

# Implementasi Random Numbers Java untuk Menciptakan Missing Value dari suatu Tabel

Jasman Pardede <sup>1)</sup>

<sup>1)</sup> Teknik Informatika Fakultas Teknologi Industri, Institut Teknologi Nasional  
Jl. PKH. Hasan Mustapa No.23, Bandung 40124 Indonesia  
email : pardede\_js@yahoo.com

## ABSTRAK

*Missing value* dapat diartikan sebagai data atau informasi yang “hilang” atau tidak tersedia mengenai subjek penelitian pada variabel tertentu akibat faktor *non sampling error*. *Missing value* merupakan hal yang sangat sering ditemui dan digunakan di dalam pengembangan aplikasi data mining dan beberapa kasus yang berhubungan dengan bidang statistik. Pada beberapa penelitian terhadap suatu algoritma yang berhubungan dengan penanganan *missing value*, seperti algoritma CB\*, TPDA, sangat membutuhkan data *missing value*. Hasil algoritma dengan *missing value* diharapkan akan menghasilkan kesimpulan yang hampir sama dengan data lengkap. Untuk membuktikan bahwa suatu algoritma dengan data lengkap dan data yang memiliki *missing value* memiliki kesimpulan yang hampir sama, maka diperlukan suatu data lengkap dan data *missing value* yang diperoleh dari data lengkapnya. Untuk itu diperlukan suatu aplikasi yang dapat menciptakan suatu data *missing value* yang diperoleh dari data lengkapnya. Pada penelitian ini akan mengimplementasikan *Random Numbers Java*, yaitu class **Random** yang terdapat pada package **java.util**, untuk menciptakan *missing value* dari suatu tabel yang memiliki data lengkap berdasarkan persentasi data yang akan dibangkitkan *missing value*-nya.

**Kata Kunci** : *Random Numbers*, Java, *missing value*, algoritma, tabel

## 1. Pendahuluan

*Missing value* dapat diartikan sebagai data atau informasi yang “hilang” atau tidak tersedia mengenai subjek penelitian pada variabel tertentu akibat faktor *non sampling error*. Hal ini disebabkan oleh beberapa hal diantaranya non-respon, kelalaian saat melakukan perekaman data atau bahkan oleh hal lain yang tidak diketahui sebabnya. *Missing value* merupakan hal yang sangat sering ditemui dan digunakan di dalam pengembangan aplikasi data mining dan beberapa kasus yang berhubungan dengan bidang statistik. Saat ini telah banyak metode yang digunakan dalam mengambil informasi di dalam kumpulan data yang lengkap diantaranya, algoritma B, algoritma PC dan lain sebagainya [7]. Disamping itu terdapat banyak metode yang digunakan di dalam menangani masalah nilai tidak lengkap (*missing value*) diantaranya algoritma K2, CB, BC, BSEM, CB\* dan TPDA.

Untuk membuktikan kedua metode itu dapat menghasilkan informasi yang hampir sama maka dapat dilakukan dengan mengimplementasikan data yang tidak ada *missing value* pada metode yang tidak ada *missing value* dan kemudian dari data awal dibangkitkan data yang *missing value* kemudian diterapkan salah satu metode yang menggunakan *missing value* apakah memberikan informasi yang hampir mendekati atau sama.

Untuk membantu di dalam pembuktian kedua metode tersebut, maka pada penelitian ini peneliti akan mengimplementasikan *Randomize Java* untuk menciptakan tabel *missing value*. Sehingga akan dapat memberikan informasi bahwa kedua metode dapat dianggap menghasilkan informasi yang hampir mendekati nilai kebenarannya. Penelitian ini bertujuan untuk melakukan analisis, perancangan, dan pengimplementasian *random numbers java* untuk menciptakan *missing value* dari suatu tabel.

## 2. Studi Konsep Penciptaan Missing Value

Pengembangan aplikasi perangkat lunak yang dapat memfasilitasi penciptaan *missing value* pada suatu tabel perlu memperhatikan beberapa hal penting yaitu pengertian bilangan acak, penciptaan bilangan acak pada java, pengertian *missing value*, serta algoritma penciptaan *missing value* yang digunakan pada penelitian ini.

### 2.1. Bilangan Acak (Random Number)

Dasar pengembangan studi simulasi adalah kemampuan untuk menghasilkan bilangan acak, dimana suatu bilangan acak mewakili nilai suatu variabel acak yang didistribusikan secara seragam pada (0,1). Bilangan acak berarti suatu bilangan yang diambil dari sekumpulan bilangan, dimana tiap-tiap elemen dari kumpulan bilangan tersebut mempunyai peluang yang sama untuk terambil [4]. Proses pengambilan bilangan acak itu sendiri merupakan suatu *random event* (peristiwa acak), yang berarti peristiwa dimana proses dan hasilnya “tidak dapat” diprediksi. Algoritma pembangkit bilangan acak harus memenuhi syarat-syarat yang telah ditentukan pada *Request For Command 1750* (RFC 1750), yaitu [6]:

1. Tidak dapat diprediksi, suatu algoritma pembangkit bilangan acak harus bersifat tidak dapat diprediksi hasil keluarannya.
2. Tahan akan *skew* (pembelokan), bila perangkat keras dimanipulasi maka algoritma dapat tahan terhadap manipulasi sumber bilangan tersebut hingga level tertentu yang tidak mungkin lagi dapat ditahan.
3. Memiliki fungsi penggabungan dan pengacakan, untuk mencegah penyerangan terhadap algoritma secara keseluruhan.

Pembangkitan bilangan acak terbagi atas dua tipe, yaitu Pembangkitan Bilangan Acak Semu (*pseudo-random*), menggunakan rutinitas algoritma yang mengimplementasikan rumus matematika untuk menghasilkan bilangan yang terlihat acak, dan Pembangkitan Bilangan Acak Penuh (*trully random*), menghasilkan baris bilangan dari ekstraksi fenomena aktivitas fisika dan kemudian mengirimkannya kepada komputer melalui interface dengan port serial atau parallel. Kebanyakan bahasa komputer sudah mempunyai pembangkit bilangan acak terpasang yang dapat dipanggil untuk membangkitkan bilangan acak [3].

### 2.2. Random Numbers Java

Java menyediakan dua pendekatan dalam pembangkitan bilangan acak, yaitu melalui class **java.lang.Math** dan **java.util.Random**. Class **java.lang.Math** mendefinisikan metode *random* yang akan mengembalikan bilangan *pseudo-random* antara 0.0 (inklusif) dan 1.0

(eksklusif). Jenis bilangan acak yang dapat diperoleh pada class **Math** adalah suatu nilai bertipe **double**. Class **Random** yang terdapat pada package **java.util** adalah pembangkit bilangan *pseudo-random* dengan menggunakan algoritma yang diambil dari *The Art of Computer Programming* oleh Donald Knuth [5]. Untuk menghasilkan bilangan acak yang lebih baik, kita dapat menggunakan class **java.util.Random**. Class ini menyediakan pendekatan yang lebih praktis dan mendukung tipe-tipe lain selain **double**, termasuk distribusi Gaussian.

### 2.3. Missing Value

*Missing value* merupakan salah satu penyebab suatu data dikatakan tidak valid atau masih merupakan data 'kotor'. *Missing value* dengan jumlah banyak akan menghilangkan sejumlah informasi yang seharusnya dibutuhkan untuk diolah. *Missing value* dapat diartikan sebagai data atau informasi yang "hilang" atau tidak tersedia mengenai subjek penelitian pada variabel tertentu akibat faktor *non sampling error*. Faktor *non sampling error* yang dimaksud adalah [1] :

1. *Interviewer recording error*, terjadi akibat kealpaan petugas pengumpul data (pewawancara), misalnya ada sejumlah pertanyaan yang terlewatkan.
2. *Respondent inability error*, terjadi akibat ketidakmampuan responden dalam memberikan jawaban akurat, misalnya karena tidak memahami pertanyaan, bosan atau kelelahan (*respondent fatigue*) akhirnya responden mengosongkan sejumlah pertanyaan atau berhenti mengisi kuesioner di tengah jalan.
3. *Unwillingness respondent error*, terjadi karena responden tidak berkenan memberikan jawaban yang akurat, misalnya pertanyaan soal penghasilan, usia, berat badan, pengalaman melakukan pelanggaran hukum, dan lain-lain.

Saat ini telah ada beberapa metode yang digunakan di dalam menangani *missing value* diantaranya **Metode K-Means Imputation**. Penggunaan *missing value* sangat penting peranannya dalam pengembangan aplikasi data mining, diantaranya algoritma yang berhubungan dengan *missing value* adalah algoritma K2, CB, BC, BSEM, CB\*, TPDA dan lain-lain.

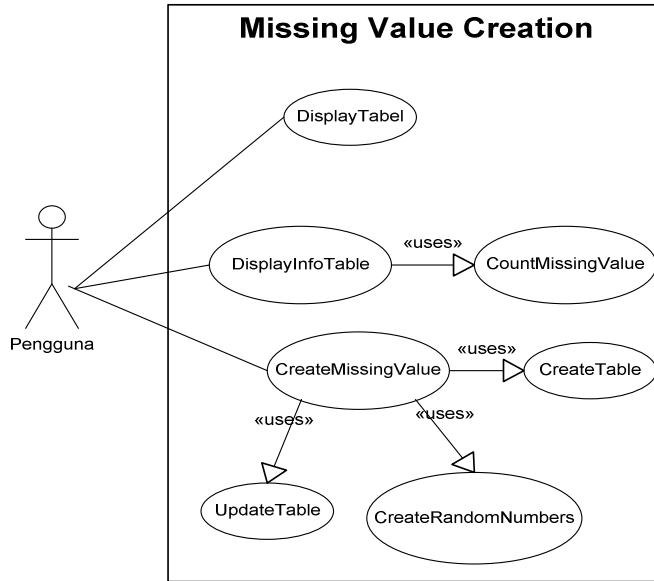
### 2.4. Algoritma Menciptakan Missing Value

Algoritma yang digunakan untuk menciptakan *missing value* pada penelitian ini adalah sebagai berikut :

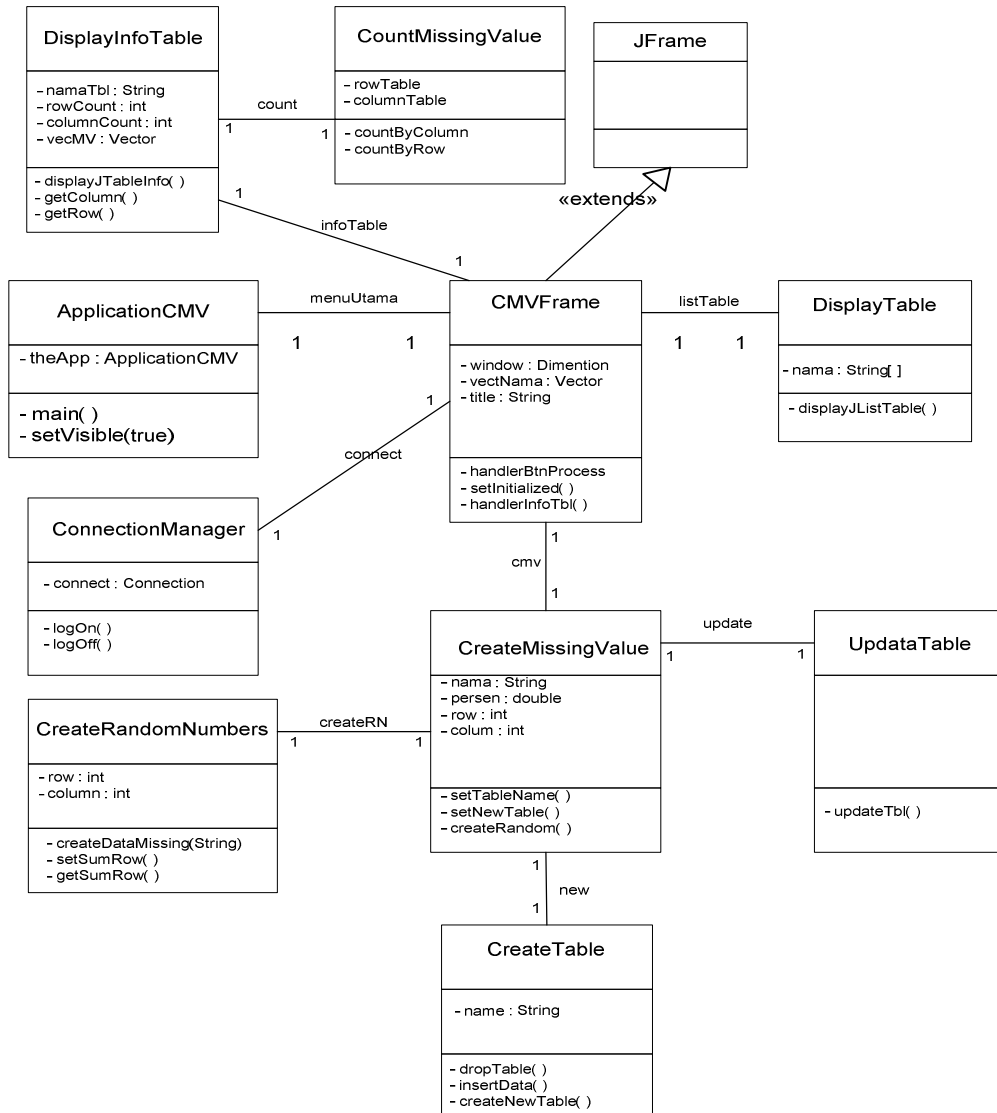
- a. Tentukan persentasi data yang akan dibangkitkan *missing value*-nya.
- b. Hitung banyaknya baris dan kolom suatu tabel yang akan dibangkitkan *missing value*-nya.
- c. Bangkitkan bilangan acak dalam menentukan baris yang menjadi baris data yang hilang.
- d. Bangkitkan bilangan acak dalam menentukan kolom yang menjadi kolom data yang hilang.
- e. Cek apakah pada baris dan kolom yang diperoleh dari (c) dan (d) telah memiliki nilai "null", jika ya ulangi langkah (c) dan (d). Jika tidak masuk ke langkah (f).
- f. Cek apakah pengubahan nilai baris dan kolom pada langkah (c) dan (d) mengakibatkan penghapusan satu baris (row) tabel, jika ya ulangi langkah (c) dan (d). Jika tidak masuk ke langkah (g).
- g. Ubah nilai tabel yang bersesuaian dengan baris dan kolom tabel yang didapatkan dari (c) dan (d) dengan nilai "null".
- h. Lakukan proses (c) sampai (g) secara berulang sampai di dapatkan persentasi yang diminta sesuai dengan yang diberikan pada (a).

## 3. Hasil Percobaan

Untuk menghasilkan bilangan acak dalam menentukan baris dan kolom suatu tabel yang akan dihilangkan, pada penelitian ini peneliti menggunakan class **java.util.Random**, yaitu class yang penggunaannya menggunakan pendekatan yang lebih praktis dan juga mendukung tipe bilangan bulat (**int**). Penciptaan nilai *missing value* dari suatu tabel pada penelitian ini, mengikuti algoritma yang dinyatakan pada sub bab 2.4. Kebutuhan perangkat lunak dalam mendukung penciptaan *missing value*, memerlukan beberapa fungsionalitas seperti yang dinyatakan pada use case diagram pada Gambar 1.



Gambar 1 Use case diagram



Gambar 2 Model konseptual aplikasi penciptaan missing value

### 3.1. Perancangan Sistem

Berdasarkan kebutuhan fungsionalitas yang harus dipenuhi dalam memfasilitasi kebutuhan perangkat lunak seperti yang dinyatakan pada use case diagram pada Gambar 1, maka peneliti mendapatkan model konseptual dari aplikasi penciptaan *missing value* seperti yang dinyatakan pada Gambar 2.

### 3.2. Implementasi Sistem

Perangkat lunak aplikasi, memerlukan dukungan perangkat lunak lain dalam implementasinya seperti, sistem operasi Windows atau Linux, bahasa pemrograman java dan database MySQL Server 5.0.

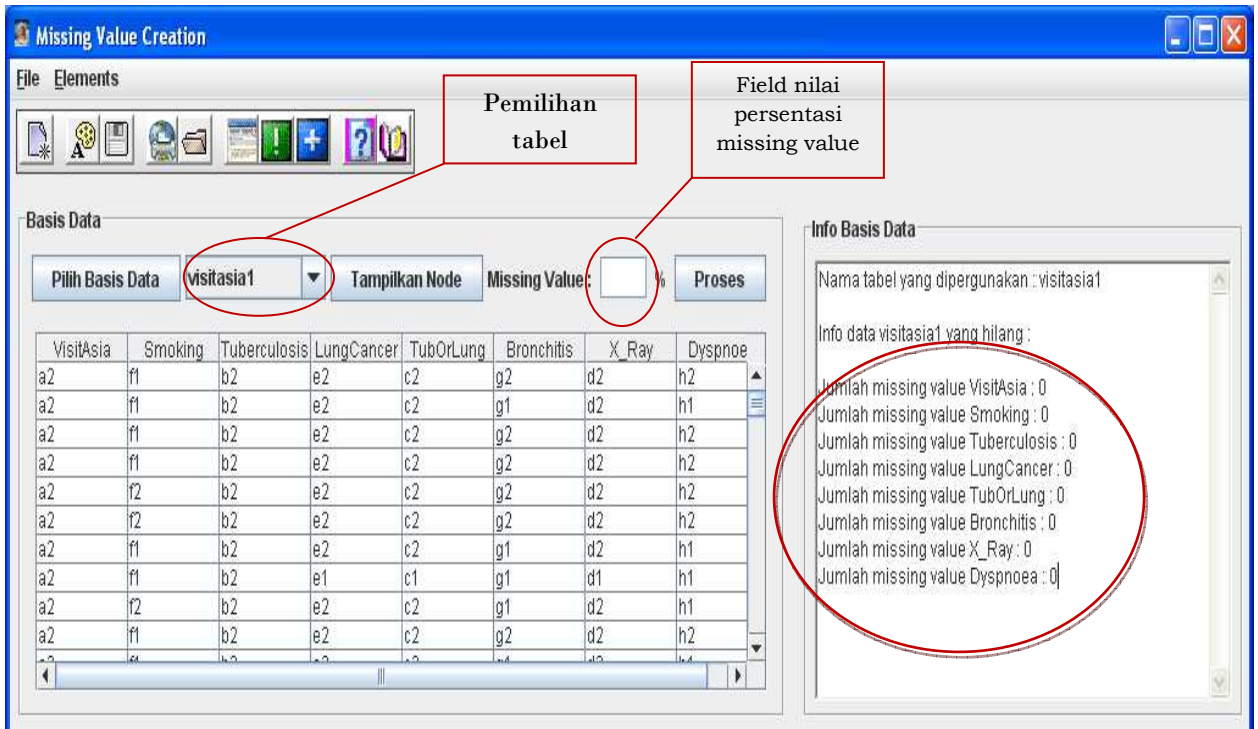
### 3.3. Teknik Pengujian

Teknik pengujian yang digunakan adalah teknik pengujian *black box testing*. Pengujian ini memungkinkan pemrogram untuk memperoleh sekumpulan kondisi masukan (*input*) yang akan secara penuh menjalankan semua kebutuhan fungsional untuk sebuah program. Dalam hal ini diambil salah satu butir uji, yaitu **CreateMissingValue** seperti yang ditunjukkan pada Tabel 1.

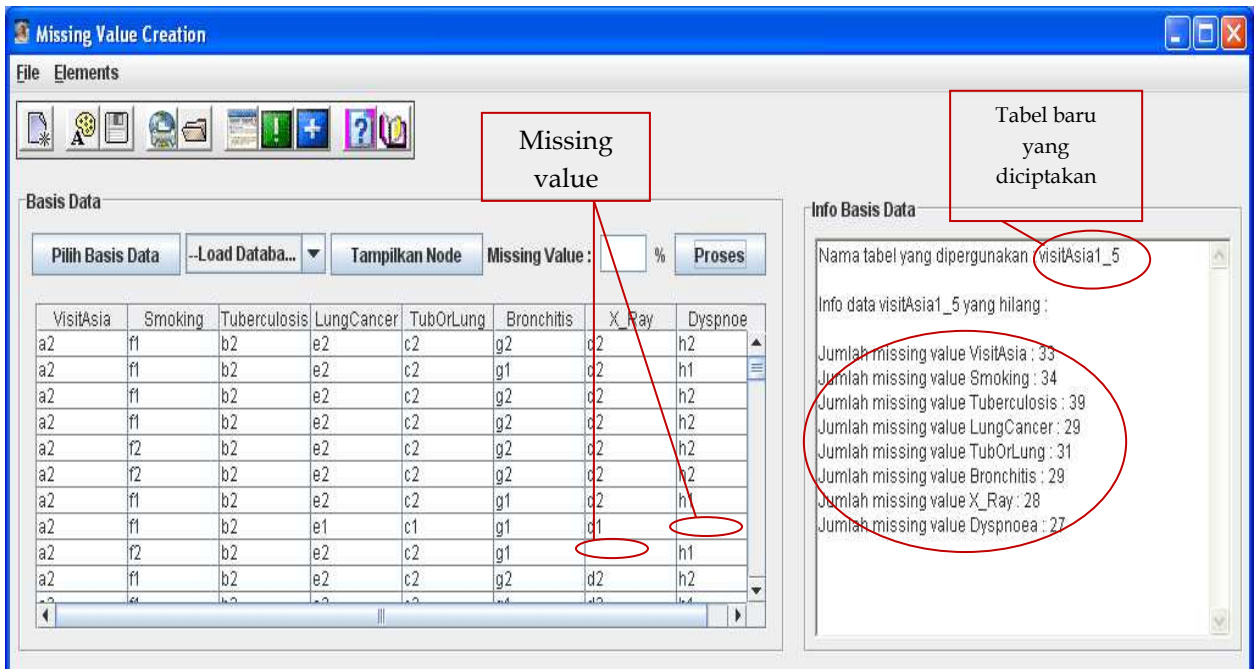
Tabel 1 Pengujian create missing value 5%

<b>Identifikasi</b>	<b>AMVC-01</b>		
<b>Nama Butir Uji</b>	CreateMissingValue		
<b>Tujuan</b>	Media dalam menciptakan tabel <i>missing value</i> dan menyimpan hasil penciptaan data <i>missing value</i> ke tabel baru		
<b>Deskripsi</b>	Pengguna menulis <b>nilai persentasi</b> ke teks field aplikasi kemudian tekan " <b>Proses</b> "		
<b>Kondisi Awal</b>	Pengguna telah memilih tabel yang akan diciptakan <i>missing value</i> -nya		
<b>Pengujian</b>			
<b>Skenario Uji</b>			
1. Masukkan <b>nilai persentasi</b> yang akan diciptakan nilai <i>missing value</i> -nya			
2. Tekan " <b>Proses</b> "			
<b>Kriteria Evaluasi Hasil</b>			
Pengguna mempunyai hak untuk menciptakan tabel baru pada suatu database			
<b>Kasus dan Hasil Uji (Data Normal)</b>			
Masukan	Harapan	Pengamatan	Kesimpulan
<b>Bilangan Real</b>	Ciptakan tabel <i>missing value</i> dari tabel yang telah dipilih dengan nama tabel, <b><i>namaTabelAwal_persentasi</i></b> , kemudian jumlahkan nilai <i>missing value</i> dari setiap field yang dimiliki, kemudian tampilkan ke pengguna.	Ciptakan tabel <i>missing value</i> dari tabel yang telah dipilih dengan nama tabel, <b><i>namaTabelAwal_persentasi</i></b> , kemudian jumlahkan nilai <i>missing value</i> dari setiap field yang dimiliki, kemudian tampilkan ke pengguna.	<b>[X]</b> Terima <b>[ ]</b> Tolak

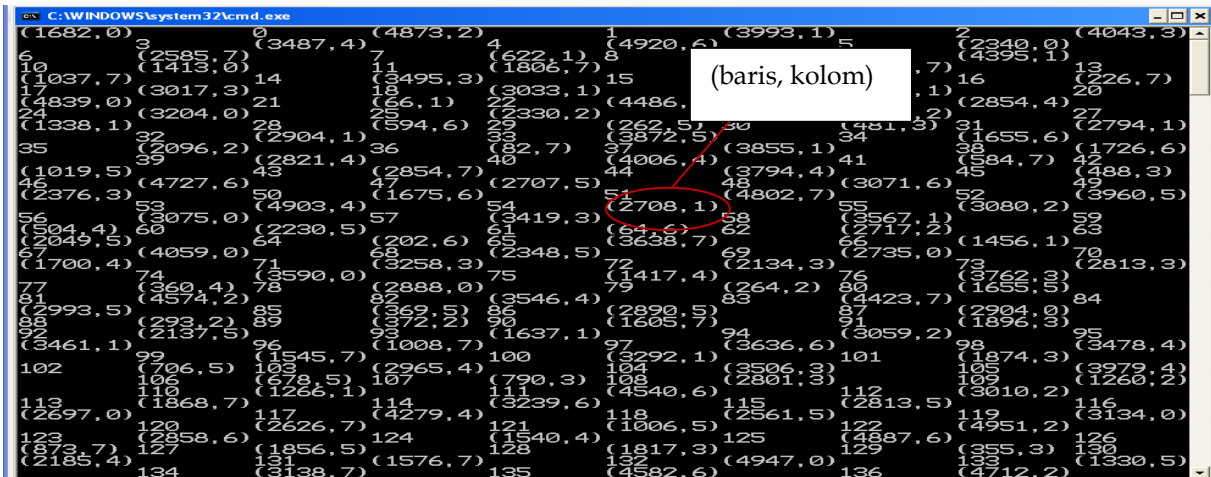
Pada pengujian ini peneliti menggunakan tabel **visitAsia1** dengan jumlah data sebanyak 5000 data. Informasi tabel **visitAsia1** direpresentasikan seperti pada Gambar 3. Hasil pengujian **create missing value** dengan persentasi 5 % yang dilakukan oleh pengguna pada tabel **visitAsia1** direpresentasikan seperti pada Gambar 4. Proses pembangkitan bilangan acak dalam penentuan baris dan kolom yang akan dihilangkan datanya saat pengujian ini, diperoleh bahwa pada penghilangan data pertama dilakukan pada baris 1683 dan kolom 1, data kedua dilakukan pada baris 4874 dan kolom 3 dan seterusnya sampai 250 data telah dihilangkan, seperti yang ditunjukkan pada Gambar 5.



Gambar 3 Display informasi tabel visitAsia1



Gambar 4 Hasil penciptaan missing value 5 % dari tabel visitAsia1



Gambar 5 Proses penciptaan bilangan acak dalam menentukan baris dan kolom

#### 4. Kesimpulan

Berdasarkan penelitian yang dilakukan, beberapa kesimpulan yang dapat diambil sebagai berikut :

1. Pada penelitian ini telah mengimplementasikan Randomize Java, class **Random** yang ada pada package **java.util**, untuk menciptakan tabel *missing value* dari suatu tabel dengan mengikuti algoritma yang dinyatakan pada sub bab 2.4.
2. Aplikasi yang dikembangkan dapat menciptakan nilai *missing value* sesuai dengan persentasi yang diharapkan pengguna dengan nilai *missing value* yang dapat berubah-ubah walaupun pada tabel yang sama dengan persentasi yang sama.

#### Daftar Pustaka

- [1] Acock, A.C (2005), Working With Missing Values, Oregon State University, <http://people.oregonstate.edu/~acock/gwoth-curves/worling%20with%20missing%20value.pdf> (diakses pada Juni 2010)
- [2] Gosling, James et al. (1996). *The Java Language Specification*, Sun Microsystems.
- [3] Knudsen, J.B. (1998). *Java Cryptography, First Edition*, O'Reilly.
- [4] LavaRnd, Terms & Definitions : Pseudo-Random Number Generator, <http://www.lavarnd.org/faq/prng.html> (diakses pada Juni 2010)
- [5] Naughton, P.(1996), *The Java HandBook*, McGraw-Hill, Inc.
- [6] Request For Commands 1750. <http://www.ietf.org/rfc/rfc1750.txt>. (diakses pada juni 2010).
- [7] Sophia, A.(2005), Algoritma PC sebagai Alternatif Pendekatan Analisis Dependensi untuk Konstruksi Struktur Bayesian Network dalam Data Mining, Laporan Tugas Akhir, Departemen Teknik Informatika ITB