

LAPORAN PRAKTIKUM MODUL KE-9
MEMBUAT PROGRAM POHON KELUARGA DENGAN FUNGSI REKURSI



Dosen Pengampu :
I Ketut Purnamawan, S.Kom., M.Kom.

Disusun Oleh :
I Gede Gelgel Abdiutama ; 2115101014

MATA KULIAH PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
UNIVERSITAS PENDIDIKAN GANESHA
SINGARAJA
TA. 2022

A. PERMASALAHAN

Buatlah program untuk menelusuri jalur keturunan seseorang, dari sebuah pohon keluarga. Pohon keluarga dibentuk dengan menggunakan array struct.

Contoh struct-nya sebagai berikut.

```
struct orang
{
    int indexAyah;

    char nama[30];

};
```

Orang ke-0 pada array merupakan root.

Untuk root, indexAyah diset -1.

Pengguna bisa menambah dan melihat anggota keluarga. Program bisa menampilkan jalur keturunan seseorang, dari root ke orang tersebut. Untuk melihat jalur keturunan seseorang dari root, user harus memasukkan index dari orang tersebut.

B. KAJIAN TEORI

1. Bahasa Pemrograman C

Bahasa pemrograman C dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Meskipun C dibuat untuk memprogram sistem dan jaringan komputer, bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan ekstensi dari C. Bahasa C terdiri dari beberapa versi seperti C K&R, ANSI C & ISO C, dan C99.

Contoh program bahasa C:

```
#include <stdio>
int main()
{
printf("Hello World");
return 0;
}
```

2. Bahasa Pemrograman C++

Bahasa pemrograman C++ adalah bahasa pemrograman komputer yang di buat oleh Bjarne Stroustrup, yang merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, bahasa itu diturunkan dari bahasa B yang ditulis oleh Ken Thompson pada tahun 1970 yang diturunkan dari bahasa sebelumnya yaitu BCL. Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix. Pada perkembangannya, versi ANSI (American National Standards Institute) pada bahasa pemrograman C menjadi versi dominan, meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk embedded system. Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan support system untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (object-oriented programming).

Contoh program bahasa C++:

```
#include <iostream>
int main()
{
std::cout << "Hello World";
return 0;
}
```

3. Tipe Data

Data types atau tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Untuk mengembangkan program dalam bahasa pemrograman C atau C++ terdapat berbagai jenis tipe data yang dapat dipilih dan digunakan sesuai dengan kebutuhan dan karakter nilai yang ingin disimpan di dalam variable. Jenis tipe data yang sering digunakan yaitu Boolean, Character, String, Integer, Floating Point, dan Double Floating Point.

a. Boolean

Tipe data Boolean merupakan tipe yang memiliki dua nilai yaitu benar (true) atau salah (false). Nilai yang digunakan pada tipe ini sangat penting dalam mengambil keputusan suatu kejadian tertentu.

b. Character

Tipe data character merupakan salah satu tipe data yang memungkinkan kita untuk memesan memori berformat text (huruf, angka, dan simbol) dengan karakter tunggal. Dibutuhkan 1 byte atau 8 bit ruang di dalam memori agar dapat menyimpan sebuah karakter.

c. String

Tipe data string terdiri dari kumpulan karakter dengan panjang tertentu, dan seringkali dianggap sebagai tipe data dasar. Hal ini dikarenakan hingga saat ini tipe data string paling sering digunakan oleh para programmer.

d. Integer

Jenis tipe data integer dapat didefinisikan sebagai bilangan bulat. Artinya, suatu program yang menggunakan tipe data integer ini tidak mendukung penggunaan huruf. Selain itu, bilangan yang digunakan juga haruslah bulat (tidak mengandung pecahan decimal).

e. Floating Point

Tipe data floating point atau real number merupakan tipe data angka yang memiliki bagian decimal di akhir angka. Tipe data float cocok digunakan untuk variable yang akan berisi angka pecahan.

f. Double Floating Point

Sama halnya dengan floating point, yang bersifat menyatakan bilangan pecahan. Bedanya adalah penyimpangan angka maksimal lebih besar daripada float, otomatis double juga akan membutuhkan memori yang lebih besar.

4. Percabangan

Percabangan adalah sebuah tahap dimana program akan melakukan pengecekan kondisi. Kondisi ini bisa digunakan untuk menentukan bagian program/statement mana yang akan dijalankan jika kondisi tertentu terpenuhi. Di dalam bahasa C, kita dapat membuat seleksi dengan if else.

a. if

Pernyataan if : “Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan.”

```
if(kondisi){  
    //statement  
}
```

b. if else

Pernyataan if else : “Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2”

```
if(kondisi){  
    //statement  
}else{  
    //statement  
}
```

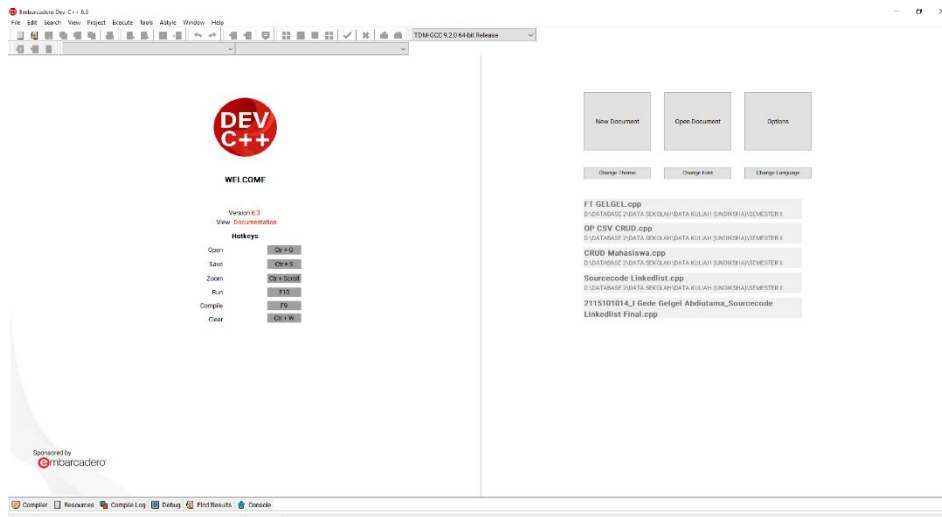
C. PEMBAHASAN

Dalam pembahasan ini, akan menjelaskan proses atau tahapan yang dilakukan pada saat melakukan Praktikum membuat program untuk menelusuri jalur keturunan seseorang, dari sebuah pohon keluarga yang dibentuk dengan menggunakan array struct. Program ini menggunakan bahasa C++ sebagai bahasa pemrogramannya, dan untuk isi program ini menggunakan fungsi rekursif. Rekursif adalah proses di mana suatu fungsi memanggil dirinya sendiri baik secara langsung maupun tidak langsung. Dalam fungsi rekursi, kode di dalam fungsi dieksekusi berulang kali hingga kontrol eksekusi mendapatkan inputan yang sesuai dari perintah user.

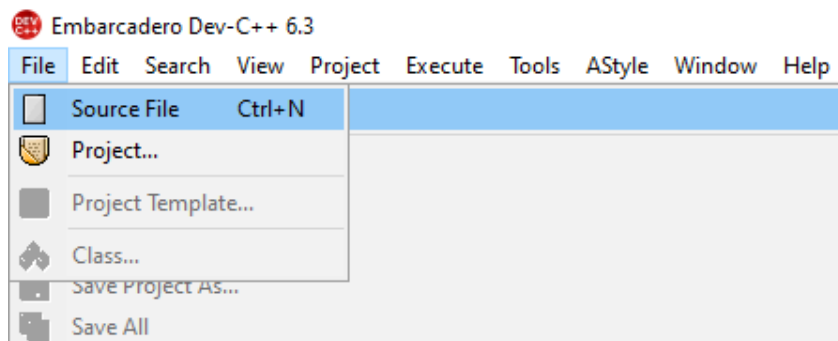
Pada program ini dibuat beberapa menu diantaranya terdapat menu untuk user menambah, melihat anggota keluarga, dan juga bisa melihat silsilah atau jalur keturunannya berdasarkan index yang dicari, sehingga ketika user menginputkan index seseorang yang akan dicari maka program akan menampilkan jalur keturunannya sesuai yang sudah diinputkan pada saat di menu tambah anggota keluarga. Secara konsep dibuatkan array struct nya terlebih dahulu, program ini mirip konsep Tree, yang dimana orang ke-0 pada array itu merupakan root, dan untuk root index Ayah nya akan di set menjadi -1. User akan menginputkan nama dan Index Parrent nya untuk menentukan seseorang itu berasal dari index Parrent mana, sehingga pada saat melakukan searching dengan index nanti akan dimunculkan data yang sesuai dengan yang telah diinputkan.

Berikut Langkah-langkah praktikum

1. Pertama, kita buka terlebih dahulu Compiler yang digunakan. Sebagai contoh disini saya menggunakan Dev C++ sebagai compiler nya.



2. Setelah terbuka, kita buat source file dengan cara memilih pada File > New > Source File, atau bisa juga dengan menggunakan shortcut pada keyboard yaitu CTRL + N.



3. Kemudian kita akan eksekusi kode programnya, karena saya menggunakan bahasa pemrograman C++, maka kita membutuhkan sebuah kerangka dasarnya untuk bisa membentuk kode program lainnya, berikut merupakan header dan main program dari bahasa C

```
#include <iostream>

int main()
{
    std::cout << "Hello World";
    return 0;
}
```

4. Pada tahap ini, kita bisa melanjutkan proses pembuatan program dengan melakukan coding sesuai dengan kode program yang akan dibuat. Untuk kode program akan dilampirkan pada halaman Lampiran paling akhir laporan ini.

```

1  #include <iostream>
2  #include <stdlib.h>
3  #include <conio.h>
4  #include <string.h>
5  #define POHON 50
6
7  using namespace std;
8
9  struct data{
10     int id_Ayah;
11     char nama[51];
12 };
13
14 int search, pilih, tmp = -1;
15 struct data keluarga[POHON];
16 char ulang, isi[POHON][POHON];
17
18 void tambahRoot();
19 void tambahNode();
20 void lihatKeluarga(struct data keluarga[], int j, int tmp);
21 void JalurInden(struct data keluarga[], int search, int x, int m);
22 void lihatJalur(int x);
23
24 int main();
25
26 void menu(){
27     system("cls");
28     cout << "PERMASALAHAN MODUL KE-9"<< endl;
29     cout << "PROGRAM FAMILY TREE"<< endl<< endl;
30     cout << "PILIHAN MENU PROGRAM"<< endl;
31     cout << "1. Tambah Keluarga"<< endl;

```

Line: 172 Col: 2 Sel: 0 Lines: 172 Length: 3852 Insert

5. Untuk melakukan compile sehingga program dapat dijalankan, kita klik Execute lalu pilih Compile atau bisa juga dengan menggunakan shortcut pada keyboard yaitu F9. Berikutnya akan muncul tampilan yang meminta kita memilih lokasi untuk menyimpan file cpp sebelum di compile. Sebagai contoh disini akan saya simpan dengan nama Modul9.cpp.

The screenshot shows a Windows IDE with a C++ program open. The program is a simple search implementation using a struct and a linear search function. A 'Save As' dialog box is open, showing the file is being saved to the 'DATABASE (D:)' directory. The file name is 'data1.cpp'. The dialog also shows a list of files in the directory, including 'DATABASE 1', 'DATABASE 2', and 'DATABASE 3'.

The C++ code in the background is as follows:

```
#include <iostream>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#define Pohon 50

using namespace std;

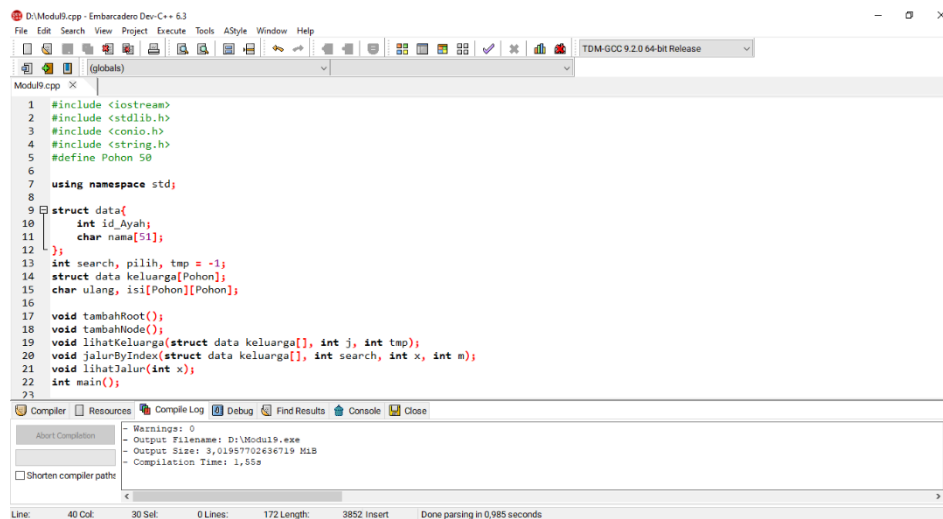
struct data{
    int id_Ayah;
    char nama[51];
};

int search, pilih, tmp = -1;
struct data keluarga[Pohon];
char ulang, isi[Pohon][Pohon];

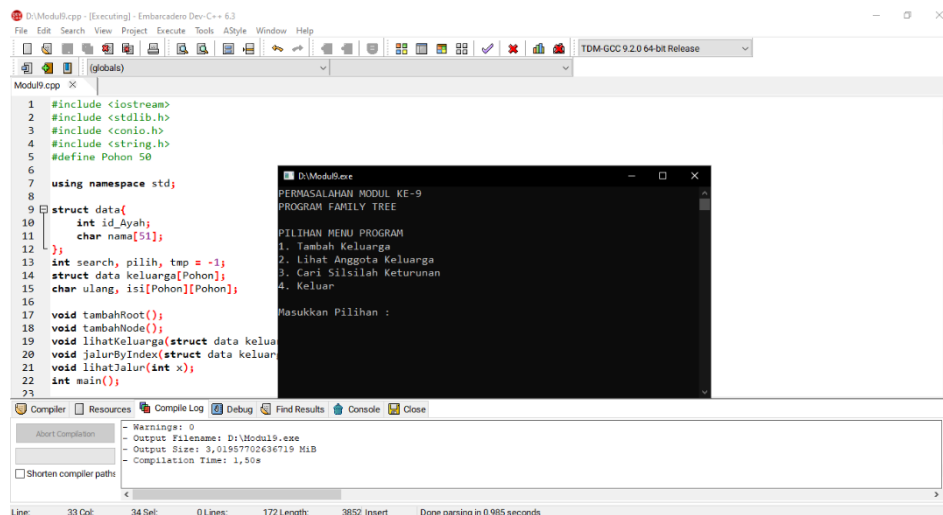
void tambahfloat();
void tambahNode();
void lihatKeluarga(struct data keluarga);
void JalurbyIndex(struct data keluarga);
void lihatJalur(int x);

int main();
```


6. Maka setelah itu hasil dari program yang telah kita buat akan di compile secara otomatis, perlu diperhatikan jika Errors 0 dan Warnings 0 maka program siap untuk di Run dengan cara klik Execute lalu pilih Run menekan tombol F10.



7. Ketika program sudah di run, maka langkah selanjutnya kita akan mengeksekusi program kita dengan cara memilih menu yang telah disediakan pada program, maka program akan secara otomatis mengarahkan ke menu yang dipilih.



D. KESIMPULAN

Pembuatan program untuk menelusuri jalur keturunan seseorang, dari sebuah pohon keluarga ini menggunakan bahasa C++ dan untuk isi program ini menggunakan fungsi rekursif untuk menelusuri garis keturunan dari seseorang. Pada program ini dibuat beberapa menu diantaranya terdapat menu untuk user menambah, melihat anggota keluarga, dan juga bisa melihat silsilah atau jalur keturunannya berdasarkan index yang dicari. Secara konsep program ini terbentuk atas array struct dan mirip seperti Tree, sehingga dalam membuat program ini perlu memahami fungsi rekursi dan konsep Tree dengan baik.

E. LAMPIRAN

Kode Program:

```
#include <iostream>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#define Pohon 50

using namespace std;

struct data{
    int id_Ayah;
    char nama[51];
};
int search, pilih, tmp = -1;
struct data keluarga[Pohon];
char ulang, isi[Pohon][Pohon];

void tambahRoot();
void tambahNode();
void lihatKeluarga(struct data keluarga[], int j, int tmp);
void jalurByIndex(struct data keluarga[], int search, int x, int m);
void lihatJalur(int x);
int main();

void menu(){
    system("cls");
    cout << "PERMASALAHAN MODUL KE-9" << endl;
    cout << "PROGRAM FAMILY TREE" << endl << endl;
    cout << "PILIHAN MENU PROGRAM" << endl;
    cout << "1. Tambah Keluarga" << endl;
    cout << "2. Lihat Anggota Keluarga" << endl;
    cout << "3. Cari Silsilah Keturunan" << endl;
    cout << "4. Keluar" << endl << endl;
    cout << "Masukkan Pilihan : ";
    cin >> pilih;
    switch (pilih){
        case 1:{
            system("cls");
            cout << "(MENU 1). TAMBAH KELUARGA" << endl << endl;
            if (tmp == -1){
                tambahRoot();
            }else{
                tambahNode();
            }
            break;
        }
        case 2:{
            system("cls");
            cout << "(MENU 2). LIHAT ANGGOTA KELUARGA" << endl << endl;
            cout << "Root \t\t\t: " << keluarga[-1].nama << endl;
            cout << "Index dari " << keluarga[-1].nama << " \t: -1" << endl;
            cout << "=====" << endl;
            cout << "| Index\t| Nama\t\t| Index Parrent |" << endl;
            cout << "=====" << endl;
            lihatKeluarga(keluarga, 0, tmp);
            cout << endl << endl << endl << "Tekan ENTER untuk kembali...";
        }
    }
}
```

```

        getch();
        system("cls");
        menu();
        break;
    }
    case 3:{
        system("cls");
        cout <<"(MENU 3). LIHAT KETURUNAN BY INDEX"<< endl<< endl;
        cout <<"\nMasukkan Index \t: ";
        cin >> search;
        cout <<endl<<"Silsilah : ";
        int m, x = 0;
        jalurByIndex(keluarga, search, x, m);
        break;
    }
    case 4:{
        system("cls");
        char ulang;
        system("cls");
        cout <<"Apakah Anda Yakin Keluar Dari Program Ini? (Y/N) : ";
        cin >> ulang;
        if(ulang=='y' || ulang=='Y')
        {
            system("cls");
            cout <<"Terimakasih telah menggunakan program ini";
        }
        else
        {
            system("cls");
            menu();
        }
        break;
    }
    default:
        system("cls");
        cout <<"ERROR: Pilihan yang anda masukkan tidak ada!\n";
        cout <<"Tekan ENTER untuk mengulang program...";
        getch();
        system("cls");
        menu();
        break;
    }
}

void tambahRoot(){
    system("cls");
    cout <<"(MENU 1). TAMBAH KELUARGA"<< endl<< endl;
    cout << "Index Parrent \t: " << tmp <<endl;
    cout <<"Nama Root\t: ";
    gets(keluarga[tmp].nama);
    gets(keluarga[tmp].nama);
    tmp++;
    cout <<endl;
    tambahNode();
}

void tambahNode(){
    node:
    cout <<"Index Parrent \t: ";

```

```

cin >> keluarga[tmp].id_Ayah;
cout <<"Nama \t\t: ";
gets(keluarga[tmp].nama);
gets(keluarga[tmp].nama);
cout <<endl;
tmp++;

tambahNode:
cout <<"Ingin menambah lagi? (Y/N) :";
cin >> ulang;
cout <<endl;
if (ulang == 'y' | ulang == 'Y'){
    goto node;
}else if (ulang == 'n' | ulang == 'N'){
    main();
}else {
    goto tambahNode;
}
}

void lihatKeluarga(struct data keluarga[], int j, int n){
    if (j>=n){
        return;
    }else{
        printf("| %i\t| %s\t\t| %i\t\t|\n", j, keluarga[j].nama,
keluarga[j].id_Ayah);
        cout <<"-----" <<endl;
        lihatKeluarga(keluarga, j+1, n);
    }
}

void jalurByIndex(struct data keluarga[], int search, int x, int m){
    int idx = search;
    if(idx<=-1){
        strcpy(isi[x], keluarga[idx].nama);
        lihatJalur(x);
        return;
    }else{
        strcpy(isi[x], keluarga[idx].nama);
        search=keluarga[idx].id_Ayah;
        x++;
        jalurByIndex(keluarga, search, x, m);
    }
}

void lihatJalur(int x){
    if (x==0){
        cout <<isi[0];
        return;
    }else{
        cout <<isi[x];
        cout <<" -> ";
        lihatJalur(x-1);
    }
}

int main(){
    menu();
}

```

Proses Run pada Command Prompt :

- Pilihan Menu Program

```
D:\Modul9.exe
PERMASALAHAN MODUL KE-9
PROGRAM FAMILY TREE

PILIHAN MENU PROGRAM
1. Tambah Keluarga
2. Lihat Anggota Keluarga
3. Cari Silsilah Keturunan
4. Keluar

Masukkan Pilihan : _
```

- Menu Tambah Keluarga

```
D:\Modul9.exe
(MENU 1). TAMBAH KELUARGA

Index Parrent : -1
Nama Root : Rudi

Index Parrent : -1
Nama : Udin

Ingin menambah lagi? (Y/N) :y

Index Parrent : 0
Nama : Herya

Ingin menambah lagi? (Y/N) :y

Index Parrent : -1
Nama : Budi

Ingin menambah lagi? (Y/N) :y

Index Parrent : 1
Nama : Ardika

Ingin menambah lagi? (Y/N) :
```

- Menu Lihat Anggota Keluarga

```
D:\Modul9.exe
(MENU 2). LIHAT ANGGOTA KELUARGA

Root : Rudi
Index dari Rudi : -1

=====
| Index | Nama      | Index Parrent |
=====
| 0     | Udin     | -1            |
| 1     | Herya    | 0             |
| 2     | Budi     | -1            |
| 3     | Ardika   | 1             |
=====

Tekan ENTER untuk kembali....
```

- Menu Mencari dan Melihat Silsilah Berdasarkan Index

```
D:\Modul9.exe
(MENU 3). LIHAT KETURUNAN BY INDEX

Masukkan Index : 2

Silsilah : Rudi -> Budi

-----
Process exited after 212 seconds with return value 0
Press any key to continue . . .
```