

# Aplikasi Pohon Keputusan dalam Membantu Mahasiswa Institut Teknologi Bandung Tingkat 1 untuk Memilih Jurusan

Fikra Hadi Ramadhan, 13518036  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13518036@std.stei.itb.ac.id

**Abstract**—Masih banyaknya mahasiswa Institut Teknologi Bandung yang bingung dan ragu dalam memilih jurusannya setelah selama setahun melaksanakan program TPB (Tahap Persiapan Bersama). Dengan menggunakan pohon keputusan, bisa setidaknya sedikit membantu mahasiswa ITB dalam memilih jurusannya.

**Keywords**—Pohon Keputusan, TPB, Jurusan, ITB.

## I. PENDAHULUAN

Seiring berjalannya waktu, Institut Teknologi Bandung pun juga menambah program studi / jurusannya. Akibatnya, beberapa mahasiswa Institut Teknologi Bandung menjadi bingung dan ragu dalam memilih jurusannya. Bahkan ada jurusan yang baru di masukan menjadi salah satu jurusan di ITB disaat mahasiswa sedang melaksanakan Tahap Persiapan Bersama (TPB).

Bertambahnya jurusan, maka lebih banyak pilihan jurusan yang bisa dipilih. Tetapi banyak factor yang harus kita pertimbangkan, baik itu dalam kurikulumnya, mata kuliahnya, dan juga prospek kerja dari jurusannya. Untuk itu dibutuhkan algoritma yang tepat untuk memutuskan memilih jurusan bagi mahasiswa Institut Teknologi Bandung. Dengan adanya pohon keputusan ini, diharapkan bisa membantu mahasiswa Institut Teknologi Bandung dalam memilih jurusannya.

## II. TEORI DASAR

### 2.1 Graf

Dalam matematika dan ilmu computer, teori graf adalah cabang yang mempelajari tentang sifat-sifat graf itu sendiri. Dalam pengertiannya, graf adalah himpunan benda-benda yang disebut simpul (*vertex* atau *node*) yang dihubungkan atau terhubung oleh sisi (*edge*) atau busur (*arc*). Biasanya graf digambarkan dalam bentuk titik-titik yang melambangkan simpul dan dihubungkan oleh garis garis yang melambangkan sisi atau garis berpanah yang melambangkan busur. Sisi bisa menghubungkan suatu simpul dengan simpul yang sama, disebut dengan gelang (*loop*).

Banyak sekali struktur yang bisa direpresentasikan dengan graf, dan banyak masalah yang bisa diselesaikan dengan

bantuan graf. Jaringan persahabatan pada [Facebook](#) bisa direpresentasikan dengan graf, yakni simpul-simpulnya adalah para pengguna Facebook dan ada sisi antar pengguna jika dan hanya jika mereka berteman. Perkembangan [algoritma](#) untuk menangani graf akan berdampak besar bagi [ilmu komputer](#).

Sebuah struktur graf bisa dikembangkan dengan memberi bobot pada tiap sisi. Graf berbobot dapat digunakan untuk melambangkan banyak konsep berbeda. Sebagai contoh jika suatu graf melambangkan jaringan jalan maka bobotnya bisa berarti panjang jalan maupun batas kecepatan tertinggi pada jalan tertentu. Ekstensi lain pada graf adalah dengan membuat sisinya berarah, yang secara teknis disebut [graf berarah](#) atau [digraf](#) (*directed graph*). Digraf dengan sisi berbobot disebut [jaringan](#).

Jaringan banyak digunakan pada cabang praktis teori graf yaitu [analisis jaringan](#). Perlu dicatat bahwa pada analisis jaringan, definisi kata "jaringan" bisa berbeda, dan sering berarti graf sederhana (tanpa bobot dan arah).

Formalnya, suatu graf misalkan  $G$  dapat dinyatakan sebagai berikut:

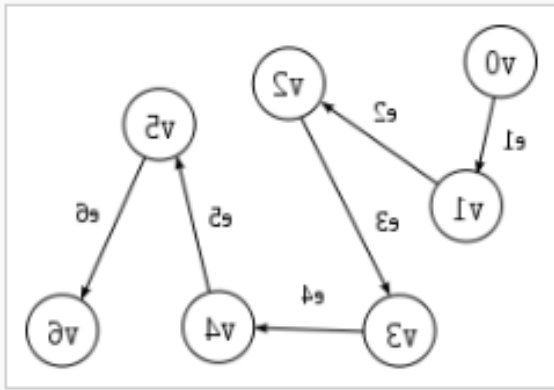
$$G = \langle V, E \rangle$$

Graph  $G$  terdiri atas himpunan  $V$  yang berisikan simpul pada graf tersebut dan himpunan dari  $E$  yang berisi sisi pada graf tersebut. Himpunan  $E$  dinyatakan sebagai pasangan dari simpul yang ada dalam  $V$ . Sebagai contoh definisi dari graf pada gambar di atas adalah :

$$V = \{1,2,3,4\}$$

$$E = \{(1,2), (1,4), (2,4), (3,4)\}$$

Pada [digraf](#) maka pasangan-pasangan ini merupakan pasangan terurut. Untuk menyatakan digraf (gambar kedua yang menggunakan tanda panah) kita dapat menggunakan himpunan *edge* sebagai berikut :

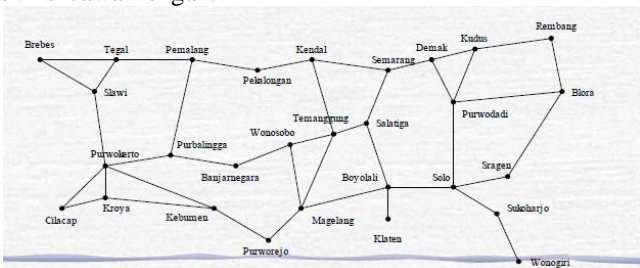


Gambar 2.1.1 Graf yang sama tetapi dalam bentuk digraf  
 Sumber : <http://danysatriokintoko.blogspot.com/2013/02/teori-dasar-graf.html>

Sebuah struktur graf bisa dikembangkan dengan memberi bobot pada tiap sisi. Graf berbobot dapat digunakan untuk melambangkan banyak konsep berbeda. Sebagai contoh jika suatu graf melambangkan jaringan jalan maka bobotnya bisa berarti panjang jalan maupun batas kecepatan tertinggi pada jalan tertentu. Ekstensi lain pada graf adalah dengan membuat sisinya berarah, yang secara teknis disebut graf berarah atau digraf (*directed graph*). Digraf dengan sisi berbobot disebut jaringan.

Jaringan banyak digunakan pada cabang praktis teori graf yaitu analisis jaringan. Perlu dicatat bahwa pada analisis jaringan, definisi kata "jaringan" bisa berbeda, dan sering berarti graf sederhana (tanpa bobot dan arah).

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Seperti contoh dibawah ini merupakan sebuah graf yang menyatakan peta jaringan jalan raya yang menghubungkan sejumlah kota di Provinsi Jawa Tengah.



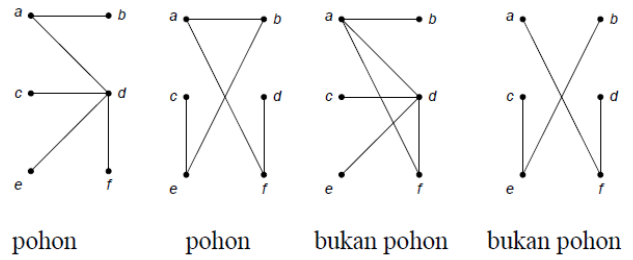
Gambar 2.1.2 Graf jaringan jalan raya di Jawa Tengah  
 Sumber : Diktat Matematika Diskrit

2.2 Pohon

Pohon (tree) telah digunakan sejak tahun 1857 oleh matematikawan Inggris yang bernama Arthur Cayley untuk menghitung jumlah senyawa kimia. Sejak ditemukannya teori tentang Pohon, banyak permasalahan yang diselesaikan menggunakan pohon dalam berbagai disiplin ilmu.

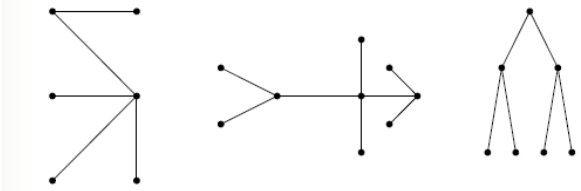
Pohon adalah graf yang tak berarah, terhubung yang tidak mengandung sirkuit, tapi bisa saja mengandung lintasan.

Diagram pohon dapat digunakan sebagai alat untuk memecahkan masalah dengan menggambarkan semua alternatif penyelesaian



Gambar 2.2.1 Ilustrasi yang merupakan pohon dan tidak  
 Sumber : Diktat Matematika Diskrit

Hutan (*forest*) adalah kumpulan pohon yang saling lepas atau suatu graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon.

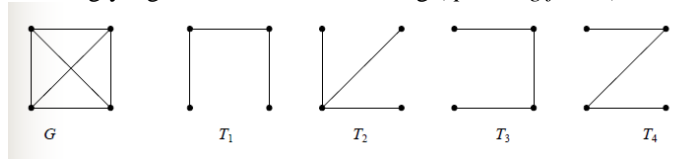


Gambar 2.2.2 Hutan yang terdiri dari tiga pohon  
 Sumber : Diktat Matematika Diskrit

Sifat-sifat pohon adalah sebagai berikut. Misalkan  $G = (V,E)$  adalah graftak-berarah sederhana dan jumlah simpulnya  $n$ . Maka, semua pernyataan di bawah ini adalah ekuivalen:

- 2.2.1  $G$  adalah pohon.
- 2.2.2 Setiap pasang simpul di dalam  $G$  terhubung dengan lintasan tunggal.
- 2.2.3  $G$  terhubung dan memiliki  $m = n - 1$  buah sisi.
- 2.2.4  $G$  tidak mengandung sirkuit dan memiliki  $m = n - 1$  buah sisi.
- 2.2.5  $G$  tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
- 2.2.6  $G$  terhubung dan semua sisinya adalah jembatan.

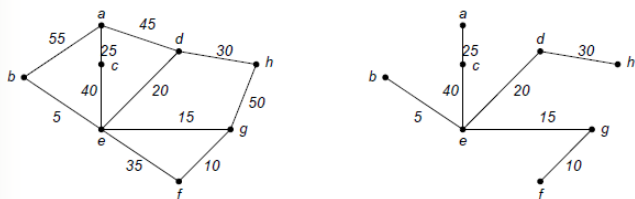
Pohon merentang dari graf terhubung adalah upagraf merentang yang berupa pohon. Pohon merentang diperoleh dengan memotong sirkuit di dalam graf. Setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang. Graf tak-terhubung dengan  $k$  komponen mempunyai  $k$  buah hutan merentang yang disebut hutan merentang (*spanning forest*).



Gambar 2.2.3 Graf  $G$  yang diputus salah satu sirkuitnya menyebabkan mengubahnya menjadi pohon merentang  
 Sumber : Diktat Matematika Diskrit

Aplikasi dari pohon merentang salah satunya adalah untuk menghitung jumlah ruas jalan seminimum mungkin yang menghubungkan semua kota sehingga setiap kota tetap terhubung satu sama lain. Selain itu juga sebagai perutean (*routing*) pesan pada jaringan computer.

Graf terhubung-berbobot mungkin mempunyai lebih dari 1 pohon merentang. Pohon merentang yang berbobot minimum dinamakan pohon merentang minimum (*minimum spanning tree*).



Gambar 2.2.4 Contoh Pohon Merentang Minimum  
Sumber : Diktat Matematika Diskrit

Untuk menentukan bobot minimum pada pohon merentang, bisa menggunakan Algoritma Prim. Algoritma Prim adalah sebuah algoritma dalam teori graf untuk mencari pohon rentang minimum untuk sebuah graf berbobot yang saling terhubung. Ini berarti bahwa sebuah himpunan bagian dari edge yang membentuk suatu pohon yang mengandung node, di mana bobot keseluruhan dari semua edge dalam pohon diminimalisasikan. Bila graf tersebut tidak terhubung, maka graf itu hanya memiliki satu pohon rentang minimum untuk satu dari komponen yang terhubung.

Langkah-langkah algoritma prim adalah sebagai berikut:

- Langkah 1 : ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T.
- Langkah 2 : pilih sisi (u,v) yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi (u,v) tidak membentuk sirkuit di T. Masukkan (u,v) ke dalam T.
- Langkah 3 : ulangi langkah 2 sebanyak  $n - 2$  kali.

```

procedure Prim(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
i, p, q, u, v : integer
Algoritma
Cari sisi (p,q) dari E yang berbobot terkecil
T ← {(p,q)}
for i←1 to n-2 do
  Pilih sisi (u,v) dari E yang bobotnya terkecil namun bersisian dengan simpul di T
  T ← T ∪ {(u,v)}
endfor

```

Gambar 2.2.5 Algoritma Prim  
Sumber : Diktat Matematika Diskrit

Selain algoritma prim, ada juga Algoritma Kruskal. Algoritma Kruskal adalah algoritma untuk mencari pohon merentang minimum secara langsung didasarkan pada algoritma MST (*Minimum Spanning Tree*) umum. Pada algoritma Kruskal sisi-sisi di dalam graf diurut terlebih dahulu berdasarkan bobotnya dari kecil ke besar. Sisi yang dimasukkan ke dalam himpunan T adalah sisi graf G sedemikian sehingga T adalah pohon. Pada

keadaan awal, sisi-sisi sudah diurut berdasarkan bobot membentuk hutan (*forest*). Hutan tersebut dinamakan hutan merentang (*spanning forest*). Sisi dari graf G ditambahkan ke T jika tidak membentuk sirkuit di T.

Perbedaan prinsip antara algoritma Prim dan Kruskal adalah jika pada algoritma Prim sisi yang dimasukkan ke dalam T harus bersisian dengan sebuah simpul di T, maka pada algoritma Kruskal sisi yang dipilih tidak perlu bersisian dengan simpul di T asalkan penambahan sisi tersebut tidak membentuk sirkuit.

Langkah-langkah Algoritma Kruskal adalah sebagai berikut:

- Langkah 1 : T masih kosong
- Langkah 2 : pilih sisi (u,v) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (u,v) ke dalam T.
- Langkah 3 : ulangi langkah 2 sebanyak  $n - 1$  kali.

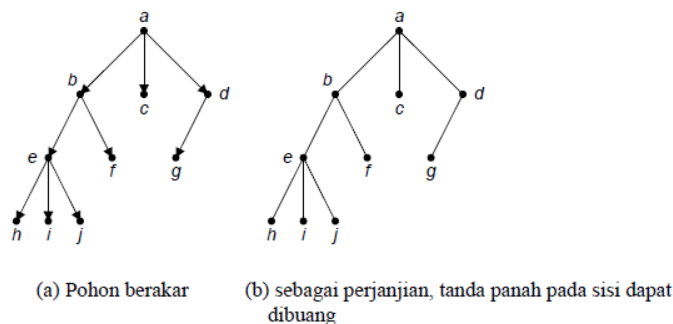
```

procedure Kruskal(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
i, p, q, u, v : integer
Algoritma
( Asumsi: sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya - dari bobot kecil ke bobot besar)
T ← {}
while jumlah sisi T < n-1 do
  Pilih sisi (u,v) dari E yang bobotnya terkecil
  if (u,v) tidak membentuk sirkuit di T then
    T ← T ∪ {(u,v)}
  endif
endfor

```

Gambar 2.2.6 Algoritma Kruskal  
Sumber : Diktat Matematika Diskrit

Pohon berakar (*rooted tree*) yaitu pohon yang satu buah simpulnya diperlakukan sebagai akar, dan sisi-sisinya diberi arah sehingga menjadi graf berarah.



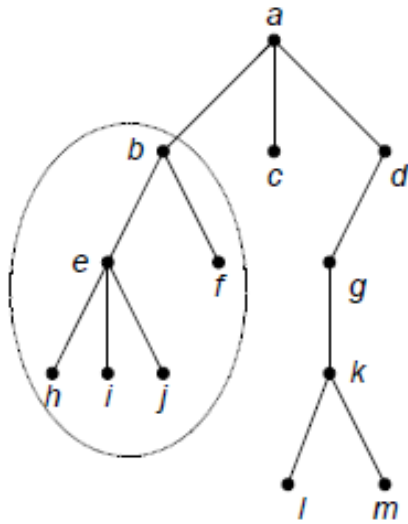
Gambar 2.2.7 Pohon berakar  
Sumber : Diktat Matematika Diskrit

Terminologi pada pohon berakar:

- Anak (*child* atau *children*) dan Orangtua (*parent*)  
*b, c,* dan *d* adalah anak-anak simpul *a*, *a* adalah orangtua dari anak-anak itu.
- Lintasan (*path*)  
Lintasan dari *a* ke *j* adalah *a, b, e, j*.  
Panjang lintasan dari *a* ke *j* adalah 3.
- Saudara Kandung (*sibling*)

$f$  adalah saudara kandung  $e$ , tetapi  $g$  bukan saudara kandung  $e$ , karena orangtua mereka berbeda.

- Upapohon (*subtree*)



Gambar 2.2.8 Menunjukkan upapohon dari suatu pohon  
 Sumber : Diktat Matematika Diskrit

- Derajat (*degree*)

Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut. Derajat  $a$  adalah 3, derajat  $b$  adalah 2, derajat  $d$  adalah 1, derajat  $c$  adalah 0.

Jadi, derajat yang dimaksud adalah derajat-keluar. Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri.

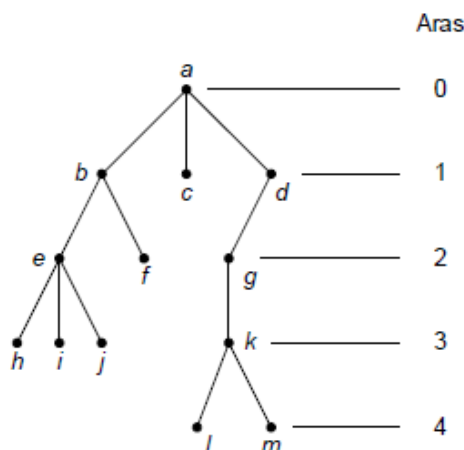
- Daun (*leaf*)

Simpul yang berderajat nol (atau tidak mempunyai anak) disebut daun. Simpul  $h, i, j, f, c, l,$  dan  $m$  adalah daun.

- Simpul Dalam (*internal nodes*)

Simpul yang mempunyai anak disebut simpul dalam. Simpul  $b, d, e, g,$  dan  $k$  adalah simpul dalam.

- Aras (*level*) atau Tingkat



Gambar 2.2.9 Aras pada pohon  
 Sumber : Diktat Matematika Diskrit

- Tinggi (*height*) atau Kedalaman (*depth*)

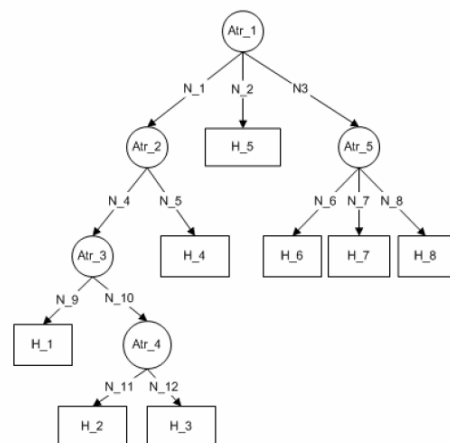
Aras maksimum dari suatu pohon disebut tinggi atau kedalaman pohon tersebut.

### III. POHON KEPUTUSAN

Pohon keputusan adalah salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi oleh manusia. Pohon keputusan adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari pohon keputusan adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan. Manfaat utama dari penggunaan pohon keputusan adalah kemampuannya untuk mem-break down proses pengambilan keputusan yang kompleks menjadi lebih simpel sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan.

Pohon Keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target. Pohon keputusan memadukan antara eksplorasi data dan pemodelan, sehingga sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain. Sering terjadi tawar-menawar antara keakuratan model dengan transparansi model. Dalam beberapa aplikasi, akurasi dari sebuah klasifikasi atau prediksi adalah satu-satunya hal yang ditonjolkan, misalnya sebuah perusahaan direct mail membuat sebuah model yang akurat untuk memprediksi anggota mana yang berpotensi untuk merespon permintaan, tanpa memperhatikan bagaimana atau mengapa model tersebut bekerja.

Pohon yang dalam analisis pemecahan masalah pengambilan keputusan adalah pemetaan mengenai alternatif-alternatif pemecahan masalah yang dapat diambil dari masalah tersebut. Pohon tersebut juga memperlihatkan faktor-faktor kemungkinan/probabilitas yang akan mempengaruhi alternatif-alternatif keputusan tersebut, disertai dengan estimasi hasil akhir yang akan didapat bila kita mengambil alternatif keputusan tersebut.



Gambar 3.1 Contoh Pohon Keputusan  
 Sumber :

<https://fairuzelsaid.wordpress.com/2009/11/24/data-mining-konsep-pohon-keputusan/>

IV. APLIKASI POHON KEPUTUSAN DALAM MEMBANTU MAHASISWA INSTITUT TEKNOLOGI BANDUNG UNTUK MEMILIH JURUSAN

Banyaknya pilihan program studi dan jurusan yang ditawarkan oleh Institut Teknologi Bandung, sering kali membuat mahasiswa Institut Teknologi Bandung kebingungan dan ragu dalam memilih. Walaupun sudah banyak faktor-faktor spesifik yang dimasukkan dalam memilih, namun terkadang pencarian tidak mengeluarkan hasil sesuai dengan apa yang diharapkan. Keadaan ini dapat mengkhawatirkan, banyak mahasiswa yang nantinya akan stress karna tidak cocok dengan jurusan yang dia pilih.

Untuk mempermudah melaksanakan pencarian manakah jurusan yang cocok untuk setiap mahasiswa, dibutuhkan sistem operasi data yang tepat dan sangat spesifik, mengingat saat melakukan pendaftaran untuk masuk jurusan, IPK adalah hal yang sangat dipakai.

Kelompok jurusan-jurusan di Institut Teknologi Bandung berdasarkan fakultas/sekolah adalah sebagai berikut:

4.1 STEI

- 4.1.1 Teknik Informatika
- 4.1.2 Teknik Elektro
- 4.1.3 Teknik Tenaga Listrik
- 4.1.4 Sistem dan Teknologi Informasi
- 4.1.5 Teknik Telekomunikasi
- 4.1.6 Teknik Biomedis

4.2 SAPPK

- 4.2.1 Arsitektur
- 4.2.2 Perencanaan Wilayah dan Kota

4.3 SF

- 4.3.1 Farmasi Klinik dan Komunitas
- 4.3.2 Sains dan Teknologi Farmasi

4.4 SBM

- 4.4.1 Kewirausahaan
- 4.4.2 Manajemen

4.5 SITH R

- 4.5.1 Rekayasa Hayati
- 4.5.2 Rekayasa Pertanian
- 4.5.3 Rekayasa Kehutanan
- 4.5.4 Teknologi Pasca Panen

4.6 SITH S

- 4.6.1 Biologi
- 4.6.2 Mikrobiologi

4.7 FITB

- 4.7.1 Meteorologi
- 4.7.2 Oseanografi
- 4.7.3 Teknik Geodesi dan Geomatika
- 4.7.4 Teknik Geologi

4.8 FTI

- 4.8.1 Manajemen Rekayasa Industri
- 4.8.2 Teknik Bionergi dan Kemurgi
- 4.8.3 Teknik Fisika

- 4.8.4 Teknik Industri
- 4.8.5 Teknik Kimia
- 4.8.6 Teknik Pangan

4.9 FTSL

- 4.9.1 Rekayasa Infrastruktut Lingkungan
- 4.9.2 Teknik dan Pengelolaan Sumber Daya Air
- 4.9.3 Teknik Kelautan
- 4.9.4 Teknik Lingkungan
- 4.9.5 Teknik Sipil

4.10 FTMD

- 4.10.1 Aeronotika dan Astronotika
- 4.10.2 Teknik Material
- 4.10.3 Teknik Mesin

4.11 FMIPA

- 4.11.1 Astronomi
- 4.11.2 Fisika
- 4.11.3 Kimia
- 4.11.4 Matematika

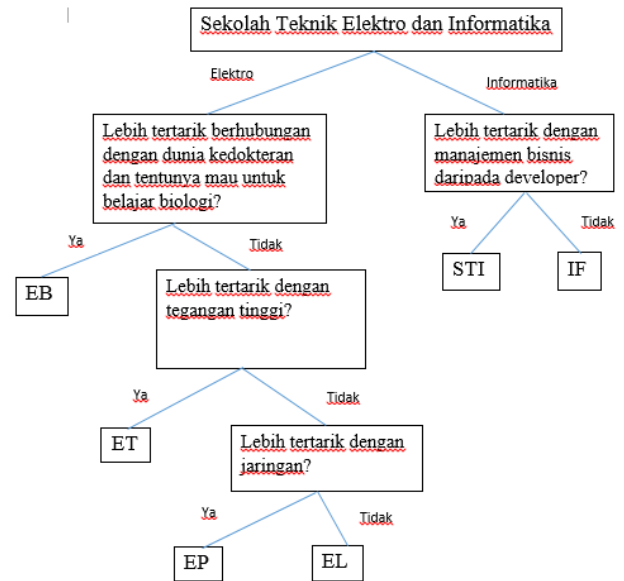
4.12 FSRD

- 4.12.1 Desain Interior
- 4.12.2 Desain Komunikasi Visual
- 4.12.3 Desain Produk
- 4.12.4 Kriya
- 4.12.5 Seni Rupa

4.13 FTTM

- 4.13.1 Teknik Geofisika
- 4.13.2 Teknik Metalurgi
- 4.13.3 Teknik Perminyakan
- 4.13.4 Teknik Pertambangan

Sebagai contoh, akan diambil jurusan-jurusan yang ada di STEI. Pohon Keputusannya adalah sebagai berikut



Gambar 4.1 Pohon Keputusan STEI

## V. KESIMPULAN

Dengan adanya aplikasi pohon keputusan untuk proses pemilihan jurusan bagi mahasiswa Institut Teknologi Bandung, mahasiswa lebih tau apa perbedaan tiap jurusannya masing-masing, dan tau dimana minatnya sendiri. Diharapkan banyak mahasiswa yang terbantu dan bisa memutuskan dengan tepat demi kelanjutan masa depannya di jurusan terbaik yang bisa didapat. Walaupun tidak semua faktor yang dimasukkan dalam pohon keputusan, namun diharapkan mahasiswa lebih tau apa keunggulan masing-masing jurusannya.

Masih banyak penerapan dari pohon keputusan untuk membantu kegiatan kita dalam keseharian. Seperti memilih untuk pekerjaan apa sesuai minat, bakat, atau title yang dimiliki.

Penggunaan pohon keputusan sangat disarankan bagi orang-orang yang susah untuk memilih sesuatu walaupun sudah tau mana yang lebih baik. Dengan pohon keputusan, bisa membuat kita lebih yakin dan sesuai dengan minat bakat diri masing-masing.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa, berkat rahmat-Nya penulis bisa menyelesaikan penulisan makalah matematika diskrit ini. Tidak lupa juga penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir, dan sebagai dosen mata kuliah matematika diskrit di program studi Teknik Informatika Institut Teknologi Bandung yang telah membimbing penulis baik dalam pengajaran di kelas maupun dalam membuat makalah matematika diskrit semester ini. Kemudian, penulis menyampaikan ucapan terima kasih kepada kedua orang tua penulis, keluarga, dan teman-teman yang telah memberikan motivasi kepada penulis sehingga penulis bisa semangat dalam menyelesaikan makalah matematika diskrit ini.

## REFERENSI

- [1] Munir, Rinaldi, 2010, "Matematika Diskrit, Edisi 3", Bandung : Penerbit Informatika.
- [2] <http://7valley.blogspot.com/2014/04/algorithm-prim.html>, diakses 6 desember 2019 pukul 03.47 WIB.
- [3] <https://docplayer.info/30279798-Pengaplikasian-graf-dalam-kehidupan-sehari-hari.html> diakses 6 desember 2019 pukul 01.00 WIB.
- [4] <http://danyatriokintoko.blogspot.com/2013/02/teori-dasar-graf.html>, diakses 5 desember 18.00 WIB.
- [5] [teknikelektroteknologiinformasi.blogspot.com/2011/12/algorithm-kruskal.html](http://teknikelektroteknologiinformasi.blogspot.com/2011/12/algorithm-kruskal.html), diakses 6 desember 04.00 WIB.
- [6] <https://dianagita.wordpress.com/2017/01/08/pengertian-dan-manfaat-pohon-keputusan/>, diakses 6 desember 04.29 WIB.
- [7] <https://fairuzelsaid.wordpress.com/2009/11/24/data-mining-konsep-pohon-keputusan/>, diakses 5 desember 17.34 WIB.

## 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, 6 Desember 2019



Fikra Hadi Ramadhan  
13518036