Aplikasi Pohon Keputusan dalam *PLL Patern Recognition Rubiks Cube*

Varian Caesar - 13514041
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514041@std.stei.itb.ac.id

***Abstrak*—Dalam dunia speedcubing atau seni menyelesaikan rubik dengan waktu secepat-cepatnya, terdapat sebuah metode lanjut yang dapat membuat seseorang menyentuh waktu keramat dibawah 20 detik setiap kali bermain. Metode itu bernama CFOP, yang terdiri dari 4 bagian yaitu *cross*, F2L, OLL dan PLL. Mayoritas *speedcuber* yang mulai beralih metode ke CFOP akan mengalami kesulitan khususnya pada saat menghafalkan algoritma PLL karena selain membutuhkan hafalan algortima yang panjang, pengenalan pola atau *pattern recognition* dari sekian banyak kasus yang muncul dalam PLL juga merepotkan. Untuk mencapai waktu yang lebih baik bahkan sampai dibawah 10 detik tentu *pattern recognition* haruslah sangat cepat. Untuk itu dapat dibuat pohon keputusan untuk membantu *pattern recognition* PLL lebih cepat.**

Kata Kunci*—Pohon Keputusan, Rubik, CFOP, PLL, *Pattern Recognition

I. PENDAHULUAN

Rubiks Cube atau kubus rubik adalah permainan *puzzle* mekanik yang berbentuk kubus dengan 6 sisi dan warna yang berbeda. Rubik ditemukan pertama kali oleh Prof. Erno Rubik, seorang arsitek dan pemahat asal Hungaria. Awalnya *puzzle* ini muncul di toko-toko mainan lokal dengan nama *magic cube*. Mulai tahun 1980, atas permintaan distributor, *magic cube* berubah nama menjadi *rubiks cube* dan dijual ke berbagai belahan dunia.

Dengan segera, *rubiks cube* laku keras. Setiap orang penasaran dan membelinya. Hingga tahun 1982 saja sudah terjual lebih dari 100 juta rubik. Erno Rubik menjadi salah satu milyuner di dunia.Satu demi satu kejuaraan rubik dunia dilaksanakan, puncaknya adalah *Rubiks Cube World Championship* di Budapest pada bulan Juni 1982. Kejuaraan ini dimenangkan oleh pelajar bernama Minh Thai dari Los Angeles dengan catatan waktu 22.95 detik.

Menjelan tahun 1990, ketertarikan terhadap rubik mulai memudar. Mereka sudah terlalu kesal karena tidak dapat menyelesaikannya, mengingat keterbatasan informasi saat itu. Sebagian lebih tertarik dengan kehadiran *video game* elektronik yang lebih modern. Akhirnya pada tahun 2000, petunjuk untuk menyelesaikan rubik sudah mulai bertebaran di internet. Sekali lagi demam rubik melanda dunia. Puncaknya adalah diadakan turnamen kelas dunia yang kedua di Kanada pada tahun 2003. Kejuaraan tahun 2003 mengubah 180 derajat cara bermain rubik setiap orang, Jessica Fridrich seorang profesor komputer di Binghamton University menemukan sebuah metode baru yang memungkinkan seseorang menyelesaikan rubik dibawah 15 detik bila mau berjuang menghafalkan banyak algoritma yang sesuai dengan masing-masing kondisi rubik. Metode itu bernama CFOP. Mulai sejak saat itulah terbentuk sebuah gelombang baru menyelesaikan rubik yang dikenal dengan *speedcubing.*

Pada makalah ini penulis akan membahas aplikasi pohon keputusan dalam menentukan pola yang didapat saat memasuki tahap bernama PLL pada metode CFOP diatas. Diharapkan penggunaan pohon keputusan ini dapat mempersingkat waktu berpikir dalam menentukan pola yang muncul sehingga waktu yang diperoleh akan semakin cepat.

II. DASAR TEORI

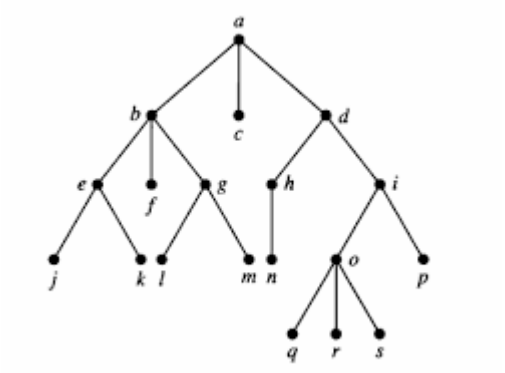
A. *Pohon*

Pohon adalah graf tak-berarah yang terhubung yang tidak mengandung sirkuit. Oleh karena itu, jumlah sisi pada pohon yaitu jumlah simpul dikurangi satu (n - 1 buah sisi).

Misalkan G = (V, E) adalah graf tak-berarah sederhana dan jumlah simpulnya n. Maka semua pernyataan dibawah ini adalah ekivalen.

1. G adalah Pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki m = n – 1 sisi.
4. G tidak mengandung sirkuit.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya ada 1 sirkuit.
6. G terhubung dan semua sisinya adalah jembatan (jembatan adalah sisi yang bila dihapus menyebabkan graf terpecah menjadi 2 bagian).

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



Gambar 2.1 Pohon berakar
(Sumber : http://mathhelpforum.com**)**

Pohon berakar memiliki beberapa terminologi penting yang perlu diketahui. Dari pohon berakar pada gambar 2.1 diatas, diketahui beberapa terminologi sebagai berikut :

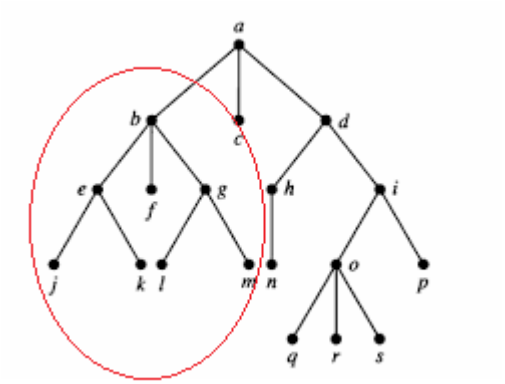
1. Anak (*child*) dan Orang tua (*parents*)
   Simpul y dikatakan anak dari simpul x apabila terdapat sisi dari simpul x ke simpul y. Pada gambar 2.1, simpul e adalah orangtua dari j dan k, sedangkan simpul m adalah anak dari g.

2. Lintasan (*path*)
   Lintasan dari simpul x ke simpul y adalah runtutan simpul-simpul x1,x2,x3,... y. Pada gambar, lintasan dari simpul a ke j adalah a,b,e,j.

3. Keturunan (*descendant*)
   Jika y adalah anak dari x maka y disebut sebagai keturunan dari x. Contoh pada gambar 2.1, simpul j adalah keturunan simpul e.

4. Saudara kandung (*sibling*)
   Simpul v1 dan v2 disebut sebagai saudara jika dan hanya jika memiliki simpul *parent* yang sama. Contoh adalah simpul j dan k pada gambar 2.1 .

5. Upapohon (*subtree)*
   Misalkan x adalah simpul didalam pohon T. Yang dimaksud dengan upapohon dengan x sebagai akarnya adalah upagraf T' = (V', E') sedemikian sehingga V' mengandung x dan semua keturunannya dan E' mengandung sisi-sisi dalam semua lintasan yang berasal dari x.



Gambar 2.2 Upapohon

Gambar 2.2 memperlihatkan contoh upapohon dari pohon pada gambar 2.1 . Lingkaran merah tersebut merupakan salah satu upapohon yang ada.

6. Derajat (*degree)*
   Derajat merupakan jumlah anak pada simpul tersebut. Derajat simpul b pada gambar adalah 3 karena memiliki 3 anak (e, f, g).

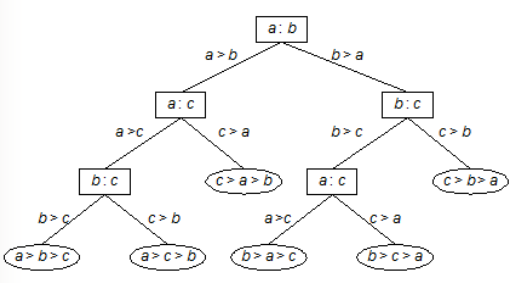
7. Daun(*leaf)*
   Merupakan simpul yang tidak memiliki anak. Contoh adalah simpul j, k, l dan m pada gambar.

8. Simpul dalam
   Yaitu simpul yang memiliki anak. Simpul b,d,e,dan g adalah contoh simpul dalam.

9. Aras(*level)*
   Akar mempunyai aras 0 sedangkan simpul lainnya mempunyai aras 1 + panjang lintasan dari akar ke simpul tersebut. Pada gambar 2.1 simpul e memiliki aras 2 dan simpul b memiliki aras 1.

10. Tinggi (*height)*
    Yaitu aras tertinggi dari suatu pohon. Pohon pada gambar 2.1 tingginya 4.

B. Pohon Keputusan
Pohon keputusan adalah aplikasi dari pohon berakar dimana setiap simpul menyatakan keputusan dan daun menyatakan solusi.

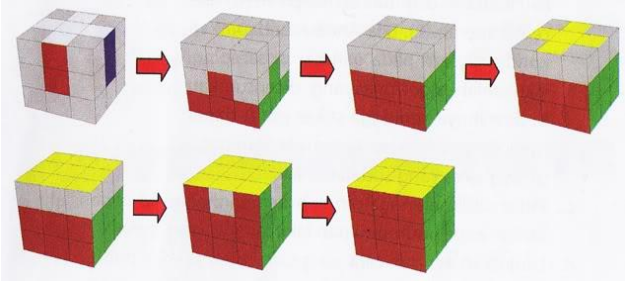


Gambar 2.3 Pohon Keputusan
(Sumber : Matematika Diskrit, Rinaldi Munir)

III. APLIKASI POHON KEPUTUSAN DALAM *PLL PATTERN RECOGNITION*

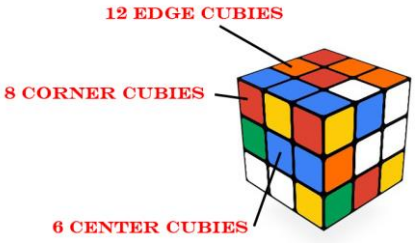
A. Metode Layer by Layer

Metode *Layer by Layer* disebut juga sebagai metode *beginner* karena banyak dipelajari oleh pemula. Inti dari metode ini ada 7 yaitu : membuat cross, menyelesaikan lapis pertama, menyelesaikan lapis kedua, membuat cross pada bagian atas, menyelesaikan sisi atas, merapikan kondisi lapis ketiga dan menyelesaikan lapis ketiga.



Gambar 3.1 Langkah pengerjaan rubik
(Sumber : Tips & Trik Jago Main Rubik, Wicaksono Adi)

Gambar diatas menunjukkan tahapan pengerjaan dengan menggunakan metode *Layer by Layer*. Tahapan pengerjaan dengan menggunakan metode dasar ini tidak akan dijelaskan lebih lanjut dalam makalah ini karena hanya berfungsi sebagai intro menuju metode *CFOP*. Jika anda tertarik mempelajarinya maka anda dapat mencari referensi terkait. Pada umumnya orang awam yang ingin belajar rubik akan memulai dengan metode ini dan kemudian untuk meningkatkan kecepatan, akan berpindah ke CFOP. Sebelum masuk ke metode *Fridrich*, perlu diketahui bahwa setiap *piece* dari rubik memiliki jenis yang berbeda, untuk mudahnya perhatikan gambar berikut:



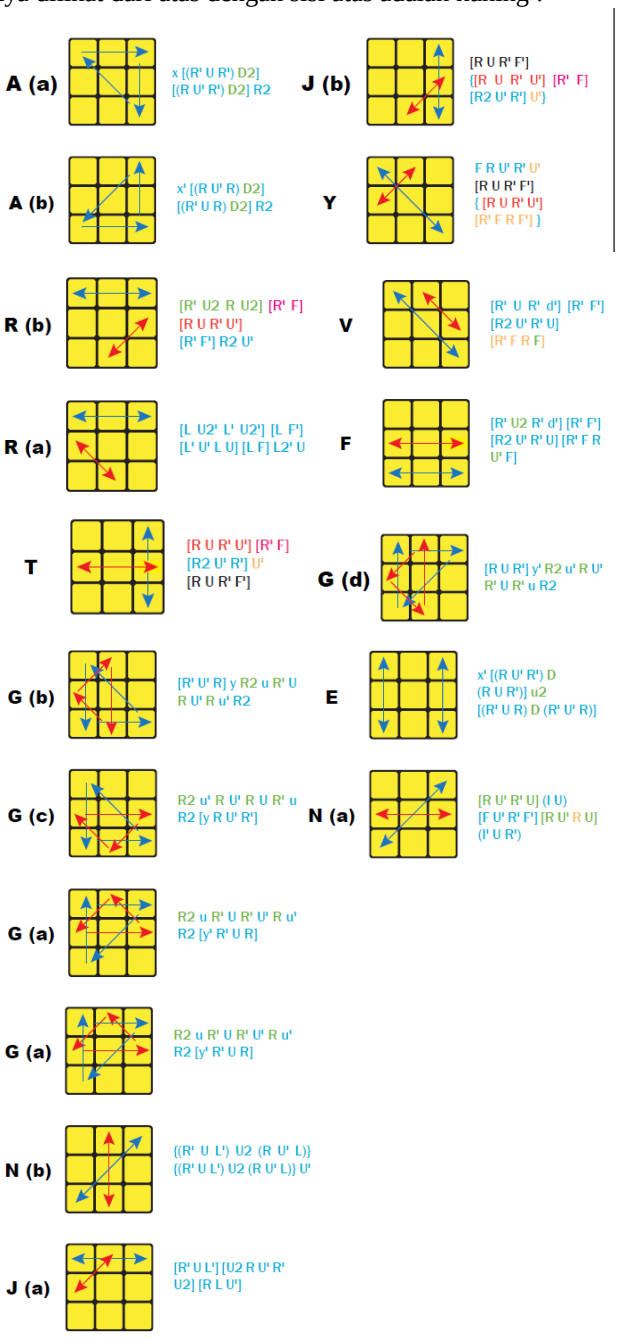
Gambar 3.2 Jenis *piece* pada rubik
(Sumber : smlab.cs.tau.il/projects/14/p1/Mindcuber.htm)

B. Metode Fridrich – CFOP (PLL)

Metode ini diberi nama sesuai dengan nama penemunya yaitu Jessica Fridrich, seorang profesor di bidang elektro dan komputer. Metode ini pertama kali dipublikasikan pada tahun 1997, namun baru terkenal pada tahun 2003 pasca *World Championship Rubiks Cube* kedua di Kanada. Inti dalam metode ini adalah mengeksekusi algoritma yang sesuai dengan pola tertentu pada rubik. Metode Fridrich juga terkenal dengan nama CFOP yang merupakan singkatan dari tahapannya yaitu :

1. *Cross*
2. F2L (*First 2 Layers)*
3. OLL (*Orientation of Last Layer)*
4. PLL (*Permutation of Last Layer)*

Tahapan pada metode Fridrich yang paling berpengaruh besar pada *solving* adalah PLL, seseorang yang telah menguasai PLL saja dan memiliki *pattern recognition* yang bagus dapat mencapai waktu kurang dari 30 detik.
PLL memotong tahap ke-6 dan ke-7 pada metode *beginner* menjadi hanya 1 gerakan saja. Oleh karena itu kita harus melakukan satu buah algoritma yang sesuai dengan kondisi yang diperoleh. Berikut ini adalah semua kemungkinan pola yang mungkin beserta algoritma PLL nya dilihat dari atas dengan sisi atas adalah kuning :

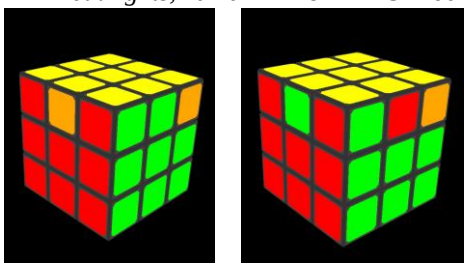


Gambar 3.3 Daftar semua PLL
(Sumber : https://Badmephisto.com)

C. Pengelompokan Algoritma PLL

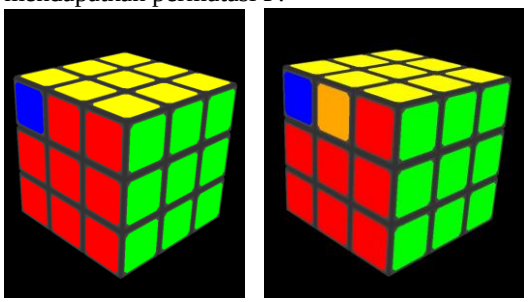
Pattern Recognition atau pengenalan pola sangatlah penting dalam PLL. Percuma saja bila hafal seluruh algoritma PLL namun pada saat eksekusinya lama karena bingung mencari pola yang didapat. Untuk itu kita dapat mengelompokan PLL berdasarkan pola umum yang didapat.

1. *Headlights*
   Headlights memiliki ciri berupa warna stiker yang sama pada sepasang corner pada sebuah sisi. Kasus yang memiliki ciri pola *Headlights* adalah permutasi A, T, R, dan G. Perhatikan juga warna stiker *edge* yang ada disamping *corner* yang memiliki warna sama itu, apabila warnanya sama dengan warna stiker *corner* maka ini disebut sebagai *connected headlights*.
   a. Permutasi T memiliki 2 buah *connected highlights*
   b. Permutasi R hanya memiliki 1 buah *connected headlights*
   c. Permutasi A dan G tidak memiliki *connected Headlights*, namun A memiliki *3 Blocks*.



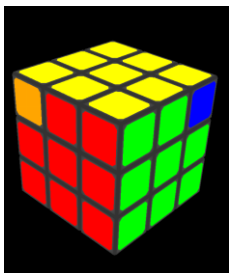
Gambar 3.4 *connected headlights dan headlights*

2. *Full Bar*
   Pada pola ini, salah satu sisi membentuk sebuah warna yang sama. Permutasi J dan F memiliki pola ini. Jika warna pada *Full bar* tersambung dengan *edge* (*connected bar*) maka kita mendapatkan permutasi J, dan apabila tidak maka kita mendapatkan permutasi F.



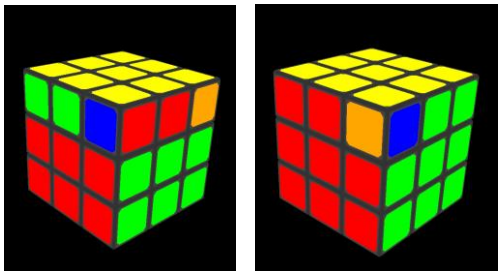
Gambar 3.5 Contoh *Full Bar* yang connected dan tidak *connected*

3. *3 Blocks*
   3 Blocks adalah ciri lain yang mudah dikenali, yaitu kondisi tersambungnya warna antara 2 *edge* dan 1 *corner* sehingga membentuk blok 1x2x2. PLL yang memiliki kondisi ini yaitu V.



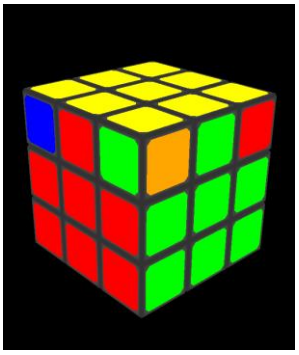
Gambar 3.6 Contoh *3 Blocks*

4. *2 pairs*
   Kondisi tersambungnya warna antara sebuah *edge* dengan sebuah *corner*. PLL yang memiliki ciri ini adalah N dan Y. Permutasi N memiliki 4 buah *2 pairs* sedangkan permutasi Y hanya punya dua buah saja dan saling mengapit sebuah *corner*.



Gambar 3.7 Contoh *2 Pairs* pada permutasi N (kiri) dan permutasi Y (kanan)

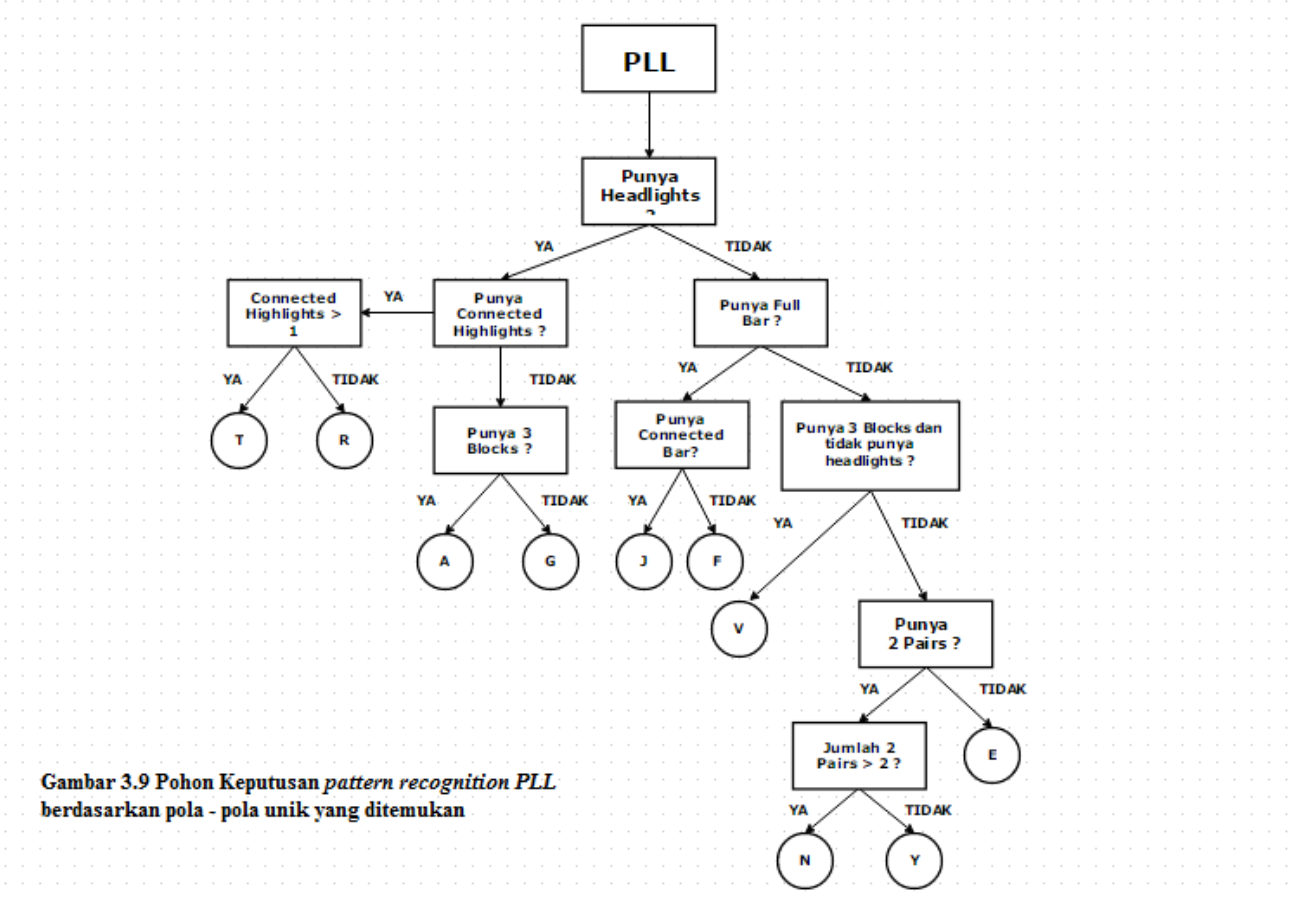
5. Permutasi E
   Satu-satunya permutasi yang belum disebutkan adalah permutasi E. Hal ini disebabkan permutasi E tidak punya ciri untuk dikenali dengan cepat. Kondisi setiap *edge* pada permutasi E sudah berada pada tempat yang benar sehingga kita tinggal menyesuaikannya saja.



Gambar 3.8 Permutasi E, semua *edges* sudah berada pada posisi benar

D. Pohon Keputusan Terhadap Pola PLL

Dari semua kemungkinan pola yang muncul pada tahap PLL yang sudah dibahas sebelumnya, kita dapat membuat sebuah *decision tree* atau pohon keputusan sebagai berikut :



Gambar 3.9 Pohon Keputusan *pattern recognition PLL* berdasarkan pola - pola unik yang ditemukan

V. KESIMPULAN

Aplikasi Pohon khusunya pohon keputusan sangat banyak. Salah satunya dapat digunakan dalam permainan rubik. *Speedcubing* atau seni menyelesaikan rubik dengan secepat-cepatnya tidak hanya membutuhkan ingatan algoritma yang baik dan presisi saja, namun juga pengenalan *pattern* yang muncul pada rubik dengan cepat agar tidak banyak waktu terbuang hanya untuk menentukan pola yang didapat. Dengan pohon keputusan, dapat dibuat skema penentuan pola dan algoritma yang harus dipakai pada tahap PLL (*Permutation of Last Layer)* pada metode tingkat lanjut yaitu *Fridrich Method CFOP*. Dengan pohon keputusan ini, pengenalan pola menjadi lebih terstruktur dan apabila diimbangin dengan latihan rutin maka kecepatan akan meningkat dengan drastis.

VII. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Tuhan Yang Maha Esa karena atas berkat dan karunia-Nya, makalah ini dapat diselesaikan tanpa ada masalah yang berarti selama proses pembuatannya. Penulis juga mengucapkan terima kasih kepada Ibu Dra. Harlili S.,M.Sc selaku dosen pembimbing dan pengajar mata kuliah IF-2120 Matematika Diskrit kelas 01, Program Studi Teknik Informatika, Institut Teknologi Bandung yang telah memberikan pengetahuan terutama materi tentang pohon yang sangat membantu dan menjadi objek dasar penyusunan makalah ini.

REFERENSI

[1] Munir, Rinaldi, *Matematika Diskrit*. Bandung: Percetakan ITB,2006, bab 9.
[2] Draw.io Online Diagram Software, https://www.draw.io diakses tanggal 5 Desember 2015, pukul 14.02 WIB
[3] Badmephisto PLL Algorithm, http://badmephisto.com/pll.php diakses tanggal 5 Desember 2015, pukul 16.30 WIB
[4] Adi, Wicaksono, *Tips & Trik Jago Main Rubik*. Yogyakarta: Gradien Mediatama,2009.
[5] Speedcubing Forum, https://www.speedsolving.com/ diakses tanggal 6 Desember 2015, pukul 1.11 WIB

[6] Khalilurrahman, Jihan, *How To Be A Sub 15 Cuber*. Yogyakarta : Gradien Mediatama,2010
[7] History of Rubiks Cube, https://www.rubiks.com/about/the-history-of-the-rubiks-cube diakses tanggal 6 Desember 2015, pukul 1.19 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, 8 Desember 2015

ttd

Varian Caesar
13514041