

LAMPIRAN STUDI KASUS BINARY-TREE (DECISION TREE)



Disusun oleh :

Wendi Kardan

2100016

Pendidikan Ilmu Komputer – A

PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER FAKULTAS

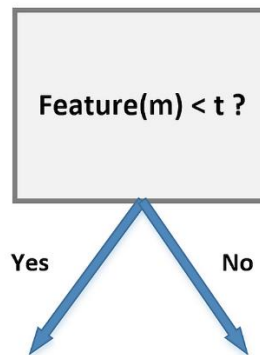
MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PENDIDIKAN INDONESIA

2022

A. PENJELASAN KASUS

Decision Tree atau pohon keputusan yang menggunakan binary tree, merupakan suatu proses keputusan yang berjalan secara sekuensia. Dari root kemudian terdapat fitur evaluasi yang dimana diminta untuk meminta memilih di antara 2 cabang antara kanan dan kiri. Proses tersebut akan terus dilakukan sampai mencapai daun yang terakhir yang merepresentasikan klasifikasi yang dicari dari decision tree ini.



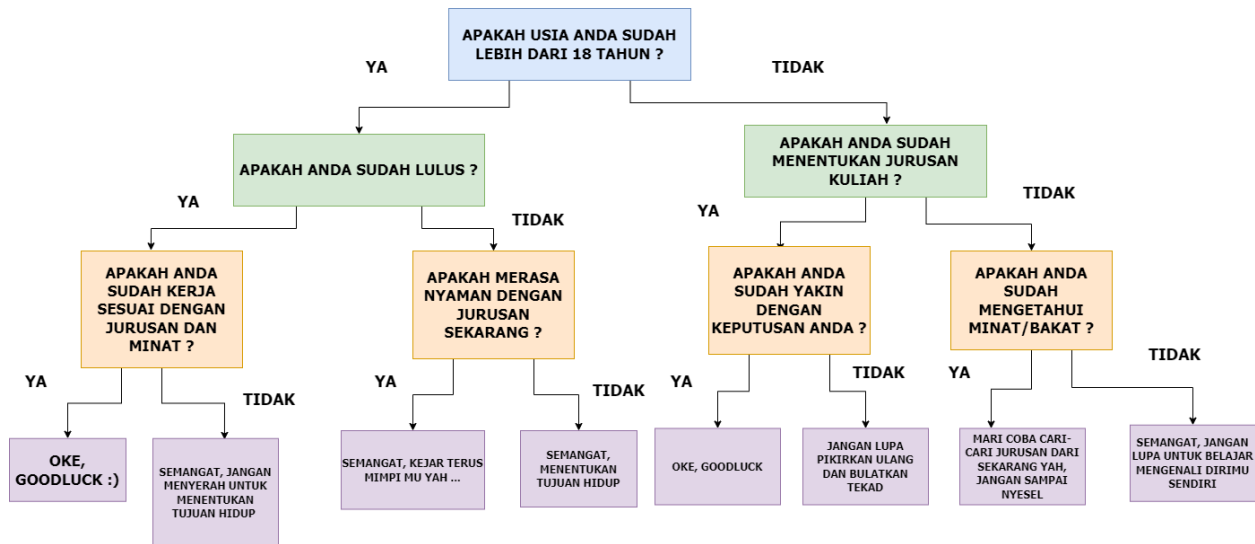
Kasus yang saya buat adalah decision tree tentang pertanyaan terkait karir atau rencana untuk kedepannya. Dimana setiap vertex dari tree tersebut nanti akan berisikan data-data pertanyaan. Adapun ADT yang diperlukan dalam struktur tree ini adalah :

Alamat child sebelah kiri (apabila Yes)	Pertanyaan (Yes/No)	Alamat child sebelah kanan (apabila No)
--	---------------------	--

Apabila proses pengambilan keputusan sudah selesai, maka nanti diakhir program nanti akan diberikan hasil dari pertanyaan yang sudah dijawab yang merupakan daun paling ujung dari tree tersebut.

B. SKEMA BINARY-TREE

Adapun skema dari decision tree menggunakan binary-tree yang nantinya akan dibuat adalah sebagai berikut :



C. PENJELASAN SOURCE CODE

1. ADT untuk vertex dari tree tersebut yang berisikan label pertanyaan, alamat child sebelah kanan dan sebelah kiri

```
// ADT untuk binary tree
typedef struct vertex{
    char label[100]; //label disini akan berisikan pertanyaan untuk setiap desicion yang dibuat
    struct vertex* right;
    struct vertex* left;
}Vertex;
```

2. ADT untuk tree yang nantinya akan menyimpan alamat vertex paling atas yang merupakan root

```
// ADT untuk tree yang berisikan root atau akar dari tree itu
typedef struct tree{
    Vertex *root;
}Tree;
```

3. Prosedur untuk membuat atau menginisialisasikan tree yang baru

```
// Menginisialisasikan tree baru untuk root yang nanti akan berisikan label/data
void createTree(char a[], Tree *T){
    Vertex *new;
    new = (Vertex *)malloc(sizeof(Vertex));
    strcpy(new->label, a);
    new->right = NULL;
    new->left = NULL;
    (*T).root = new;
}
```

4. Prosedur untuk menambahkan child baru baik sebelah kanan maupun sebelah kiri

```
// Menambahkan daun/vertex yang berada disebelah kanan
void addRight(char c[], Vertex *root){
    if(root->right == NULL){ //apabila sebelah kanan masih kosong
        Vertex *new = (Vertex *)malloc(sizeof(Vertex));
        strcpy(new->label, c);
        new->right = NULL;
        new->left = NULL;
        root->right = new;
    }else{ //apabila sebelah kanan sudah di isi
        printf("The right side of the tree already contain\n");
    }
}

// Menambahkan daun/vertex yang berada disebelah kiri
void addLeft(char c[], Vertex *root){
    if(root->left == NULL){ //apabila sebelah kiri masih kosong
        Vertex *new = (Vertex *)malloc(sizeof(Vertex));
        strcpy(new->label, c);
        new->right = NULL;
        new->left = NULL;
        root->left = new;
    }else{ //apabila sebelah kiri sudah terpenuhi
        printf("The left side of the tree already contain\n");
    }
}
```

5. Proses membuat struktur tree yang akan diisi dengan pertanyaan sesuai dengan skema tree yang sudah dibuat

```
// Proses penambahan pertanyaan kedalam struktur tree
Tree T;
createTree("Apakah usia mu diatas 18 tahun ? ", &T);
addLeft("Apakah anda sudah lulus dari sekolah/kuliah ? ", T.root);
addRight("Apakah anda sudah menentukan jurusan kuliah ?", T.root);
Vertex* atas18 = T.root->left;
Vertex* bawah18 = T.root->right;
addLeft("Apakah anda sudah merasa kerja sesuai dengan jurusan dan passion yang anda miliki ?", atas18);
addRight("Apakah sekarang anda merasa nyaman dengan jurusan kuliah anda ?", atas18);
addLeft("Fine, berarti anda berada di jalan yang tepat dalam hidup anda, semoga anda bahagia :)", atas18->left);
addRight("Semangat ! jangan putus asa, jangan menyerah untuk menentukan tujuan hidup anda", atas18->left);
addLeft("Semangat, mari bekerja keras untuk menggapai mimpi mu :)", atas18->right);
addRight("Mari semangat untuk menentukan tujuan hidup, rencanakan apa yang akan kau lakukan kedepannya", atas18->right);
addLeft("Apakah anda yakin dengan keputusan anda ? ", bawah18);
addRight("Apakah anda sudah menemukan minat dan bakat mu ? ", bawah18);
addLeft("Oke, Goodluck :) ", bawah18->left);
addRight("jangan lupa pikirkan ulang dan bulatkan tekad yah :)", bawah18->left);
addLeft("Mari coba cari-cari jurusan dari sekarang yah, jangan sampai nyesel :)", bawah18->right);
addRight("Semangat, jangan lupa untuk belajar untuk mengenali diri mu sendiri yah :)", bawah18->right);
```

- Proses input dari user untuk menentukan hasil dari decision tree yang sudah dibuat sampai menemukan akar dari tree tersebut yang merupakan jawaban dari decision yang sudah dibuat. Apabila user menjawab ya, akan mengarahkan ke child sebelah kiri, apabila tidak maka akan mengarahkan ke child yang sebelah kanan.

```
// Proses input pertanyaan dari user
int desicion;
printf("=====\n");
printf("== QUOTES DESICION ==\n");
printf("=====\n");

Vertex *question = T.root;

int j;

// perulangan untuk melakukan pertanyaan dan penyelusuran tree
// perulangan akan diulang sampai mencapai daun paling ujung dari tree tersebut
while(question != NULL && question != NULL){
    printf("\n%s ", question->label);
    if(question->left != NULL && question->right != NULL){
        printf("\n1. Ya\n2. Tidak\n");
        scanf("%d", &desicion);
        if(desicion == 1){
            question = question->left;
        }else{
            question = question->right;
        }
    }else{
        break;
    }
}
}
```

D. OUTPUT

Berdasarkan program yang sudah dibuat maka output yang dihasilkan dari program tersebut adalah :

Sample 1:

```
=====
== QUOTES DESICION ==
=====

Apakah usia mu diatas 18 tahun ?
1. Ya
2. Tidak
1

Apakah anda sudah lulus dari sekolah/kuliah ?
1. Ya
2. Tidak
1

Apakah anda sudah merasa kerja sesuai dengan jurusan dan passion yang anda miliki ?
1. Ya
2. Tidak
1

Fine, berarti anda berada di jalan yang tepat dalam hidup anda, semoga anda bahagia :)
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 13\Praktikum Binary Tree> █
```

Sample 2 :

```
=====
== QUOTES DESICION ==
=====

Apakah usia mu diatas 18 tahun ?
1. Ya
2. Tidak
1

Apakah anda sudah lulus dari sekolah/kuliah ?
1. Ya
2. Tidak
2

Apakah sekarang anda merasa nyaman dengan jurusan kuliah anda ?
1. Ya
2. Tidak
2

Mari semangat untuk menentukan tujuan hidup, rencanakan apa yang akan kau lakukan kedepanny
a
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 13\Praktikum Binary Tree> █
```

Sample 3 :

```
=====
== QUOTES DESICION ==
=====

Apakah usia mu diatas 18 tahun ?
1. Ya
2. Tidak
2

Apakah anda sudah menentukan jurusan kuliah ?
1. Ya
2. Tidak
2

Apakah anda sudah menemukan minat dan bakat mu ?
1. Ya
2. Tidak
2

Semangat, jangan lupa untuk belajar untuk mengenali diri mu sendiri yah :)
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 13\Praktikum Binary Tree> █
```

E. KESIMPULAN

Salah satu bentuk struktur data yang sering digunakan adalah tree, salah satu bentuk tree adalah binary tree dimana tree ini hanya dapat menampung 2 buah child saja yang berada disebelah kiri dan kanan. Struktur binary tree pada umumnya biasanya menampung data atau label dari vertex tersebut, kemudian alamat dari child kiri, dan alamat dari child kanan. Contoh implementasi dari binary tree ada banyak salah satunya adalah pohon keluarga (akan tetapi setiap parents hanya boleh memiliki 2 anak saja maksimal). Contoh studi kasus yang saya ambil disini adalah Decision Tree yang merupakan pohon keputusan yang nanti akan berisikan pertanyaan yang jawabannya adalah ya dan tidak. User nanti akan diberikan beberapa pertanyaan sesuai skema tree yang sudah dibuat, nanti setelah menjawab beberapa pertanyaan nanti akan muncul quotes yang merupakan decision yang sudah diisikan diinputan sebelumnya.