

ALGORITMA DAN PEMROGRAMAN

(CSF101)

**APLIKASI MEMBUAT INVENTARIS UNTUK SEBUAH
PERUSAHAAN**

LAPORAN TUGAS AKHIR



Disusun oleh:

KELOMPOK 2

David boy

20230801547

Happy Adrian Pradana P

20230801539

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS

ESA UNGGUL JAKARTA

2023

KATA PENGANTAR

Puji dan syukur senantiasa saya ucapkan atas kehadiran Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penyusunan laporan tugas pembuatan aplikasi membuat inventaris untuk sebuah Perusahaan dapat dilakukan dengan lancar.

Tujuan dari penulisan laporan ini adalah untuk memenuhi nilai tugas akhir semester mata kuliah Algortima dan Pemrograman. Kami mengucapkan terima kasih kepada pihak yang telah berkontribusi dan juga kepada Ibu Hani Dewi Ariessanti S,kom, M.Kom selaku dosen pengampu pada mata kuliah Algortima dan Pemrograman.

Kami menyadari bahwa masih banyak kekurangan pada laporan ini. Maka dari itu kami memohon maaf kepada pembaca apabila masih menemukan kekurangan dalam penulisan. Serta kami sangat mengharapkan kritik dan saran yang dapat membangun laporan ini.

Jakarta, 23 Juli 2024

Kelompok 2

DAFTAR ISI

BAB I.....	1
1.1 Latar Belakang Laporan	1
1.2 Batasan Kerja Penelitian	1
BAB II.....	2
2.1 Pembagian kerja.....	2
2.2 Aktifitas Kegiatan	2
BAB III	4
3.1 Gambaran Project.....	4
3.2 Konsep Aplikasi	4
3.3 Alur aplikasi	4
BAB IV	5
4.1 Ruang Lingkup Project	5
4.2 Tujuan Proyek	5
4.3 Fungsi Utama	5
4.4 Flowchart	7
4.5 Code	13
4.6 Output	21
BAB V	27
5.1 Kesimpulan	27
5.2 Saran	28

BAB I

1.1 Latar Belakang Laporan

Teknologi ialah ilmu pengetahuan yang mempelajari tentang keterampilan dalam membuat sebuah alat sampai metode pengolahan untuk membantu menyelesaikan berbagai pekerjaan manusia. Perkembangan teknologi adalah hal yang selalu berkembang setiap saat. Dengan adanya perkembangan teknologi dapat memudahkan manusia dalam menjalankan aktivitas sehari-hari. Teknologi juga dapat menjadi bagian yang penting dalam dunia bisnis. Hampir dibanyak perusahaan baik perusahaan yang besar ataupun kecil telah banyak menggunakan teknologi dalam menjalankan bisnisnya. Dengan adanya teknologi tersebut maka ke-efektifan aktivitas dalam sebuah perusahaan besar ataupun kecil dapat berjalan secara efektif dan efisien. Contoh dari teknologi untuk menyelesaikan sebuah permasalahan pada suatu perusahaan adalah dengan menciptakan sebuah aplikasi pada sebuah komputer yang dapat memudahkan perusahaan dalam mengoprasionalkan sebuah pekerjaan seperti membuat aplikasi membuat inventaris untuk sebuah Perusahaan.

Aplikasi inventaris adalah perangkat lunak yang digunakan untuk mengatur dan memantau persediaan barang di dalam suatu perusahaan atau organisasi. Aplikasi ini membantu melacak pergerakan barang, mulai dari penerimaan, penyimpanan, penggunaan, hingga penghapusan.

1.2 Batasan Kerja Penelitian

1. Pengerjaan aplikasi ini menggunakan bahasa pemrograman C++
- 2 Aplikasi ini berjalan menggunakan PC/Komputer
- 3 Aplikasi ini untuk melakukan mengatur dan memantau persediaan barang dalam suatu Perusahaan

BAB II

PROJECT TIM AND MANAGEMENT

2.1 Pembagian kerja

NO	Nama	Kegiatan
1	David boy (20230801547)	<ul style="list-style-type: none"> • Membuat laporan bab I - III • Menyusun laporan • Mengecek apakah aplikasi sudah sesuai atau belum • Memperbaiki bug jika ditemukan • Membuat coding
2	Happy Adrian Pradana (20230801539)	<ul style="list-style-type: none"> • Mengecek apakah codingan sudah siap digunakan • Membuat alur kerja codingan • Mengecek apakah sudah sesuai standar yang diperlukan • Membuat laporan bab IV - VI

2.2 Aktifitas Kegiatan

Waktu	Kegiatan	Diselesaikan
22 July 2024 (Hari ke -1)	<ul style="list-style-type: none"> • Merancang flowchart • Membuat laporan • Melihat Tutorial Youtube • Membuat rangkaian program dimulai dari fungsi sederhana 	<ul style="list-style-type: none"> • Melihat Tutorial youtube • Membuat rangkaian program sederhana
23 July 2024 (Hari ke -2)	<ul style="list-style-type: none"> • Membuat fungsi Create • Melihat Tutorial di youtube • Membuat flowchart • Memebuat fungsi read 	<ul style="list-style-type: none"> • Membuat rangkaian program sederhana • Memabuat fungsi create • Membuat fungsi read
24 July 2024 (Hari ke -3)	<ul style="list-style-type: none"> • Membuat fungsi Undo • Membuat fungsi Delete • Melihat Tutorial dari youtube/ChatGPT • Membuat tampilan program 	<ul style="list-style-type: none"> • Fungsi undo dan delete sudah selesai • Membuat tampilan awal program • Selesai Flowchart

25 July 2024 (Hari ke -4)	<ul style="list-style-type: none"> • Menyusun laporan • Membuat semua struktur program 	<ul style="list-style-type: none"> • Membuat semua struktur program
26 July 2024 (Hari ke -5)	<ul style="list-style-type: none"> • Menemukan sebuah bug pada program • Menyusun laporan • Memperbaiki bug 	<ul style="list-style-type: none"> • Memperbaiki bug yang sudah di temukan • Menyelesaikan laporan

BAB III CONCEPT PROJECT

3.1 Gambaran Project

Konsep project yang telah diangkat kelompok 2 yaitu mengenai tentang mengenai aplikasi inventaris. Apa itu inventaris?, inventaris adalah persediaan atau inventori menurut kajian industri dan manufaktur mengacu pada stok dari suatu item atau sumber daya yang digunakan dalam suatu organisasi perusahaan. Persediaan dalam manufaktur umumnya berupa item atau barang yang berkontribusi atau akan menjadi bagian dari keluaran produk perusahaan.

3.2 Konsep Aplikasi

Aplikasi yang kami buat memiliki fitur dimana Pengguna bisa mengelola data barang yang dapat mengakses sebuah menu. Dalam menu, Pengguna dapat menentukan pilihan-pilihan yang mau dipilih. Di menu Pengguna bisa menambahkan barang, mencari barang berdasarkan ID, menghapus barang, mengupdate barang, dan menampilkan barang yang sudah di datakan, menggunakan system CRUD (Create,Read,Update,Delete).

3.3 Alur aplikasi

Berdasarkan penjelasan gambaran dan konsep diatas, maka dibuatlah program aplikasi inventaris dengan alur/tahapan sebagai berikut:

1. Seseorang Pengguna akan dihadapkan dengan menu utama, Pengguna harus memilih salah satu diantara sub-menu sebagai berikut
 - Input barang
Pengguna harus memasukan data lengkap barang yang akan di datakan.
 - Tampil data barang
Menampilkan keseluruhan table barang yang telah diinput sebelumnya oleh Pengguna.
 - Menghapus data barang
Menghapus data barang yang sudah tidak tersedia.
 - Mengedit data barang
Mengedit atau melakukan perubahan stok, harga barang
 - Kembali ke main menu

BAB IV

RUANG LINGKUP PROJECT, IMPLEMENTASI APLIKASI, DAN OVERVIEW

4.1 Ruang Lingkup Project

Proyek ini bertujuan untuk membuat sebuah aplikasi berbasis konsol yang memungkinkan pengguna untuk mengelola data barang di sebuah toko. Aplikasi ini akan memungkinkan pengguna untuk menambah, menghapus, mencari, mengedit, dan menampilkan data barang yang disimpan dalam file biner "data.bin". Program ini menggunakan bahasa pemrograman C++ dan mengimplementasikan struktur data barang menggunakan struc.

4.2 Tujuan Proyek

1. Membuat sistem untuk mengelola data barang.
2. Mengimplementasikan operasi CRUD (Create, Read, Update, Delete) untuk data barang.
3. Menyimpan dan membaca data barang dari file.

4.3 Fungsi Utama

1. Menambah Data Barang

- Fungsi: MenambahData(fstream &data)
- Menambahkan data barang baru ke dalam file "data.bin".
- Meminta input dari pengguna untuk nama barang, ID barang, jumlah barang, dan harga barang.
- Menyimpan data barang yang baru ditambahkan ke dalam file biner.

2. Menghapus Data Barang

- Fungsi: HapusData(fstream &data, int id)
- Menghapus data barang berdasarkan ID yang diberikan oleh pengguna.
- Menulis ulang data barang yang tidak dihapus ke file sementara, kemudian mengganti file lama dengan file sementara.

3. Mencari Barang Berdasarkan ID

- Fungsi: CariData(fstream &data, int id)
- Mencari dan menampilkan data barang berdasarkan ID yang dimasukkan oleh pengguna.
- Menampilkan detail barang yang ditemukan (nama, ID, jumlah, harga).

4. Mengedit atau Memperbarui Data Barang

- Fungsi: EditData(fstream &data, int id)
- Mengedit data barang yang ada berdasarkan ID yang dimasukkan oleh pengguna.
- Meminta input baru dari pengguna untuk memperbarui data barang (nama, jumlah, harga).
- Menyimpan perubahan ke dalam file biner.

5. Menampilkan Data Barang

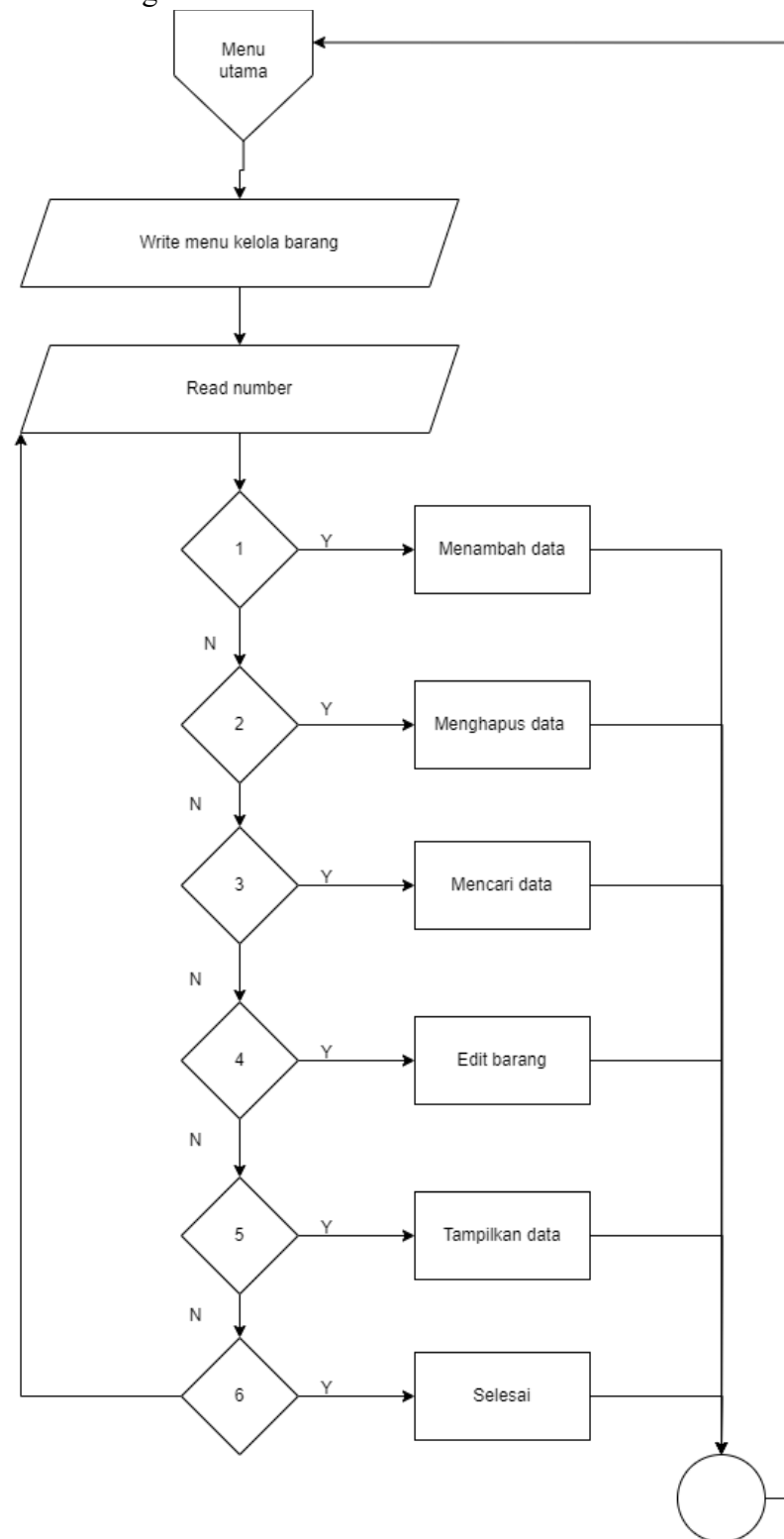
- Fungsi: TampilData(fstream &data)
- Menampilkan semua data barang yang ada dalam file "data.bin".
- Menampilkan data dalam format tabel yang rapi dengan nomor, nama barang, ID, dan harga barang.

4.4 Flowchart

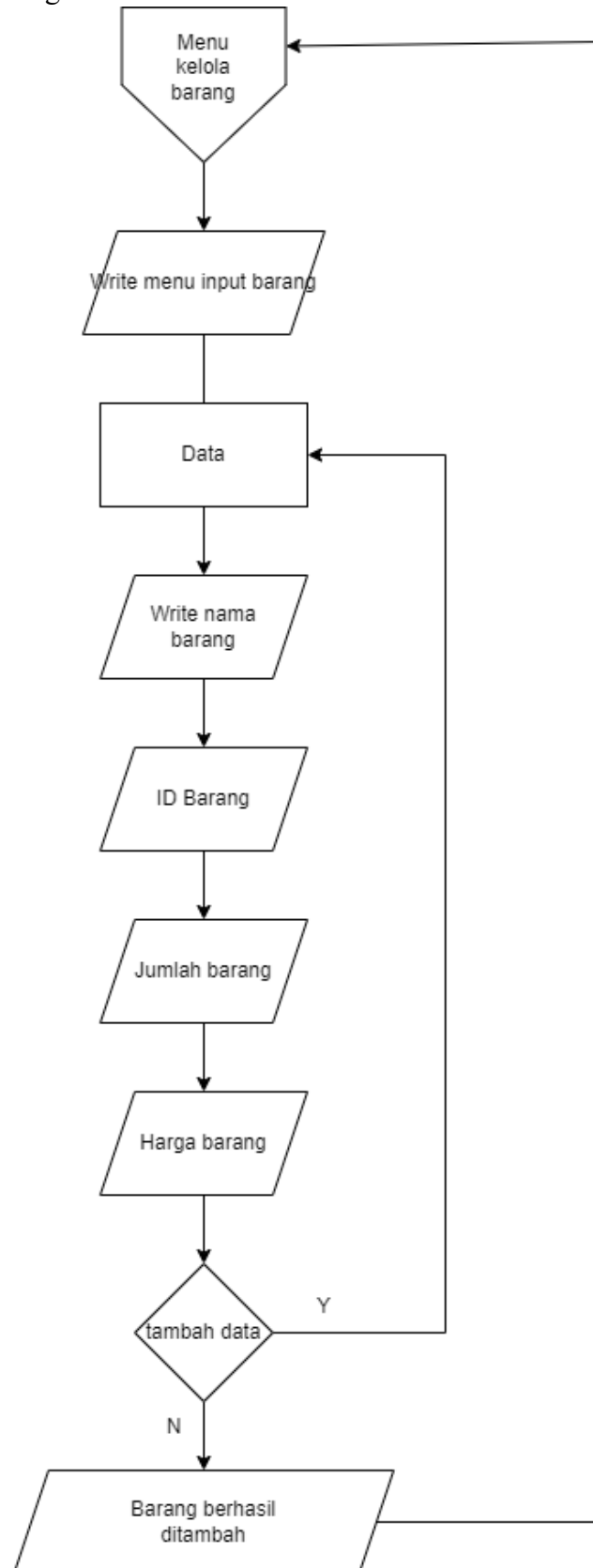
1. Menu utama



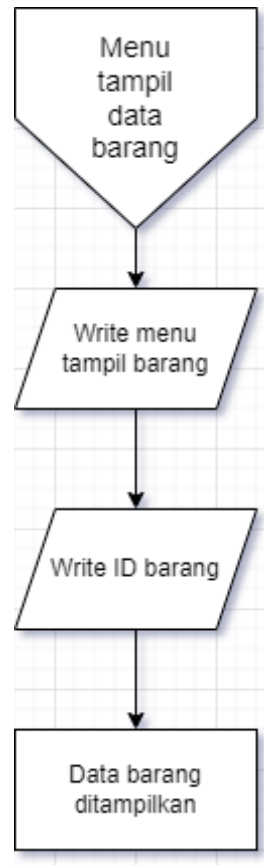
2. Menu Kelola barang utama



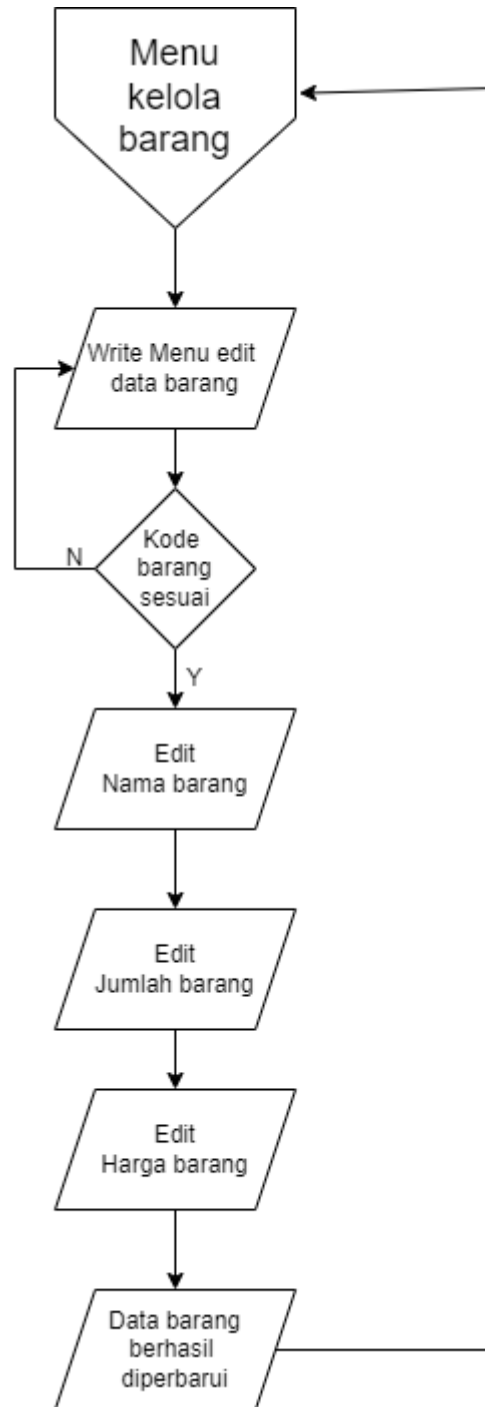
3. Menu Input barang



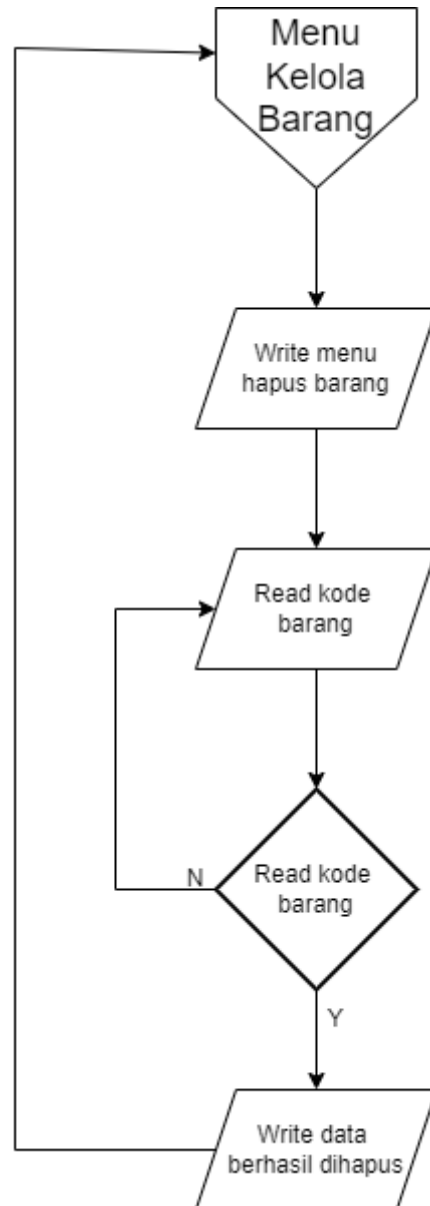
4. Menu Tampil Barang



5. Menu Edit Barang



6. Menu Hapus Barang



4.5 Code

LINK SOURCE CODE : <https://github.com/ReysGG/Algortima-dan-pemograman-bahasa>

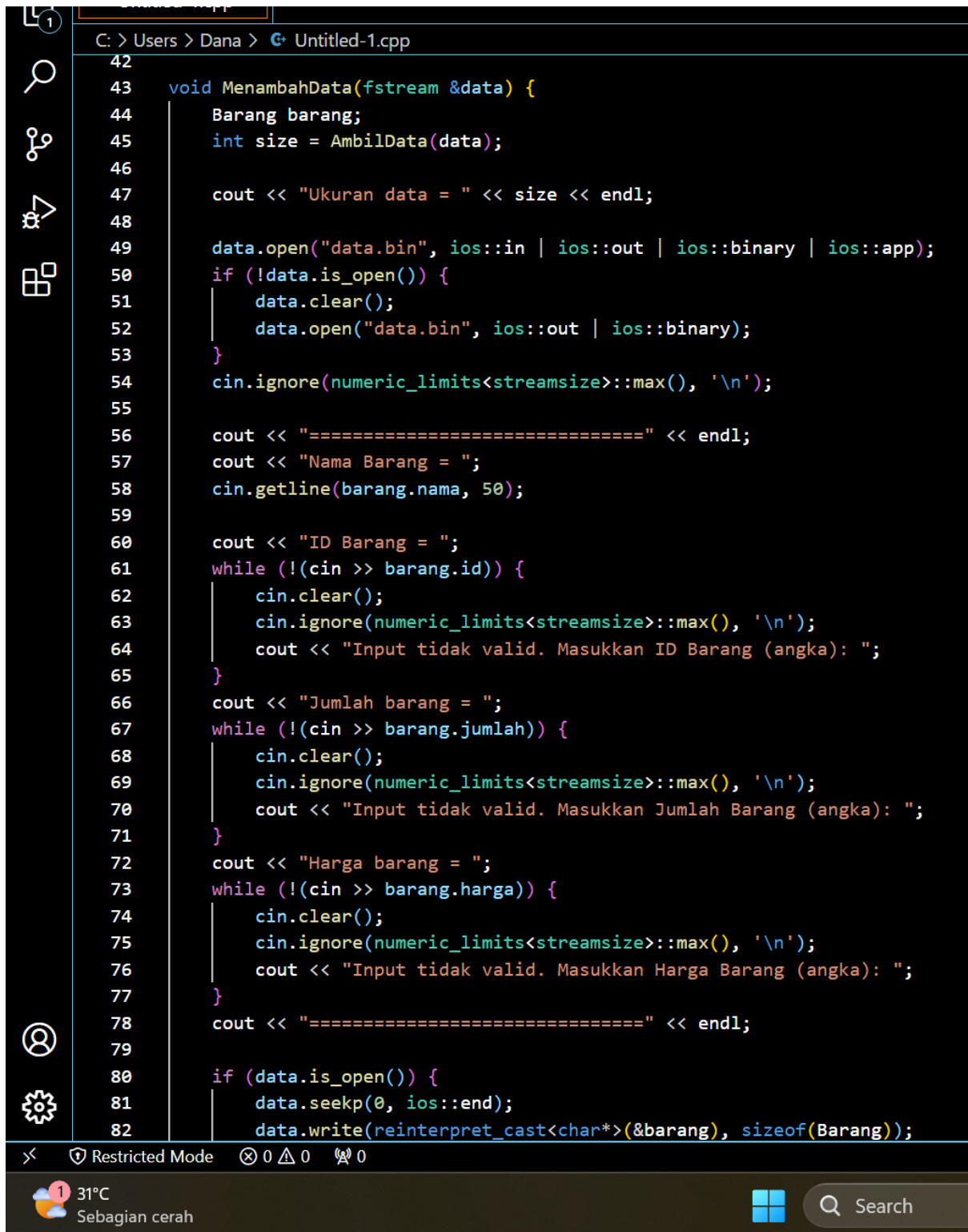
1. Fungsi Menu

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <limits>
5  #include <iomanip>
6  #include <cstring>
7
8  using namespace std;
9
10 struct Barang {
11     char nama[50];
12     int id, jumlah;
13     double harga;
14 };
15
16 int menu() {
17     int input;
18     cout << "==== PROGRAM CRUD BARANG =====< endl;
19     cout << "=====< endl;
20     cout << "[1]. Menambah data barang" << endl;
21     cout << "[2]. Menghapus data barang" << endl;
22     cout << "[3]. Mencari barang berdasarkan Id" << endl;
23     cout << "[4]. Mengedit atau memperbarui data barang" << endl;
24     cout << "[5]. Menampilkan data barang" << endl;
25     cout << "[6]. Selesai" << endl;
26     cout << "=====< endl;
27     cout << "Masukkan nomor yang ingin dipilih = ";
28     cin >> input;
29     return input;
30 }
31
```

2. Fungsi Ambil Data

```
30 }
31
32 int AmbilData(fstream &data) {
33     data.open("data.bin", ios::in | ios::binary);
34     if (!data.is_open()) {
35         return 0;
36     }
37     data.seekg(0, ios::end);
38     int size = data.tellg() / sizeof(Barang);
39     data.close();
40     return size;
41 }
```


3. Fungsi Menambah Data



```
42
43 void MenambahData(fstream &data) {
44     Barang barang;
45     int size = AmbilData(data);
46
47     cout << "Ukuran data = " << size << endl;
48
49     data.open("data.bin", ios::in | ios::out | ios::binary | ios::app);
50     if (!data.is_open()) {
51         data.clear();
52         data.open("data.bin", ios::out | ios::binary);
53     }
54     cin.ignore(numeric_limits<streamsize>::max(), '\n');
55
56     cout << "=====" << endl;
57     cout << "Nama Barang = ";
58     cin.getline(barang.nama, 50);
59
60     cout << "ID Barang = ";
61     while (!(cin >> barang.id)) {
62         cin.clear();
63         cin.ignore(numeric_limits<streamsize>::max(), '\n');
64         cout << "Input tidak valid. Masukkan ID Barang (angka): ";
65     }
66     cout << "Jumlah barang = ";
67     while (!(cin >> barang.jumlah)) {
68         cin.clear();
69         cin.ignore(numeric_limits<streamsize>::max(), '\n');
70         cout << "Input tidak valid. Masukkan Jumlah Barang (angka): ";
71     }
72     cout << "Harga barang = ";
73     while (!(cin >> barang.harga)) {
74         cin.clear();
75         cin.ignore(numeric_limits<streamsize>::max(), '\n');
76         cout << "Input tidak valid. Masukkan Harga Barang (angka): ";
77     }
78     cout << "=====" << endl;
79
80     if (data.is_open()) {
81         data.seekp(0, ios::end);
82         data.write(reinterpret_cast<char*>(&barang), sizeof(Barang));
```

```

65     }
66     cout << "Jumlah barang = ";
67     while (!(cin >> barang.jumlah)) {
68         cin.clear();
69         cin.ignore(numeric_limits<streamsize>::max(), '\n');
70         cout << "Input tidak valid. Masukkan Jumlah Barang (angka): ";
71     }
72     cout << "Harga barang = ";
73     while (!(cin >> barang.harga)) {
74         cin.clear();
75         cin.ignore(numeric_limits<streamsize>::max(), '\n');
76         cout << "Input tidak valid. Masukkan Harga Barang (angka): ";
77     }
78     cout << "======" << endl;
79
80     if (data.is_open()) {
81         data.seekp(0, ios::end);
82         data.write(reinterpret_cast<char*>(&barang), sizeof(Barang));
83         data.close();
84         cout << "Barang berhasil ditambahkan :)" << endl;
85     } else {
86         cout << "Gagal membuka file untuk menambah data." << endl;
87     }
88 }

```

4. Fungsi Hapus Data

```
88     },
89
90     void HapusData(fstream &data, int id) {
91         fstream tempFile;
92         tempFile.open("temp.bin", ios::out | ios::binary);
93         if (!tempFile.is_open()) {
94             cout << "Gagal membuka file sementara untuk menghapus data." << endl;
95             return;
96         }
97
98         data.open("data.bin", ios::in | ios::binary);
99         if (!data.is_open()) {
100             cout << "Gagal membuka file data." << endl;
101             tempFile.close();
102             return;
103         }
104
105         Barang barang;
106         bool found = false;
107         while (data.read(reinterpret_cast<char*>(&barang), sizeof(Barang))) {
108             if (barang.id != id) {
109                 tempFile.write(reinterpret_cast<char*>(&barang), sizeof(Barang));
110             } else {
111                 found = true;
112             }
113         }
114
115         data.close();
116         tempFile.close();
117         remove("data.bin");
118         rename("temp.bin", "data.bin");
119
120         if (found) {
121             cout << "Barang dengan ID " << id << " berhasil dihapus." << endl;
122         } else {
123             cout << "Barang dengan ID " << id << " tidak ditemukan." << endl;
124         }
125     }
126
127     void CariData(fstream &data, int id) {
128         data.open("data.bin", ios::in | ios::binary);
```

5. Fungsi Cari Data

```
126  
127 void CariData(fstream &data, int id) {  
128     data.open("data.bin", ios::in | ios::binary);  
129     if (!data.is_open()) {  
130         cout << "Gagal membuka file data." << endl;  
131         return;  
132     }  
133  
134     Barang barang;  
135     bool found = false;  
136     while (data.read(reinterpret_cast<char*>(&barang), sizeof(Barang))) {  
137         if (barang.id == id) {  
138             cout << "Data Barang ditemukan:" << endl;  
139             cout << "Nama: " << barang.nama << endl;  
140             cout << "ID: " << barang.id << endl;  
141             cout << "Jumlah: " << barang.jumlah << endl;  
142             cout << "Harga: Rp " << fixed << setprecision(2) << barang.harga << endl;  
143             found = true;  
144             break;  
145         }  
146     }  
147  
148     if (!found) {  
149         cout << "Barang dengan ID " << id << " tidak ditemukan." << endl;  
150     }  
151  
152     data.close();  
153 }  
154
```

6. Fungsi Edit Data

```
155 void EditData(fstream &data, int id) {
156     fstream tempFile;
157     tempFile.open("temp.bin", ios::out | ios::binary);
158     if (!tempFile.is_open()) {
159         cout << "Gagal membuka file sementara untuk mengedit data." << endl;
160         return;
161     }
162
163     data.open("data.bin", ios::in | ios::binary);
164     if (!data.is_open()) {
165         cout << "Gagal membuka file data." << endl;
166         tempFile.close();
167         return;
168     }
169
170     Barang barang;
171     bool found = false;
172     while (data.read(reinterpret_cast<char*>(&barang), sizeof(Barang))) {
173         if (barang.id == id) {
174             cout << "Edit data Barang:" << endl;
175             cout << "Nama Barang = ";
176             cin.ignore(numeric_limits<streamsize>::max(), '\n');
177             cin.getline(barang.nama, 50);
178             cout << "Jumlah barang = ";
179             while (!(cin >> barang.jumlah)) {
180                 cin.clear();
181                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
182                 cout << "Input tidak valid. Masukkan Jumlah Barang (angka): ";
183             }
184             cout << "Harga barang = ";
185             while (!(cin >> barang.harga)) {
186                 cin.clear();
187                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
188                 cout << "Input tidak valid. Masukkan Harga Barang (angka): ";
189             }
190             found = true;
191         }
192         tempFile.write(reinterpret_cast<char*>(&barang), sizeof(Barang));
193     }
194 }
```

```
193     }
194
195     data.close();
196     tempFile.close();
197     remove("data.bin");
198     rename("temp.bin", "data.bin");
199
200     if (found) {
201         cout << "Barang dengan ID " << id << " berhasil diperbarui." << endl;
202     } else {
203         cout << "Barang dengan ID " << id << " tidak ditemukan." << endl;
204     }
205 }
206 }
```

7. Fungsi Tampil Data

```
204     }
205 }
206
207 void TampilData(fstream &data) {
208     data.open("data.bin", ios::in | ios::binary);
209     if (!data.is_open()) {
210         cout << "Gagal membuka file data." << endl;
211         return;
212     }
213
214     Barang barang;
215     int index = 1;
216     cout << "===== " << endl;
217     cout << "Daftar Barang:" << endl;
218     cout << "===== " << endl;
219     cout << setw(5) << "No" << setw(20) << "Nama" << setw(10) << "ID" << setw(15) << "Harga" << endl;
220     cout << "===== " << endl;
221     while (data.read(reinterpret_cast<char*>(&barang), sizeof(Barang))) {
222         cout << setw(5) << index++
223             << setw(20) << barang.nama
224             << setw(10) << barang.id
225             << setw(15) << fixed << setprecision(2) << barang.harga << endl;
226     }
227     cout << "===== " << endl;
228
229     data.close();
230 }
```

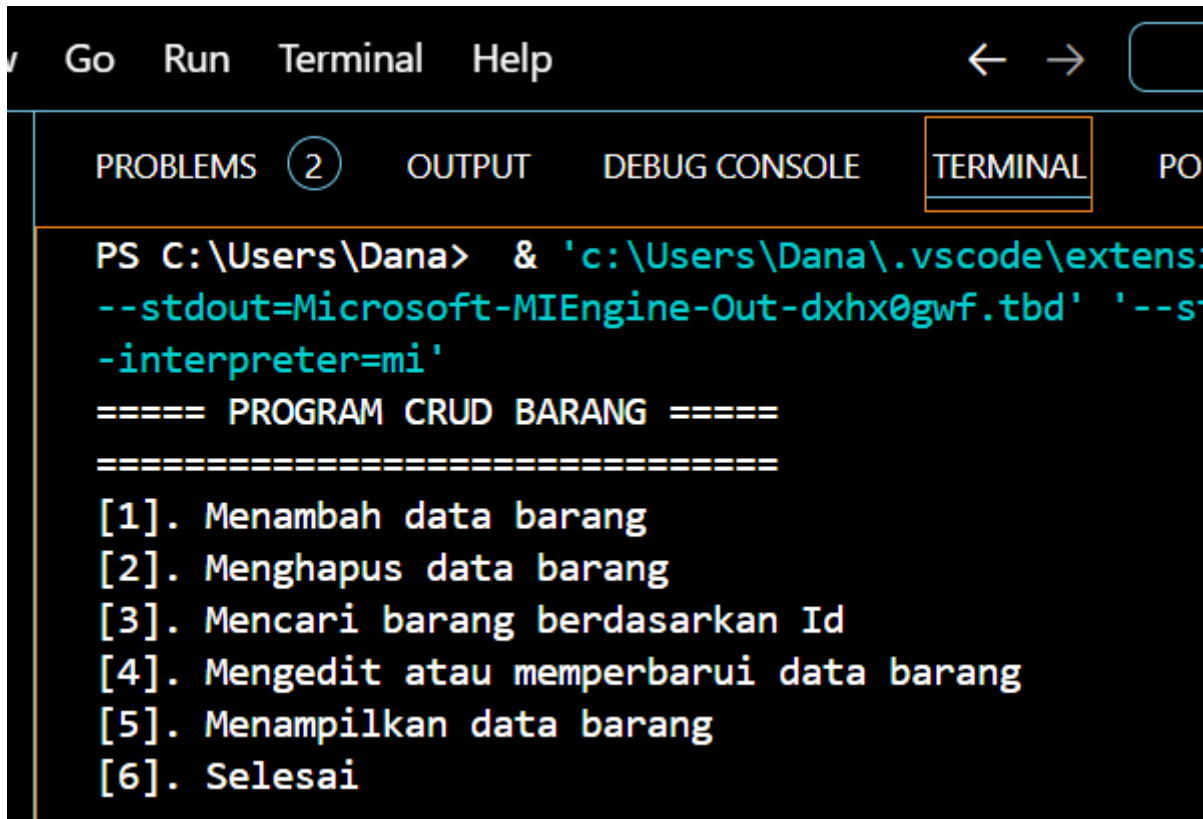
8. Fungsi Main

```
231
232 int main() {
233     fstream data;
234     char lanjut;
235     int pilihan, id;
236
237     enum optin { menambah = 1, menghapus, mencari, edit, menampilkan, selesai };
238
239     do {
240         pilihan = menu();
241         cout << endl;
242
243         switch (pilihan) {
244             case menambah:
245                 cout << "Menambah data barang" << endl;
246                 MenambahData(data);
247                 break;
248             case menghapus:
249                 cout << "Menghapus data barang" << endl;
250                 cout << "Masukkan ID Barang yang ingin dihapus: ";
251                 cin >> id;
252                 HapusData(data, id);
253                 break;
254             case mencari:
255                 cout << "Mencari barang berdasarkan Id" << endl;
256                 cout << "Masukkan ID Barang yang ingin dicari: ";
257                 cin >> id;
258                 CariData(data, id);
259                 break;
260             case edit:
261                 cout << "Mengedit atau memperbarui data barang" << endl;
262                 cout << "Masukkan ID Barang yang ingin diedit: ";
263                 cin >> id;
264                 EditData(data, id);
265                 break;
266             case menampilkan:
267                 cout << "Menampilkan data barang" << endl;
268                 TampilData(data);
269                 break;
```

```
270         }
271     } while (pilihan != selesai);
272
273     cout << endl;
274
275     cout << "Lanjutkan [Y/N]: ";
276     cin >> lanjut;
277     cin.ignore(numeric_limits<streamsize>::max(), '\n');
278     while (lanjut == 'Y' || lanjut == 'y');
279
280     return 0;
281 }
```

4.6 Output

1. Menu utama



```
PS C:\Users\Dana> & 'c:\Users\Dana\.vscode\extensions\ms-vscode.cmake-tools\bin\cmake\bin\cmake.exe' --stdout=Microsoft-MIEngine-Out-dxhx0gwf.tbd' '--s
-interpreter=mi'

===== PROGRAM CRUD BARANG =====
=====
[1]. Menambah data barang
[2]. Menghapus data barang
[3]. Mencari barang berdasarkan Id
[4]. Mengedit atau memperbarui data barang
[5]. Menampilkan data barang
[6]. Selesai
```

Pengguna diminta untuk memasukkan nomor pilihan mereka.

2. Menambah Data Barang.

```
-interpreter=mi'
===== PROGRAM CRUD BARANG =====
=====
[1]. Menambah data barang
[2]. Menghapus data barang
[3]. Mencari barang berdasarkan Id
[4]. Mengedit atau memperbarui data barang
[5]. Menampilkan data barang
[6]. Selesai
=====
Masukkan nomor yang ingin dipilih = 1

Menambah data barang
Ukuran data = 0
=====
Nama Barang = HP
ID Barang = 001
Jumlah barang = 10
Harga barang = 1000000
=====
Barang berhasil ditambahkan :)

Lanjutkan [Y/N]: Y
===== PROGRAM CRUD BARANG =====
=====
[1]. Menambah data barang
[2]. Menghapus data barang
[3]. Mencari barang berdasarkan Id
[4]. Mengedit atau memperbarui data barang
[5]. Menampilkan data barang
[6]. Selesai
=====
Masukkan nomor yang ingin dipilih = 1

Menambah data barang
Ukuran data = 1
=====
Nama Barang = LAPTOP
ID Barang = 002
Jumlah barang = 20
Harga barang = 15000000
=====
Barang berhasil ditambahkan :)

Lanjutkan [Y/N]: Y
===== PROGRAM CRUD BARANG =====
=====
```

Jika pengguna memilih opsi 1, mereka akan diminta untuk memasukkan data barang, termasuk nama, ID, jumlah, dan harga barang. Data ini kemudian disimpan ke dalam file

3. Edit Data Barang

```
Lanjutkan [Y/N]: Y
===== PROGRAM CRUD BARANG =====
=====
[1]. Menambah data barang
[2]. Menghapus data barang
[3]. Mencari barang berdasarkan Id
[4]. Mengedit atau memperbarui data barang
[5]. Menampilkan data barang
[6]. Selesai
=====
Masukkan nomor yang ingin dipilih = 4

Mengedit atau memperbarui data barang
Masukkan ID Barang yang ingin diedit: 001
Edit data Barang:
Nama Barang = MOTHERBOARD
Jumlah barang = 5
Harga barang = 8000000
Barang dengan ID 1 berhasil diperbarui.
```

Jika pengguna memilih opsi 4, mereka akan diminta untuk memasukkan ID barang yang ingin diedit. Program kemudian menampilkan data barang tersebut dan memungkinkan pengguna untuk memperbarui informasinya.

4. Mencari berdasarkan ID

```
===== PROGRAM CRUD BARANG =====  
=====  
[1]. Menambah data barang  
[2]. Menghapus data barang  
[3]. Mencari barang berdasarkan Id  
[4]. Mengedit atau memperbarui data barang  
[5]. Menampilkan data barang  
[6]. Selesai  
=====  
Masukkan nomor yang ingin dipilih = 3  
  
Mencari barang berdasarkan Id  
Masukkan ID Barang yang ingin dicari: 002  
Data Barang ditemukan:  
Nama: LAPTOP  
ID: 2  
Jumlah: 20  
Harga: Rp 15000000.00
```

Jika pengguna memilih opsi 3, mereka akan diminta untuk memasukkan ID barang yang ingin dicari. Program akan menampilkan detail barang jika ditemukan.

5. Menampilkan Data Barang

```
Lanjutkan [Y/N]: Y
===== PROGRAM CRUD BARANG =====
=====
[1]. Menambah data barang
[2]. Menghapus data barang
[3]. Mencari barang berdasarkan Id
[4]. Mengedit atau memperbarui data barang
[5]. Menampilkan data barang
[6]. Selesai
=====
Masukkan nomor yang ingin dipilih = 5

Menampilkan data barang
=====
Daftar Barang:
=====
      No           Nama           ID           Harga
=====
      1      MOTHERBOARD           1      8000000.00
      2           LAPTOP           2     15000000.00
=====
```

Jika pengguna memilih opsi 5, program akan menampilkan semua barang yang tersimpan di file data.bin.

6. Menghapus Data barang

```
Lanjutkan [Y/N]: Y
===== PROGRAM CRUD BARANG =====
=====
[1]. Menambah data barang
[2]. Menghapus data barang
[3]. Mencari barang berdasarkan Id
[4]. Mengedit atau memperbarui data barang
[5]. Menampilkan data barang
[6]. Selesai
=====
Masukkan nomor yang ingin dipilih = 2

Menghapus data barang
Masukkan ID Barang yang ingin dihapus: 001
Barang dengan ID 1 berhasil dihapus.
```

Jika pengguna memilih opsi 2, mereka akan diminta untuk memasukkan ID barang yang ingin dihapus. Program kemudian mencari barang tersebut dan menghapusnya jika ditemukan.

7. Selesai:

Jika pengguna memilih opsi 6, program akan selesai dan menampilkan pesan:

Selesai

Setelah setiap operasi, program akan menanyakan apakah pengguna ingin melanjutkan menggunakan program (Lanjutkan [Y/N]:). Jika pengguna memilih 'Y', program akan kembali ke menu utama. Jika pengguna memilih 'N', program akan berhenti.

BAB V

PENUTUP

5.1 Kesimpulan

Program di atas adalah aplikasi CRUD (Create, Read, Update, Delete) sederhana untuk mengelola data barang dalam sebuah file biner. Program ini menampilkan menu interaktif yang memungkinkan pengguna untuk melakukan beberapa operasi dasar pada data barang. Berikut adalah fitur-fitur utama yang disediakan oleh program ini:

1. Menu Utama : Saat menjalankan program, pengguna akan disambut dengan menu utama yang menawarkan beberapa pilihan: menambah data barang, menghapus data barang, mencari barang berdasarkan ID, mengedit atau memperbarui data barang, menampilkan semua data barang, dan keluar dari program.
2. Menambah Data Barang : Fungsi `MenambahData` memungkinkan pengguna untuk menambah informasi barang baru ke dalam file. Pengguna akan diminta untuk memasukkan nama barang, ID barang, jumlah barang, dan harga barang. Data yang dimasukkan akan disimpan dalam file biner.
3. Menghapus Data Barang : Fungsi `HapusData` digunakan untuk menghapus data barang dari file berdasarkan ID yang dimasukkan oleh pengguna. Proses penghapusan dilakukan dengan menyalin semua data barang kecuali barang yang akan dihapus ke file sementara, lalu menggantikan file asli dengan file sementara.
4. Mencari Barang Berdasarkan ID : Fungsi `CariData` memungkinkan pengguna untuk mencari dan menampilkan informasi barang berdasarkan ID. Jika barang dengan ID yang dicari ditemukan, informasi lengkap tentang barang tersebut akan ditampilkan.
5. Mengedit atau Memperbarui Data Barang : Fungsi `EditData` memungkinkan pengguna untuk mengedit informasi barang yang sudah ada berdasarkan ID. Sama seperti fungsi hapus, data barang yang diperbarui disalin ke file sementara, lalu file asli digantikan dengan file sementara.
6. Menampilkan Semua Data Barang : Fungsi `TampilData` akan menampilkan daftar lengkap semua barang yang tersimpan dalam file biner. Informasi yang ditampilkan meliputi nomor urut, nama barang, ID barang, dan harga barang.

7. Kontrol Program : Program ini menggunakan loop do-while yang memastikan menu utama terus ditampilkan sampai pengguna memilih untuk keluar. Setelah setiap operasi, pengguna ditanya apakah ingin melanjutkan atau tidak.

Dengan fitur-fitur tersebut, program ini memberikan cara yang mudah dan efisien bagi pengguna untuk mengelola data barang melalui antarmuka terminal berbasis teks.

5.2 Saran

Saran untuk Pengguna:

1. Validasi Input : Pastikan untuk selalu memasukkan data yang benar dan valid sesuai dengan prompt yang diberikan oleh program. Misalnya, ID barang harus berupa angka, begitu juga dengan jumlah dan harga barang.
2. Backup Data : Sebelum melakukan operasi yang bisa mengubah atau menghapus data, seperti mengedit atau menghapus barang, sebaiknya buat salinan cadangan (backup) dari file data binernya untuk mencegah kehilangan data.
3. Mengikuti Petunjuk : Bacalah instruksi dan pesan yang ditampilkan program dengan saksama. Program ini memberikan panduan yang jelas tentang apa yang harus dilakukan, termasuk mengingatkan pengguna untuk memasukkan input yang valid.
4. Penggunaan Program: Gunakan program ini untuk mengelola data barang dengan lebih teratur dan rapi. Ini sangat bermanfaat untuk inventaris sederhana atau aplikasi kecil lainnya.

Saran untuk Mahasiswa:

1. Memahami Struktural Data : Pelajari bagaimana struktur data, seperti struct dalam C++, digunakan untuk menyimpan informasi yang terorganisir. Struktur Barang adalah contoh yang baik tentang bagaimana mengelompokkan data yang berhubungan.
2. Manajemen File Biner : Pelajari cara membaca dan menulis ke file biner. Ini penting karena file biner lebih efisien dalam penyimpanan dan pemrosesan data dibandingkan dengan file teks.
3. Error Handling : Perhatikan bagaimana program menangani kesalahan saat membuka file. Menangani kesalahan dengan baik adalah aspek penting dalam pemrograman untuk membuat aplikasi yang lebih robust.
4. Loop dan Kondisi : Perhatikan penggunaan loop do-while dan switch-case untuk mengelola menu dan operasi program. Ini adalah teknik dasar tetapi sangat efektif untuk banyak aplikasi interaktif.
5. Praktik Pengkodean : Perhatikan penggunaan fungsi-fungsi modular seperti MenambahData, HapusData, CariData, EditData, dan TampilData. Memisahkan logika program ke dalam fungsi-fungsi kecil yang bisa digunakan kembali membantu dalam menjaga kode tetap bersih dan terorganisir.