LAPORAN PRAKTIKUM

PEMROGRAMAN DASAR

Operasi Filedan Vector

NAMA: FALIK FABIO AULIA NIM: 25104410030

PERIODE: SEMESTER GANJIL 2024/2025

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ISLAM BALITAR

LAPORAN PROGRAM C++
MEMBACAFILE TXT KEVECTORMENGGUNAKAN PEMROGRAMAN C++

I. PENDAHULUAN

1.1LatarBelakang

Program ini dibuat untuk mendemonstrasikan cara membaca data dari file teks eksternal dan menyimpannya ke dalam struktur data Vector menggunakan bahasa pemrograman C++. Program ini merupakan implementasi dari operasi file I/O (Input/Output) yang merupakan salah satu konsep penting dalam pemrograman.

1.2 Tujuan

- Membaca data dari file eksternal (bacaAku.txt)
- Memproses dan memisahkan data berdasarkan delimiter
- Menyimpan data ke dalam struktur data Vector
- Menampilkan data yang telah dibaca dengan format yang rapi

II. ANALISIS PROGRAM

2.1 Library yang Digunakan

```
1   #include<iostream>
2   #include<fstream>
3   #include<string>
4   #include<sstream>
5   #include<vector>
6   #include<iomanip> //untuk setw
```

Penjelasan:

- ***iostream***: Library standar untuk input/output, digunakan untuk cout
- ***fstream***: Library untuk file stream, digunakan untuk membaca file (ifstream)
  string: Library untuk tipe data string
  sstream: Library untuk string stream, digunakan untuk parsing/memisahkan data
- ***vector***: Library untuk container vector (array dinamis)

- ***iomanip***: Library untuk manipulasi format output seperti `setw()` dan `left`

2.2 Struktur Data

```
struct Siswa {
    int nomor;
    string nama;
    int nilai;
```

Penjelasan:

- Mendefinisikan tipe data baru bernama Siswa
- Memiliki 3 atribut: nomor (integer), nama (string), dan nilai (integer)
- Struktur ini digunakan untuk menyimpan informasi setiap siswa

III. ALUR PROGRAM

3.1 Deklarasi Vector

```
vector<Siswa> daftarSiswa;
```

Penjelasan:

- Membuat vector dengan nama daftarSiswa
- Vector ini bertipe Siswa, artinya dapat menyimpan banyak objek Siswa
- Vector adalah container dinamis yang dapat bertambah ukurannya secara otomatis

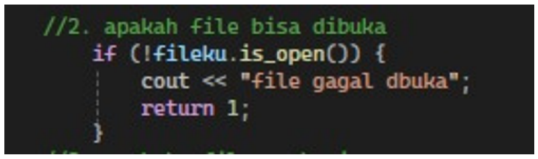
3.2 Membuka File

```
//1. membuka file
ifstream fileku("bacaAku.txt");
```

Penjelasan:

ifstream adalah input file stream (untuk membaca file)
Membuka file bernama "bacaAku.txt"
File harus berada di direktori yang sama dengan executable program

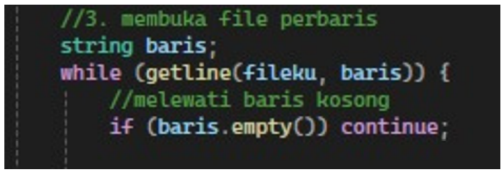
3.3 Validasi File



Penjelasan:

- Mengecek apakah file berhasil dibuka
- is_open() mengembalikan true jika file terbuka, false jika gagal
- Jika gagal, program menampilkan pesan error dan keluar dengan kode 1

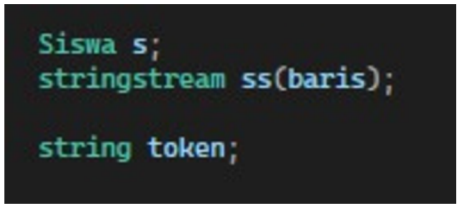
3.4 Membaca File Per Baris



Penjelasan:

- getline(fileku, baris) membaca satu baris dari file dan menyimpannya ke variabel `baris`
- Loop akan berjalan selama masih ada baris yang bisa dibaca
- `if (baris.empty()) continue; melewati baris kosong`

3.5 Parsing Data



Penjelasan:

- Membuat objek Siswa bernama s untuk menyimpan data sementara
  stringstream ss(baris) membuat stream dari string baris
  token digunakan untuk menyimpan potongan data sementara

a. Membaca Nomor

```
    //nampilkan nomor
    getline(ss, token, ';');
    s.nomor = stoi(token);
```

Penjelasan:

- getline(ss, token, ';') membaca dari stringstream sampai menemukan delimiter ;
- stoi(token) mengkonversi string menjadi integer
- Hasilnya disimpan ke s.nomor

• Contoh: "1; Tono; 82;" → token = "1" → s.nomor = 1

b. Membaca Nama

```
    //nampilkan nama
    getline(ss, s.nama, ';');
```

Penjelasan:

- Membaca langsung ke s.nama sampai menemukan ; Tidak perlu konversi karena sudah bertipe string

• Contoh: " Tono; 82;" → s.nama = " Tono" (dengan spasi)

c. Membaca Nilai

```
    //nampilkan nilai
    getline(ss, token, ';');
    s.nilai = stoi(token);
```

Penjelasan:

- Membaca nilai sampai menemukan ;
- Mengkonversi string ke integer dengan stoi()

• Contoh: " 82;" → token = " 82" → s.nilai = 82

3.6 Menyimpan ke Vector

```
    //4. simpan kevector
    daftarSiswa.push_back(s);
}
```

Penjelasan:

- push_back() menambahkan elemen baru di akhir vector
- Objek s yang berisi data siswa ditambahkan ke `da ftarSiswa`
- Vector akan otomatis bertambah ukurannya

3.7 Menutup File

```
fileku.close();

//5. tampilkan data
cout << "\nData siswa:\n";
for (const auto& student : daftarSiswa) {
    cout << left
        << setw(6) << student.nomor
        << setw(15) << student.nama
        << setw(8) << student.nilai << "\n";
    }
return 0;
```

Penjelasan:

- Menutup file setelah selesai dibaca
- Membebaskan resource yang digunakan
- Best practice dalam pemrograman ***3.8Menampilkan Data***

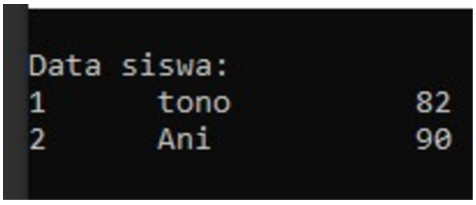
Penjelasan:
`for (const auto& student : daftarSiswa)` adalah range-based for loop
`const auto&` artinya referensi konstan ke elemen vector (efisien, tidak copy)
`left` membuat teks rata kiri
`setw(6)` mengatur lebar kolom menjadi 6 karakter
Output akan tampil dalam bentuk tabel yang rapi

IV. CONTOH INPUT DAN OUTPUT

4.1 Input File (bacaAku.txt)

```
1; tono; 82 ;
2; Ani; 90;
```

4.2 Output Program



V. KESIMPULAN

Program ini berhasil mendemonstrasikan cara membaca data dari file eksternal, melakukan parsing data dengan delimiter tertentu, menyimpan data ke dalam struktur data Vector, dan menampilkannya dengan format yang rapi. Program ini merupakan dasar yang baik untuk memahami konsep file I/O dan manipulasi data dalam C++.

Konsep-konsep penting yang dipelajari:

- Operasi file menggunakan ifstream Parsing string dengan stringstream
  Penggunaan struktur data vector
- Manipulasi output dengan iomanip
- Penggunaan struct untuk mengelompokkan data