STRUKTUR DATA
Tugas Queue

NAMA: Bagas Diatama Wardoyo
NPM: 140810230061

Dikumpulkan tanggal :
19 Mei 2024

Universitas Padjadjaran
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
Program Studi S-1 Teknik Informatika
2024

Queue_Array

- **Source Code**

```
/* Nama program : queue_array
   Nama         : Bagas Diatama Wardoyo
   NPM          : 140810230061
   Tanggal buat : 13/05/2024
   Deskripsi    : queue (Array)
**********************************************************************/
#include <iostream>
#include <string>

using namespace std;

const int maxElement = 255;

struct pegawai
{
    string nama;
    string NIP;
    int golongan;
};

struct Queue
{
    pegawai data[maxElement];
    int head;
    int tail;
};

Queue Q;

void createQueue(Queue &Q);
void createElement(pegawai &data);
int gajiGolongan(int golongan);
int average(Queue Q);
int MAXGaji(Queue Q);
int MINGaji(Queue Q);
void inQueue(Queue &Q, pegawai data);
void deQueue(Queue &Q, pegawai &deletePegawai);
void traversal(Queue Q);

int main()
{
    pegawai newPegawai, deletePegawai;

    createQueue(Q);

    int choice;

    do
    {
        cout << "\n=================Main Menu=================\n";
        cout << "1. Tambah Pegawai\n";
        cout << "2. Tampilkan Data Pegawai dan Rata-rata Gaji\n";
        cout << "3. Hapus Data Pegawai\n";
        cout << "4. Keluar\n";
        cout << "Masukkan pilihan Anda: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            createElement(newPegawai);
            inQueue(Q, newPegawai);
            cout << "Pegawai berhasil ditambahkan!\n";
            break;
        case 2:
            traversal(Q);
            break;
        case 3:
            deQueue(Q, deletePegawai);
            cout << "Pegawai dengan NIP " << deletePegawai.NIP << " telah 
dihapus!\n";
            break;
        case 4:
            cout << "Program Keluar\n";
            break;
        default:
            cout << "Pilihan tidak valid. Silakan coba lagi.\n";
            break;
        }
    } while (choice != 4);

    return 0;
}

void createQueue(Queue &Q)
{
    Q.head = 0;
    Q.tail = -1;
}

void createElement(pegawai &data)
{
    cout << "NIP          :    ";
    cin >> data.NIP;
    cout << "Nama         :    ";
    cin >> data.nama;
    cout << "Golongan     :    ";
    cin >> data.golongan;
}

int gajiGolongan(int golongan)
```

```
{
    int gaji = 0;
    switch (golongan)
    {
    case 1:
        gaji = 3000000;
        break;
    case 2:
        gaji = 4000000;
        break;
    case 3:
        gaji = 5000000;
        break;
    default:
        gaji = 0;
        break;
    }
    return gaji;
}

int average(Queue Q)
{
    if (Q.tail == -1)
    {
        return -1;
    }
    else
    {

        int banyakData = Q.tail + 1;
        int totalGaji = 0;
        for (int i = 0; i < banyakData; i++)
        {
            totalGaji += gajiGolongan(Q.data[i].golongan);
        }
        return totalGaji / banyakData;
    }
}

int MAXGaji(Queue Q)
{
    if (Q.tail == -1)
    {
        return -1;
    }
    else
    {
        int Max = 0;
        int gaji;
        int banyakData = Q.tail + 1;
        for (int i = 0; i < banyakData; i++)
        {
            gaji = gajiGolongan(Q.data[i].golongan);
            if (Max < gaji)
            {
                Max = gaji;
            }
        }

        return Max;
    }
}

int MINGaji(Queue Q)
{
    if (Q.tail == -1)
    {
        return -1;
    }
    else
    {
        int Min = INT32_MAX;
        int gaji;
        int banyakData = Q.tail + 1;
        for (int i = 0; i < banyakData; i++)
        {
            gaji = gajiGolongan(Q.data[i].golongan);
            if (Min > gaji)
            {
                Min = gaji;
            }
        }

        return Min;
    }
}

void inQueue(Queue &Q, pegawai data)
{
    if (Q.tail == maxElement - 1)
    {
        cout << "List sudah penuh" << endl;
        return;
    }
    else
    {
        Q.tail = Q.tail + 1;
        Q.data[Q.tail] = data;
    }
}

void deQueue(Queue &Q, pegawai &deletePegawai)
{
    if (Q.tail == -1)
    {
        cout << "Tidak ada pegawai yang dapat dihapus" << endl;
    }
    else
    {
        deletePegawai = Q.data[Q.tail];
        for (int i = 0; i < Q.tail; i++)
        {
            Q.data[i] = Q.data[i + 1];
        }
        Q.tail--;
    }
}

void traversal(Queue Q)
{
    int banyakElement = Q.tail + 1;
    int n = 0;
    int gaji = 0;
    cout << "======================================================" << endl;
    cout << "NO\tNIP\tNama\tGolongan\tGaji" << endl;
    cout << "======================================================" << endl;
    for (int i = 0; i < banyakElement; i++)
    {
        n++;
        gaji = gajiGolongan(Q.data[i].golongan);
        cout << n << "\t" << Q.data[i].NIP << "\t" << Q.data[i].nama << "\t"
<< Q.data[i].golongan << "\t\t" << gaji << endl;
    }
    cout << "======================================================" << endl;
    cout << "Rata-rata Gaji  :  " << average(Q) << endl;
    cout << "Gaji Tertinggi  :  " << MAXGaji(Q) << endl;
    cout << "Gaji Terendah   :  " << MINGaji(Q) << endl;
}
```

- **Hasil Run**

```
======================================================
NO      NIP     Nama    Golongan        Gaji
======================================================
1       001     A       1               3000000
2       002     B       2               4000000
3       003     C       3               5000000
4       004     D       3               5000000
5       005     E       3               5000000
======================================================
Rata-rata Gaji  :  4400000
Gaji Tertinggi  :  5000000
Gaji Terendah   :  3000000
```

```
=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 3
Pegawai dengan NIP 001 telah dihapus!

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 3
Pegawai dengan NIP 002 telah dihapus!

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 2
======================================================
NO      NIP     Nama    Golongan        Gaji
======================================================
1       003     C       3               5000000
2       004     D       3               5000000
3       005     E       3               5000000
======================================================
Rata-rata Gaji  :  5000000
Gaji Tertinggi  :  5000000
Gaji Terendah   :  5000000
```

Queue_ArrayCircular

- **Source Code**

```
/* Nama program : queue_array
   Nama         : Bagas Diatama Wardoyo
   NPM          : 140810230061
   Tanggal buat : 13/05/2024
   Deskripsi    : queue (Array Circular)
**********************************************************************/
#include <iostream>
#include <string>

using namespace std;

const int maxElement = 5;

struct pegawai
{
    string nama;
    string NIP;
    int golongan;
};

struct Queue
{
    pegawai data[maxElement];
    int head;
    int tail;
};

Queue Q;

void createQueue(Queue &Q);
void createElement(pegawai &data);
int gajiGolongan(int golongan);
int average(Queue Q);
int MAXGaji(Queue Q);
int MINGaji(Queue Q);
void inQueue(Queue &Q, pegawai data);
void deQueue(Queue &Q, pegawai &deletePegawai);
void traversal(Queue Q);
bool isFull(Queue Q);
bool isEmpty(Queue Q);

int main()
{
    pegawai newPegawai, deletePegawai;

    createQueue(Q);

    int choice;

    do
    {
        cout << "\n================Main Menu================\n";
        cout << "1. Tambah Pegawai\n";
        cout << "2. Tampilkan Data Pegawai dan Rata-rata Gaji\n";
        cout << "3. Hapus Data Pegawai\n";
        cout << "4. Keluar\n";
        cout << "Masukkan pilihan Anda: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            createElement(newPegawai);
            inQueue(Q, newPegawai);
            cout << "Pegawai berhasil ditambahkan!\n";
            break;
        case 2:
            traversal(Q);
            break;
        case 3:
            deQueue(Q, deletePegawai);
            if (deletePegawai.NIP != "")
                cout << "Pegawai dengan NIP " << deletePegawai.NIP << " telah
dihapus!\n";
            break;
        case 4:
            cout << "Program Keluar\n";
            break;
        default:
            cout << "Pilihan tidak valid. Silakan coba lagi.\n";
            break;
        }
    } while (choice != 4);

    return 0;
}

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

void createElement(pegawai &data)
{
    cout << "NIP          : ";
    cin >> data.NIP;
    cout << "Nama         : ";
    cin >> data.nama;
    cout << "Golongan     : ";
    cin >> data.golongan;
}

int gajiGolongan(int golongan)
{
    int gaji = 0;
    switch (golongan)
    {
    case 1:
        gaji = 3000000;
        break;
    case 2:
        gaji = 4000000;
        break;
    case 3:
        gaji = 5000000;
        break;
    default:
        gaji = 0;
        break;
    }
    return gaji;
}

int average(Queue Q)
{
    if (isEmpty(Q))
    {
        return -1;
    }
    else
    {
        int totalGaji = 0;
        int count = 0;
        for (int i = Q.head; i != Q.tail; i = (i + 1) % maxElement)
        {
            totalGaji += gajiGolongan(Q.data[i].golongan);
            count++;
        }
        totalGaji += gajiGolongan(Q.data[Q.tail].golongan); // Add the last element
        count++;
        return totalGaji / count;
    }
}

int MAXGaji(Queue Q)
{
    if (isEmpty(Q))
    {
        return -1;
    }
    else
    {
        int Max = 0;
        for (int i = Q.head; i != Q.tail; i = (i + 1) % maxElement)
        {
            int gaji = gajiGolongan(Q.data[i].golongan);
            if (Max < gaji)
            {
                Max = gaji;
            }
        }
        int gaji = gajiGolongan(Q.data[Q.tail].golongan);
        if (Max < gaji)
        {
            Max = gaji;
        }
        return Max;
    }
}

int MINGaji(Queue Q)
{
    if (isEmpty(Q))
    {
        return -1;
    }
    else
    {
        int Min = INT32_MAX;
        for (int i = Q.head; i != Q.tail; i = (i + 1) % maxElement)
        {
            int gaji = gajiGolongan(Q.data[i].golongan);
            if (Min > gaji)
            {
                Min = gaji;
            }
        }
        int gaji = gajiGolongan(Q.data[Q.tail].golongan);
        if (Min > gaji)
        {
            Min = gaji;
        }
        return Min;
    }
}

void inQueue(Queue &Q, pegawai data)
{
    if (isFull(Q))
    {
        cout << "List sudah penuh" << endl;
        return;
    }
    if (isEmpty(Q))
    {
        Q.head = 0;
        Q.tail = 0;
        Q.data[Q.tail] = data;
    }
    else
    {
        Q.tail = (Q.tail + 1) % maxElement;
        Q.data[Q.tail] = data;
    }
}

void deQueue(Queue &Q, pegawai &deletePegawai)
{
    if (isEmpty(Q))
    {
        cout << "Tidak ada pegawai yang dapat dihapus" << endl;
        deletePegawai.NIP = "";
    }
    else if (Q.head == Q.tail)
    {
        deletePegawai = Q.data[Q.head];
        Q.head = -1;
        Q.tail = -1;
    }
    else
    {
        deletePegawai = Q.data[Q.head];
        Q.head = (Q.head + 1) % maxElement;
    }
}

void traversal(Queue Q)
{
    if (isEmpty(Q))
    {
        cout << "Tidak ada data pegawai" << endl;
        return;
    }

    int n = 0;
    int gaji = 0;
    cout << "====================================================" << endl;
    cout << "NO\tNIP\tNama\tGolongan\tGaji" << endl;
    cout << "====================================================" << endl;
    for (int i = Q.head; i != Q.tail; i = (i + 1) % maxElement)
    {
        n++;
        gaji = gajiGolongan(Q.data[i].golongan);
        cout << n << "\t" << Q.data[i].NIP << "\t" << Q.data[i].nama << "\t"
<< Q.data[i].golongan << "\t\t" << gaji << endl;
    }
    n++;
    gaji = gajiGolongan(Q.data[Q.tail].golongan);
    cout << n << "\t" << Q.data[Q.tail].NIP << "\t" << Q.data[Q.tail].nama <<
"\t" << Q.data[Q.tail].golongan << "\t\t" << gaji << endl;
```

```
    cout << "======================================================" << endl;
    cout << "Rata-rata Gaji  :  " << average(Q) << endl;
    cout << "Gaji Tertinggi  :  " << MAXGaji(Q) << endl;
    cout << "Gaji Terendah   :  " << MINGaji(Q) << endl;
}

bool isFull(Queue Q)
{
    return (Q.tail + 1) % maxElement == Q.head;
}

bool isEmpty(Queue Q)
{
    return Q.head == -1;
}
```

- **Hasil Run**

```
======================================================
NO      NIP     Nama    Golongan        Gaji
======================================================
1       001     A       1               3000000
2       002     B       2               4000000
3       003     C       3               5000000
4       004     D       3               5000000
5       005     E       3               5000000
======================================================
Rata-rata Gaji  :  4400000
Gaji Tertinggi  :  5000000
Gaji Terendah   :  3000000

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 3
Pegawai dengan NIP 001 telah dihapus!

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
```

```
Masukkan pilihan Anda: 3
Pegawai dengan NIP 002 telah dihapus!

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 2
======================================================
NO      NIP     Nama    Golongan        Gaji
======================================================
1       003     C       3               5000000
2       004     D       3               5000000
3       005     E       3               5000000
======================================================
Rata-rata Gaji  :  5000000
Gaji Tertinggi  :  5000000
Gaji Terendah   :  5000000
```

Queue_List

- **Source Code**

```
/* Nama program : queue_list
   Nama         : Bagas Diatama Wardoyo
   NPM          : 140810230061
   Tanggal buat : 13/05/2024
   Deskripsi    : queue (Linked List)
*********************************************************************/

#include <iostream>
#include <string>

using namespace std;

struct pegawai
{
    string NIP;
    string nama;
    int golongan;
};

struct node
{
    pegawai data;
    node *next;
};

typedef node *Pointer;
typedef Pointer List;

struct Queue
{
    List Head;
    List Tail;
};

Queue Q;

void createList(Queue &Q);
void createElement(List &newNode);
void inQueue(Queue &Q, Pointer newNode);
void deQueue(Queue &Q, Pointer &pHapus);
int gajiGolongan(int golongan);
int average(Queue Q);
int MAXGaji(Queue Q);
int MINGaji(Queue Q);
void traversal(Queue Q);

int main()
{
    Pointer pHapus;
    List newNode;

    createList(Q);

    int choice;

    do
    {
        cout << "\n==============Main Menu==============\n";
        cout << "1. Tambah Pegawai\n";
        cout << "2. Tampilkan Data Pegawai dan Rata-rata Gaji\n";
        cout << "3. Hapus Data Pegawai\n";
        cout << "4. Keluar\n";
        cout << "Masukkan pilihan Anda: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            createElement(newNode);
            inQueue(Q, newNode);
            cout << "Pegawai berhasil ditambahkan!\n";
            break;
        case 2:
            traversal(Q);
            break;
        case 3:
            deQueue(Q, pHapus);
            cout << "Pegawai dengan NIP " << pHapus->data.NIP << " telah
dihapus!\n";
            break;
        case 4:
            cout << "Program Keluar\n";
            break;
        default:
            cout << "Pilihan tidak valid. Silakan coba lagi.\n";
            break;
        }
    } while (choice != 4);

    return 0;
}

void createList(Queue &Q)
{
    Q.Head = nullptr;
    Q.Tail = nullptr;
}

void createElement(List &newNode)
{
    newNode = new node;
    cout << "NIP          :    ";
    cin >> newNode->data.NIP;
    cout << "Nama         :    ";
    cin >> newNode->data.nama;
    cout << "Golongan     :    ";
    cin >> newNode->data.golongan;
    newNode->next = nullptr;
}

void inQueue(Queue &Q, Pointer newNode)
{
    if (Q.Head == nullptr)
    {
        Q.Head = newNode;
        Q.Tail = newNode;
    }
    else
    {
        Q.Tail->next = newNode;
        Q.Tail = newNode;
    }
}

void deQueue(Queue &Q, Pointer &pHapus)
{
    if (Q.Head == nullptr)
    {
        cout << "List kosong! Tidak ada data yang bisa dihapus.\n";
        pHapus = nullptr;
        return;
    }
    else if (Q.Head == Q.Tail)
    {
        pHapus = Q.Head;
        Q.Head = nullptr;
        Q.Tail = nullptr;
    }
    else
    {
        pHapus = Q.Head;
        Q.Head = Q.Head->next;
    }
}

int gajiGolongan(int golongan)
{
    int gaji = 0;
    switch (golongan)
    {
    case 1:
        gaji = 3000000;
        break;
    case 2:
        gaji = 4000000;
        break;
    case 3:
        gaji = 5000000;
        break;
    default:
        gaji = 0;
        break;
    }
    return gaji;
}

int average(Queue Q)
{
    if (Q.Head == nullptr)
    {
        return 0;
    }
    else
    {
        int banyakData = 0;
        int totalGaji = 0;
        Pointer temp = Q.Head;
        do
        {
            banyakData++;
            totalGaji += gajiGolongan(temp->data.golongan);
            temp = temp->next;
        } while (temp != nullptr);
        return totalGaji / banyakData;
    }
}

int MAXGaji(Queue Q)
{
    int Max = 0;
    if (Q.Head == nullptr)
    {
        return 0;
    }
    else
    {
        Pointer temp = Q.Head;
        do
        {
            if (Max < gajiGolongan(Q.Head->data.golongan))
            {
                Max = gajiGolongan(Q.Head->data.golongan);
            }
            // Q.Head = Q.Head->next;
            temp = temp->next;

        } while (temp != nullptr);
    }
    return Max;
}

int MINGaji(Queue Q)
{
    int Min = INT32_MAX;
    if (Q.Head == nullptr)
    {
        return 0;
    }
    else
    {
        Pointer temp = Q.Head;
        do
        {
            if (Min > gajiGolongan(Q.Head->data.golongan))
            {
                Min = gajiGolongan(Q.Head->data.golongan);
            }
            // Q.Head = Q.Head->next;
            temp = temp->next;

        } while (temp != nullptr);
    }
    return Min;
}

void traversal(Queue Q)
{
    Pointer pHelp = Q.Head;
    int gaji;
    int n = 0;
    cout << "=======================================================" << endl;
    cout << "NO\tNIP\tNama\tGolongan\tGaji" << endl;
    cout << "=======================================================" << endl;

    do
    {
        n++;
        gaji = gajiGolongan(pHelp->data.golongan);
        cout << n << "\t" << pHelp->data.NIP << "\t" << pHelp->data.nama <<
"\t" << pHelp->data.golongan << "\t\t" << gaji << endl;
        pHelp = pHelp->next;
    } while (pHelp != nullptr);

    cout << "=======================================================" << endl;
    cout << "Rata-rata Gaji  :  " << average(Q) << endl;
    cout << "Gaji Tertinggi  :  " << MAXGaji(Q) << endl;
    cout << "Gaji Terendah   :  " << MINGaji(Q) << endl;
}
```

- **Hasil Run**

```
=======================================================
NO      NIP     Nama    Golongan        Gaji
=======================================================
1       001     A       1               3000000
2       002     B       2               4000000
3       003     C       3               5000000
4       004     D       3               5000000
5       005     E       3               5000000
=======================================================
Rata-rata Gaji  :  4400000
Gaji Tertinggi  :  5000000
Gaji Terendah   :  3000000
```

```
=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 3
Pegawai dengan NIP 001 telah dihapus!

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 3
Pegawai dengan NIP 002 telah dihapus!

=================Main Menu=================
1. Tambah Pegawai
2. Tampilkan Data Pegawai dan Rata-rata Gaji
3. Hapus Data Pegawai
4. Keluar
Masukkan pilihan Anda: 2
======================================================
NO      NIP     Nama    Golongan        Gaji
======================================================
1       003     C       3               5000000
2       004     D       3               5000000
3       005     E       3               5000000
======================================================
Rata-rata Gaji  :  5000000
Gaji Tertinggi  :  5000000
Gaji Terendah   :  5000000
```