

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 8
QUEUE**



Disusun Oleh :

NAMA : Muhammad Azzam Satria

NIM : 103112400112

Dosen

FAHRUDIN MUKTI WIBOWO S. Kom., M. Eng

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

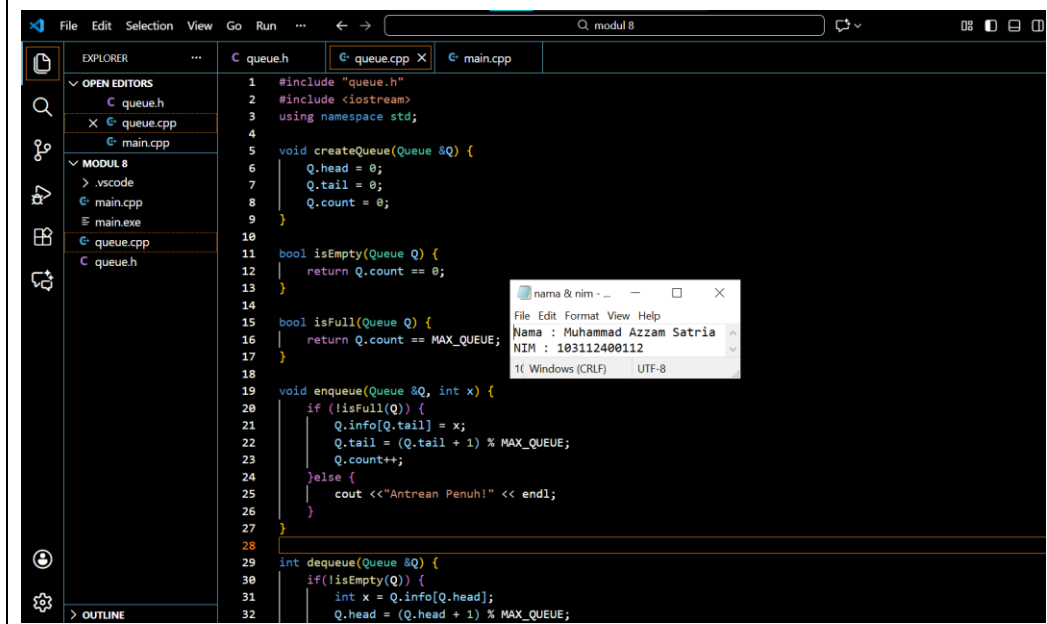
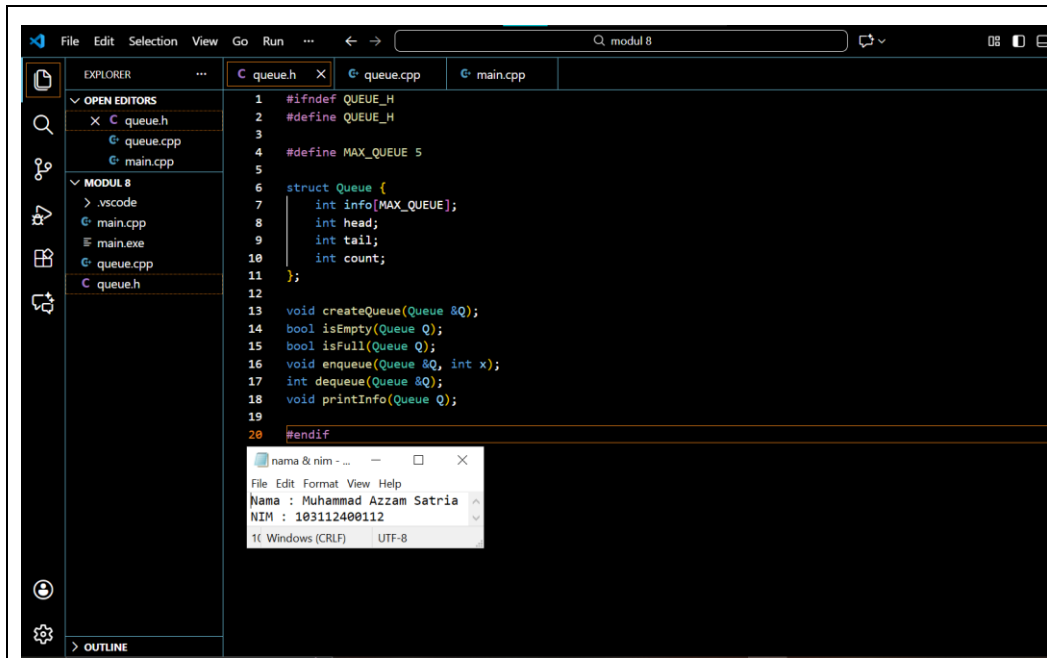
A. Dasar Teori

Queue merupakan salah satu struktur data yang cara kerjanya menerapkan prinsip FIFO (First In First Out), yaitu elemen yang pertama masuk ke dalam antrean maka akan menjadi elemen yang pertama keluar dari antrean juga. Contoh antrian dalam kehidupan di sekitar kita yaitu jalur lalu lintas satu arah di mana kendaraan yang pertama masuk ke jalur lebih dulu maka akan menjadi kendaraan pertama yang keluar jalur lebih dulu juga. Penerapan antrean dalam kinerja sistem komputer memiliki banyak kelebihan, diantaranya yaitu sistem menjadi lebih teratur dan mudah dalam melakukan pengawasan serta perbaikan.

Pada struktur data queue terdapat beberapa operasi dasar, diantaranya adalah enqueue yang berfungsi untuk menambahkan elemen ke bagian paling belakang antrian, dequeue untuk menghapus elemen yang ada di bagian depan antrean, isEmpty fungsinya untuk mengecek array antrian apakah masih terdapat tempat kosong dan isFull untuk memeriksa array antrian apakah sudah penuh.

Di dalam struktur data queue terdapat beberapa variasi bentuk, dua diantaranya yaitu linear queue dan circular queue. Linear queue memiliki dua cara kerja, yang pertama dengan menggeser elemen setiap kali melakukan penghapusan data dan yang kedua yaitu dengan memindahkan posisi head dan tail tanpa perlu menggeser elemen lain. Selanjutnya circular queue yang merupakan bentuk variasi queue yang paling efisien. Cara kerja variasi bentuk ini yaitu memungkinkan indeks bergerak memutar saat mencapai batas array, keuntungan variasi ini yaitu ruang penyimpanan dapat digunakan lebih efektif.

B. Guided



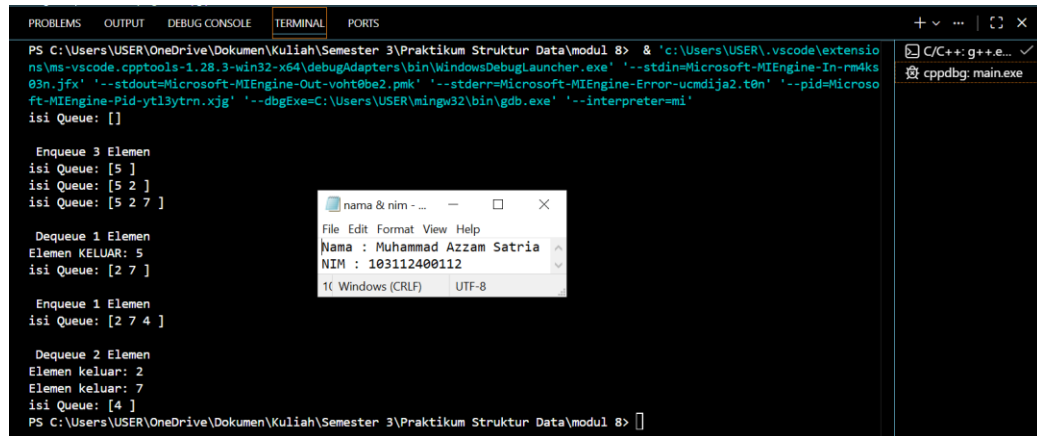
The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'EXPLORER' view shows the project structure with files: queue.h, queue.cpp, and main.cpp. The 'main.cpp' file is selected, and the code editor displays the implementation of the queue functions. The code includes a function `isEmpty()` that checks if the queue is empty by comparing the head and count. It also includes a `printInfo()` function that prints the queue's state, including the head, count, and the elements in the queue. A small window titled 'nama & nim' is visible in the foreground, displaying the student's name and NIM.

```
30 if(!isEmpty(Q)) {
31     int x = Q.info[Q.head];
32     Q.head = (Q.head + 1) % MAX_QUEUE;
33     Q.count--;
34     return x;
35 } else {
36     cout << "Antrean Kosong!" << endl;
37     return -1;
38 }
39 }
40
41 void printInfo(Queue Q) {
42     cout << "isi Queue: [";
43     if (!isEmpty(Q)) {
44         int i = Q.head;
45         int n = 0;
46         while (n < Q.count) {
47             cout << Q.info[i] << " ";
48             i = (i + 1) % MAX_QUEUE;
49             n++;
50         }
51     }
52     cout << "]" << endl;
53 }
```

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'main.cpp' file is selected, and the code editor displays the implementation of the `main()` function. The code includes the necessary headers and uses the `std` namespace. The `main()` function creates a queue, enqueues three elements (5, 2, 7), prints the queue, dequeues one element, enqueues one element (4), dequeues two elements, and prints the queue again. A small window titled 'nama & nim' is visible in the foreground, displaying the student's name and NIM.

```
1 #include <iostream>
2 #include "queue.h"
3 #include "queue.cpp"
4 using namespace std;
5
6 int main()
7 {
8     Queue Q;
9     createQueue(Q);
10    printInfo(Q);
11    cout << "\n Enqueue 3 Elemen" << endl;
12    enqueue(Q, 5);
13    printInfo(Q);
14    enqueue(Q, 2);
15    printInfo(Q);
16    enqueue(Q, 7);
17    printInfo(Q);
18
19    cout << "\n Dequeue 1 Elemen" << endl;
20    cout << "Elemen KELUAR: " << dequeue(Q) << endl;
21    printInfo(Q);
22
23    cout << "\n Enqueue 1 Elemen" << endl;
24    enqueue(Q, 4);
25    printInfo(Q);
26
27    cout << "\n Dequeue 2 Elemen" << endl;
28    cout << "Elemen keluar: " << dequeue(Q) << endl;
29    cout << "Elemen keluar: " << dequeue(Q) << endl;
30    printInfo(Q);
31    return 0;
32 }
```

Screenshots Output



```
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\modul 8> & 'c:\Users\USER\.vscode\extensions\ms-vscode.cpptools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-rm4ks03n.jfx' '--stdout=Microsoft-MIEngine-Out-voht0be2.pmk' '--stderr=Microsoft-MIEngine-Error-ucmdija2.t0n' '--pid=Microsoft-MIEngine-Pid-ytl3ytrn.xjg' '--dbgExe=C:\Users\USER\mingw32\bin\gdb.exe' '--interpreter=mi'
isi Queue: []

Enqueue 3 Elemen
isi Queue: [ 5 ]
isi Queue: [ 5 2 ]
isi Queue: [ 5 2 7 ]

Dequeue 1 Elemen
Elemen KELUAR: 5
isi Queue: [ 2 7 ]

Enqueue 1 Elemen
isi Queue: [ 2 7 4 ]

Dequeue 2 Elemen
Elemen keluar: 2
Elemen keluar: 7
isi Queue: [ 4 ]
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\modul 8>
```

nama & nim - ...

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 103112400112

1(Windows (CRLF) UTF-8

Deskripsi :

Program ini merupakan implementasi dari ADT queue yang menggunakan struktur data array statis dan menerapkan konsep circular queue. Pada program terdapat struktur queue yaitu head, tail dan count untuk mengelola elemen dalam antrian. Kemudian terdapat juga fungsi-fungsi dasar seperti createQueue untuk membuat antrian, isEmpty dan isFull untuk memeriksa antrian sebelum berjalannya operasi, enqueue untuk menambahkan elemen ke posisi tail, dequeue berfungsi mengambil elemen dari posisi head. Setiap perubahan isi antrian dapat dilihat dengan fungsi printInfo yang akan mencetak seluruh isi output.

C. Unguided 1 (head diam tail bergerak)

```
1  #ifndef QUEUE_H
2  #define QUEUE_H
3
4  const int Max = 5;
5  typedef int infotype;
6
7  struct Queue {
8      infotype info[Max];
9      int head;
10     int tail;
11 };
12
13 void createQueue(Queue &Q);
14 bool isEmptyQueue(Queue Q);
15 bool isFullQueue(Queue Q);
16 void enqueue(Queue &Q, infotype x);
17 infotype dequeue(Queue &Q);
18 void printInfo(Queue Q);
19 #endif
```

nama & nim - ...

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 10311240112

1(Windows (CRLF) UTF-8

```
1  #include <iostream>
2  #include "unguidedqueue.h"
3  using namespace std;
4
5  void createQueue(Queue &Q) {
6      Q.head = -1;
7      Q.tail = -1;
8  }
9
10 bool isEmptyQueue(Queue Q) {
11     return(Q.head == -1 && Q.tail == -1);
12 }
13 bool isFullQueue(Queue Q) {
14     return(Q.tail == Max - 1);
15 }
16
17 void enqueue(Queue &Q, infotype x) {
18     if(isFullQueue(Q)) {
19         cout << "Queue Penuh" << endl;
20     }else{
21         if(isEmptyQueue(Q)) {
22             Q.head = 0;
23             Q.tail = 0;
24         }else{
25             Q.tail++;
26         }
27         Q.info[Q.tail] = x;
28     }
29 }
30
31 infotype dequeue(Queue &Q) {
32     infotype temp;
```

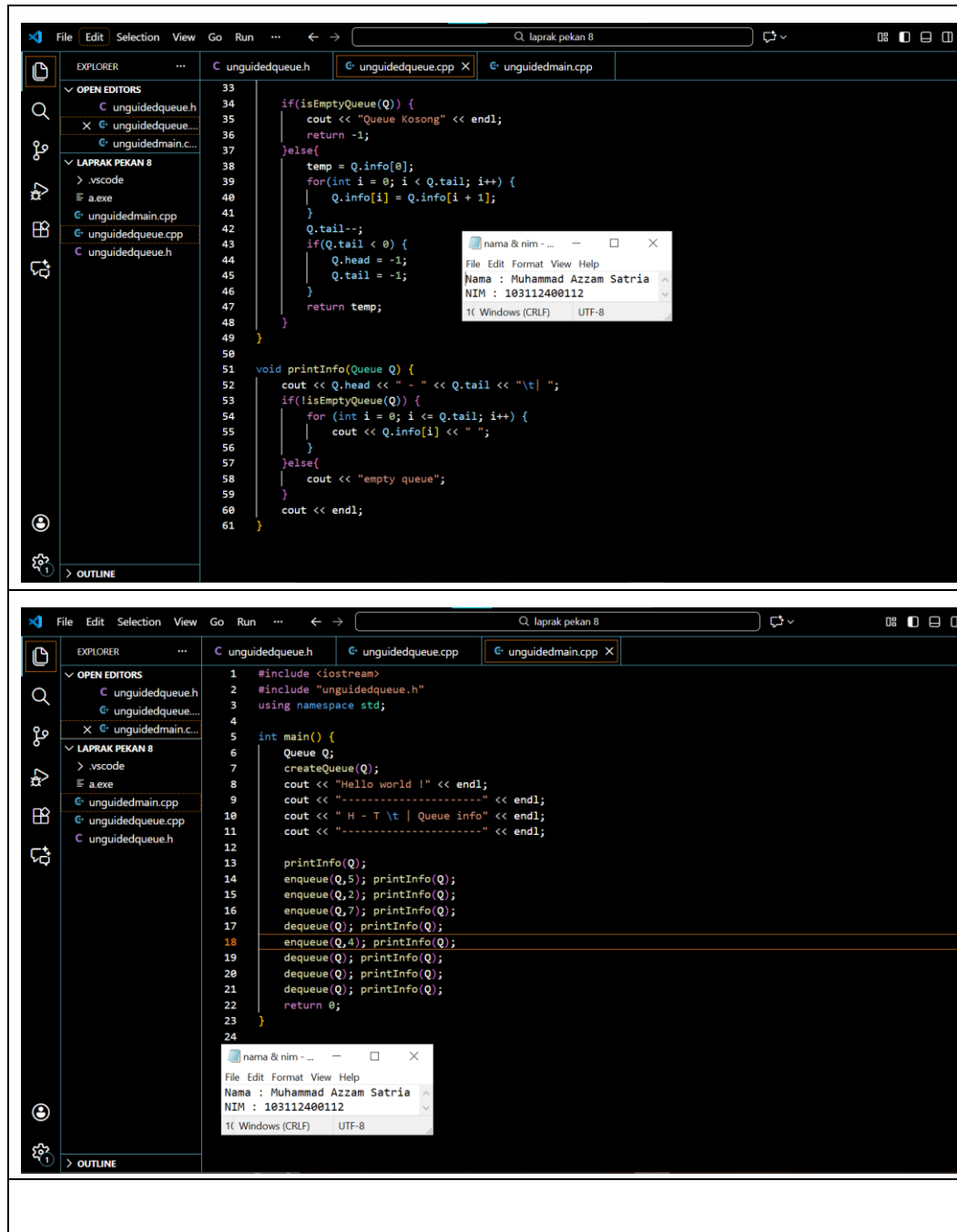
nama & nim - ...

File Edit Format View Help

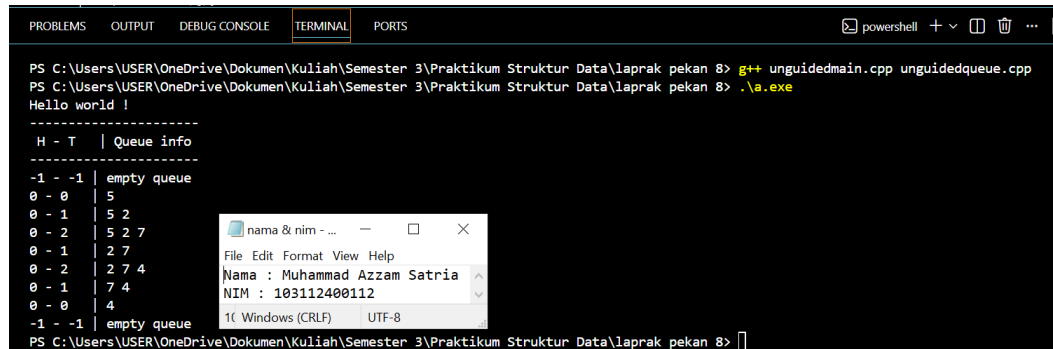
Nama : Muhammad Azzam Satria

NIM : 10311240112

1(Windows (CRLF) UTF-8



Screenshots Output



```
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> g++ unguidedmain.cpp unguidedqueue.cpp
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> .\a.exe
Hello world !

-----
H - T | Queue info
-----
-1 - -1 | empty queue
0 - 0 | 5
0 - 1 | 5 2
0 - 2 | 5 2 7
0 - 1 | 2 7
0 - 2 | 2 7 4
0 - 1 | 7 4
0 - 0 | 4
-1 - -1 | empty queue

PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8>
```

nama & nim - ...

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 103112400112

1(Windows (CRLF) UTF-8

Deskripsi:

Program ini merupakan struktur data queue dengan array berukuran tetap. Di dalamnya terdapat 2 penanda untuk mengatur program yaitu head yang menunjukkan posisi elemen paling depan dan tail yang menunjukkan posisi elemen paling akhir data. Beberapa operasi dasar yang terdapat pada program yaitu enqueue yang berfungsi untuk menambah elemen ke antrean, dequeue yang fungsinya untuk mengambil elemen dari antrean, isEmptyQueue dan isFullQueue untuk mengecek antrean sedang kosong atau penuh serta printInfo yang berfungsi untuk menampilkan isi antrean di output.

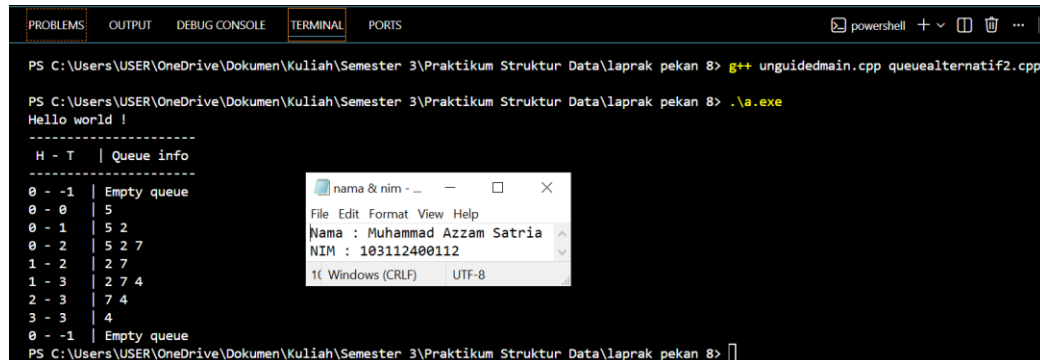
Cara kerja program yang baru dijalankan, antrean masih kosong sehingga output menampilkan (-1 -1 | kosong), ketika terdapat penambahan elemen, nilai baru akan ditempatkan di posisi paling belakang sesuai urutan masuk. Ketika elemen diambil, nilai paling depan dihapus dan semua data setelahnya digeser satu langkah ke kiri. Setelah semua elemen diambil, antrean akan kembali ke kondisi kosong seperti awal program.

Unguided 2 (head bergerak tail bergerak)


```
1 #include <iostream>
2 #include "unguidedqueue.h"
3 using namespace std;
4
5 void createQueue(Queue &Q) {
6     Q.head = 0;
7     Q.tail = -1;
8 }
9
10 bool isEmptyQueue(Queue Q) {
11     return(Q.tail < Q.head);
12 }
13 bool isFullQueue(Queue Q) {
14     return(Q.tail == Max - 1);
15 }
16
17 void enqueue(Queue &Q, infotype x) {
18     if(isFullQueue(Q)) {
19         cout << "Queue penuh" << endl;
20     }else{
21         if(isEmptyQueue(Q)) {
22             Q.head = 0;
23             Q.tail = 0;
24         }else{
25             Q.tail++;
26         }
27         Q.info[Q.tail] = x;
28     }
29 }
30
31 infotype dequeue(Queue &Q) {
32     if(isEmptyQueue(Q)) {
```

```
33     cout << "Queue kosong" << endl;
34     return -1;
35 }else{
36     infotype temp = Q.info[Q.head];
37     Q.head++;
38
39     if(Q.head > Q.tail) {
40         Q.head = 0;
41         Q.tail = -1;
42     }
43     return temp;
44 }
45
46 void printInfo(Queue Q) {
47     cout << Q.head << " - " << Q.tail << "\t| ";
48     if(!isEmptyQueue(Q)) {
49         for (int i = Q.head; i <= Q.tail; i++) {
50             cout << Q.info[i] << " ";
51         }
52     }else{
53         cout << "Empty queue";
54     }
55     cout << endl;
56 }
57 }
```

Screenshots Output

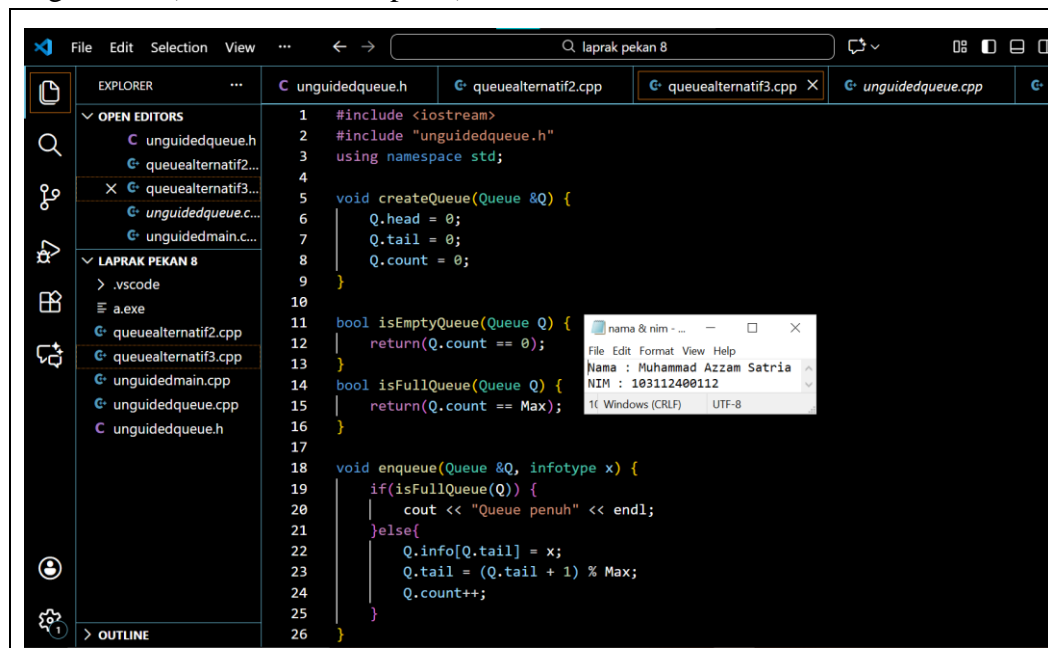


```
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> g++ unguidedmain.cpp queuealternatif2.cpp
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> .\a.exe
Hello world !
-----
H - T | Queue info
-----
0 - -1 | Empty queue
0 - 0 | 5
0 - 1 | 5 2
0 - 2 | 5 2 7
1 - 2 | 2 7
1 - 3 | 2 7 4
2 - 3 | 7 4
3 - 3 | 4
0 - -1 | Empty queue
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8>
```

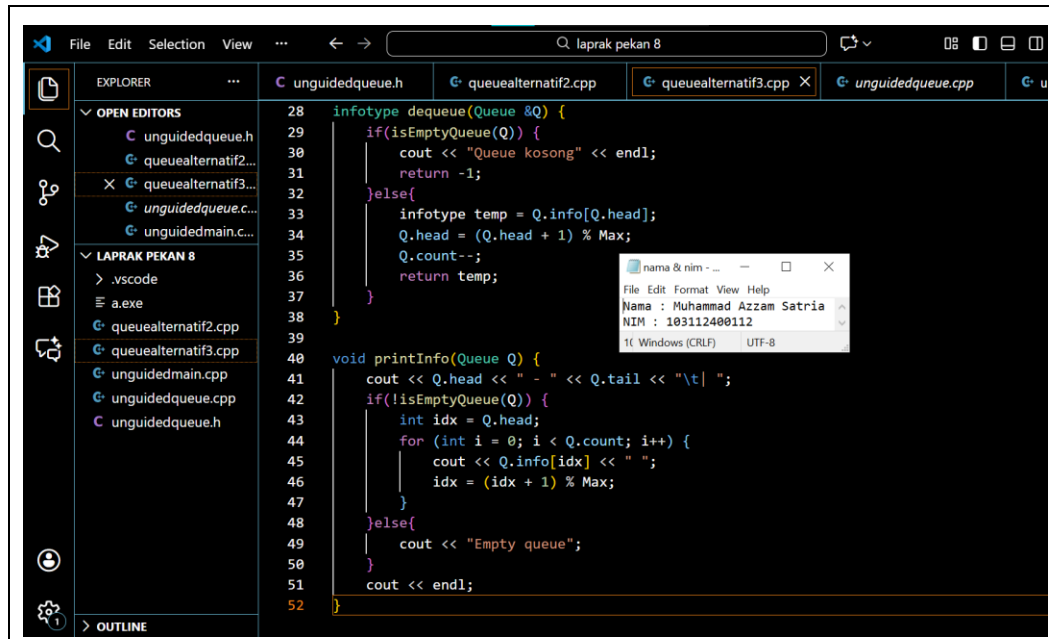
Deskripsi :

Pada alternatif 2, program dibuat supaya keduanya (head dan tail) bisa maju ke kanan tanpa perlu menggeser isi array setiap kali ada dequeuer. Setiap ada penghapusan elemen, posisi yang berubah hanya head nya saja.

Unguided 3 (head dan tail berputar)

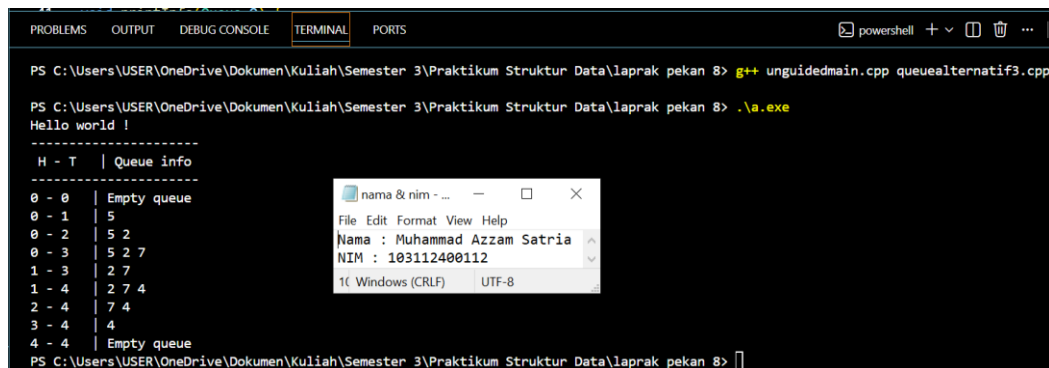


```
1 #include <iostream>
2 #include "unguidedqueue.h"
3 using namespace std;
4
5 void createQueue(Queue &Q) {
6     Q.head = 0;
7     Q.tail = 0;
8     Q.count = 0;
9 }
10
11 bool isEmptyQueue(Queue Q) {
12     return(Q.count == 0);
13 }
14 bool isFullQueue(Queue Q) {
15     return(Q.count == Max);
16 }
17
18 void enqueue(Queue &Q, infotype x) {
19     if(isFullQueue(Q)) {
20         cout << "Queue penuh" << endl;
21     }else{
22         Q.info[Q.tail] = x;
23         Q.tail = (Q.tail + 1) % Max;
24         Q.count++;
25     }
26 }
```



```
28 infotype dequeue(Queue &Q) {
29     if(isEmptyQueue(Q)) {
30         cout << "Queue kosong" << endl;
31         return -1;
32     }else{
33         infotype temp = Q.info[Q.head];
34         Q.head = (Q.head + 1) % Max;
35         Q.count--;
36         return temp;
37     }
38 }
39
40 void printInfo(Queue Q) {
41     cout << Q.head << " - " << Q.tail << "\t| ";
42     if(!isEmptyQueue(Q)) {
43         int idx = Q.head;
44         for (int i = 0; i < Q.count; i++) {
45             cout << Q.info[idx] << " ";
46             idx = (idx + 1) % Max;
47         }
48     }else{
49         cout << "Empty queue";
50     }
51     cout << endl;
52 }
```

Screenshots Output



```
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> g++ unguidedmain.cpp queuealternatif3.cpp
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> ./a.exe
Hello world !

H - T | Queue info
-----|-----
0 - 0 | Empty queue
0 - 1 | 5
0 - 2 | 5 2
0 - 3 | 5 2 7
1 - 3 | 2 7
1 - 4 | 2 7 4
2 - 4 | 7 4
3 - 4 | 4
4 - 4 | Empty queue

PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak pekan 8> 
```

Deskripsi :

Pada alternatif 3 antrean dibuat melingkar agar head dan tail berputar kembali ke indeks awal ketika mencapai ujung array.

D. Kesimpulan

Queue merupakan salah satu struktur data yang menyimpan data menggunakan prinsip FIFO yaitu data yang pertama masuk akan menjadi data yang keluar pertama juga. Melalui contoh dan latihan pada modul, saya belajar cara Membuat dan mengelola queue menggunakan array dengan beberapa

operasi dasar yaitu enqueue yang fungsinya untuk menambah elemen, dequeue yang berfungsi menghapus elemen, isEmpty untuk mengecek apakah terdapat array kosong, isFull untuk mengecek apakah array sudah penuh dan printInfo untuk menampilkan seluruh data yang ada di dalam program. Melalui modul tersebut, saya menjadi paham bagaimana cara kerja head dan tail serta mengerti bagaimana alur data dalam antrian diatur dengan terstruktur.

Pada bagian latihan saya juga diperkenalkan dengan adanya beberapa variasi queue seperti queue dengan shifting, queue tanpa shifting dan circular queue yang lebih efisien karena terdapat indeks yang bisa berputar kembali ke awal. Setiap variasi memperlihatkan bagaimana pengelolaan antrian dapat dibuat lebih efisien tergantung pada cara penerapannya. Dengan melakukan praktik secara langsung dari ketiga bentuk tersebut, saya dapat mengetahui perbedaan dan menentukan suatu jenis queue digunakan pada kondisi yang sesuai.

E. Referensi

Erkamim, E., Abdurrohim, I., Yuliyanti, S., Karim, R., Rahman, A., Admira, T. M. A., & Ridwan, A. (2024). *Buku ajar algoritma dan struktur data*. PT Sonpedia Publishing Indonesia.

Ginting, S. H. N., Effendi, H., Kumar, S., Marsisno, W., Sitanggang, Y. R. U., Anwar, K., ... & Smrti, N. N. E. (2024). *Pengantar struktur data*. Mifandi Mandiri Digital.