

# **STRUKTUR DATA**

## **Latihan Queue**



**Nama : Alfrandiano Imanuel Rionaldo Holle**

**NPM : 140810247001**

**Dikumpulkan Tanggal:**

**20 Mei 2025**

**PROGRAM STUDI S-1 TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**JATINANGOR**

**2025**

1. Source cpp (Queue List) :

```
#include <iostream>
#include <iomanip>

struct Barang
{
    std :: string kode_brg;
    std :: string nama_brg;
    unsigned int harga;
    unsigned int banyak;
};

struct Node
{
    Barang info;
    Node* next;
};

typedef Node* Ptr;

struct Queue
{
    Ptr head;
    Ptr tail;
};

void createQueue(Queue &q)
{
    q.head = nullptr;
    q.tail = nullptr;
}

bool isEmpty(Queue &q)
{
    return q.head == nullptr;
}

Ptr createNode()
{
    Ptr newNode = new Node;
    std :: cout << "Masukkan Kode Barang : ";
    std :: cin >> newNode->info.kode_brg;
    std :: cout << "Masukkan Nama Barang : ";
    std :: cin.ignore();
    getline(std :: cin, newNode->info.nama_brg);
    std :: cout << "Masukkan Harga Barang : ";
    std :: cin >> newNode->info.harga;
    std :: cout << "Masukkan Banyak Barang: ";
    std :: cin >> newNode->info.banyak;
    newNode->next = nullptr;
    return newNode;
}

void insertQueue(Queue &q, Ptr newNode)
{
    if (isEmpty(q))
    {
        q.head = q.tail = newNode;
    }
}
```

```

        else
        {
            q.tail->next = newNode;
            q.tail = newNode;
        }

        std :: cout << newNode->info.nama_brg << " dimasukkan ke queue.\n";
    }
}

void deleteQueue(Queue &q)
{
    if (isEmpty(q))
    {
        std :: cout << "Queue kosong!\n";
        return;
    }

    if (q.head == q.tail)
    {
        std :: cout << q.head->info.nama_brg << " dikeluarkan dari queue (1
elemen).\n";
        delete q.head;
        q.head = q.tail = nullptr;
    }
    else
    {
        Ptr temp = q.head;
        std :: cout << temp->info.nama_brg << " dikeluarkan dari queue.\n";
        q.head = q.head->next;
        delete temp;
    }
}

void display(Queue &q)
{
    if (isEmpty(q))
    {
        std :: cout << "Queue kosong!\n";
        return;
    }

    std :: cout << "\nDaftar Barang PT Informatika\n";
    std :: cout << "-----\n";
    std :: cout << std :: left << std :: setw(6) << "No"
        << std :: setw(12) << "Kode"
        << std :: setw(35) << "Nama Barang"
        << std :: setw(15) << "Harga"
        << std :: setw(12) << "Banyak"
        << std :: setw(15) << "Jumlah" << '\n';
    std :: cout << "-----\n";

    int i = 1;
    Ptr temp = q.head;
    while (temp != nullptr)
    {
        std :: cout << std :: left << std :: setw(6) << i++
            << std :: setw(12) << temp->info.kode_brg
            << std :: setw(35) << temp->info.nama_brg

```

```

        << std :: setw(15) << temp->info.harga
        << std :: setw(12) << temp->info.banyak
        << std :: setw(15) << temp->info.harga * temp->info.banyak <<
'\n';
        temp = temp->next;
    }

    std :: cout << "-----\n";
}

void menu(Queue &q)
{
    int pilihan;
    do
    {
        std :: cout << "\n=== MENU QUEUE BARANG ===\n";
        std :: cout << "1. Tambah Barang (Enqueue)\n";
        std :: cout << "2. Hapus Barang (Dequeue)\n";
        std :: cout << "3. Tampilkan Queue\n";
        std :: cout << "4. Keluar\n";
        std :: cout << "Pilih: ";
        std :: cin >> pilihan;

        switch (pilihan)
        {
            case 1:
            {
                Ptr node = createNode();
                insertQueue(q, node);
                break;
            }
            case 2:
                deleteQueue(q);
                break;
            case 3:
                display(q);
                break;
            case 4:
                std :: cout << "Program selesai.\n";
                break;
            default:
                std :: cout << "Pilihan tidak valid.\n";
        }
    } while (pilihan != 4);
}

int main()
{
    Queue q;
    createQueue(q);
    menu(q);
    return 0;
}
2.

```

Output Program :

- Insert Queue :

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 1
Masukkan Kode Barang : 001
Masukkan Nama Barang : Pensil
Masukkan Harga Barang : 3000
Masukkan Banyak Barang: 5
Pensil dimasukkan ke queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3

Daftar Barang PT Informatika
-----
No  Kode  Nama Barang      Harga  Banyak  Jumlah
-----
1   001    Pensil           3000   5        15000
-----
```

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 1
Masukkan Kode Barang : 002
Masukkan Nama Barang : Kado
Masukkan Harga Barang : 5000
Masukkan Banyak Barang: 2
Kado dimasukkan ke queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3

Daftar Barang PT Informatika
-----
No  Kode  Nama Barang      Harga  Banyak  Jumlah
-----
1   001    Pensil           3000   5        15000
2   002    Kado             5000   2        10000
-----
```

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 1
Masukkan Kode Barang : 003
Masukkan Nama Barang : Tas
Masukkan Harga Barang : 50000
Masukkan Banyak Barang: 4
Tas dimasukkan ke queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3

Daftar Barang PT Informatika
-----
No  Kode  Nama Barang      Harga  Banyak  Jumlah
-----
1   001    Pensil           3000   5        15000
2   002    Kado             5000   2        10000
3   003    Tas              50000   4       200000
-----
```

- Delete Queue

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 2
Pensil dikeluarkan dari queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3

Daftar Barang PT Informatika
-----
No  Kode  Nama Barang      Harga  Banyak  Jumlah
-----
1   002    Kado             5000   2        10000
2   003    Tas              50000   4       200000
-----

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 1
Masukkan Kode Barang : 004
Masukkan Nama Barang : Panci
Masukkan Harga Barang : 10000
Masukkan Banyak Barang: 5
Panci dimasukkan ke queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3

Daftar Barang PT Informatika
-----
No  Kode  Nama Barang      Harga  Banyak  Jumlah
-----
1   002    Kado             5000   2        10000
2   003    Tas              50000   4       200000
3   004    Panci            10000   5        50000
-----
```

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 2
Kado dikeluarkan dari queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3

Daftar Barang PT Informatika
-----
No  Kode  Nama Barang      Harga  Banyak  Jumlah
-----
1   003    Tas              50000   4       200000
2   004    Panci            10000   5        50000
-----
```

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 2
Tas dikeluarkan dari queue.

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 2
Panci dikeluarkan dari queue (1 elemen).

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 2
Queue kosong!

=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 3
Queue kosong!
```

- Exit

```
=== MENU QUEUE BARANG ===
1. Tambah Barang (Enqueue)
2. Hapus Barang (Dequeue)
3. Tampilkan Queue
4. Keluar
Pilih: 4
Program selesai.
```