Laporan Praktikum 10

Struktur Data Algoritma

Materi
"Queue"

Nama :
Muhammad Azhar Rasyad

NIM :
0110217029

Program Studi :
Teknik Informatika 1

Queue

Berikut contoh program Queue dengan C++ :

```
#include <iostream>

#define max 5

using namespace std;

struct node
{
  string nama;
  node *next;
  node *prev;
} *head, *tail, *baru, *bantu;

void create(string value);
// Antrian Depan
void enqueue_belakang();
void dequeue_depan();
// Antrian Belakang
void enqueue_depan();
void dequeue_belakang();
void read();
void clear();

string value;
int counter = 0;

int main()
{
  int menu;

  menu:
  cout << "\n----- Queue Linked List -----" << endl;
  cout << "\n----- Antrian Depan -----";
  cout << "\n1. Enqueue Belakang";
  cout << "\n2. Dequeue Depan" << endl;
  cout << "\n----- Antrian Belakang -----";
  cout << "\n3. Enqueue Depan";
  cout << "\n4. Dequeue Belakang" << endl;
  cout << "\n5. Read";
  cout << "\n6. Clear";
  cout << "\n7. Exit" << endl;
  cout << "\nChoose Function : ";
  cin >> menu;

  switch(menu)
  {
```

```
    case 1:
      cout << "\nInput Nama = ";
      cin >> value;
      create(value);
      enqueue_belakang();
      goto menu;
      break;

    case 2:
      dequeue_depan();
      goto menu;
      break;

    case 3:
      cout << "\nInput Nama = ";
      cin >> value;
      create(value);
      enqueue_depan();
      goto menu;
      break;

    case 4:
      dequeue_belakang();
      goto menu;
      break;

    case 5:
      read();
      goto menu;
      break;

    case 6:
      clear();
      goto menu;
      break;

    case 7:
          cout << "\n---------- Thanks For Using The Program
----------" << endl;
      break;

    default:
      cout << "\n---------- Error No Function ----------" << endl;

      cout << "\nPress Any Key To Continue...";
      cin.ignore();
      cin.get();
      goto menu;
      break;
  }
```

```
}

void create(string value)
{
  baru = new node;
  baru -> nama = value;
  baru -> next = NULL;
  baru -> prev = NULL;
}

void enqueue_belakang()
{
  if(head == NULL)
  {
    head = baru;
    tail = baru;
    counter++;

    cout << "\n<---Input Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(counter == max)
  {
    cout << "\n<---Queue Is Full--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(head -> next == NULL)
  {
    head -> next = baru;
    tail = baru;
    tail -> prev = head;
    counter++;

    cout << "\n<---Input Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    bantu = tail;
    tail -> next = baru;
    tail = baru;
```

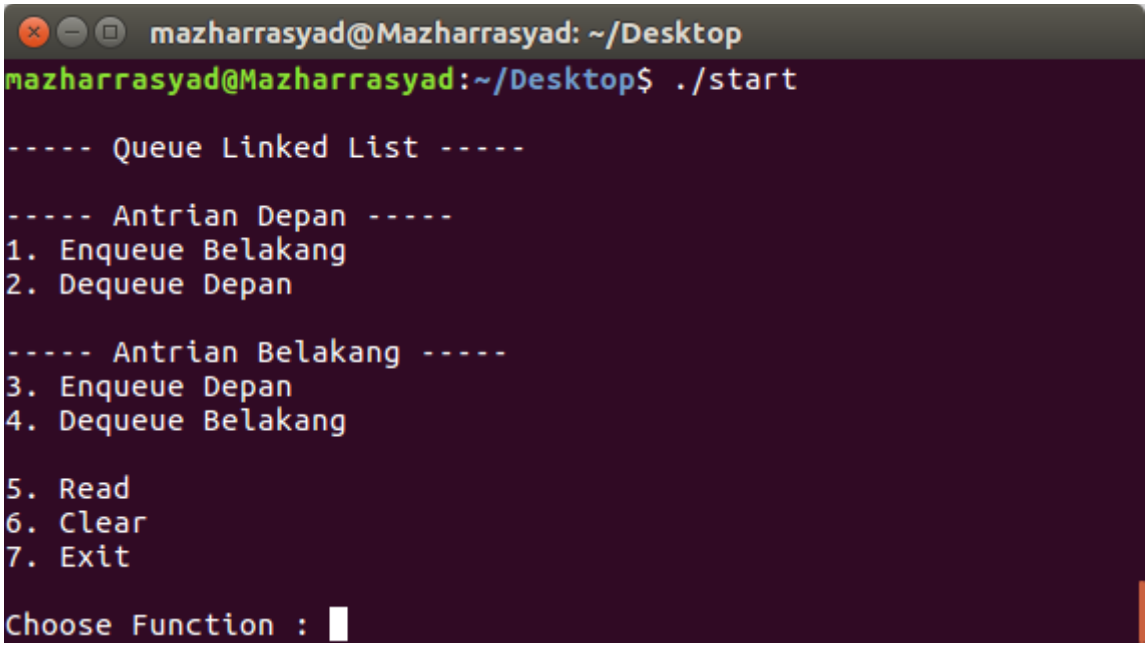
```
    tail -> prev = bantu;
    counter++;

    cout << "\n<---Input Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void dequeue_depan()
{
  if(head == NULL)
  {
    cout << "\n<---Queue Is Empty--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    bantu = head;
    head = head -> next;
    counter--;

    cout << "\n<---Delete Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void enqueue_depan()
{
  if(head == NULL)
  {
    head = baru;
    tail = baru;
    counter++;

    cout << "\n<---Input Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(counter == max)
```

```
  {
    cout << "\n<---Queue Is Full--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(head -> next == NULL)
  {
    baru -> next = head;
    head = baru;
    tail -> prev = head;
    counter++;

    cout << "\n<---Input Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    bantu = head;
    baru -> next = head;
    head = baru;
    bantu -> prev = head;
    counter++;

    cout << "\n<---Input Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void dequeue_belakang()
{
  if(head == NULL)
  {
    cout << "\n<---Queue Is Empty--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    tail = tail -> prev;
    tail -> next = NULL;
```

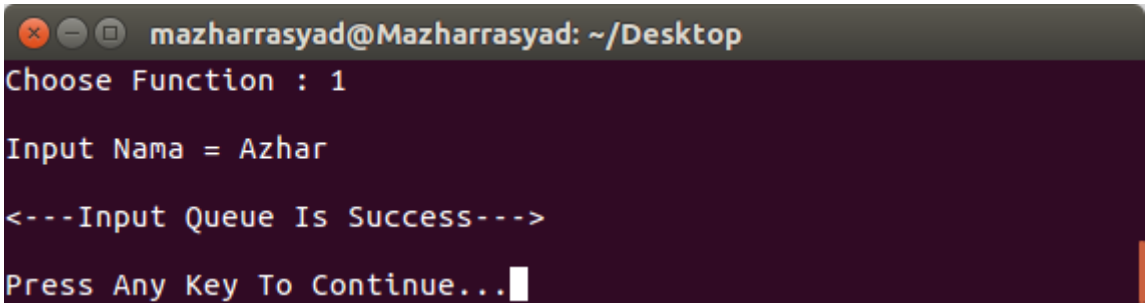
```
    counter--;

    cout << "\n<---Delete Queue Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void read()
{
  if(head == NULL)
  {
    cout << "\n<---Queue Is Empty--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(head -> next == NULL)
  {
     cout << "\nNama Queue : " << "|" << tail -> nama << "|" <<
endl;
    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    bantu = head;
    cout << "\nNama Queue : ";
    while(bantu != NULL)
    {
      cout << "|" << bantu -> nama;
      if(bantu -> next == NULL)
      {
        cout << "|";
      }
      else
      {
        cout << "| <-> ";
      }
      bantu = bantu -> next;
    }
    cout << endl;
    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
```

```
}

void clear()
{
  if(head == NULL)
  {
    cout << "\n<---Queue Is Empty--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    head = NULL;
    tail = NULL;
    counter = 0;

    cout << "\n<---Queue Is Clear--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}
```

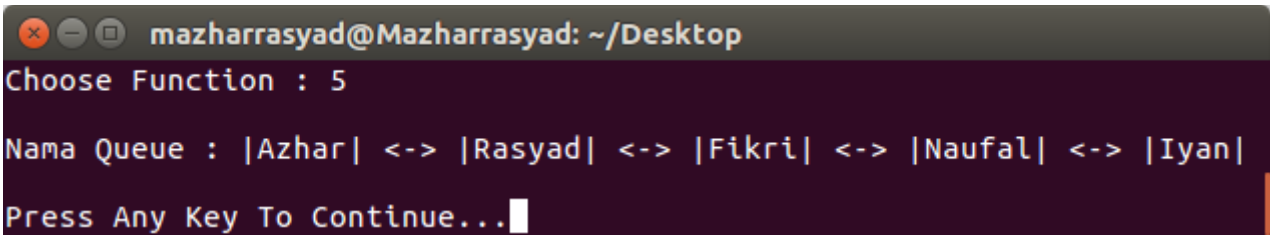
Berikut penjelasan dari program Queue diatas :



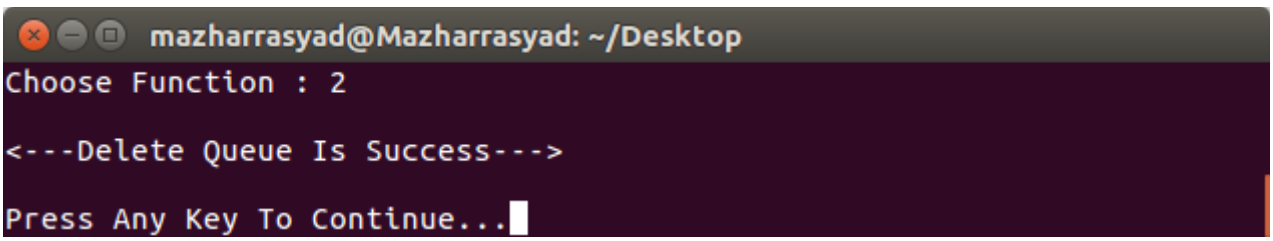
Tampilan diatas merupakan menu program queue.



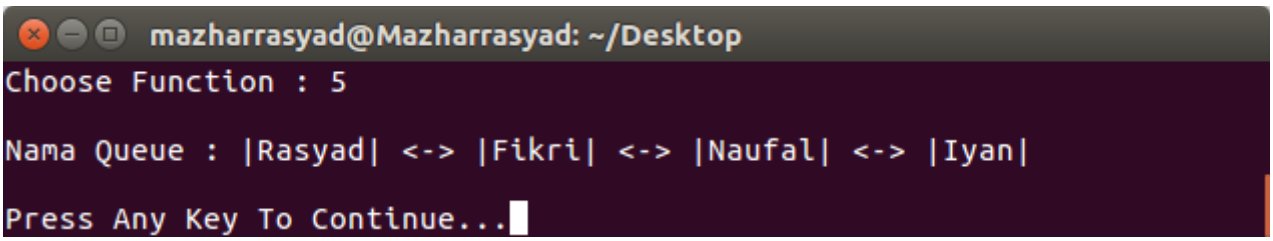
Tampilan diatas merupakan fungsi enqueue belakang.



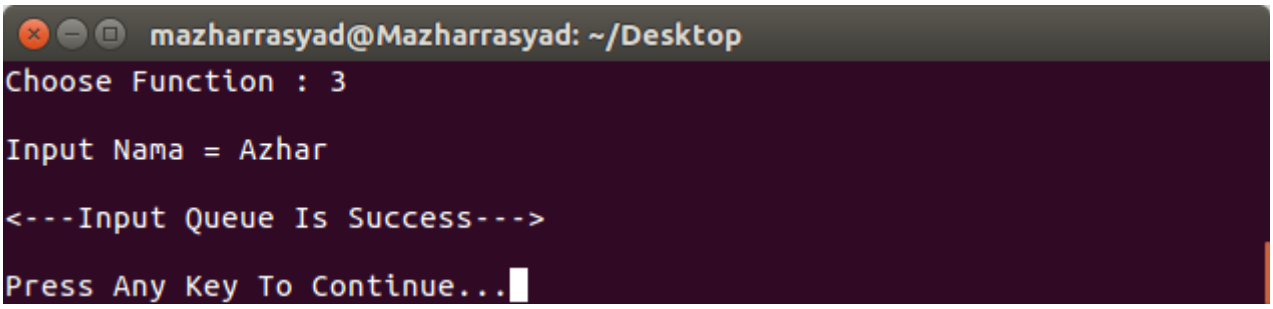
Tampilan diatas merupakan hasil dari fungsi enqueue belakang.



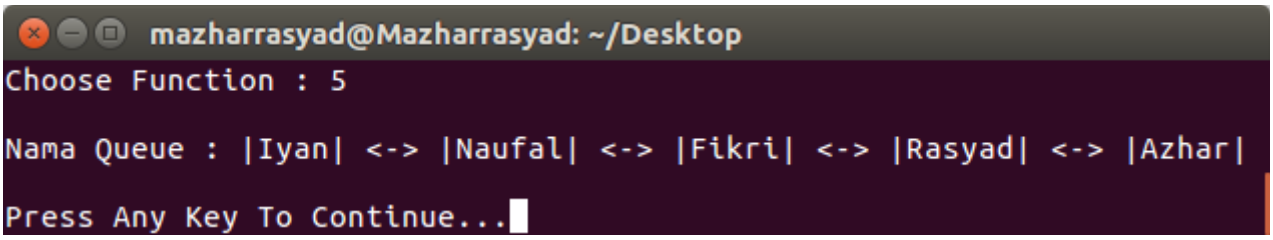
Tampilan diatas merupakan fungsi dequeue depan.



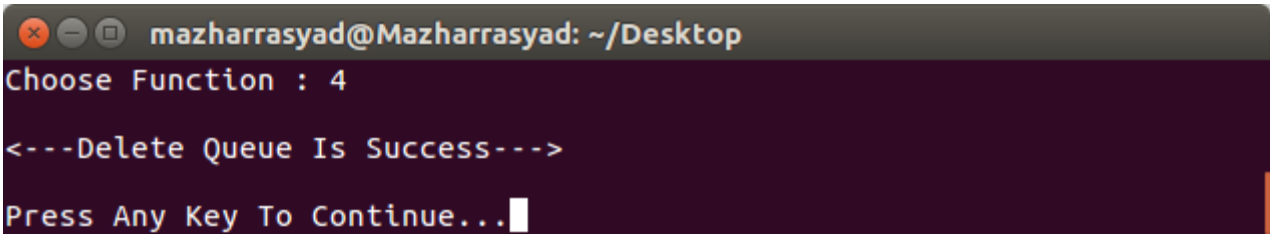
Tampilan diatas merupakan hasil dari fungsi dequeue belakang.



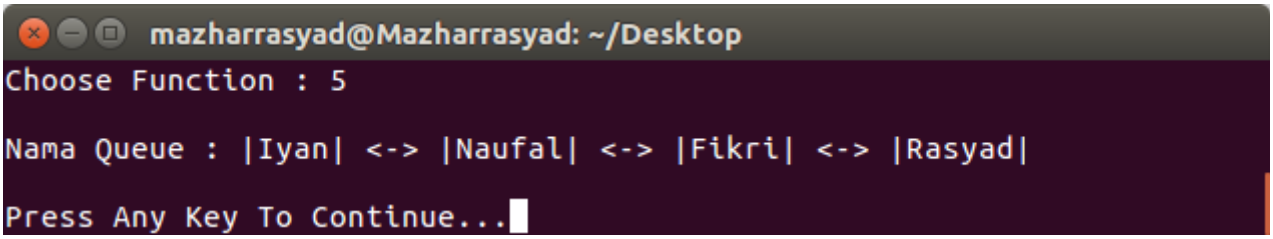
Tampilan diatas merupakan fungsi enqueue depan.



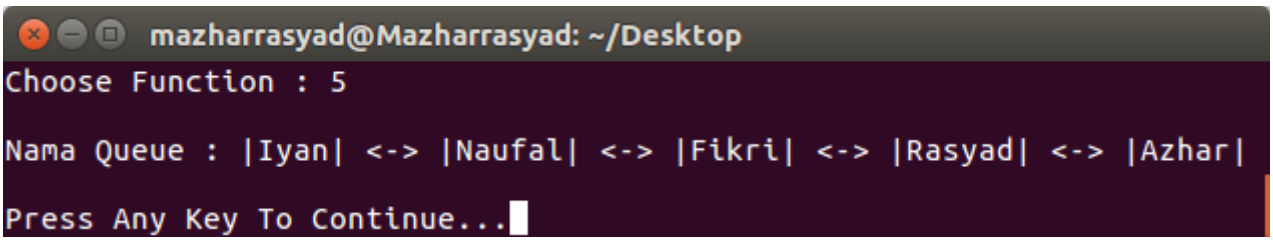
Tampilan diatas merupakan hasil dari fungsi enqueue depan.



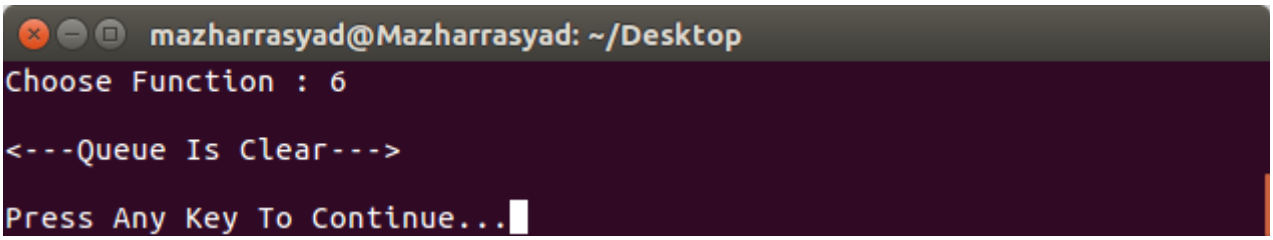
Tampilan diatas merupakan fungsi dequeue belakang.



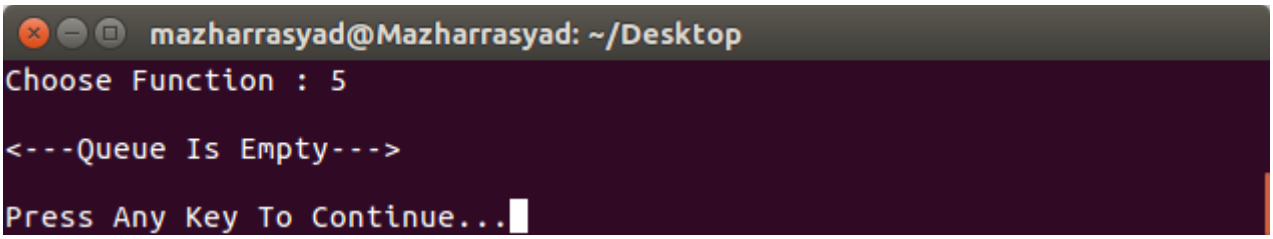
Tampilan diatas merupakan hasil dari fungsi dequeue belakang.



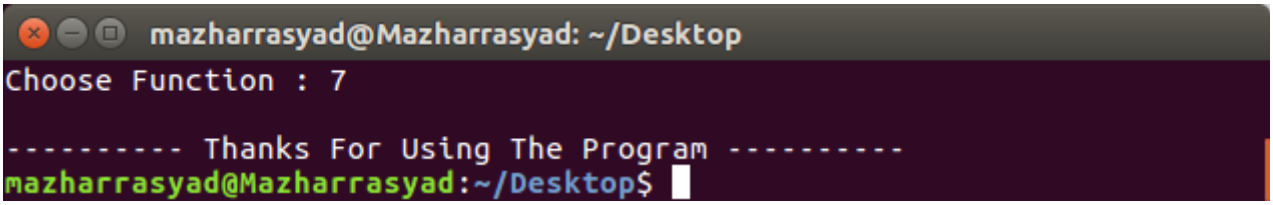
Tampilan diatas merupakan fungsi read.



Tampilan diatas merupakan fungsi clear.



Tampilan diatas merupakan hasil dari fungsi clear.



Tampilan diatas merupakan hasil dari fungsi exit.

Kesimpulan

Queue merupakan kumpulan data yang tersusun secara berurutan serta memiliki satu ujung keluar dan satu ujung masuk sehingga data yang duluan masuk dan keluar duluan dan data yang masuk belakangan akan keluar belakangan.

- - - - - - - - - - Sekian - - - - - - - - - -