LAPORAN

UJIAN SEMESTER LAB 1 DESAIN DAN ANALISIS ALGORITMA

Disusun untuk memenuhi ujian mata kuliah Lab Desain dan Analisis Algoritma 1

Asisten: Petrus Marcelino H. Tampubolon

Disusun Oleh:

1. Elin Betsey Br Ginting (211401050)
2. Muhammad Saffa Wardana (211401056)
3. Shinta Arjanti (211401065)

PRODI S-1 ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

T.A GANJIL 2022

Jl. Dr. T. Mansur No. 9, Kampus Padang Bulan, Medan, 20155, Sumatera Utara

DAFTAR ISI

SOAL I

ALGORITMA

1.1. Algoritma Data Statis

1. Menu Awal

```
int main () {
    int pilihan, identify, owner, option, linked;
    iden:
    cout <<"+=========================================+"<<endl;
    cout <<"|      Welcome To Restaurant Community    |"<<endl;
    cout <<"+=========================================+"<<endl;
    cout <<"|        How would we identify you?       |"<<endl;
    cout <<"+=========================================+"<<endl;
    cout <<"|                                         |"<<endl;
    cout <<"| 1. I'm looking for some restaurant      |"<<endl;
    cout <<"| 2. I'm an restaurant owner              |"<<endl;
    cout <<"| 3. Exit                                 |"<<endl;
    cout <<"|                                         |"<<endl;
    cout <<"+=========================================+"<<endl;
    cout << "I am: ";
    cin >> identify;
    system ("CLS");

    switch (identify){
```

```
+=========================================+
|      Welcome To Restaurant Community    |
+=========================================+
|        How would we identify you?       |
+=========================================+
|                                         |
| 1. I'm looking for some restaurant      |
| 2. I'm an restaurant owner              |
| 3. Exit                                 |
|                                         |
+=========================================+
I am:
```

Algoritma pertama adalah untuk menampilkan menu. Disini pengguna/user diminta untuk memilih sesuai dengan kebutuhan. User hanya perlu memasukkan nomor sebagai kode identifikasi. Pilihan pertama hanya berfungsi untuk menampilkan data seluruh restoran, pilihan kedua untuk proses CRUD, dan pilihan ketiga untuk keluar dari program. Kita sebagai user memilih opsi kedua.

2. Menu Struktur Data

```
+======================+
|  Please choose one!  |
+======================+
|                      |
| 1. Static            |
| 2. Dynamic           |
| 3. Back              |
| 4. Exit              |
|                      |
+======================+
Your Option:
```

```
case 2:
    opsi:
    cout << "+======================+"<<endl;
    cout << "|  Please choose one!  |"<<endl;
    cout << "+======================+"<<endl;
    cout << "|                      |"<<endl;
    cout << "| 1. Static            |"<<endl;
    cout << "| 2. Dynamic           |"<<endl;
    cout << "| 3. Back              |"<<endl;
    cout << "| 4. Exit              |"<<endl;
    cout << "|                      |"<<endl;
    cout << "+======================+"<<endl;
    cout << "Your Option: ";
    cin >> option;
    system("CLS");

    switch (option){
        case 1:
            menu:
            cout <<"+=============================================================+"<<endl;
            cout <<"|                    Hi Owner, what do you want               |"<<endl;
            cout <<"+=============================================================+"<<endl;
            cout <<"|                                                             |"<<endl;
            cout <<"| 1. I want to add my restaurant to the community             |"<<endl;
            cout <<"| 2. I'm looking for some restaurant information              |"<<endl;
            cout <<"| 3. I want to update my restaurant information               |"<<endl;
            cout <<"| 4. I want to remove my restaurant from the community        |"<<endl;
            cout <<"| 5. Back                                                     |"<<endl;
            cout <<"| 6. Exit                                                     |"<<endl;
            cout <<"|                                                             |"<<endl;
            cout <<"+=============================================================+"<<endl;
            cout <<" I want: ";
            cin >> owner;
            system ("CLS");
```

```
+============================================================+
|                  Hi Owner, what do you want                |
+============================================================+
|                                                            |
|  1. I want to add my restaurant to the community           |
|  2. I'm looking for some restaurant information            |
|  3. I want to update my restaurant information             |
|  4. I want to remove my restaurant from the community      |
|  5. Back                                                   |
|  6. Exit                                                   |
|                                                            |
+============================================================+
 I want: _
```

Selanjutnya kita akan dibawa kedalam menu untuk memilih ingin menggunakan struktur data statis atau dinamis. Untuk sekarang kita akan memilih opsi pertama yaitu struktur data statis. Dan kita akan dibawa ke menu CRUD

3. Algoritma Create

```
+==================================+
|             Welcome              |
+==================================+
How many restaurant you have: _
```

```
=====================================================
Restaurant Name         : PakDin
Restaurant ID           : 123451
Restaurant Location     : Pembagunan
Opening Days Schedule   : Everyday
Business Hours          : 24Hours
=====================================================


=====================================================
Restaurant Name         : Gacoan
Restaurant ID           : 495012
Restaurant Location     : Dr.Mansyur
Opening Days Schedule   : Everyday
Business Hours          : 8AM-10PM
=====================================================


=====================================================
Restaurant Name         : Chatime
Restaurant ID           : 853301
Restaurant Location     : Dimana-mana
Opening Days Schedule   : Everyday
Business Hours          : 10AM-7PM
=====================================================


Press any key to continue . . .
```

```
void create()
{
    int banyak, sisa;
    if (jumlah<MAKSIMAL)
    {
        cout << "+==================================+"<<endl;
        cout << "|             Welcome              |"<<endl;
        cout << "+==================================+"<<endl;
        cout << "How many restaurant you have: ";
        cin >> banyak;
        system ("CLS");
        sisa= MAKSIMAL-jumlah-banyak;
        if (sisa>=banyak)
        {
            for (int i=0; i<banyak;i++)
            {
                cout <<"====================================================="<<endl;
                cout << "Restaurant Name\t\t: ";
                cin >> restoran[jumlah].nama_resto;
                cout << "Restaurant ID\t\t: ";
                cin >> restoran[jumlah].id_resto;
                cout << "Restaurant Location\t: ";
                cin >> restoran[jumlah].lokasi_resto;
                cout << "Opening Days Schedule\t: ";
                cin >> restoran[jumlah].hari_buka;
                cout << "Business Hours\t\t: ";
                cin >> restoran[jumlah].jam_buka;
                cout <<"====================================================="<<endl;
                jumlah++;
                cout <<endl<<endl;
            }
        }
        else
        {
            cout <<"====================================================="<<endl;
            cout <<"          Sorry! The remaining free space is: " <<sisa<<endl;
            cout <<"====================================================="<<endl;
        }
    }
    else
    {
        cout <<"====================================================="<<endl;
        cout <<"      Sorry our community database has already full"<<endl;
        cout <<"====================================================="<<endl;
    }
```

Jika user memilih opsi pertama pada menu struktur data, maka user akan dibawa untuk algoritma proses Create, program pertama kali akan mengecek apakah kapasitas data sudah penuh apa belum. Jika sudah penuh maka akan ada pemberitahuan bahwa database sudah penuh. Jika belum penuh maka user akan diminta untuk memasukkan jumlah data yang ingin dimasukkan. Untuk kembali kehalaman selanjutnya user hanya perlu menekan sembarang tombol.

Jika jumlah data yang ingin dimasukkan sudah melebihi sisa kapasitas database maka akan ada pemberitahuan jumlah kapasitas yang masih kosong. Jika jumlah data yang ingin dimasukkan tidak melebihi sisa kapasitas database maka user akan dibawa ketampilan berikutnya, yaitu tampilan untuk mengisi data. Program akan mengulangi format pengisian data sebanyak jumlah data yang ingin dimasukkan. Untuk kembali kehalaman selanjutnya user hanya perlu menekan sembarang tombol.

4. Algoritma Read

```
void read ()
{
    if (jumlah==0){
        cout <<"============================================================"<<endl;
        cout <<"      OH NO! There are no restaurants that have registered yet"<<endl;
        cout <<"                          BE THE FIRST"<<endl;
        cout <<"============================================================"<<endl;
    }
    else
    {
        for (int i=1;i<jumlah;i++)
        {
            for (int j= jumlah-1;j>=i;j--)
            {
                if(restoran[j].id_resto<restoran[j-1].id_resto)
                {
                    swap(restoran[j],restoran[j-1]);
                }
            }
        }
    }
        cout << "======================================================"<<endl;
        cout << "                All Restaurant Information "<<endl;
        cout << "======================================================"<<endl<<endl;
        for (int i=0; i<jumlah;i++)
        {   cout <<"======================================================"<<endl;
            cout <<"Restaurant Name\t\t: "<<restoran[i].nama_resto<<endl;
            cout <<"Restaurant ID\t\t: "<<restoran[i].id_resto<<endl;
            cout <<"Restaurant Location\t: "<<restoran[i].lokasi_resto<<endl;
            cout <<"Opening Days Schedule\t: "<<restoran[i].hari_buka<<endl;
            cout <<"Business Hours\t\t: "<<restoran[i].jam_buka<<endl;
            cout <<"======================================================"<<endl<<endl;
        }
}
```

```
============================================================
                   All Restaurant Information
============================================================

======================================================
Restaurant Name         : PakDin
Restaurant ID           : 123451
Restaurant Location     : Pembagunan
Opening Days Schedule   : Everyday
Business Hours          : 24Hours
======================================================

======================================================
Restaurant Name         : Gacoan
Restaurant ID           : 495012
Restaurant Location     : Dr.Mansyur
Opening Days Schedule   : Everyday
Business Hours          : 8AM-10PM
======================================================

======================================================
Restaurant Name         : Chatime
Restaurant ID           : 853301
Restaurant Location     : Dimana-mana
Opening Days Schedule   : Everyday
Business Hours          : 10AM-7PM
======================================================

Press any key to continue . . .
```

Jika user memilih opsi kedua pada menu struktur data, maka user akan dibawa untuk algoritma proses Read, yaitu untuk menampilkan data. Jika sebelumnya belum ada data yang diinput, akan muncul pemberitahuan bahwa belum ada data/restoran dalam database.

Jika sudah ada data yang diinput kedalam database, maka program pertama-tama akan mengurutkan data berdasarkan “Restaurant ID” secara ascending. Setelahnya, baru data akan ditampilkan.

5. Algoritma Update

Jika user memilih opsi ketiga pada menu struktur data, maka user akan dibawa untuk algoritma proses Update, yaitu untuk mengganti data yang sudah ada sebelumnya. Setelahnya user akan diminta untuk memasukkan posisi/urutan restoran/datanya pada database. Jika nomor posisi yang dimasukkan ada, maka user akan dibawa pada tampilan untuk memasukkan data perbaikan. Jika posisi yang dimasukkan melebihi jumlah data yang sudah diinput kedalam database, maka program akan memberi peringatan bahwa data dengan posisi tersebut tidak ada didalam database.

```
Enter your restaurant position number:
```

```
Enter your restaurant position number: 2
Restaurant Name          : Gacoan
Restaurant ID            : 99999
Restaurant Location      : Indonesia
Opening Days Schedule    : Everyday
Business Hours           : 9AM-8PM
```

```
==========================================
          Data has been updated
==========================================
Press any key to continue . . .
```

```
void update ()
{
    int pilih;
    cout << "Enter your restaurant position number: ";
    cin >> pilih;
    if (pilih<=jumlah && pilih>0)
    {
        cout << "Restaurant Name\t\t: ";
        cin >> restoran[pilih-1].nama_resto;
        cout << "Restaurant ID\t\t: ";
        cin >> restoran[pilih-1].id_resto;
        cout << "Restaurant Location\t: ";
        cin >> restoran[pilih-1].lokasi_resto;
        cout << "Opening Days Schedule\t: ";
        cin >> restoran[pilih-1].hari_buka;
        cout << "Business Hours\t\t: ";
        cin >> restoran[pilih-1].jam_buka;
        system("CLS");
        cout <<"=========================================="<<endl;
        cout <<"          Data has been updated"<<endl;
        cout <<"=========================================="<<endl;
    }
    else
    {   cout <<"=================================================="<<endl;
        cout <<"       Hmm! We do not have any data for that number"<<endl;
        cout <<"=================================================="<<endl;
    }
}
```

6. Algoritma Delete

```
void del ()
{
     int pilih;
     cout <<"Enter your restaurant position number: ";
     cin >> pilih;
     if (pilih<jumlah && pilih>0)
     {
          for (int i =pilih; i<jumlah;i++)
          {
               strcpy(restoran[i-1].nama_resto,restoran[i].nama_resto);
               restoran[i-1].id_resto=restoran[i].id_resto;
               strcpy(restoran[i-1].lokasi_resto,restoran[i].lokasi_resto);
               strcpy(restoran[i-1].hari_buka,restoran[i].hari_buka);
               strcpy(restoran[i-1].jam_buka,restoran[i].jam_buka);
          }
          system ("CLS");
          cout <<"====================================="<<endl;
          cout <<"       We will miss you so much!"<<endl;
          cout <<"====================================="<<endl;
          jumlah--;
     }
     else if (pilih==jumlah)
          {jumlah--;}
     else
     {
          cout <<"================================================"<<endl;
          cout <<"          Oh no! There is no restaurant"<<endl;
          cout <<"================================================"<<endl;
     }
}
```

```
Enter your restaurant position number:
```

```
=====================================
       We will miss you so much!
=====================================
Press any key to continue . . .
```

Pilihan keenam adalah pilihan untuk proses Delete. User akan diminta untuk memasukkan posisi/urutan restoran/datanya pada database. Jika nomor posisi yang dimasukkan ada, maka program akan menghapus semua data yang berada pada posisi tersebut dan memindahkan data setelah data yang dihapus ke posisi data yang telah dihapus. Jika melakukan Delete, maka kapasitas dari database akan bertambah. Jika pada posisi yang dimasukkan tidak ada data, program akan memberitahu bahwa tidak ada data pada posisi yang dipilih.

7. Algoritma Read Kembali

Karena kita telah melakukan proses editing dan juga deleting, kita dapat melihat kembali data-data hasil editing dan deleting. Caranya masih sama, kita cukup memilih opsi kedua pada menu struktur data, atau opsi pertama pada menu awal.

1.2. Algoritma Data Dinamis

```
using namespace std;
const int MAKSIMAL= 50;
int jumlah=0;

struct node
{
    char nama_resto [100];
    int  id_resto;
    char lokasi_resto [100];
    char hari_buka [100];
    char jam_buka [100];
    node *next;
};

node *awal_ptr = NULL;
node *posisi;
int option = 0;
```

Pada struktur data statis dan dinamis, sebenarnya tidak banyak perbedaannya. Yang paling utama adalah penggunaan pointer dan codingan sorting yang berbeda. Dimana untuk proses sorting pada data dinamis, codingannya lebih rumit, karena kita harus membuat 3 fungsi. Untuk tampilan, struktur data statis dan dinamis sama persis, hanya algoritma pemrogramannya saja yang berbeda.

Sebelumnya setelah kita membuat struct, kita jangan lupa untuk memberika nilai pada node awal dan variabel pointer untuk menunjukan posisi dari node. Untuk memilih data dinamis, silahkan pilih opsi nomor 2 pada menu kedua.

1. Algoritma Create

Untuk pilihan pertama, Creat, setelah jumlah data yang ingin dimasukkan telah diperiksa, maka saat user memasukkan data, program akan menerima data dan dimasukkan dengan cara pointer, jadi program tidak menyalin isi data yang dimasukkan kedalam node tetapi menunjukan bahwa posisi data pada node.

Selanjutnya posisi awal pointer akan dipindahkan ke node selanjutnya, sehingga kita melakukan penambahan data pada posisi node selanjutnya. Untuk selebihnya, algoritma struktur data statis dan dinamis sama.

```
void buat()
{
     int banyak, sisa;
     if (jumlah<MAKSIMAL)
     {
          cout << "+=======================================+"<<endl;
          cout << "|              Welcome                  |"<<endl;
          cout << "+=======================================+"<<endl;
          cout << "How many restaurant do you have: ";
          cin >> banyak;
          system ("CLS");
          sisa= MAKSIMAL-jumlah-banyak;
          if (sisa>=banyak)
          {
               for (int i=0; i<banyak;i++)
               {
                    node *temp, *temp2;
                    temp = new node;
                    cout <<"============================================================"<<endl;
                    cout << "Restaurant Name\t\t: ";
                    cin >> temp -> nama_resto;
                    cout << "Restaurant ID\t\t: ";
                    cin >> temp -> id_resto;
                    cout << "Restaurant Location\t: ";
                    cin >> temp -> lokasi_resto;
                    cout << "Opening Days Schedule\t: ";
                    cin >> temp -> hari_buka;
                    cout << "Business Hours\t\t: ";
                    cin >> temp -> jam_buka;
                    cout <<"============================================================"<<endl;
                    cout <<endl;
                    jumlah++;
                    temp->next = NULL;
                    if (awal_ptr == NULL)
                    {
                         awal_ptr = temp;
                         posisi = awal_ptr;
                    }
                    else
                    {
                         temp2 = awal_ptr;
                         while (temp2->next != NULL)
                         {
                              temp2 = temp2->next;
                         }
                         temp2->next = temp;
                    }
               }
          }
          else
          {
               cout <<"============================================================"<<endl;
               cout <<"          Sorry! The remaining free space is: " <<sisa<<endl;
               cout <<"============================================================"<<endl;
          }
     }
     else
     {
          cout <<"============================================================"<<endl;
          cout <<"       Sorry! Our community database has already full"<<endl;
          cout <<"============================================================"<<endl;
     }
```

2. Algoritma Sorting

```
void swapnode(node *left, node *right)
{
    node *temp;
    temp = new node;
    *temp->nama_resto = *left->nama_resto;
    temp -> id_resto = left -> id_resto;
    *temp -> lokasi_resto = *left -> lokasi_resto;
    *temp -> hari_buka = *left -> hari_buka;
    *temp -> jam_buka = *left -> jam_buka;

    *left -> nama_resto = *right -> nama_resto;
    left -> id_resto = right -> id_resto;
    *left -> lokasi_resto = *right -> lokasi_resto;
    *left -> hari_buka = *right -> hari_buka;
    *left -> jam_buka = *right -> jam_buka;

    *right -> nama_resto = *temp -> nama_resto;
    right -> id_resto = temp -> id_resto;
    *right -> lokasi_resto = *temp -> lokasi_resto;
    *right -> hari_buka = *temp -> hari_buka;
    *right -> jam_buka = *temp -> jam_buka;
}
```

```
    *right -> nama_resto = *temp -> nama_resto;
    right -> id_resto = temp -> id_resto;
    *right -> lokasi_resto = *temp -> lokasi_resto;
    *right -> hari_buka = *temp -> hari_buka;
    *right -> jam_buka = *temp -> jam_buka;
}

node* carilist(node* awal_ptr, int urutan)
{
    node* temp = awal_ptr;
    for (int i = 0; i < urutan; i++){
        temp = temp->next;
    }
    return temp;
}

void selectionSort(node* awal, int size)
{
    int i, j, imin;
    for (i = 0; i < size - 1; i++) {
        imin = i;   //get index of minimum data
        for (j = i + 1; j < size; j++) {
            if (carilist(awal, j)->id_resto < carilist(awal, imin)->id_resto) {
                imin = j;
            }
        }
        //placing in correct position
        swapnode(carilist(awal, i), carilist(awal, imin));
    }
}
```

Sesuai dengan penjelasan sebelumnya, untuk sorting struktur data dinamis memerlukan 3 fungsi. Fungsi pertama yaitu “selectionSort” untuk membandingkan “Restauran ID” pada node satu dengan node selanjutnya, fungsi kedua yaitu “swapnode” untuk menukar data antar node yang akan terjadi tergantung nilai dari fungsi pertama,

dan fungsi ketiga “carilist” untuk mencari node beserta seluruh datanya. Setelah melalui 3 fungsi ini, baru data kita dapat terurut.

3. Algoritma Read

```
void baca()
{
    node *temp;
    temp = awal_ptr;
    cout << endl;
    if (temp == NULL)
    {
        cout <<"=============================================================================="<<endl;
        cout <<"       OH NO! There are no restaurants that have registered yet"<<endl;
        cout <<"                                  BE THE FIRST"<<endl;
        cout <<"=============================================================================="<<endl;
    }
    else
    {
        cout << "======================================================"<<endl;
        cout << "                    All Restaurant Information "<<endl;
        cout << "======================================================"<<endl<<endl;
        while (temp != NULL)
        {
            cout <<"======================================================"<<endl;
            cout << "Restaurant Name\t\t: "<<temp -> nama_resto<<endl;
            cout << "Restaurant ID\t\t: "<<temp -> id_resto<<endl;
            cout << "Restaurant Location\t: "<<temp -> lokasi_resto<<endl;
            cout << "Opening Days Schedule\t: "<<temp -> hari_buka<<endl;
            cout << "Business Hours\t\t: "<<temp -> jam_buka<<endl;
            cout <<"======================================================"<<endl;
            cout << endl;
            temp = temp->next;
        }
    }
}
```

Setelah data diurutkan, barulah kita akan menampilkan data yang telah dimasukkan. Yaitu dengan memilih opsi kedua seperti pada data dinamis. Program akan menampilkan data yang telah diurutkan. Bedanya pada data statis, program akan mengambil atau bisa dibilang mengcopy paste nilai data-data, tetapi pada data statis, program cukup menunjuk alamat yang berisi data yang telah dimasukkan sebelumnya.

4. Algoritma Edit

```
void perbaru(int y){
    bool cari = false;
    node *temp;
    temp = awal_ptr;

    while(temp != NULL)
    {
        if(temp->id_resto==y)
            {
            cari = true;

                cout <<"============================================================"<<endl;
                cout << "Restaurant Name\t\t: ";
                cin >> temp -> nama_resto;
                cout << "Restaurant ID\t\t: ";
                cin >> temp -> id_resto;
                cout << "Restaurant Location\t: ";
                cin >> temp -> lokasi_resto;
                cout << "Opening Days Schedule\t: ";
                cin >> temp -> hari_buka;
                cout << "Business Hours\t\t: ";
                cin >> temp -> jam_buka;
                cout <<"============================================================"<<endl;
                cout <<endl;
                system ("CLS");
                cout <<"=============================================="<<endl;
                cout <<"             Data has been updated"<<endl;
                cout <<"=============================================="<<endl;

            getch();
        }
        temp = temp->next;
    }
}
```

Kembali lagi, untuk editing, untuk data dinamis dan statis tidak jauh berbeda. Ketika pengeditan dilakukan, pada data dinamis nilai data yang baru tidak menggantikan data yang lama, tetapi penghubung ataupun alamat yang sebelumnya menuju data lama sekarang dialamatkan kepada alamat baru.

5. Algoritma Delete

```
void hapus()
{
     int banyakdata,posisi_hapus,poshapus;
     node *hapus, *bantu;
     if(awal_ptr != NULL)
     {
          cout<<"Enter your restaurant position number: ";
          cin>>posisi_hapus;
          banyakdata=1;
          bantu=awal_ptr;
          while(bantu->next != NULL)
          {
               bantu=bantu->next;
               banyakdata++;
          }
     if((posisi_hapus<1)||(posisi_hapus>banyakdata))
     {
          cout <<"================================================"<<endl;
          cout <<"          Oh no! There is no restaurant"<<endl;
          cout <<"================================================"<<endl;
     }
     else
     {
          bantu=awal_ptr;
          poshapus=1;
          while(poshapus<(posisi_hapus-1))
          {
               bantu=bantu->next;
               poshapus++;
          }
          hapus=bantu->next;
          bantu->next=hapus->next;
          delete hapus;
     }
               banyakdata++;
          }
     if((posisi_hapus<1)||(posisi_hapus>banyakdata))
     {
          cout <<"================================================"<<endl;
          cout <<"          Oh no! There is no restaurant"<<endl;
          cout <<"================================================"<<endl;
     }
     else
     {
          bantu=awal_ptr;
          poshapus=1;
          while(poshapus<(posisi_hapus-1))
          {
               bantu=bantu->next;
               poshapus++;
          }
          hapus=bantu->next;
          bantu->next=hapus->next;
          delete hapus;
     }
     system ("CLS");
     cout <<"======================================"<<endl;
     cout <<"       We will miss you so much!"<<endl;
     cout <<"======================================"<<endl;
     jumlah--;
     }
     else
     {
          cout <<"================================================"<<endl;
          cout <<"       Oh no! There is no restaurant left"<<endl;
          cout <<"================================================"<<endl;
     }
}
```

Kemudian untuk melakukan delete, user memilih opsi keempat, kemudian sama seperti data statis, user diminta untuk memasukkan posisi data. Yang membedakan dengan data statis adalah pada data dinamis, penghapusan menggunakan pointer. Jadi alamat yang sebelumnya mengarah kepada node yang akan dihapus, diputus dan dialih alamatkan kepada node/data selanjutnya. Ini sama seperti saat pergandengan tangan, kita melepas gandengan dan menggandeng tangan orang selanjutnya.

SOAL II

KOMPRESI DATA

2.1. Kompresi Data: Algoritma End Tagged Dense Code

2.1.1. Data

Adapun data yang dipakai berjumlah 5 data, dengan nilai data sebagai berikut:

```
==========================================================
                  All Restaurant Information
==========================================================

=====================================================
Restaurant Name          : padin
Restaurant ID            : 101
Restaurant Location      : medan
Opening Days Schedule    : everyday
Business Hours           : 24
=====================================================

=====================================================
Restaurant Name          : padin2
Restaurant ID            : 102
Restaurant Location      : bandung
Opening Days Schedule    : everyday
Business Hours           : 12
=====================================================

=====================================================
Restaurant Name          : padin3
Restaurant ID            : 103
Restaurant Location      : medan
Opening Days Schedule    : everyday
Business Hours           : 5
=====================================================

=====================================================
Restaurant Name          : padin4
Restaurant ID            : 104
Restaurant Location      : medan
Opening Days Schedule    : everyday
Business Hours           : 9
=====================================================

=====================================================
Restaurant Name          : padin5
Restaurant ID            : 105
Restaurant Location      : medan
Opening Days Schedule    : everyday
Business Hours           : 12
=====================================================
```

2.1.2. Tabel Data Sebelum Dikompresi

Char	Freq	ASCII Code	Bit	Bit x Freq
Sp	24	00100000	8	192
a	15	01100001	8	120
d	15	01100100	8	120
e	14	01100101	8	112
n	11	01101110	8	88

1	8	00000001	8	64
p	5	01110000	8	40
i	5	01101001	8	40
v	5	01110110	8	40
r	5	01110010	8	40
y	5	01111001	8	40
0	5	00000000	8	40
2	5	00000010	8	40
m	4	01101101	8	32
4	3	00000100	8	24
5	3	00000101	8	24
3	2	00000011	8	16
b	1	01100010	8	8
u	1	01110101	8	8
g	1	01100111	8	8
9	1	00001001	8	8
TOTAL				**1104**

2.1.3. Tabel Data Setelah Dikompresi

Pada kompresi ini kita menggunakan b= 3, karena dinilai lebih efisien. Berikut adalah hasil kompresi yang didapat.

Char	Freq	ETDC	Bit	Bit x Freq
Sp	24	100	3	72
a	15	101	3	45
d	15	110	3	45
e	14	111	3	42
n	11	000100	6	66
1	8	000101	6	48
p	5	000110	6	30
i	5	001100	6	30
v	5	001101	6	30
r	5	001110	6	30
y	5	001111	6	30
0	5	010100	6	30
2	5	010101	6	30
m	4	010110	6	24
4	3	010111	6	18
5	3	011100	6	18
3	2	011101	6	12
b	1	011110	6	6
u	1	011111	6	6
g	1	000000100	9	9
9	1	000000101	9	9
TOTAL				**630**

2.1.4. String Bit

Setelah melakukan kompresi, langkah selanjutnya adalah menghitung keseluruhan string bit setelah penambahan padding dan flag bit. Kemudian kita akan menghitung efisiensi dari kompresi ini.

Total bit setelah dikompresi= 630

Karena total bit setelah dikompresi tidak habis dibagi delapan, maka kita akan menambahkan padding bit sebanyak dua angka nol (0) agar total bit setelah dikompresi habis dibagi delapan.

Total bit setelah dikompresi= 632

Selanjutnya agar padding bit tidak dibaca sebagai karakter oleh komputer, kita perlu menambahkan flag bit, yaitu delapan buat bit. Dengan cara, mengubah banyak angka nol pada padding bit. Karena jumlah angka nol pada padding bit adalah 2 maka flag bit nya adalah 00000010. Maka sekarang:

Total bit setelah dikompresi= 640

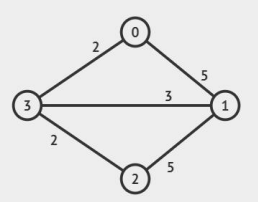
Compression Ratio(Cr)
Cr = (ukuran bit sebelum dikompresi)/(ukuran bit setelah dikompresi)
Cr = 1104/640
Cr = 1.725

Space Savings
SS = 1-(ukuran bit setelah dikompresi)/(ukuran bit sebelum dikompresi) x100%
SS = 1 – 640/1104 x 100%
SS = 42,02898%

Bit Rate
Bit Rate = (jumlah bit terkompresi)/(jumlah jenis karakter)
Bit Rate = 640/21 = 30.66
Bit Rate = 30.476 Bit/char

2.2. Kompresi Data: Visualisasi Minimum Spanning Tree

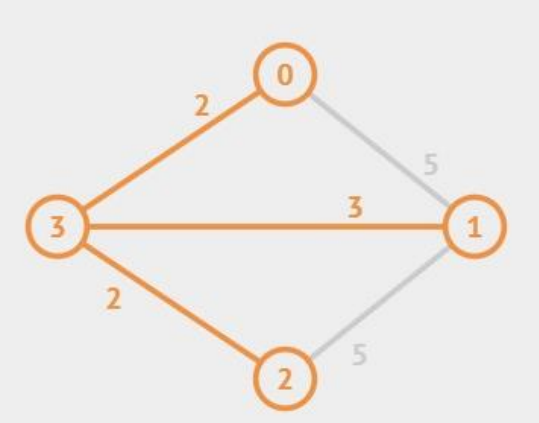
2.1.1. Graf Awal



Keterangan:
0= create
1= read
2= delete
3= update

2.1.1. Graf Awal

Graf awal dinilai tidak memenuhi minumum spanning tree, dikarenakan belum efisien dan juga masih ada siklus pada grafnya. Sehingga graf awal diubah menjadi:



Keterangan:
0= create
1= read
2= delete
3= update

Dengan garis abu-abu adalah sisi yang dihapus, sehingga menghasilkan graf baru dengan sisi berwarna jingga.

LAMPIRAN

Lampiran Link Youtube:
https://youtu.be/gEK3PG3Z71I