Pembuktian Benford's Law dengan Algoritma Brute Force terhadap Suatu Barisan Geometri Sembarang

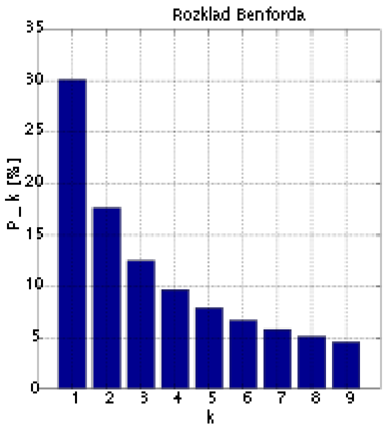
Azka Hanif Imtiyaz - 13514086
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514086@std.stei.itb.ac.id

Abstract**—***Benford's Law* atau sering juga disebut sebagai *first-digit law*, adalah suatu hukum mengenai distribusi frekuensi dari digit pertama dalam banyak tetapi tidak semua set data numerik. Pada makalah ini penulis mencoba menbahas mengenai pembuktian *Benford's Law* tersebut pada suatu barisan geometri sembarang.

Keywords—Benford's Law; Deret Geometri; Brute Force; Dataset

I. PENDAHULUAN

Data merupakan suatu hal yang kita jumpai setiap hari, mulai dari data populasi manusia dari suatu kota hingga data dari jumlah uang yang dimiliki oleh setiap manusia tersebut. Namun tanpa disangka, ternyata data yang seharusnya acak dan tidak memiliki pola ini ternyata justru memiliki pola. Pola ini didefinisikan dalam Hukum Benford, dimana untuk bilangan desimal, angka 1 sebagai digit pertama dalam frekuensi dari suatu data akan muncul kira-kira 30,1% dibandingkan dengan angka lainnya.



Gambar 1.
Distribusi dari digit pertama berdasarkan Hukum Benford. Setiap batang menyatakan suatu digit dan tinggi dari batang menyatakan persentase dari bilangan dengan digit batang tersebut sebagai digit pertama.
Sumber: Gknor.

Namun ternyata Hukum Benford tidak berlaku untuk seluruh data yang ada. Salah satu faktor yang mempengaruhi adalah angka yang dipengaruhi oleh pikiran manusia, sebagai contoh: harga barang yang ditetapkan oleh pedagang.

Dalam makalah ini, penulis akan mencoba menbahas mengenai pembuktian Hukum Benford tersebut pada suatu barisan geometri sembarang, dimana barisan tersebut memiliki rasio lebih besar dari 1 dan dimulai dari angka 1. Karena Hukum Benford telah dibuktikan tidak berlaku terhadap deret bilangan rasional yang mengecil. Dengan rasio lebih besar dari 1, maka deret bilangan akan selalu membesar.

II. LANDASAN TEORI

A. *Algoritma Brute Force*

Algoritma *brute force* adalah algoritma dengan pendekatan yang lempang (straightforward) untuk memecahkan suatu persoalan. Algoritma ini biasanya didasarkan pada pernyataan pada persoalan dan definisi konsep yang dilibatkan. Karakteristik *brute force* antara lain:

- Memecahkan persoalan dengan sangat sederhana.
- Memecahkan persoalan dengan langsung.
- Memecahkan persoalan dengan jalan yang jelas.

Algoritma *brute* force menyelesaikan suatu persoalan dengan cara menelusuri seluruh kemungkinan solusi secara ekstensif. Hal ini menyebabkan algoritma *brute force* pasti menemukan solusi yang optimum (jika ada). Namun apabila ruang lingkup solusi sangat besar, sebagai contoh dalam permainan catur, maka algoritma *brute force* akan memakan waktu yang sangat lama karena penulusuran semua kemungkinan solusi.

Source-code dari algoritma *greedy* pada perhitungan bilangan pertama adalah sebagai berikut.

```
int leadDigit (double a){
    double temp = a;
    while (temp > 9) temp /= 10;
    return temp;
}
```

Sedangkan *source-code* dari algoritma *greedy* pada perhitungan barisan geometri adalah sebagai berikut.

```
int main()
{
    int count1 = 0, count2 = 0, count3 = 0, count4
= 0, count5 = 0, count6 = 0, count7 = 0, count8 =
0, count9 = 0;
    double x = 1;
    int lead;

    float r;
    cin >> r;
    for (int i = 0; i < 500; i++){
        lead = leadDigit(x);
        cout << x << " " << lead << endl;
        if (lead == 1) count1++;
        else if (lead == 2) count2++;
        else if (lead == 3) count3++;
        else if (lead == 4) count4++;
        else if (lead == 5) count5++;
        else if (lead == 6) count6++;
        else if (lead == 7) count7++;
        else if (lead == 8) count8++;
        else if (lead == 9) count9++;
        x *= r;
    }
    cout << "Banyaknya 1: " << count1 << endl;
    cout << "Banyaknya 2: " << count2 << endl;
    cout << "Banyaknya 3: " << count3 << endl;
    cout << "Banyaknya 4: " << count4 << endl;
    cout << "Banyaknya 5: " << count5 << endl;
    cout << "Banyaknya 6: " << count6 << endl;
    cout << "Banyaknya 7: " << count7 << endl;
    cout << "Banyaknya 8: " << count8 << endl;
    cout << "Banyaknya 9: " << count9 << endl;
}
```

B. *Benford's Law*

Hukum Benford menyatakan bahwa, untuk suatu set bilangan yang memenuhi hukum benford, maka digit pertama dari bilangan tersebut akan muncul dengan probabilitas:

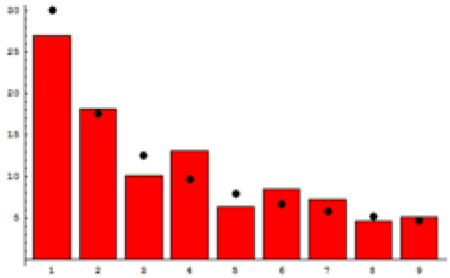
$$P(d) = \log_{10}(d+1) - \log_{10}(d) = \log_{10}\left(\frac{d+1}{d}\right) = \log_{10}\left(1+\frac{1}{d}\right).$$

Secara numerik, digit pertama bilangan mempunyai distribusi sebagai berikut, dimana d adalah digit pertama dan P(d) adalah probabilitas kemunculan dari d.

d	***P*(*d*)**
1	30.1%
2	17.6%
3	12.5%
4	9.7%
5	7.9%
6	6.7%
7	5.8%
8	5.1%
9	4.6%

Tabel 1.
Hukum Benford pada bilangan basis 10 (desimal)

Besarnya P(d) dan P(d+1) proporsional terhadap d dan d+1 dalam skala logaritmik.

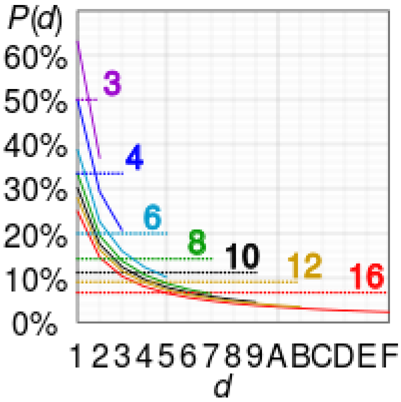


Gambar 2.
Distribusi dari digit pertama (dalam %) pada populasi di 237 negara di dunia pada Juli 2010. Titik hitam menyatakan prediksi distribusi oleh Hukum Benford.
Sumber: https://www.cia.gov/library/publications/the-world-factbook/rankorder/2119rank.html

Hukum Benford juga berlaku pada bilangan dengan basis lainnya, tidak hanya dalam basis 10. Rumus secara umumnya:

$$P(d) = \log_b(d+1) - \log_b(d) = \log_b\left(1+\tfrac{1}{d}\right).$$

Dimana b adalah basis dari bilangan yang digunakan.



Gambar 3.
Graf dari P(d) dalam basis yang bermacam-macam.
Sumber: HB.

Hukum Benford ini diketahui dapat diaplikasikan ke beberapa kriteria sebagai berikut.

- Ketika mean lebih besar dari median dan kemiringan dari statistik positif.
- Angka-angka yang dihasilkan dari kombinasi angka-angka, contoh: jumlah × harga.
- Data tingkat transaksi, contoh: penjualan.
- Angka-angka yang dihasilkan dari perhitungan kalkulasi apapun pada *Oughtred slide rule*, karena hasilnya akan masuk ke dalam distribusi logaritmik.

Hukum Benford diketahui tidak dapat diaplikasikan ke beberapa kriteria sebagai berikut.

- Ketika angka ditetapkan secara sekuensial.
- Ketika penetapan angka dipengaruhi oleh pikiran manusia.
- Rekening dengan minimum dan maksimum yang ditetapkan sebelumnya.
- Ketika tidak ada transaksi yang direkam.

Hukum Benford sebenarnya memungkinkan untuk menghitung probabilitas bertemunya suatu digit bilangan setelah pertama, atau digit ke-n, dengan rumus pada basis 10:

$$\log_{10}(n+1) - \log_{10}(n) = \log_{10}\left(1 + \frac{1}{n}\right)$$

C. *Barisan Geometri*

Barisan Geometri dapat didefinisikan sebagai barisan yang tiap-tiap sukunya didapatkan dari hasil perkalian suku sebelumnya dengan sebuah rasio tertentu.

Rasio pada suatu barisan dapat dirumuskan menjadi:

$$\mathbf{r = a_{k+1}/a_k}$$

Dimana a_k adalah sembarang suku dari barisan geometri yang ada. sementaraa_{k+1} adalah suku selanjutnya setelah a_k.

Dntuk menentukan suku ke-n dari sebuah barisan geometri, kita dapat menggunakan rumus:

$$\mathbf{U_n = ar^{n-1}}$$

Dimana ***a*** merupakan suku awal dan ***r*** adalah nilai rasio dari sebuah barisan geometri.

III. Analisis Dan Pembahasan

Pada analisis kali ini, penulis akan mencoba untuk menerapkan Hukum Benford pada barisan geometri dengan rasio 2, 3, dan 4, dengan nilai pertama pada barisan tersebut adalah 1.

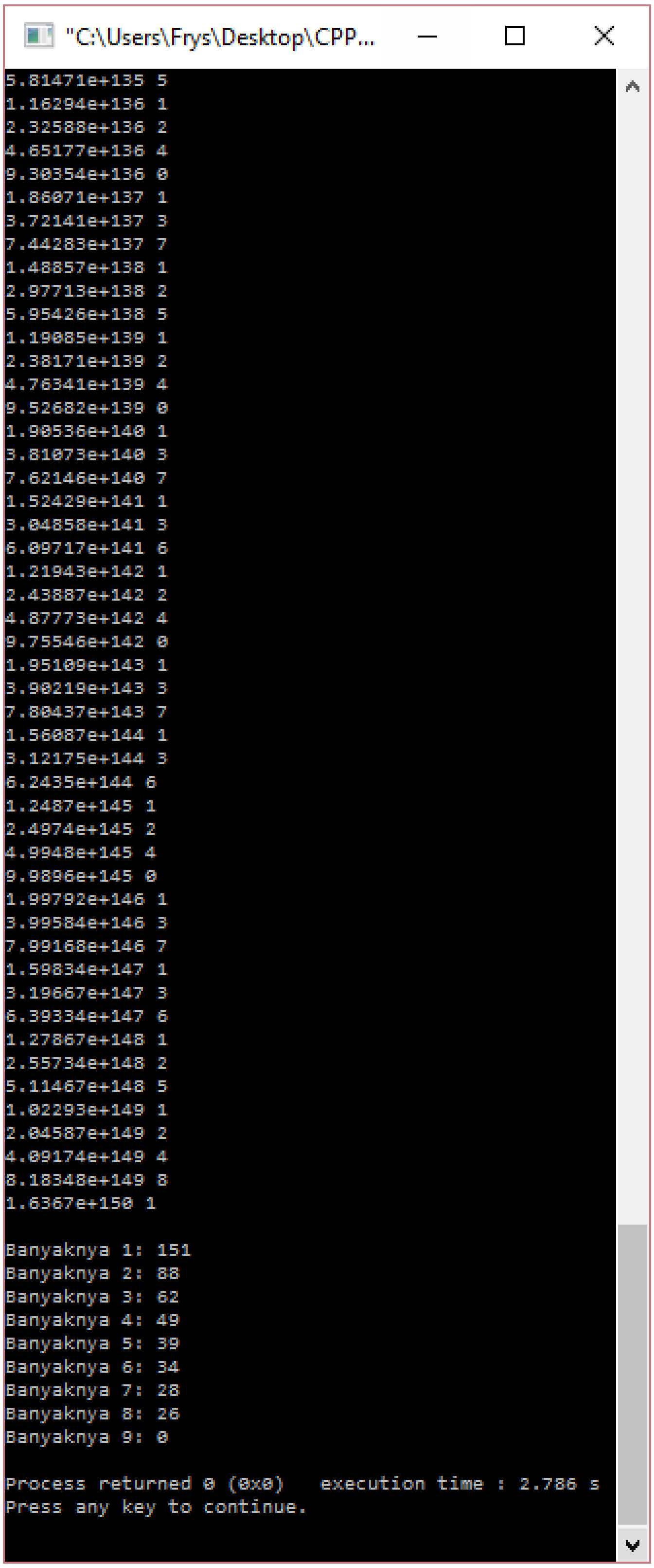
Analisis akan dilakukan dengan menguji data dengan barisan geometri yang berukuran 500 (jumlah sukunya 500), dengan rasio 2, 3, 4, dan nilai pertama pada barisan tersebut 1.

Setelah perhitungan barisan geometri, maka akan dihitung nilai dari *leading digit* atau digit pertama pada masing-masing bilangan. Kemudian akan dihitung totalnya dari masing masing bilangan (1,2,3,4,…). Setelah itu, akan dihitung probabilitas kemunculan dari masing masing bilangan sebagai *leading digit.*

Perhitungan dari *leading digit* akan dilakukan dalam suatu fungsi *leadDigit* dimana fungsi ini akan terus membagi nilai dengan 10, hingga akhirnya tidak bisa dibagi lagi. Hasilnya adalah *leading digit* dari bilangan tersebut.

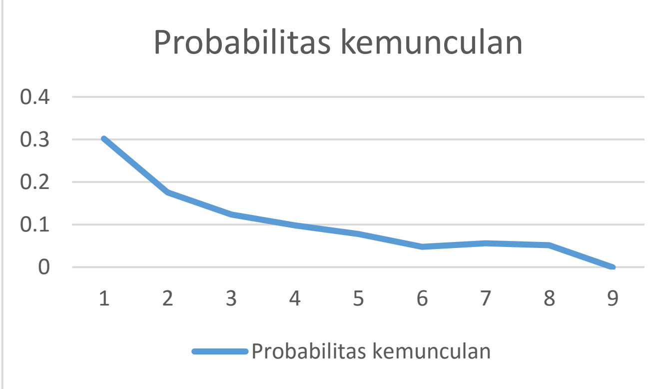
Ukuran 500 digunakan karena untuk menghindari dan mengurangi galat perhitungan (karena digunakan representasi *double* untuk variabel barisan geometri). Selain itu, untuk mengurangi waktu komputasi. Dan juga ukuran 500 sudah cukup besar untuk merepresentasikan Hukum Benford.

A. Barisan Geometri dengan Rasio 2



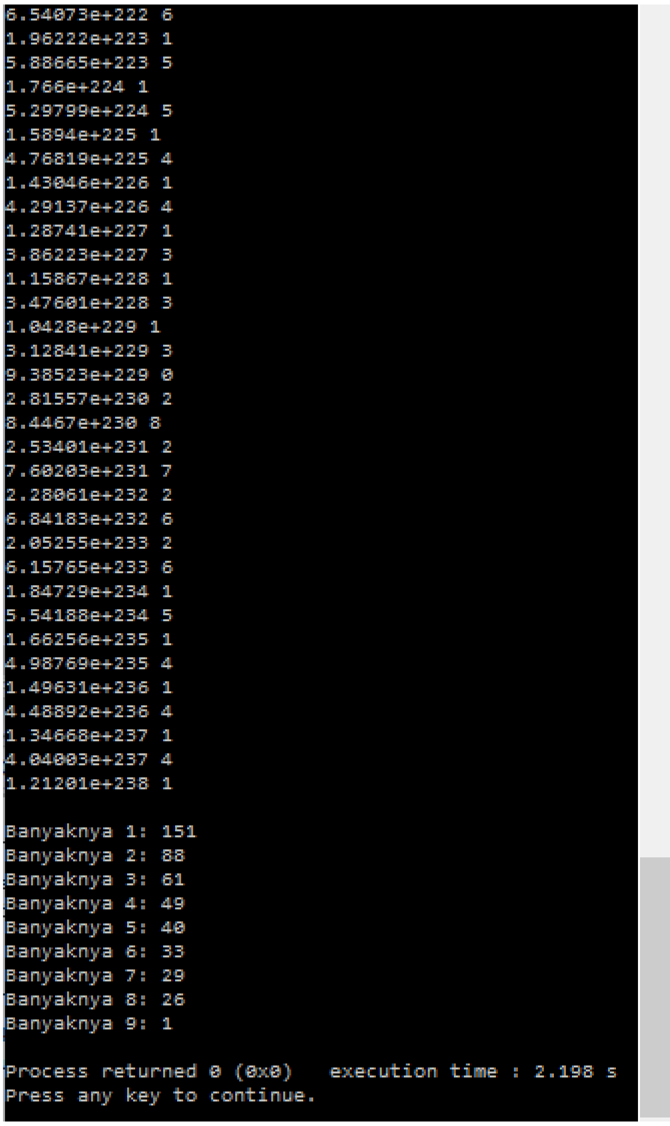
```
"C:\Users\Frys\Desktop\CPP...
5.81471e+135 5
1.16294e+136 1
2.32588e+136 2
4.65177e+136 4
9.30354e+136 0
1.86071e+137 1
3.72141e+137 3
7.44283e+137 7
1.48857e+138 1
2.97713e+138 2
5.95426e+138 5
1.19085e+139 1
2.38171e+139 2
4.76341e+139 4
9.52682e+139 0
1.90536e+140 1
3.81073e+140 3
7.62146e+140 7
1.52429e+141 1
3.04858e+141 3
6.09717e+141 6
1.21943e+142 1
2.43887e+142 2
4.87773e+142 4
9.75546e+142 0
1.95109e+143 1
3.90219e+143 3
7.80437e+143 7
1.56087e+144 1
3.12175e+144 3
6.2435e+144 6
1.2487e+145 1
2.4974e+145 2
4.9948e+145 4
9.9896e+145 0
1.99792e+146 1
3.99584e+146 3
7.99168e+146 7
1.59834e+147 1
3.19667e+147 3
6.39334e+147 6
1.27867e+148 1
2.55734e+148 2
5.11467e+148 5
1.02293e+149 1
2.04587e+149 2
4.09174e+149 4
8.18348e+149 8
1.6367e+150 1

Banyaknya 1: 151
Banyaknya 2: 88
Banyaknya 3: 62
Banyaknya 4: 49
Banyaknya 5: 39
Banyaknya 6: 34
Banyaknya 7: 28
Banyaknya 8: 26
Banyaknya 9: 0

Process returned 0 (0x0)   execution time : 2.786 s
Press any key to continue.
```

Gambar 4.
Beberapa bilangan terakhir dari barisan geometri dengan rasio 2 diikuti dengan digit pertamanya serta hasil dari banyaknya digit pertama pada barisan.



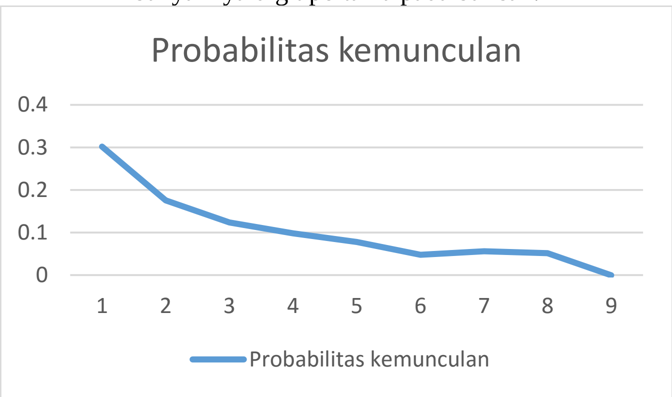
Grafik 1.
Grafik digit pertama dari bilangan terhadap probabilitas kemunculannya. Probabilitas kemunculan dihitung dengan banyaknya digit pertama bilangan tersebut muncul dibagi dengan besar dari barisan geometri.

B. Barisan Geometri dengan Rasio 3



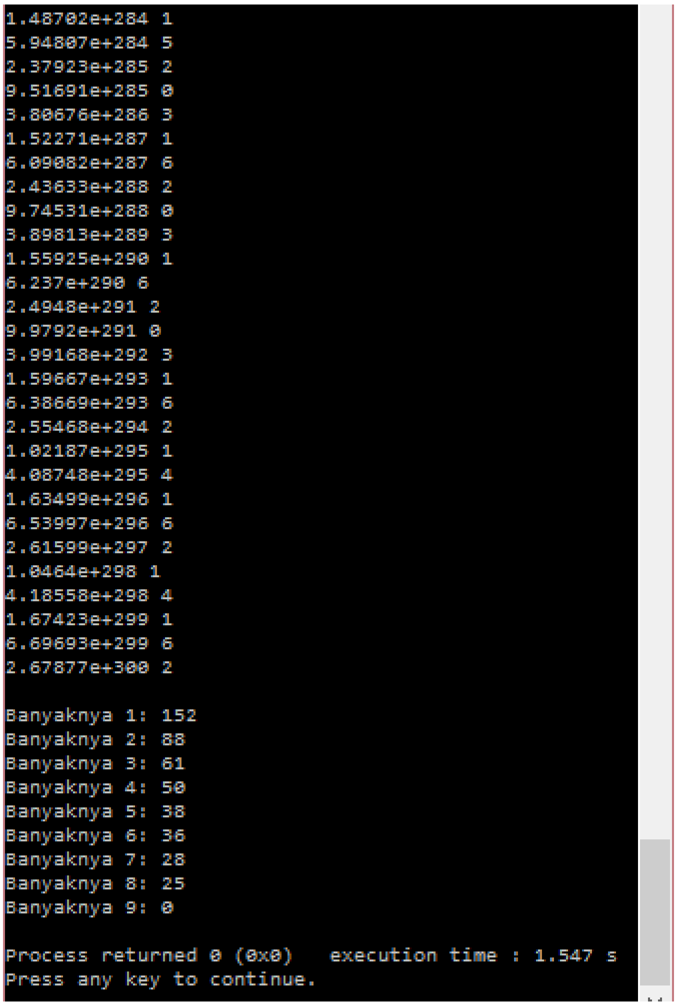
```
6.54073e+222 6
1.96222e+223 1
5.88665e+223 5
1.766e+224 1
5.29799e+224 5
1.5894e+225 1
4.76819e+225 4
1.43046e+226 1
4.29137e+226 4
1.28741e+227 1
3.86223e+227 3
1.15867e+228 1
3.47601e+228 3
1.0428e+229 1
3.12841e+229 3
9.38523e+229 0
2.81557e+230 2
8.4467e+230 8
2.53401e+231 2
7.60203e+231 7
2.28061e+232 2
6.84183e+232 6
2.05255e+233 2
6.15765e+233 6
1.84729e+234 1
5.54188e+234 5
1.66256e+235 1
4.98769e+235 4
1.49631e+236 1
4.48892e+236 4
1.34668e+237 1
4.04003e+237 4
1.21201e+238 1

Banyaknya 1: 151
Banyaknya 2: 88
Banyaknya 3: 61
Banyaknya 4: 49
Banyaknya 5: 40
Banyaknya 6: 33
Banyaknya 7: 29
Banyaknya 8: 26
Banyaknya 9: 1

Process returned 0 (0x0)   execution time : 2.198 s
Press any key to continue.
```

Gambar 5.
Beberapa bilangan terakhir dari barisan geometri dengan rasio 3 diikuti dengan digit pertamanya serta hasil dari banyaknya digit pertama pada barisan.



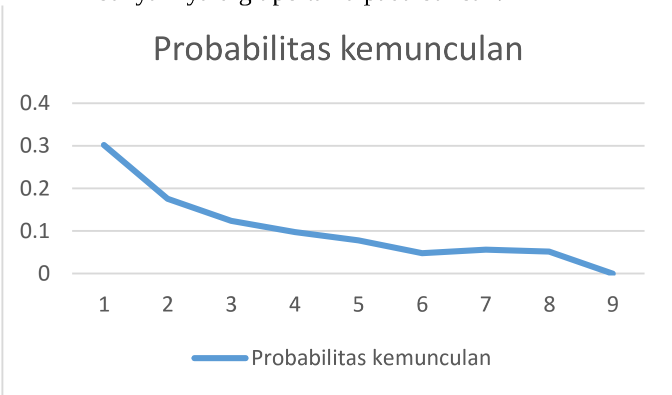
Grafik 2.
Grafik digit pertama dari bilangan terhadap probabilitas kemunculannya. Probabilitas kemunculan dihitung dengan banyaknya digit pertama bilangan tersebut muncul dibagi dengan besar dari barisan geometri.

C. Barisan Geometri dengan Rasio 4



```
1.48702e+284 1
5.94807e+284 5
2.37923e+285 2
9.51691e+285 0
3.80676e+286 3
1.52271e+287 1
6.09082e+287 6
2.43633e+288 2
9.74531e+288 0
3.89813e+289 3
1.55925e+290 1
6.237e+290 6
2.4948e+291 2
9.9792e+291 0
3.99168e+292 3
1.59667e+293 1
6.38669e+293 6
2.55468e+294 2
1.02187e+295 1
4.08748e+295 4
1.63499e+296 1
6.53997e+296 6
2.61599e+297 2
1.0464e+298 1
4.18558e+298 4
1.67423e+299 1
6.69693e+299 6
2.67877e+300 2

Banyaknya 1: 152
Banyaknya 2: 88
Banyaknya 3: 61
Banyaknya 4: 50
Banyaknya 5: 38
Banyaknya 6: 36
Banyaknya 7: 28
Banyaknya 8: 25
Banyaknya 9: 0

Process returned 0 (0x0)   execution time : 1.547 s
Press any key to continue.
```

Gambar 6.
Beberapa bilangan terakhir dari barisan geometri dengan rasio 4 diikuti dengan digit pertamanya serta hasil dari banyaknya digit pertama pada barisan.



Grafik 3.
Grafik digit pertama dari bilangan terhadap probabilitas kemunculannya. Probabilitas kemunculan dihitung dengan banyaknya digit pertama bilangan tersebut muncul dibagi dengan besar dari barisan geometri.

D. Uraian

Terdapat hal menarik yang terdapat pada barisan geometri sebelumnya, yaitu terdapat pola pengulangan pada *leading digit* dari barisan geometri.

Barisan *leading digit* dari barisan geometri dengan rasio 2:

1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 6 1 2 5 1 2 4 8 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 2 5 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 4 0 1 3 7 1 3 6 1 2 5 1 2 4 8 1

Barisan *leading digit* dari barisan geometri dengan rasio 3:

1 3 1 3 0 2 8 2 7 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 5 1 4 1 4 1 3 1 3 0 2 8 2 8 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 4 1 4 1 4 1 3 1 3 0 2 8 2 7 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 5 1 4 1 4 1 3 1 3 0 2 8 2 8 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 4 1 4 1 4 1 3 1 3 0 2 8 2 7 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2

6 1 5 1 5 1 4 1 4 1 3 1 3 0 2 8 2 8 2 7 2 6 1 5 1 5 1 4 1 4 1 3 1 3 1 3 0 2 8 2 7 2 6 2 6 1 5 1 4 1 4 1 4 1

Barisan *leading digit* dari barisan geometri dengan rasio 4:

1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 3 1 4 1 7 3 1 4 1 7 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 0 3 1 5 2 0 3 1 5 2 0 3 1 6 2 0 3 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 3 1 4 1 7 3 1 4 1 7 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 0 3 1 5 2 0 3 1 6 2 0 3 1 6 2 0 3 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 3 1 4 1 7 3 1 4 1 7 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 0 3 1 5 2 0 3 1 6 2 0 3 1 6 2 0 3 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 3 1 4 1 7 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 0 3 1 5 2 0 3 1 5 2 0 3 1 6 2 0 3 1 6 2 0 3 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 6 2 1 4 1 7 2 1 4 1 7 2 1 4 1 7 3 1 4 1 7 3 1 4 1 7 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 8 3 1 5 2 0 3 1 5 2 0 3 1 5 2 0 3 1 6 2 0 3 1 6 2 0 3 1 6 2 1 4 1 6 2 1 4 1 6 2

Pola ini, walaupun tidak tetap sama dari awal sampai akhir barisan, namun akan berulang beberapa kali sebelum pola tersebut berganti.

Dari hasil-hasil sebelumnya dapat dilihat bahwa distribusi frekuensi *leading digit* dari masing-masing barisan geometri serupa, dan tidak berbeda jauh.

Nilai dari frekuensi *leading digit* 1 adalah 151 untuk barisan geometri dengan rasio 2 dan 3, dan 152 untuk barisan geometri dengan rasio 4. Dengan demikian, probabilitas kemunculan untuk *leading digit* 1 pada barisan geometri kurang lebih sama.

IV. Kesimpulan

Setelah dilakukan analisis di atas dapat diambil kesimpulan bahwa Hukum Benford berlaku pada barisan geometri dengan rasio lebih besar dari 1 dan bilangan awal 1.

V. Ucapan Terima Kasih

Penulis ingin mengucapkan syukur kepada Tuhan Yang Maha Esa, karena hanya oleh karena rahmat-Nya penulis dapat menyelesaikan tulisan ini. Penulis juga ingin megucapkan berterima kasih kepada dosen mata kuliah IF2211 Strategi Algoritma, yaitu Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc. dan Bapak Dr. Ir. Rinaldi Munir atas bimbingan dan jasa beliau yang selama ini telah mengajar dan memberikan ilmu bagi penulis, sehingga penulis mampu membuat tulisan ini. Tak lupa juga penulis berterima kasih atas rekan - rekan mahasiswa Teknk Informatika 2014 yang senantiasa memberikan dorongan serta semangat kepada penulis.

Referensi

[1] Arno Berger and Theodore P Hill, Benford's Law Strikes Back: No Simple Explanation in Sight for Mathematical Gem, 2011

[2] http://mathworld.wolfram.com/BenfordsLaw.html (Diakses pada 8 Mei 2016)

[3] I Scott, P.D.; Fasli, M. (2001) "Benford's Law: An empirical investigation and a novel explanation". CSM Technical Report 349, Department of Computer Science, Univ. Essex

[4] Munir, Rinaldi. (2004). *Diktat Strategi Algoritmik*. Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung

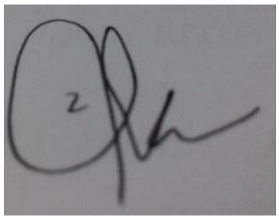
[5] Slide kuliah IF2211 Strategi Algoritma Algoritma Brute Force (2014).ppt

[6] http://www.rumusmatematikadasar.com/2015/01/materi-rumus-barisan-dan-deret-geometri-lengkap.html (diakses pada 8 Mei 2016)

Referensi

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 Mei 2016



Azka Hanif Imtiyaz - 13514086