UNIVERSITAS ESA UNGGUL
CSF101 ALGORITMA DAN PEMROGRAMAN KJ1001 7174

BUILDING CONCEPT OF PROJECT INITIATION – APLIKASI PEMINJAMAN DAN PENGEMBALIAN BUKU

TUGAS KELOMPOK 7 PROGRESS 2

Dosen Pengampu:
7174 - Ir. Sawali Wahyu, S.Kom., M.Kom

Kelompok 7:

1.	Muhamad Akbar Fadilah	-	20200801269
2.	Christian Niko Saputra	-	20240801295
3.	Denis Prastya Putra	-	20240801319
4.	Davina Tri Febriyanti	-	20240801361
5.	Arva Raihan Javier	-	20240801344

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL
TAHUN 2024

DAFTAR ISI

PROJECT OVERVIEW

1. **Deskripsi Proyek**

   Aplikasi Peminjaman dan Pengembalian Buku adalah sistem manajemen perpustakaan digital yang dirancang untuk memudahkan proses administrasi perpustakaan. Sistem ini mengotomatisasi proses peminjaman dan pengembalian buku, serta memberikan visibilitas yang lebih baik terhadap inventaris perpustakaan.

2. **Tujuan Proyek**

   Dalam era digital yang terus berkembang, perpustakaan tradisional menghadapi tantangan signifikan dalam mengelola operasional mereka secara efisien dan efektif. Dengan mengadopsi teknologi terkini dan metodologi pengembangan yang tepat, sistem ini diharapkan dapat menjadi solusi komprehensif untuk berbagai permasalahan yang dihadapi perpustakaan. Adapun tujuan spesifik dari pengembangan proyek ini adalah:

   - Meningkatkan efisiensi pengelolaan perpustakaan
   - Meminimalisir kesalahan dalam pencatatan
   - Mempermudah pelacakan status buku
   - Menghasilkan laporan yang akurat tentang aktivitas perpustakaan

3. **Fitur**

   1. Manajemen Buku
      i. Pendaftaran buku baru
      ii. Katalog buku digital
      iii. Pencarian buku
      iv. Pembaruan status buku
   2. Manajemen Anggota
      i. Pendaftaran anggota baru
      ii. Pengelolaan profil anggota
      iii. Riwayat peminjaman
      iv. Status keanggotaan
   3. Transaksi
      i. Proses peminjaman buku
      ii. Proses pengembalian buku
      iii. Perpanjangan masa pinjam
      iv. Perhitungan denda keterlambatan
   4. Pelaporan
      i. Laporan peminjaman
      ii. Statistik buku populer
      iii. Laporan keterlambatan

4. **Pengguna Sistem**

   1. Admin Perpustakaan

2. Petugas Perpustakaan
3. Supervisor/Kepala Perpustakaan

5. Teknologi yang Digunakan

1. Bahasa Pemrograman:
    i. C++
2. Struktur Data:
    i. Stack

6. Development Tools

1. IDE dan Text Editor:
    i. Dev-C++
2. Compiler:
    i. GNU G++

7. Version Control

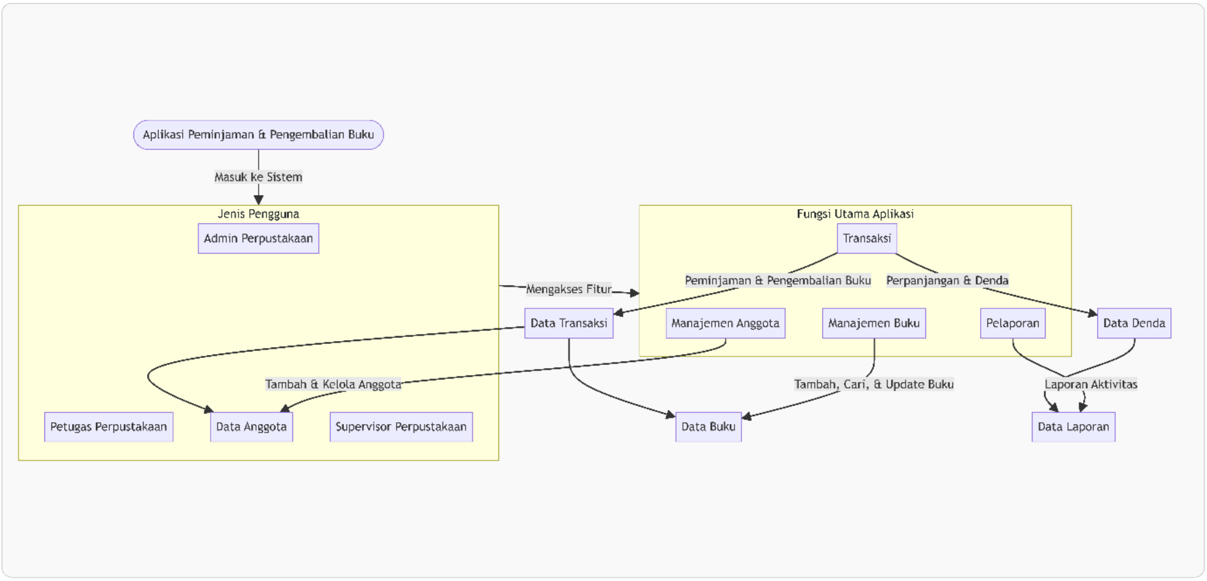
1. Git untuk manajemen versi kode
2. GitHub untuk repositori dan kolaborasi

8. Testing Tools

1. Unit Testing
    i. Pengujian fungsi-fungsi individual
    ii. Validasi operasi stack

FLOWCHART KONSEP APLIKASI

Flowchart ini menggambarkan alur kerja aplikasi Peminjaman & Pengembalian Buku yang mencakup interaksi pengguna, fungsi utama, dan hubungan antar data



Gambar 1 – Flowchart Konsep Aplikasi

PSEUDOCODE APLIKASI

```
// Global Data Structures
STRUCT Book
   id: INTEGER
   title: STRING
   author: STRING
   isAvailable: BOOLEAN
   category: STRING
END STRUCT

STRUCT Member
   id: INTEGER
   name: STRING
   membershipStatus: STRING
   borrowedBooks: STACK OF INTEGER
END STRUCT

STRUCT Transaction
   bookId: INTEGER
   memberId: INTEGER
   transactionType: STRING
   date: STRING
   dueDate: STRING
END STRUCT
STRUCT User
   username: STRING
   password: STRING
   role: STRING
END STRUCT
```

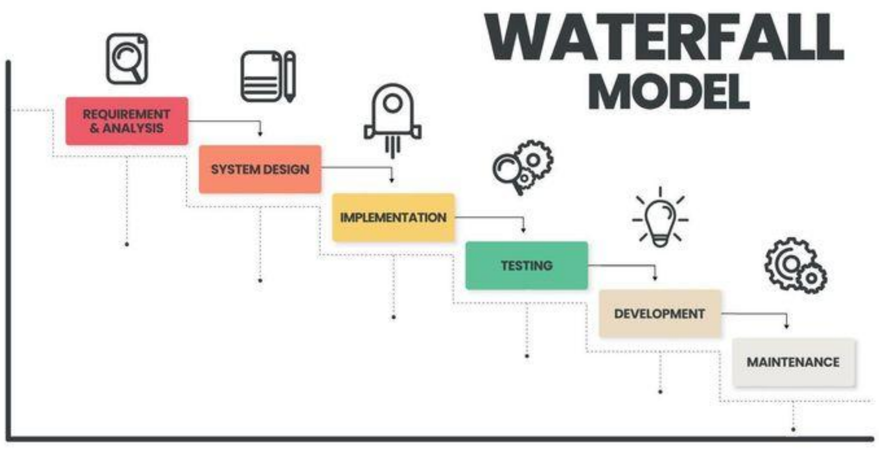
```
// Main Program
PROCEDURE Main()
  DECLARE loggedInUser: User

  // Initialize default users
  ADD admin credentials to users
  ADD petugas credentials to users
  ADD supervisor credentials to users
  IF login(loggedInUser) THEN
    DISPLAY "Login successful"
    showMainMenu(loggedInUser)
  ELSE
    DISPLAY "Login failed"
  END IF
END PROCEDURE

// Authentication
FUNCTION login(OUT loggedInUser: User) RETURNS BOOLEAN
  INPUT username
  INPUT password
  FOR EACH user IN users DO
    IF user.username = username AND user.password = password THEN
      loggedInUser ← user
      RETURN TRUE
    END IF
  END FOR
  RETURN FALSE
END FUNCTION
```

```
// Main Menu
PROCEDURE showMainMenu(IN loggedInUser: User)
  REPEAT
    IF loggedInUser.role = "Admin" OR loggedInUser.role = "Supervisor" THEN
      DISPLAY menu options for admin/supervisor
    ELSE IF loggedInUser.role = "Petugas" THEN
      DISPLAY menu options for petugas
    END IF
    INPUT choice
    CASE choice OF
      1: IF Admin/Supervisor THEN bookManagementMenu()
        IF Petugas THEN transactionManagementMenu()
      2: IF Admin/Supervisor THEN memberManagementMenu()
        IF Petugas THEN EXIT
      3: IF Admin/Supervisor THEN transactionManagementMenu()
      4: IF Admin/Supervisor THEN reportMenu()
      5: EXIT
    END CASE
  UNTIL choice = EXIT
END PROCEDURE

// Book Management
PROCEDURE bookManagementMenu()
  REPEAT
    DISPLAY book management options
    INPUT choice
    CASE choice OF
      1: registerNewBook()
```

```
            2: viewBookCatalog()
            3: searchBook()
            4: updateBookStatus()
            5: deleteBook()
            6: RETURN to main menu
        END CASE
    UNTIL choice = 6
END PROCEDURE

// Member Management
PROCEDURE memberManagementMenu()
    REPEAT
        DISPLAY member management options
        INPUT choice
        CASE choice OF
            1: registerNewMember()
            2: viewMember()
            3: searchMember()
            4: updateMemberProfile()
            5: deleteMember()
            6: RETURN to main menu
        END CASE
    UNTIL choice = 6
END PROCEDURE

// Transaction Management
PROCEDURE transactionManagementMenu()
    REPEAT
        DISPLAY transaction options
    INPUT choice
    CASE choice OF
      1: borrowBook()
      2: processReturning()
      3: extendBorrowing()
      4: deleteTransaction()
      5: RETURN to main menu
    END CASE
  UNTIL choice = 5
END PROCEDURE

// Report Management
PROCEDURE reportMenu()
  REPEAT
    DISPLAY report options
    INPUT choice
    CASE choice OF
      1: calculateLateFee()
      2: viewTransactions()
      3: RETURN to main menu
    END CASE
  UNTIL choice = 3
END PROCEDURE
END PROGRAM
```

METODOLOGI PENGEMBANGAN SISTEM

Metode Waterfall adalah model pengembangan perangkat lunak yang mengikuti pendekatan linier dan berurutan, di mana setiap fase—mulai dari perencanaan, analisis kebutuhan, desain, implementasi, pengujian, hingga pemeliharaan—diselesaikan sepenuhnya sebelum beralih ke fase berikutnya.



Gambar 2 - Waterfall Model

(sumber: vecteezy.com)

HASIL AKHIR

1. **Source Code**

```
#include <iostream>
#include <stack>
#include <vector>
#include <string>
#include <sstream>
#include <iomanip>
#include <stdexcept>
#include <ctime>

using namespace std;

struct Book {
  int id;
  string title;
  string author;
  bool isAvailable;
  string category;
};

struct Member {
  int id;
  string name;
  string membershipStatus;
  stack<int> borrowedBooks;
};

struct Transaction {
  int bookId;
  int memberId;
```

```
    string transactionType; // Type: Borrow, Return, Extend
    string date;
    string dueDate;
};

struct User {
    string username;
    string password;
    string role; // Role: Admin, Petugas, Supervisor
};

// Global Data
vector<Book> books;
vector<Member> members;
vector<Transaction> transactions;
vector<User> users;

// Utility functions
time_t parseDate(const string& dateStr) {
    struct tm tm = {};
    if (sscanf(dateStr.c_str(), "%d-%d-%d", &tm.tm_year, &tm.tm_mon, &tm.tm_mday) !=
3) {
        throw std::runtime_error("Invalid date format: " + dateStr);
    }
    tm.tm_year -= 1900; // Adjust year (tm_year is years since 1900)
    tm.tm_mon -= 1;     // Adjust month (tm_mon is 0-based)
    return mktime(&tm);
}

string formatDate(time_t date) {
    char buffer[11];
    strftime(buffer, 11, "%Y-%m-%d", localtime(&date));
    return string(buffer);
}

bool login(User& loggedInUser) {
    string username, password;
    cout << "Enter username: ";
    cin >> username;
    cout << "Enter password: ";
    cin >> password;

    for (const auto& user : users) {
        if (user.username == username && user.password == password) {
            loggedInUser = user;
            return true;
        }
    }
    return false;
}

// Function declarations
void registerNewBook();
void viewBookCatalog();
void searchBook();
void updateBookStatus();
void deleteBook();

void registerNewMember();
void viewMember();
void searchMember();
void updateMemberProfile();
void deleteMember();
void manageMemberProfile();
void viewBorrowHistory();

void processBorrowing();
void processReturning();
void extendBorrowing();
void calculateLateFee();

void borrowBook();
void viewTransactions();
void deleteTransaction();

void bookManagementMenu() {
  int choice;
  do {
    cout << "\nBook Management\n";
    cout << "1. Register New Book\n";
    cout << "2. View Book Catalog\n";
    cout << "3. Search Book\n";
    cout << "4. Update Book Information\n";
    cout << "5. Delete Book\n";
    cout << "6. Back to Main Menu\n";
    cout << "Enter your choice: ";
    cin >> choice;

    switch (choice) {
    case 1:
      registerNewBook();
      break;
    case 2:
            viewBookCatalog();
            break;
        case 3:
            searchBook();
            break;
        case 4:
            updateBookStatus();
            break;
        case 5:
            deleteBook();
            break;
        case 6:
            return;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (true);
}

void registerNewBook() {
    Book newBook;
    cout << "Enter book ID: ";
    cin >> newBook.id;
    cin.ignore();
    cout << "Enter book title: ";
    getline(cin, newBook.title);
    cout << "Enter author: ";
    getline(cin, newBook.author);
    cout << "Enter category: ";
    getline(cin, newBook.category);
    newBook.isAvailable = true;
    books.push_back(newBook);
    cout << "Book registered successfully.\n";
}

void viewBookCatalog() {
    cout << "\nBook Catalog:\n";
    cout << "+------------+-----------------------------------------------+---------------------------------
--------------+---------------------+----------------------+\n";
    cout << "| Book ID    | Title                                         | Author                                        | Category
| Availability         |\n";
    cout << "+------------+-----------------------------------------------+---------------------------------
--------------+---------------------+----------------------+\n";

    bool found = false;

    for (size_t i = 0; i < books.size(); ++i) {
        cout << "| " << setw(10) << left << books[i].id
             << " | " << setw(45) << left << books[i].title
             << " | " << setw(45) << left << books[i].author
             << " | " << setw(19) << left << books[i].category
             << " | " << setw(20) << left << (books[i].isAvailable ? "Yes" : "No") << " |\n";

        found = true;
    }

    if (!found) {
        cout << "| Data books not found.\n";
    }

    cout << "+------------+-----------------------------------------------+---------------------------------
--------------+---------------------+----------------------+\n";
}
```

```
void searchBook() {
  string searchQuery;
  cout << "Enter book title or author to search: ";
  cin.ignore();
  getline(cin, searchQuery);

  bool found = false;

  cout << "\nSearch Results:\n";
  cout << "+------------+-------------------------------------------+------------------------------
-------------+--------------------+---------------------+\n";
  cout << "| Book ID    | Title                                     | Author                                    | Category
| Availability       |\n";
  cout << "+------------+-------------------------------------------+------------------------------
-------------+--------------------+---------------------+\n";

  for (size_t i = 0; i < books.size(); ++i) {
    if (books[i].title.find(searchQuery) != string::npos ||
books[i].author.find(searchQuery) != string::npos) {
      found = true;
      cout << "| " << setw(10) << left << books[i].id
         << " | " << setw(45) << left << books[i].title
         << " | " << setw(45) << left << books[i].author
         << " | " << setw(19) << left << books[i].category
         << " | " << setw(20) << left << (books[i].isAvailable ? "Yes" : "No") << " |\n";
    }
  }

  if (!found) {
    cout << "| No books found matching your search criteria.\n";
  }
```

```
    cout << "+------------+----------------------------------------------+---------------------------------
--------------+---------------------+----------------------+\n";
}

void updateBookStatus() {
    int bookId;
    cout << "Enter book ID to update: ";
    cin >> bookId;
    cin.ignore();

    for (size_t i = 0; i < books.size(); ++i) {
        if (books[i].id == bookId) {
            cout << "Enter new title (leave blank to keep current): ";
            string newTitle;
            getline(cin, newTitle);
            if (!newTitle.empty()) books[i].title = newTitle;

            cout << "Enter new author (leave blank to keep current): ";
            string newAuthor;
            getline(cin, newAuthor);
            if (!newAuthor.empty()) books[i].author = newAuthor;

            cout << "Enter new category (leave blank to keep current): ";
            string newCategory;
            getline(cin, newCategory);
            if (!newCategory.empty()) books[i].category = newCategory;

            cout << "Is the book available? (1 for Yes, 0 for No): ";
            cin >> books[i].isAvailable;
            cout << "Book updated successfully.\n";
            return;
        }
    }
    cout << "Book with ID " << bookId << " not found.\n";
}

void deleteBook() {
    int bookId;
    cout << "Enter book ID to delete: ";
    cin >> bookId;

    for (size_t i = 0; i < books.size(); ++i) {
        if (books[i].id == bookId) {
            books.erase(books.begin() + i);
            cout << "Book deleted successfully.\n";
            return;
        }
    }
    cout << "Book with ID " << bookId << " not found.\n";
}

void memberManagementMenu() {
    int choice;
    do {
        cout << "\nMember Management\n";
        cout << "1. Register New Member\n";
        cout << "2. View Member\n";
        cout << "3. Search Member\n";
        cout << "4. Update Member Profile\n";
        cout << "5. Delete Member\n";
        cout << "6. Back to Main Menu\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            registerNewMember();
            break;
        case 2:
            viewMember();
            break;
        case 3:
            searchMember();
            break;
        case 4:
            updateMemberProfile();
            break;
        case 5:
            deleteMember();
            break;
        case 6:
            return;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (true);
}

void registerNewMember() {
    Member newMember;
    cout << "Enter member ID: ";
    cin >> newMember.id;
    cin.ignore();
    cout << "Enter member name: ";
    getline(cin, newMember.name);
    newMember.membershipStatus = "Active";
    members.push_back(newMember);
    cout << "Member registered successfully.\n";
}

void viewMember() {
    cout << "\nMember Catalog:\n";
    cout << "+------------+---------------------------------------------+---------------------+\n";
    cout << "| Member ID  | Name                                        | Status              |\n";
    cout << "+------------+---------------------------------------------+---------------------+\n";

    bool found = false;

    for (size_t i = 0; i < members.size(); ++i) {
        cout << "| " << setw(10) << left << members[i].id
             << " | " << setw(45) << left << members[i].name
             << " | " << setw(19) << left << members[i].membershipStatus << " |\n";

        found = true;
    }


    if (!found) {
        cout << "| Data members not found.\n";
    }

    cout << "+------------+---------------------------------------------+---------------------+\n";
```

```
}

void searchMember() {
    string searchQuery;
    cout << "Enter member name to search: ";
    cin.ignore();
    getline(cin, searchQuery);

    bool found = false;

    cout << "\nSearch Results:\n";
    cout << "+------------+-----------------------------------------------+---------------------+\n";
    cout << "| Member ID  | Name                                          | Status              |\n";
    cout << "+------------+-----------------------------------------------+---------------------+\n";

    for (size_t i = 0; i < members.size(); ++i) {
        if (members[i].name.find(searchQuery) != string::npos) {
            found = true;
            cout << "| " << setw(10) << left << members[i].id
                 << " | " << setw(45) << left << members[i].name
                 << " | " << setw(19) << left << members[i].membershipStatus << " |\n";
        }
    }

    if (!found) {
        cout << "| No members found matching your search criteria.\n";
    }

    cout << "+------------+-----------------------------------------------+---------------------+\n";
}
```

```
void updateMemberProfile() {
  int memberId;
  cout << "Enter member ID to update: ";
  cin >> memberId;
  cin.ignore();

  for (size_t i = 0; i < members.size(); ++i) {
    if (members[i].id == memberId) {
      cout << "Enter new name (leave blank to keep current): ";
      string newName;
      getline(cin, newName);
      if (!newName.empty()) members[i].name = newName;

      cout << "Enter new membership status (leave blank to keep current): ";
      string newStatus;
      getline(cin, newStatus);
      if (!newStatus.empty()) members[i].membershipStatus = newStatus;

      cout << "Member updated successfully.\n";
      return;
    }
  }
  cout << "Member with ID " << memberId << " not found.\n";
}

void deleteMember() {
  int memberId;
  cout << "Enter member ID to delete: ";
  cin >> memberId;

  for (size_t i = 0; i < members.size(); ++i) {
        if (members[i].id == memberId) {
            members.erase(members.begin() + i);
            cout << "Member deleted successfully.\n";
            return;
        }
    }
    cout << "Member with ID " << memberId << " not found.\n";
}

void transactionManagementMenu() {
    int choice;
    do {
        cout << "\nTransaction Management\n";
        cout << "1. Borrow Book\n";
        cout << "2. Return Book\n";
        cout << "3. Extend Borrowing\n";
        cout << "4. Delete Transaction\n";
        cout << "5. Back to Main Menu\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            borrowBook();
            break;
        case 2:
            processReturning();
            break;
        case 3:
            extendBorrowing();
            break;
        case 4:
            deleteTransaction();
            break;
        case 5:
            return;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (true);
}

void borrowBook() {
    int bookId, memberId;
    cout << "Enter Book ID to borrow: ";
    cin >> bookId;
    cout << "Enter Member ID: ";
    cin >> memberId;

    for (size_t i = 0; i < books.size(); ++i) {
        if (books[i].id == bookId) {
            if (!books[i].isAvailable) {
                cout << "Book is currently not available.\n";
                return;
            }

            for (size_t j = 0; j < members.size(); ++j) {
                if (members[j].id == memberId) {
                    Transaction newTransaction;
                    newTransaction.bookId = bookId;
                    newTransaction.memberId = memberId;
                    newTransaction.transactionType = "Borrow";
                    time_t now = time(nullptr);
                    newTransaction.date = formatDate(now);

                    // Set tanggal jatuh tempo (7 hari dari sekarang)
                    time_t dueDate = now + 7 * 24 * 60 * 60;
                    newTransaction.dueDate = formatDate(dueDate);

                    // Simpan transaksi
                    transactions.push_back(newTransaction);

                    // Tandai buku sebagai tidak tersedia
                    books[i].isAvailable = false;

                    // Tambahkan buku ke daftar peminjaman anggota
                    members[j].borrowedBooks.push(bookId);

                    cout << "Book borrowed successfully. Due date: " << newTransaction.dueDate
<< "\n";
                    return;
                }
            }
            cout << "Member not found.\n";
            return;
        }
    }
    cout << "Book not found.\n";
}

void processReturning() {
    int bookId, memberId;
    cout << "Enter Book ID to return: ";
```

```
    cin >> bookId;
    cout << "Enter Member ID: ";
    cin >> memberId;

    for (auto& transaction : transactions) {
        if (transaction.bookId == bookId && transaction.memberId == memberId &&
transaction.transactionType == "Borrow") {
            transaction.transactionType = "Return";
            transaction.date = formatDate(time(nullptr));

            for (auto& book : books) {
                if (book.id == bookId) {
                    book.isAvailable = true;
                    break;
                }
            }

            for (auto& member : members) {
                if (member.id == memberId) {
                    if (!member.borrowedBooks.empty() && member.borrowedBooks.top() ==
bookId) {
                        member.borrowedBooks.pop();
                    }
                    break;
                }
            }

            cout << "Book returned successfully.\n";
            return;
        }
    }
    cout << "No active borrowing transaction found for the given Book ID and Member ID.\n";
}

void deleteTransaction() {
    int bookId, memberId;
    cout << "Enter Book ID: ";
    cin >> bookId;
    cout << "Enter Member ID: ";
    cin >> memberId;

    for (size_t i = 0; i < transactions.size(); ++i) {
        if (transactions[i].bookId == bookId && transactions[i].memberId == memberId) {
            transactions.erase(transactions.begin() + i);
            cout << "Transaction deleted successfully.\n";
            return;
        }
    }
    cout << "Transaction not found.\n";
}

void extendBorrowing() {
    int bookId, memberId;
    cout << "Enter Book ID: ";
    cin >> bookId;
    cout << "Enter Member ID: ";
    cin >> memberId;

    for (auto& t : transactions) {
        if (t.bookId == bookId && t.memberId == memberId && t.transactionType == "Borrow") {
            time_t dueDate = parseDate(t.dueDate);
            dueDate += 7 * 24 * 60 * 60; // Extend by 7 days
            t.dueDate = formatDate(dueDate);
            t.transactionType = "Extend";
            cout << "Borrowing period extended successfully. New due date: " << t.dueDate <<
"\n";
            return;
        }
    }
    cout << "Transaction not found or not eligible for extension.\n";
}

// Reports Menu
void reportMenu() {
    int choice;
    do {
        cout << "\nReports\n";
        cout << "1. Calculate Late Fee\n";
        cout << "2. View Transactions\n";
        cout << "3. Back to Main Menu\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            calculateLateFee();
            break;
        case 2:
                viewTransactions();
            break;
        case 3:
            return;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (true);
}

void viewTransactions() {
    cout << "\nTransaction History:\n";
    cout << "+------------+------------+----------------+------------+------------+\n";
    cout << "| Book ID    | Member ID  | Transaction    | Date       | Due Date   |\n";
    cout << "+------------+------------+----------------+------------+------------+\n";

    for (const auto& t : transactions) {
        cout << "| " << setw(10) << left << t.bookId
             << " | " << setw(10) << left << t.memberId
             << " | " << setw(14) << left << t.transactionType
             << " | " << setw(10) << left << t.date
             << " | " << setw(10) << left << t.dueDate << " |\n";
    }

    cout << "+------------+------------+----------------+------------+------------+\n";
}

void calculateLateFee() {
    int bookId, memberId;
    cout << "Enter Book ID: ";
    cin >> bookId;
    cout << "Enter Member ID: ";
    cin >> memberId;

    for (const auto& t : transactions) {
        if (t.bookId == bookId && t.memberId == memberId && t.transactionType ==
"Borrow") {
            time_t dueDate = parseDate(t.dueDate);
            time_t currentDate = time(nullptr);

            if (currentDate > dueDate) {
                int daysLate = (currentDate - dueDate) / (24 * 60 * 60);
                int lateFee = daysLate * 5000; // Late fee: 5000 per day
                cout << "Book is late by " << daysLate << " days. Late fee: Rp " << lateFee << "\n";
            } else {
                cout << "No late fee. Book is returned on time.\n";
            }
            return;
        }
    }
    cout << "Transaction not found or not eligible for late fee calculation.\n";
}

void showMainMenu(User& loggedInUser) {
    int choice;
    do {
        cout << "\nLibrary Management System\n";

        if (loggedInUser.role == "Admin") {
            cout << "1. Book Management\n";
            cout << "2. Member Management\n";
            cout << "3. Transactions\n";
            cout << "4. Reports\n";
            cout << "5. Exit\n";
        }
        else if (loggedInUser.role == "Petugas") {
```

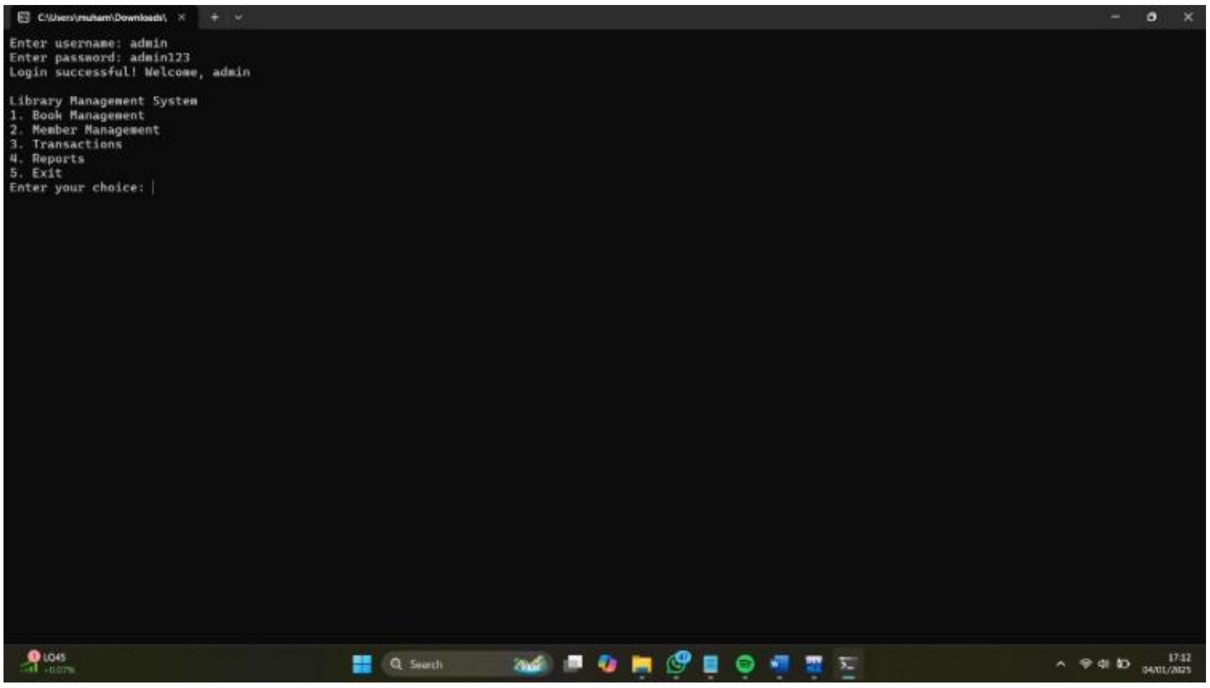
```
    cout << "1. Transactions\n";
    cout << "2. Exit\n";
  }
  else if (loggedInUser.role == "Supervisor") {
    cout << "1. Book Management\n";
    cout << "2. Member Management\n";
    cout << "3. Transactions\n";
    cout << "4. Reports\n";
    cout << "5. Exit\n";
  }

  cout << "Enter your choice: ";
  cin >> choice;

  switch (choice) {
  case 1:
    if (loggedInUser.role == "Admin" || loggedInUser.role == "Supervisor") {
      bookManagementMenu();
    }
    else if (loggedInUser.role == "Petugas") {
      transactionManagementMenu();
    }
    break;
  case 2:
    if (loggedInUser.role == "Admin" || loggedInUser.role == "Supervisor") {
      memberManagementMenu();
    }
    else if (loggedInUser.role == "Petugas") {
      transactionManagementMenu();
    }
    break;
```

```
        case 3:
          if (loggedInUser.role == "Admin" || loggedInUser.role == "Supervisor") {
            transactionManagementMenu();
          }
          break;
        case 4:
          if (loggedInUser.role == "Admin" || loggedInUser.role == "Supervisor") {
            reportMenu();
          }
          break;
        case 5:
          if (loggedInUser.role == "Admin" || loggedInUser.role == "Supervisor") {
            cout << "Exiting the system.\n";
            return;
          }
          else if (loggedInUser.role == "Petugas") {
            cout << "Exiting the system.\n";
            return;
          }
          break;
        default:
          cout << "Invalid choice. Please try again.\n";
      }
    } while (true);
}

// Main function
int main() {
    users.push_back({"admin", "admin123", "Admin"});
    users.push_back({"petugas", "petugas123", "Petugas"});
    users.push_back({"supervisor", "supervisor123", "Supervisor"});
    User loggedInUser;

    if (login(loggedInUser)) {
      cout << "Login successful! Welcome, " << loggedInUser.username << "\n";
      showMainMenu(loggedInUser);
    }
    else {
      cout << "Login failed! Please check your credentials.\n";
    }

    return 0;
  }
```

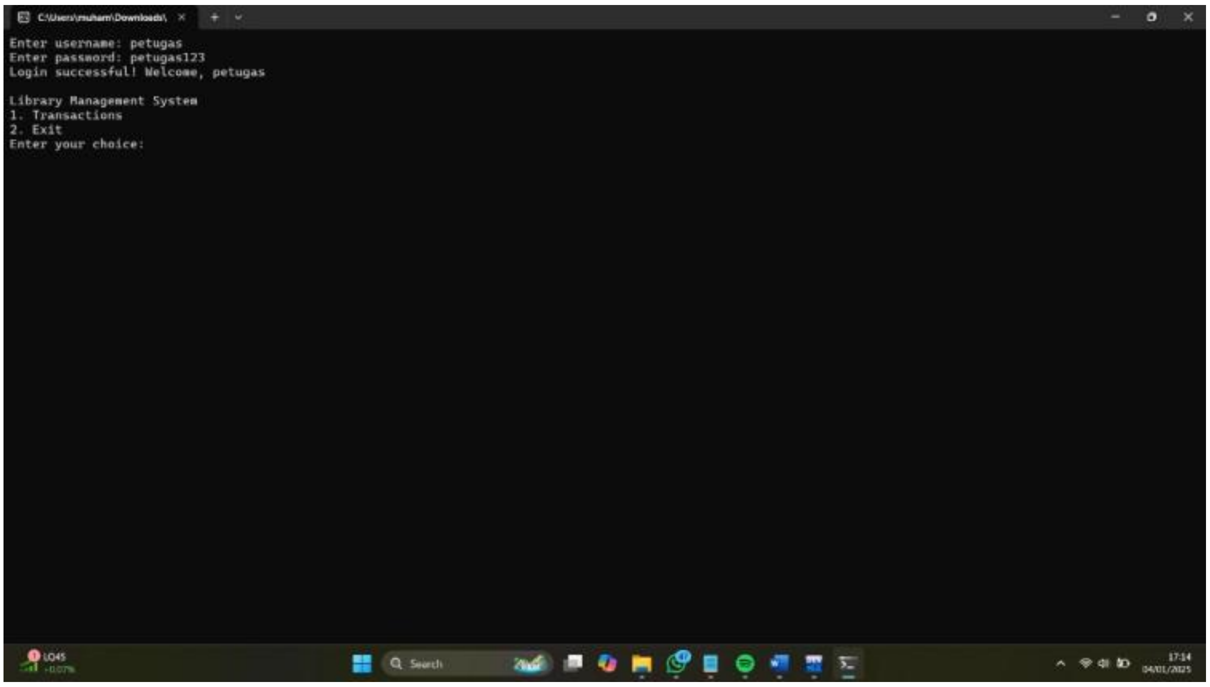
2. **Link Github**

   https://github.com/akbarfdlh2/Tugas-Kelompok-7

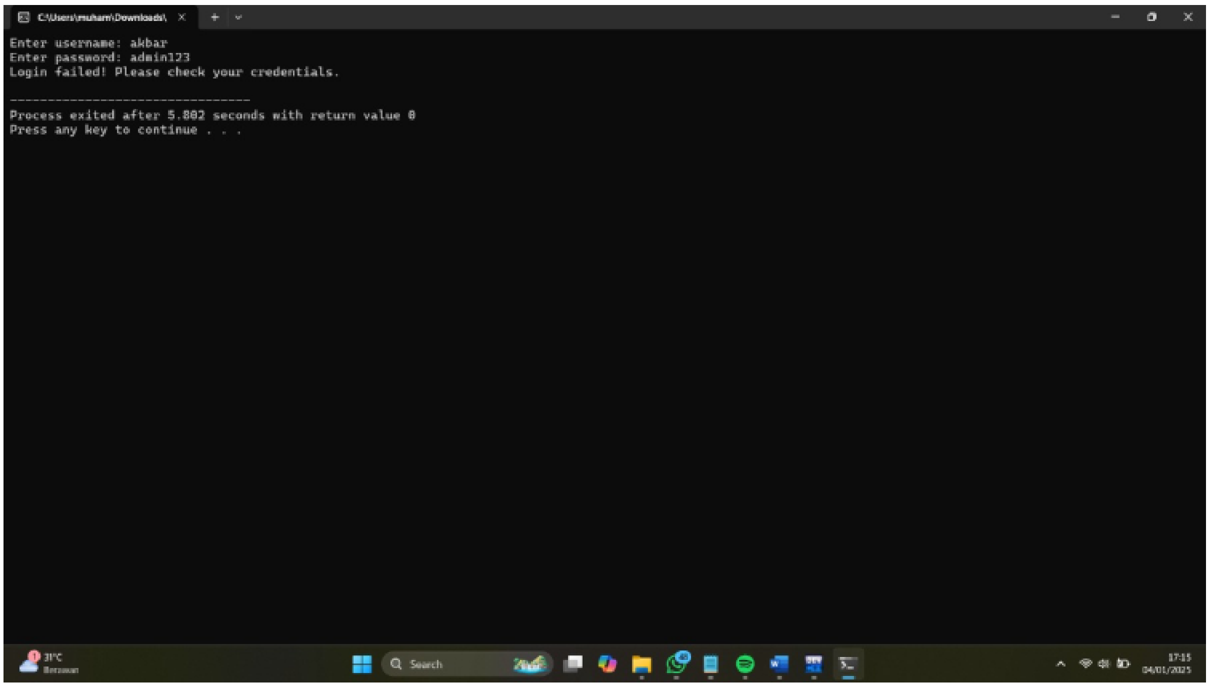
3. **Screenshot Aplikasi**



```
Enter username: admin
Enter password: admin123
Login successful! Welcome, admin

Library Management System
1. Book Management
2. Member Management
3. Transactions
4. Reports
5. Exit
Enter your choice: 
```

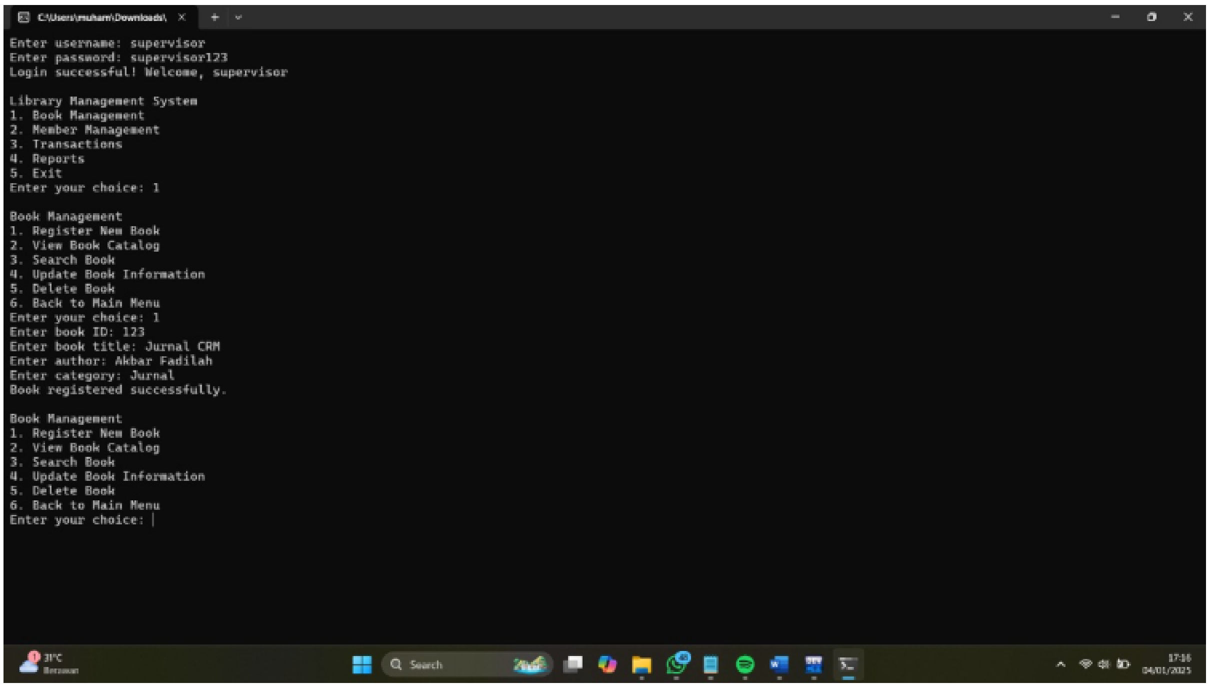


```
Enter username: petugas
Enter password: petugas123
Login successful! Welcome, petugas

Library Management System
1. Transactions
2. Exit
Enter your choice:
```



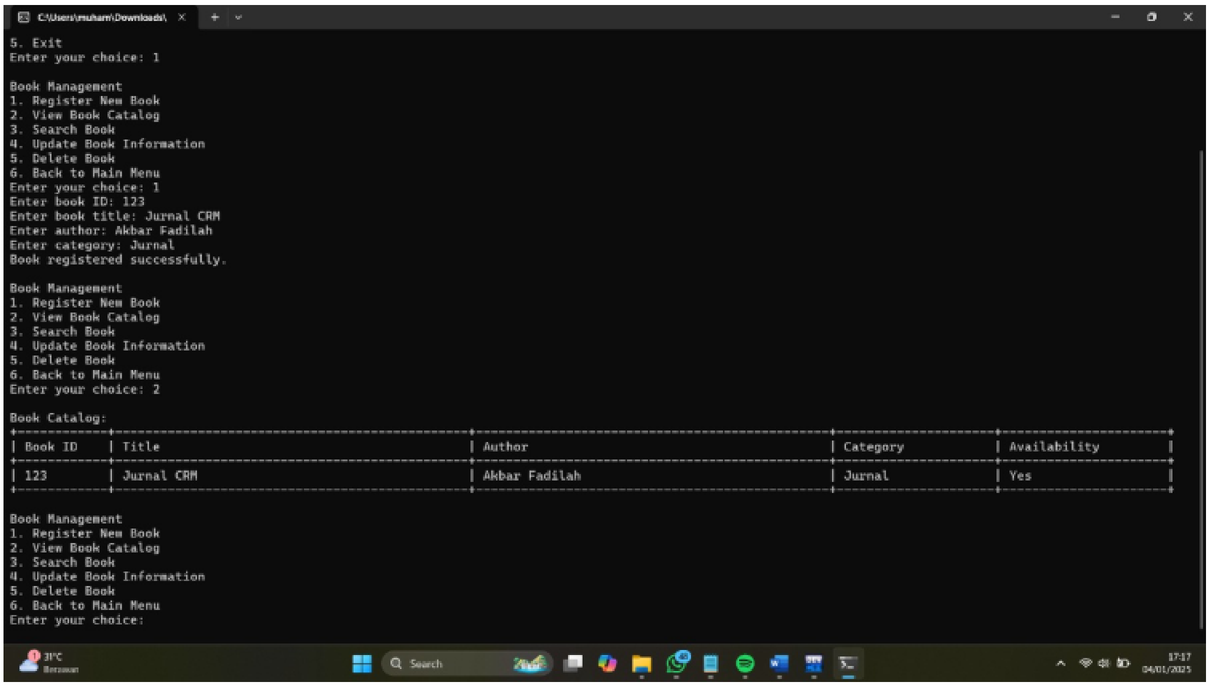
```
Enter username: akbar
Enter password: admin123
Login failed! Please check your credentials.

--------------------------------
Process exited after 5.802 seconds with return value 0
Press any key to continue . . .
```

Gambar 3 - Login



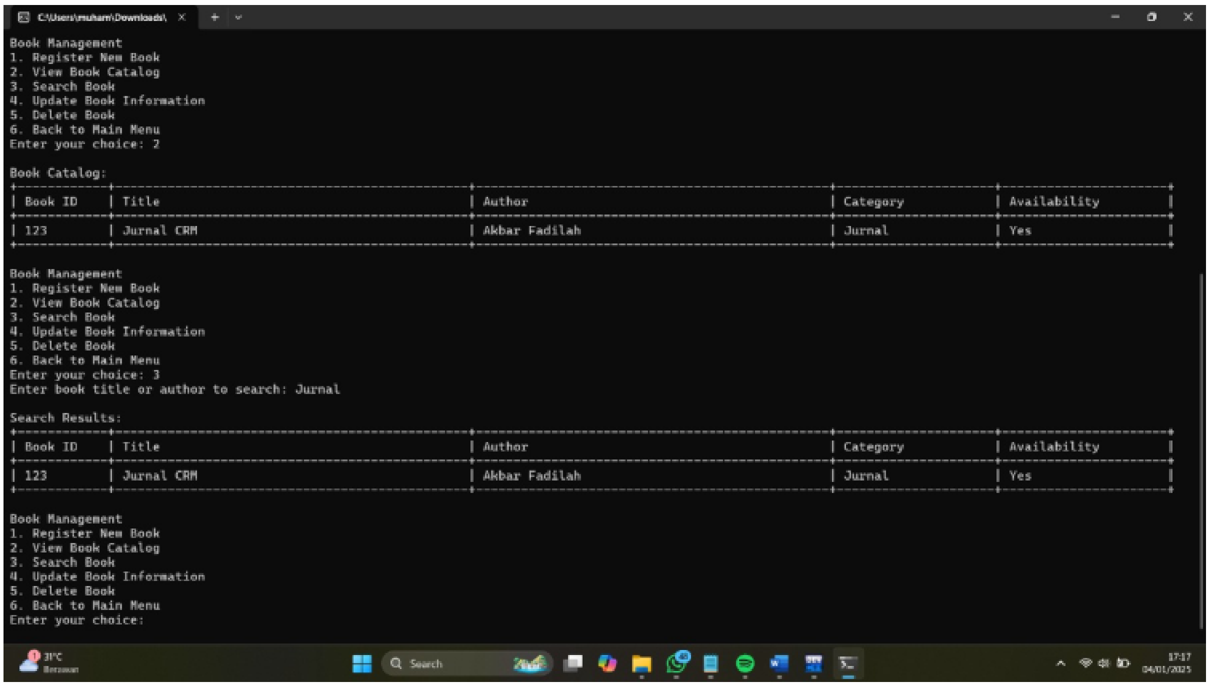
```
Enter username: supervisor
Enter password: supervisor123
Login successful! Welcome, supervisor

Library Management System
1. Book Management
2. Member Management
3. Transactions
4. Reports
5. Exit
Enter your choice: 1

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 1
Enter book ID: 123
Enter book title: Jurnal CRM
Enter author: Akbar Fadilah
Enter category: Jurnal
Book registered successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: |
```

Gambar 4 - Register New Book



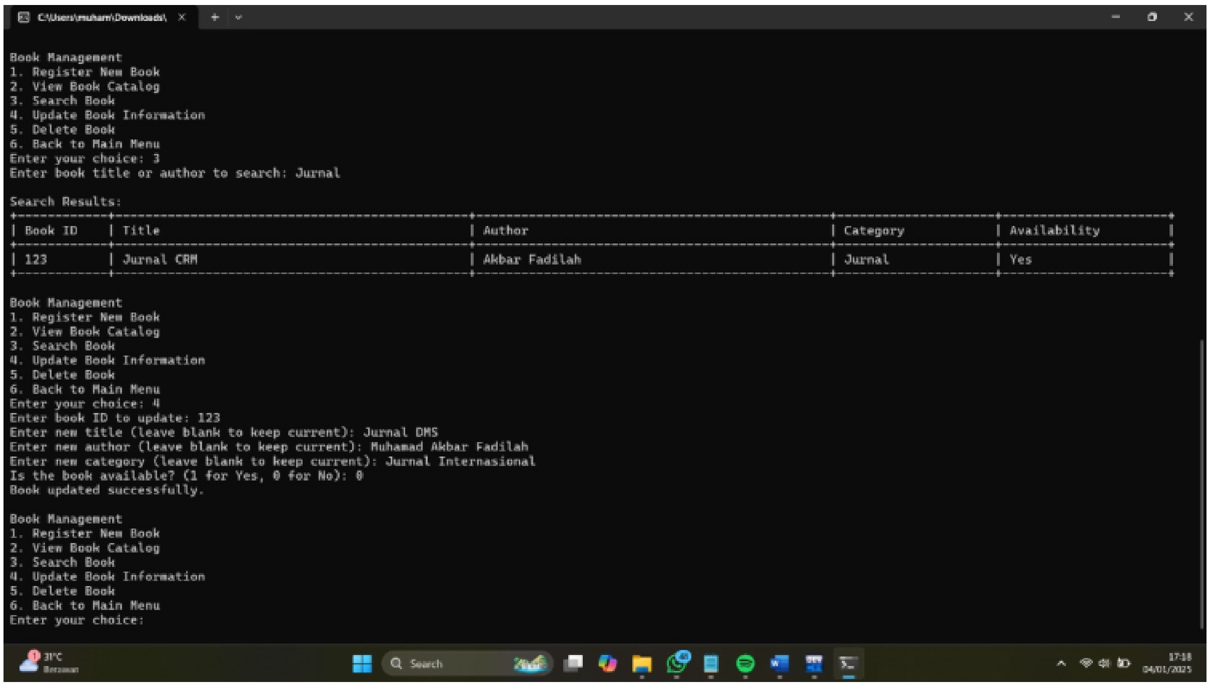
```
5. Exit
Enter your choice: 1

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 1
Enter book ID: 123
Enter book title: Jurnal CRM
Enter author: Akbar Fadilah
Enter category: Jurnal
Book registered successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 2

Book Catalog:
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID    | Title                                          | Author                                         | Category             | Availability          |
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123        | Jurnal CRM                                     | Akbar Fadilah                                  | Jurnal               | Yes                   |
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice:
```

Gambar 4 - View Book Catalog



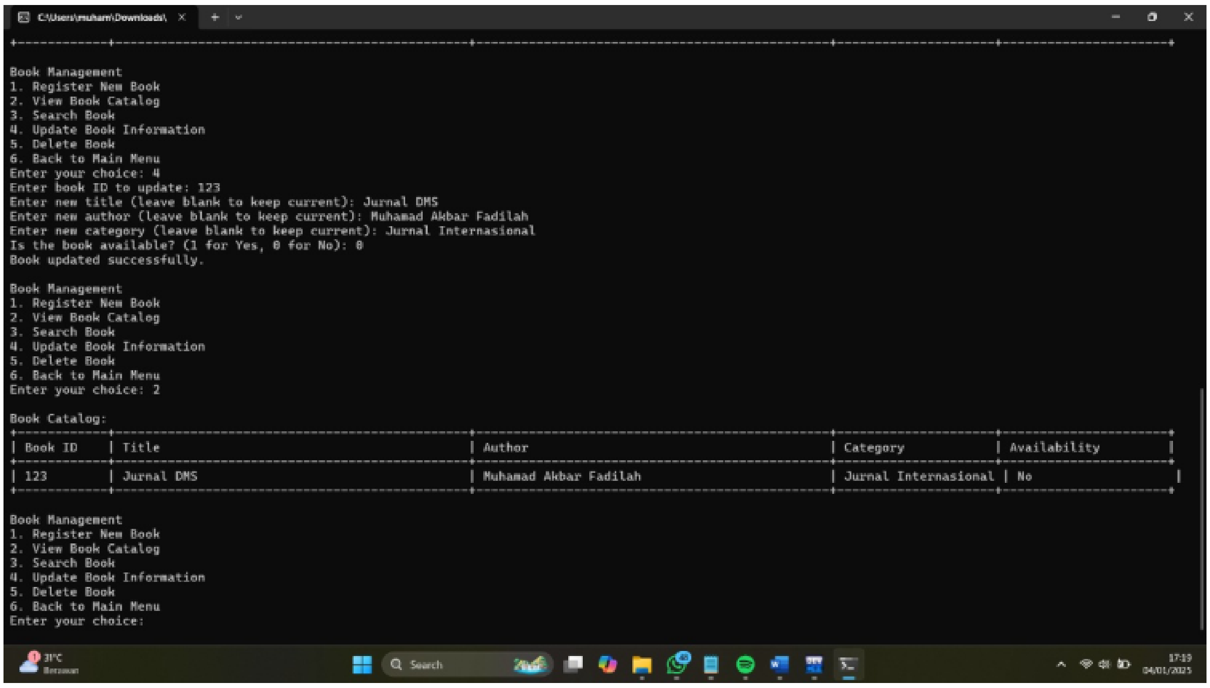
```
Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 2

Book Catalog:
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID    | Title                                          | Author                                         | Category             | Availability          |
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123        | Jurnal CRM                                     | Akbar Fadilah                                  | Jurnal               | Yes                   |
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 3
Enter book title or author to search: Jurnal

Search Results:
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID    | Title                                          | Author                                         | Category             | Availability          |
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123        | Jurnal CRM                                     | Akbar Fadilah                                  | Jurnal               | Yes                   |
+------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice:
```

Gambar 5 - Search Book



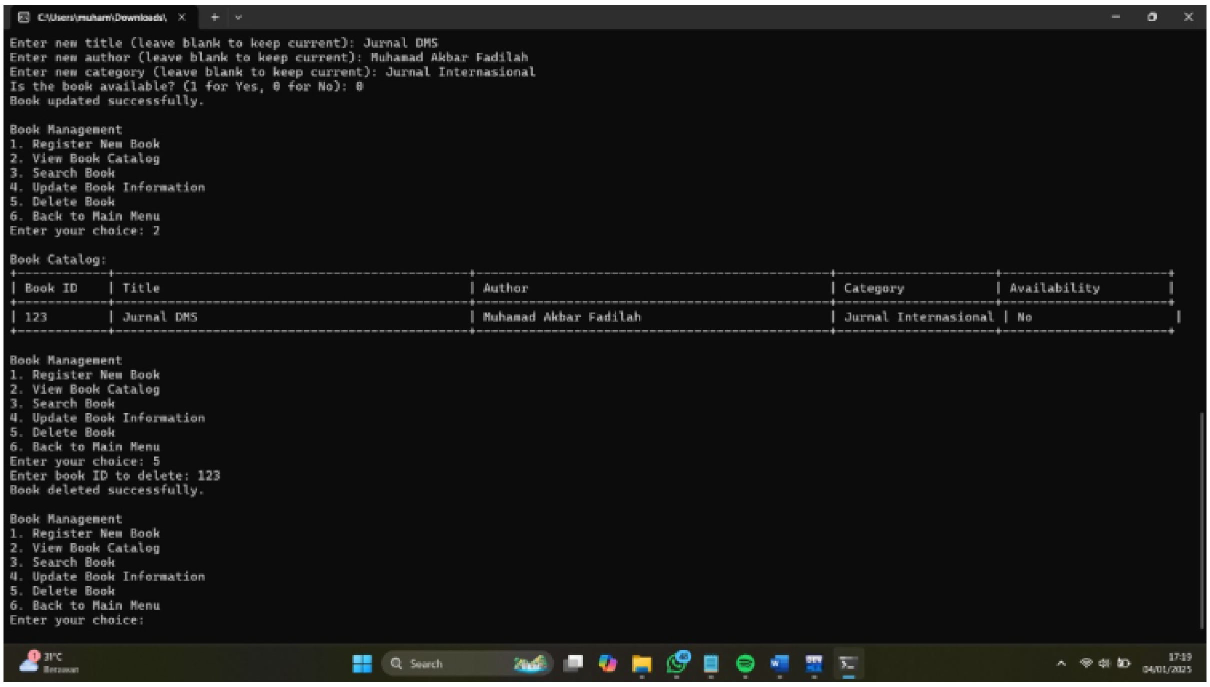
```
Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 3
Enter book title or author to search: Jurnal

Search Results:
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID     | Title                                          | Author                                         | Category             | Availability          |
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123         | Jurnal CRM                                     | Akbar Fadilah                                  | Jurnal               | Yes                   |
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 4
Enter book ID to update: 123
Enter new title (leave blank to keep current): Jurnal DMS
Enter new author (leave blank to keep current): Muhamad Akbar Fadilah
Enter new category (leave blank to keep current): Jurnal Internasional
Is the book available? (1 for Yes, 0 for No): 0
Book updated successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice:
```

Gambar 6 - Update Book Information



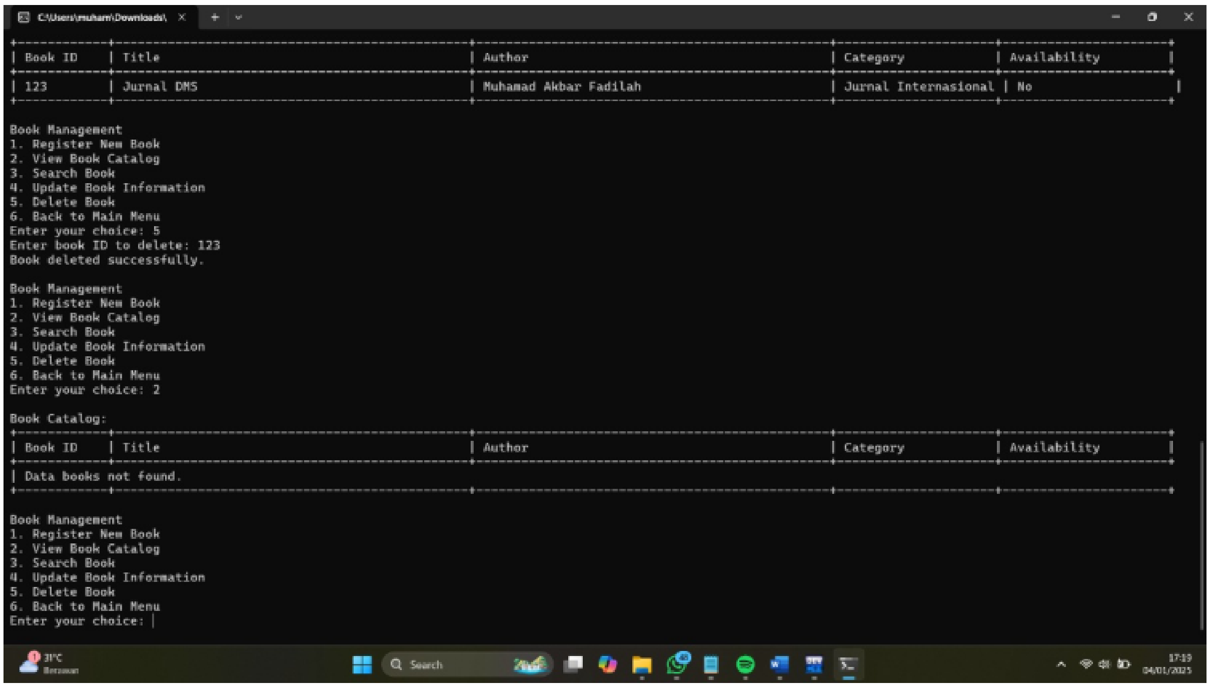
```
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 4
Enter book ID to update: 123
Enter new title (leave blank to keep current): Jurnal DMS
Enter new author (leave blank to keep current): Muhamad Akbar Fadilah
Enter new category (leave blank to keep current): Jurnal Internasional
Is the book available? (1 for Yes, 0 for No): 0
Book updated successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 2

Book Catalog:
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID     | Title                                          | Author                                         | Category             | Availability          |
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123         | Jurnal DMS                                     | Muhamad Akbar Fadilah                          | Jurnal Internasional | No                     |
+-------------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice:
```

Gambar 7- View Book Catalog (After Update)



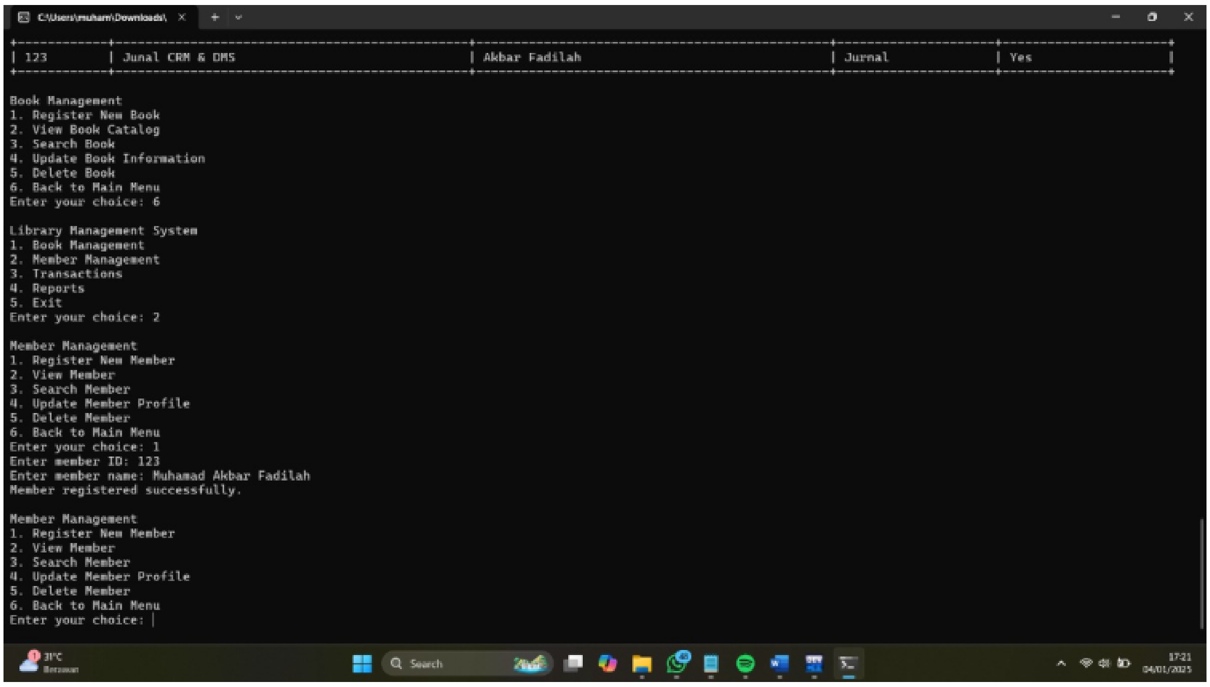
```
Enter new title (leave blank to keep current): Jurnal DMS
Enter new author (leave blank to keep current): Muhamad Akbar Fadilah
Enter new category (leave blank to keep current): Jurnal Internasional
Is the book available? (1 for Yes, 0 for No): 0
Book updated successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 2

Book Catalog:
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID   | Title                                          | Author                                         | Category             | Availability          |
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123       | Jurnal DMS                                     | Muhamad Akbar Fadilah                          | Jurnal Internasional | No                     |
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 5
Enter book ID to delete: 123
Book deleted successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice:
```

Gambar 8 - Delete Book



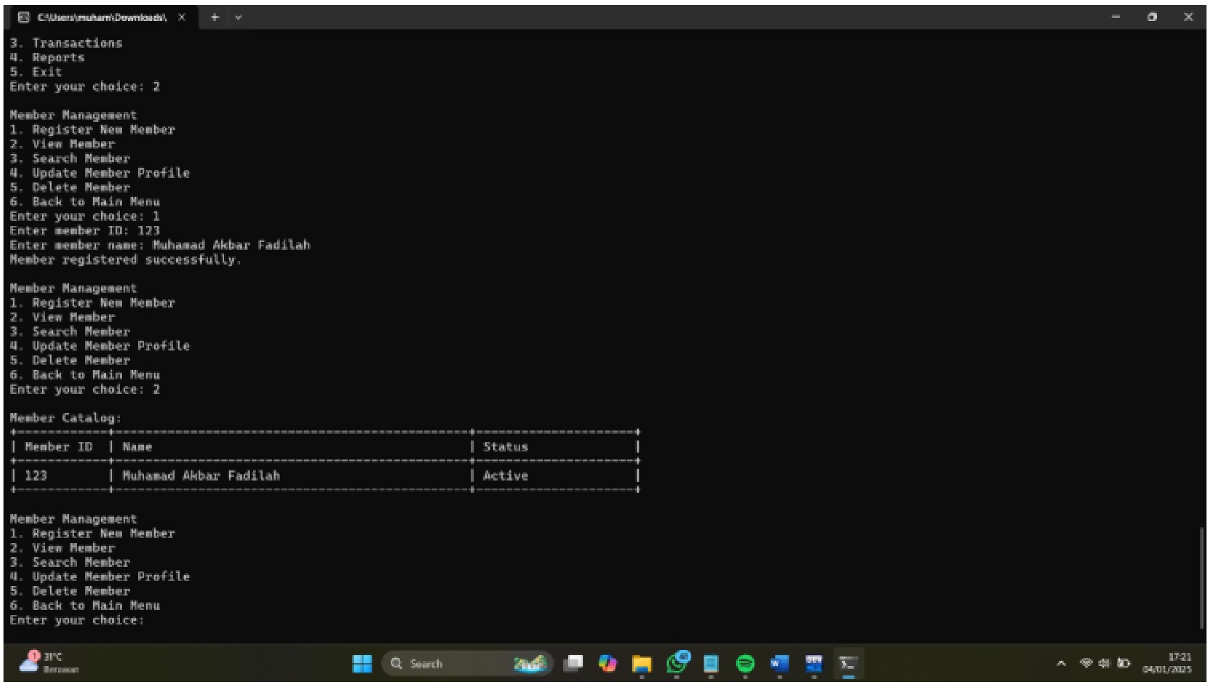
```
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID   | Title                                          | Author                                         | Category             | Availability          |
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| 123       | Jurnal DMS                                     | Muhamad Akbar Fadilah                          | Jurnal Internasional | No                     |
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 5
Enter book ID to delete: 123
Book deleted successfully.

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 2

Book Catalog:
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Book ID   | Title                                          | Author                                         | Category             | Availability          |
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+
| Data books not found.
+-----------+------------------------------------------------+------------------------------------------------+----------------------+-----------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: |
```

Gambar 8 - View Book Catalog (After Delete)



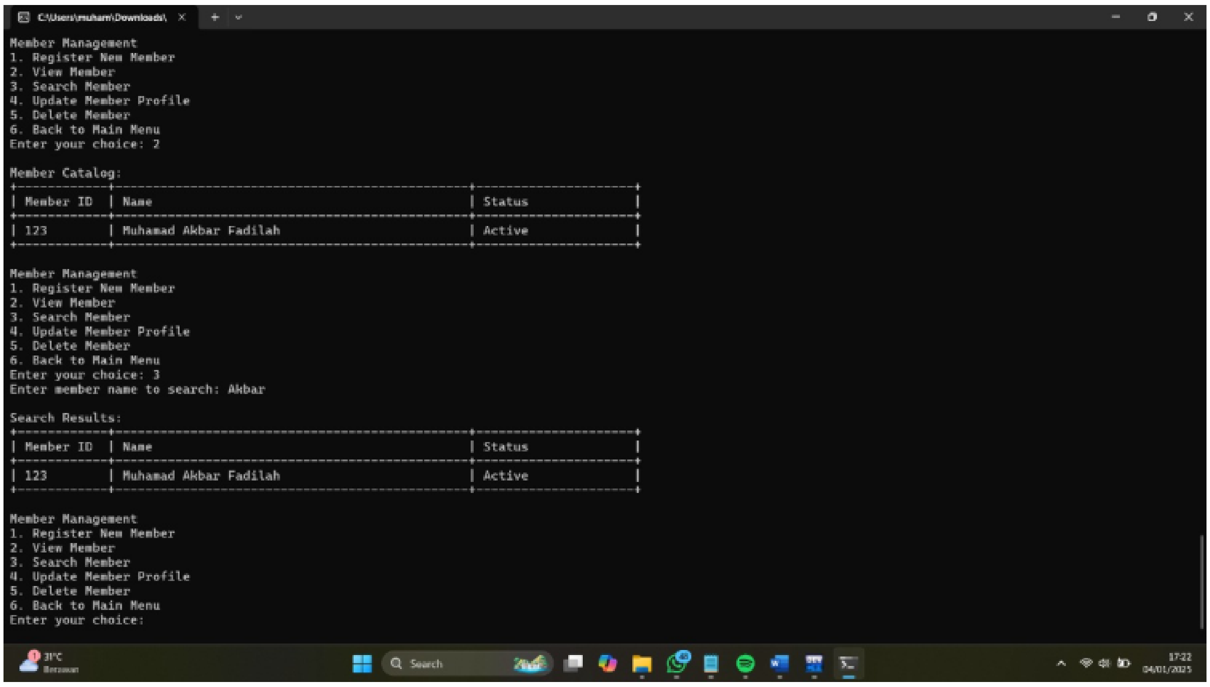
```
+-----------+----------------------------------------------+------------------------------------------------+----------------------+------------------------+
| 123       | Junal CRM & DMS                              | Akbar Fadilah                                  | Jurnal               | Yes                    |
+-----------+----------------------------------------------+------------------------------------------------+----------------------+------------------------+

Book Management
1. Register New Book
2. View Book Catalog
3. Search Book
4. Update Book Information
5. Delete Book
6. Back to Main Menu
Enter your choice: 6

Library Management System
1. Book Management
2. Member Management
3. Transactions
4. Reports
5. Exit
Enter your choice: 2

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 1
Enter member ID: 123
Enter member name: Muhamad Akbar Fadilah
Member registered successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: |
```

Gambar 9 - Register New Member



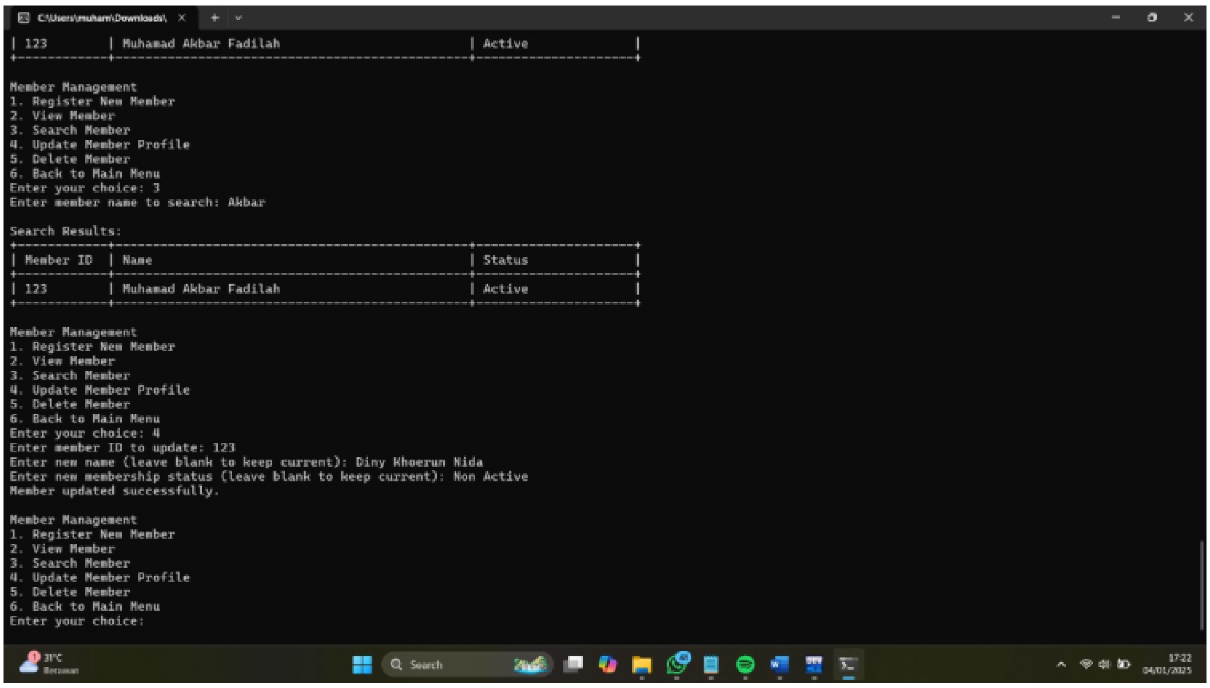
```
3. Transactions
4. Reports
5. Exit
Enter your choice: 2

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 1
Enter member ID: 123
Enter member name: Muhamad Akbar Fadilah
Member registered successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 2

Member Catalog:
+-----------+----------------------------------------------+---------------------+
| Member ID | Name                                         | Status              |
+-----------+----------------------------------------------+---------------------+
| 123       | Muhamad Akbar Fadilah                        | Active              |
+-----------+----------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice:
```

Gambar 9 - View Member



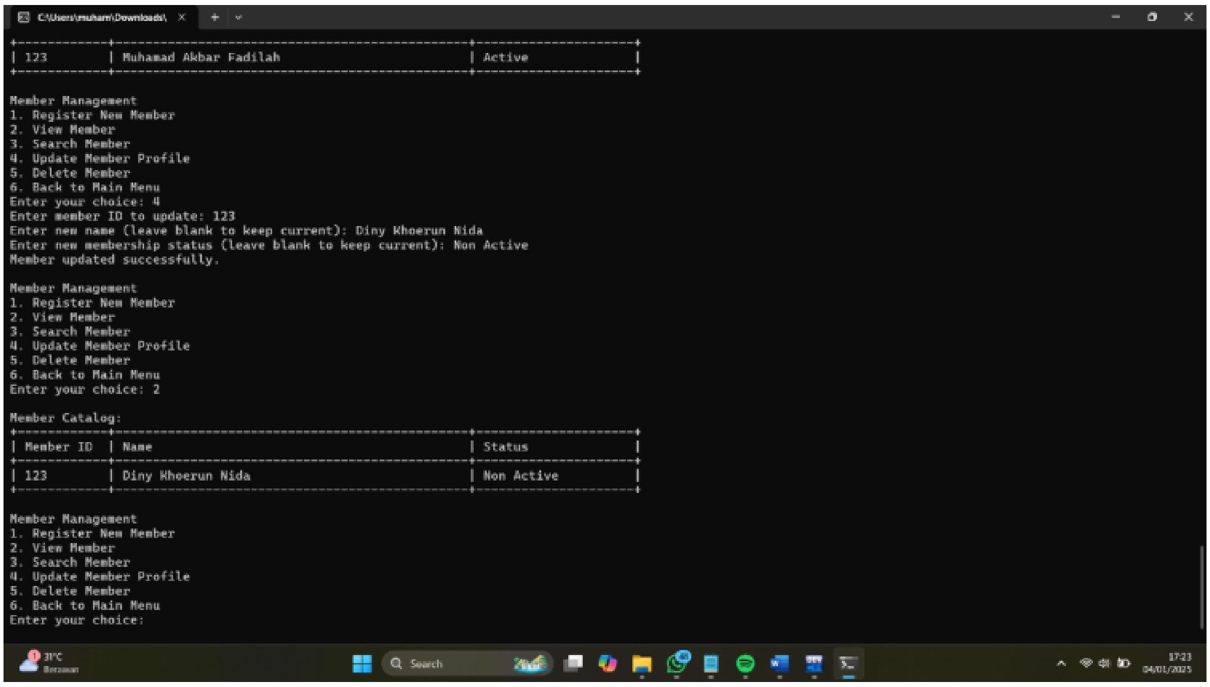
```
Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 2

Member Catalog:
+------------+------------------------------------------------+---------------------+
| Member ID  | Name                                           | Status              |
+------------+------------------------------------------------+---------------------+
| 123        | Muhamad Akbar Fadilah                          | Active              |
+------------+------------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 3
Enter member name to search: Akbar

Search Results:
+------------+------------------------------------------------+---------------------+
| Member ID  | Name                                           | Status              |
+------------+------------------------------------------------+---------------------+
| 123        | Muhamad Akbar Fadilah                          | Active              |
+------------+------------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice:
```

Gambar 10 - Search Member



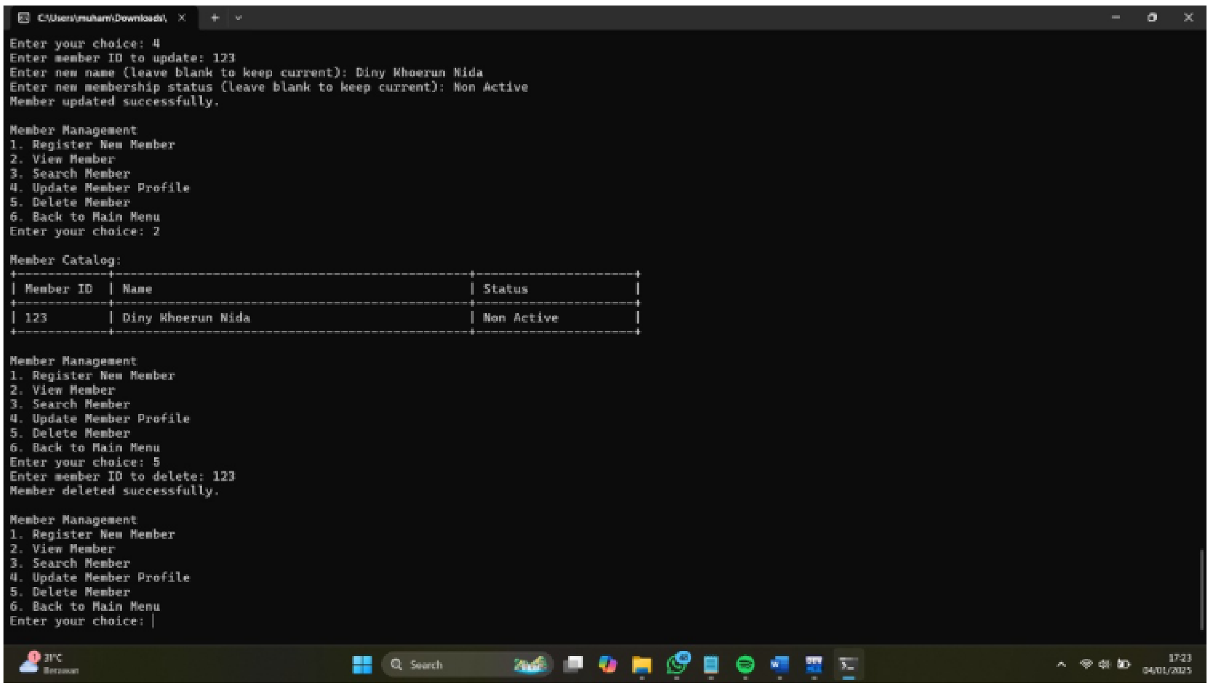
```
| 123        | Muhamad Akbar Fadilah                          | Active              |
+------------+------------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 3
Enter member name to search: Akbar

Search Results:
+------------+------------------------------------------------+---------------------+
| Member ID  | Name                                           | Status              |
+------------+------------------------------------------------+---------------------+
| 123        | Muhamad Akbar Fadilah                          | Active              |
+------------+------------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 4
Enter member ID to update: 123
Enter new name (leave blank to keep current): Diny Khoerun Nida
Enter new membership status (leave blank to keep current): Non Active
Member updated successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice:
```

Gambar 11 - Update Member Profile



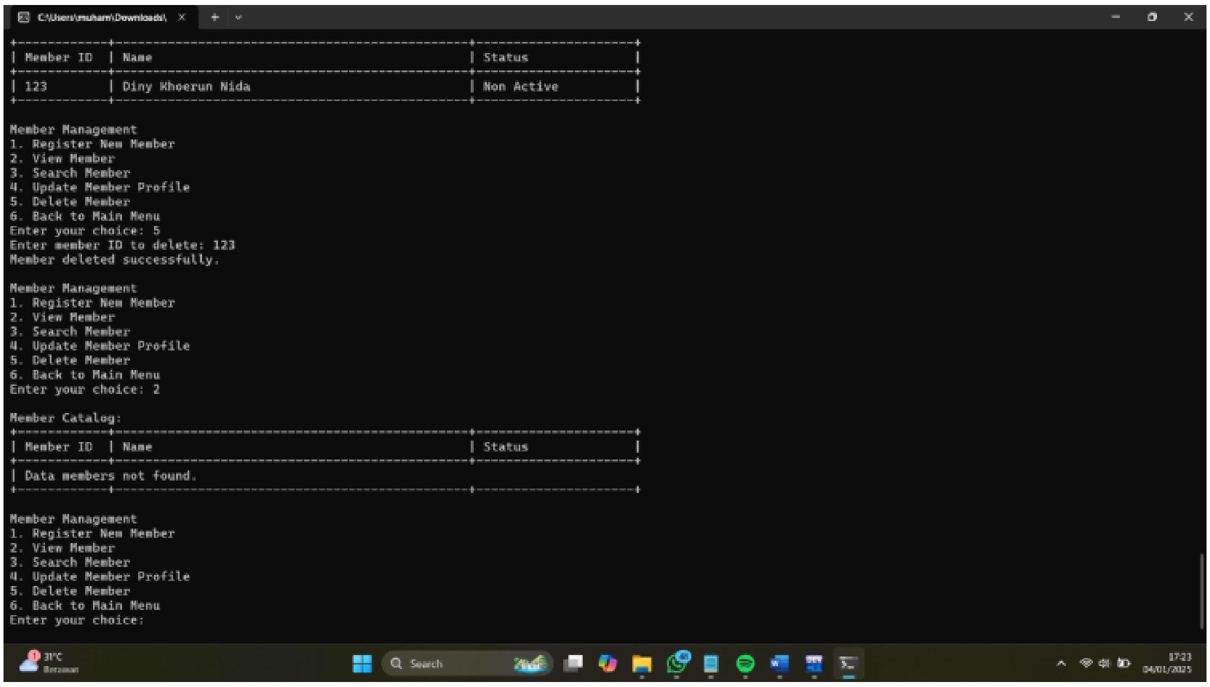
```
+------------+-----------------------------------------------+---------------------+
| 123        | Muhamad Akbar Fadilah                         | Active              |
+------------+-----------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 4
Enter member ID to update: 123
Enter new name (leave blank to keep current): Diny Khoerun Nida
Enter new membership status (leave blank to keep current): Non Active
Member updated successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 2

Member Catalog:
+------------+-----------------------------------------------+---------------------+
| Member ID  | Name                                          | Status              |
+------------+-----------------------------------------------+---------------------+
| 123        | Diny Khoerun Nida                             | Non Active          |
+------------+-----------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice:
```

Gambar 12 - View Member (After Update)



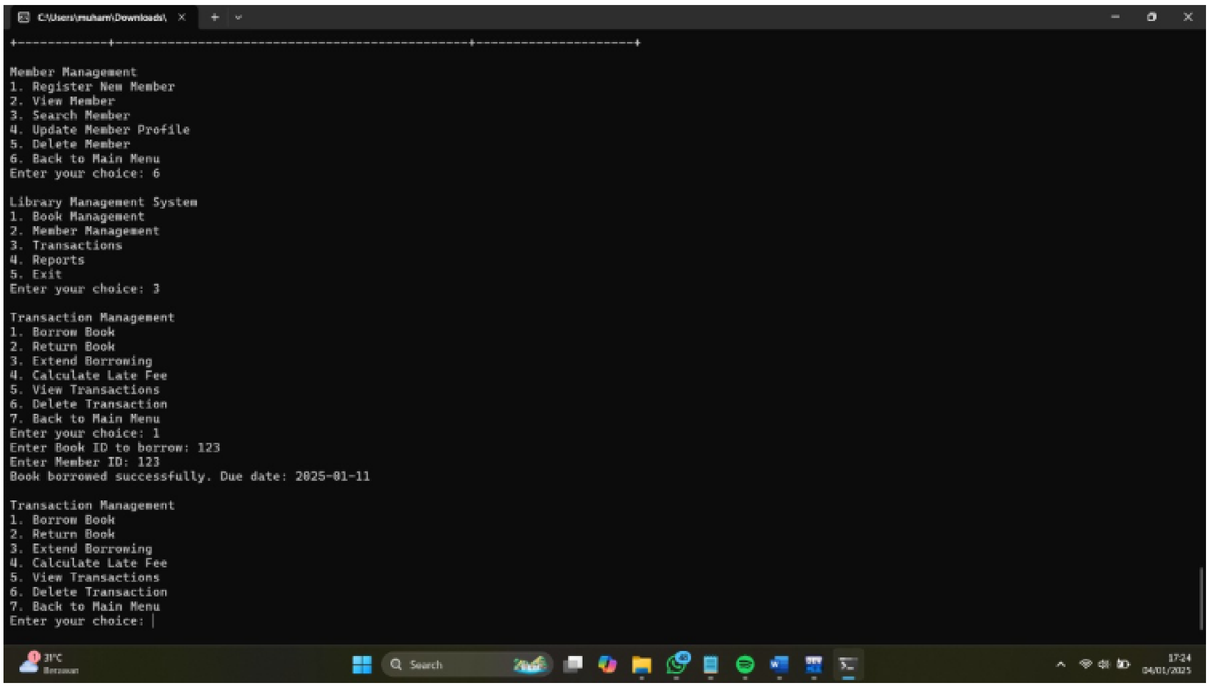
```
Enter your choice: 4
Enter member ID to update: 123
Enter new name (leave blank to keep current): Diny Khoerun Nida
Enter new membership status (leave blank to keep current): Non Active
Member updated successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 2

Member Catalog:
+------------+-----------------------------------------------+---------------------+
| Member ID  | Name                                          | Status              |
+------------+-----------------------------------------------+---------------------+
| 123        | Diny Khoerun Nida                             | Non Active          |
+------------+-----------------------------------------------+---------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 5
Enter member ID to delete: 123
Member deleted successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: |
```

Gambar 13 - Delete Member



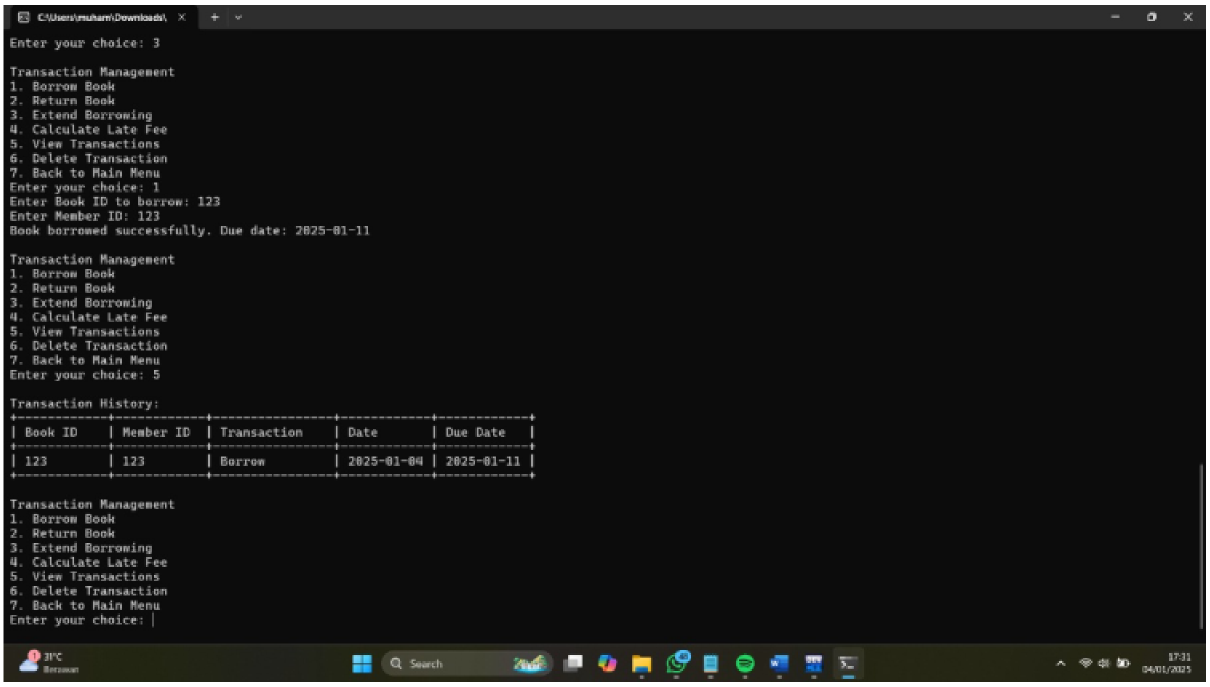
```
+------------+-----------------------------------------------+----------------------+
| Member ID  | Name                                          | Status               |
+------------+-----------------------------------------------+----------------------+
| 123        | Diny Khoerun Nida                             | Non Active           |
+------------+-----------------------------------------------+----------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 5
Enter member ID to delete: 123
Member deleted successfully.

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 2

Member Catalog:
+------------+-----------------------------------------------+----------------------+
| Member ID  | Name                                          | Status               |
+------------+-----------------------------------------------+----------------------+
| Data members not found.
+------------+-----------------------------------------------+----------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice:
```

Gambar 14 - View Member (After Delete)



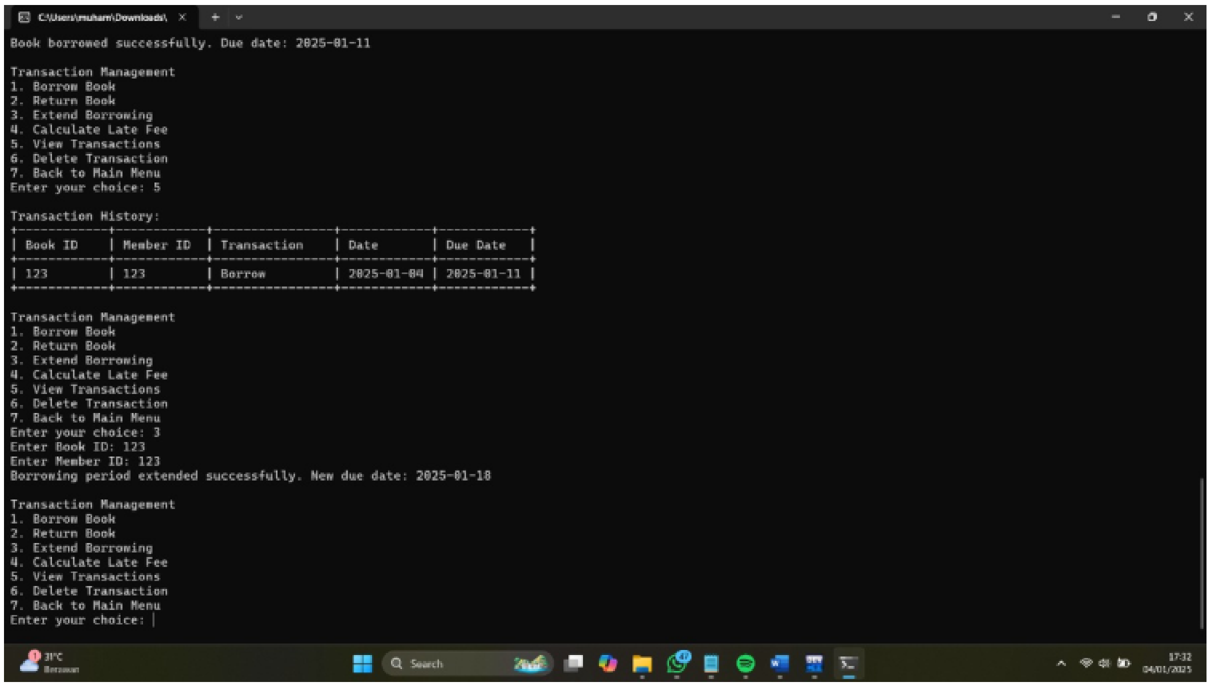
```
+------------+-----------------------------------------------+----------------------+

Member Management
1. Register New Member
2. View Member
3. Search Member
4. Update Member Profile
5. Delete Member
6. Back to Main Menu
Enter your choice: 6

Library Management System
1. Book Management
2. Member Management
3. Transactions
4. Reports
5. Exit
Enter your choice: 3

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 1
Enter Book ID to borrow: 123
Enter Member ID: 123
Book borrowed successfully. Due date: 2025-01-11

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice:
```

Gambar 15 - Borrow Book



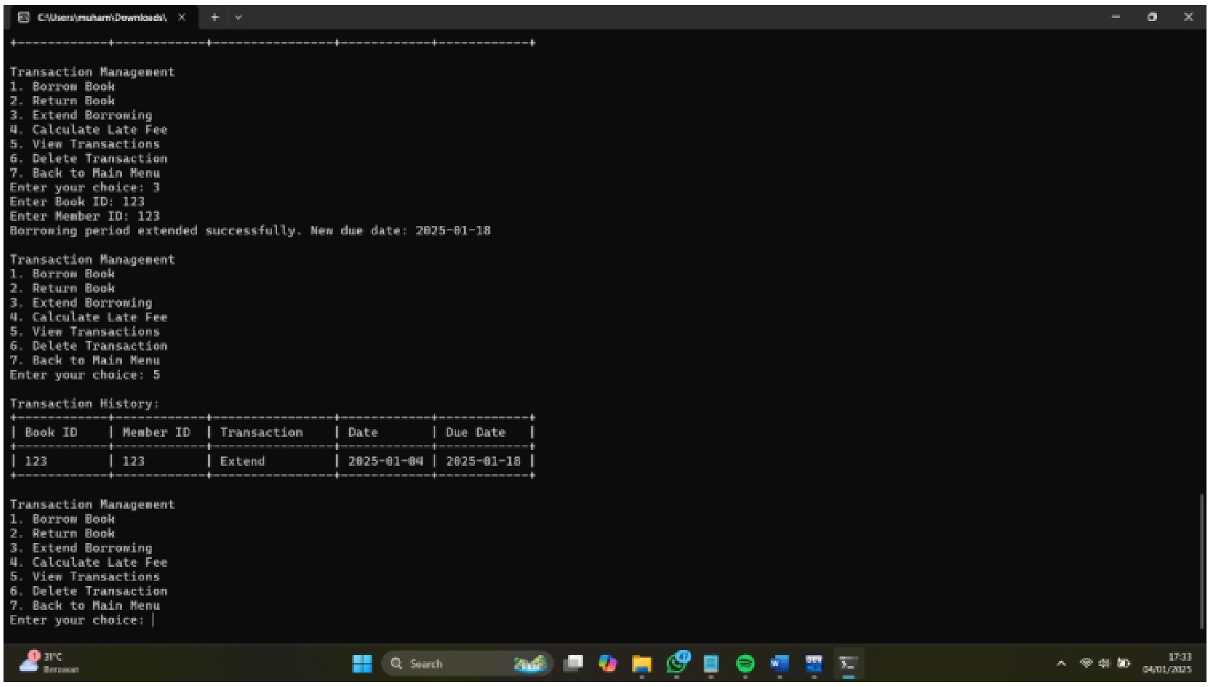
```
Enter your choice: 3

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 1
Enter Book ID to borrow: 123
Enter Member ID: 123
Book borrowed successfully. Due date: 2025-01-11

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 5

Transaction History:
+------------+------------+-----------------+------------+------------+
| Book ID    | Member ID  | Transaction     | Date       | Due Date   |
+------------+------------+-----------------+------------+------------+
| 123        | 123        | Borrow          | 2025-01-04 | 2025-01-11 |
+------------+------------+-----------------+------------+------------+

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice:
```

Gambar 16 - View Transactions (After Borrow Book)



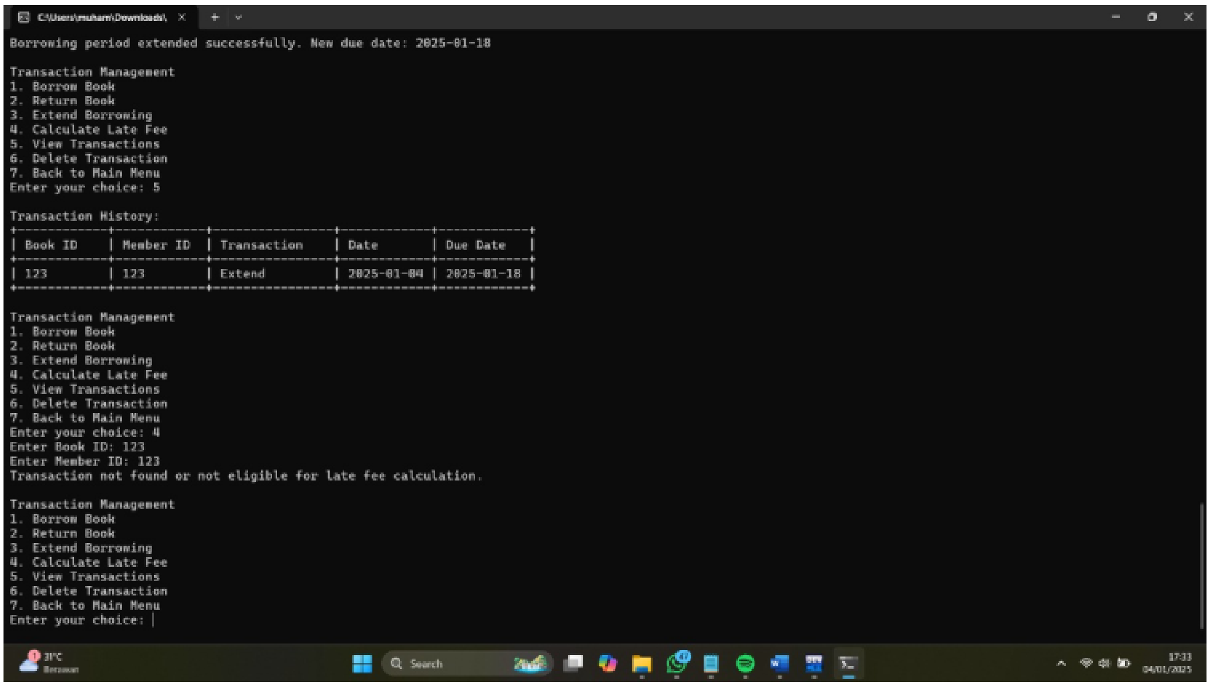
```
Book borrowed successfully. Due date: 2025-01-11

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 5

Transaction History:
+------------+------------+-----------------+------------+------------+
| Book ID    | Member ID  | Transaction     | Date       | Due Date   |
+------------+------------+-----------------+------------+------------+
| 123        | 123        | Borrow          | 2025-01-04 | 2025-01-11 |
+------------+------------+-----------------+------------+------------+

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 3
Enter Book ID: 123
Enter Member ID: 123
Borrowing period extended successfully. New due date: 2025-01-18

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice:
```

Gambar 17 - Extend Borrowing



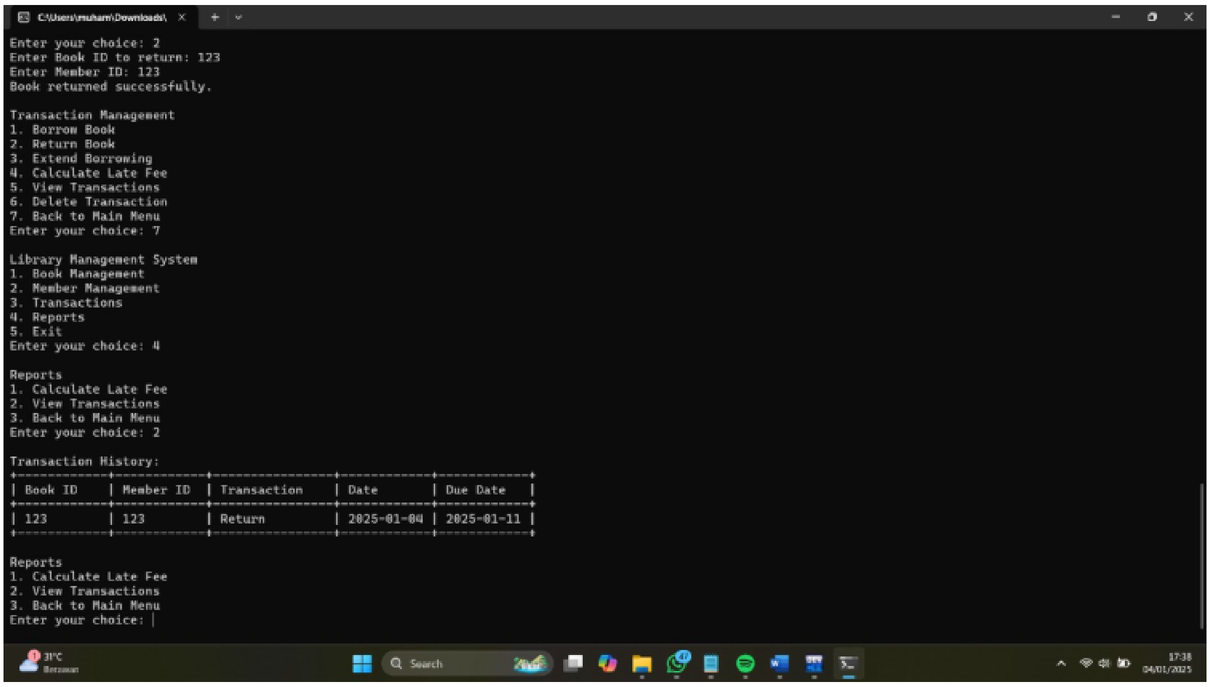
```
+------------+------------+----------------+------------+------------+


Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 3
Enter Book ID: 123
Enter Member ID: 123
Borrowing period extended successfully. New due date: 2025-01-18

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 5

Transaction History:
+------------+------------+----------------+------------+------------+
| Book ID    | Member ID  | Transaction    | Date       | Due Date   |
+------------+------------+----------------+------------+------------+
| 123        | 123        | Extend         | 2025-01-04 | 2025-01-18 |
+------------+------------+----------------+------------+------------+

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: |
```

Gambar 18 - View Transactions (After Extend Borrowing)



```
Borrowing period extended successfully. New due date: 2025-01-18

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 5

Transaction History:
+------------+------------+----------------+------------+------------+
| Book ID    | Member ID  | Transaction    | Date       | Due Date   |
+------------+------------+----------------+------------+------------+
| 123        | 123        | Extend         | 2025-01-04 | 2025-01-18 |
+------------+------------+----------------+------------+------------+

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 4
Enter Book ID: 123
Enter Member ID: 123
Transaction not found or not eligible for late fee calculation.

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: |
```

Gambar 19 - Return Book



```
Enter your choice: 2
Enter Book ID to return: 123
Enter Member ID: 123
Book returned successfully.

Transaction Management
1. Borrow Book
2. Return Book
3. Extend Borrowing
4. Calculate Late Fee
5. View Transactions
6. Delete Transaction
7. Back to Main Menu
Enter your choice: 7

Library Management System
1. Book Management
2. Member Management
3. Transactions
4. Reports
5. Exit
Enter your choice: 4

Reports
1. Calculate Late Fee
2. View Transactions
3. Back to Main Menu
Enter your choice: 2

Transaction History:
+------------+------------+----------------+------------+------------+
| Book ID    | Member ID  | Transaction    | Date       | Due Date   |
+------------+------------+----------------+------------+------------+
| 123        | 123        | Return         | 2025-01-04 | 2025-01-11 |
+------------+------------+----------------+------------+------------+

Reports
1. Calculate Late Fee
2. View Transactions
3. Back to Main Menu
Enter your choice: 
```

Gambar 20 - View Transaction (After Return)