_transaksi),

**add** **constraint** fk\_id\_petugas\_pengembalian **foreign** **key** (id\_petugas) **references** petugas (id\_petugas);

## DML (Data Manipulation Language)

Adalah salah satu subbahasa dalam SQL (Sctructured Query Language) yang digunakan untuk mengelola data dalam database. DML digunakan untuk mengubah, menghapus, dan mengambil data dari tabel dalam database.

1. Halaman Users

Halaman ini bertugas untuk mengatur user login ke halaman website sistem manajemen perpustakaan

1. Query login dari tabel “petugas”

**select**

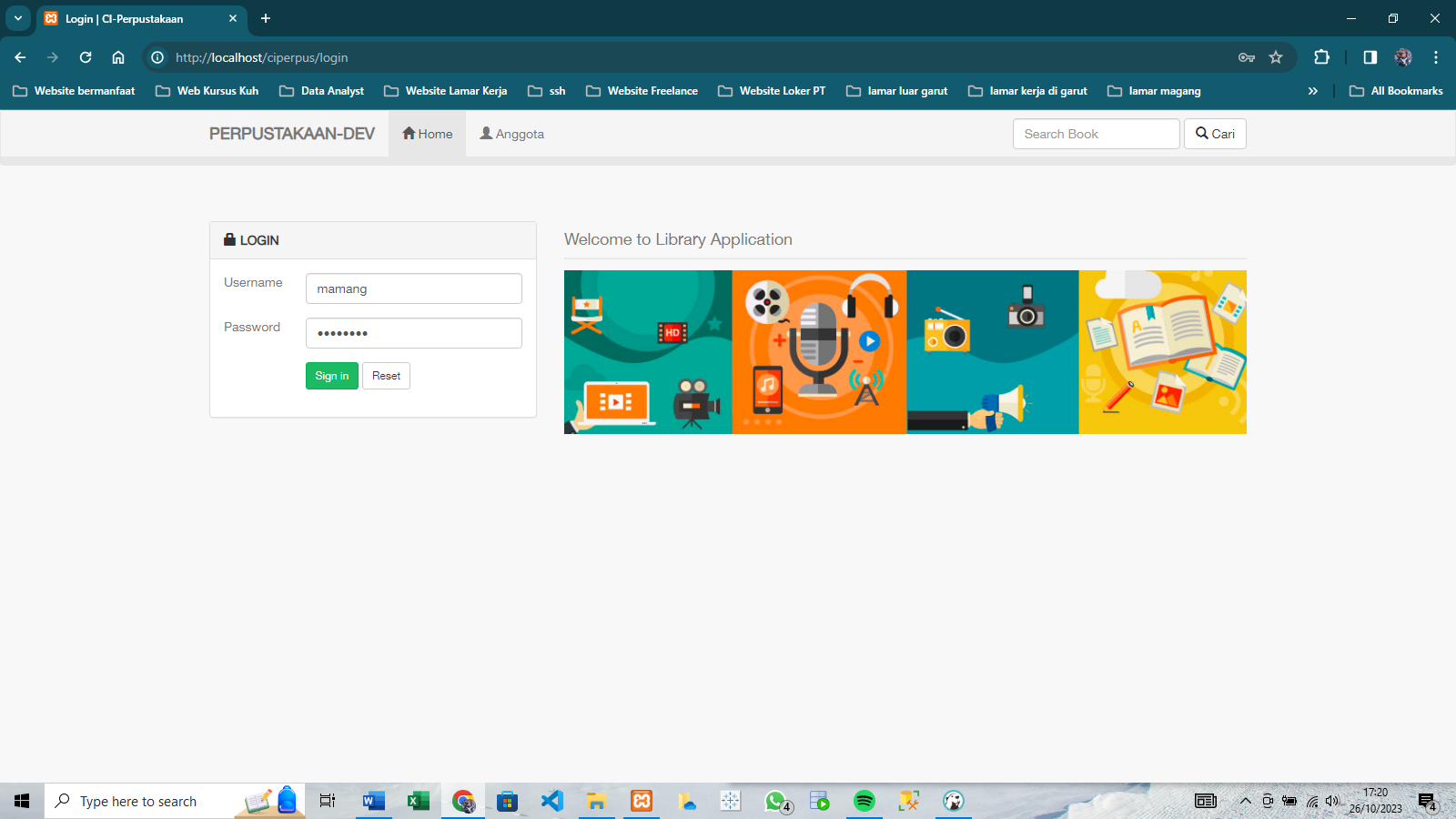
\*

**from** petugas

**where**

username = "mamang"

**and** **password** = "qwertyui";



1. Query menambahkan data ke tabel “petugas”

**insert**

**into**

petugas (username,

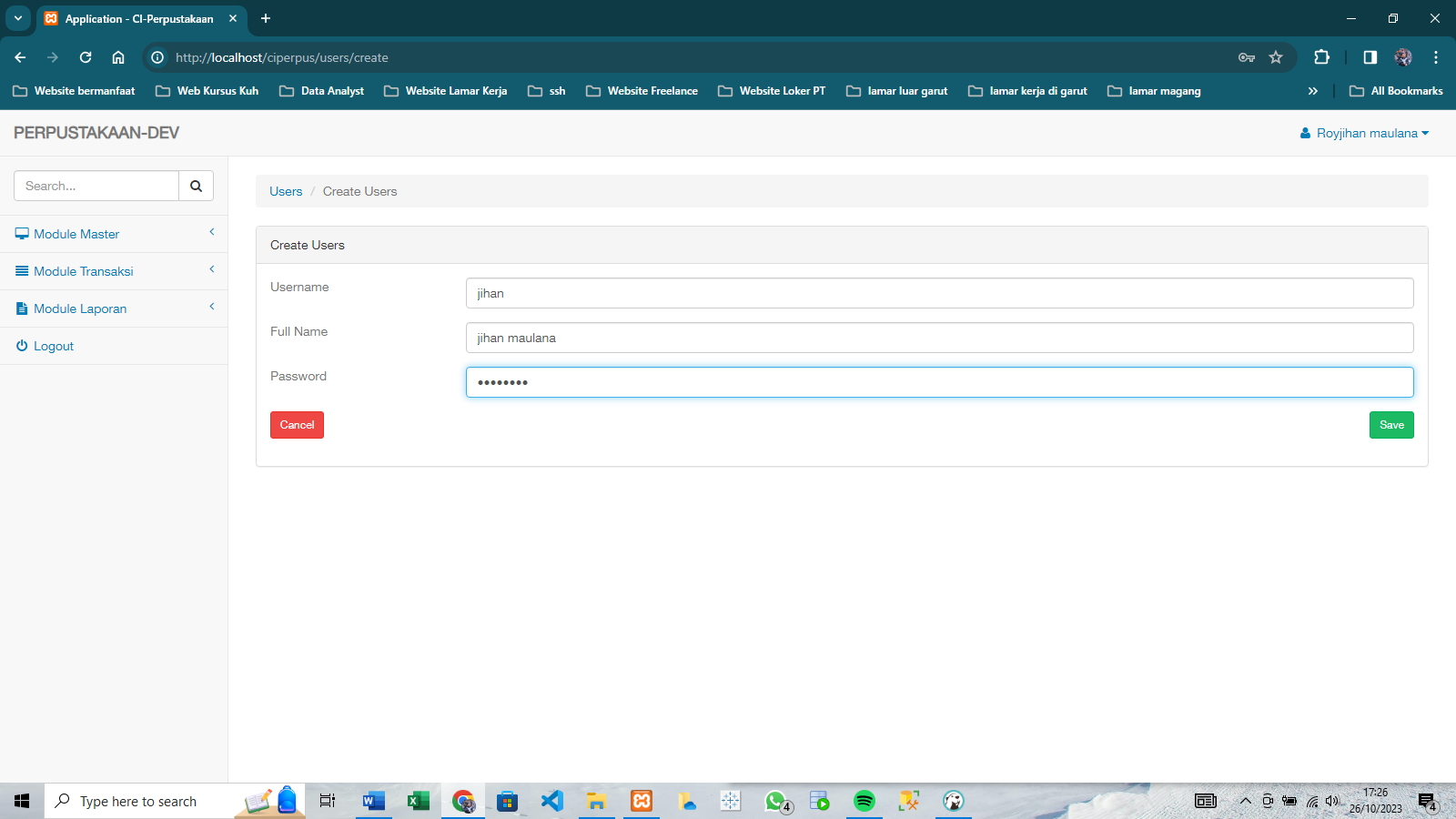
full\_name,

**password**)

**values** ("jihan",

"jihan maulana",

"qwertyui");

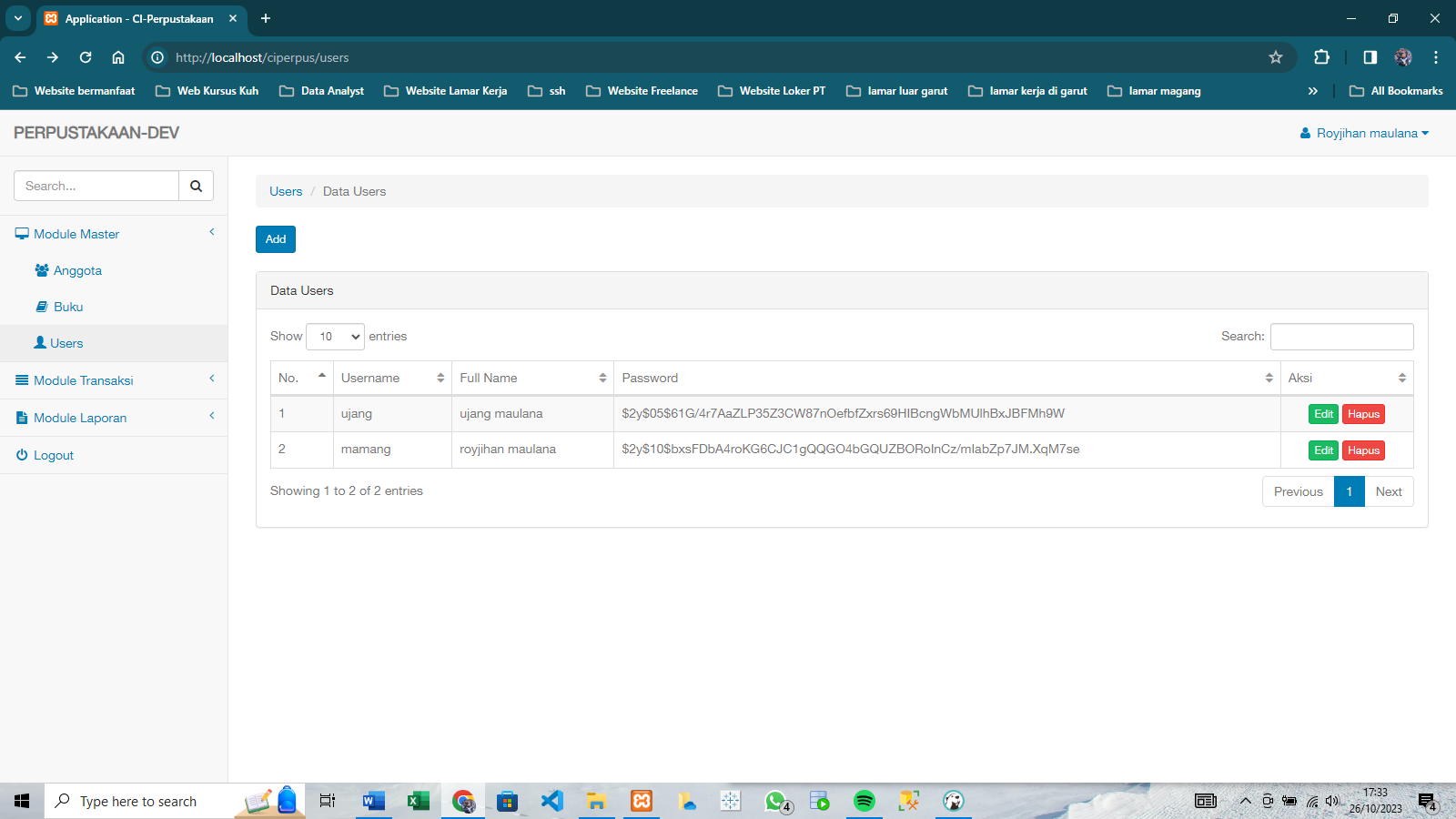


1. Query melihat semua data tabel “petugas”

**select**

\*

**from** petugas;



1. Query update data pada tabel “petugas”

**update**

petugas

**set**

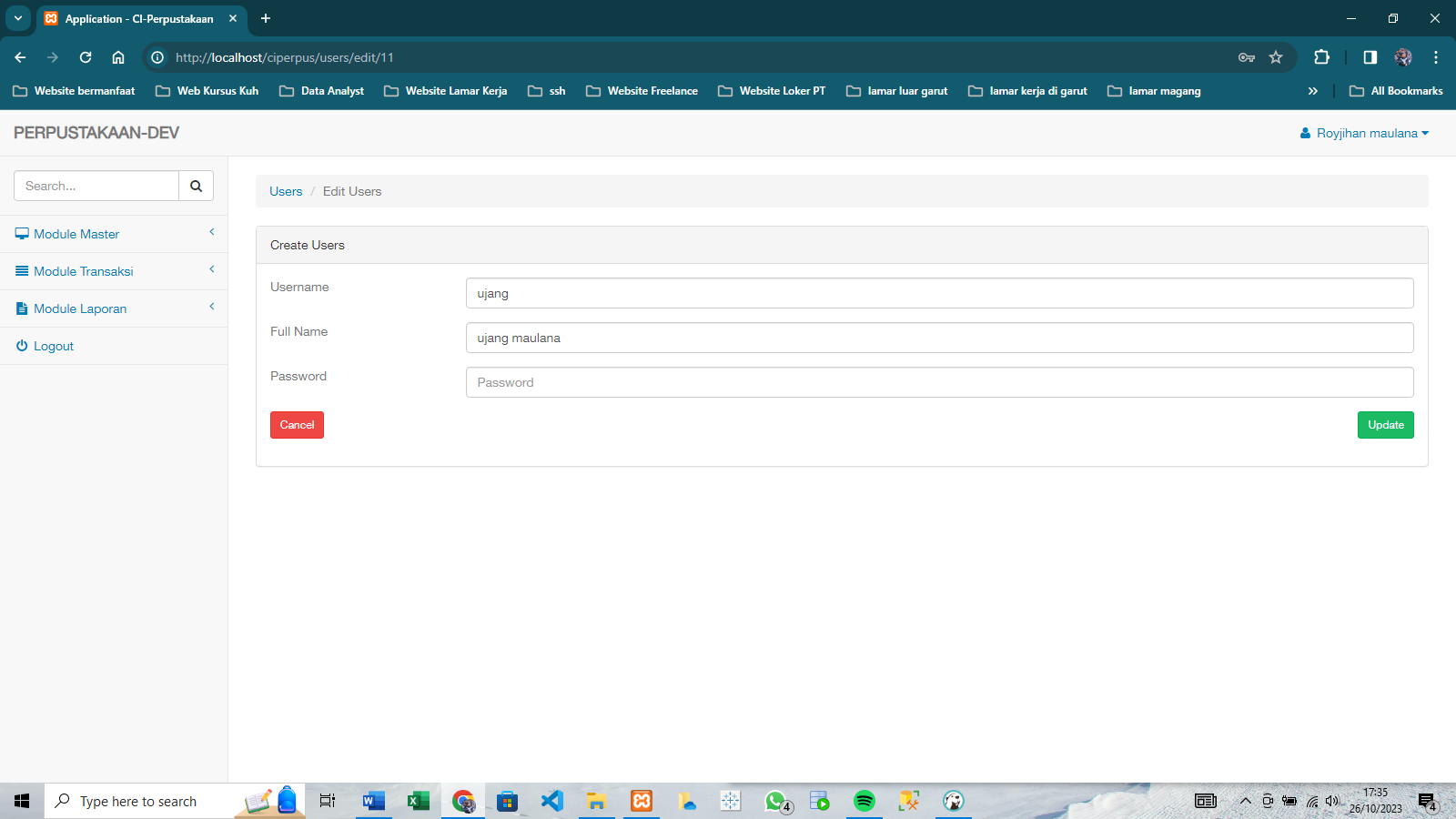
username = "ujang",

full\_name = "ujang maulana",

**password** = " "

**where**

id\_petugas = 11



1. Query menghapus data tabel “petugas”

**delete**

**from**

petugas

**where**

id\_petugas = 11;



1. Query mencari data pada tabel “petugas”

**select**

\*

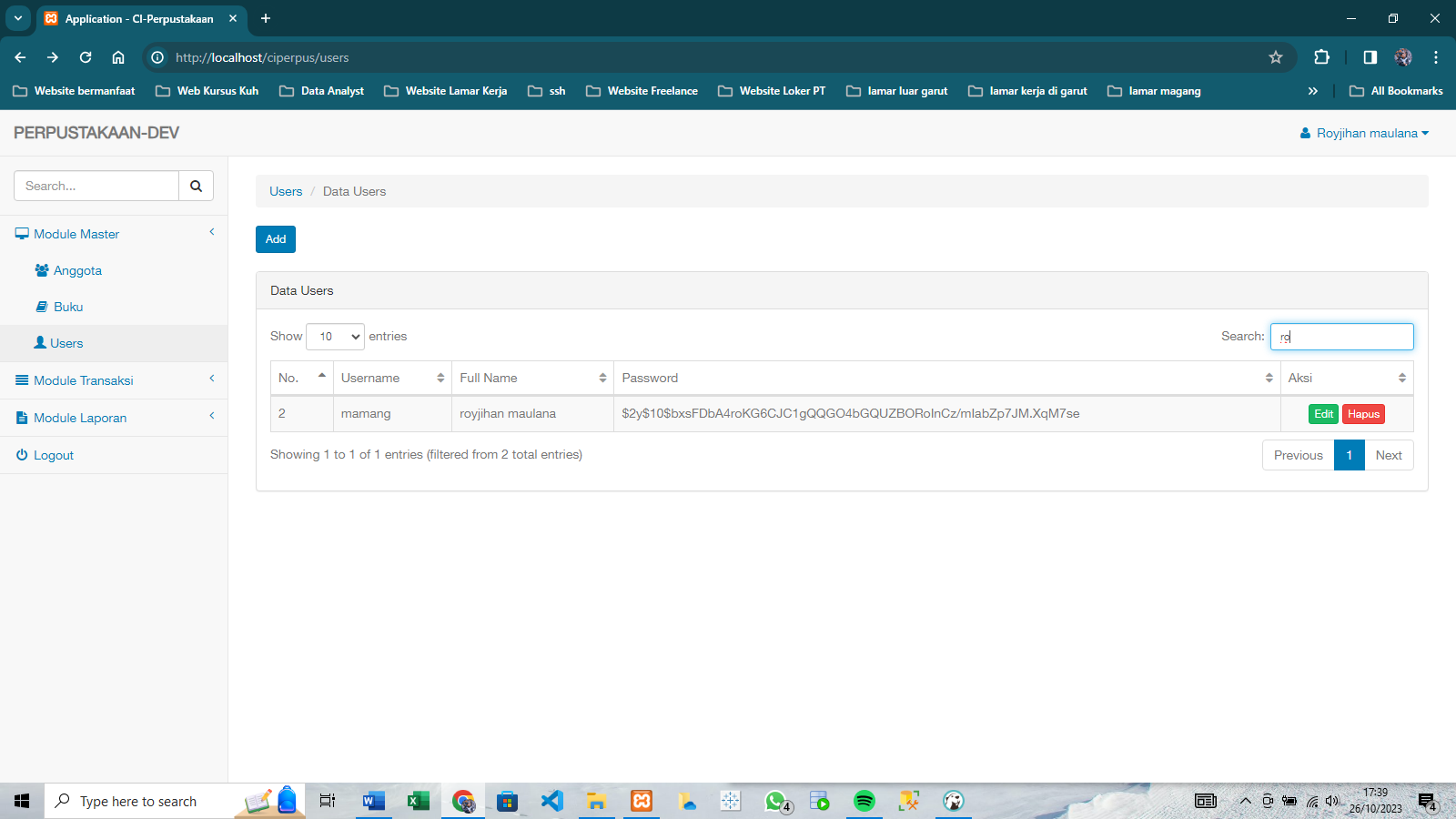
**from**

petugas

**where**

full\_name **like** "r%"

**or** username **like** "u%";



1. Halaman Anggota

Halaman ini berisi informasi semua anggota diperpustakaan yang bisa meminjam buku.

1. Query menambahkan data ke tabel “anggota”

**insert**

**into**

anggota (nis,

nama,

jk,

ttl,

kelas,

image)

**values** ("200137",

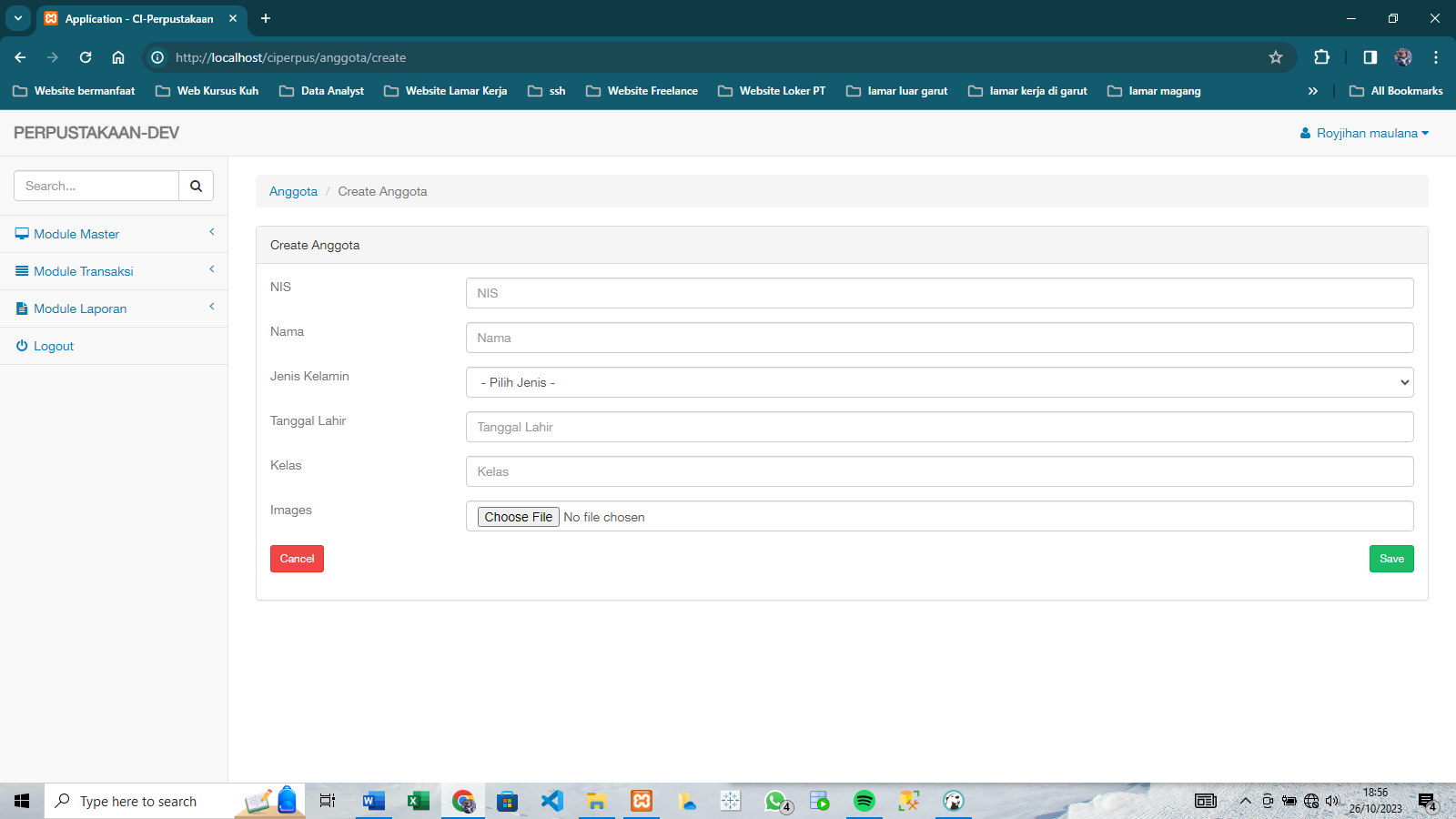
"siti silvi",

"P",

"2001-02-19",

"VII",

"siti.jpg");

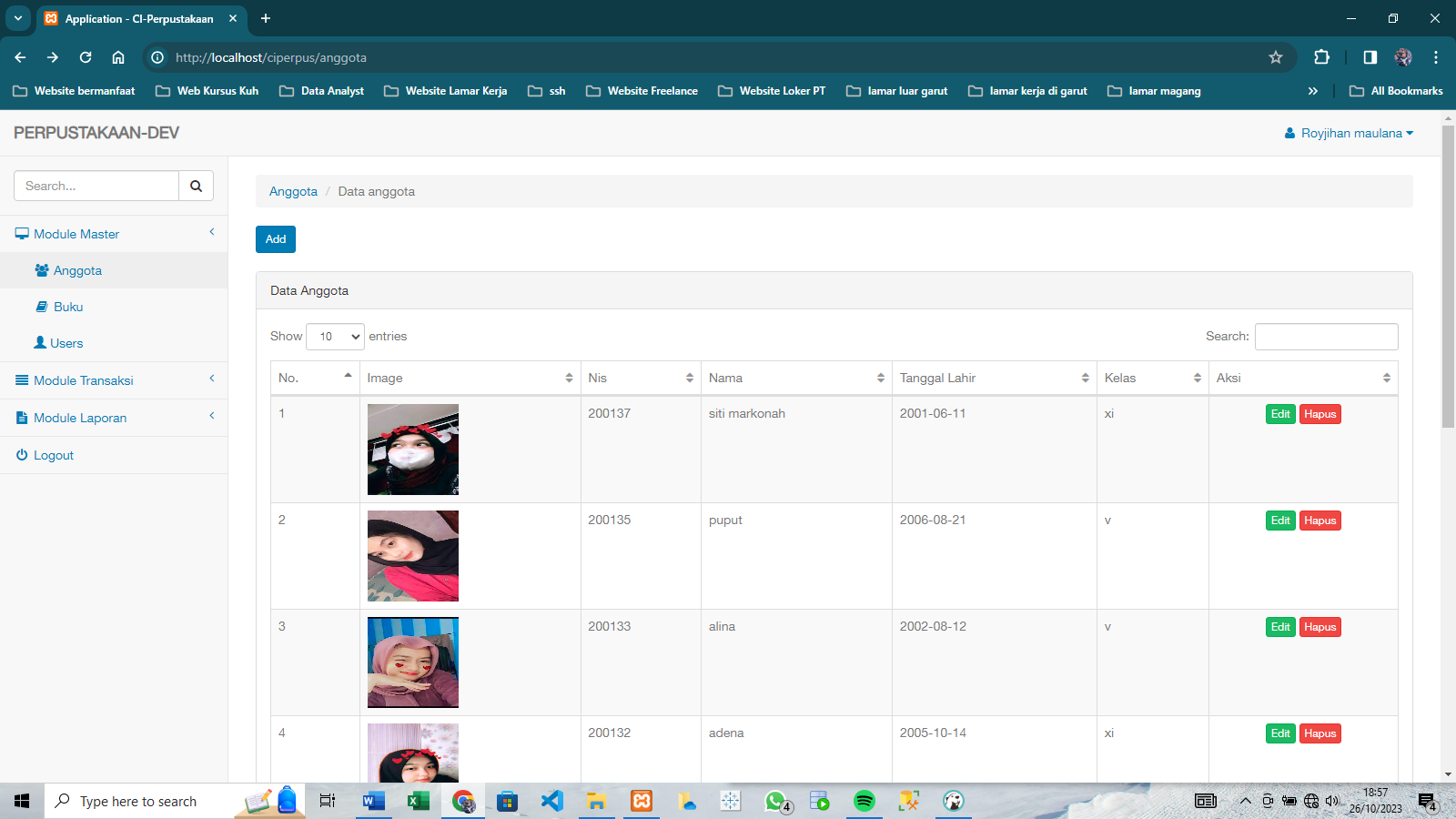


1. Query melihat data pada tabel “anggota”

**select**

\*

**from** anggota;



1. Query untuk update data pada tabel “anggota”

**update**

anggota

**set**

nama = "siti markonah",

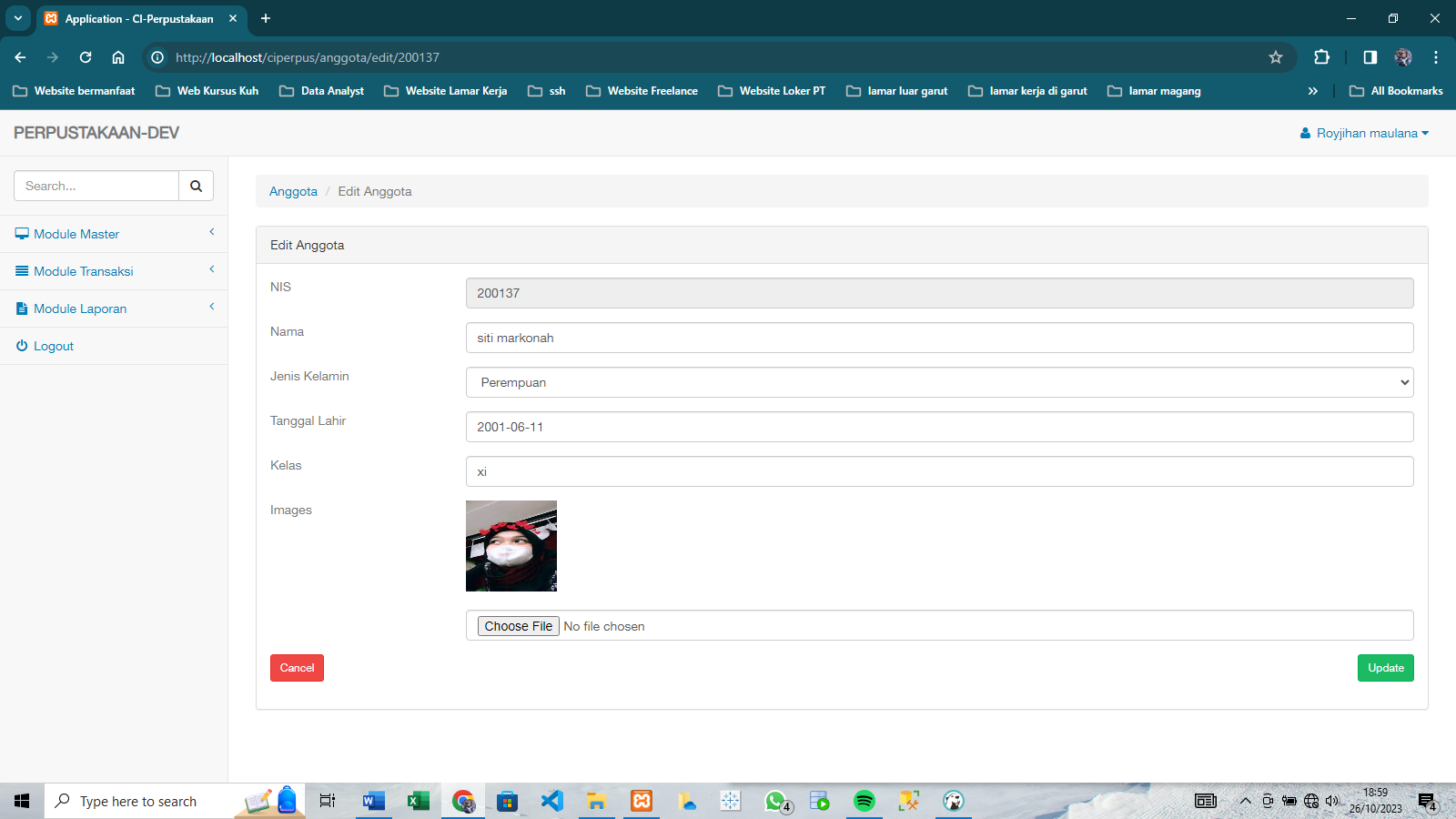
jk = "P",

ttl = "2001-06-11",

kelas = "xi", image = "siti.jpg"

**where**

nis = "200137";



1. Query untuk menghapus data pada tabel “anggota”

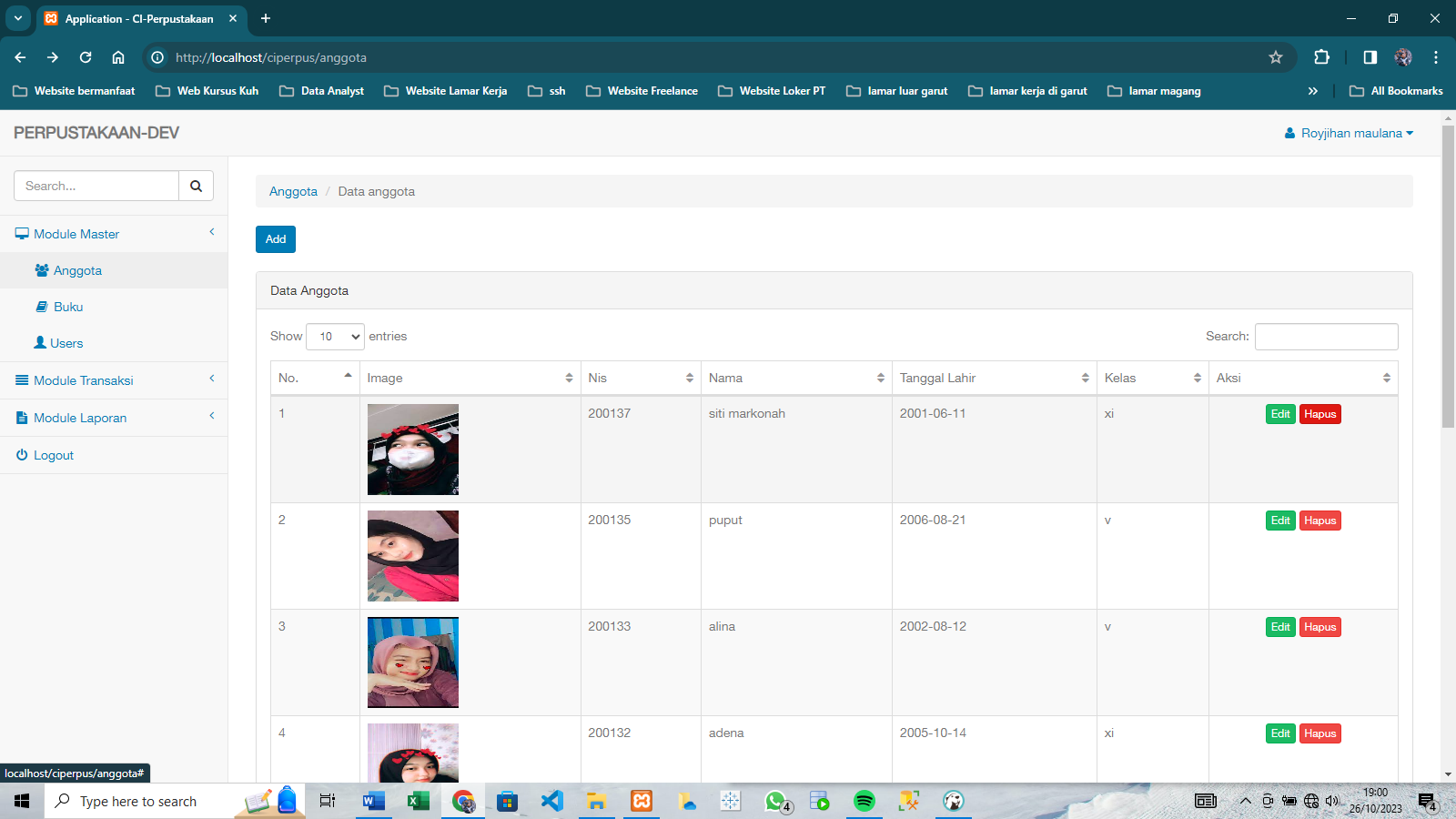
**delete**

**from**

anggota

**where**

nis = "200137";



1. Query mencari data pada tabel “anggota”

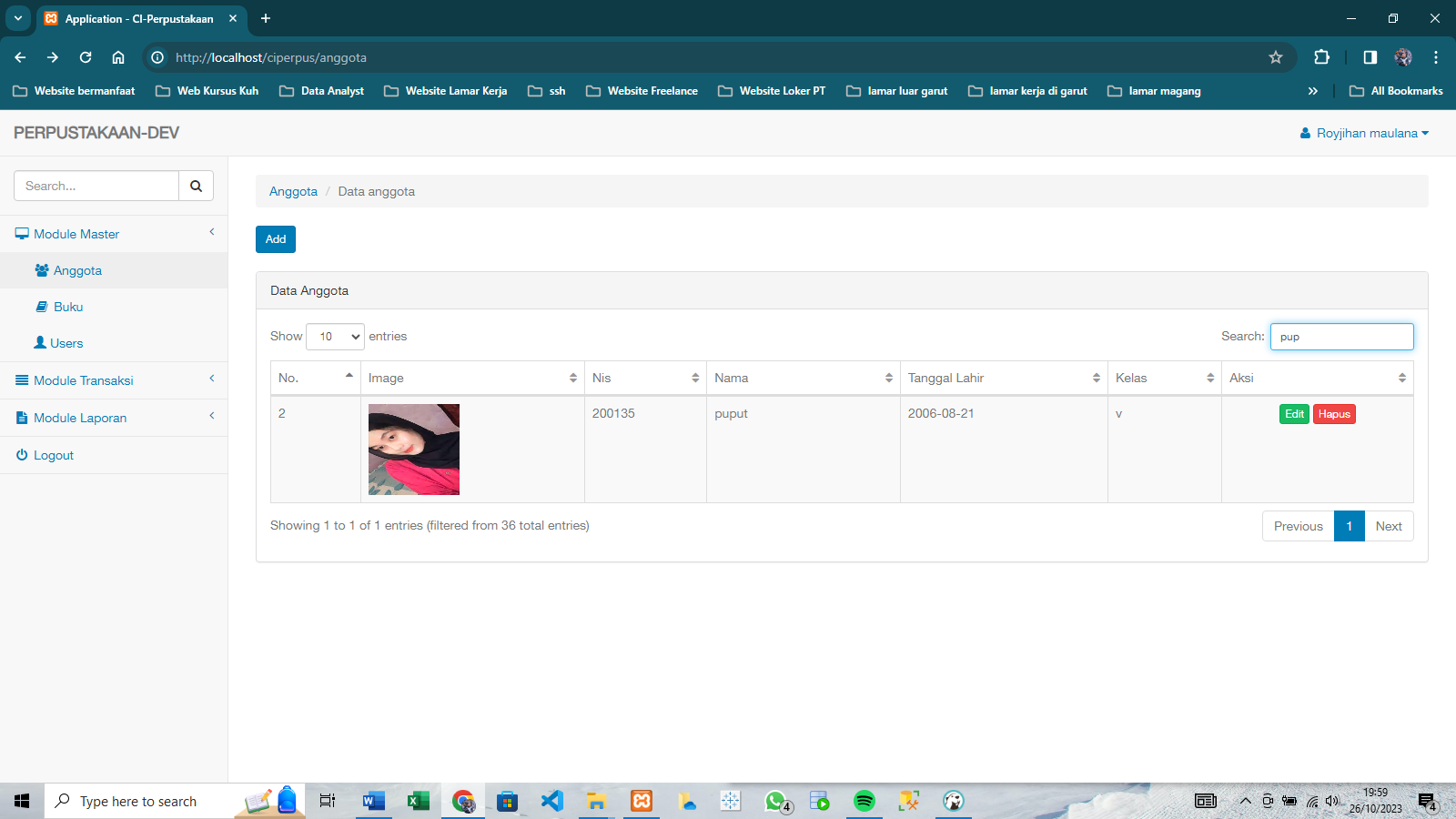
**select**

\*

**from** anggota

**where**

nama **like** "pup%";



1. Halaman Buku

Yang menampilkan daftar buku dan informasi mengenai buku tersebut yang ada di perpustakaan.

1. Query menambahkan data ke tabel "buku"

**insert**

**into**

buku (kode\_buku,

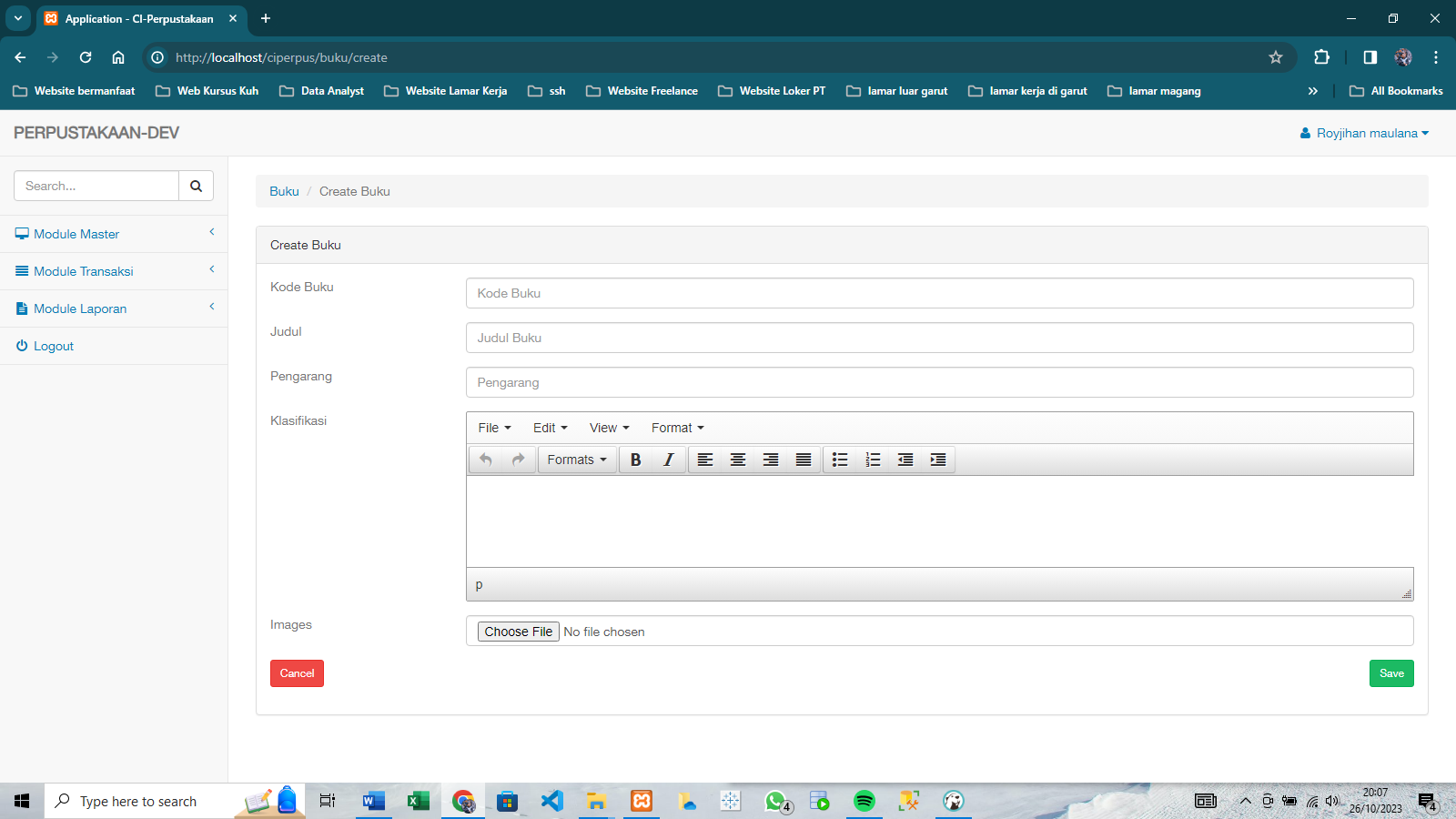
judul,

pengarang,

klasifikasi,

image)

**values** ("2004","PHP Beginner","jihan","<p>buku belajar website dari php</p>","php-advanced1.jpg");

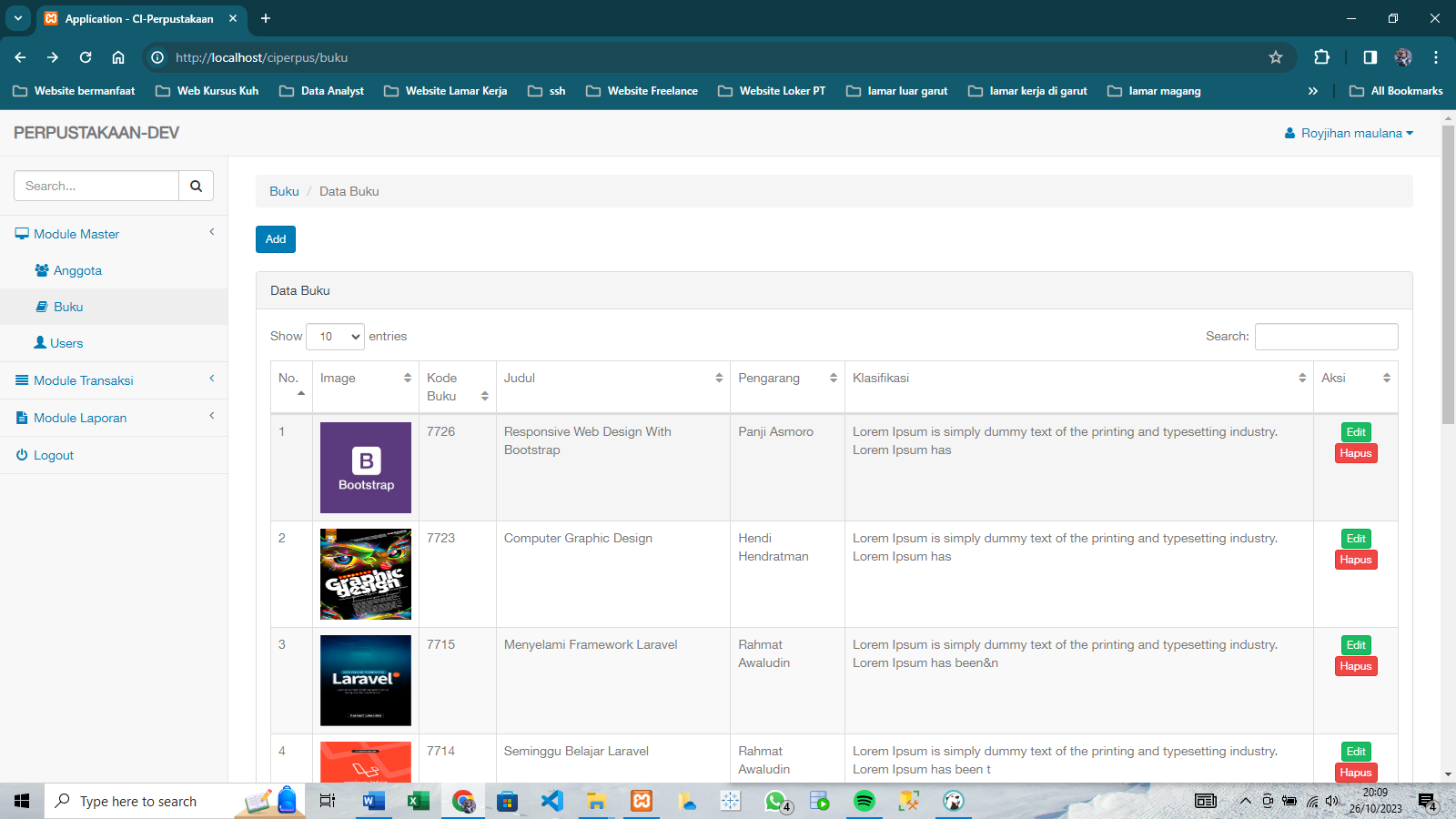


1. Query melihat data pada tabel “buku”

**SELECT**

\*

**FROM** buku;



1. Query untuk update data pada tabel “buku”

**update**

buku

**set**

judul = "PHP Untuk Pemula",

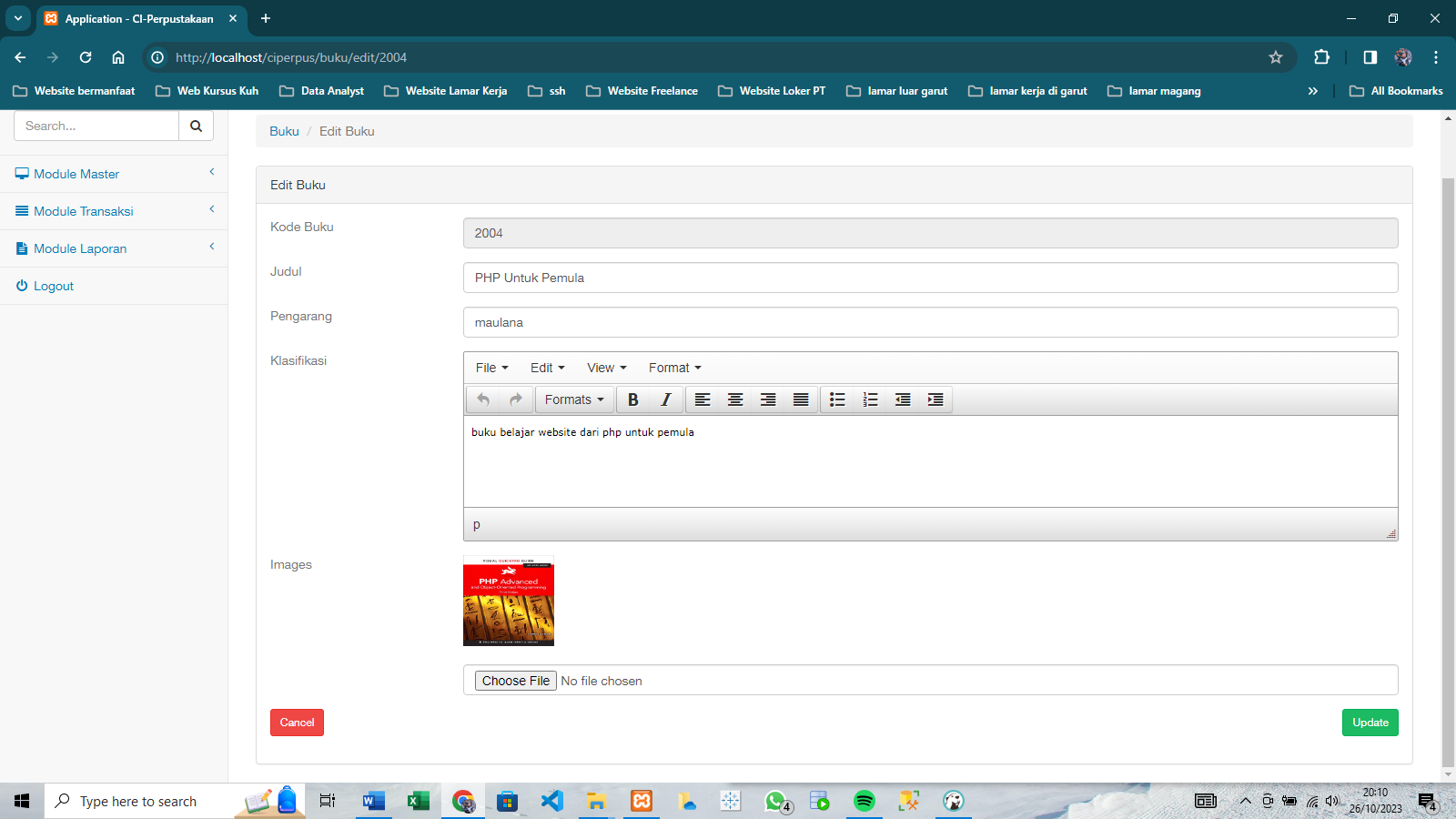
pengarang = "maulana",

klasifikasi = "<p>buku belajar website dari php untuk pemula</p>",

image = "php-advanced1.jpg"

**where**

kode\_buku = "2004";



1. Query untuk menghapus data pada tabel “buku”

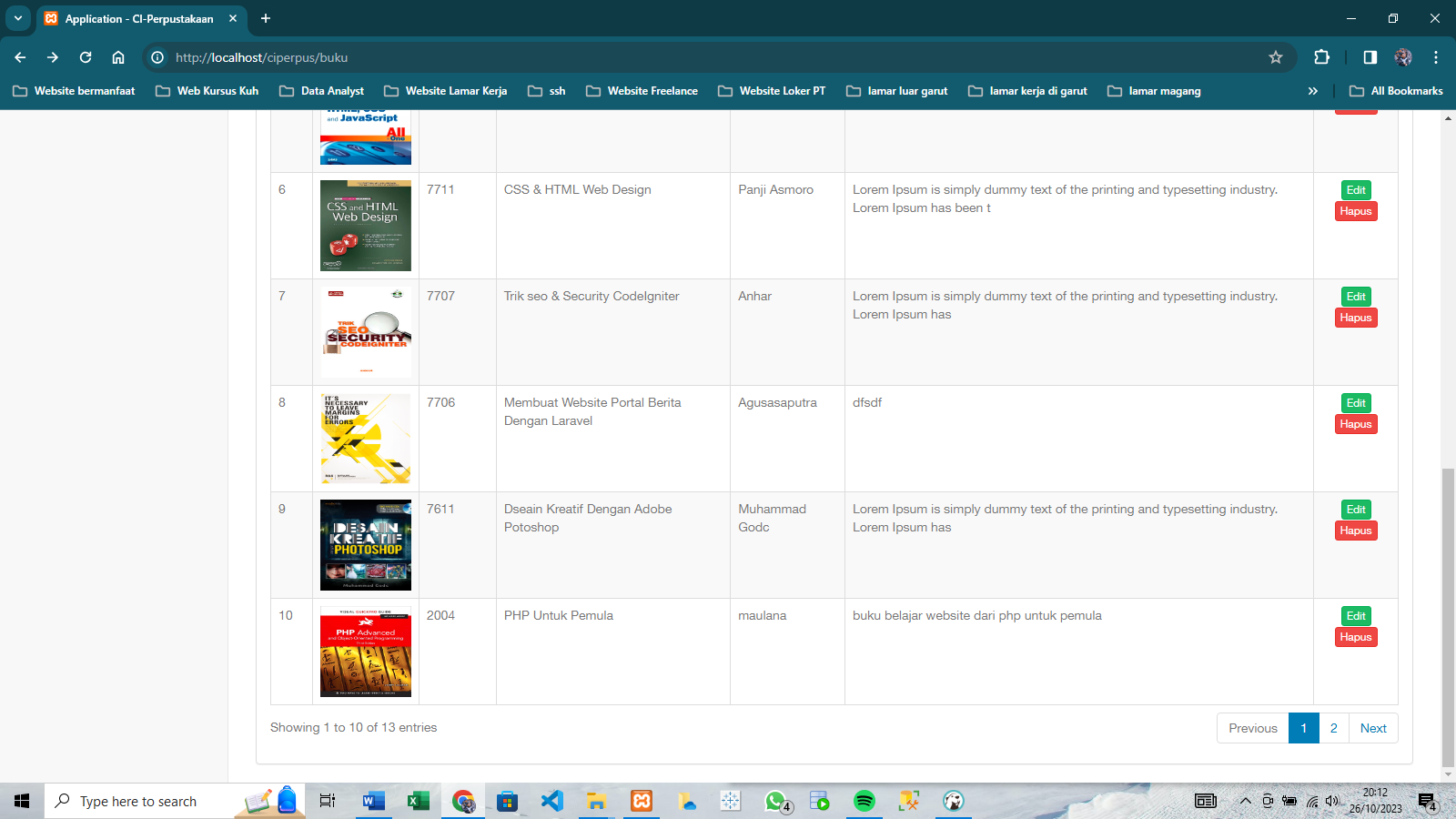
**delete**

**from**

buku

**where**

kode\_buku = "2004";



1. Query mencari data pada tabel “buku”

**select**

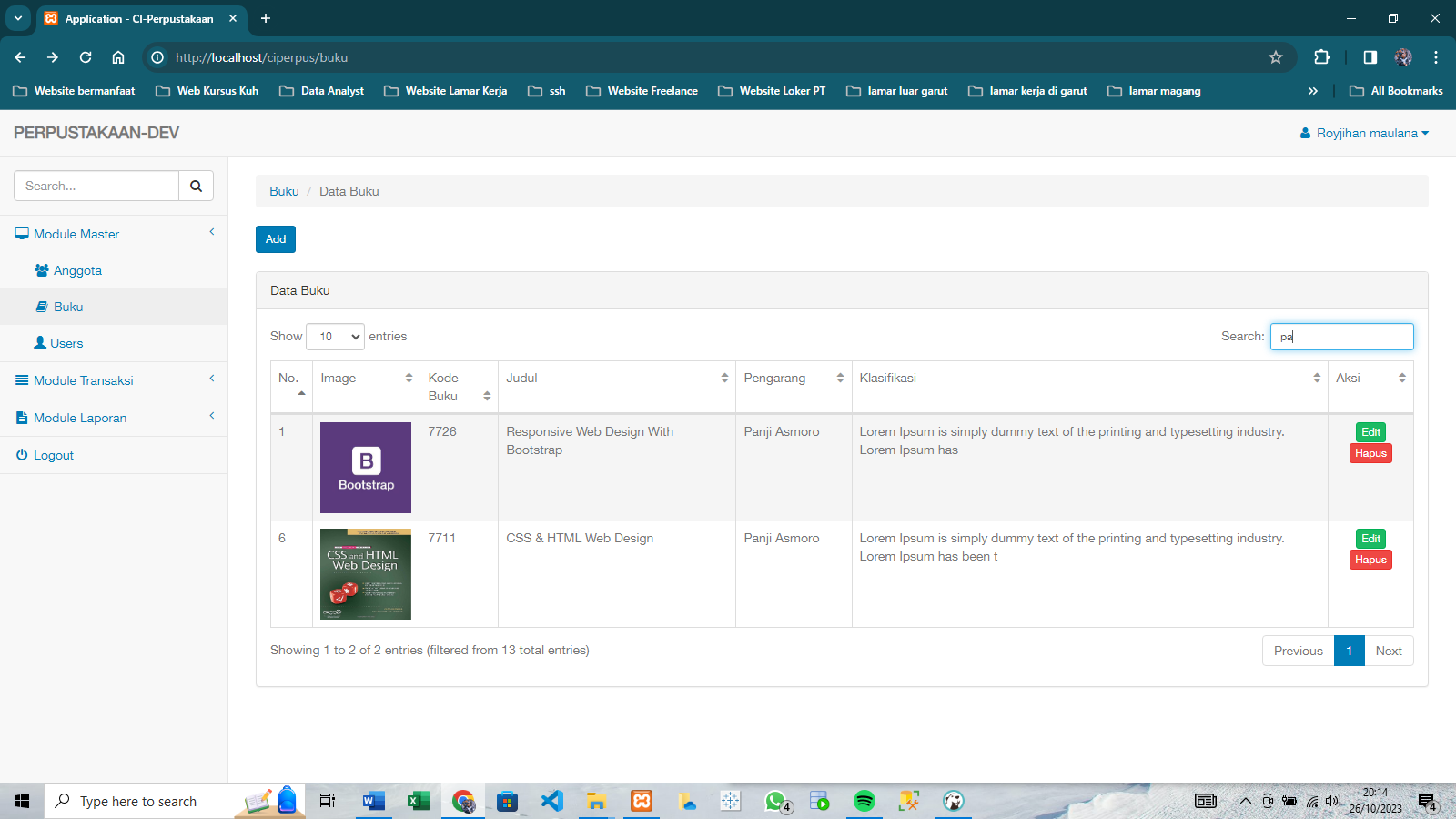
\*

**from** buku

**where**

judul **like** "p%"

**or** pengarang **like** "w%";



1. Halaman Peminjaman

Halaman ini bertugas untuk menyimpan data dari tiap anggota yang mau meminjam buku dari perpustakaan.

1. Query mencari data dari tabel “anggota”

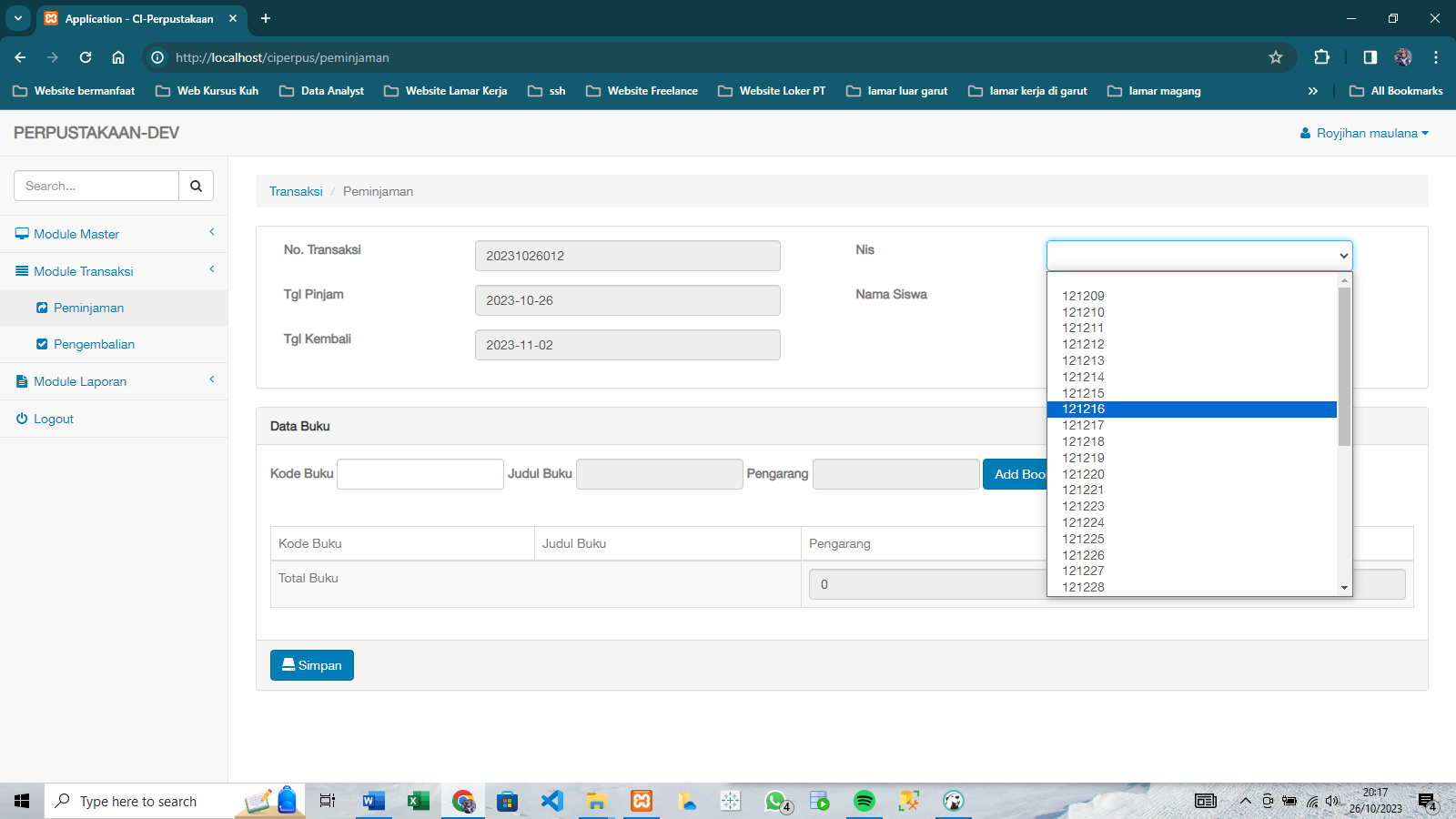
**select**

nis,

nama

**from**

anggota;



1. Query memilih data dari tabel “anggota”

**select**

nis,

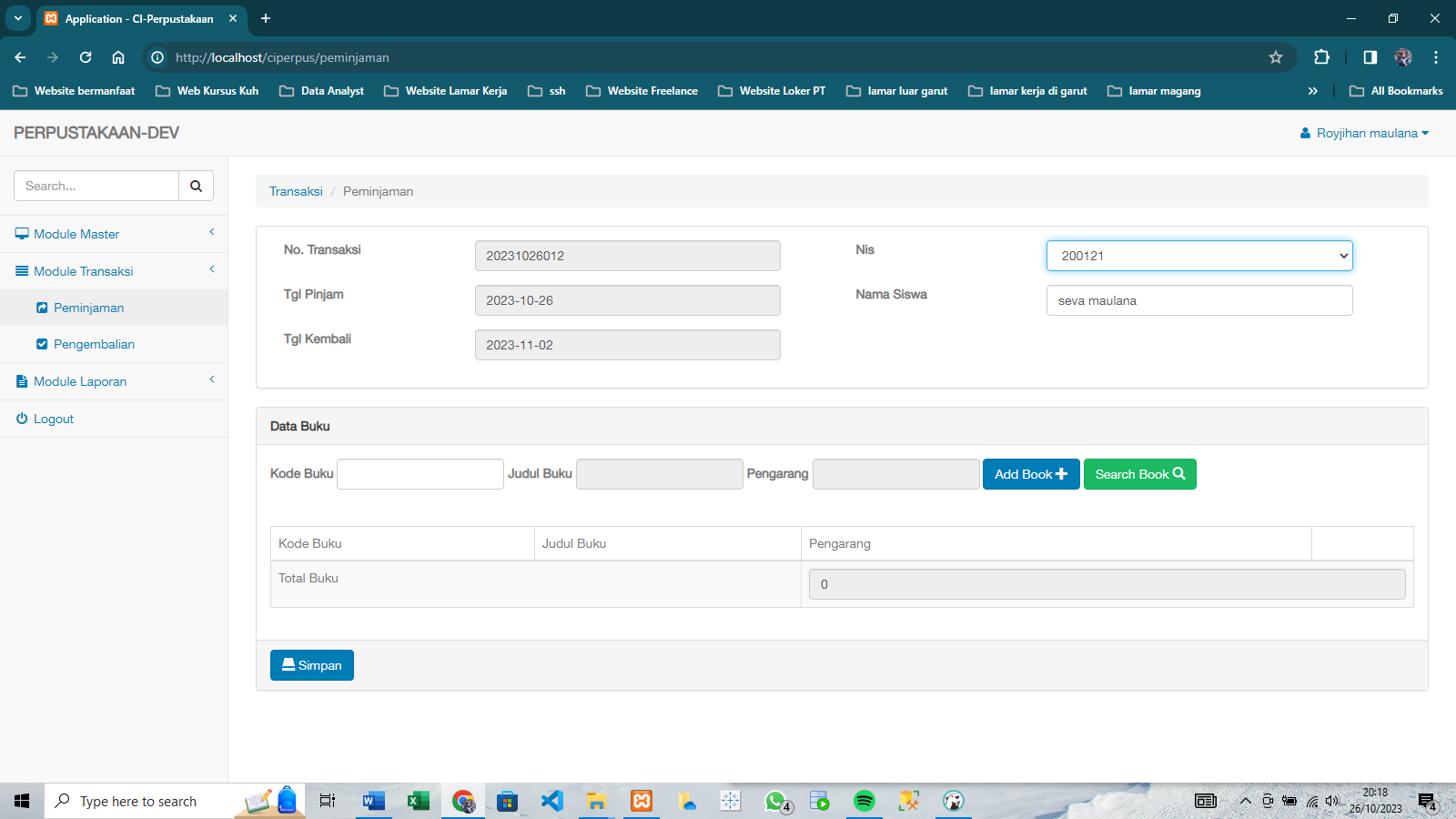
nama

**from**

anggota

**where**

nis = "200121";



1. Query untuk mencari data dari tabel “buku”

**select**

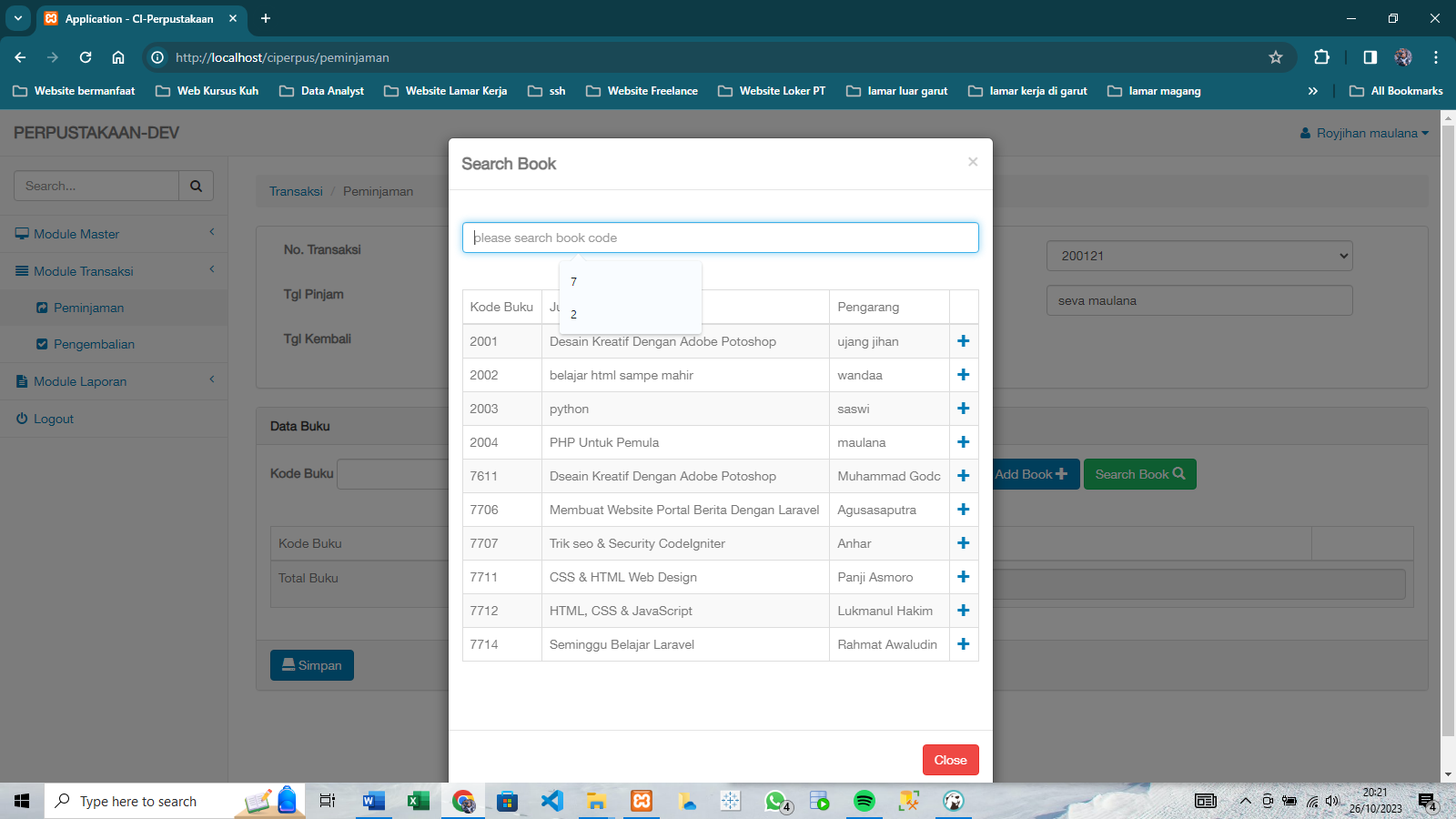
kode\_buku,

judul,

pengarang

**from**

buku **where** kode\_buku **like** "%";



1. Query memilih data dari tabel “buku”

**select**

kode\_buku,

judul,

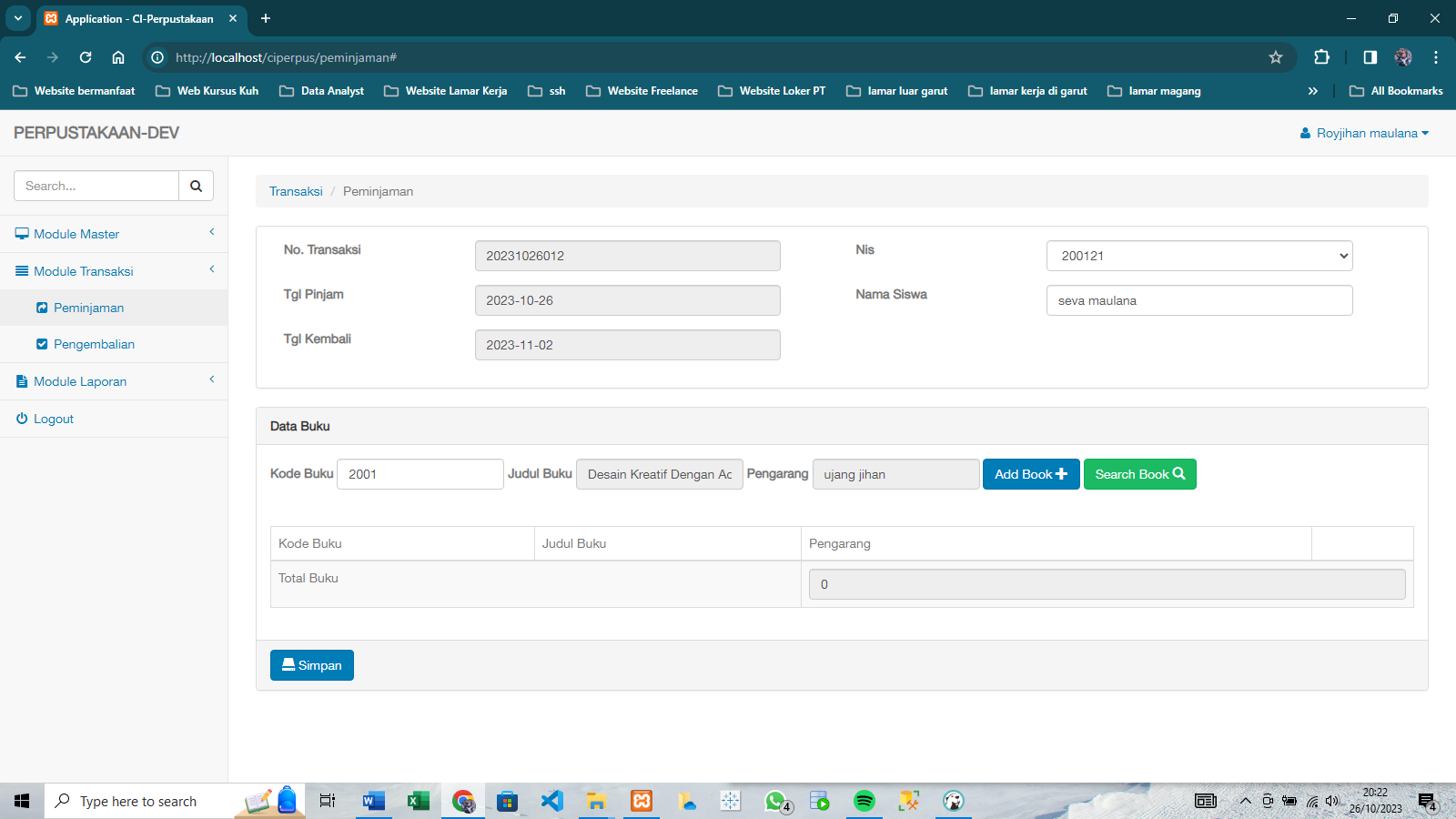
pengarang

**from**

buku

**where**

kode\_buku = "2001";



1. Query mencari data pada tabel “buku”

**insert**

**into**

tmp (kode\_buku,

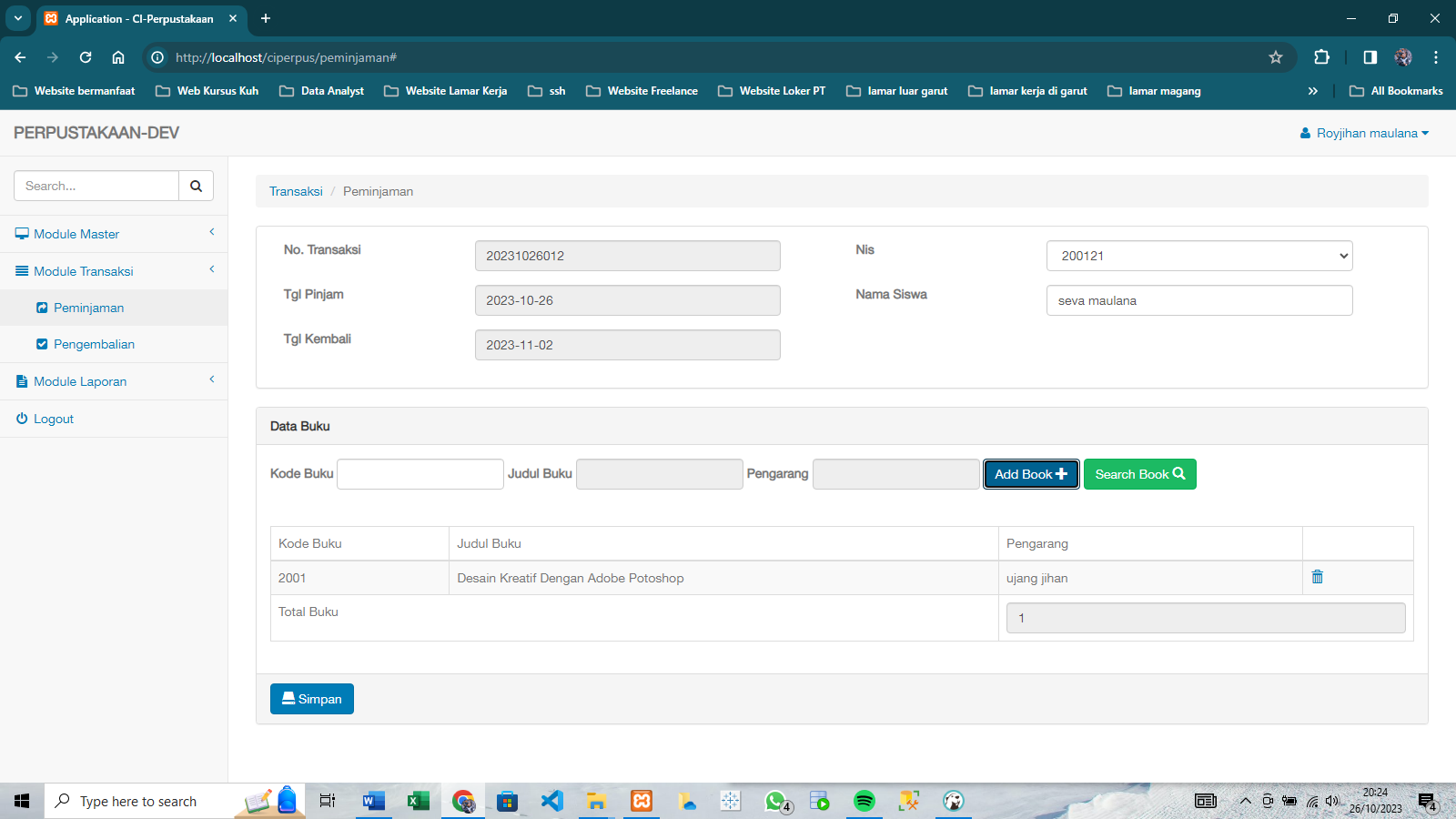
judul,

pengarang)

**values** ("2001",

"Desain Kreatif Dengan Adobe Potoshop",

"ujang jihan");



1. Query menambahkan data untuk tabel “transaksi”

**insert**

**into**

transaksi (id\_transaksi,

nis,

kode\_buku,

tanggal\_pinjam,

tanggal\_kembali,

status,

id\_petugas)

**values**

("20231026011","200121","2001","2023-10-26","2023-11-02","N",10);

1. Halaman Pengembalian

Halaman ini bertujuan untuk mencatat semua transaksi pengembalian buku dari tiap anggota perpustakaan.

1. Query mencari data dari tabel transaksi

**select**

id\_transaksi,

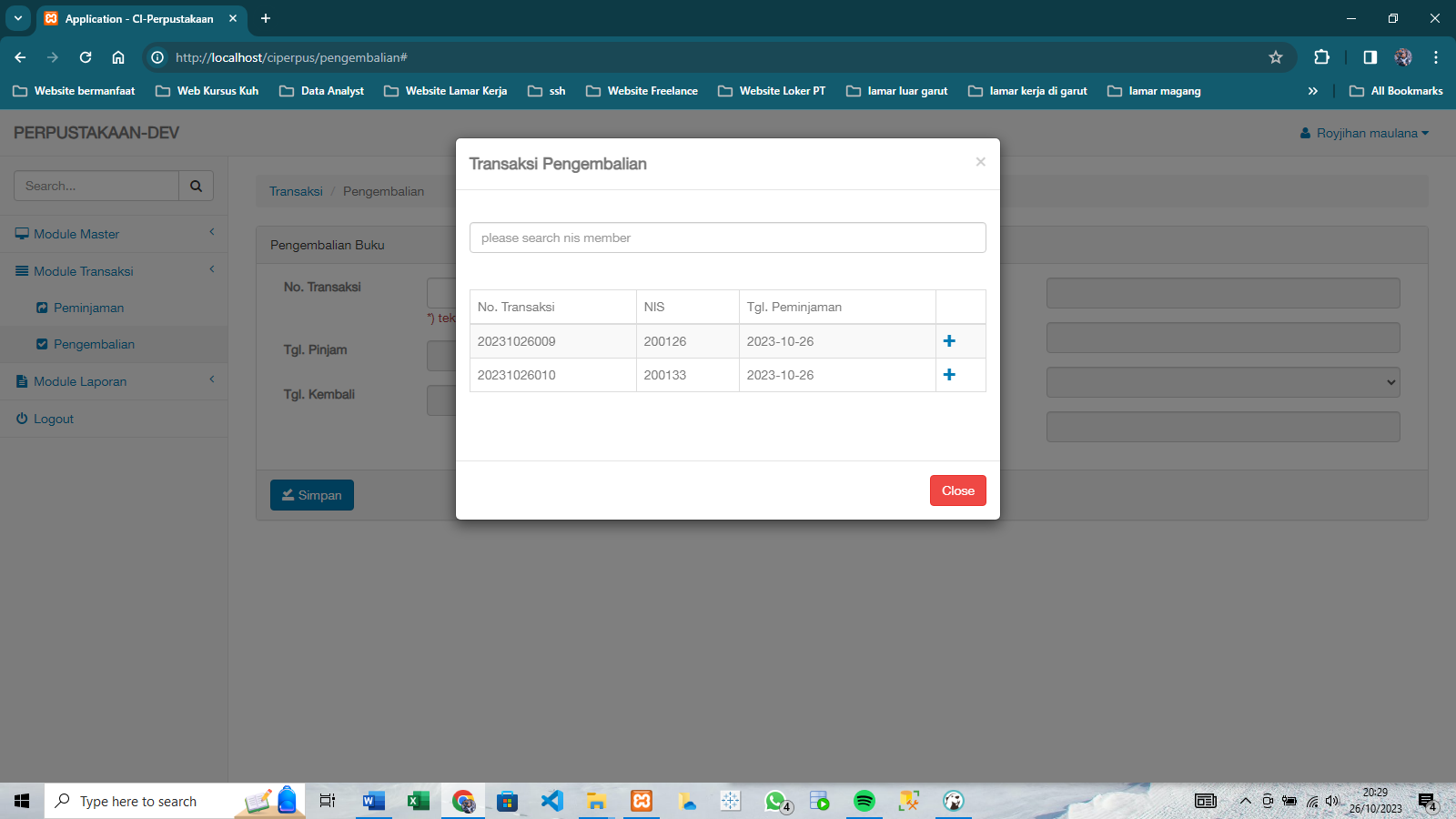
nis,

tanggal\_pinjam

**from**

transaksi

**where** status = "N" **or** nis = "%";



1. Query menampilkan data dari tabel “transaksi dan anggota”

**select**

t.id\_transaksi,

t.tanggal\_kembali,

t.tanggal\_pinjam,

t.nis,

a.nama,

b.kode\_buku,

b.judul,

b.pengarang

**from**

transaksi t

**inner** **join** anggota a **on**

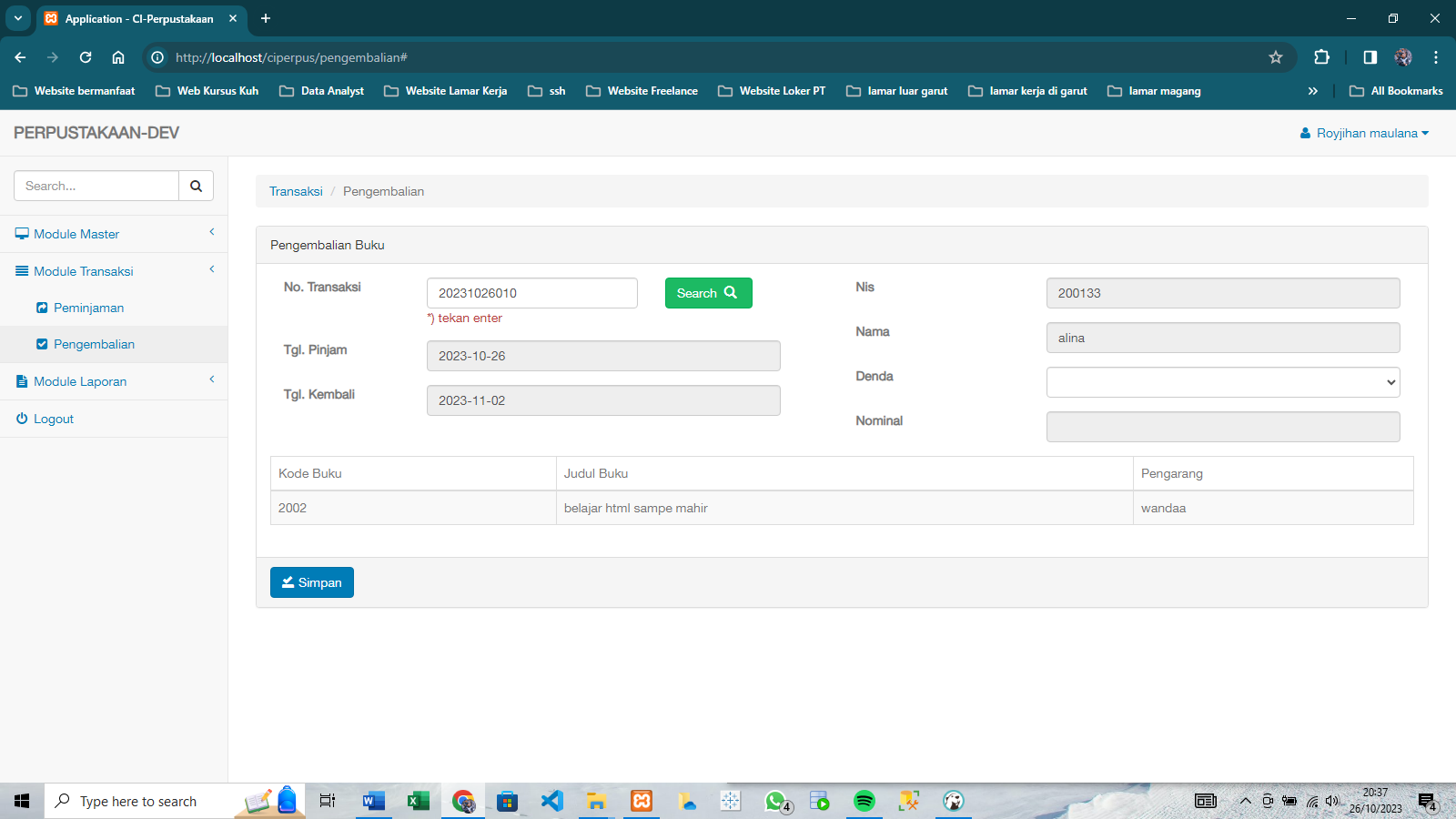
t.nis = a.nis

**inner** **join** buku b **on**

b.kode\_buku = t.kode\_buku

**where**

id\_transaksi = "20231026010";



1. Query untuk memasukan data ke tabel “pengembalian” sekaligus update data tabel “transaksi”

**insert**

**into**

pengembalian (id\_transaksi,

tgl\_pengembalian,

denda,

nominal,

id\_petugas)

**values** ("20231026011",

"2023-10-26",

"N",

"0",

10);

-- update table transaksi

**update**

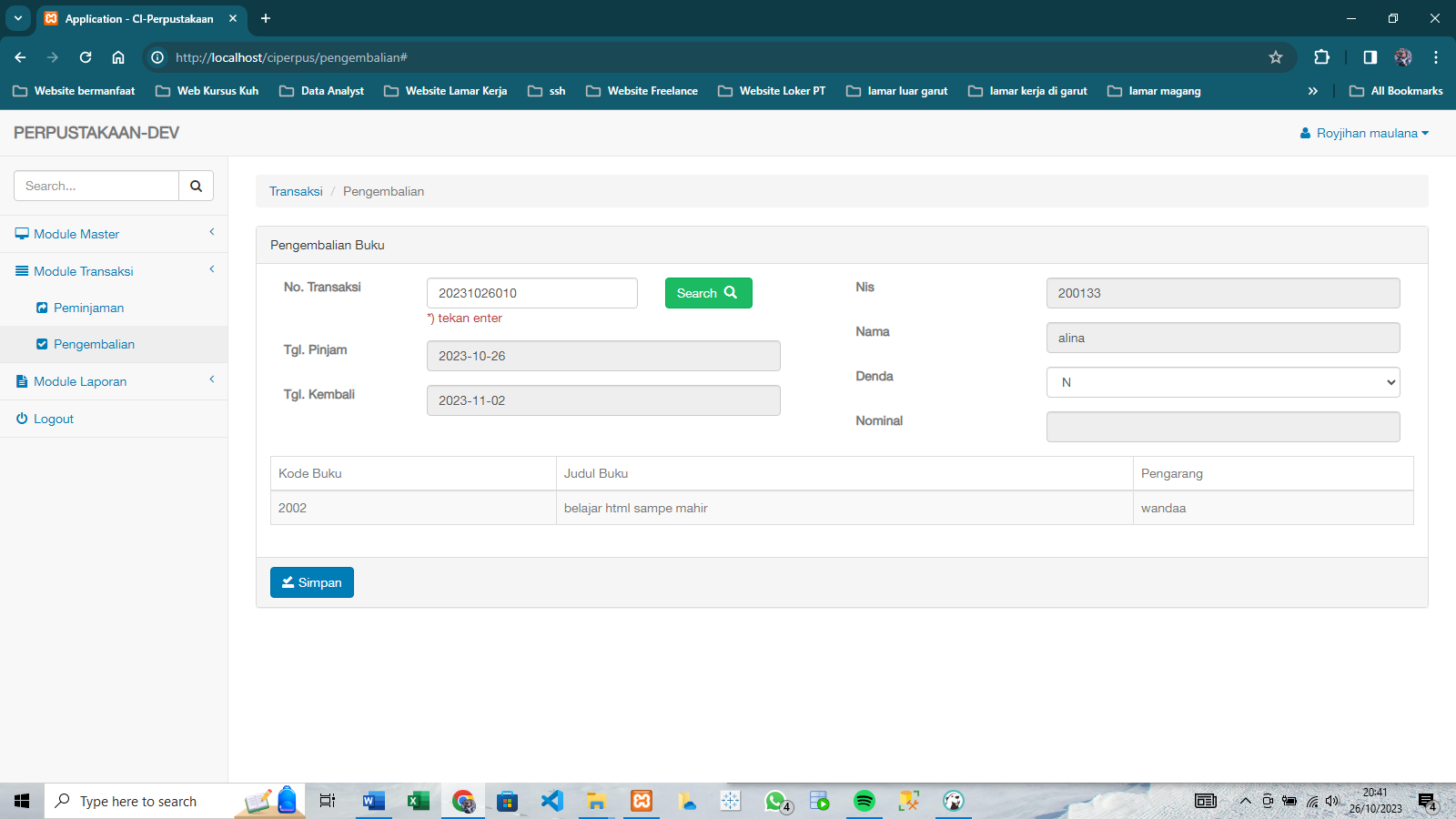
transaksi

**set**

status = "Y"

**where**

id\_transaksi = "20231026011";



1. Halaman Data Peminjaman

Pada halaman ini akan menampilkan semua data peminjaman buku lewat tanggal yang telas di tentukan.

1. Query untuk melihat data pada tabel “transaksi”

**select**

id\_transaksi,

tanggal\_pinjam,

tanggal\_kembali,

status,

nis

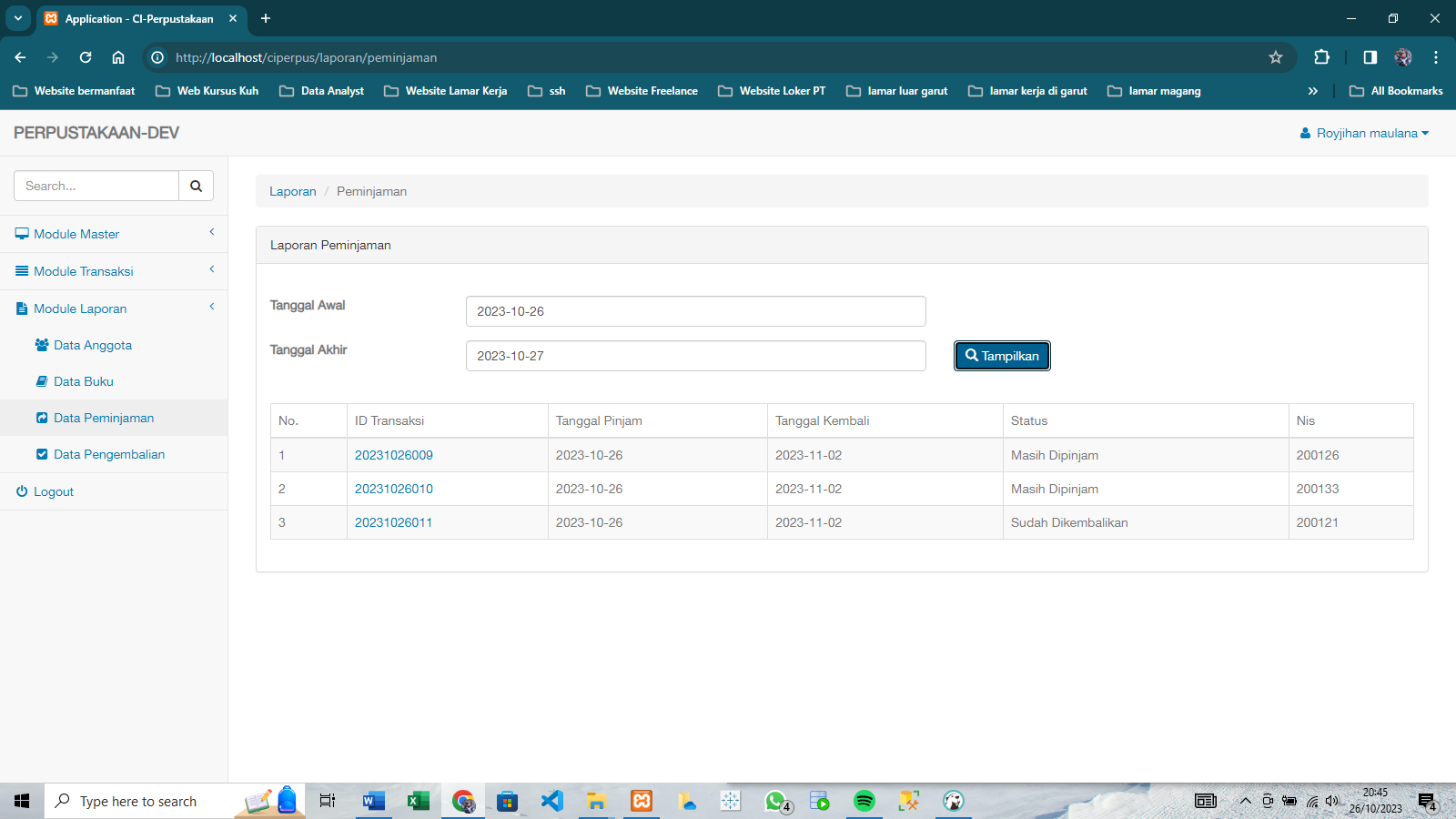
**from**

transaksi

**where**

tanggal\_pinjam **between** "2023-10-25" **and** "2023-10-27"

**or** tanggal\_kembali **between** "2023-10-25" **and** "2023-10-27";



1. Query melihat data pada tabel “transaksi dan buku”

**select**

t.id\_transaksi,

t.tanggal\_pinjam,

t.nis,

t.status,

b.kode\_buku,

b.judul,

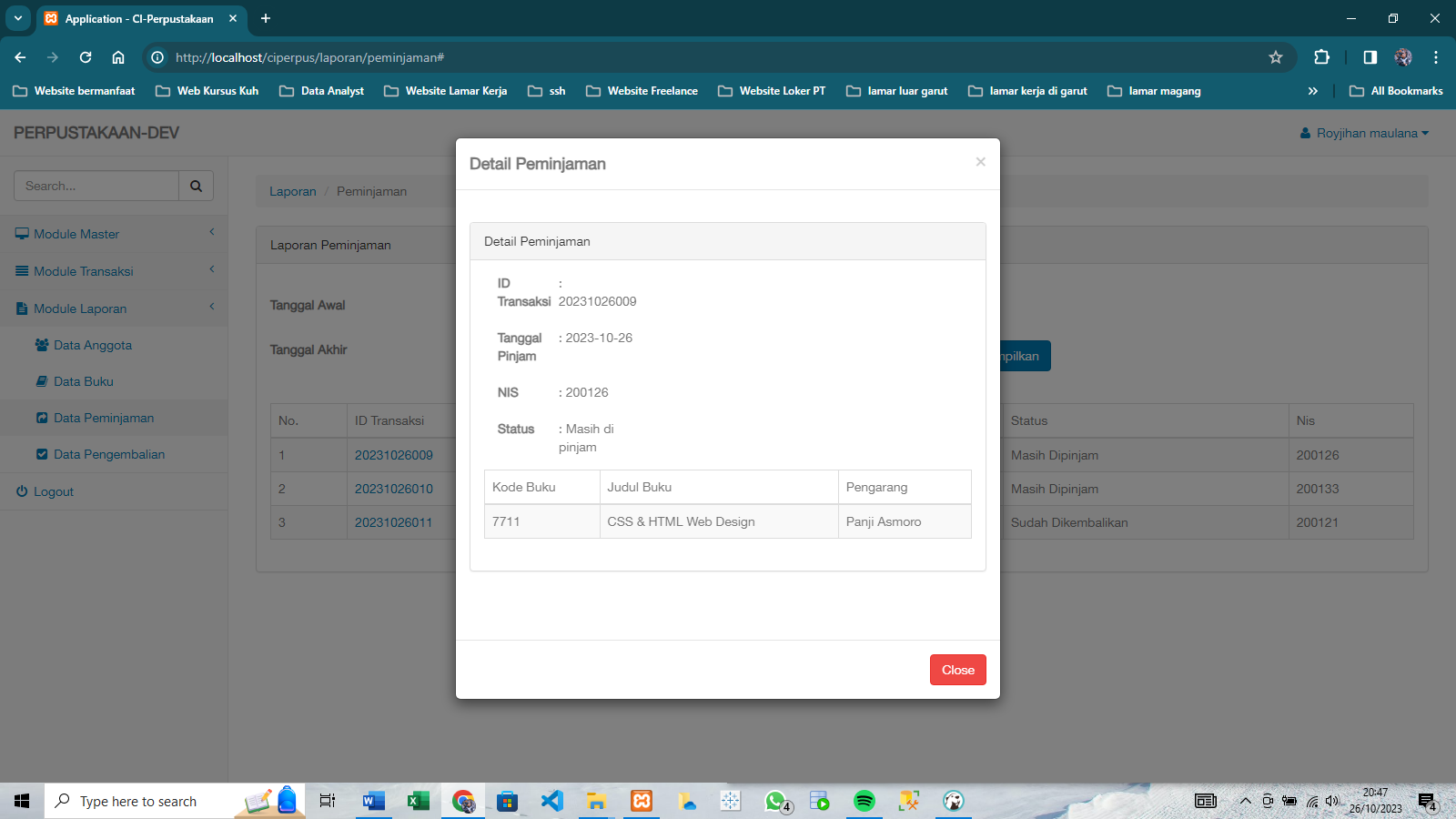
b.pengarang

**from**

transaksi t **inner** **join** buku b **on** t.kode\_buku = b.kode\_buku

**where**

id\_transaksi = "20231026009";



1. Halaman Data Pengembalian

Dihalaman data pengembalian ini akan menampilkan semua data transaksi pengembalian buku lewat tanggal yang telah ditentukan.

1. Query melihat data pada tabel “pengembalian”

**select**

id\_transaksi,

tgl\_pengembalian,

denda,

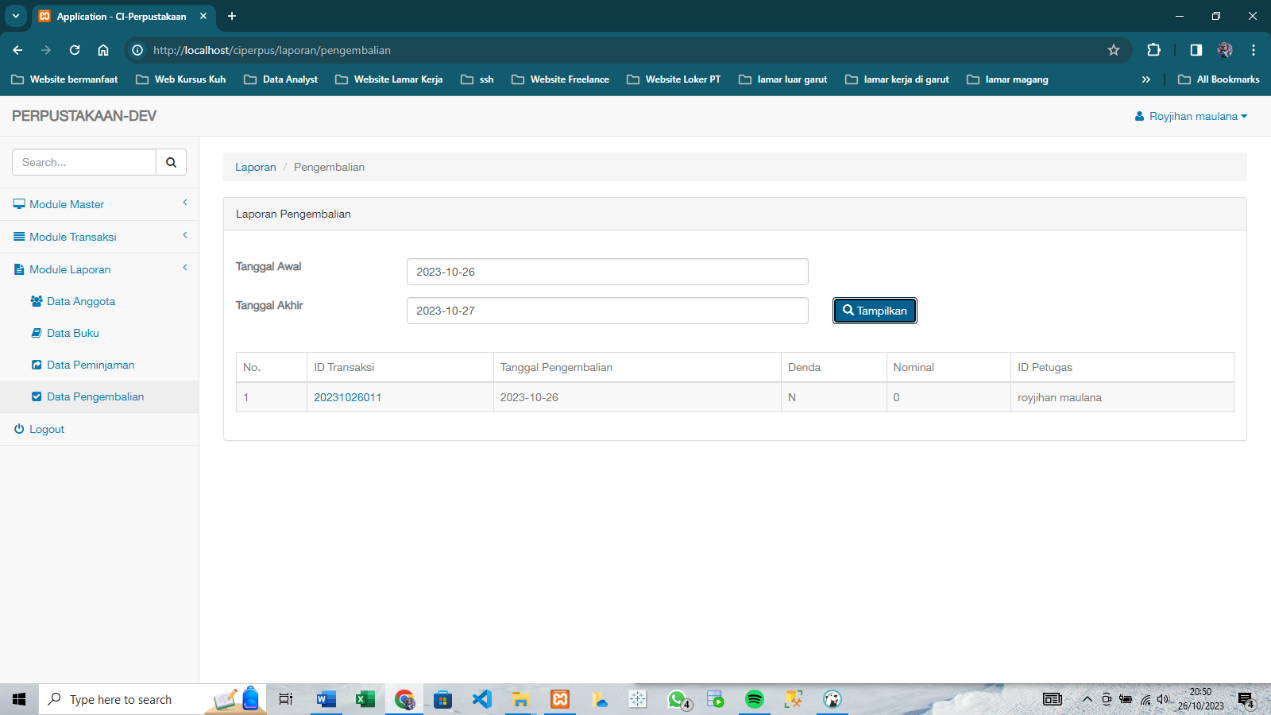
nominal

**from**

pengembalian

**where**

tgl\_pengembalian **between** "2023-10-25" **and** "2023-10-27";



1. Query melihat data pada tabel “pengembalian, transaksi dan buku”

**select**

p.id\_transaksi,

p.tgl\_pengembalian,

p.denda,

t.status,

b.kode\_buku,

b.judul,

b.pengarang

**from**

pengembalian p

**inner** **join** transaksi t **on**

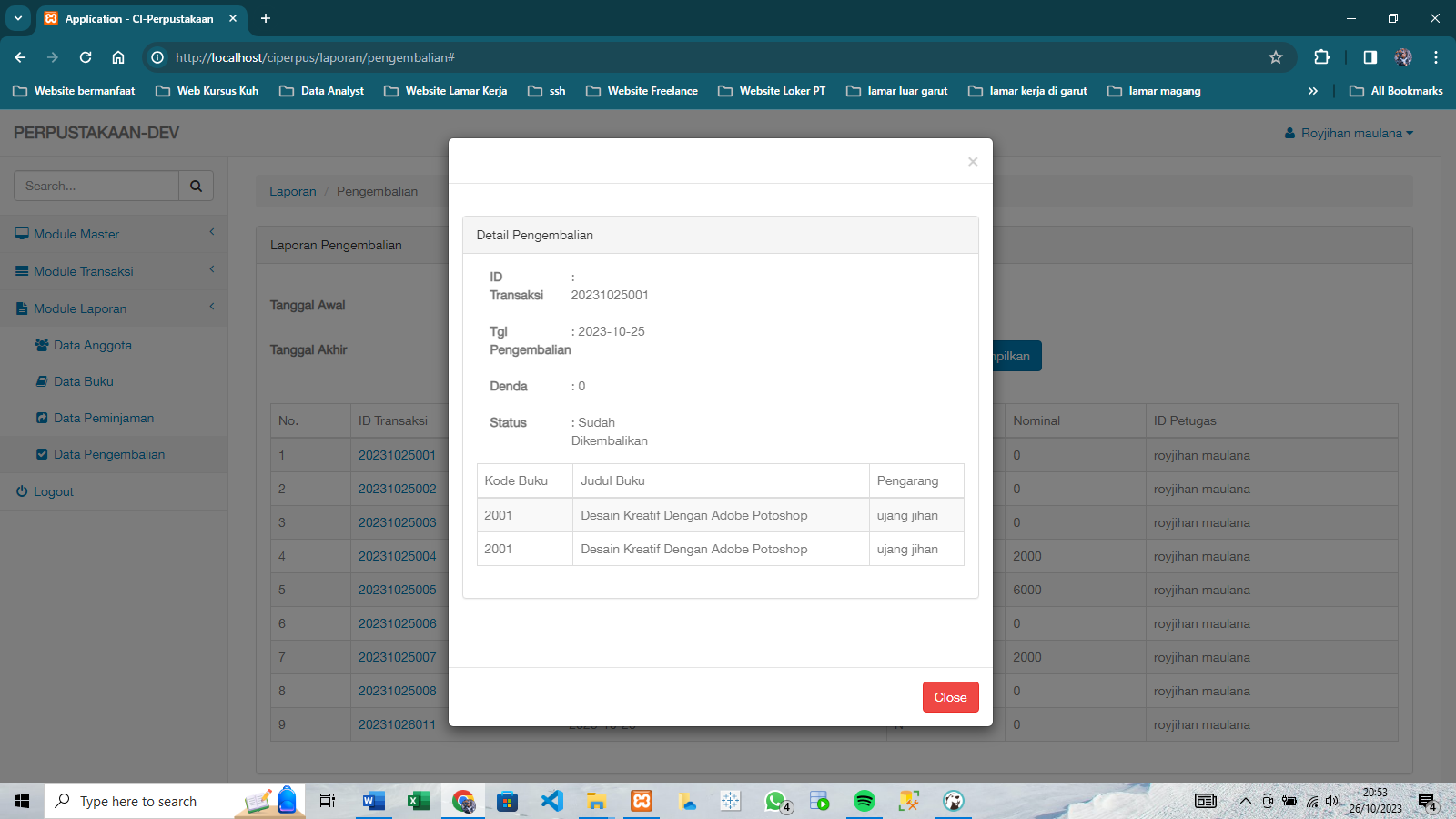
p.id\_transaksi = t.id\_transaksi

**inner** **join** buku b **on**

b.kode\_buku = t.kode\_buku

**where**

t.id\_transaksi = "20231025001";



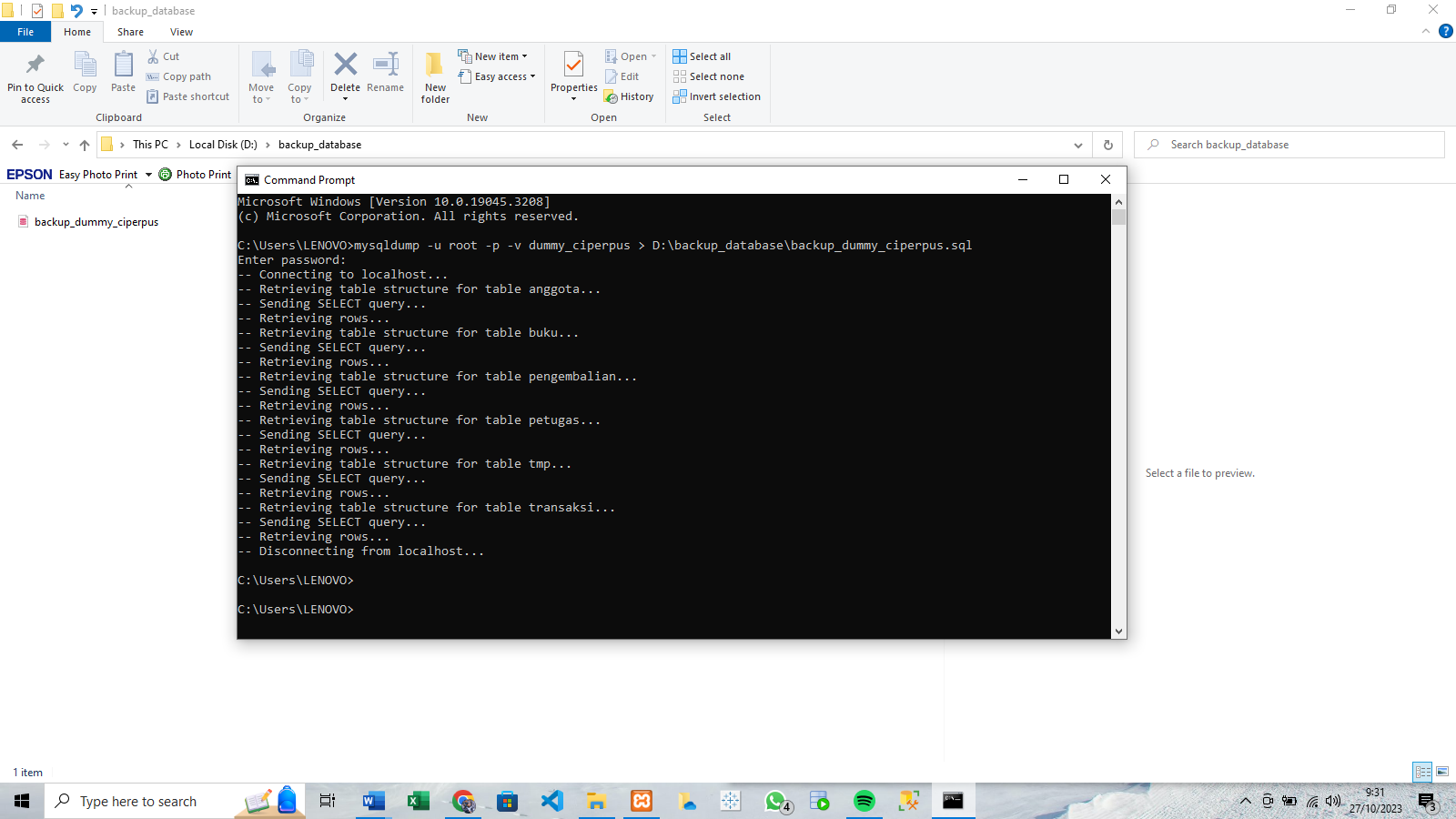
## Backup dan Restore

Backup dalam database adalah proses pembuatan salinan data dari database yang ada, yang bertujuan untuk melindungi data dari kehilangan atau kerusakan. Backup umumnya dilakukan secara berkala, dan ada berbagai jenis backup, seperti backup lengkap, backup diferensial, dan backup transaksional. Data backup disimpan di lokasi yang aman untuk pemulihan data dalam situasi darurat.

Restore dalam database adalah proses pengembalian data dari salinan cadangan (backup) ke dalam database utama setelah terjadi kerusakan atau kehilangan data. Pemulihan dapat dilakukan dalam berbagai tingkatan, tergantung pada jenis backup yang digunakan. Ini memungkinkan pengguna untuk mengembalikan data ke keadaan sebelumnya setelah insiden seperti kegagalan perangkat keras, serangan siber, atau kesalahan manusia. Backup dan restore merupakan praktek penting untuk menjaga keberlanjutan data dalam lingkungan database.

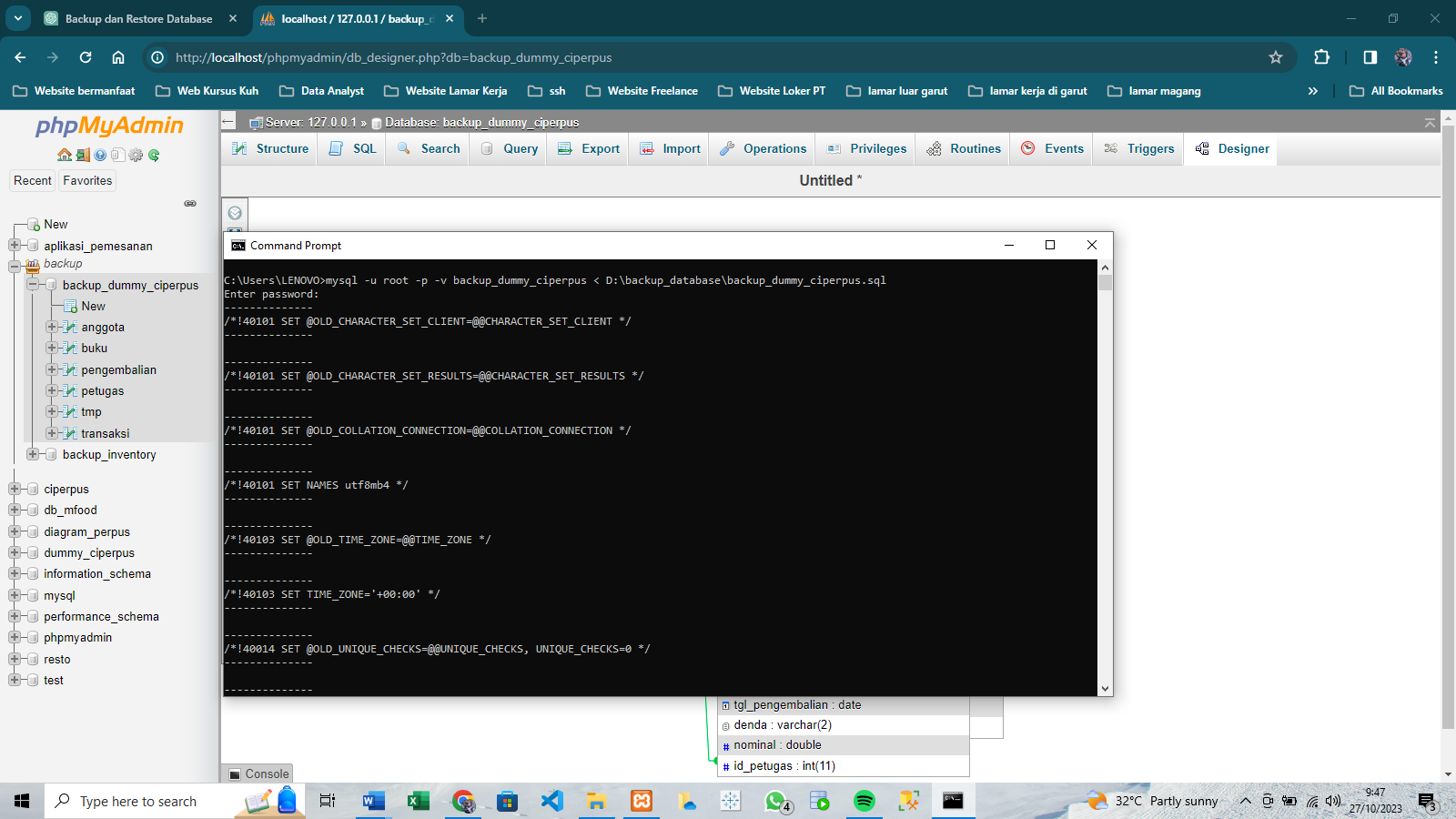
1. Backup Database

Printah cmd : mysqldump -u root -p -v dummy\_ciperpus > D:\backup\_database\backup\_dummy\_ciperpus.sql



1. Restore Database

Printah cmd : mysql -u root -p -v backup\_dummy\_ciperpus < D:\backup\_database\backup\_dummy\_ciperpus.sql



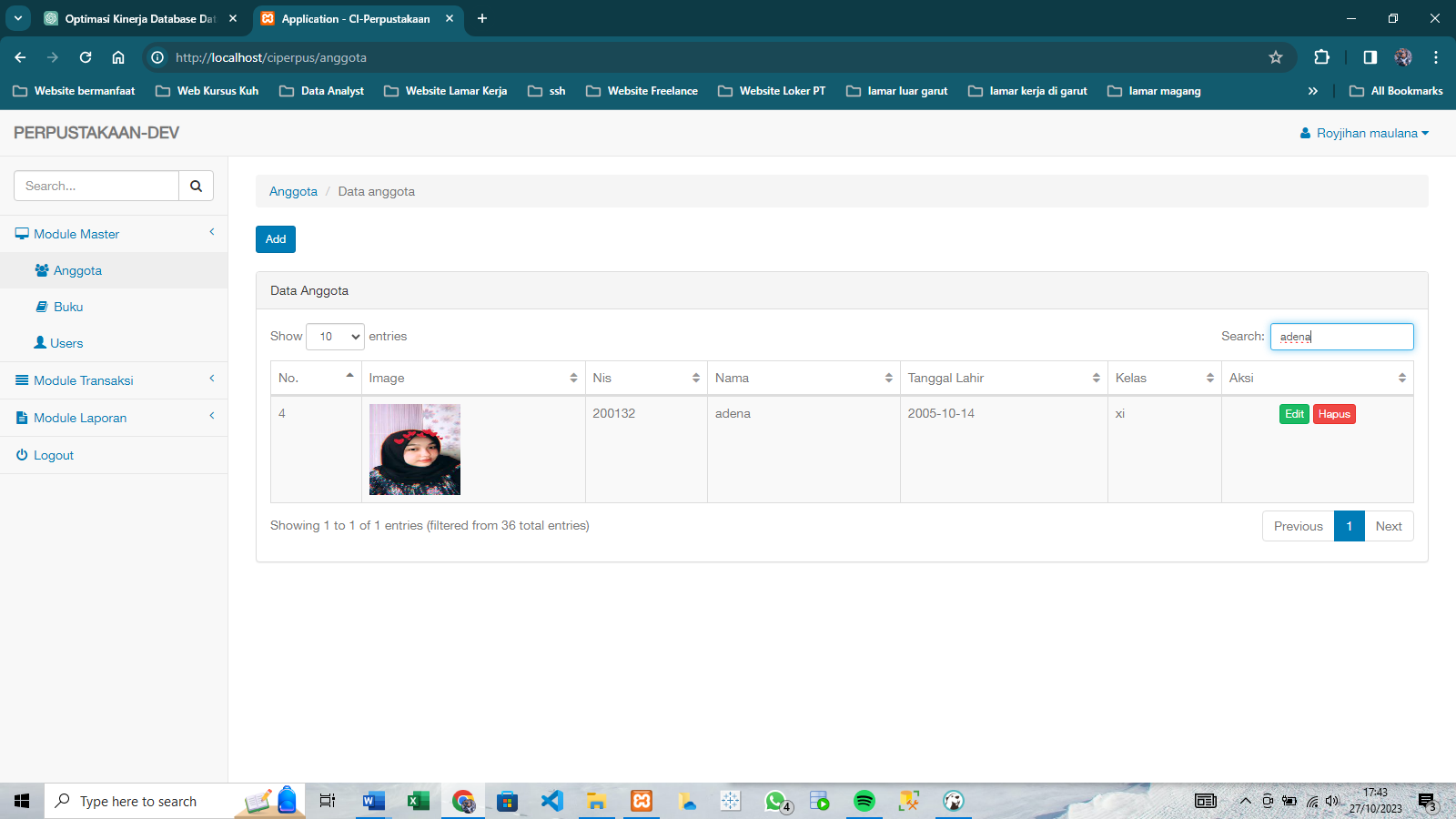
## Optimasi Kinerja

Optimasi kinerja database adalah proses meningkatkan kinerja sistem database agar lebih efisien, responsif, dan mampu mengatasi beban kerja yang lebih tinggi. Tujuan utama dari optimasi kinerja database adalah mengurangi waktu eksekusi query, menghemat sumber daya, dan menjaga konsistensi data.

1. Tabel “anggota” menambahkan index pada kolom “nama”

**create** **index** idx\_nama **on**

anggota (nama);



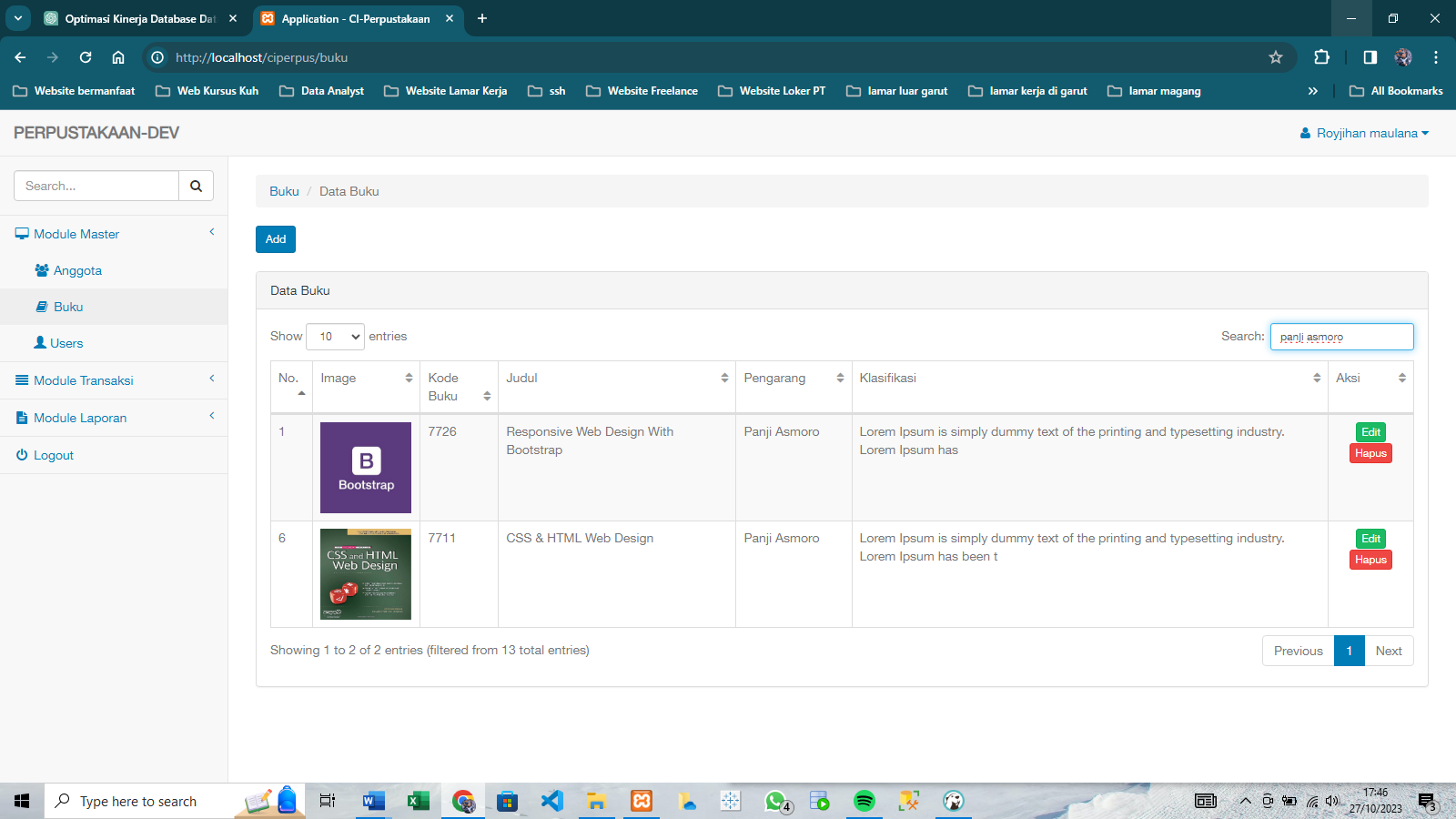
1. Tabel “buku” menambahkan index pada kolom “judul dan pengarang”

**create** **index** idx\_judul **on**

buku (judul);

**create** **index** idx\_pengarang **on**

buku (pengarang);



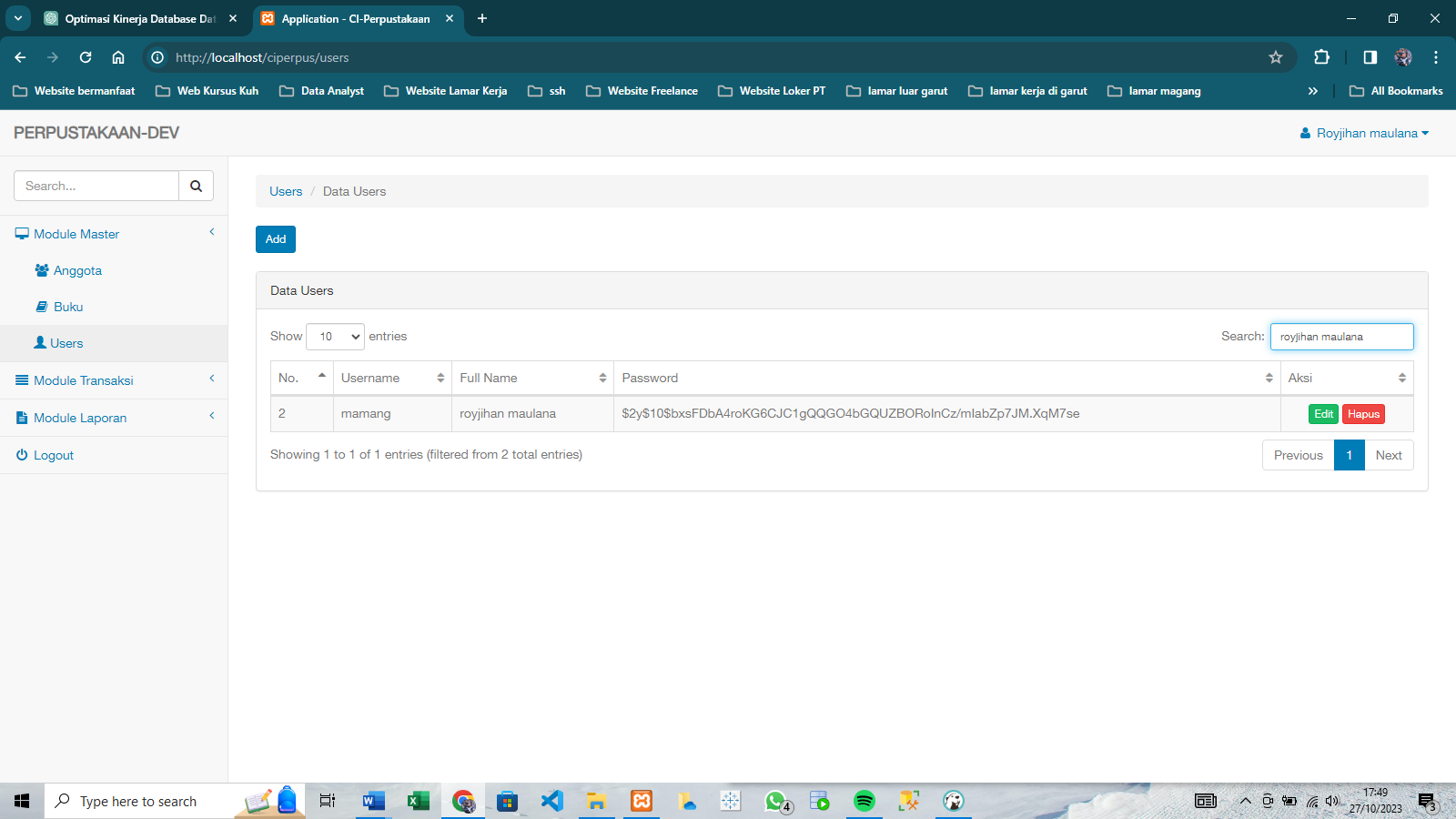
1. Tabel “petugas” menambahkan index pada kolom “username dan full\_name”

**create** **index** idx\_username **on**

petugas (username);

**create** **index** idx\_full\_name **on**

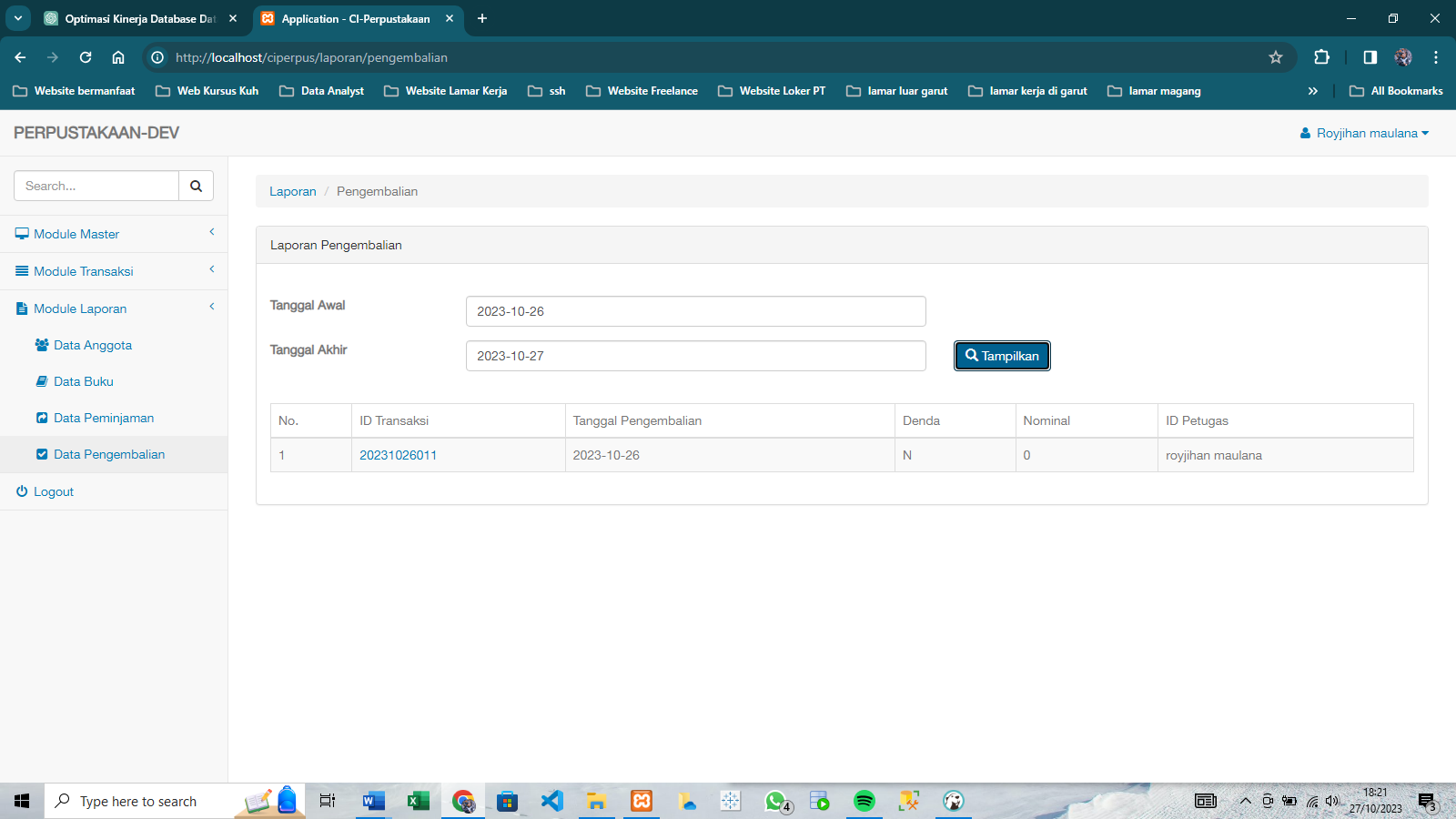
petugas (full\_name);



1. Tabel “pengembalian” menambahkan index pada kolom “tgl\_pengembalian”

**create** **index** idx\_tgl\_pengembalian **on**

pengembalian (tgl\_pengembalian);



1. Tabel “Transaksi” menambahkan index pada kolom “tanggal\_kembali dan tanggal\_pinjam”

**create** **index** idx\_tanggal\_kembali **on**

transaksi (tanggal\_kembali);

**create** **index** idx\_tanggal\_pinjam **on**

transaksi (tanggal\_pinjam);



## Skrip Otomatisasi

Skrip otomatisasi ini dibuat untuk melakukan backup database dummy\_ciperpus secara otomatis yan dijadwalkan setiap hari pada pukul 7 pagi. Skrip ini digunakan menggunakan bahasa pemrogaraman python.

Skrip :

import os

import subprocess

import schedule

import time

from datetime import datetime

def backup\_database():

    # Konfigurasi koneksi ke basis data MySQL/MariaDB

    db\_config = {

        'user': 'root',

        'password': '',

        'host': 'localhost',

        'database': 'dummy\_ciperpus'

    }

    # Nama file pencadangan dengan timestamp

    backup\_filename = f"backup\_dummy\_ciperpus\_{datetime.now().strftime('%Y-%m-%d\_%H-%M-%S')}.sql"

    # Eksekusi perintah mysqldump

    try:

        subprocess.run([

            'mysqldump',

            f"--user={db\_config['user']}",

            f"--password={db\_config['password']}",

            f"--host={db\_config['host']}",

            db\_config['database'],

            f"--result-file={backup\_filename}"

        ], check=True)

        print(f"Pencadangan berhasil dibuat: {backup\_filename}")

    except subprocess.CalledProcessError as e:

        print(f"Error: {e}")

# Menjadwalkan pencadangan setiap hari pukul 7 pagi

schedule.every().day.at("07:00").do(backup\_database)

# Loop utama untuk menjalankan penjadwalan

while True:

    schedule.run\_pending()

    time.sleep(1)

Output :

