

**LAPORAN PRAKTIKUM KE-2**  
**PROYEK STRUKTUR DATA DAN ALGORITMA**



**DISUSUN OLEH :**

Nama : I Nyoman Dimas Kresna Adryan  
NPM : G1A02377  
Kelas : A2

**Nama Asisten Dosen :**

- |                             |             |
|-----------------------------|-------------|
| 1. Davi Sulaiman            | (G1A022001) |
| 2. Attiya Dianti Fadli      | (G1A022002) |
| 3. Abdi Agung Kurniawan     | (G1A022011) |
| 4. Sophina Shafa Salsabila  | (G1A022021) |
| 5. Evelyn Funiike Aritonang | (G1A022024) |
| 6. Diodo Arrahman           | (G1A022027) |
| 7. Sinta Ezra Wati Gulo     | (G1A022040) |
| 8. Wahyu Ozorah Manurung    | (G1A022060) |
| 9. Alif Nurhidayat          | (G1A022073) |
| 10. Ahmad Radesta           | (G1A022086) |

**Dosen Pengampu :**

1. Mochammad Yusa, S.Kom., M.Kom.
2. Arie Vatesia, S.T., M.TI, Ph.D.

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS BENGKULU**  
**2024**

## LANDASAN TEORI

Bahasa pemrograman, atau sering diistilahkan dengan bahasa komputer, merupakan teknik komando atau instruksi standar untuk memerintah komputer. Bahasa pemrograman dapat dimanfaatkan untuk membangun sebuah sistem aplikasi sesuai dengan kebutuhan yang dapat dimanfaatkan dalam berbagai bidang baik dalam bidang pendidikan, ekonomi, bisnis bahkan dalam bidang sosial budaya. Sampai saat ini terdapat banyak jenis Bahasa pemrograman. Salah satu di antaranya adalah Bahasa pemrograman C++ di mana bahasa pemrograman C++ ini merupakan hybrid dari bahasa C.

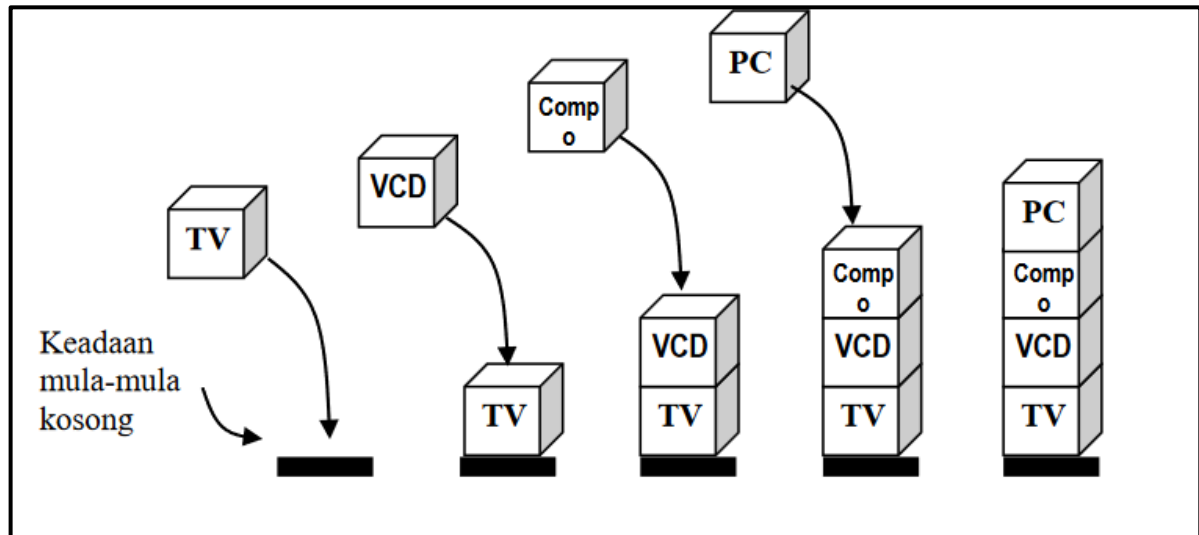
C ++ adalah bahasa pemrograman tingkat tinggi yang digunakan untuk membangun aplikasi perangkat lunak dan sistem komputer. C++ merupakan turunan dari bahasa C dan menambahkan fitur-fitur seperti abstraksi data, *overloading* operator, dan *polymorphism*, sehingga dapat digunakan untuk pengembangan aplikasi yang kompleks dan besar. Tujuan dari bahasa pemrograman C++ yaitu untuk memberikan bahasa pemrograman yang kuat dan fleksibel untuk mengembangkan perangkat lunak tingkat tinggi yang efisien dan cepat.

C++ dikenal sebagai bahasa pemrograman yang cepat dan efisien karena kemampuan untuk mengakses memori langsung dan mengoptimalkan kode untuk meningkatkan kecepatan dan efisiensi. Dalam C++, pengembang dapat melakukan optimasi kode dan mengakses memori langsung untuk meningkatkan kinerja. Selain itu, C++ didesain untuk mendukung portabilitas lintas platform, sehingga kode yang dibuat di satu sistem operasi dapat dijalankan di sistem operasi yang berbeda tanpa perlu memodifikasi kodenya.

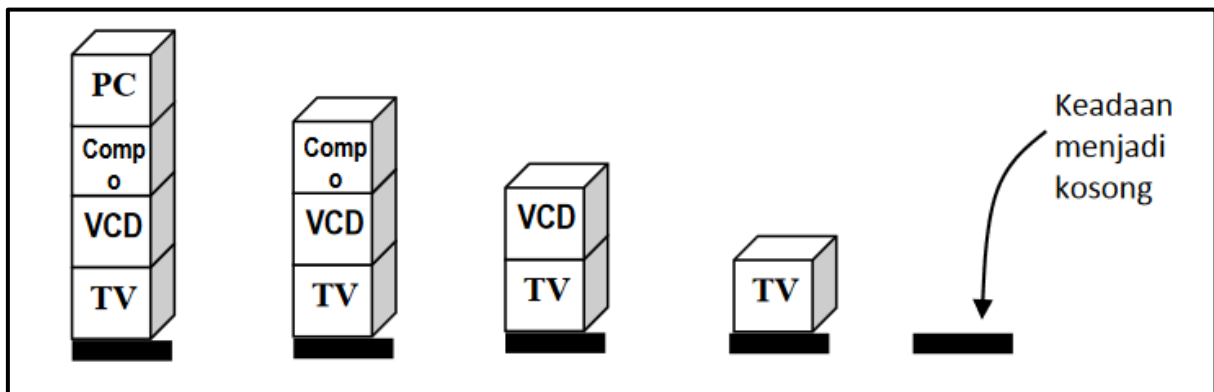
pada tahun 1983, seorang doktor bernama Bjarne Stroustrup menciptakan bahasa baru yaitu bahasa C++ yang merupakan bahasa hybrid dari bahasa C. Bahasa C++ didasarkan atas bahasa C sehingga kita dapat melakukan kompilasi program-program yang ditulis dalam bahasa C dengan menggunakan kompiler C++. Pada mulanya C++ disebut “a better C”. Nama C++ sendiri diberikan oleh Rick Mascitti pada musim panas 1983. Adapun tanda ++ berasal dari nama operator penaikan pada bahasa C. Keistimewaan dari bahasa C++ adalah dapat mendukung pemrograman berorientasi objek atau dikenal dengan istilah Object Oriented Programming (OOP).

Stack atau tumpukan adalah suatu struktur data yang penting dalam pemrograman dengan metode pemrosesan yang bersifat LIFO (Last In First Out) dimana objek/benda yang terakhir masuk ke dalam stack akan menjadi benda pertama yang dikeluarkan dari stack. Dengan model demikian, maka hanya bagian paling atas saja dari stack (TOP) yang bisa di akses. Salah satu kelebihan stack adalah bahwa struktur data tersebut dapat di implementasikan baik pada array maupun pada linked list. Tumpukan banyak digunakan dalam ilmu komputer dan pemrograman untuk mengelola pemanggilan fungsi, operasi pembatalan-pengulangan, evaluasi ekspresi, dan

banyak lagi. Misalnya, ketika suatu fungsi dipanggil, parameter dan variabel lokal dimasukkan ke dalam tumpukan, dan ketika fungsi tersebut kembali, nilainya akan dikeluarkan dari tumpukan. Tumpukan juga digunakan dalam operasi undo-redo, dimana status objek sebelumnya dimasukkan ke dalam tumpukan, dan status saat ini dikeluarkan dari tumpukan untuk membatalkan perubahan. Ilustrasi stack sebagai berikut :



Jika ingin mengambil sesuatu pada kumpulan stack, maka pengambilan akan diambil dari bagian atas, Sebagai contoh pada gambar diatas misalnya ingin diambil VCD secara langsung maka Compo dan PC akan jatuh.



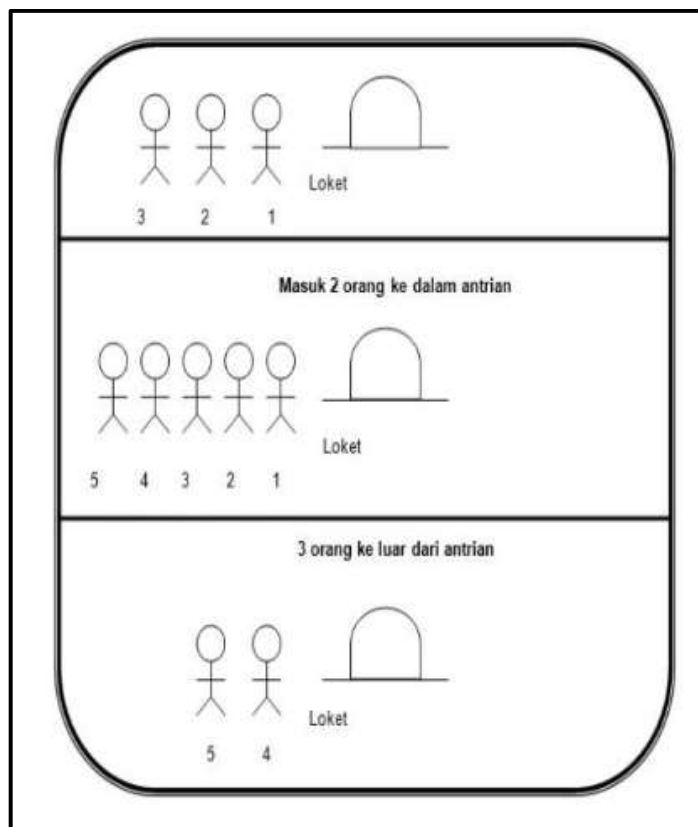
Stack memiliki beberapa operasi atau fungsi yang dapat dilakukan, diantaranya yaitu Push : digunakan untuk menambah item pada stack pada tumpukan paling atas, Pop : digunakan untuk mengambil item pada stack pada tumpukan paling atas, Clear : digunakan untuk mengosongkan stack, IsEmpty : fungsi yang digunakan untuk mengecek apakah stack sudah kosong dan IsFull : fungsi yang digunakan untuk mengecek apakah stack sudah penuh.

Kebalikan dari stack, queue (antrian) adalah suatu jenis struktur data yang dapat diproses dengan sifat FIFO (First In First Out), dimana elemen yang pertama kali masuk ke antrian akan keluar pertama kalinya. Ada dua jenis operasi yang bias dilakukan di antrian : enqueue (memasukkan elemen baru ke dalam elemen) dan dequeue (adalah mengeluarkan satu elemen

dari suatu antrian). Antrian dapat dibuat dengan menggunakan: Linier Array dan Circular Array.

Dalam implementasi struktur data, queue adalah struktur data dengan penyisipan di satu ujung dan penghapusan di ujung lainnya. Ujung penyisipan disebut Rear/Tail/ Ekor sedangkan ujung penghapusan disebut Front/Head/Depan. Ketika elemen terdepan (pertama) dilayani, maka secara logika terjadi pergeseran posisi elemen-elemen dibelakangnya satu langkah, yaitu :

- elemen ke dua akan berpindah ke posisi/menjadi elemen pertama
- elemen ke tiga akan berpindah ke posisi/menjadi elemen ke dua
- elemen ke empat akan berpindah ke posisi/menjadi elemen ke tiga...
- elemen ke (n) akan berpindah ke posisi/menjadi elemen ke (n-1)



Antrian banyak digunakan dalam ilmu komputer dan pemrograman untuk mengatur penjadwalan pekerjaan, sistem pesan, antrian printer, dan banyak lagi. Misalnya, dalam sistem penjadwalan pekerjaan, pekerjaan ditambahkan ke antrian, dan pekerjaan dengan prioritas tertinggi dikeluarkan dari antrian terlebih dahulu. Dalam sistem pesan, pesan ditambahkan ke antrian, dan pesan dikeluarkan dari antrian sesuai urutan penerimaannya.

Perbedaan utama antara struktur data tumpukan dan antrian adalah urutan elemen diakses dan dihapus. Dalam tumpukan, elemen terakhir yang ditambahkan adalah elemen pertama yang dihapus, sedangkan dalam antrian, elemen pertama yang ditambahkan adalah elemen pertama yang dihapus.

## SOAL DAN PEMBAHASAN\

1. Buatlah sebuah kasus serta kode program stack dan queue dalam kehidupan sehari-hari selain yang dicontohkan saat praktikum (masing masing minimal 1 contoh)

### A. Program stack

Pada hari minggu, Eca sedang membantu ibu untuk mengerjakan pekerjaan rumah yaitu melipat pakaian, Setelah melipat pakaian, Eca menaruh pakaian yang telah dilipatnya ke almari, pakaian yang ia masukan pertama kali yaitu polo, dan dilanjutkan baju jersey, dan kemeja.

#### Printscreen Source Code :

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Stack {
6     string data[5];
7     int bottom, top;
8 };
9
10 Stack st = { {}, 0, -1 };
11
12 void push() {
13     if (st.top == 4) {
14         cout << "STACK PLEMAN!\n";
15     } else {
16         st.top = st.top + 1;
17         cout << "\nINPUT DATA : ";
18         cin >> st.data[st.top];
19     }
20 }
21
22 void display() {
23     int j, i;
24     if (st.bottom > st.top)
25         cout << "STACK KOSONG!\n";
26 }
```

G1A023077

Gambar 1.1 Source Code

```
27
28     j = (st.top + 1) - st.bottom;
29     cout << "\nTotal Stack terdapat " << j << " elemen, yaitu " << endl;
30     for (i = 0; i <= st.top; i++) {
31         cout << st.data[i] << endl;
32     }
33     cout << "\n\n";
34 }
35
36 void pop() {
37     if (st.bottom > st.top)
38         cout << "Stack sudah kosong!\n";
39     else {
40         st.top = st.top - 1;
41         cout << "Penghapusan Berhasil!\n";
42     }
43 }
44
45 int main() {
46     int choice = 0;
47
48     while (choice != 4) {
49         cout << "PILIHAN KAMI FROM UAS1!\n";
50         cout << "1. Tambah Baju ke dalam lemari!\n";
51         cout << "2. Lihat Susunan Baju pada lemari!\n";
52         cout << "3. Hapus Baju dari lemari!\n";
53     }
```

G1A023077

Gambar 1.2 Source Code

```
54     cout << "4. Keluar!\n";
55
56     cout << "Pilihkan!";
57     cin >> choice;
58
59     switch (choice) {
60         case 1:
61             push();
62             break;
63         case 2:
64             display();
65             break;
66         case 3:
67             pop();
68             break;
69         case 4:
70             cout << "\nSELESAI!\n";
71             break;
72         default:
73             cout << "\nPilihan tidak valid. silakan pilih kembali!\n";
74     }
75
76     return 0;
77 }
```

G1A023077

Gambar 1.3 Source Code

**Source Code :**

```
#include <iostream>
#include <string>
using namespace std;
struct Stack {
    string data[5];
    int bottom, top;
};
Stack st = { {}, 0, -1 };
void push() {
    if (st.top == 4)
        cout << "STACK PENUH\n";
    else {
        st.top = st.top + 1;
        cout << "\nINPUT DATA :";
        cin >> st.data[st.top];
    }
}
void display() {
    int je, i;
    if (st.bottom > st.top)
        cout << "STACK KOSONG\n";
    else {
        je = (st.top + 1) - st.bottom;
        cout << "\n\nIsi Stack terdapat " << je << " elemen, yaitu "<< endl;
        for (i = 0; i <= st.top; i++) {
            cout << st.data[i] << endl;
        }
        cout << "\n\n";
    }
}
void pop() {
    if (st.bottom > st.top)
        cout << "Stack Sudah Kosong\n";
    else {
```

```

        st.top = st.top - 1;
        cout << "Penghapusan Berhasil\n";
    }
}

int main() {
    int choice = 0;
    while (choice != 4) {
        cout << "TUMPUKAN BAJU PADA LEMARI\n\n";
        cout << "1. Tambah Baju ke Dalam Lemari\n";
        cout << "2. Lihat Susunan Baju pada Lemari\n";
        cout << "3. Ambil Baju dari Lemari\n";
        cout << "4. Keluar\n";
        cout << "Pilihan:";
        cin >> choice;
        switch (choice) {
            case 1:
                push();
                break;
            case 2:
                display();
                break;
            case 3:
                pop();
                break;
            case 4:
                cout << "\nSELESAI\n";
                break;
            default:
                cout << "\nPilihan tidak valid. Silakan pilih kembali.\n";
        }
    }
}

```

### **Pembahasan Source Code :**

Program ini diawali dengan `#include <iostream>` yang dimana ini merupakan pernyataan preprocessor yang menginstruksikan compiler untuk menyertakan library input-output standar C++ yang akan digunakan dalam program dan dilanjutkan dengan `using namespace std;` untuk menghilangkan kebutuhan untuk menuliskan `std::` di depan objek dan fungsi dari library standar C++ yang akan digunakan. Selanjutnya yaitu `int main()` ini adalah fungsi utama dari program

C++, yang akan dieksekusi terlebih dahulu saat program di mulai. Selanjutnya yaitu **struct Stack** { **string data[5]; int bottom, top;** berfungsi sebagai struct yang menyimpan data baju dalam array 'data' dengan ukuran 5 dengan **Stack st = { {}, 0, -1 }**; yaitu bottom awalnya yaitu 0 dan top awalnya yaitu -1. Selanjutnya yaitu **void push()** { berfungsi untuk menambahkan baju kedalam lemari dengan permissalan **if (st.top == 4) cout << "STACK PENUH\n";** jika top sudah mencapai 4, maka stack penuh, dan **else { st.top = st.top + 1; cout << "\nINPUT DATA :"; cin >> st.data[st.top];** yaitu jika tidak penuh maka user diminta untuk memasukkan data baju baru kedalam 'data' pada indeks 'top'. Selanjutnya yaitu **void display()** { berfungsi untuk digunakan menampilkan isi susunan baju dalam lemari dengan permissalan **if (st.bottom > st.top) cout << "STACK KOSONG\n";** yaitu jika bottom lebih besar dari 'top', itu berarti stack kosong, dan **else { je = (st.top + 1) - st.bottom; cout << "\nIsi Stack terdapat " << je << " elemen, yaitu " << endl;** jika tidak kosong, jumlah elemen dihitung dengan 'top + 1) – bottom', lalu setiap elemen stack ditampilkan. Selanjutnya **void pop()** { berfungsi untuk mengambil baju dari lemari (menghapus stack), dengan permissalan **if (st.bottom > st.top) cout << "Stack Sudah Kosong\n";** yaitu jika 'bottom' lebih besar dari 'top' itu berarti stack sudah kosong, dan **else { st.top = st.top - 1; cout << "Penghapusan Berhasil\n";** yaitu jika tidak kosong, to dikurangi 1 untuk menghapus baju teratas. Selanjutnya yaitu **int main()** { adalah tempat Dimana program berjalan dengan isi perulangan while, dan percabangan switch yang sudah dirancang.

### Output :

```

PROGRAM : OUTPUT DERESCONSOLE TERMINAL FILE
INPUT DATA : POLO
TUMPUKAN BAJU PADA LEMARI
1. Tambah Baju ke Dalam Lemari
2. Lihat Susunan Baju pada Lemari
3. Ambil Baju dari Lemari
4. Keluar
Pilihan1
INPUT DATA : JERSEY
TUMPUKAN BAJU PADA LEMARI
1. Tambah Baju ke Dalam Lemari
2. Lihat Susunan Baju pada Lemari
3. Ambil Baju dari Lemari
4. Keluar
Pilihan2

Isi stack terdapat 2 elemen, yaitu
POLO
JERSEY

TUMPUKAN BAJU PADA LEMARI
1. Tambah Baju ke Dalam Lemari
2. Lihat Susunan Baju pada Lemari
3. Ambil Baju dari Lemari
4. Keluar
Pilihan

```

Gambar 1.4 Output

### Pembahasan Output :

Pada pemrograman kali ini merupakan bentuk stack tentang tumpukan baju dalam lemari dengan menggunakan stack dengan konsep stack yaitu LIFO (Last In First Out) yaitu elemen yang terakhir masuk akan dieksekusi pertama kali, pada output kali ini terdapat daftar pilihan yaitu : 1. Tambah baju ke dalam lemari, 2. Lihat susunan baju pada lemari, dan 3. Ambil baju pada lemari, Dimana jika pengguna memilih nomer 1, maka pengguna akan diminta untuk



mengisi data baju untuk dimasukan ke dalam tumpukan baju. Jika pengguna memilih nomer 2 maka akan memperlihatkan elemen yang telah masuk kedalam stack. Jika pengguna memilih nomer 3 maka data stack paling atas yang telah dimasukkan akan dihapus secara otomatis dan data dibawahnya otomatis naik. Dengan penggunaan stack ini data yang dimasukan paling akhir adalah data yang akan dihapus atau diambil paling awal.

## B. Program queue

Pada hari senin, Zaki, Rahman, dan Yohanes ingin pergi ke rumah sakit untuk melakukan cek Kesehatan sebagai syarat untuk tes kedinasan. Setelah sampai di rumah sakit, mereka masuk keruang cek Kesehatan dan disuruh masuk kedalam antrian dengan yang pertama mengantri yaitu Yohanes, di ikuti Zaki dan Rahman.

### Printscreen Souce Code:

```

3  #include<iostream>
4  using namespace std;
5
6  #define max 5
7  string Data[max];
8  int top = 0;
9
10 bool isEmpty() {
11     return (top == 0);
12 }
13
14 bool isFull() {
15     return (top == 5);
16 }
17
18 void view() {
19     if (isEmpty()) {
20         cout << "Antrian kosong" << endl;
21     } else {
22         for (int a = 0; a < top; a++) {
23             cout << a + 1 << ". " << Data[a] << endl;
24         }
25     }
26 }
27
28 if (isFull()) {

```

G1A023077

Gambar 1.5 Source Code

```

29     cout << "Antrian penuh" << endl;
30 }
31
32 void enqueue() {
33     if (isFull()) {
34         cout << "Masukan data : ";
35         cin >> Data[top];
36         top++;
37     } else {
38         cout << "Antrian penuh, tidak bisa enqueue." << endl;
39     }
40 }
41
42 void dequeue() {
43     if (isEmpty()) {
44         for (int a = 0; a < top-1; a++) {
45             Data[a] = Data[a+1];
46         }
47         top--;
48     } else {
49         cout << "Antrian kosong, tidak bisa dequeue." << endl;
50     }
51 }
52
53 }

```

G1A023077

Gambar 1.6 Source Code

```

54 int main() {
55     int pil;
56     string isi;
57
58     queue;
59     view();
60     cout << "ANTRIAN CEK KESKATAN YAKI, NASIR, YUZ, KELUAR (pilih) : ";
61     cin >> pil;
62
63     if (pil == 1) {
64         cout << "Masuk" << endl;
65         enqueue();
66         goto queue;
67     } else if (pil == 2) {
68         cout << "Antrian" << endl;
69         dequeue();
70         goto queue;
71     }
72
73     return 0;
74 }

```

G1A023077

Gambar 1.7 Source Code

**Source Code :**

```
using namespace std;

#define max 5

string Data[max];
int top = 0;

bool isFull() {
    return (top >= max);
}

bool isEmpty() {
    return (top == 0);
}

void view() {
    if (!isEmpty()) {
        cout << "Data Antrian : " << endl;
        for (int a = top - 1; a >= 0; a--) {
            cout << a + 1 << ". " << Data[a] << endl;
        }
    } else {
        cout << "Antrian Kosong" << endl;
    }

    if (isFull()) {
        cout << "Antrian Penuh" << endl;
    }

    cout << endl;
}

void enqueue() {
    if (!isFull()) {
        cout << "Masukkan data : ";
        cin >> Data[top];
        top++;
    } else {
```

```

        cout << "Antrian Penuh, tidak bisa enqueue." << endl;
    }
}

void dequeue() {
    if (!isEmpty()) {
        for (int a = 0; a < top-1; a++) {
            Data[a] = Data[a+1];
        }
        top--;
    } else {
        cout << "Antrian Kosong, tidak bisa dequeue." << endl;
    }
}

int main() {
    int pil;
    string isi;

    queue:
    view();
    cout << "ANTRIAN CEK KESEHATAN \n1. MASUK \n2. KELUAR \nPilihan : ";
    cin >> pil;
    if (pil == 1) {
        cout << "MASUK" << endl;
        enqueue();
        goto queue;
    } else if (pil == 2) {
        cout << "KELUAR" << endl;
        dequeue();
        goto queue;
    }
}

```

#### **Pembahasan Source Code :**

Program ini diawali dengan `#include <iostream>` yang dimana ini merupakan pernyataan preprocessor yang menginstruksikan compiler untuk menyertakan library input-output standar C++ yang akan digunakan dalam program dan dilanjutkan dengan `using namespace std;` untuk menghilangkan kebutuhan untuk menuliskan `std::` didepan objek dan fungsi dari library standar C++ yang akan digunakan. Selanjutnya yaitu `int main()` ini adalah fungsi utama dari program

### Output :

G1A023077

---

I NYOMAN DIMAS KRESNA ADRYAN

## Pembahasan Output :

Program ini merupakan bentuk queue yang dimana data yang paling pertama masuk maka akan data yang pertama kali keluar, dalam pemrograman ini mengambil tema antrian cek Kesehatan dimana data tiap orang yang ingin masuk dapat dimasukan dan akan keluar dahulu sesuai dengan posisi setelah masuk. Dalam outputnya memperlihatkan judul dari program ini yaitu “Antrian Cek Kesehatan” dimana ada dua pilihan didalam tersebut yaitu : 1. Masuk, dan 2. Keluar, dimana jika pengguna memilih nomer 1 maka nanti pengguna akan diminta untuk mengisi data untuk di input ke antrian, dan jika pengguna memilih nomer 2 maka nanti antrian yang paling baawah atau paling awal masuk akan otomatis terhapus, dengan begitu kita dapat memasukan data kedalam antrian dan menghapus atau mengeluarkan antrian sesuai dengan data yang paling lama didalam antrian dalam hal ini yaitu antrian cek Kesehatan.

2. Di sebuah kidzstation, Baim membawa list untuk membeli tumpukan mainan. Mobil 2 buah, Robot 4 buah, Lego 3 buah, Motor 1 buah, Puzzle 6 Buah dan play-doh 3 buah. Namun di toko tersebut hanya menyediakan 5 buah puzzle. Buatlah program mengenai ilustrasi soal kemudian tunjukkan jumlah barang yang dibeli, serta kondisi barangnya (boolean)

## Printline Source Code :

```
1 // Antrian Cek Kesehatan
2 #include <iostream>
3 using namespace std;
4
5 int maksimal = 0;
6 string mainstack[10];
7 int top = 0;
8
9 bool isEmpty() {
10     return top == maksimal;
11 }
12
13 bool isFull() {
14     return top == 0;
15 }
16
17 void push(string data) {
18     mainstack[top] = data;
19     top++;
20 }
21
22 void pop() {
23     if (top > 0) {
24         for (int i = start_index; i < top; i++) {
25             mainstack[i] = "";
26         }
27         top = start_index;
28     }
29 }
30
31 void displayStack() {
32     cout << "Tumpukan Mainan:\n";
33     for (int i = top - 1; i >= 0; i--) {
34         cout << mainstack[i] << endl;
35     }
36     cout << "\n";
37 }
38
39 int main() {
40     push("Mobil"); push("Mobil");
41     push("Robot"); push("Robot"); push("Robot"); push("Robot");
42     push("Lego"); push("Lego"); push("Lego");
43     push("Motor");
44     push("Puzzle"); push("Puzzle"); push("Puzzle"); push("Puzzle"); push("Puzzle"); push("Puzzle");
45     push("Play Doh"); push("Play Doh"); push("Play Doh");
46     displayStack();
47
48     int mainan_keluar = 4;
49     cout << "Mengeluarkan " << mainan_keluar << " buah mainan" << endl;
50     pop(mainan_keluar);
51     displayStack();
52
53     cout << "Masukkan mainan kembali:" << endl;
54     push("Play Doh");
55     push("Play Doh");
56     push("Play Doh");
57     displayStack();
58     return 0;
59 }
```

G1A023077

Gambar 2.1 Source Code

```
16 }
17
18 int main() {
19     push("Mobil"); push("Mobil");
20     push("Robot"); push("Robot"); push("Robot"); push("Robot");
21     push("Lego"); push("Lego"); push("Lego");
22     push("Motor");
23     push("Puzzle"); push("Puzzle"); push("Puzzle"); push("Puzzle"); push("Puzzle"); push("Puzzle");
24     push("Play Doh"); push("Play Doh"); push("Play Doh");
25     displayStack();
26
27     int mainan_keluar = 4;
28     cout << "Mengeluarkan " << mainan_keluar << " buah mainan" << endl;
29     pop(mainan_keluar);
30     displayStack();
31
32     cout << "Masukkan mainan kembali:" << endl;
33     push("Play Doh");
34     push("Play Doh");
35     push("Play Doh");
36     displayStack();
37     return 0;
38 }
```

G1A023077

Gambar 2.2 Source Code

**Source Code :**

```
#include <iostream>
using namespace std;

int maksimal = 19;
string mainanstack[19];
int top = 0;

bool isFull() {
    return top == maksimal;
}

bool isEmpty() {
    return top == 0;
}

void push(string data) {
    mainanstack[top] = data;
    top++;
}

void pop(int count) {
    int start_index = top - count;
    for (int i = start_index; i < top; ++i) {
        mainanstack[i] = "";
    }
    top = start_index;
}

void displayStack() {
    cout << "Tumpukan Mainan:\n";
    for (int i = top - 1; i >= 0; i--) {
        cout << mainanstack[i] << endl;
    }
    cout << "\n";
}
```

```

    }

    int main() {
        push("Mobil"); push("Mobil");
        push("Robot"); push("Robot"); push("Robot"); push("Robot");
        push("Lego"); push("Lego"); push("Lego");
        push("Motor");

        push("Puzzle");push("Puzzle");push("Puzzle");push("Puzzle");push("Puzzle");push(
        "Puzzle");
        push("Play Doh");push("Play Doh");push("Play Doh");
        displayStack();

        cout << "Masukkan mainan kembali:" << endl;
        displayStack();
        return 0;
    }

```

#### **Pembahasan Source Code :**

Program ini diawali dengan `#include <iostream>` yang dimana ini merupakan pernyataan preprocessor yang menginstruksikan compiler untuk menyertakan library input-output standar C++ yang akan digunakan dalam program setelah itu `using namespace std;` untuk menghilangkan kebutuhan untuk menuliskan `std::` didepan objek dan fungsi dari library standar C++ yang akan digunakan. Selanjutnya yaitu `int maksimal = 19;` merupakan variabel ukuran maksimum tumpukan, `string mainanstack[19];` merupakan array menyimpan nama-nama mainan, dan `int top = 0;` merupakan variabel teratas dari tumpukan. Selanjutnya yaitu `bool isFull() { return top == maksimal;` berfungsi sebagai menjalankan true jika tumpukan penuh dan menjalankan false jika belum. Selanjutnya `bool isEmpty() { return top == 0;` berfungsi mengembalikan permisalan true jika tumpukan kosong dan false jika tidak. Selanjutnya `void push(string data) { mainanstack[top] = data; top++;` berfungsi sebagai untuk menambahkan elemen baru ke tumpukan, elemen dimasukkan ke posisi top dan ditingkatkan sehingga menuju ke elemen selanjutnya. `void pop(int count) { int start_index = top - count; for (int i = start_index; i < top; ++i) { mainanstack[i] = "";` berfungsi untuk menghapus sejumlah count elemen dari tumpukan yang dimulai dari elemen teratas. selanjutnya `void displayStack() { cout << "Tumpukan Mainan:\n";` berfungsi untuk menampilkan seluruh isi tumpukan, dimulai dari elemen teratas dengan menggunakan loop for. Selanjutnya yaitu `int main() {` untuk mencetak pesan ke layar output.

### Output :

```
Mengeluarkan 4 buah mainan
Tumpukan Mainan:
Puzzle
Puzzle
Puzzle
Puzzle
Puzzle
Motor
Lego
Lego
Lego
Robot
Robot
Robot
Mobil
Mobil
```

G1A023077

Gambar 2.3 Output

```
Mengeluarkan mainan ke dalam
Tumpukan Mainan:
Play Doh
Play Doh
Play Doh
Puzzle
Puzzle
Puzzle
Puzzle
Motor
Lego
Lego
Lego
Robot
Robot
Robot
Mobil
Mobil
```

G1A023077

Gambar 2.4 Output

### Pembahasan Output :

Pada program ini merupakan gabungan dari stack dan queue dimana stack dipakai untuk menyimpan daftar barang yang ingin dibeli dan queue dipakai untuk menyimpan informasi ketersediaan barang, dimana pada program ini semua data dari variabel yaitu mainan yang ingin dibeli sudah diinput langsung pada program, ada dua dua output yaitu mainan yang ingin dibeli oleh pembeli dan juga mainan yang tersedia di took, dimana pada toko mainan puzzle hanya tersedia 5 buah sedangkan permintaan pembeli menginginkan 6 puzzle.

- Adryan, Ghazi, Rayhan, Habib, dan Yohannes mengantri untuk membeli tiket shikansen dengan tujuan Tokyu-Shibuya, dengan jumlah 5 tiket, 6 tiket, 8 tiket, 1 tiket dan 7 tiket. Ternyata 1 teman Adryan tidak jadi berangkat. Buatlah antrian dari ilustrasi diatas kemudian tunjukkan jumlah tiket yang dibeli dan kondisi tiket (boolean). Serta diurutkan dari jumlah tiket terbanyak.

### Printline Source Code :

```
1 // Stack dan Queue
2 // Stack: Last In First Out
3 // Queue: First In First Out
4
5 // Stack
6 // Push: Menambahkan elemen ke stack
7 // Pop: Menghapus elemen dari stack
8 // IsEmpty: Mengecek apakah stack kosong
9 // Size: Mendapatkan jumlah elemen di stack
10
11 // Queue
12 // Enqueue: Menambahkan elemen ke queue
13 // Dequeue: Menghapus elemen dari queue
14 // IsEmpty: Mengecek apakah queue kosong
15 // Size: Mendapatkan jumlah elemen di queue
16
17 // Contoh penggunaan
18 // Stack
19 Stack s = new Stack();
20 s.Push(1);
21 s.Push(2);
22 s.Push(3);
23 s.Pop();
24 s.Pop();
25 s.Pop();
26
27 // Queue
28 Queue q = new Queue();
29 q.Enqueue(1);
30 q.Enqueue(2);
31 q.Enqueue(3);
32 q.Dequeue();
33 q.Dequeue();
34 q.Dequeue();
```

G1A023077

Gambar 3.1 Source Code



```

1 // Queue using Array using STL & Linked List & Array
2
3 #include <iostream>
4
5 using namespace std;
6
7 #define max 27
8 string Data[max];
9 int top = 0;
10
11 bool isFull() {
12     return (top >= max);
13 }
14
15 bool isEmpty() {
16     return (top == 0);
17 }
18
19 void enqueue(string item) {
20     Data[top] = item;
21     top++;
22 }
23
24 void dequeue(int count) {
25     for (int j = 0; j < count; j++) {
26         for (int a = 0; a < top - 1; a++) {
27             Data[a] = Data[a + 1];
28         }
29         top--;
30     }
31 }
32
33 int main() {
34     enqueue("Adrian"); enqueue("Adrian"); enqueue("Adrian"); enqueue("Adrian"); enqueue("Adrian");
35     enqueue("Budi"); enqueue("Budi"); enqueue("Budi"); enqueue("Budi"); enqueue("Budi");
36     enqueue("Ayah"); enqueue("Ayah"); enqueue("Ayah"); enqueue("Ayah"); enqueue("Ayah");
37     enqueue("Akhi");
38     enqueue("Nahar"); enqueue("Nahar"); enqueue("Nahar"); enqueue("Nahar"); enqueue("Nahar"); enqueue("Nahar");
39     display();
40
41     int arr[5], k, l, m = 0;
42     cout << "Masukkan k : "; cin >> k;
43     dequeue(k);
44     display();
45
46     cout << "Masukkan k : "; cin >> k;
47     enqueue(k);
48     display();
49
50     cout << "Masukkan k : "; cin >> k;
51     enqueue(k);
52     display();
53
54     cout << "Masukkan k : "; cin >> k;
55     enqueue(k);
56     display();
57
58     cout << "Masukkan k : "; cin >> k;
59     enqueue(k);
60     display();
61
62     cout << "Masukkan k : "; cin >> k;
63     enqueue(k);
64     display();
65
66     cout << "Masukkan k : "; cin >> k;
67     enqueue(k);
68     display();
69
70     cout << "Masukkan k : "; cin >> k;
71     enqueue(k);
72     display();
73
74     cout << "Masukkan k : "; cin >> k;
75     enqueue(k);
76     display();
77
78     cout << "Masukkan k : "; cin >> k;
79     enqueue(k);
80     display();
81
82     cout << "Masukkan k : "; cin >> k;
83     enqueue(k);
84     display();
85
86     cout << "Masukkan k : "; cin >> k;
87     enqueue(k);
88     display();
89
90     cout << "Masukkan k : "; cin >> k;
91     enqueue(k);
92     display();
93
94     cout << "Masukkan k : "; cin >> k;
95     enqueue(k);
96     display();
97
98     cout << "Masukkan k : "; cin >> k;
99     enqueue(k);
100    display();
101
102    return 0;
103 }

```

G1A023077

Gambar 3.2 Source Code

**Source Code :**

```

#include <iostream>
using namespace std;

#define max 27
string Data[max];
int top = 0;

bool isFull() {
    return (top >= max);
}

bool isEmpty() {
    return (top == 0);
}

void enqueue(string item) {
    Data[top] = item;
    top++;
}

void dequeue(int count) {
    for (int j = 0; j < count; j++) {
        for (int a = 0; a < top - 1; a++) {
            Data[a] = Data[a + 1];
        }
        top--;
    }
}

```

```

    }
}

void displayQueue() {
    cout << "Antrian Tiket:\n";
    for (int i = 0; i < top; i++) {
        cout << Data[i] << endl;
    }
    cout << "\n";
}

int main() {
    enqueue("Adryan"); enqueue("Adryan"); enqueue("Adryan"); enqueue("Adryan");
    enqueue("Adryan");

    enqueue("Ghazi"); enqueue("Ghazi"); enqueue("Ghazi"); enqueue("Ghazi");
    enqueue("Ghazi"); enqueue("Ghazi");

    enqueue("Rayhan"); enqueue("Rayhan"); enqueue("Rayhan"); enqueue("Rayhan");
    enqueue("Rayhan"); enqueue("Rayhan"); enqueue("Rayhan"); enqueue("Rayhan");

    enqueue("Habib");

    enqueue("Yohanes"); enqueue("Yohanes"); enqueue("Yohanes");
    enqueue("Yohanes"); enqueue("Yohanes"); enqueue("Yohanes"); enqueue("Yohanes");

    displayQueue();

    int antrian_keluar = 5;
    cout << "Mengeluarkan " << antrian_keluar << " orang" << endl;
    dequeue(antrian_keluar);
    displayQueue();

    cout << "Antrian Baru:" << endl;
    enqueue("Adryan"); enqueue("Adryan"); enqueue("Adryan"); enqueue("Adryan");

```

```
displayQueue();
```

```
cout << "Rayhan : 8 " << endl;  
cout << "Yohanes : 7 " << endl;  
cout << "Ghazi : 6 " << endl;  
cout << "Adryan : 4 " << endl;  
cout << "Habib : 1 " << endl;  
  
return 0;
```

#### Pembahasan Source Code :

Program ini diawali dengan `#include <iostream>` yang dimana ini merupakan pernyataan preprocessor yang menginstruksikan compiler untuk menyertakan library input-output standar C++ yang akan digunakan dalam program dan dilanjutkan dengan `# using namespace std;` untuk menghilangkan kebutuhan untuk menuliskan `std::` didepan objek dan fungsi dari library standar C++ yang akan digunakan. Selanjutnya yaitu `bool isFull() { return (top >= max);` untuk mengembalikan true jika antrian penuh dan false jika tidak. Selanjutnya yaitu `bool isEmpty() { return (top == 0);` berfungsi untuk mengembalikan true jika antrian kosong dan false jika antrian tidak kosong. `void enqueue(string item) {Data[top] = item; top++;` berfungsi untuk menambahkan elemen baru kedalam antrian, elemen ditambahkan di posisi top dalam array kemudian ditingkatkan. `void dequeue(int count) { for (int j = 0; j < count; j++) { for (int a = 0; a < top - 1; a++) { Data[a] = Data[a + 1];` berfungsi sebagai menghapus sejumlah count elemen pertama antrian. Selanjutnya yaitu `void displayQueue() { cout << "Antrian Tiket:\n"; for (int i = 0; i < top; i++) { cout << Data[i] << endl;` berfungsi sebagai menampilkan seluruh antrian, loop for mencetak setiap elemen dalam array data hingga top. Selanjutnya yaitu `int main() {` berfungsi untuk mencetak pesan yang diinginkan ke layar output. Pada output akan mengeluarkan daftar mainan yang ingin dibeli di toko mainan yang di buat dalam stack pada code diatas dan daftar mainan yang tersedia pada took mainan tersebut

#### Output :



Gambar 3.3 Output

```
Mengeluarkan 5 orang
Antrian Tiket:
Ghazi
Ghazi
Ghazi
Ghazi
Ghazi
Rayhan
Rayhan
Rayhan
Rayhan
Rayhan
Rayhan
Rayhan
Yohanes
Yohanes
Yohanes
Yohanes
Yohanes
Yohanes
```

G1A023077

Gambar 3.4 Output

```
Antrian Baru:
Antrian Tiket:
Ghazi
Ghazi
Ghazi
Ghazi
Ghazi
Rayhan
Rayhan
Rayhan
Rayhan
Rayhan
Rayhan
Rayhan
Yohanes
Yohanes
Yohanes
Yohanes
Yohanes
Yohanes
Adryan
Adryan
Adryan
```

G1A023077

Gambar 3.5 Output

```
Rayhan : 8
Yohanes : 7
Ghazi : 6
Adryan : 4
Habib : 1
PS C:\Users\dimas\dryn\coding\C++\LAPRAK 2>
```

G1A023077

Gambae 3.6 Output

**Pembahasan Output :**

Pada program ini merupakan program yang menampilkan sebuah pembelian tiket dimana pada output pertama adalah antrian dalam queue dimana orang yang paling pertama hingga paling terakhir dimana pada kondisi pada soal yaitu Adryan urutan paling pertama dan Yohannes urutan paling akhir, setelah itu dilanjutkan antrian dengan tidak adanya Adryan

karena ada satu yang tidak membeli dan terakhir ada urutan pemnelli dengan jumlah pembelian tiket paling banyak hingga paling sedikit.

## KESIMPULAN DAN SARAN

### A. KESIMPULAN

C ++ adalah bahasa pemrograman tingkat tinggi yang digunakan untuk membangun aplikasi perangkat lunak dan sistem komputer. C++ merupakan turunan dari bahasa C dan menambahkan fitur-fitur seperti abstraksi data, *overloading* operator, dan *polymorphism*, sehingga dapat digunakan untuk pengembangan aplikasi yang kompleks dan besar. Tujuan dari bahasa pemrograman C++ yaitu untuk memberikan bahasa pemrograman yang kuat dan fleksibel untuk mengembangkan perangkat lunak tingkat tinggi yang efisien dan cepat.

C++ merupakan bahasa pemrograman yang cepat dan efisien, terkenal karena kemampuannya mengakses memori langsung dan mengoptimalkan kode untuk meningkatkan kecepatan dan efisiensi. Didukung oleh desainnya yang memperhatikan portabilitas lintas platform, kode yang dibuat dalam C++ dapat dijalankan di berbagai sistem operasi tanpa perlu dimodifikasi. Diciptakan oleh Bjarne Stroustrup pada tahun 1983 sebagai bahasa hybrid dari bahasa C, C++ awalnya disebut "a better C" sebelum diberi nama "C++" oleh Rick Mascitti pada musim panas tahun yang sama, dengan "++" berasal dari operator penambahan pada bahasa C. Keistimewaan utama C++ adalah kemampuannya mendukung pemrograman berorientasi objek (Object Oriented Programming/OOP).

Stack atau tumpukan adalah suatu stuktur data yang penting dalam pemrograman dengan metode pemrosesan yang bersifat LIFO (Last In First Out) dimana objek/benda yang terakhir masuk ke dalam stack akan menjadi benda pertama yang dikeluarkan dari stack. Dengan model demikian, maka hanya bagian paling atas saja dari stack (TOP) yang bisa di akses. Salah satu kelebihan stack adalah bahwa struktur data tersebut dapat di implementasikan baik pada array maupun pada linked list.

Kebalikan dari stack, queue (antrian) adalah suatu jenis struktur data yang dapat diproses dengan sifat FIFO (First In First Out), dimana elemen yang pertama kali masuk ke antrian akan keluar pertama kalinya. Ada dua jenis operasi yang bias dilakukan di antrian : enqueue (memasukkan elemen baru ke dalam elemen) dan dequeue (adalah mengeluarkan satu elemen dari suatu antrian). Antrian dapat dibuat dengan menggunakan: Liniear Array dan Circular Array.

### B. SARAN

Saran saya pada pembelajaran proyek sistem data algoritma pada minggu kedua yaitu harus lebih memahami lagi fungsi tiap kode yang telah dipelajari dalam setiap praktikum karena jika

tidak memahami apa fungsi tiap kode yang dimasukan, maka ketika memasukan kode kita tidak dapat mendeklarasikan kode dengan benar dan maksimal.

#### **DAFTAR PUSTAKA**

Aziz, L (2019). Meningkatkan Proses Pemahaman Operasi Perhitungan Bilangan Bulat Menggunakan Pseudocode dan Program pada Bahasa Pemrograman C++. *Jurnal Bisnis, Manajemen dan Informatika*. (Vol. 16, No 1, 108 – 120)

Joni, L (2010). Media Pembelajaran Bahasa Pemrograman C++. *Jurnal Manajemen Informatika* (Vol. 7, No. 1, 63 – 71)

Soemantri, M. (2005). Membangun Sistem Komputasi Terdistribusi dengan Pemrograman C++. *Jurnal Teknik Elektro*. (Vol. 9, No. 1, 26 – 30)

Effendi, Y. (2022). Buku Ajar Pendidikan Algoritma dan Struktur Data.

Septian, F. (2022). MODEL PEMBELAJARAN PEMROGRAMAN BAHASA C++ UNTUK TUNANETRA DENGAN METODE PROBLEM BASED LEARNING. *Jurnal Maklumatika* ( No. 119 – 125)



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS BENGKULU  
FAKULTAS TEKNIK  
PROGRAM STUDI INFORMATIKA**

Jl. Wr. Supratman Kandang Limun, Bengkulu Bengkulu 38371 A  
Telp: (0736) 344087, 22105 - 227

**LEMBAR ASISTENSI  
LAPORAN PRAKTIKUM KE-2 PROYEK STRUKTUR DATA DAN ALGORITMA**

Nama Mahasiswa : I Nyoman Dimas Kresna Adryan  
NPM : G1A023077  
Dosen : 1. Mochammad Yusa, S.Kom., M.Kom.  
2. Arie Vatesia, S.T., M.TI, Ph.D.  
Asisten : 1. Davi Sulaiman (G1A022001)  
2. Attiya Dianti Fadli (G1A022002)  
3. Abdi Agung Kurniawan (G1A022011)  
4. Sophina Shafa Salsabila (G1A022021)  
5. Evelyn Funike Aritonang (G1A022024)  
6. Diodo Arrahman (G1A022027)  
7. Sinta Ezra Wati Gulo (G1A022040)  
8. Wahyu Ozorah Manurung (G1A022060)  
9. Alif Nurhidayat (G1A022073)  
10. Ahmad Radesta (G1A022086)

Laporan Praktikum

Catatan dan Tanda Tangan

Laporan Praktikum Ke – 2	
--------------------------	--