

**STRUKTUR DATA**  
**Tugas Stack**



**NAMA: Bagas Diatama Wardoyo**  
**NPM: 140810230061**

**Dikumpulkan tanggal :**  
**12 Mei 2024**

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

## Stack\_List:

Source Code :

```
/* Nama program : Stack_LinkedList
   Nama          : Bagas Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 10/05/2024
   Deskripsi     : stack (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 Stack;

void createStack(Stack &Top);
void createElement(Stack &newNode);
void push(Stack &Top, Pointer newNode);
void pop(Stack &Top, Pointer &pHapus);
int gajiGolongan(int golongan);
int average(Stack Top);
int MAXGaji(Stack Top);
int MINGaji(Stack Top);
void traversal(Pointer Top);

int main()
```

```

{
    Pointer Top, pHapus;
    Stack newNode;

    createStack(Top);

    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);
                push(Top, newNode);
                cout << "Pegawai berhasil ditambahkan!\n";
                break;
            case 2:
                traversal(Top);
                break;
            case 3:
                pop(Top, 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 createStack(Stack &Top)
    {
        Top = nullptr;
    }

    void createElement(Stack &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 push(Stack &Top, Pointer newNode)
    {
        if (Top == nullptr)
        {
            Top = newNode;
        }
        else
        {
            newNode->next = Top;
            Top = newNode;
        }
    }

    void pop(Stack &Top, Pointer &pHapus)
    {
        if (Top == nullptr)
        {
            cout << "Stack kosong! Tidak ada data yang bisa
dihapus.\n";
            pHapus = nullptr;
            return;
        }
    }

```

```

else
{
    pHapus = Top;
    Top = Top->next;
    pHapus->next = nullptr;
}
}

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(Stack Top)
{
    if (Top == nullptr)
    {
        return 0;
    }
    else
    {
        int banyakData = 0;
        int totalGaji = 0;
        Pointer temp = Top;
        do
        {

```

```

        banyakData++;
        totalGaji += gajiGolongan(temp->data.golongan);
        temp = temp->next;
    } while (temp != nullptr);
    return totalGaji / banyakData;
}
}

```

```

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

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

```

```

int MINGaji(Stack Top)
{
    int Min = INT32_MAX;
    if (Top == nullptr)
    {
        return 0;
    }
    else
    {
        Pointer temp = Top;

```

```

        do
        {
            if (Min > gajiGolongan(Top->data.golongan))
            {
                Min = gajiGolongan(Top->data.golongan);
            }
            Top = Top->next;
            temp = temp->next;

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

void traversal(Pointer Top)
{
    Pointer pHelp = Top;
    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(Top) << endl;
    cout << "Gaji Tertinggi : " << MAXGaji(Top) << endl;
}

```

```
    cout << "Gaji Terendah    : " << MINGaji(Top) << endl;
}
```

Hasil Run :

**\*\*Main Menu\*\***

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

Masukkan pilihan Anda: 1

NIP : 001

Nama : Bagus

Golongan : 1

Pegawai berhasil ditambahkan!

**\*\*Main Menu\*\***

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

Masukkan pilihan Anda: 1

NIP : 002

Nama : Diatama

Golongan : 2

Pegawai berhasil ditambahkan!

**\*\*Main Menu\*\***

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

Masukkan pilihan Anda: 1

NIP : 003

Nama : Wardoyo

Golongan : 3

Pegawai berhasil ditambahkan!

**\*\*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	Wardoyo	3	5000000
2	002	Diatama	2	4000000
3	001	Bagas	1	3000000

Rata-rata Gaji : 4000000  
 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 003 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	001	Bagas	1	3000000

Rata-rata Gaji : 3000000  
 Gaji Tertinggi : 3000000  
 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: 4

Program Keluar

**Stack\_Array:**

**Source Code :**

```
/* Nama program : Stack_Array
   Nama          : Bagus Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 10/05/2024
   Deskripsi     : stack (Array)
   ****/
#include <iostream>
#include <string>

using namespace std;

const int maxElement = 256;

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

struct node
{
    pegawai data[maxElement];
    int Top;
};

typedef node *Pointer;
typedef Pointer Stack;

void createStack(Stack &S);
void createElement(pegawai &data);
```

```

int gajiGolongan(int golongan);
int average(Stack S);
int MAXGaji(Stack S);
int MINGaji(Stack S);
void push(Stack &S, pegawai data);
void pop(Stack &S, pegawai &deletePegawai);
void traversal(Stack &S);

int main()
{
    pegawai newPegawai, deletePegawai;
    Stack Top;

    createStack(Top);

    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);
                push(Top, newPegawai);
                cout << "Pegawai berhasil ditambahkan!\\n";
                break;
            case 2:
                traversal(Top);
                break;
            case 3:
                pop(Top, 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 createStack(Stack &S)
{
    S = new node;
    S->Top = -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(Stack S)
    {
        if (S->Top == -1)
        {
            return -1;
        }
        else
        {
            int banyakData = S->Top + 1;
            int totalGaji = 0;
            for (int i = 0; i < banyakData; i++)
            {
                totalGaji += gajiGolongan(S->data[i].golongan);
            }
            return totalGaji / banyakData;
        }
    }

    int MAXGaji(Stack S)
    {
        if (S->Top == -1)
        {
            return -1;
        }
        else
        {
            int Max = 0;
            int gaji;
            int banyakData = S->Top + 1;
            for (int i = 0; i < banyakData; i++)
            {
                gaji = gajiGolongan(S->data[i].golongan);
                if (Max < gaji)
                {
                    Max = gaji;
                }
            }

            return Max;
        }
    }

```

```

}

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

        return Min;
    }
}

void push(Stack &S, pegawai data)
{
    if (S->Top == maxElement - 1)
    {
        cout << "Stack sudah penuh" << endl;
        return;
    }
    else
    {
        S->Top = S->Top + 1;
        S->data[S->Top] = data;
    }
}

void pop(Stack &S, pegawai &deletePegawai)
{
    if (S->Top == -1)
    {

```

```

        cout << "Tidak ada pegawai yang dapat dihapus" << endl;
    }
    else
    {
        deletePegawai = S->data[S->Top];
        S->Top = S->Top - 1;
    }
}

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

```

**Hasil Run :**

```

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

```

Golongan : 1  
Pegawai berhasil ditambahkan!

=====Main Menu=====

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

Masukkan pilihan Anda: 1

NIP : 0002  
Nama : Diatama  
Golongan : 2  
Pegawai berhasil ditambahkan!

=====Main Menu=====

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

Masukkan pilihan Anda: 1

NIP : 003  
Nama : Wardoyo  
Golongan : 3  
Pegawai berhasil ditambahkan!

=====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	Wardoyo	3	5000000
2	0002	Diatama	2	4000000
3	001	Bagas	1	3000000

=====

Rata-rata Gaji : 4000000  
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 003 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	0002	Diatama	2	4000000
---	------	---------	---	---------

2	001	Bagas	1	3000000
---	-----	-------	---	---------

=====

Rata-rata Gaji : 3500000

Gaji Tertinggi : 4000000

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: 4

Program Keluar