Pembuktian Cayley's Formula dengan Prüfer Sequence

Muntaha Ilmi (13512048)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
muntahailmi@students.itb.ac.id

***Abstrak*—Cayley's Formula adalah salah satu teorema dalam graf yang digunakan untuk menghitung banyaknya kemungkinan pohon rentang yang dapat dibentuk dari suatu graf lengkap. Terdapat banyak cara untuk membuktikan bahwa Cayley's Formula ini benar, yang akan dibahas di makalah ini adalah pembuktian dengan bantuan Prüfer sequence. Prüfer sequence sendiri adalah suatu cara merepresentasikan sebuah pohon, yang pertama kali digunakan untuk membuktikan kebenaran dari Cayley's Formula.**

***Kata Kunci*—Cayley's Formula, graf lengkap, pohon rentang, Prüfer sequence.**

I. PENDAHULUAN

Pohon rentang adalah suatu pohon yang dibentuk dari suatu graf yang mengandung semua simpul di graf tersebut. Salah satu masalah yang sering dipertanyakan adalah banyaknya cara untuk membuat pohon rentang ini. Terdapat beberapa teorema yang dapat menghitungnya, bahkan untuk bentuk graf yang lebih umum, namun yang dibahas di sini adalah salah satu teorema yang menyelesaikan masalah tersebut dalam lingkup yang lebih spesifik, yaitu khusus untuk graf lengkap.

Cayley's Formula, seperti yang sudah dikatakan sebelumnya, adalah suatu rumus untuk menghitung banyaknya cara pohon rentang dapat dibuat dari suatu graf lengkap. Rumus ini sebenarnya pertama ditemukan oleh Carl Wilhelm Borchardt pada tahun 1860, Arthur Cayley kemudian mengembangkannya di tahun 1889. Namun, pembuktian Cayley untuk rumus ini tidak meyakinkan karena tidak dapat digeneralisasi untuk graf yang lebih besar. Beberapa tahun kemudian, pembuktian untuk teorema ini dikemukakan. Salah satunya adalah dengan bantuan Prüfer sequence yang ditemukan oleh Heinz Prüfer pada tahun 1918 khusus untuk membuktikan rumus ini.

.

II. DASAR TEORI

2.1 Pohon

Pohon adalah suatu graf tak berarah yang tidak memiliki sirkuit.

Menurut teorema, misalkan G= (V, E) adalah graf tak-berarah sederhana dan jumlah simpulnya n. Maka, semua pernyataan di bawah ini adalah ekivalen :

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

Teorema di atas dapat juga disebut definisi dari sebuah pohon.

2.2 Pohon Rentang

Pohon rentang adalah pohon yang dibentuk dari sebuah graf yang mengandung semua simpul dari graf tersebut. Tentu saja, pohon rentang harus memiliki semua sifat pohon, terutama sifat tidak mengandung sirkuit dan menghubungkan setiap pasang simpul.

2.3 Graf Lengkap

Graf lengkap adalah graf sederhana yang semua simpulnya memunyai sisi ke semua simpul lainnya. Graf lengkap dengan *n* buah simpul dilambangkan dengan K_n. Jumlah sisi pada graf lengkap yang terdiri dari n buah simpul adalah $n(n-1)/2$.[1]

2.4 Cayley's Formula

Cayley's Formula ditemukan pertama oleh Carl Wilhelm Borchardt pada tahun 1860,[2] dan dikembangkan oleh Arthur Cayley pada tahun 1889.

Teorema ini mengatakan bahwa untuk semua bilangan bulat positif *n*, jumlah pohon dengan *n* buah simpul berlabel yang bisa dibentuk adalah n^{n-2}. [3]

Dengan kata lain, teorema ini menghitung banyaknya pohon rentang yang dapat dibentuk dari sebuah graf lengkap.

2.5 Prüfer Sequence

Prüfer sequence secara mudahnya adalah urutan n-2 angka yang setiap angkanya bernilai antara 1 sampai n. Prüfer sequence adalah salah satu cara untuk merepresentasikan sebuah pohon berlabel dengan n buah simpul dengan n-2 urutan angka. Prüfer sequence pertama ditemukan oleh Heinz Prüfer pada tahun 1918 dan dipakai untuk membuktikan Cayley's Formula.[4] Agak sulit untuk melihat pohon yang direpresentasikan oleh suatu Prüfer sequence dan sebaliknya. Berikut adalah algoritma untuk membentuk Prüfer sequence dari suatu pohon dan sebaliknya.

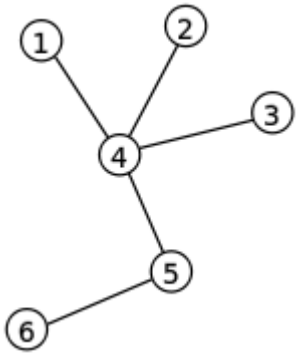
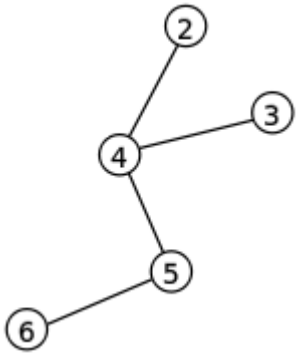
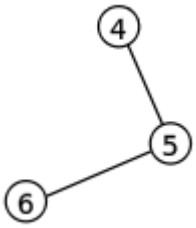
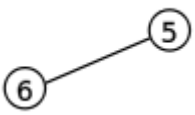
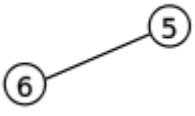
2.5.1 Membentuk Prüfer Sequence

Algoritma untuk membentuk Prüfer Sequence dari sebuah pohon P adalah sebagai berikut.

1. Ambil simpul dari pohon P dengan derajat 1. Jika ada lebih dari satu simpul, ambil yang labelnya paling kecil. Sebut saja simpul ini simpul V.
2. Masukkan simpul yang bertetangga dengan simpul V pada Prüfer Sequence. Seharusnya hanya ada satu simpul, karena V berderajat 1.
3. Hilangkan simpul V dan busur yang berhubungan dengan simpul V dari pohon P.
4. Ulangi langkah 1-3 hingga hanya tersisa tepat dua simpul.

Berikut adalah salah satu contoh pembuatan Prüfer Sequence dari sebuah pohon.

Tabel 2.1 Langkah pembentukan Prüfer Sequence

Pohon P	Prufer Sequence	Ket.
	{}	Kondisi awal. Simpul berderajat 1: {1,2,3,6}
	{4}	Simpul berderajat 1 dengan label terkecil adalah 1. Simpul 4 dimasukkan ke dalam sekuens. Simpul berderajat 1: {2,3,6}
	{4,4,4}	Simpul berikutnya adalah 2 dan 3. Simpul 4 dimasukkan 2 kali ke dalam sekuens. Simpul 4 menjadi berderajat 1. Simpul berderajat 1: {4,6}
	{4,4,4,5}	Simpul berderajat 1 berikutnya adalah simpul 4. Simpul 5 dimasukkan dalam sekuens. Simpul berderajat 1: {5,6}
	{4,4,4,5}	Kondisi akhir. Hanya tersisa 2 simpul, sehingga algoritma selesai.

2.5.2 Membentuk Pohon dari Prüfer Sequence

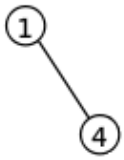
Algoritma untuk membentuk sebuah pohon P dari suatu Prüfer Sequence dengan panjang n adalah sebagai berikut.

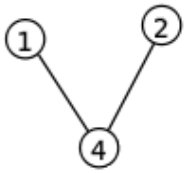
1. Buat sebuah table ‘derajat’ dengan ukuran n+2. Isi tabel ini dengan jumlah kemunculan simpul ke-i pada Prufer Sequence +1.
2. Mulai dari elemen pertama dari Prüfer Sequence, anggap saja simpul A. Ambil dari tabel ‘derajat’, suatu simpul dengan yang nilai ‘derajat’-nya 1. Jika ada lebih dari satu, ambil yang labelnya paling kecil. Ebut saja simpul tersebut simpul B.
3. Masukkan simpul A dan B ke dalam pohon P jika belum ada, lalu masukkan busur yang menghubungkan dua simpul tersebut.
4. Kurangi 1 dari nilai dari simpul A dan B di tabel derajat.
5. Ulangi langkah 2-4 untuk semua elemen dari Prufer Sequence.
6. Dipastikan akan tersisa tepat dua buah simpul dengan nilai tabel ‘derajat’ 1. Anggap saja simpul U dan V. Masukkan busur yang menghubungkan U dan V ke dalam Pohon P. [5]

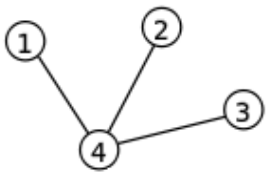
Berikut adalah salah satu contoh pembuatan pohon dari suatu Prufer Sequence.

Tabel 2.2 Langkah algoritma pembentukan pohon

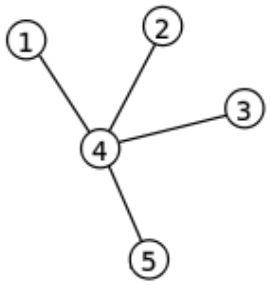
<table>
<tr><th>Pohon P</th><th>Tabel Derajat</th><th>Ket.</th></tr>
<tr><td>Kosong</td><td><table><tr><th>Simpul</th><th>Nilai</th></tr><tr><td>1</td><td>0</td></tr><tr><td>2</td><td>1</td></tr><tr><td>3</td><td>1</td></tr><tr><td>4</td><td>3</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>1</td></tr></table>Sisa sekuens:
{4,4,4,5}</td><td>Prüfer Sequence:
{4,4,4,5}

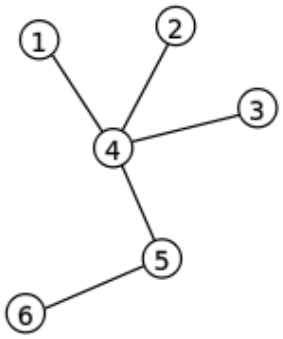
Simpul dengan nilai 1 di tabel ‘derajat’:
{1,2,3,6}</td></tr>
<tr><td>  </td><td><table><tr><th>Simpul</th><th>Nilai</th></tr><tr><td>1</td><td>0</td></tr><tr><td>2</td><td>1</td></tr><tr><td>3</td><td>1</td></tr><tr><td>4</td><td>3</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>1</td></tr></table>Sisa sekuens:
{4,4,5}</td><td>Proses elemen pertama dari sekuens, yaitu simpul 4.
Busur (1,4) dimasukkan ke pohon.

Simpul dengan nilai 1 di tabel ‘derajat’:
{2,3,6}</td></tr>
<tr><td>  </td><td><table><tr><th>Simpul</th><th>Nilai</th></tr><tr><td>1</td><td>0</td></tr><tr><td>2</td><td>0</td></tr><tr><td>3</td><td>1</td></tr><tr><td>4</td><td>2</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>1</td></tr></table>Sisa sekuens:
{4,5}</td><td>Proses elemen kedua dari sekuens, yaitu simpul 4.
Busur (2,4) dimasukkan ke pohon.

Simpul dengan nilai 1 di tabel ‘derajat’:
{3,6}</td></tr>
<tr><td>  </td><td><table><tr><th>Simpul</th><th>Nilai</th></tr><tr><td>1</td><td>0</td></tr><tr><td>2</td><td>0</td></tr><tr><td>3</td><td>0</td></tr><tr><td>4</td><td>1</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>1</td></tr></table>Sisa sekuens:
{5}</td><td>Proses elemen ketiga dari sekuens, yaitu simpul 4.
Busur (3,4) dimasukkan ke pohon.

Simpul dengan nilai 1 di tabel ‘derajat’:
{4,6}</td></tr>
</table>

<table>
<tr>
<td>  </td>
<td>
<table>
<tr><th>Simpul</th><th>Nilai</th></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>2</td><td>0</td></tr>
<tr><td>3</td><td>0</td></tr>
<tr><td>4</td><td>0</td></tr>
<tr><td>5</td><td>1</td></tr>
<tr><td>6</td><td>1</td></tr>
</table>
Sisa sekuens:
{}
</td>
<td>Proses elemen keempat dari sekuens, yaitu simpul 5.
Busur (4,5) dimasukkan ke pohon.

Simpul dengan nilai 1 di tabel ‘derajat’:
{5,6}</td>
</tr>
<tr>
<td>  </td>
<td>
<table>
<tr><th>Simpul</th><th>Nilai</th></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>2</td><td>0</td></tr>
<tr><td>3</td><td>0</td></tr>
<tr><td>4</td><td>0</td></tr>
<tr><td>5</td><td>1</td></tr>
<tr><td>6</td><td>1</td></tr>
</table>
</td>
<td>Hanya tersisa tepat dua simpul dengan nilai 1 di tabel ‘derajat’.
Masukkan busur (5,6)

Kondisi akhir.
Algoritma Selesai.</td>
</tr>
</table>

III. PEMBUKTIAN CAYLEY’S FORMULA

Pembuktian untuk teorema Cayley’s Formula dimulai dengan mendapatkan dua Lemma berikut.

1. Lemma 1: Untuk setiap pohon, terdapat tepat satu buah Prüfer Sequence unik untuk merepresentasikan pohon tersebut.
2. Lemma 2: Untuk setiap Prüfer Sequence, terdapat terdapat tepat satu buah pohon unik yang merupakan representasi dari Prüfer Sequence tersebut.

Kedua Lemma tersebut ditarik dari algoritma pembentukan Prüfer Sequence dari pohon dan sebaliknya.

Lemma 1 didapat dari algoritma pembentukan Prüfer Sequence. Satu-satunya kemungkinan pohon tersebut dapat menghasilkan lebih dari satu jenis Prüfer Sequence ialah ketika memilih simpul berderajat 1 untuk dimasukkan pada sekuens. Tapi, hal tersebut tidak akan terjadi, karena yang dipilih adalah sekuens dengan label terkecil, yang sudah pasti hanya akan ada satu buah.

Lemma 2 ditarik dari pembentukan pohon dari Prüfer Sequence. Satu-satunya kemungkinan Prüfer Sequence akan menghasilkan lebih dari satu jenis pohon adalah ketika memilih simpul dengan nilai di tabel ‘derajat’-nya sama dengan 1. Namun, algoritma akan memilih simpul dengan label terkecil, sehingga hanya akan menghasilkan satu jenis pohon.

Dengan adanya Lemma 1 dan Lemma 2, dapat ditarik kesimpulan bahwa jumlah pohon dengan *n* buah simpul berlabel, sama dengan jumlah Prüfer Sequence yang valid dengan *n*–2 buah elemen. Kesimpulan tersebut ditarik karena dengan adanya Lemma 1 dan Lemma 2, berarti untuk setiap sekuens dari set Prüfer Sequence yang valid memiliki tepat satu pasangan pohon dari set pohon rentang yang mungkin dibuat, yang berarti jumlah kedua set tersebut sama.

Kemudian, untuk menghitung jumlah Prüfer Sequence yang valid, perlu adanya satu lemma tambahan, yaitu sebagai berikut.

3. Lemma 3: Semua kemungkinan Prüfer Sequence yang dapat dibentuk adalah valid, dalam artian dapat dibuat pohon dari prufer sequence tersebut.

Untuk membuktikan Lemma 3 ini, dipakai metode kontradiksi. Anggap Lemma 3 tersebut salah, sehingga terdapat suatu Prüfer Sequence yang tidak dapat dibuat pohon, dengan kata lain yang membuat algoritma pembentukan pohon gagal. Kemungkinan algoritma pembentukan pohon gagal adalah ketika memilih simpul dari tabel yang bernilai satu, dan pada bagian akhir, ketika memilih dua simpul yang bernilai satu di tabel.

Untuk bagian pertama, algoritma akan gagal jika tidak menemukan simpul yang nilainya 1 dari tabel, sebut saja simpul yang bernilai satu di tabel tersebut simpul bebas. Hal itu tidak akan pernah terjadi. Pertama, karena Prüfer Sequence hanya memiliki *n*–2 elemen, sehingga dapat dipastikan pada kondisi awal, akan terdapat setidaknya 2 simpul bebas. Jumlah simpul bebas ini akan bertambah jika ada simpul yang muncul lebih dari satu kali pada sekuens. Hal ini juga didukung oleh fakta bahwa jumlah daun paling minimal dari sebuah pohon adalah 2.

Alasan kedua, pada saat pemrosesan sekuens, jumlah simpul bebas ini tidak akan pernah berkurang menjadi lebih kecil dari dua. Untuk memastikannya, lihat kedua kasus yang mungkin terjadi pada saat pemrosesan Sekuens.

Kasus pertama adalah elemen yang sedang diproses tidak ada lagi di sekuens. Pada kasus ini, satu simpul bebas akan dipakai satu, namun akan bertambah satu, karena elemen yang sedang diproses akan menjadi simpul bebas (karena tidak ada lagi di sekuens). Sehingga kasus ini tidak akan mengubah jumlah simpul bebas.

Kasus kedua adalah ketika elemen yang diproses masih ada di sekuens. Jumlah simpul bebas akan berkurang satu, tetapi tidak akan berkurang menjadi kurang dari dua. Penyebabnya adalah, untuk setiap duplikat elemen pada sekuens, terdapat tambahan satu buah simpul bebas di awal algoritma. Tambahan inilah yang dipakai ketika memroses elemen pada kasus ini. Sehingga jumlah simpul bebas tidak akan kurang dari 2, yang menyebabkan kemungkinan algoritma ini gagal akibat tidak ada simpul bebas tidak mungkin terjadi.

Kemungkinan kedua adalah algoritma gagal akibat pada akhir jumlah simpul bebas tidak sama dengan dua. Kemungkinan ini juga tidak mungkin terjadi, dengan alasan mirip dengan analisis sebelum ini. Pertama, jumlah simpul bebas tidak pernah kurang dari dua, berdasar analisis sebelumnya. Kedua, di akhir, jumlah simpul bebas tidak mungkin lebih dari dua. Karena, berdasar analisis sebelumnya, kemungkinan jumlah simpul lebih dari dua adalah ketika masih terdapat duplikat pada sekuens yang diproses. Di akhir algoritma, tidak ada lagi duplikat sekuens, karena semua sudah diproses, sehingga jumlah simpul bebas di akhir algoritma tidak mungkin lebih dari dua. Karena kedua hal itulah, jumlah simpul bebas di akhir pasti akan tepat dua buah.

Karena kedua kemungknan algoritma tersebut tidak mungkin terjadi, maka terdapat kontradiksi dengan asumsi bahwa algoritma pembentukan pohon ini akan gagal, sehingga Lemma 3 dapat dipastikan benar.

Karena semua kemungkinan Prüfer Sequence adalah valid, maka cukup menghitung kemungkinan Prüfer Sequence yang dapat dibentuk. Karena terdapat *n*–2 elemen, dan masik-masing elemen dapat bernilai 1 sampai *n*, maka jumlah kemungkinan Prüfer Sequence yang dapat dibentuk adalah:

$$\prod_{i=1}^{n-2} n$$

Yang ekivalen dengan:

$$n^{n-2}$$

Sesuai dengan Cayley's Formula. Sehingga terbukti bahwa Cayley's Formula adalah benar.

IV. KESIMPULAN

Menghitung banyaknya kemungkinan pohon rentang yang dapat dibuat dalam suatu graf lengkap dapat menggunakan Cayley's Formula. Cayley's Formula sudah terbukti pasti benar untuk semua graf lengkap berukuran *n*. Pembuktian dilakukan dengan bantuan Prüfer Sequence.

REFERENCES

[1] Munir, Rinaldi, Diktat Kuliah IF 2153, Matematika Diskrit, Edisi Keempat, Program Studi Teknik Informatika, STEI, ITB, 2008.

[2] Borchardt, C. W. (1860). "Über eine Interpolationsformel für eine Art Symmetrischer Functionen und über Deren Anwendung". Math. Abh. der Akademie der Wissenschaften zu Berlin: 1–20.

[3] Cayley, A. (1889). "A theorem on trees". Quart. J. Math 23: 376–378.

[4] Prüfer, H. (1918). "Neuer Beweis eines Satzes über Permutationen". Arch. Math. Phys. 27: 742–744.

[5] Jens Gottlieb, Bryant A. Julstrom, Günther R. Raidl, and Franz Rothlauf. (2001). "Prüfer numbers: A poor representation of spanning trees for evolutionary search". Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001): 343–350.

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, 17 Desember 2013

Muntaha Ilmi (13512048)