Pemanfaatan Pohon Keputusan dalam Menentukan Tempat Ternyaman untuk Mahasiswa Mengerjakan Tugas

Muhammad Fauzan Al-Ghifari 13518112
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518112@std.stei.itb.ac.id

***Abstract*—Pada masa perkuliahan mahasiswa seringkali mendapat banyak tugas. Mulai dari tugas kecil yang mudah untuk diselesaikan hingga tugas besar yang harus dikerjakan secara berkelompok. Untuk mengerjakan tugas-tugas tersebut, tentu mahasiswa membutuhkan lingkungan yang nyaman dan kondusif. Banyak mahasiswa memilih untuk mengerjakan tugas di kamar kosnya. Tetapi tidak sedikit juga mahasiswa yang tidak suka mengerjakan tugas di kamar dengan alasan mahasiswa tersebut seringkali tidak sengaja tertidur. Café atau Coworking Space sering menjadi alternatif yang dipilih untuk mengatasi permasalahan "ketiduran", selain itu Coworking space juga memiliki fasilitas yang menunjang seperti koneksi internet. Banyaknya Coworking space di lingkungan kampus terkadang membuat mahasiswa bingung tempat mana yang paling tepat untuknya mengerjakan tugas. Untuk membuat keputusan yang lebih tepat dan mudah, kita dapat mengaplikasikan pembelajaran Matematika Diskrit yaitu Decision Tree.**

***Keywords*—Decision, Tree, Tempat, Belajar.**

I. PENDAHULUAN

Graph sering digunakan untuk merepresentasikan sebuah objek dan hubungannya dengan objek lain. Graph pada dasarnya mempunyai komponen berupa simpul dan sisi sehingga diklasifikasikan menjadi graph terbuka dan graph tertutup. Graph terbuka memiliki kemungkinan untuk memiliki lintasan dan graph tertutup memiliki kemungkinan untuk memiliki sirkuit dan/atau lintasan.

Graph terhubung yang tidak mengandung sirkuit disebut pohon. Dari banyak konsep dalam teori graph, konsep pohon adalah yang paling banyak memiliki penerapan langsung pada kehidupan sehari-hari. Misalnya pohon silsilah keluarga, struktur organisasi, dan diagram pertandingan.

Salah satu penerapan konsep pohon adalah pohon keputusan (*decision tree*), yaitu pemodelan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi. Pohon keputusan dalam aturan keputusan (*decision rule*) merupakan metodologi data mining yang banyak diterapkan sebagai solusi untuk klasifikasi.

Pohon keputusan membantu manusia untuk mengambil sebuah keputusan dari beberapa pilihan yang tersedia berdasarkan spesifikasi yang diinginkan. Salah satu kegunaan dari pohon keputusan adalah dalam penentuan tempat yang kondusif dan nyaman bagi mahasiswa untuk menyelesaikan tugas-tugasnya. Dengan bantuan google maps dan *Decision Tree* permasalahan akan dengan mudah diselesaikan.

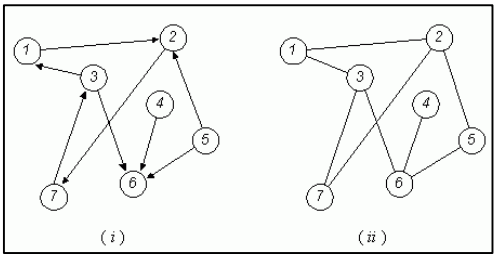
II. LANDASAN TEORI

A. Graph

Secara matematis graf didefinisikan sebagai himpunan (V, E), ditulis dengan notasi G = (V, E), yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*ventrices* atau *nodes*) dan E adalah sisi (*edges*) yang menghubungkan sepasang simpul. Sisi pada graf dapat mempunyai orientasi arah Berdasarkan orientasi arah pada sisinya, graf dikelompokkan menjadi dua.

1. Graf tak-berarah (*undirected graph*)
   Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Pada graf tak-berarah, urutan pasangan simpul yang dihubungkan oleh sis tidak diperhatikan. (u, v) = (v, u) adalah sisi yang sama

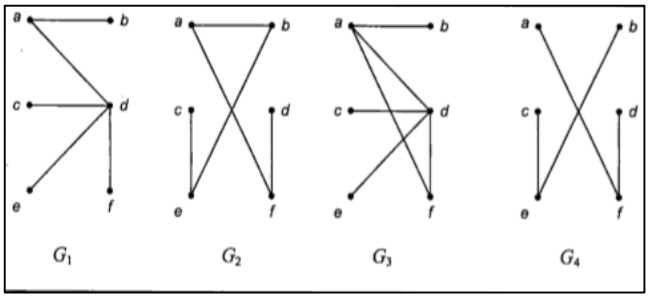
2. Graf berarah (*directed graph* atau *digraph*)
   Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Pada graf berarah (*u, v*) *dan* (*v, u*) menyatakan dua buah busur yang berbeda. Untuk busur (*u, v*) simpul *u* dinamakan simpul asal (*initial vertex*) dan simpul *v* dinamakan simpul terminal (*terminal vertex*).



Gambar 1. (i) Graf berarah (ii) graf tidak berarah
(Sumber: http://aimyaya.com/images/contohgraf.gif)

B. Pohon

Pohon adalah graf yang khusus, pohon termasuk graf tak-terhubung yang tidak mengandung sirkuit. Pada Gambar 2, hanya G_1 dan G_2 yang dapat dikategorikan sebagai pohon karena memenuhi karakteristik khusus pohon, sedangkan G_3 dan G_4 tidak tergolong sebagai pohon karena G_3 memiliki sirkuit a-d-f dan G_4 merupakan graf yang tidak terhubung.



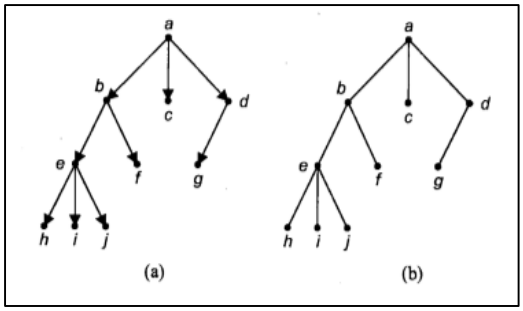
Gambar 2. G_1 dan G_2 Pohon, G_3 dan G_4 bukan pohon (Sumber: Munir, Rinaldi, Matematika Diskrit. Bandung: Informatika, Bandung, 2006.)

1. Sifat-sifat Pohon

   Misalkan *G = (V, E)* adalah graf tak-berarah sederhana dan jumlah simpulnya *n*. Maka pernyataan di bawah ini adalah benar:
   a. *G* adalah pohon.
   b. Setiap pasang simpul di dalam *G* terhubung dengan lintasan tunggal.
   c. *G* adalah terhubung dan memiliki *m = n-1* buah sisi
   d. *G* tidak mengandung sirkuit dan memiliki *m=n-1* buah sisi
   e. *G* tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit
   f. *G* terhubung dan semua sisinya adalah jembatan

2. Pohon Berakar

   Pohon yang sebuah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah menjauh dari akar dinamakan pohon berakar (*rooted tree*). Pada kebanyakan aplikasi pohon, simpul diperlakukan sebagai akar (*root*). Saat simpul sudah ditetapkan menjadi akar, maka simpul-simpul lainnya dapat dicapai dari akar dengan memberi arah pada sisi-sisi pohon yang mengikutinya.

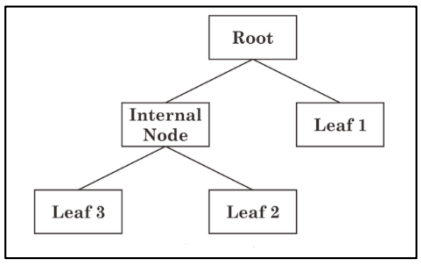


Gambar 3. Pohon Berakar (Sumber: Munir, Rinaldi, Matematika Diskrit. Bandung: Informatika, Bandung, 2006.)

Pada penggambaran pohonnya tanda panah yang menunjukkan arah pada pohon berakar dapat diabaikan. Gambar 3(b) menunjukkan pohon berakar 3(a) yang tanda panahnya sudah dihilangkan.

3. Pohon Keputusan

   Pohon keputusan merupakan suatu metode klasifikasi yang menggunakan struktur pohon, dimana setiap *node* menunjukkan atribut dan cabangnya menunjukkan nilai dari atribut, sedangkan daunnya digunakan untuk merepresentasikan kelas. *Node* teratas dari pohon keputusan disebut dengan akar.



Gambar 4. Representasi dari pohon keputusan. (Sumber: medium.com/@mimubarok.mim)

Metode pohon keputusan merupakan metode yang sangat populer untuk digunakan karena hasil yang diperoleh mudah untuk dipahami. Pada pohon keputusan terdapat tiga jenis node, antara lain:

a. Akar

   Merupakan *node* teratas, pada *node* ini tidak ada input dan dapt tidak mempunyai output atau mempunyai output lebih dari satu
b. *Internal node*

   Merupakan *node* percabangan, pada *node* ini hanya terdapat satu input dan mempunyai output minimal dua.
c. Daun

   Merupakan *node* akhir atau terminal *node*, pada *node* ini hanya terdapat satu input dan tidak mempunyai output (*terinal*).

Konsep dari pohon keputusan adalah serangkaian aturan if-then, pembentukan pohon keputusan terdiri dari beberapa tahap

a. Konstruksi pohon

   Diawali dengan pembentukan akar, yaitu node yang paling atas, lalu dilanjutkan dengan membagi berdasarkan atribut yang akan dijadikan daun atau pun internal node.
b. Pemangkasan pohon

   Mengidentifikasi dan membuang cabang yang tidak diperlukan pada pohon yang telah terbentuk. Jika pohon yang dikonstruksi terlalu besar makan dapat dilakukan pemangkasan pohon berdasarkan *confident level.*
c. Pembentukan aturan keputusan

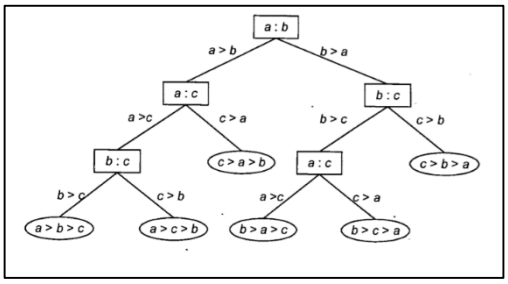
   Yaitu membuat aturan keputusan dari pohon yang telah dibentuk. Aturan tersebut dapat dalam bentuk if-then dan diturunkan sampai ke daun.

Kelebihan pohon keputusan:

a. Mudah menghubungkannya dengan basis data.
b. Memiliki tingkat ketelitian yang baik.
c. Daerah pengambilan keputusan menjadi sangat sederhana.
d. Dapat melakukan eliminasi untuk data yang tidak diperlukan.
e. Fleksibel dalam menentukan fitur.

Kekurangan pohon keputusan:

a. Meningkatnya waktu pengambilan keputusan ketika kriteria yang digunakan jumlahnya sangat banyak.
b. Sulit membuat desain pohon keputusan yang optimal
c. Hasil keputusan sangat bergantung kepada desain dari pohon keputusan yang dibuat.

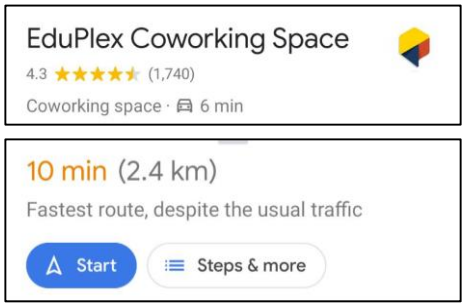


Gambar 5. Pohon keputusan untuk mengurutkan 3 buah integer
(Sumber: Munir, Rinaldi, Matematika Diskrit. Bandung: Informatika, Bandung, 2006.)

III. DATA *COWORKING SPACE* DI BANDUNG

Penulis menggunakan aplikasi *Google Maps* untuk mendapat data jarak dan *rating* dari *coworking space* dan menggunakan aplikasi *Go-Food* untuk memeriksa harga makanan atau biaya minimal yang harus dikeluarkan untuk dapat menggunakan *coworking space* tersebut.

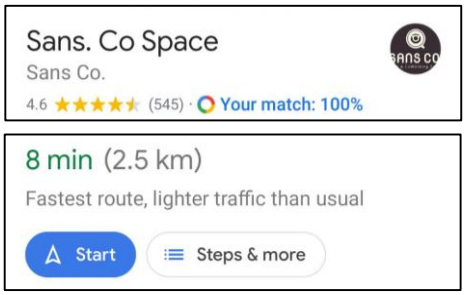
1. Eduplex



Gambar 6. Eduplex
(Sumber: maps.google.com)

Comfort Level : 4.3 *Star*
Jarak : 2.4 Km
Harga : Rp. 18.000, -

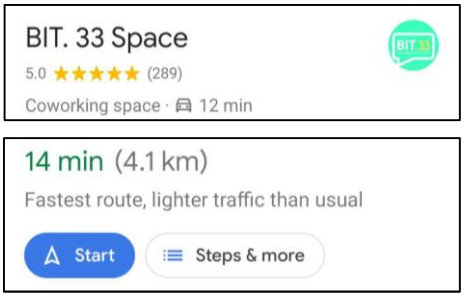
2. Sans Co Space



Gambar 7. Sans Co Space
(Sumber: maps.google.com)

Comfort Level : 4.6 *Star*
Jarak : 2.5 Km
Harga : Rp. 19.000, -

3. Bit 33 Space

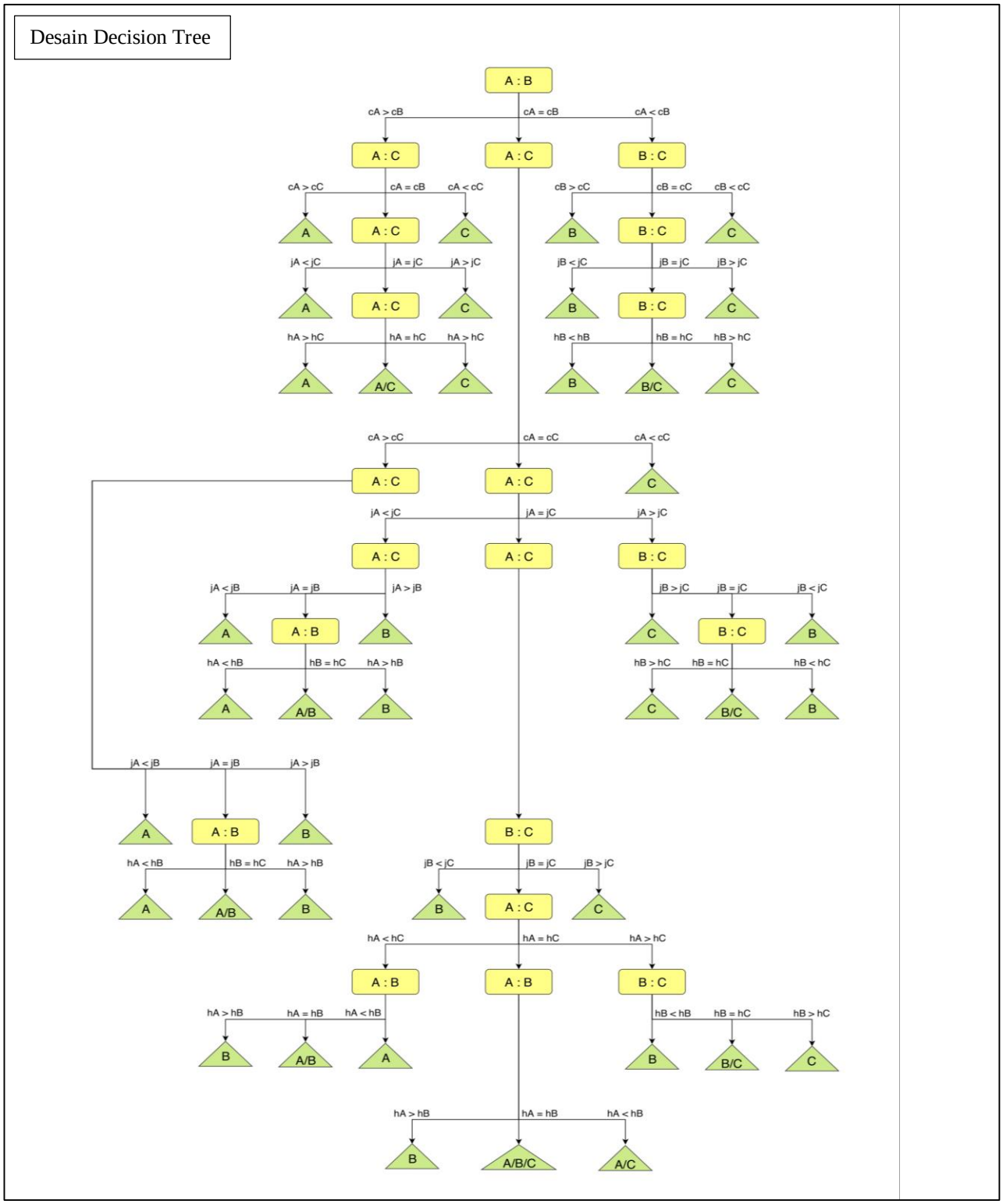


Gambar 8. BIT 33 Space
(Sumber: maps.google.com)

Comfort Level : 5.0 *Star*
Jarak : 4.1 Km
Harga : Rp. 20.000, -

IV. PEMBAHASAN

Penulis mengambil 3 buah sample *coworking space* yang paling sering dikunjungi oleh mahasiswa, yaitu Eduplex, San Co Space, dan Bit 33 Space. Penulis menggunakan tiga buah indikator untuk mengurutkan ketiga buah *coworking space* tersebut, yaitu *confort level* yang diambil dari rating pengunjung, jarak dari kampus dan harga minimum yang harus dikeluarkan perorang untuk dapat menikmati fasilitas yang diberikan oleh *coworking space*. Urutan prioritas yang penulis pilih adalah *comfort level*, Jarak dan yang terakhir adalah harga.



Gambar 9. Desain Decision Tree
(Sumber: Penulis)

Keterangan:
1. Segitiga hijau adalah daun
2. kotak kuning adalah *internal node*.
3. c = *comfort level*, j = jarak, h = harga
4. jA artinya jarak *coworoking space* A dsb.

Bandingkan A:B

1. Jika A lebih nyaman dari B, bandingkan A:C
   A. Jika A lebih nyaman dari C maka pergi ke A
   B. Jika A sama nyamannya dengan C maka bandingkan lagi A:C
      1) Jika A lebih dekat, pergi ke A
      2) Jika A jaraknya sama dengan C, bandingkan lagi A:C
         a) Jika A lebih murah, pergi ke A
         b) Jika harga A dan C sama, Anda bisa pergi ke A atau ke C
         c) Jika C lebih murah, pergi ke C
      3) Jika C jaraknya lebih dekat maka pergi ke C
   C. Jika C lebih nyaman dari A maka pergi ke C

2. Jika A dan B sama kenyamanannya bandingkan A:C
   A. Jika A lebih nyaman dari C, bandingnya A:B
      1) Jika jarak A lebih dekat pergi ke A
      2) Jika jarak A dan B sama, bandingkan A:B
         a) Jika A lebih murah, pergi ke A
         b) Jika A dan B sama harganya, anda bisa pergi ke A atau ke B
         c) Jika B lebih murah, pergi ke B
      3) Jika jarak B lebih dekat, pergi ke B

   B. Jika A dan C sama tingkat kenyamanannya bandingkan A:C
      1) Jika jarak A lebih dekat bandingkan A:C
         a) Jika jarak A lebih dekat, pergi A
         b) Jika jarak A dan B sama, bandingkan A:B
            A) Jika harga A lebih murah, pergi ke A
            B) Jika harga A dan B sama, anda bisa pergi ke A atau B
            C) Jika harga B lebih murah, pergi ke B
         c) Jika jarak B lebih dekat, pergi ke B
      2) Jika jarak A dan C sama, bandingkan B:C
         a) Jika jarak B lebih dekat, pergi ke B
         b) Jika jarak B dan C sama, bandingkan A:C
            A) Jika A lebih murah dari C bandingkan A:B
               1. Jika A lebih murah, pergi ke A
               2. Jika A dan B harganya sama, anda dapat pergi ke A atau B
               3. Jika B lebih murah, pergi ke B
            B) Jika harga A sama dengan harga C bandingkan A:B
               1. Jika harga A lebih murah, anda dapat pergi ke A atau C
               2. Jika harga A dan B sama, maka anda dapat pergi ke A, B atau C
               3. Jika harga B lebih murah, pergi ke B
            C) Jika harga C lebih murah dari C, bandingkan B:C
               1. Jika harga B lebih murah, pergi ke B
               2. Jika harga B dan C sama, anda dapat pergi ke B atau C
               3. Jika C lebih murah, pergi ke C
         c) Jika jarak B lebih dekat pergi ke C
      3) Jika jarak C lebih dekat, badningkan B:C
   C. Jika C lebih nyaman dari A, pergi ke C

3. Jika B lebih nyaman dari A, bandingkan B:C
   A. Jika B lebih nyaman, pergi ke B
   B. Jika B dan C sama nyamannya, bandingkan B:C
      1) Jika B lebih dekat, pergi ke B
      2) Jika B dan C sama jaraknya, bandingkan B:C
         a) Jika B harganya lebih murah, pergi ke B
         b) Jika B dan C memiliki harga yang sama, Anda bisa pergi ke B atau C
         c) Jika C harganya lebih murah pergi ke C
      3) Jika C lebih dekat, pergi ke C
   C. Jika C lebih nyaman, pergi ke C

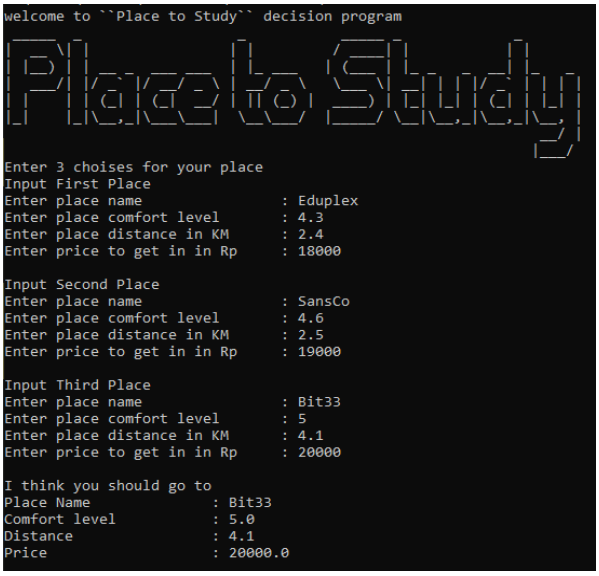
Jika penulis memasukkan parameter, parameter *comfort level*, jarak dan harga dari Eduplex, SansCo space dan bit 33 Space maka berdasarkan design pohon keputusan di atas *output* yang diterima adalah Bit 33 Space.

Untuk mempermudah penggunaan pohon keputusan di atas penulis juga menyediakan algoritmanya dalam bahasa C yang dapat dilihat pada https://github.com/mfauzanalg/DecisionTree.

Berikut adalah sedikit cuplikan dari source code dan cara menggunakan program yang dibuat oleh penulis.

```
printf("I think you should go to \n");
if (A.comfort > B.comfort && A.comfort > C.comfort){
    Print(A);
}
else if (B.comfort > A.comfort && B.comfort > C.comfort){
    Print(B);
}
else if (C.comfort > A.comfort && C.comfort > B.comfort){
    Print(C);
}
else if (A.comfort == B.comfort && A.comfort > C.comfort){
    if(A.distance < B.distance){
        Print(A);
    }
    else if (A.distance > B.distance){
        Print(B);
    }
    else{ //comfort level and distance are the same between A and B
        if(A.price < B.price){
            Print(A);
        }
        else{
            Print(B);
        }
    }
}
else if (A.comfort == C.comfort && A.comfort > B.comfort){
```

Gambar 10. Algoritma dari decision tree (hanya sedikit potongan, untuk lengkapnya kunjungi link github) (Sumber: Penulis)



Gambar 11. Hasil output dari program
(Sumber: Penulis)

V. KESIMPULAN

Pohon adalah graf khusus yang memiliki banyak kegunaan dalam kehidupan manusia. Salah satu konsep pohon yang paling sering digunakan adalah konsep pohon keputusan. Dengan menggunakan pohon keputusan, manusia dapat membuat pilihan yang kompleks menjadi lebih sederhana dengan mempertimbangkan parameter-parameter yang terdapat pada pilihan terkait.

Salah satu contoh penggunaan konsep pohon keputusan adalah dalam pemilihan tempat yang kondusif dan nyaman untuk mahasiswa mengerjakan tugas. Pohon keputusan dalam mempermudah *user* dalam menentukan tempat berdasarkan tingkat kenyamanan, jarak dan harga.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunianya makalah berjudul “Pemanfaatan Pohon Keputusan dalam Menentukan Tempat Ternyaman untuk Mahasiswa Mengerjakan Tugas” dapat penulis selesaikan. Penulis mengucapkan terima kasih sebesar-besarnya kepada Dr. Ir. Rinaldi Munir, MT. yang telah membimbing saya dalam memahami mata kuliah matematika diskrit. Tidak lupa juga penulis ucapkan kepada keluarga dan teman-teman teknik informatika ITB 2018, terutama teman-teman di K1 yang sudah membuat perkuliahan matematika diskrit menjadi lebih menyenangkan.

REFERENSI

[1] Rinaldi, Munir, Matematika Diskrit, 3rd ed. Bandung: Penerbit INFORMATIKA Bandung, 2010.
[2] Tim penulis http://aimyaya.com/images/contohgraf.gif Diakses pada 3 Desember pukul 22.00 GMT+7
[3] Tim penulis https://medium.com/@mimubarok.mim/decision-tree-pohon-keputusan-6484ad30c289 Diakses pada 3 Desember pukul 22.10 GMT+7
[4] Tim penulis https://mti.binus.ac.id/2018/03/05/teori-graph-sejarah-dan-manfaatnya/ Diakses pada 4 Desember pukul 13.30 GMT+7
[5] Breiman, L., Friedman, JH., Olshen, RA., Stone, CJ., 1984, Classification and Regression Trees, Chapman &Hall/CRC, New York.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 4 Desember 2019

Muhammad Fauzan Al-Ghifari
13518112