**PROYEK PRAKTIKUM**

**(Project Team)**

* LO 1: Apply syntax and functions in the C language
* LO 2: Analyzing a program to solve problems using the C language.
* LO 3: Evaluate the kind of algorithms, syntax, and function for problem-solving

**Relevansi**

|  |  |  |
| --- | --- | --- |
| Week | Topic | Relevansi Praktikum |
| Week 1 | Introduction to Algorithm and Programming | Memberikan dasar konsep program, struktur kontrol, dan struktur data dasar yang mendasari seluruh tugas |
| Week 2 | Format Input and Output | Langsung digunakan dalam menu 1 dan 7: penggunaan scanf, printf untuk menerima dan menampilkan data |
| Week 3 | Operator, Operand, and Arithmetic | Diterapkan dalam menu 7: perhitungan total harga = jumlah × harga, serta validasi input numerik |
| Week 4 | Program Control: Selection and Repetition | Digunakan dalam navigasi menu, validasi input, pengulangan proses di seluruh menu 1–8 |
| Week 5 | Pointers and Array | Digunakan untuk menyimpan dan memanipulasi data buku dan transaksi menggunakan array of struct |
| Week 6 | Function and Recursion | Fungsi modular diperlukan untuk pemisahan logika tiap menu, termasuk sorting di menu 8 |
| Week 7 | Structures and Union | Struct digunakan untuk mendefinisikan tipe data buku dan histori penjualan |
| Week 8 | Memory Allocation | Diperlukan dalam manajemen file saat file tidak tersedia, serta alokasi memori dinamis jika diimplementasi |
| Week 9 | File Processing and Searching | Esensial untuk semua fitur yang membaca/menulis file (menu 2, 3, 4, 5, 6, 7) |
| Week 10 | Sorting | Langsung digunakan dalam menu 8: fitur pengurutan nama (bubble sort) dan harga (selection sort) |

**Toko Buku “Literasi Nusantara Jaya”** adalah toko buku skala menengah di kota Anda. Pemiliknya, Bapak Anwar, selama ini mencatat penjualan di buku tulis. Untuk mendukung digitalisasi, Anda dan tim diminta membuat **aplikasi monitoring transaksi** sederhana menggunakan **Bahasa C**.

**Kebutuhan:**

* Saat program dijalankan, baca data buku dari file databuku.txt. Pada data tersebut terdapat data buku antara lain: *kode buku, nama buku, jenis buku, dan harga buku*.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Selain itu, program juga akan membaca file history.txt untuk menampilkan histori transaksi penjualan. Format history.txt:

A white background with black text

AI-generated content may be incorrect.

**PENILAIAN (SCORING)**

* Tersedia menu utama:

|  |  |  |
| --- | --- | --- |
| Menu | Deskripsi | Bobot |
| 1️⃣ | Input Data Buku Baru | 20% |
| 2️⃣ | View History Transaksi Penjualan | 10% |
| 3️⃣ | View Buku (Seluruh Data Buku yang Tersedia) | 10% |
| 4️⃣ | Delete History (Hapus Histori Penjualan Tertentu) | 10% |
| 5️⃣ | Delete Buku (Hapus Data Buku Tertentu) | 10% |
| 6️⃣ | Exit (Simpan data terbaru & keluar) | 10% |
| 7️⃣ | Input Transaksi Penjualan | 15% |

Tabel 1. Field buku dan tipe data inputan buku baru

|  |  |
| --- | --- |
| Field Buku | Tipe Data |
| Kode Buku | String |
| Nama Buku | String |
| Jenis Buku | String |
| Harga Buku | float atau double |

**💡 *Saran: Kode buku dibuat secara otomatis oleh program.***

Tabel 2. Field history transaksi penjualan (*view*)

|  |  |
| --- | --- |
| Field History | Tipe Data |
| Kode Buku | String |
| Jumlah Terjual | Integer |
| Total Harga | float atau double |

**💡 *Saran: "Gunakan float atau double untuk menyimpan harga dengan desimal" (lebih cocok untuk rupiah yang bisa punya pecahan sen).***

**Penjelasan Menu dan Fungsionalitas**

1. **Menu 1:** Jika user memilih menu Input data buku (tekan tombol ‘1’), maka program akan:

* Meminta inputan buku (*nama buku, jenis buku, dan harga)*
* Tambahkan data buku ke daftar buku.
* Sistem akan **menghasilkan kode buku secara otomatis** dengan format: BK001, BK002, dst.
* Kode buku dibuat berdasarkan jumlah data buku yang sudah ada, ditambah 1.
* Contoh: jika buku terakhir adalah BK005, maka buku baru mendapat kode BK006.

1. **Menu 2:** Jika user memilih menu *View History* (tekan tombol ‘2’), maka program akan menampilkan seluruh data histori transaksi yang pernah dilakukan pada history.txt.
2. **Menu 3:** Jika user memilih menu *View Buku* (tekan tombol ‘3’), maka program akan menampilkan seluruh data buku dari databuku.txt.
3. **Menu 4:** Jika user memilih menu *Delete History* (tekan tombol ‘4’), maka program akan menampilkan list data history penjualan.
   * Tampilkan daftar histori transaksi.
   * Meminta inputan index. Validasikan input minimal 1 dan maksimal sebanyak jumlah data.
   * Hapus data sesuai dengan index yang diinput. Contoh: Jika user memilih index 1, maka hapus data yang pertama. Lalu Tampilkan pesan “Data Successfully delete..”.
4. **Menu 5:** Jika user memilih menu *Delete Buku* (tekan tombol ‘5’), maka program akan menampilkan list data buku.
   * Tampilkan daftar buku.
   * Meminta inputan index. Validasikan input minimal 1 dan maksimal sebanyak jumlah data
   * Hapus data sesuai dengan index yang diinput. Contoh: Jika user memilih index 1, maka hapus data yang pertama. Lalu Tampilkan pesan “Data Successfully delete..”.
5. **Menu 6:** Jika user memilih menu *Exit* (tekan tombol ‘6’), maka program akan **menulis data** tersebut ke dalam file ”databuku.txt” dan program selesai dijalankan.

* Simpan semua perubahan ke databuku.txt dan history.txt
* Keluar dari program.

1. **Menu 7:** Jika user memilih menu *Input Transaksi Penjualan* (tekan tombol ‘7’).

* User diminta input **kode buku** dan **jumlah terjual**.
* Program mencari harga buku dari daftar.
* Hitung total harga = jumlah × harga.
* Tambahkan data ke daftar history (dan simpan ke history.txt).
* Tampilkan pesan: Transaksi berhasil dicatat.

**Instruksi Umum – Penting Dibaca Sebelum Mengerjakan**

Untuk proyek praktikum ini, **file databuku.txt dan history.txt tidak akan disediakan di awal oleh dosen**.

Oleh karena itu, perhatikan hal berikut:

1. **Program harus mampu menangani kondisi di mana file databuku.txt dan history.txt belum ada** (misalnya saat pertama kali program dijalankan).
2. Jika file tidak ditemukan saat dibuka (misalnya dengan mode "r" atau "r+"), maka program harus:
   * **Tidak error atau crash**
   * **Menginisialisasi data dari kondisi kosong** (array/list kosong)
3. Setelah pengguna melakukan input buku atau transaksi dan memilih menu Exit, program harus:
   * **Membuat file databuku.txt dan history.txt secara otomatis**
   * **Menyimpan data ke dalam file sesuai format yang telah ditentukan**
4. Dengan pendekatan ini, **program akan benar-benar fleksibel**, bisa berjalan meskipun belum ada file data dari awal, dan **dapat digunakan kembali di sesi berikutnya**.

**Soal Praktikum Tambahan – (Sorting Algorithms)**

**Judul:**  
*Optimalisasi Proses Sorting Data Buku pada Literasi Nusantara Jaya*

**Deskripsi Studi Kasus:**

Setelah menjalankan aplikasi transaksi buku selama 2 bulan, pemilik toko *Literasi Nusantara Jaya* meminta tim pengembang untuk menambahkan fitur **pengurutan data buku berdasarkan harga dan nama buku** agar lebih mudah dilihat dan dianalisis.

**Tugas Anda:**

Tambahkan fitur **menu baru** dengan deskripsi berikut:

|  |  |  |
| --- | --- | --- |
| Menu | Deskripsi | Bobot |
| 8️⃣ | Sort Buku (berdasarkan nama dan harga) | 15% |

Fitur ini akan:

1. Menyediakan sub-menu:
   * (a) Urutkan berdasarkan **nama buku** (ascending, bubble sort)
   * (b) Urutkan berdasarkan **harga buku** (descending, selection sort)
2. Menampilkan hasil pengurutan ke layar setelah selesai.

**Spesifikasi Teknis:**

* Gunakan **fungsi terpisah** untuk masing-masing algoritma sorting.
* Gunakan **struktur data array of struct** untuk menyimpan buku.
* Validasi jika data buku masih kosong → tampilkan pesan: *"Tidak ada data buku untuk diurutkan."*
* Tambahkan opsi kembali ke menu utama.

## PENGUMPULAN TUGAS:

1. Tulis kode program menggunakan bahasa pemrograman C.
2. Simpan file dengan nama sesuai instruksi, misalnya: algo\_project.c.
3. Kompres file .c, beserta file data (jika terbentuk), ke dalam folder .zip.  
   Sertakan informasi anggota kelompok di bagian komentar atas program. Jika LMS tidak memungkinkan untuk pengunggahan .zip, maka kirimkan dalam bentuk link Google Drive dan unggah file pengerjaan Anda disitu.
4. Tambahkan **komentar pada setiap fungsi utama**.
5. Pastikan program dapat **dikompilasi dan dijalankan tanpa kesalahan**.
6. Kumpulkan file melalui platform yang telah ditentukan oleh dosen.