LAPORAN PRAKTIKUM STRUKTUR DATA

MODUL VII (8)
QUEUE

Disusun Oleh :
NAMA : Gien Darrel Adli
NIM : 10312430008

Dosen
FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

A. Dasar Teori

Algoritma merupakan fondasi dalam pembuatan program komputer. Secara sederhana, algoritma adalah serangkaian langkah logis dan sistematis yang disusun untuk menyelesaikan suatu masalah. C++, berfungsi sebagai alat untuk mengimplementasikan algoritma tersebut agar dapat dimengerti dan dieksekusi oleh komputer. C++ sering digunakan sebagai bahasa pengantar untuk mempelajari konsep pemrograman dasar karena strukturnya yang terorganisir dan kemampuannya untuk menangani operasi tingkat rendah.

Untuk membangun logika dalam program sesuai dengan algoritma yang dirancang, C++ menyediakan struktur kontrol, yang terbagi menjadi dua jenis utama:

Struktur Percabangan (Conditional): Digunakan untuk pengambilan keputusan, di mana program akan menjalankan blok kode tertentu jika suatu kondisi terpenuhi. Struktur ini mencakup if-else untuk mengevaluasi kondisi boolean dan switch-case untuk memilih blok kode berdasarkan nilai dari sebuah variabel.

Struktur Perulangan (Looping): Digunakan untuk mengeksekusi blok kode yang sama secara berulang kali selama kondisi tertentu masih terpenuhi. C++ menyediakan tiga jenis perulangan utama: for, while, dan do-while, yang masing-masing memiliki karakteristik penggunaan yang spesifik dalam implementasi algoritma.

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1 (queue.h)

```
#ifndef QUEUE_H
#define QUEUE_H

#define MAX_QUEUE 5

struct Queue
{
   int info[MAX_QUEUE];
   int head;
   int tail;
   int count;
};

void createQueue(Queue &Q);

bool isEmpty(Queue Q);

bool isFull(Queue Q);
```

```
void enqueue(Queue &Q, int x);

int dequeue(Queue &Q);

void printInfo(Queue Q);

#endif
```

Deskripsi:

File queue.h adalah sebuah header file yang berisi deklarasi struktur dan fungsi untuk mengelola struktur data antrian (queue) dengan ukuran maksimum lima elemen. Di dalamnya didefinisikan sebuah struct bernama Queue yang menyimpan array info sebagai penampung data, serta variabel head, tail, dan count untuk menandai posisi awal, akhir, dan jumlah elemen antrian. File ini juga mendeklarasikan beberapa fungsi penting seperti createQueue untuk inisialisasi antrian, isEmpty dan isFull untuk mengecek kondisi antrian, enqueue untuk menambahkan data ke antrian, dequeue untuk mengeluarkan data dari antrian, serta printInfo untuk menampilkan isi antrian. Dengan kata lain, queue.h menyediakan blueprint dari seluruh operasi dasar yang diperlukan untuk menggunakan queue dalam program.

Guided 2 (queue.cpp)

```
#include "queue.h"
#include <iostream>

using namespace std;

void createQueue(Queue &Q)
{
    Q.head = 0;
    Q.tail = 0;
    Q.count = 0;
}

bool isEmpty(Queue Q)
{
    return Q.count == 0;
}

bool isFull(Queue Q)
{
    return Q.count == MAX_QUEUE;
}

void enqueue(Queue &Q, int x)
{
    if (!isFull(Q))
    {
```

```
        Q.info[Q.tail] = x;
        Q.tail = (Q.tail + 1) % MAX_QUEUE;
        Q.count++;
    }
    else
    {
        cout << "Antrean Penuh! " << endl;
    }
}

int dequeue(Queue &Q)
{
    if (!isEmpty(Q))
    {
        int x = Q.info[Q.head];
        Q.head = (Q.head + 1) % MAX_QUEUE;
        Q.count--;
        return x;
    }
    else
    {
        cout << "Antrean kosong! " << endl;
        return -1;
    }
}

void printInfo(Queue Q)
{
    cout << "isi queue: [";
    if (!isEmpty(Q))
    {
        int i = Q.head;
        int n = 0;
        while (n < Q.count)
        {
            cout << Q.info[i] << " ";
            i = (i + 1) % MAX_QUEUE;
            n++;
        }
    }
    cout << "]" << endl;
}
```

Deskripsi:

File queue.cpp adalah file implementasi dari queue.h. Di dalamnya berisi kode program yang menjalankan fungsi-fungsi queue seperti inisialisasi antrian, pengecekan kosong atau penuh, menambah data dengan enqueue, menghapus data dengan dequeue, serta menampilkan isi antrian melalui printInfo. File ini juga menerapkan konsep circular queue dengan memutar posisi head dan tail menggunakan operasi modulus.

Guided 3 (main.cpp)

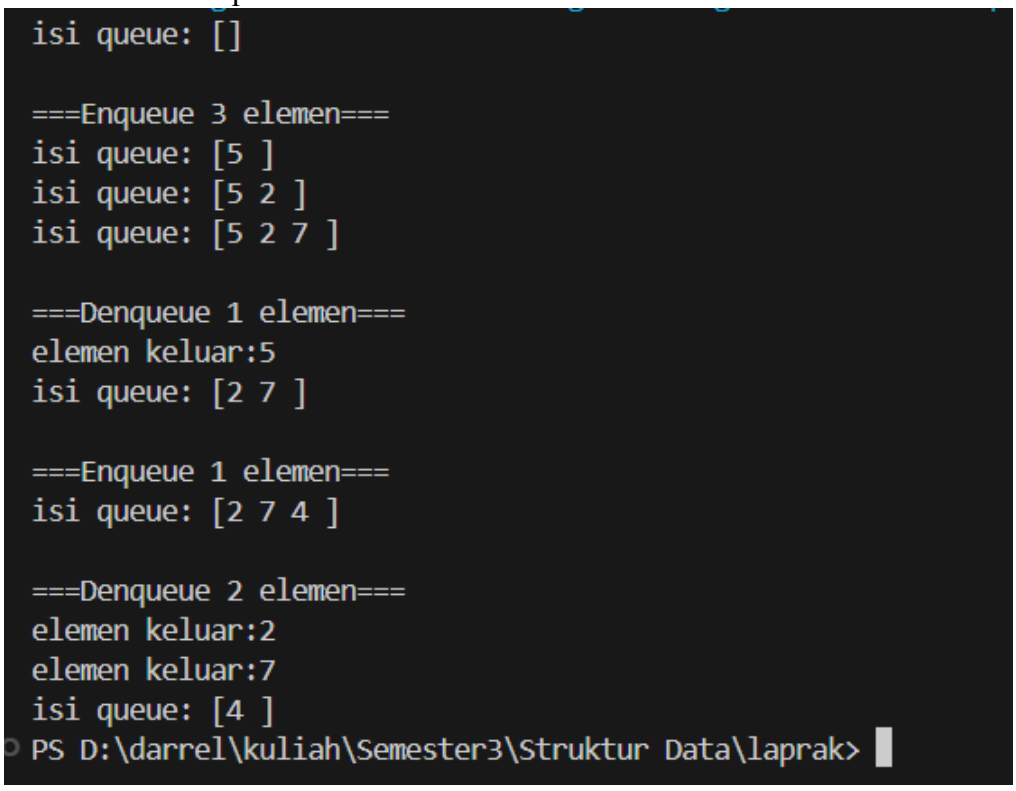
```
#include
<iostream>
#include
"queue.h"
#include
"queue.cpp"

using namespace
std;

int main()
{
   Queue Q;

   createQueue(Q)
;
   printInfo(Q);

   cout <<
"\n===Enqueue 3
elemen===" <<
endl;
   enqueue(Q, 5);
   printInfo(Q);
   enqueue(Q, 2);
   printInfo(Q);
   enqueue(Q, 7);
   printInfo(Q);

   cout <<
"\n===Denqueue
1 elemen===" <<
endl;

   cout <<
"elemen keluar:"
```

```
<< dequeue(Q) <<
endl;
   printInfo(Q);

   cout <<
"\n===Enqueue 1
elemen===" <<
endl;
   enqueue(Q, 4);
   printInfo(Q);

   cout <<
"\n===Denqueue
2 elemen===" <<
endl;

   cout <<
"elemen keluar:"
<< dequeue(Q) <<
endl;

   cout <<
"elemen keluar:"
<< dequeue(Q) <<
endl;
   printInfo(Q);

   return 0;
}
```

Deskripsi:

File main.cpp merupakan file utama yang berfungsi sebagai driver program untuk menguji seluruh operasi queue yang telah dibuat. Di dalamnya dibuat sebuah objek Queue Q, kemudian dilakukan inisialisasi menggunakan createQueue. Setelah itu program mencoba berbagai operasi seperti enqueue beberapa elemen untuk memasukkan data ke antrian, dequeue untuk mengeluarkan elemen dari antrian, serta memanggil printInfo setiap selesai operasi agar terlihat perubahan kondisi queue. Program ini juga menampilkan output berupa elemen yang keluar dari antrian serta kondisi isi queue setelah setiap proses, sehingga main.cpp berfungsi sebagai simulasi dan demonstrasi cara kerja circular queue dalam program.

Screenshots Output :



C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

queue.h

```
#ifndef QUEUE_H_INCLUDED
#define QUEUE_H_INCLUDED

#include <iostream>
using namespace std;

typedef int infotype;

struct Queue {
   infotype info[5];
   int head;
   int tail;
};

void createQueue(Queue &Q);
bool isEmptyQueue(Queue Q);
bool isFullQueue(Queue Q);
void enqueue(Queue &Q, infotype x);
infotype dequeue(Queue &Q);
void printInfo(Queue Q);
```

```
#endif
```

Deskripsi:

File queue.h ini adalah header file yang mendefinisikan struktur dan operasi dasar untuk antrian (queue). Di dalamnya terdapat deklarasi struct Queue yang menyimpan array info sebagai tempat data serta variabel head dan tail sebagai penanda posisi awal dan akhir antrian. File ini juga mendeklarasikan fungsi-fungsi penting seperti createQueue untuk inisialisasi, isEmptyQueue dan isFullQueue untuk pengecekan kondisi antrian, enqueue untuk menambah data, dequeue untuk mengeluarkan data, serta printInfo untuk menampilkan isi antrian. Dengan kata lain, queue.h berfungsi sebagai blueprint dari queue sebelum diimplementasikan di file .cpp.

queue.cpp

```
#include "queue.h"

void createQueue(Queue &Q) {
   Q.head = -1;
   Q.tail = -1;
}

bool isEmptyQueue(Queue Q) {
   return (Q.head == -1 && Q.tail == -1);
}

bool isFullQueue(Queue Q) {
   return (Q.tail == 4);
}

void enqueue(Queue &Q, infotype x) {
   if (isFullQueue(Q)) {
      cout << "Queue penuh!" << endl;
      return;
   }

   if (isEmptyQueue(Q)) {
      Q.head = 0;
      Q.tail = 0;
   } else {
      Q.tail++;
   }

   Q.info[Q.tail] = x;
}

infotype dequeue(Queue &Q) {
   if (isEmptyQueue(Q)) {
```

```
        cout << "Queue kosong!" << endl;
        return -1;
    }

    infotype x = Q.info[Q.head];

    if (Q.head == Q.tail) {
        Q.head = -1;
        Q.tail = -1;
    } else {
        Q.head++;
    }

    return x;
}

void printInfo(Queue Q) {
    cout << Q.head << " - " << Q.tail << " | ";

    if (isEmptyQueue(Q)) {
        cout << "empty queue" << endl;
        return;
    }

    for (int i = Q.head; i <= Q.tail; i++) {
        cout << Q.info[i] << " ";
    }
    cout << endl;
}
```

Deskripsi:

File queue.cpp adalah file implementasi yang menjalankan fungsi-fungsi queue yang dideklarasikan di queue.h. Di dalamnya terdapat proses inisialisasi queue, pengecekan kosong atau penuh, penambahan data dengan enqueue, penghapusan data dengan dequeue, serta penampilan isi queue melalui printInfo. File ini mengatur kerja antrian menggunakan indeks head dan tail sehingga operasi queue dapat berjalan dengan benar.

main.cpp

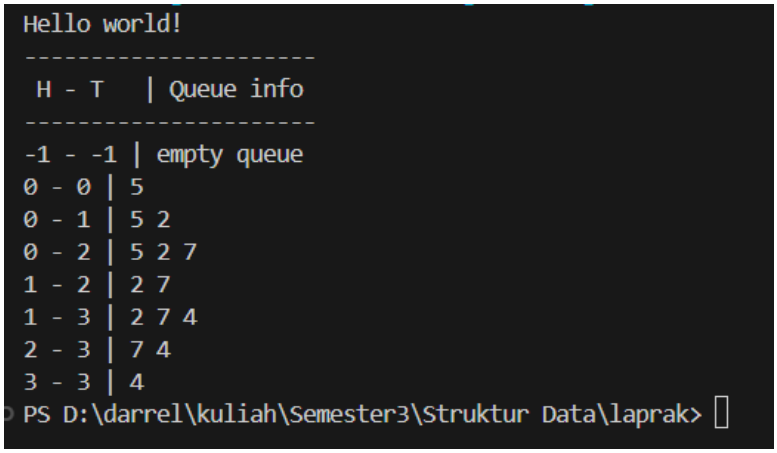
```
#include <iostream>

#include "queue.h"

#include "queue.cpp"


using namespace std;
```

```
int main() {
    cout << "Hello world!" << endl;

    Queue Q;
    createQueue(Q);

    cout << "----------------------" << endl;
    cout << " H - T \t | Queue info" << endl;
    cout << "----------------------" << endl;

    printInfo(Q);
    enqueue(Q, 5); printInfo(Q);
    enqueue(Q, 2); printInfo(Q);
    enqueue(Q, 7); printInfo(Q);
    dequeue(Q);   printInfo(Q);
    enqueue(Q, 4); printInfo(Q);
    dequeue(Q);   printInfo(Q);
    dequeue(Q);   printInfo(Q);

    return 0;
}
```

Deskripsi:

File main.cpp ini merupakan file utama yang digunakan untuk menguji dan menjalankan operasi queue yang telah dibuat. Di dalamnya dibuat sebuah objek Queue, kemudian dilakukan inisialisasi dengan createQueue, setelah itu dilakukan beberapa operasi seperti enqueue untuk menambah data, dequeue untuk mengeluarkan data, serta printInfo untuk melihat perubahan isi queue setelah setiap operasi. Program ini juga menampilkan posisi head dan tail sehingga pengguna bisa melihat secara jelas kondisi queue selama proses berlangsung.

Screenshots Output Unguided:



D. Kesimpulan

Kesimpulan modul 8 ini adalah mempelajari dan mengimplementasikan struktur data queue (antrian) menggunakan C++, mulai dari pembuatan header untuk deklarasi, file implementasi untuk logika operasi queue seperti inisialisasi, pengecekan kosong dan penuh, enqueue, dequeue, hingga penampilan isi queue, serta menguji seluruh fungsi tersebut melalui program utama sehingga queue dapat bekerja sesuai konsep FIFO (First In First Out).

E. Referensi

Kaswar, A. B., & Zain, S. G. (2021). Mudah Belajar Pemrograman Dasar C++. Syiah Kuala University Press.

Hanief, S., Jepriana, I. W., & Kom, S. (2020). Konsep Algoritme dan Aplikasinya dalam Bahasa Pemrograman C++. Penerbit Andi.

Imamuddin, A., & Sobarnas, M. A. (2021). PEMBELAJARAN JARAK JAUH PEMROGRAMAN DASAR MENGGUNAKAN BAHASA C++ UNTUK UMUM: SEBUAH PROGRAM PENGABDIAN KEPADA MASYARAKAT. BEMAS: Jurnal Bermasyarakat, 1(2), 59-67.