

STUDI DAN APLIKASI POHON KEPUTUSAN SEBAGAI SUPPORT TOOL DALAM PENGAMBILAN KEPUTUSAN

Yuandra Ismiraldi

13505069

*Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2007*

Email : if15069@students.if.itb.ac.id

Abstraksi

Makalah ini membahas tentang studi dan implementasi sebuah skema pohon keputusan yang digunakan sebagai sebuah alat bantu / pendukung (support tool) dalam membuat suatu keputusan di dalam sebuah sistem pengambilan keputusan. Studi dari pohon keputusan yang dibahas dalam makalah ini adalah dari segi skema pohon keputusan, dari segi struktur pohon keputusan, serta dari aplikasi pohon keputusan dengan memperhatikan analisa resiko, nilai utilitas, serta nilai informasi yang dapat mempengaruhi alternatif keputusan yang dapat diambil dengan pohon keputusan tersebut.

Secara struktural, 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.

Pohon keputusan juga dapat digunakan untuk memperhitungkan analisa resiko dan tingkat utilitas yang ada pada suatu alternatif pengambilan keputusan. Selain itu, pohon keputusan juga dapat memperhitungkan nilai dari informasi tambahan yang mungkin akan kita pergunakan dalam mengambil salah satu dari alternatif keputusan yang ada di dalam pohon keputusan tersebut.

Kemampuan untuk menganalisa alternatif pengambilan keputusan yang ada dengan memperhitungkan faktor-faktor yang ada termasuk analisa resiko dan analisa informasi yang ada serta mampu memberikan estimasi hasil akhir dari suatu alternatif pengambilan keputusan membuat pohon keputusan ini sebuah support tool yang sangat berguna dalam memecahkan masalah pengambilan keputusan.

Kata kunci : pohon keputusan, decision tree, decision tree learning, decision analysis, data mining, dataset, pohon structure, graf theory

1. Pendahuluan

Di dalam kehidupan manusia sehari-hari, manusia selalu dihadapkan oleh berbagai macam masalah dari berbagai macam bidang. Masalah-masalah ini yang dihadapi oleh manusia tingkat kesulitan dan kompleksitasnya sangat bervariasi, mulai dari yang teramat sederhana dengan sedikit faktor-faktor / hal-hal berkaitan dengan masalah tersebut dan perlu diperhitungkan sampai dengan yang sangat rumit dengan banyak sekali faktor-faktor / hal-hal yang turut serta berkaitan dengan masalah tersebut dan perlu untuk diperhitungkan.

Untuk menghadapi masalah-masalah ini, manusia mulai mengembangkan sebuah sistem / cara yang dapat membantu manusia agar dapat dengan mudah mampu untuk menyelesaikan masalah-masalah tersebut. Adapun pohon keputusan ini adalah sebuah jawaban akan sebuah sistem / cara yang manusia kembangkan untuk membantu mencari dan membuat keputusan untuk masalah-masalah tersebut dan dengan memperhitungkan berbagai macam faktor yang ada di dalam lingkup masalah tersebut. Dengan pohon keputusan, manusia dapat dengan mudah melihat mengidentifikasi dan melihat hubungan antara faktor-faktor yang mempengaruhi suatu masalah dan dapat mencari penyelesaian terbaik dengan memperhitungkan faktor-faktor tersebut. Pohon keputusan ini juga dapat menganalisa nilai resiko dan nilai suatu informasi yang terdapat dalam suatu alternatif pemecahan masalah.

Peranan pohon keputusan ini sebagai alat bantu dalam mengambil keputusan (*decision support tool*) telah dikembangkan oleh manusia sejak perkembangan teori pohon yang dilandaskan pada teori graf. Kegunaan pohon keputusan yang sangat banyak ini membuatnya telah dimanfaatkan oleh manusia dalam berbagai macam sistem pengambilan keputusan.

2. Skema Pohon keputusan

Skema dan struktur pohon keputusan adalah suatu permodelan dari struktur pohon menurut teori graf.

2.1. Struktur Dasar Pohon

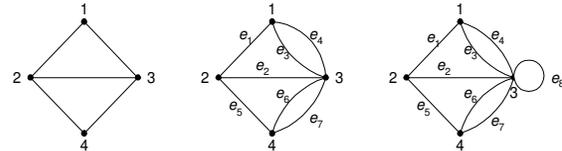
Berdasarkan teori graf, definisi pohon adalah “sebuah graf, tak-berarah, terhubung, yang tidak mengandung sirkuit”

Graf adalah suatu representasi visual dari objek-objek diskrit yang dinyatakan dengan noktah, bulatan, atau titik, serta hubungan yang ada antara objek-objek tersebut.

Secara matematis, graf didefinisikan sebagai pasangan himpunan (V,E) yang dalam hal ini :

V = himpunan tidak-kosong dari simpul-simpul (*vertices*) = $\{ v_1, v_2, \dots, v_n \}$

E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul = $\{ e_1, e_2, \dots, e_n \}$

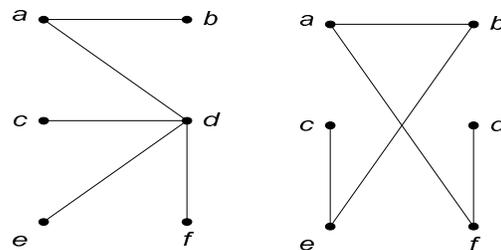


Gambar 1. Beberapa Contoh Graf

Tak-Berarah di sini adalah graf tersebut sisinya tidak mempunyai arah.

Terhubung maksudnya adalah semua simpulnya dapat terhubung oleh sisi-sisi yang ada pada graf tersebut.

Tidak mengandung sirkuit maksudnya adalah graf tersebut tidak mengandung suatu lintasan yang melalui masing-masing sisi/simpul dalam graf tepat satu kali yang kembali ke simpul asal.



Gambar 2. Contoh Pohon

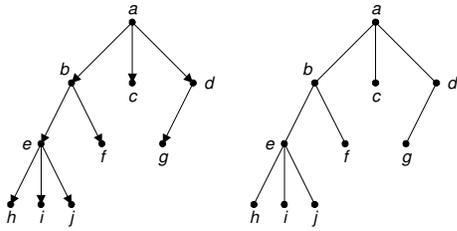
Selain itu, sebuah pohon juga memenuhi salah satu dari pernyataan-pernyataan yang ekuivalen di bawah ini :

Misal G adalah sebuah pohon.

1. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
2. G terhubung dan memiliki $m = n - 1$ buah sisi.
3. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
5. G terhubung dan semua sisinya adalah jembatan.

Adapun permodelan pohon yang biasa dipakai dalam pohon keputusan adalah rooted tree (pohon berakar)

Pohon berakar adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah.



Gambar 3. Pohon Berakar

Sebagai konvensi, tanda arah panah dapat diabaikan

Beberapa terminologi dalam pohon berakar :

Anak/Child atau Orangtua/Parent : b,c, dan d adalah anak dari a dan a adalah orangtua dari b,c, dan d.

Lintasan/Path : lintasan dari a ke j adalah a,b,e,j. Panjang lintasan dari a ke j adalah jumlah sisi yang dilalui, yaitu 3.

Saudara kandung/Sibling : b,c,dan d adalah saudara kandung sebab mempunyai orangtua yang sama yaitu a.

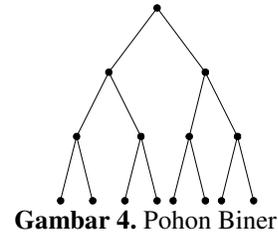
Derajat : derajat adalah jumlah anak yang ada pada simpul tersebut. A berderajat 3, dan b berderajat 2. Derajat suatu pohon adalah derajat maksimum dari semua simpul yang ada. Pohon pada gambar 3 berderajat 3.

Daun : daun adalah simpul yang tidak mempunyai anak. c,f,g,h,i,dan j adalah daun

Simpul dalam/Internal nodes : simpul yang mempunyai anak. Simpul a,b, dan d adalah simpul dalam.

Tingkat/Level : adalah 1 + panjang lintasan dari simpul teratas ke simpul tersebut. Simpul teratas mempunyai tingkat = 1.

Pohon n-ary : pohon yang tiap simpul cabangnya mempunyai banyaknya n buah anak disebut pohon n-ary. Jika n=2, pohonnya disebut pohon biner.

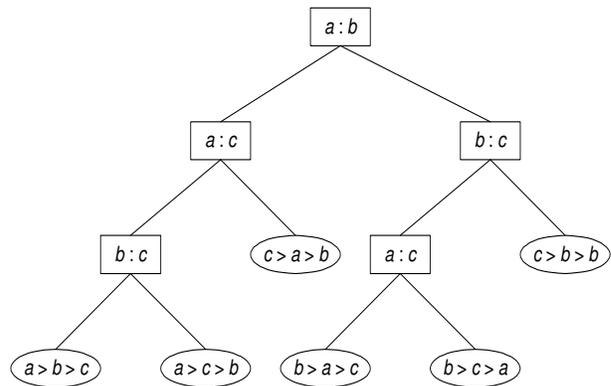


Gambar 4. Pohon Biner

2.2. Struktur Pohon Keputusan

Secara umum, pohon keputusan adalah suatu gambaran permodelan dari suatu persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi. Tiap simpul dalam menyatakan keputusan dan daun menyatakan solusi.

Permodelan pohon keputusan di sini berupa permodelan pohon n-ary, dengan jumlah anak yang dapat berbeda-beda tiap simpulnya.



Gambar 5. Pohon keputusan untuk mengurutkan 3 buah bilangan a,b, dan c.

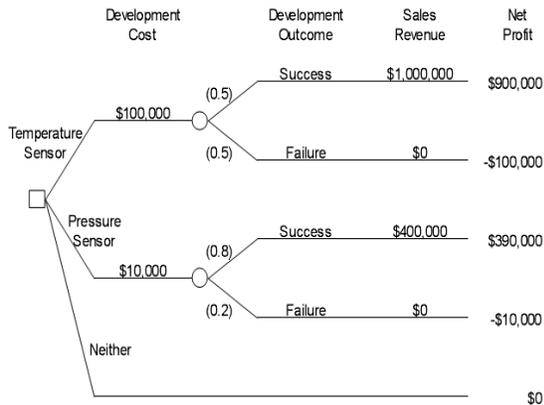
Pohon keputusan pada gambar 5 dibaca dari atas ke bawah. Simpul paling atas pada pohon keputusan ini adalah *simpul akar*. Simpul yang ditandai dengan tanda kotak di simpul tersebut disebut dengan *simpul keputusan*. Cabang-cabang yang mengarah ke kanan dan ke kiri dari sebuah cabang keputusan merepresentasikan kumpulan dari alternatif keputusan yang bisa diambil. Hanya satu keputusan yang dapat diambil dalam suatu waktu.

Dalam beberapa pohon keputusan, juga sering disertakan simpul tambahan, yaitu *simpul probabilitas*. Simpul ini biasa ditandai dengan

gambar lingkaran kecil yang disertai dengan angka pada cabang-cabang yang mengakar pada simpul probabilitas itu. Angka-angka yang terletak pada cabang-cabang tersebut merupakan probabilitas kesempatan akan munculnya keputusan yang ada di cabang tersebut dalam pilihan.

Sebagai sebuah catatan, pohon keputusan tidak hanya dapat ditulis secara vertikal, namun juga dapat secara horizontal. Pada penulisan secara horizontal, pembacaan pohon keputusan dimulai dari kiri ke kanan.

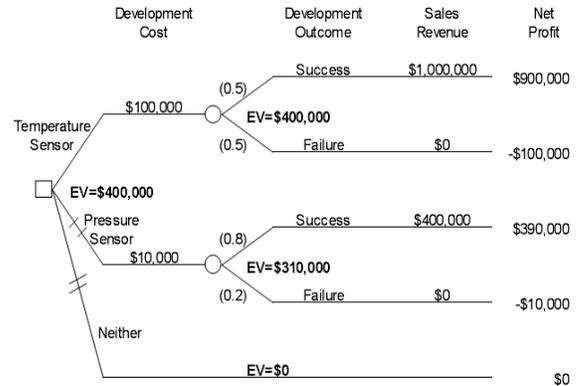
Selain itu, di posisi paling bawah sebuah pohon keputusan juga dapat ditambahkan sebuah *titik akhir* (*endpoint*), yang merepresentasikan hasil akhir dari sebuah lintasan dari akar pohon keputusan pohon tersebut sampai ke titik akhir itu.



Gambar 6. Pohon keputusan yang ditulis secara horizontal dan mengandung simpul probabilitas.

Expected value / hasil estimasi adalah sebuah estimasi hasil dari sebuah keputusan tertentu. Hasil ini didapatkan dari mengkalikan setiap kemungkinan peluang terjadinya suatu kemungkinan lalu menambahkan hasilnya menjadi suatu jumlah.

Expected value decision criterion / kriteria keputusan hasil estimasi adalah suatu seleksi agar dapat memilih sebuah alternatif keputusan yang mempunyai hasil estimasi yang paling baik / yang paling diinginkan. Dalam situasi bila “more is better” atau lebih banyak itu lebih baik, maka pilihan keputusan dengan hasil estimasi paling tinggi adalah yang terbaik, sedangkan dalam situasi bila “less is better”, maka pilihan keputusan dengan hasil estimasi paling rendah adalah yang terbaik.



Gambar 7. Pohon keputusan dengan *expected value / hasil estimasi*

Di dalam pohon keputusan pada gambar 7, cabang pohon keputusan yang mengandung alternatif pilihan yang kurang disarankan/kurang baik menurut kriteria hasil estimasi ditandai dengan tanda ‘//’ pada cabang tersebut. Hasil estimasi pada setiap simpul probabilitas ditandai dengan tanda ‘EV’. Hasil estimasi yang terdapat pada simpul keputusan bernilai sama dengan hasil estimasi bila kita mengikuti cabang tersebut sampai mencapai keputusan akhir.

Decision tree rollback adalah suatu teknik untuk menghitung selanjutnya hasil estimasi yang ada dari simpul keputusan di akhir pohon sampai kembali ke akar pohon keputusan tersebut.

Decision strategy/ strategi pengambilan keputusan adalah semua spesifikasi lengkap dari semua kemungkinan pilihan yang sesuai dengan kriteria hasil dari sebuah pengambilan keputusan suatu masalah secara sekuensial dengan menggunakan sebuah pohon keputusan.

Dalam *decision analysis*, pohon keputusan dapat diartikan sebagai sebuah alat untuk membuat ide yang secara umum dapat mengacu kepada sebuah graf atau sebuah model dari keputusan-keputusan dan akibat-akibat yang dapat muncul dari keputusan-keputusan tersebut, termasuk peluang terjadinya suatu kejadian, biaya yang dibutuhkan, dan utilitas. Melalui pohon ini strategi terbaik untuk menyelesaikan suatu masalah dapat diketahui. Pohon keputusan juga digunakan untuk mengkalkulasikan peluang kondisi-kondisi yang mungkin akan terjadi disertai dengan analisis- analisis faktor-faktor yang mempengaruhi keputusan yang diambil dengan menggunakan pohon keputusan tersebut.

3. Aplikasi Pohon Keputusan

3.1. Analisis Resiko

Hasil estimasi sebagai sebuah kriteria untuk membuat sebuah keputusan menimbulkan kesan bahwa resiko yang harus ditanggung dengan mengambil keputusan tersebut cukup kecil sehingga dapat diterima. Untuk kumpulan keputusan secara umum yang sesuai dengan hasil ekspektasi tersebut asumsi seperti ini dapat diterima, namun untuk perhitungan yang lebih terinci asumsi seperti ini tidak dapat dipakai. Oleh karena itu, harus ditambahkan pula analisis resiko yang memadai dalam pengambilan keputusan.

Hasil dari sebuah keputusan dengan resiko yang cukup tinggi mungkin dapat berbeda dengan hasil estimasi yang dihitung dalam alternatif pilihan itu karena adanya potensi resiko kerugian/kehilangan nilai yang cukup besar.

Certainty equivalent / hasil pasti adalah sebuah hasil yang kurang lebih sama dengan hasil estimasi yang ada pada suatu keputusan, namun telah memperhitungkan resiko-resiko yang ada.

Certainty equivalent ini digunakan untuk memperhitungkan tingkat resiko yang ada berdasarkan perbandingannya dengan hasil estimasi pada keputusan tersebut. Hal ini akan menentukan *risk attitude/sikap resiko* yang akan kita pakai saat kita mengambil suatu keputusan.

Risk averse adalah bila *certainty equivalent* lebih kecil daripada hasil estimasi, maka kita disebut *risk averse* bila kita mengambil keputusan ini.

Risk neutral adalah bila *certainty equivalent* sama besar / equal dengan hasil estimasi, maka kita disebut *risk neutral* bila kita mengambil keputusan ini.

Risk seeking adalah bila *certainty equivalent* lebih besar daripada hasil estimasi, maka kita disebut *risk seeking* bila kita mengambil keputusan ini.

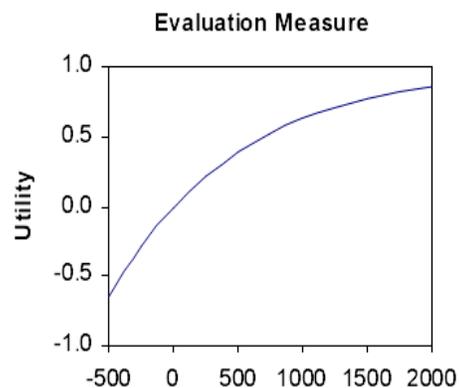
Risk attitude/sikap resiko yang kita pakai dalam mengambil keputusan akan sangat berpengaruh kepada penilaian kita akan suatu keputusan dan akan menentukan apakah kita akan mengambil keputusan tersebut atau tidak sesuai dengan sikap resiko yang kita pergunakan.

3.1.1. Fungsi Utilitas

Karena *certainty equivalent* dapat ditentukan untuk berbagai macam alternatif keputusan yang ada dalam sebuah masalah pengambilan keputusan, maka lebih

baiklah bila alternatif-alternatif pilihan keputusan yang ada langsung diseleksi berdasarkan *certainty equivalent* yang terbaik.

Utility function/fungsi utilitas adalah sebuah prosedur/metode yang mentranslasikan hasil akhir suatu keputusan menjadi angka-angka sehingga hasil estimasi dari angka utilitas yang dihasilkan tersebut dapat digunakan untuk mengkalkulasikan *certainty equivalent* dari alternatif-alternatif keputusan yang ada dan tetap konsisten/sejalan dengan sikap resiko sang pengambil keputusan.



Gambar 8. Ilustrasi fungsi utilitas

Di dalam ilustrasi fungsi utilitas pada gambar 8, sumbu horizontal merepresentasikan tingkatan skala ukuran evaluasi, dan sumbu vertikal merepresentasikan utilitas dari setiap tingkatan skala ukuran evaluasi. Angka-angka utilitas yang terletak pada sumbu vertikal menunjukkan tingkatan level evaluasi yang paling disarankan, semakin besar angkanya, semakin baik pula level evaluasi itu.

Secara intuitif, dapat dilihat pada gambar 8, hasil fungsi utilitas turun secara drastis ketika level perhitungan evaluasi menjadi lebih negatif / memburuk, dan penurunan nilai tersebut menjadi tidak begitu drastis sejalan dengan level perhitungan evaluasi yang menjadi lebih positif / membaik. Hal ini menunjukkan bahwa nilai yang hilang dari setiap penurunan perhitungan evaluasi menjadi lebih besar sejalan dengan level perhitungan evaluasi menjadi lebih negatif. Sehingga, jika kita mengambil hasil estimasi utilitas dari perhitungan evaluasi tersebut, alternatif-alternatif yang punya kemungkinan probabilitas cukup besar untuk menghasilkan hal-hal yang tidak menguntungkan akan dipenalti lebih besar dalam perhitungan kalkulasi dibandingkan dengan bila menggunakan hasil estimasi biasa untuk melakukan perhitungan evaluasi. Sehingga, sebuah alternatif dengan probabilitas yang cukup tinggi

untuk menghasilkan hasil yang tidak menguntungkan akan diturunkan nilainya dengan menggunakan fungsi utilitas dari apa yang akan menjadi benar jika hasil estimasi biasa digunakan untuk melakukan perhitungan alternatif-alternatif keputusan yang ada.

Ide utama pendekatan dengan mengkalkulasikan certainty equivalent adalah untuk pertama-pertama mengkonversi kemungkinan-kemungkinan hasil yang ada dalam sebuah masalah pengambilan keputusan yang ada ke dalam nilai utilitas dengan menggunakan fungsi utilitas, lalu mengkalkulasi hasil estimasi dari nilai-nilai utilitas yang ada dari setiap alternatif menggunakan prosedur yang sama yang dipakai untuk menghitung nilai estimasi. Setelah hasil estimasi utilitas dihitung untuk setiap kemungkinan pilihan keputusan yang ada, maka setelah itu harus ditentukan certainty equivalent dari setiap kemungkinan pilihan itu.

Bentuk fungsi utilitas yang dipakai adalah sebuah fungsi utilitas eksponensial. Untuk masalah pengambilan keputusan yang menitik beratkan pada keuntungan / makin sedikit resiko semakin baik, dengan menggunakan banyak perhitungan evaluasi, maka fungsi eksponensialnya adalah :

$$u(x) = 1 - e^{-x/R}, \quad R > 0$$

Dengan $u(x)$ merepresentasikan fungsi utilitas, x adalah level perhitungan utilitas, R adalah sebuah konstanta yang disebut toleransi resiko, dan e merepresentasikan fungsi eksponensial (fungsi e).

Dalam sebuah situasi pengambilan keputusan dimana perhitungan evaluasi yang lebih sedikit lebih diinginkan, maka fungsi utilitas eksponensial akan mempunyai bentuk :

$$u(x) = 1 - e^{x/R}, \quad R > 0$$

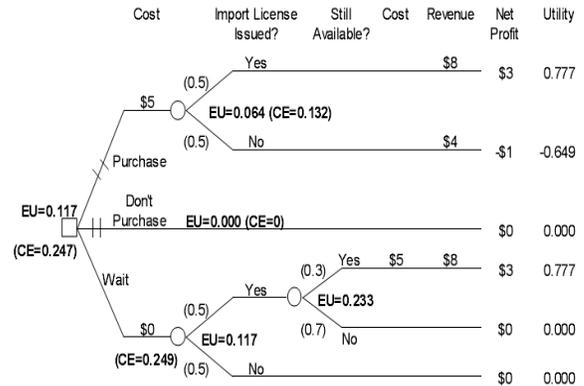
Dan dalam fungsi ini nilai yang lebih besar dari x mempunyai nilai utilitas yang lebih rendah.

R dalam fungsi ini juga menentukan tingkat toleransi resiko si pengambil keputusan.

3.1.2. Toleransi Resiko

Nilai toleransi resiko / R dapat dihitung dengan cara berikut. Pertama-tama tentukan sebuah alternatif fiktif yang punya peluang yang sama untuk menghasilkan hasil positif R_0 atau hasil negatif $R_0/2$. Lalu tentukan nilai R_0 sehingga kita tidak akan

bermasalah bila kita mengambil alternatif tersebut maupun tidak mengambil alternatif tersebut atau tentukan nilai R_0 sehingga nilai certainty equivalent pada alternatif keputusan fiktif ini bernilai 0. Setelah nilai R_0 dapat ditentukan, maka itulah nilai R yang kita pakai.



Gambar 9. Pohon keputusan dengan menggunakan nilai utilitas.

Di dalam pohon keputusan dalam gambar 9, dimisalkan kita mengambil R_0 dengan 2 sehingga fungsi utilitasnya menjadi.

$$u(x) = 1 - e^{-x/2},$$

Kemudian dapat dihitung nilai utilitas dari setiap nilai akhir yang ada dengan menggunakan fungsi utilitas eksponensial dengan mengambil $R=2$. Sebagai contoh, nilai fungsi utilitas dari nilai akhir yang terletak paling atas dapat dihitung dengan fungsi

$$u(x) = 1 - e^{-3/2} = 0.777.$$

Nilai estimasi utilitas juga dapat dihitung sama dengan hasil estimasi. Sebagai contoh hasil estimasi dari simpul probabilitas yang paling atas dapat dihitung dengan fungsi :

$$EU = 0.5 \times (0.777) + 0.5 \times (-0.649) = 0.064.$$

Untuk menghitung certainty equivalent dari fungsi utilitas eksponensial dapat digunakan fungsi :

$$CE = -R \times \ln(1 - EU),$$

Dimana CE adalah certainty equivalent, EU adalah estimasi utilitas, R adalah toleransi resiko, dan \ln adalah logaritma natural. Hasil certainty equivalent

yang lebih besar adalah hasil yang lebih baik bila menggunakan fungsi ini.

Bila perhitungan evaluasi yang lebih kecil diinginkan, maka fungsi untuk certainty equivalent tersebut akan menjadi :

$$CE = R \times \ln(1 - EU)$$

Dengan keterangan yang sama sesuai dengan fungsi diatas. Dalam fungsi ini, hasil fungsi certainty equivalent yang lebih kecil adalah hasil yang diinginkan.

Estimasi utilitasi ini dapat digunakan secara langsung untuk menentukan urutan/prioritas dari alternatif-alternatif pengambilan keputusan yang ada dalam sebuah masalah pengambilan keputusan. Alternatif dengan estimasi utilitas yang paling besar akan juga mempunyai certainty equivalent yang paling baik.

| Alternative | Expected Value | Certainty Equivalent | Difference |
|----------------|----------------|----------------------|------------|
| Purchase | 1.000 | 0.132 | 0.868 |
| Don't Purchase | 0.000 | 0.000 | 0.000 |
| Wait | 0.450 | 0.249 | 0.201 |

Gambar 10. Tabel perbandingan nilai estimasi dan certainty equivalent

Tabel pada gambar 10 menunjukkan perbandingan nilai estimasi dan certainty equivalent dengan data-data yang diambil dari pohon keputusan pada gambar 9. Dapat dilihat bahwa 3 alternatif yang ada mempunyai tingkat resiko yang berbeda-beda. Pada alternatif “don't purchase”, tidak ada perbedaan dari nilai estimasi dan certainty equivalent karena tidak ada ketidakpastian / tidak ada resiko dengan alternatif keputusan ini. Sedangkan dalam alternatif “purchase”, perbedaan antara nilai estimasi dan certainty equivalent adalah paling besar di antara ketiganya. Ini menunjukkan bahwa alternatif “purchase” tersebut mempunyai resiko yang paling besar. Hal ini bisa membuat nilai alternatif “purchase” tersebut menjadi berkurang dan dapat membuatnya tidak menjadi dipilih sebagai alternatif keputusan yang paling baik/menguntungkan. Alternatif yang lain, “wait”, juga mempunyai nilai certainty equivalent yang lebih rendah daripada nilai estimasinya. Hal ini menunjukkan bahwa alternatif tersebut juga mempunyai resiko, sama seperti alternatif “purchase”, namun dalam hal ini karena nilai perbedaannya lebih kecil daripada alternatif “purchase” maka dapat dikatakan bahwa alternatif

“wait” mempunyai peluang resiko yang lebih kecil daripada dengan alternatif “purchase”. Bila sikap resiko sang pengambil keputusan tidak ingin mengambil resiko, maka alternatif “wait” ini akan menjadi alternatif keputusan pilihan yang paling baik/menguntungkan bagi sang pembuat keputusan tersebut.

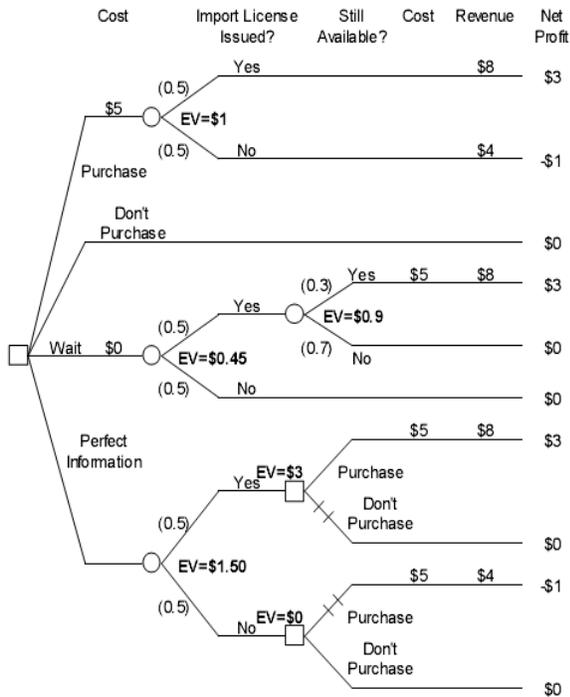
3.2. Analisis Informasi

Saat kita dihadapkan dalam sebuah masalah pengambilan keputusan, saat akan mengambil keputusan tersebut kita dapat diberikan sebuah opsi untuk mengumpulkan informasi/data-data lebih banyak untuk membantu kita mengambil keputusan dalam masalah pengambilan keputusan tersebut, Yang menjadi masalah adalah nilai dari informasi yang bisa kita kumpulkan tersebut dan perlu/tidaknya kita mengumpulkan informasi lebih banyak berdasarkan nilai informasi tersebut.

Perfect information / informasi sempurna adalah informasi yang dapat menghilangkan segala keraguan dan ketidakpastian yang ada mengenai hasil akhir dari kemungkinan alternatif keputusan-keputusan yang ada. Dengan kata lain, jika kita mempunyai informasi ini, maka kita tidak perlu ragu lagi akan hasil akhir yang akan kita dapatkan bila kita memilih suatu alternatif keputusan dalam sebuah masalah pengambilan keputusan. Nilai dari informasi sempurna ini adalah yang paling tinggi diantara informasi-informasi yang lain.

Imperfect information/informasi tidak sempurna adalah informasi yang bernilai lebih rendah daripada informasi sempurna.

Bila semua opsi-opsi pilihan yang ada untuk mengumpulkan informasi lebih banyak mengharuskan kita untuk mengeluarkan sebuah nilai yang lebih besar daripada nilai informasi sempurna, maka kita tidak perlu menganalisa pilihan-pilihan yang ada tersebut, karena tidak ada nilai yang lebih besar daripada informasi sempurna.



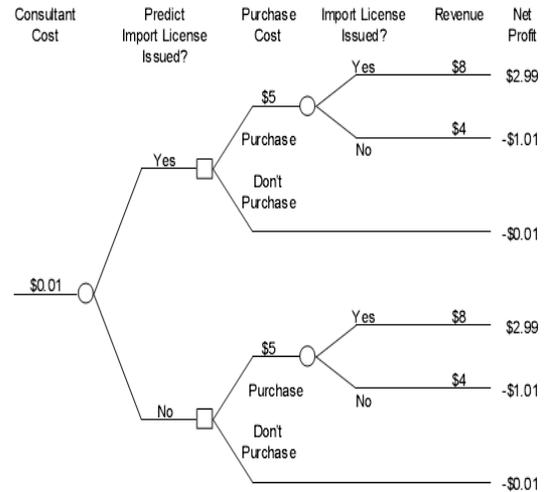
Gambar 11. Pohon keputusan dengan informasi sempurna

Dalam pohon keputusan yang terdapat di dalam gambar 11, ditunjukkan sebuah pohon keputusan yang di dalamnya terdapat cabang informasi sempurna. Di dalam pohon ini 3 cabang teratas menunjukkan cabang keputusan biasa sedangkan cabang terbawah menunjukkan cabang keputusan informasi sempurna. Dengan adanya pilihan yang mengandung informasi sempurna, dapat diketahui dengan segera informasi yang krusial dan menentukan pilihan apa yang diambil. Dengan informasi sempurna ini, tingkat resiko dalam mengambil suatu pilihan keputusan akan dapat dikurangi, atau malah dapat dihilangkan sama sekali.

Dengan kembali melihat gambar 11, dapat ditentukan nilai dari informasi sempurna ini melalui hasil estimasi cabang yang mempunyai informasi sempurna tersebut. Dari gambar 11, dapat dilihat hasil estimasinya adalah \$1.5. Karena alternatif terbaik tanpa informasi sempurna yaitu alternatif “purchase” mempunyai hasil estimasi sebesar \$1.0 sehingga nilai di dari informasi tersebut adalah $1.5 - 1.0 = 0.5$. Nilai \$0.5 yang diperoleh ini adalah batas maksimal berapa nilai biaya yang harus kita keluarkan untuk mendapatkan suatu informasi, apapun informasinya. Kita tidak perlu membayar biaya lebih mahal daripada itu sebab nilai \$0.5 itu adalah nilai dari informasi sempurna dan tidak ada

nilai yang lebih tinggi daripada nilai informasi sempurna.

Informasi tidak sempurna juga dapat dimasukkan ke dalam pohon keputusan sebagai asas pertimbangan.

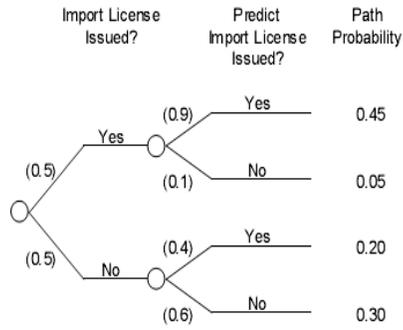


Gambar 12. Pohon keputusan dengan informasi tidak sempurna

Pada gambar 12, ditunjukkan sebuah pohon keputusan yang mengandung informasi tidak sempurna, yang ditunjukkan dengan label “Consultant Cost”. Mendapatkan informasi lebih dari konsultan adalah suatu informasi tidak sempurna karena masih mengandung resiko dan adanya ketidakpastian dari keakuratan informasi konsultan tersebut. Nilai dari informasi tidak sempurna ini adalah \$0.01. Hal ini sesuai dengan ketentuan bahwa nilai informasi tidak sempurna tidak boleh lebih besar daripada informasi sempurna.

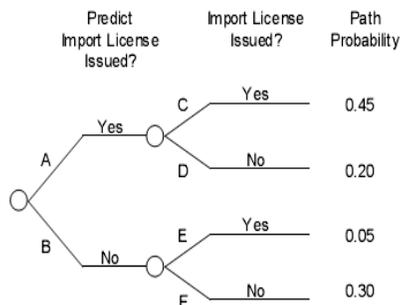
Akan tetapi, untuk lebih akurat dalam memasukkan perhitungan mengenai informasi tidak sempurna tersebut dalam sebuah pohon keputusan, kita harus juga memperhitungkan probabilitas kemungkinan dari cabang-cabang keputusan dalam pohon keputusan tersebut yang mengandung/berdasarkan informasi tidak sempurna itu.

Oleh karena itu, maka perlu untuk dibuat pohon yang menggambarkan probabilitas kemungkinan informasi tidak sempurna itu benar, dan probabilitas kemungkinan informasi tidak sempurna itu salah.



Gambar 13. Contoh pohon keputusan mengenai probabilitas kemungkinan yang ada pada informasi sempurna

Untuk menganalisis suatu informasi tidak sempurna maka digunakanlah teknik *tree flipping* / *pembalikan pohon*, yaitu suatu proses untuk mengkalkulasi probabilitas untuk sebuah pohon kemungkinan dengan urutan simpul probabilitas yang dibalik.

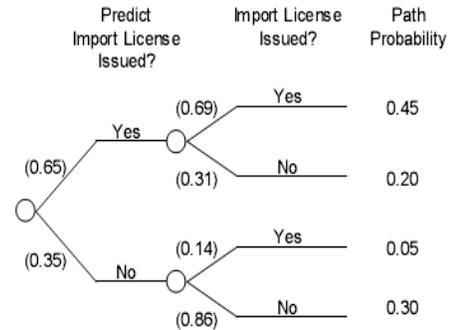


Gambar 14. Pohon dari gambar 13 yang telah dibalik dengan teknik *tree flipping*.

Di dalam gambar 14, di ujung kanan dari pohon tersebut terdapat path probability / probabilitas lintasan, yang dapat diperoleh dari perkalian peluang biasa, yaitu dengan mengkalikan probabilitas yang ada dari simpul akar sampai ke simpul ujung pada lintasan tersebut.

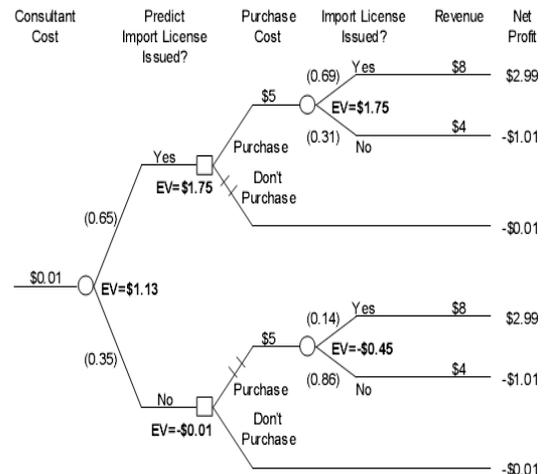
Path probability / *probabilitas lintasan* adalah probabilitas dari suatu lintasan tertentu dari simpul akar sampai ke simpul paling ujung / simpul akhir dalam pohon probabilitas.

Setelah semua probabilitas lintasan diketahui, dapat ditentukan probabilitas cabang-cabang sebelumnya, sehingga pohon pada gambar 14 dapat ditulis lagi menjadi :



Gambar 15. Pohon dengan probabilitas lintasan yang telah lengkap

Setelah mengetahui probabilitas informasi tidak sempurna tersebut, maka informasi mengenai informasi tidak sempurna itu dapat digabungkan dalam pohon keputusan pada gambar 12, hingga terbentuklah pohon keputusan seperti :



Gambar 16. Pohon keputusan yang telah dilengkapi dengan nilai informasi tidak sempurna

Dengan melihat pohon dalam gambar 16, makadapat dilihat hasil estimasi dari alternatif menggunakan informasi tidak sempurna tersebut yaitu \$1.13. Bila dibandingkan dengan hasil estimasi terbaik dengan tidak menggunakan informasi tidak sempurna tersebut adalah \$1.0, maka akan didapatkan selisih sebesar \$0.13. Hal ini menunjukkan bahwa lebih baik menggunakan keputusan yang menggunakan informasi tidak sempurna tersebut, karena menghasilkan hasil yang lebih tinggi. Nilai dari informasi tidak sempurna tersebut juga dapat bernilai lebih tinggi dari \$.01, yaitu dapat sebesar \$0.13+\$0.01 = \$0.14.

4. Kesimpulan

Dapat dilihat bahwa menggunakan pohon keputusan sebagai support tool dalam menganalisis suatu masalah pengambilan keputusan dapat sangat membantu kita dalam melakukan pengambilan keputusan.

Kegunaan pohon keputusan yang dapat melihat berbagai macam alternatif keputusan-keputusan yang dapat kita ambil serta mampu memperhitungkan nilai-nilai dari faktor-faktor yang mempengaruhi alternatif-alternatif keputusan tersebut adalah sangat penting dan berguna, karena membuat kita dapat mengetahui alternatif mana yang paling menguntungkan untuk kita ambil.

Pohon keputusan juga dapat dipergunakan untuk memperhitungkan dan melakukan analisa terhadap resiko-resiko yang mungkin muncul dalam suatu alternatif pemilihan keputusan. Selain itu, pohon keputusan juga dapat dipakai untuk memperhitungkan berapa nilai suatu informasi tambahan yang mungkin kita perlukan agar kita dapat lebih mampu dalam membuat suatu pilihan keputusan dari suatu alternatif-alternatif keputusan yang ada.

Dengan melihat kegunaan pohon keputusan dan kemampuannya dalam memperhitungkan berbagai alternatif pemecahan masalah termasuk faktor-faktor yang mempengaruhinya serta nilai resiko dan nilai informasi dalam alternatif keputusan itu, maka jelaslah bahwa pohon keputusan ini dapat menjadi alat bantu yang sangat berguna dalam pengambilan keputusan.

Daftar Pustaka

[1] Munir, Rinaldi. (2006) .Diktat Kuliah IF2153 Matematika Diskrit Edisi Keempat. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[2] Khodra, Masayu Leylia . (2006) . Bahan Kuliah IF5133 Artificial Intelligence – Learning. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[3] Wikipedia. (2006) . Decision Tree. http://en.wikipedia.org/wiki/Decision_tree. Tanggal akses : 29 Desember 2006 pukul 20.48 WIB.

[4] Wikipedia. (2006) . Decision Tree Learning. http://en.wikipedia.org/wiki/Decision_tree_learning. Tanggal akses : 29 Desember 2006 pukul 21.05 WIB.

[5] Wikipedia. (2006) . Decision Analysis. http://en.wikipedia.org/wiki/Decision_analysis. Tanggal akses : 29 Desember 2006 pukul 20.50 WIB.

[6] Wikipedia. (2006) .Data Mining. http://en.wikipedia.org/wiki/Data_mining. Tanggal akses : 29 Desember 2006 pukul 21.05 WIB.

[7] Wikipedia. (2006) . Influence Diagrams. http://en.wikipedia.org/wiki/Influence_diagrams. Tanggal akses : 29 Desember 2006 pukul 20.50 WIB.

[8] Wikipedia. (2006) . Tree (Graph Theory). [http://en.wikipedia.org/wiki/Tree_\(Graph_Theory\)](http://en.wikipedia.org/wiki/Tree_(Graph_Theory)). Tanggal akses : 29 Desember 2006 pukul 20.46 WIB.

[9] KirkWood, Craig W. (2002) . Decision Tree Primer. <http://www.public.asu.edu/~kirkwood/DASstuff/decisiontrees/index.html>. Tanggal akses : 2 Desember 2006 pukul 22.03 WIB.

[10] Hamilton, Howard J. (2006) . CS Course. http://www2.cs.uregina.ca/%7Ehamilton/courses/831/notes/ml/dtrees/4_dtrees2.html. Tanggal akses : 2 Desember 2006 pukul 22.15 WIB.