

**PROJEK MAKALAH**  
**UJIAN TENGAH SEMESTER**  
**MATA KULIAH PRAKTIKUM STRUKTUR DATA**

**“PEMBUATAN PROGRAM MANAJEMEN DATA BARANG ”**



**DISUSUN OLEH :**  
**ZALDY SENO YUDHANTO**  
-2310631170123  
2C-INFORMATIKA

**INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS SINGAPERBANGSA KARAWANG**  
**2024**

## **BAB 1**

### **LATAR BELAKANG**

Program pendataan barang ini dibuat untuk memberikan kemudahan dalam manajemen data barang pada sebuah toko. Dengan program ini, pengguna dapat dengan cepat dan efisien melakukan berbagai operasi seperti membuat, membaca, memperbarui, dan menghapus data barang, serta melihat stok barang yang tersedia. Program ini menggunakan struktur data linked list untuk menyimpan data barang, di mana setiap barang memiliki atribut nama, harga, kode, dan stok. Dengan demikian, program ini dapat menjadi solusi yang praktis dalam mempermudah kegiatan manajemen inventarisasi barang di toko.

Dalam penggunaannya, program ini memberikan fleksibilitas dan kenyamanan bagi pengguna untuk mengelola informasi barang dengan lebih terstruktur. Dengan adanya fitur-fitur yang disediakan, seperti pencatatan data barang secara detail dan pembaruan stok secara langsung, diharapkan dapat meningkatkan efisiensi dalam pengelolaan toko. Selain itu, pengguna juga dapat dengan mudah memantau perubahan dalam persediaan barang dan mengambil keputusan yang tepat dalam hal pengadaan atau pengelolaan stok. Dengan demikian, program ini diharapkan dapat menjadi alat yang berguna bagi pemilik atau pengelola toko dalam mengoptimalkan manajemen inventarisasi barang.

Disini saya menggunakan linkedlist sebagai penyimpanan data, serta structure agar membuat data lebih rapih.

## BAB II

## PEMBAHASAN

### 2.1. RESOURCE PROJECT

```
=====
Selamat datang di program pendataan barang!

Menu Utama
=====
[1]Create      [2]Read      [3]Update      [4]Delete      [5]Exit
=====

Masukan Pilihan :
```

Tampilan awal.

```
Masukan jumlah barang yang ingin di data : 2
=====
Masukan nama barang: Beras
Masukan harga barang: 10000
Masukan kode barang: 1
Masukan stok barang: 10
```

```
Data berhasil dibuat.

=====

Ketik [<] untuk kembali ke menu
Pilihan :
```

Tampilan menu 1 setelah kita menginputkan barang.

```
Data Barang:

Data ke-1:
Nama Barang: Beras
Harga Barang: Rp.10000
Kode Barang: 1
Stok Barang: 10
Alamat Barang: 0x671760

Data ke-2:
Nama Barang: Pensil
Harga Barang: Rp.3000
Kode Barang: 2
Stok Barang: 5
Alamat Barang: 0x675ac0

=====

Ketik [<] untuk kembali ke menu
Pilihan :
```

Tampilan menu 2.

```
Data Barang:

Data ke-1:
Nama Barang: Beras
Harga Barang: Rp.10000
Kode Barang: 1
Stok Barang: 10
Alamat Barang: 0x671760

Data ke-2:
Nama Barang: Pensil
Harga Barang: Rp.3000
Kode Barang: 2
Stok Barang: 5
Alamat Barang: 0x675ac0

Masukkan kode barang yang ingin diupdate: 2
Data barang ditemukan.
Masukkan nama barang yang baru: Pulpen
Masukkan harga barang yang baru: 3500
Masukkan stok barang yang baru: 10

=====

Ketik [<] untuk kembali ke menu
Pilihan :
```

Tampilan menu 3

```
Data Barang:

Data ke-1:
Nama Barang: Beras
Harga Barang: Rp.10000
Kode Barang: 1
Stok Barang: 10
Alamat Barang: 0x671760

Data ke-2:
Nama Barang: Pulpen
Harga Barang: Rp.3500
Kode Barang: 2
Stok Barang: 10
Alamat Barang: 0x675ac0

Masukkan kode barang yang ingin dihapus: 1
Data barang ditemukan dan dihapus.

=====

Ketik [<] untuk kembali ke menu
Pilihan :
```

Tampilan menu 4.

```
Masukan jumlah barang yang ingin di data : 1
```

```
=====
Masukan nama barang: Emas
Masukan harga barang: 1000000
Masukan kode barang: 1
Masukan stok barang: 1
```

Saya menambahkan satu barang lagi.

```
Data Barang:
```

```
Data ke-1:
Nama Barang: Pulpen
Harga Barang: Rp.3500
Kode Barang: 2
Stok Barang: 10
Alamat Barang: 0x675ac0
```

```
Data ke-2:
Nama Barang: Emas
Harga Barang: Rp.1000000
Kode Barang: 1
Stok Barang: 1
Alamat Barang: 0x671760
```

```
=====
Ketik [<] untuk kembali ke menu
Pilihan :
```

Tampilan setelah saya update, hapus, dan menambahkan barang.

```
=====
Selamat datang di program pendataan barang!
```

```
Menu Utama
```

```
=====
[1]Create      [2]Read      [3]Update      [4]Delete      [5]Exit
=====
```

```
Masukan Pilihan :5
Program dibuat oleh Zaldy Seno Yudhanto (2310631170123)
```

Tampilan menu 5.

## 2.2. SCREENSHOT CODE DAN PENJELASAN

```
1  #include <iostream>
2  #include <windows.h>
3  #include <iomanip>
4  #include <string>
5
6  using namespace std;
7
8  const int MAX_BARANG = 100; // Maksimum jumlah barang yang dapat disimpan dalam array
9
10 void kembali();
11
12 struct barang {
13     string nama;
14     int harga, kode;
15     int stok; // Menambahkan variabel stok
16
17     barang* next;
18 };
19
20 barang* input = NULL;
21
22 //membuat fungsi hiasan garis
23 void garis () {
24     for (int i = 0; i < 72; i++){
25         cout << "=";
26     }
27     cout << endl;
28 }
```

Saya menggunakan 4 library, yang pertama ada *iostream* library standar Bahasa pemrograman c++, kemudian *windows.h* untuk mengakses berbagai fitur sistem operasi Windows. Beberapa fungsi yang tersedia antara lain untuk mengatur tampilan layar, mengakses fungsi sistem, dan berinteraksi dengan file system. Lalu ada *iomanip* digunakan pada fungsi menu utama yang saya buat. Kemudian ada *string* digunakan untuk menyediakan berbagai fungsi untuk bekerja dengan string dalam C++. Fungsi-fungsi untuk mengolah string seperti menggabungkan string, mencari panjang string, dan melakukan manipulasi string lainnya tersedia di dalamnya.

Kemudian saya mendeklarasikan struct yang juga digunakan untuk membuat node barang yang digunakan untuk menyimpan data barang yang akan saya inputkan kemudian saya mendeklarasikan *input = NULL* agar setiap null yang saya isikan bisa disambung sambungkan.

Lalu saya juga membuat fungsi untuk hiasan garis agar mempercantik output yang dihasilkan.

```

30 void data(){
31     if (input == NULL) {
32         cout << "Belum ada data barang yang dimasukkan.\n";
33     } else {
34         cout << "Data Barang:\n\n";
35         barang* temp = input;
36         int counter = 1;
37         while (temp != NULL) {
38             cout << "Data ke-" << counter++ << ":\n";
39             cout << "Nama Barang: " << temp->nama << endl;
40             cout << "Harga Barang: Rp." << temp->harga << endl;
41             cout << "Kode Barang: " << temp->kode << endl;
42             cout << "Stok Barang: " << temp->stok << endl;
43             cout << "Alamat Barang: " << temp << endl;
44             cout << endl;
45             cout << endl;
46             temp = temp->next;
47         }
48     }
49 }

```

Fungsi *data* saya gunakan untuk output bacaan dari data yang sudah saya inputkan agar bisa kembali dipanggil di fungsi yang lain seperti *update* dan *delete* agar sebelum update dan delete bisa tahu kode barang apa yang harus diinputkan sebelum melakukan aksi.

Disitu saya menggunakan pengkondisian dimana jika **input == NULL**, atau input masih kosong belum ada data yang saya isikan maka akan keluar output *belum ada data barang yang dimasukan*. Jika selain itu maka data yang diinputkan akan dipanggil.

Dimana saya deklarasikan di awal **temp = input** dan akan melakukan pengulangan dimana jika temp masih ada isinya maka akan terus looping, looping akan berhenti jika node **temp-> next** sudah **NULL**. Disitu saya memanggil data yang sudah saya simpan di structure.

```

51 void create (){
52     int jumlah;
53     do {
54         cout << "Masukan jumlah barang yang ingin di data : ";
55         cin >> jumlah;
56         garis();
57     }
58     for(int i = 0; i < jumlah; i++){
59         barang* newNode = new barang;
60         newNode->next = NULL;
61         cin.ignore(); // Membersihkan newline karakter dari buffer
62         cout << "Masukan nama barang: ";
63         getline(cin, newNode->nama); // Membaca seluruh baris sebagai input nama barang
64         cout << "Masukan harga barang: ";
65         cin >> newNode->harga;
66         cout << "Masukan kode barang: ";
67         cin >> newNode->kode;
68         cout << "Masukan stok barang: ";
69         cin >> newNode->stok; // Membaca stok barang
70     }

```

```

71     if(input == NULL){
72         input = newNode;
73     }
74     else{
75         barang* temp = input;
76         while(temp->next != NULL){
77             temp = temp->next;
78         }
79         temp->next = newNode;
80     }
81     system("cls");
82 }
83 cout << "Data berhasil dibuat.\n";
84 kembali();
85 } while (true);
86 }

```

Fungsi *create* digunakan untuk menginputkan data yang nanti saya panggil di fungsi *data*. Yang pertama saya menginput berapa data barang yang mau diinputkan, kemudian melakukan perulangan, dimana perulangan akan berhenti jika nilai *i* yang sudah saya deklarasikan **0**, kurang dari jumlah yang tadi diinputkan.

Pertama saya deklarasikan **newNode** sebagai node baru, lalu mendeklarasikan next nodenya sebagai **NULL**. Agar bisa berhenti jika tidak ada data baru yang diinputkan saat dipanggil. Kemudian saya menginputkan data kedalam **struct barang** yang sudah saya buat, kemudian saya menyambungkan data yang sudah saya buat di pengkondisian.

Dimana jika **input == NULL**, kita jadikan input sebagai node baru, jika **input** sudah ada isinya maka kita jadikan **temp = input** lalu melakukan looping selama **next** belum **NULL** maka kita jadikan **temp->next** dijadikan node baru. Jika **next** sudah **NULL** maka looping berhenti. Kemudian memanggil fungsi **kembali** yang digunakan untuk kembali ke menu utama.

```

88 void read() {
89     data();
90     kembali();
91 }

```

Disini ada fungsi *read* saya memanggil fungsi *data* dan fungsi *kembali*. Untuk fungsi *data* sudah dijelaskan diatas, dimana fungsi *data* digunakan untuk memanggil data yang sudah diinputkan. Dan fungsi *kembali* digunakan untuk kembali ke menu utama yang belum saya jelaskan.



```

93 void update() {
94     data();
95     int kode;
96     cout << "Masukkan kode barang yang ingin diupdate: ";
97     cin >> kode;
98
99     barang* temp = input;
100     bool found = false;
101     while (temp != NULL) {
102         if (temp->kode == kode) {
103             cout << "Data barang ditemukan.\n";
104             cout << "Masukkan nama barang yang baru: ";
105             cin.ignore(); // Mengabaikan newline karakter dari buffer
106             getline(cin, temp->nama); // Membaca seluruh baris sebagai input nama barang
107             cout << "Masukkan harga barang yang baru: ";
108             cin >> temp->harga;
109             cout << "Masukkan stok barang yang baru: ";
110             cin >> temp->stok; // Memasukkan stok barang
111             found = true;
112             break;
113         }
114         temp = temp->next;
115     }
116
117     if (!found) {
118         cout << "Data barang dengan kode tersebut tidak ditemukan.\n";
119     }
120
121     kembali();
122 }

```

Fungsi *update* digunakan untuk mengubah data yang sudah kita inputkan tadi di awal. Kita mengubah datanya dengan cara mencari kode barang yang sudah kita inputkan tadi lalu menggunakan pengkondisian dimana di awal kita deklarasikan **bool found = false** lalu kita melakukan perulangan, perulangan akan berhenti jika **temp NULL**. Lalu kita melakukan pengkondisian dimana jika kita mencari kode dan tidak ada di inputan yang sudah kita input tadi maka akan ada outputan *data barang dengan kode tersebut tidak ditemukan*. Jika kode yang kita cari ada maka kita akan menginputkan nama, harga, dan stok baru lalu kita jadikan **found = true**, agar pengulangan berhenti setelah kita menemukan kodenya.

Kemudian saya memanggil fungsi **kembali** untuk kembali ke menu utama.

```

124 void del() {
125     data();
126     int kode;
127     cout << "Masukkan kode barang yang ingin dihapus: ";
128     cin >> kode;
129
130     barang* prev = NULL;
131     barang* temp = input;
132     bool found = false;
133
134     while (temp != NULL) {
135         if (temp->kode == kode) {
136             cout << "Data barang ditemukan dan dihapus.\n";
137             if (prev == NULL) {
138                 // Jika barang yang ingin dihapus berada di awal linked list
139                 input = input->next;
140             } else {
141                 // Jika barang yang ingin dihapus berada di tengah atau akhir linked list
142                 prev->next = temp->next;
143             }
144             delete temp;
145             found = true;
146             break;
147         }
148         prev = temp;
149         temp = temp->next;
150     }
151
152     if (!found) {
153         cout << "Data barang dengan kode tersebut tidak ditemukan.\n";
154     }
155
156     kembali();
157 }

```

Disini ada fungsi *del* yang berfungsi untuk menghapus data yang kita inginkan. Diawal kita memanggil fungsi *data* untuk mengetahui nomer kode barang yang kita punya. Kita deklarasikan **prev = NULL**, **temp = input** dan **found = false**. Lalu kita menginput kode barang yang kita cari kemudian disana ada perulangan Dimana perulangan akan terus berlangsung jika **temp != NULL** dan ada pengkondisian seperti di menu update, dimana jika barang yang kita tidak ketemu maka akan mengeluarkan output *data barang dengan kode tersebut tidak ditemukan*. Jika kode barang ditemukan maka akan menghapus kode yang kita cari tadi.

Dimana jika **prev == NULL**. Maka input akan di next sampai nextnya NULL kemudian **temp** akan dihapus. Kemudian jika sudah dihapus maka akan dipanggil fungsi *kembali*.

```

159 void menuUtama() {
160     int pil;
161     garis();
162     cout << "Selamat datang di program pendataan barang!\n\n";
163     cout << "Menu Utama\n";
164     garis();
165     cout << setiosflags(ios::left) << setw(15) << "[1]Create";
166     cout << setiosflags(ios::left) << setw(15) << "[2]Read";
167     cout << setiosflags(ios::left) << setw(15) << "[3]Update";
168     cout << setiosflags(ios::left) << setw(15) << "[4]Delete";
169     cout << setiosflags(ios::left) << setw(15) << "[5]Exit";
170     cout << endl;
171     garis();
172
173     do {
174         cout << "\nMasukan Pilihan :";
175         cin >> pil;
176
177         switch (pil){
178             case 1 :
179                 system ("cls");
180                 create ();
181                 break;
182             case 2 :
183                 system ("cls");
184                 read ();
185                 break;
186             case 3 :
187                 system ("cls");
188                 update ();
189                 break;
190             case 4 :
191                 system ("cls");
192                 del ();
193                 break;
194             case 5 :
195                 return;
196                 break;
197             default:
198                 cout << "Pilihan tidak valid. Silahkan masukan pilihan antara 1 hingga 5\n";
199             }
200         } while (true);
201     }

```

Lalu ini adalah fungsi *menuUtama* yang digunakan untuk memanggil fungsi fungsi menu yang sudah kita buat tadi. Saya memanfaatkan switch case untuk mengaksesnya. Dimana pilihan satu akan ke menu *create*, lalu pilihan kedua akan ke menu *read*, pilihan ketiga akan ke menu *update*, meny keempat akan ke meny *del*, dan pilihan 5 akan keluar dari program. Lalu saya memanfaatkan perulangan *do while* agar jika inputan tidak sesuai akan terus ulang.

```

203 void kembali () {
204     char menu;
205
206     cout << endl;
207     garis();
208     cout << "\nKetik [<] untuk kembali ke menu\n";
209     do {
210         cout << "Pilihan : ";
211         cin >> menu;
212
213         switch (menu) {
214             case '<' :
215                 system("cls");
216                 menuUtama();
217                 break;
218             default :
219                 cout << "Pilihan tidak valid. Silahkan masukan pilihan [<] untuk kembali\n";
220         }
221     } while (menu != '<');
222
223

```

Kemudian ini adalah fungsi *kembali* Dimana user harus menginput < agar kembali ke menu utama dan melanjutkan programnya.

```

226 int main() {
227     menuUtama();
228     cout << "Program dibuat oleh Zaldy Seno Yudhanto (2310631170123)\n";
229     return 0;
230 }
231

```

Ini adalah fungsi utamanya, disitu saya hanya memanggil fungsi *menuUtama* saja kemudian setelah keluar program akan keluar output *program dibuat oleh Zaldy Seno Yudhanto (2310631170123)*.

### **BAB III**

#### **KESIMPULAN**

program ini memberikan solusi efektif dalam mengelola data barang, mulai dari pembuatan data, pembacaan, pembaruan, hingga penghapusan. Penggunaan struktur data linked list sebagai basis penyimpanan data memungkinkan program untuk mengelola jumlah barang yang cukup besar dengan efisien.

Selanjutnya, program ini memberikan fleksibilitas bagi pengguna untuk mengatur stok barang dengan memperbarui data yang ada. Dengan adanya fitur ini, pemilik bisnis dapat dengan mudah mengelola persediaan barang mereka. Selain itu, antarmuka pengguna yang sederhana membuat program mudah digunakan oleh berbagai kalangan, meskipun mungkin masih memerlukan peningkatan untuk aspek keseluruhan pengalaman pengguna.

Dalam pengembangan selanjutnya, program ini dapat ditingkatkan dengan penambahan fitur tambahan seperti validasi input yang lebih ketat, manajemen pengguna dengan login, atau bahkan integrasi dengan sistem database untuk penyimpanan data yang lebih kuat. Dengan demikian, program ini tidak hanya dapat digunakan sebagai alat manajemen persediaan barang yang efektif, tetapi juga sebagai platform yang dapat terus berkembang sesuai dengan kebutuhan bisnis.

## **LAMPIRAN**

*Berisi link repository github*