Diktat Kuliah

Pengantar Metode Simulasi Statistika dengan Aplikasi R dan S^{+}

Oleh ::

Drs I Made Tirta, M.Sc. Ph.D.

Jurusan Matematika-Fakultas Matematika dan IPA

Universitas Jember

Tahun 2003

Tirta, I Made

Pengantar Metode Simulasi

Diterbitkan oleh Unit Penerbit FMIPA Universitas Jember

ALamat	:	Jalan Kalimantan No 37 Jember 68121
No. Tlp	:	0331 330 225,; 0331 334 293
Fax.	:	0331 330 225
Email	:	fmipa.unej@telkom.net

Naskah diktat ini sepenuhnya ditulis dengan menggunakan LaTeX, sedangkan grafik dihasilkan dengan S-Plus atau R. Naskah dicetak dengan HP Laser Jet 4050.

PRAKATA

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa yang telah memberi kekuatan dan kesempatan sehingga diktat kuliah ini bisa terselesaikan meskipun setelah kuliah dimulai beberapa minggu. Tujuan utama penulisan diktat ini adalah sebagai bahan bacaan bagi mahasiswa yang menempuh mata kuliah **Metode Simulasi** dan diktat ini disusun sedemikian sehingga diharapkan dapat memudahkan mahasiswa kalaupun mau belajar sendiri.

Untuk membantu pemahaman yang lebih baik, ada beberapa hal yang harus diperhatikan mahasiswa dalam menggunakan diktat ini diantaranya:

1. pada setiap awal bab, diberikan tujuan umum dan tujuan khusus, yang diharapkan dapat membantu mahasiswa memusatkan perhatian yang lebih banyak kepada hal-hal yang dianggap penting;

2. pada setiap akhir bab diberikan sumber bacaan yang bisa dicari mahasiswa untuk lebih mendalami hal-hal yang menarik perhatian dan minatnya;

3. kepada para mahasiswa diharapkan menyempatkan diri untuk membaca, baik

sebelum maupun sesudah kuliah berlangsung, sehingga selain diharapkan dapat mengikuti kuliah lebih baik, juga akan terjadi pengendapan yang lebih baik terhadap materi yang diajarkan.

Disadari betul bahwa pada terbitan pertama ini, masih banyak hal-hal yang perlu mendapat perhatian untuk disempurnakan. Kepada pembaca umumnya, teman sejawat dan mahasiswa peserta kuliah khususnya, diharapkan dapat memberikan masukan berupa saran, kritik dan koreksi demi kesempurnaan diktat ini pada cetakan berikutnya.

Kepada semua pihak yang telah membantu sampai tercetaknya diktat ini penulis sampaikan terimakasih dan penghargaan yang sebesar- besarnya. Semoga diktat ini dapat memberikan manfaat sebagaimana diharapkan.

Jember, Maret 2004 Penulis

DAFTAR ISI

1 PRINSIP DAN MANFAAT SIMULASI 1

1.1 Prinsip Simulasi 3

1.2 Manfaat dan Peran Simulasi 5

1.3 Langkah-langkah simulasi 6

1.4 Aplikasi Simulasi 7

2 TEKNIK DASAR SIMULASI PEUBAH ACAK 9

2.1 Membangkitkan Bilangan Acak 10

2.2 Membangkitkan Data dari Distribusi Tertentu 17

2.3 Pemeriksaan Secara Emperik 17

2.4 Daftar Bacaan 20

2.5 Soal-soal Latihan 20

3 MEMBANGKITKAN DATA DENGAN TRANSFORMASI LANGSUNG 23

3.1 Transformasi Peubah Acak 24

3.2 Data dengan Transformasi Box-Muller 26
3.3 Membangkitkan Data $N(\mu, \sigma^2)$ 28
3.4 Membangkitkan Data Keluarga $G(\alpha, \beta)$ 30
3.5 Membangkitkan Data Distribusi t dan F 32
3.6 Data dengan Distribusi Normal 34
3.7 Daftar Bacaan 37
3.8 Latihan Soal-soal 37

4 MEMBANGKITKAN DATA DENGAN TRAFORMASI TAK LANGSUNG 39
4.1 Metode Transformasi Invers 40
4.2 Metode Monte Carlo 46
4.2.1 Sejarah Singkat 46
4.2.2 Metode Penerimaan dan Penolakan (*Acceptance-Rejection*) 48
4.3 Menghitung Pendekatan Pi 55
4.3.1 Sejarah Perhitungan Pi 55
4.3.2 Percobaan Jarum Buffon 57
4.3.3 Menghitung π dengan Monte Carlo 60
4.4 Daftar Bacaan 63
4.5 Latihan Soal-soal 64

DAFTAR TABEL

2.1 Barisan Bilangan dari Cara Kongruen dengan $m = 6, a = 5, c = 3$. . 12

2.2 Barisan Bilangan dari Cara Kongruen dengan $m = 7, a = 1, c = 5$. 14

2.3 Daftar berbagai nilai parameter a, c dan m 16

4.1 Perhitungan π secara Analitik 55

4.2 Perhitungan π dengan Mesin 56

DAFTAR GAMBAR

2.1 Rataan dan Ragam sampel untuk Berbagai Ukuran Sampel 19

2.2 Grafik Densitas Data dibandingkan dengan Densitas Normal Standar 21

3.1 Grafik Densitas dari Berbagai Data Bangkitan Berdistribusi Normal 28

3.2 Rataan Data Berdistribusi Normal denganBerbagai Ukuran Sampel 29

3.3 Grafik Sebaran Data dengan Distribusi Bivariat Normal 36

4.1 Grafik Fungsi Kepadatan dan Fungsi Kumulatif 41

4.2 Grafik Densitas dan Kumulatif Distribusi Eksponensial 44

4.3 Ilustrasi Peluang dengan Luas Daerah 49

4.4 Ilustrasi Prinsip Penerimaan dan Penolakan 49

4.5 Ilustrasi Percobaan Jarum dari Buffon 59

4.6 *Hit-Miss* Monte Carlo meniru Percobaan Buffon 59

4.7 Ilustrasi Penerimaan dan Penolakan untuk Menghitung π dengan Metode Monte Carlo 62

BAB 1

PRINSIP DAN MANFAAT SIMULASI

Tujuan Umum

Untuk memahami prinsip dan jenis simulasi serta peran dan manfaatnya dalam berbagai bidang.

Tujuan Khusus

Pada akhir pembahasan mahasiswa diharapkan dapat menyebutkan:

1. prinsip dan jenis simulasi;
2. peran dan manfaat simulasi;
3. langkah-langkah dalam melakukan simulasi;
4. beberapa aplikasi simulasi.

Materi

1. Pendahuluan

2. Prinsip dan jenis simulasi

3. Peran dan manfaat simulasi

4. Langkah-langkah simulasi

5. Aplikasi simulasi

Pendahuluan

Salah satu peran yang harus dijalankan oleh para statistisi (ahli statistika) adalah mempelajari berbagai prosedur pengambilan keputusan, mencari prediktor atau prosedur pengambilan keputusan terbaik untuk berbagai situasi. Lebih jauh lagi ahli statistika harus dapat memberikan informasi berkaitan dengan derajat kecocokan dari masing masing prosedur yang diberikan.

Selanjutnya prosedur dan metode yang dihasilkan harus diuji validitasnya. Salah satu cara untuk menguji validitas suatu prosedur atau metode statistika adalah dengan mencobanya pada data yang parameternya diketahui (terkendali). Untuk itu perlu dikembangkan berbagai tehnik yang bertujuan membangkitkan data dengan sifat-sifat atau parameter yang diinginkan. Misalnya kita ingin membangkitkan data univariate dengan rataan dan ragam tertentu, data multivariate dengan rataan, ragam dan korelasi tertentu. Tehnik-tehnik ini dipelajari dalam Metode Simulasi.

Selain itu, kemajuan di bidang komputer menyebabkan komputer tidak saja berguna sebagai alat bantu untuk menghitung dan memverifikasi hasil analisa statistika. Akhir- akhir ini berkembang tehnik statistika yang disebut analisis berbasis komputer atau simulasi yang biasa disebut *computer intensive statistics* khususnya Monte Carlo dan *resampling method*. Jika secara tradisional interval kepercayaan suatu penduga ditetapkan berdasarkan asumsi distribusinya (Normal, t, χ^2 atau F), maka dengan metode *resampling* (sampling ulang), kita sepenuhnya menggunakan komputer untuk menghitung interval kepercayaan dari suatu estimator tanpa perlu mengasumsikan distribusi tertentu pada datanya.

Jelaslah bahwa kecanggihan teknologi komputer memberikan manfaat yang sangat besar untuk mensimulasikan suatu sistem, termasuk kelakuan dinamisnya. Simulasi dinamis dapat dikembangkan untuk mempelajari secara detail karakteristik dinamis (proses) dari suatu sistem (meskipun karakteristik tersebut sulit diperoleh melalui eksperimen), sekaligus untuk memprediksi apa yang akan terjadi pada sistem tersebut di masa mendatang.

1.1 Prinsip Simulasi

Untuk mendapatkan gambaran yang lebih baik tentang makna simulasi, berikut diberikan definisi beberapa ahli tentang simulasi.

Definisi 1.1 (Banks (Banks 1998))**.** Simulasi adalah tiruan dari proses dunia nyata atau sistem. Simulasi menyangkut pembangkitan proses serta pengamatan dari proses untuk menarik kesimpulan dari sistem yang diwakili.

Definisi 1.2 (Nailor (1966) dalam Rubinstein & Melamed (Melamed 1998))**.** Simulasi adalah tehnik numerik untuk melakukan eksperimen pada komputer, yang melibatkan jenis matematika dan model tertentu yang menjelaskan prilaku bisnis atau ekonomi pada suatu periode waktu tertentu.

Menurut Borowski & Borwein (Borwein 1989) simulasi didefinisikan sebagai berikut ini.

Definisi 1.3. Simulasi adalah tehnik untuk membuat konstruksi model matematika untuk suatu proses atau situasi, dalam rangka menduga secara karakteristik atau menyelesaikan masalah berkaitan dengannya dengan menggunakan model yang diajukan.

Jadi simulasi mempelajari atau memprediksi sesuatu yang belum terjadi dengan cara meniru atau membuat model sistem yang dipelajari dan selanjutnya mengadakan eksperimen secara numerik dengan menggunakan komputer. Dalam simulasi matematika atau statistika ada beberapa komponen yang mutlak diperlukan diantaranya adalah:*model* dari permasalahan yang dipelajari dan *komputer* yang dijadikan alat untuk melakukan eksperimen. Dalam persoalan model diperlukan kemampuan konseptual matematika dan statistika atau teori peluang, sedangkan dalam hal penggunaan komputer diperlukan kemampuan metode numerik ataupun pengetahuan komputasi lainnya, sehingga dihasilkan algoritma atau program komputer yang efisien.

Pada masa lalu prediksi masa depan hampir sepenuhnya hanya mengandalkan kajian deduktif (teori) seperti dinyatakan oleh Nailor (1966) dalam Rubinstein & Melamed (Melamed 1998) bahwa alasan mendasar menggunakan simulasi adalah kebutuhan manusia yang semakin mendesak akan masa depan. Pencarian pada pengetahuan tersebut dan keinginan agar mampu memprediksi masa depan sudah dimulai sejak adanya manusia. Tetapi sebelum abad ke 17 kemampuan memprediksi masa depan hanya terbatas seluruhnya pada metode deduktif yang dikembangkan oleh para pemikir seperti Plato, Aristotles, Euklid.

1.2 Manfaat dan Peran Simulasi

Simulasi dapat bermanfaat untuk mempelajari berbagai hal dalam berbagai bidang di antaranya seperti berikut ini.

Pilihan tepat. Simulasi memungkinkan kita menguji setiap aspek dari perubahan yang diinginkan tanpa menempatkan objek yang dipelajari pada posisi dimaksud. Simulasi dapat digunakan untuk menguji kekuatan rancang bangun gedung. Dalam hal pembangunan jembatan, misalnya jembatan dapat disimulasikan sebelum benar-benar memulai pembangunan riil di lapangan. Hasil simulasi memungkinan kita memilih ukuran yang tepat.

Pengaturan waktu. Dalam simulasi kita bisa mengatur waktu yaitu mempercepat atau memperlambat suatu proses yang memungkinkan kita mengamatinya secara keseluruhan.

Mencari penyebab. Melalui simulasi kita dapat melihat mengapa suatu fenomena muncul. Mengapa suatu proses tidak berjalan sebagaimana mestinya.

Eksplorasi kemungkinan. Dengan mengatur nilai-nilai dalam simulasi, kita dapat mempelajari atau mengeksplorasi kemungkinan-kemungkinan pengembangan tanpa banyak mengeluarkan biaya.

Pemahaman Sistem. Simulasi memungkinkan kita memiliki pemahaman yang lebih baik tentang hubungan antara variabel-variabel yang mempengaruhi suatu sistem yang kompleks. Simulasi tidak sekedar menduga bagaimana suatu sistem akan beroperasi, tetapi lebih menunjukan bagaimana suatu sistem beroperasi.

1.3 Langkah-langkah simulasi

Simulasi sebagai suatu cara menyelesaikan masalah, mempunyai tahapan-tahapan atau langkah-langkah penting yang harus dilalui diantaranya:

1. Formulasi masalah
2. Menyusun tujuan
3. Pembuatan model
4. Pengumpulan data
5. Pembuatan kode/skrip komputer
6. Verifikasi program komputer
7. Validasi model
8. Mendesain eksperimen

9. Analisis
10. Dokumentasi dan laporan
11. Implementasi dan Generalisasi

1.4 Aplikasi Simulasi

Aplikasi simulasi meliputi berbagai bidang diantaranya adalah bidang-bidang yang disebutkan berikut ini.

1. Bidang manufaktur
   (a) Menganalisis alur material dalam parbrik asembling mobil.
   (b) Mempelajari penanganan material (fabrikasi atau distribusi) dalam jumlah besar.
   (c) Menghitung biaya dari perbaikan kualitas
2. Bidang Kesehatan
   (a) Mempelajari organ dalam manusia
   (b) Evaluasi kebijakan di berbagai unit di rumah sakit
   (c) Analisis layanan *ambulance*
3. Bidang Meliter
   (a) Latihan perang dan penerbangan
   (b) Analisis pengangkutan peralatan

4. Layanan umum
   (a) Analisis layanan pemadam kebakaran
   (b) Analisis kemacetan jalan
5. Pendidikan
   (a) Simulasi dan animasi susunan atom kimia atau reaksi kimia
   (b) Simulasi dan animasi organ dalam binatang atu manusia
6. Bidang Statistika
   (a) Membangkitkan data dengan distribusi dan parameter yang diketahui
   (b) Memvisualisasikan sifat-sifat distribusi atau uji statistika sehingga menjadi lebih mudah difahami.
   (c) Uji statistika berbasis simulasi, jika dukungan teori tentang distribusi tidak cukup.

BAB 2

TEKNIK DASAR SIMULASI PEUBAH ACAK

Tujuan Umum

Memahami tehnik dasar pembangkitan bilangan acak serta menggunakannya dalam membangkitkan data peubah acak

Tujuan Khusus

Pada akhir pembahasan pada bab ini mahasiswa diharapkan:

1. dapat membangkitkan bilangan acak yang baik;
2. dapat menyebutkan tiga tehnik utama dalam membangkitkan data dari distribusi tertentu;
3. dapat mengilustrasikan secara grafis kecocokan antara data yang diinginkan dengan data yang dihasilkan.

Materi

1. Tehnik membangkitkan bilangan acak
2. Tehnik Membangkitkan Data dari Distribusi Tertentu
3. Pemertiksan Secara Emperik

2.1 Membangkitkan Bilangan Acak

Dalam simulasi stokastik, sangat dibutuhkan serangkaian peubah acak yang berdistribusi seragam (misalnya $U(0,1)$). Oleh karena itu membangkitkan bilangan acak seperti ini menjadi bagian yang sangat penting dalam simulasi stokastik. Saat ini dapat dipastikan bahwa semua paket pemrograman sudah dilengkapi (*built in*) perintah cara membangkitkan bilangan acak dengan distribusi seragam.

Salah satu cara yang dapat dipergunakan membangkitkan bilangan acak adalah dengan menggunakan prinsip bilangan kongruensi modulo. Tehnik ini dikenal dengan sebutan *Linier Congruential Generator (LCG).* Bentuk umumnya adalah

$$X < -i + 1 = aX < -i + c(\text{mod } m)$$

dengan X_0 sebagai *seed* dan $a, m \in Z^+,\ c$ adalah nol atau bilangan postif. Jika $c \neq 0$ disebut *mixed congruential generator* sedangkan jika $c = 0$ disebut *Multiplicative Linear Congruential Generator (MLCG).* Maka $X_i \in \{0, 1, 2, \cdots, m-1\}$. X disebut menyerupai bilangan acak (*pseudo random numbers*). Disebut "menyerupai" karena sesungguhnya barisan bilangan yang terjadi bersifat deterministik dengan pola yang berulang dengan panjang tertentu. Algoritma untuk menghasilkan bilangan bulat acak pada interval (0,m) dinyatakan pada Algoritma 2.1Membangkitkan Bilangan AcakItem.52

Algoritma 2.1 (Membangkitkan bilangan acak pada interval (0,m)). *Untuk membangkitkan n bilangan acak pada interval 0 dan m ditempuh langkah-langkah berikut*

1. *Tentukan seed* X_0
2. *Untuk* $i = 1, 2, \cdots, n$ *hitung* $X_i = aX_{i-1} + c \; (mod \; m)$

Selanjutnya algoritma di atas dapat diterjemahkan ke dalam salah satu bahasa pemrograman. Dalam diktat ini, untuk kode-kode pemrograman dipergunakan bahasa S-Plus atau R (Lihat Tirta (Tirta 2001)). Untuk membangkitkan bilangan acak dengan cara di atas dengan menggunakan S-Plus atau R maka dapat ditempuh dengan cara berikut

Program 2.1.

```
# Program Membangkitkan bilangan acak bulat antara (0,m)

m<-
x0<-
a<-
c<-
n<-
for(i in 1:n){
    y<-a*x0+c
    y1<-(y/m)-floor(y/m)
```

```
    x<-y1*m
    x0<-x
    print(x)
}
```

Hasil yang diperoleh untuk $m = 6; x_0 = 2; a = 5; c = 3$, hanya berupa barisan bilangan 1 dan 2 seperti dapat dilihat pada Tabel 2.1Membangkitkan Bilangan Acaktable.2.1

Tabel 2.1: Barisan Bilangan dari Cara Kongruen dengan $m = 6, a = 5, c = 3$.

i	X_i	aX_i	$aX_i + c$	$aX_i + c(\text{mod } m)$
0	2	10	13	1
1	1	5	8	2
2	2	10	13	1
3	1	5	8	2
...	...	...	...	...

Karena pada dasarnya bilangan acak yang diperoleh bukanlah bilangan acak yang sesungguhnya, maka supaya lebih menyerupai bilangan acak, Coddington (Coddington n.d.) mengatakan beberapa syarat penting yang harus dipenuhi oleh bilangan acak adalah seperti berikut ini.

Dapat diulang. Sekumpulan (barisan) bilangan yang sama harus bisa diperoleh (diulang) dengan menggunakan *seed* yang sama, hal ini kadang-kadang diperlukan untuk pemeriksaan dan penelusuran program (*debugging.*)

Keacakan. Barisan bilangan harus memenuhi syarat keacakan secara seragam (*uniform*) yang dapat diuji melalui uji statistika.

Periode panjang. Karena pada dasarnya bilangan acak itu merupakan barisan berulang dengan berbagai periode, maka periode pengulangan harus sangat besar atau lama melebihi banyaknya bilangan acak yang diperlukan.

Tidak peka *seed*. Sekalipun barisan bilangannya bergantung pada *seed* tetapi sifat keacakan dan periodisasi sedapat mungkin tidak bergantung pada seed-nya.

Agar barisan bilangan yang diperoleh lebih memenuhi sebagai bilangan acak seperti disyaratkan di atas, maka perlu diperhatikan hal-hal berikut ini (Rubinstein & Melamed (Melamed 1998)).

1. Jika diinginkan sekelompok bilangan yang berbeda pada setiap permintaan, maka harus digunakan *seed* yang berbed-beda yang dapat ditentukan secara acak. Untuk ini dapat digunakan bagian dari waktu (menit atau detik) pada saat program itu dijalankan sehingga setiap saat diperoleh *seed* yang berbeda-beda.

2. Bilangan c merupakan prima relatif terhadap m, artinya m dan c tidak mempunyai pembagi bersama kecuali 1.

3. Bilangan a memenuhi $a \equiv 1(\text{mod } g)$ dengan g adalah faktor dari m.

4. Bilangan a memenuhi $a \equiv 1(\text{mod } 4)$ jika m adalah kelipatan 4.

Dengan mengubah nilai pada Program 2.1. menjadi $m = 7,\ x_0 = 2,\ a = 1$ dan $c = 5$, maka diperoleh hasil yang lebih baik (pengulangan dengan panjang 7 bilangan) seperti pada Tabel 2.2Membangkitkan Bilangan Acaktable.2.2. Namun, jika sampel yang diminta 10 atau lebih tentu saja sekelompok bilangan tersebut masih belum bisa dianggap acak karena pengulangannya lebih pendek dari pada banyaknya bilangan yang diminta. Hasil yang lebih baik diperoleh dengan mengambil m yang lebih besar tetapi tetap merupakan bilangan prima (misalnya 11, 17, 31 dan seterusnya).

Tabel 2.2: Barisan Bilangan dari Cara Kongruen dengan $m = 7, a = 1, c = 5$

i	X_i	aX_i	$aX_i + c$	$aX_i + c(\text{mod } m)$
0	2	2	7	0
1	0	0	5	5
2	5	5	10	3
3	3	3	8	1
4	1	1	6	6
5	6	6	11	4
6	4	4	9	2
7	2	2	7	0
8	0	0	5	5
9	5	5	10	3
10	3	3	8	1
11	1	1	6	6
12	6	6	11	4
13	4	4	9	2
14	2	2	7	0
15	0	0	5	5
...	...	...	...	...

Untuk membangkitkan bilangan acak kontinu pada interval (0,1) pada dasarnya tinggal membagi bilangan tersebut dengan bilangan modulernya. Ini dapat dilakukan dengan Algoritma 2.1Membangkitkan Bilangan AcakItem.59.

Algoritma 2.2 (Membangkitkan bilangan acak pada interval (0,1)). *Untuk membangkitkan bilangan n acak pada interval 0 dan 1 ditempuh langkah-langkahberikut*

1. *Tentukan seed* X_0
2. *Untuk* $i = 1, 2, \cdots, n$ *hitung* $X_i = aX_{i-1} + c\ (mod\ m)$
3. *Selanjutnya tentukan* $U_i = \frac{X_i}{m}$

Keacakan U akan semakin baik jika kita bisa mengambil m bilangan prima sebesar mungkin. Berikut adalah 30 bilangan random kontinu antara (0,1) yang diperoleh dengan $m = 97$, $a = 98$, $x_0 = 2$, $c = 5$. Dari hasil yang diperoleh terlihat bahwa dari ketiga puluh bilangan tersebut tidak ada yang merupakan pengulangan dari bilangan sebelumnya. Ini berarti periode pengulangan sekelompok bilangan yang dihasilkan lebih panjang dari 30.

```
0.0721649484536084 0.123711340206185  0.175257731958762
0.226804123711339  0.278350515463917  0.329896907216494
0.381443298969074  0.432989690721648  0.484536082474229
0.536082474226802  0.587628865979383  0.639175257731956
0.690721649484537  0.742268041237111  0.793814432989692
0.845360824742272  0.896907216494839  0.948453608247419
```

```
0.000000000000000  0.0515463917525773 0.103092783505154
0.154639175257731  0.206185567010309  0.257731958762886
0.309278350515463  0.36082474226804   0.412371134020617
0.463917525773198  0.515463917525771  0.567010309278352
```

Coddinngton (Coddington n.d.) memberi daftar pasangan pilihan a, c dan m yang telah dicoba untuk berbagai jenis mesin, sebagaimana ditulis pada Tabel 2.3Membangkitkan Bilangan Acaktable.2.3.

Tabel 2.3: Daftar berbagai nilai parameter a, c dan m yang telah dicobakan pada beberapa mesin

32-bit		
A=69069	C=0 atau 1	M= 2^{32}(VAX)
A=1664525	C=0	M= 2^{32}(transputers)
A=16807	C=0	M= $2^{31}-1$ (IBM)
A=1103515245	C=12345	M= 2^{31} (UNIX rand routine)
48-bit		
A=$5DEECE66D^{16}$,	C=B^16,	M= 2^{48} (UNIX drand 48 routine)
A=5^{15}	C=0,	M= 2^{47} (CDC vector machines)
A= $2875A2E7B175$,	C=0,	M= 2^{48} (Cray vector machines)
64-bit		
A=13^{13},	C=0,	M= 2^{59} (Numerical Algorithms Group)

2.2 Membangkitkan Data dari Distribusi Tertentu

Pada bagian sebelumnya kita telah membahas cara membangkitkan data dari distribusi seragam $U(0,1)$. Untuk membangkitkan data dari distribusi lain, misalnya dengan fungsi kepadatan $f(x)$, maka ada beberapa kondisi yang mungkin kita hadapi, yaitu seperti berikut ini.

1. Ada transformasi langsung dari $U(0,1)$ ke X yang memiliki fungsi kepadatan $f(x)$. Untuk kondisi ini maka kita tinggal mencari fungsi $T(u)$ yang mentransformasikan $U(u) = 1,\ 0 < u < 1$ ke X dengan $f(x),\ x \in R_X$.

2. Tidak ada transformasi langsung yang menghubungkan $U(0,1)$ ke X yang memiliki fungsi kepadatan $f(x)$ tetapi invers fungsi kumulatifnya $F^{-1}(x)$ dapat ditentukan. Untuk kondisi ini kita dapat menggunakan tenik yang disebut metode invers transform.

3. Tidak ada transformasi langsung yang menghubungkan antara $U(0,1)$ dengan $f(x)$ dan invers fungsi kumulatifnya $F^{-1}(x)$ tidak dapat ditentukan. Untuk kondisi ini kita dapat membangkitkan X dengan menggunakan prinsip Monte Carlo.

2.3 Pemeriksaan Secara Emperik

Untuk meyakinkan kita bahwa data yang kita peroleh memenui sifat yang diharapkan, yaitu bahwa X benar-benar berasal dari distribusi yang diharapkan, maka perlu dilakukan pemeriksaan sifat-sifat data dengan cara antara lain sebagai berikut ini.

1. Menghitung rataan dan ragam data dan dibandingkan dengan rataan dan ragam teoritiknya. Pemeriksaan ini dapat dilakukan untuk berbagai ukuran data.

2. Menggambar sebaran (densitas) emperiknya dengan menggunakan `plot(density(x))` yang selanjutnya dibandingkan dengan $f(x)$ untuk berbagai ukuran sampel.

Sebagai contoh diambil secara acak data dari distribusi $N(10, 5)$. Pada Gambar 2.1Pemeriksaan Secara Emperikfigure.2.1 dapat dilihat bahwa semakin besar ukuran sampel, rata-rata sampel semakin dekat dengan nilai rataan $\mu = 10$. Demikian juga semakin besar ukuran sampel ragam sampel semakin dekat dengan ragam teoritik $\sigma^2 = 5$. Selain itu dapat dilihat bahwa rata-rata lebih cepat konvergen dibandingkan dengan ragam. Untuk menghasilkan grafik rataan dan ragam seperti pada Gambar 2.1Pemeriksaan Secara Emperikfigure.2.1, dapat dibuat program seperti berikut.

Program 2.2.

```
m<-6
ydat<-matrix(0,m,5)
ydat[,2]<-10
ydat[,3]<-5
for(i in 1:m){
    n<-10*(i+4)
    ydat[i,1]<-n
    y<-rnorm(n,10,sqrt(5))
    ydat[i,4]<-mean(y)
```

Gambar 2.1: Rataan dan Ragam sampel untuk Berbagai Ukuran Sampel

```
    ydat[i,5]<-var(y)
}
plot(ydat[,1],ydat[,4],type='b',xlab='n',ylab='ragam dan
mean',ylim=c(3,12))
lines(ydat[,1],ydat[,2])
points(ydat[,1],ydat[,5])
lines(ydat[,1],ydat[,5])
lines(ydat[,1],ydat[,3])
```

Dari data yang sama selain bisa dibuat grafik momennya (rata-rata dan ragam), juga dapat dibuat grafik sebarannya. Sebaran data yang berupa titik dapat dibandingkan dengan grafik fungsi kepadatan, dalam hal ini dari distribusi $N(10, 5)$. Untuk menghasilkan grafik densitas seperti pada Gambar 2.2Pemeriksaan Secara Emperikfigure.2.2, dapat dibuat program seperti berikut

Program 2.3.

```
m<-6
par(mfrow=c(2,3),mar=rep(3,4))
ydat<-matrix(0,m,5)

for(i in 1:m){
    n<-10*(i+4)
    ydat[i,1]<-n
    y<-rnorm(n,10,sqrt(5))
    ydat[i,4]<-mean(y)
    ydat[i,5]<-var(y)
    xd<-seq(min(y),max(y),0.1)
    yd<-dnorm(xd,10,sqrt(5))
    plot(density(y))
    lines(xd,yd) }
```

2.4 Daftar Bacaan

Untuk memahami lebih jauh tentang pembangkitan bilangan acak, dapat dilihat pada Rubinstein & Melamed (Melamed 1998, Bab 2) dan Coddington (Coddington

Gambar 2.2: Grafik Densitas Data dibandingkan dengan Densitas Normal Standar

n.d., Bab 9-10)

2.5 Soal-soal Latihan

1. Bangkitkan 20 bilangan bulat acak dengan distribusi seragam (persegi panjang), antara 0 dan 10. Selanjutnya selidiki:

   (a) dengan berbagai pasangan a, c dan m selidiki apakah bilangan yang anda peroleh ada yang berulang?

   (b) berapa panjang barisan yang berulang?

2. Bangkitkan 10 bilangan acak real dengan distribusi seragam antara 0 dan 1. Selanjutnya selidiki:

   (a) dengan berbagai pasangan a, c dan m selidiki apakah bilangan yang anda peroleh ada yang berulang?

   (b) berapa panjang barisan yang berulang?

3. Bangkitkan 15 bilangan acak real X yang berdistribusi seragam antara 0 dan 1.

   (a) tentukan rataan dan ragam dari X

   (b) buat grafik densitas dari X

   Misalkan $Y = 1 - X$, maka

   (a) tentukan rataan dan ragam dari Y

   (b) buat grafik densitas dari Y

   Dari penampilan grafik tersebut apakah menurut anda Y juga berdistribusi seragam $U(0, 1)$. (Bukti formal dapat diturunkan secara matematis.

BAB 3

MEMBANGKITKAN DATA DENGAN TRANSFORMASI LANGSUNG

Tujuan Umum

Memahami dan mengingat kembali jenis-jenis transformasi peubah acak serta dapat menggunakannya dalam simulasi, khususnya dalam membangkitkan sampel dari peubah acak dengan distribusi tertentu.

Tujuan Khusus

Pada akhir pembahasan bab ini mahasiswa diharapkan dapat:

1. menyebutkan berbagai macam transformasi peubah acak, yang memetakan suatu peubah ke peubah lain yang banyak dikenal;
2. dapat membangkitkan data dari $N(0,1)$;

3. dapat membangkitkan data dari $N(\mu, \sigma^2)$;
4. dapat membangkitkan data dari $G(\alpha, \beta)$;
5. dapat membangkitkan data dari t_ν;
6. dapat membangkitkan data dari F_{ν_1,ν_2}
7. dapat membangkitkan data dari Normal bivariat atau normal multi variat

Materi

1. Transformasi peubah acak;
2. Membangkitkan data dari $N(0, 1)$;
3. Membangkitkan data dari $N(\mu, \sigma^2)$;
4. Membangkitkan data dari $G(\alpha, \beta)$;
5. Membangkitkan data dari t_ν;
6. Membangkitkan data dari F_{ν_1,ν_2}
7. Membangkitkan data dari Normal bivariat atau normal multi variat

3.1 Transformasi Peubah Acak

Simulasi komputer sering digunakan untuk memeriksa tehnik statistik yang diajukan. Simulasi mensyaratkan bahwa kita memperoleh nilai pengamatan dari suatu peubah acak dengan distriusi dan parameternya yang telah ditentukan. Kebanyakan sistem komputer memuat subrutin yang menyediakan nilai pengamatan dari suatu peubah acak Y yang memiliki distribusi uniform pada selang [0,1]. Ini berarti dari distribusi uniform ini kita harus dapat memanfaatkannya untuk mensimulasikan data dari suatu distribusi yang kita inginkan. Prinsip transformasi dapat digunakan untuk membangkitkan sejumlah pengamatan distribusi lain, misalnya distribusi normal, eksponensial dan lain-lain. Berikut diberikan rangkuman beberapa transformasi yang bermanfaat dalam mensimulasikan pengamatan atau data dari suatu distribusi. Transformasi-transformasi ini dibahas dan dibuktikan dalam Statistika Matematika, sehingga di sini hanya dikutip hasilnya sebagai prosedur untuk membangkitkan data acak dari suatu distribusi tertentu. Transformasi yang ada dapat dibedakan menjadi dua kelompok besar.

1. Transformasi T dari $U(0,1)$ ke suatu peubah acak X. Termasuk dalam kelompok ini adalah transformasi Box-Muller yang memetakan $U(0,1)$ ke $N(0,1)$.

2. Transformasi T_1 dari peubah acak X yang memiliki fungsi kepadatan $f(x)$ ke peubah acak Y dengan fungsi peluang $g(y)$. Termasuk dalam transformai ini adalah transformasi linier dari $N(0,1)$ ke $N(\mu,\sigma^2)$, transformasi dari $N(0,1)$ ke χ_1^2 dan lain-lain.

Semua transformasi di atas pada dasarnya telah dibahas pada matakuliah Statistika Matematika.

3.2 Membangkitkan Data dari $N(0,1)$ dengan Transformasi Box-Muller

Transformasi dari distribusi uniform ke distribusi normal standar dapat dilakukan dengan transformasi Box-Muller.

Hasil 3.1 (Transformasi Box-Muller)**.** *Jika $U_1||U_2$ masing masing dari $U(0,1)$, maka*

$$Z_1 = \sqrt{(-2\ln U_1)}\cos(2\pi U_2), \ \ dan$$
$$Z_2 = \sqrt{(-2\ln U_1)}\sin(2\pi U_2)$$

saling bebas dan masing- masing dengan distribusi $N(0,1)$.

Dari hasil di atas selanjutnya dapat dibuat algoritma untuk membangkitkan data normal setandar dengan menggunakan transformasi Box-Muller seperti berikut ini.

Algoritma 3.1. *Langkah langkah untuk membangkitkan data dari $N(0,1)$ dengan transformasi Box-Muller adalah*

1. *Bangkitkan U_1 dan U_2 dari $U(0,1)$*

2. *Tentukan*

   (a) $Z_1 = \sqrt{(-2\ln U_1)}\cos(2\pi U_2)$, *dan* *(b)* $Z_2 = \sqrt{(-2 \ln U_1)} \sin(2\pi U_2)$

maka Z_1, Z_2 *adalah data acak dari* $N(0,1)$

Program inti untuk membangkitkan data normal standar dari distribusi uniform $U(0,1)$ diberikan pada Program 3.2Data dengan Transformasi Box-MullerItem.95. Pemeriksaan grafik terhadap data yang dihasilkan diberikan dengan menggambar grafik densitasnya, pada Gambar 3.1Data dengan Transformasi Box-Mullerfigure.3.1 dan rataan sampel pada Gambar 3.2Data dengan Transformasi Box-Mullerfigure.3.2 untuk berbagai ukuran sampel. Sebagai pembanding pada Gambar 3.1Data dengan Transformasi Box-Mullerfigure.3.1 dan Gambar 3.2Data dengan Transformasi Box-Mullerfigure.3.2 dibuat juga grafik densitas dan rataan dari data yang dibangkitkan dengan fungsi internal S-Plus atau R. Dari kedua grafik tersebut dapat dilihat bahwa program yang ditulis menghasilkan data sebaik fungsi internal S-Plus atau R.

Program 3.1.

```
#Program inti untuk membangkitkan 2*n data normal standar
u1<-runif(n,0,1)
u2<-runif(n,0,1)
z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)
z<-c(z1,z2)
plot(density(z),xlab='x',ylab='p')
x<-seq(min(z),max(z),0.1)
y<-dnorm(x,0,1)
```

Gambar 3.1: Grafik Densitas dari Berbagai Data Bangkitan Berdistribusi Normal

```
lines(x,y)
```

3.3 Membangkitkan Data $N(\mu, \sigma^2)$

Selanjutnya dari normal standar ke normal yang lebih umum dapat dilakukan dengan menggunakan transformasi linier yang sudah dikenal dengan baik.

Hasil 3.2. *Jika $Z \sim N(0, 1)$, maka $Y = \mu + \sigma Z$ berdistribusi $N(\mu, \sigma^2)$.*

Gambar 3.2: Rataan Data Berdistribusi Normal denganBerbagai Ukuran Sampel

Algoritma 3.2. *Langkah untuk membangkitkan data dari $N(\mu, \sigma^2)$ adalah:*

1. *bangkitkan Z dari $N(0, 1)$ seperti pada algoritma sebelumnya;*

2. *tentukan $X = \sigma Z + \mu$.*

maka X adalah data acak dari $N(\mu, \sigma^2)$

Program 3.2.

```
#Program inti membangkitkan 2*n data normal
u1<-runif(n,0,1)
u2<-runif(n,0,1)
z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)
z<-c(z1,z2)
x<-sgm*z+mu
```

3.4 Membangkitkan Data Keluarga $G(\alpha, \beta)$

Salah satu keluarga distribusi gamma yang dapat dibangkitkan dari distribusi normal adalah χ^2. Distribusi normal standar dapat ditransformasikan menjadi distribusi χ^2 dengan transformasi kuadrat.

Hasil 3.3. *jika $Z \sim N(0, 1)$, maka $Y = Z^2$ berdistribusi χ^2_1.*

Selanjutnya jumlah beberapa χ^2 yang saling independen akan menghasilkan χ^2 dengan derajat kebebasan yang merupakan jumlah dari derajat kebebasan masing-masing.

Algoritma 3.3. *Untuk membangkitkan data dari χ_1^2 dapat dilakukan langkah-langkah berikut:*

1. *bangkitkan Z dari $N(0,1)$ seperti pada algoritma sebelumnya;*
2. *tentukan $Y = Z^2$.*

maka Y adalah data dari χ_1^2

Program 3.3.

```
#Program inti membangkitkan n data chi-kuadrat (1)
u1<-runif(n,0,1)
u2<-runif(n,0,1)
z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)
z<-c(z1,z2)
y<-z^2
```

Hasil 3.4. *Jika $X_i \sim \chi_1^2$, dan saling bebas satu sama lain, maka $Y = \sum_{i=1}^{r} X_i \sim \chi_r^2$.*

Program 3.4.

```
#Program inti membangkitkan m data chi-kuadrat (r=2n)
for(i in 1:m){
    u1<-runif(n,0,1)
    u2<-runif(n,0,1)
    z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
    z2<-sqrt(-2*log(u1))*sin(2*pi*u2)
    z<-c(z1,z2)
    x<-z^2
    y<-sum(x)
}
```

Hasil 3.5. *Jika* $X \sim U(0,1)$, *maka* $Y = \dfrac{1}{\lambda} \ln X \sim \exp(\lambda)$.

Program 3.5.

```
#Program inti membangkitkan n data eksponensial
x<-runif(n,0,1)
y<-1/lbd*log(x)
```

Hasil 3.6. *Jika* $U_1, U_2, \cdots, U_m$ *berdistribusi i.i.d* $U(0,1)$, *maka* $\dfrac{1}{\beta}\sum_{i=1}^{m} \ln U_i$ *berdistribusi Gamma* (m, β).

3.5 Membangkitkan Data Distribusi t dan F

Distribusi t dan F dapat dibangkitkan dari transformasi distribusi norma dan chi-kuadrat seperti ditunjukkan pada hasil berikut.

Hasil 3.7. *Jika $Z \sim N(0,1)$ dan $X \sim \chi^2_\nu$, maka*

$$Y = \frac{Z}{\sqrt{\frac{\chi^2_\nu}{\nu}}}$$

berdistribusi t_ν.

Hasil 3.8. *Jika $X_1 \sim \chi^2_{\nu_1}$ dan $X_2 \sim \chi^2_{\nu_2}$, maka*

$$Y = \frac{X_1/\nu_1}{X_2/\nu_2}$$

berdistribusi F_{ν_1,ν_2}.

Algoritma untuk membangkitkan data dari distribusi t dann F adalah seperti berikut ini.

Algoritma 3.4. *Membangkitkan data dari distribusi t_ν adalah*

1. *bangkitkan Z dari $N(0,1)$*
2. *bangkitkan X dari χ^2_ν*

3. *transformasikan*

$$Y = \frac{Z}{\sqrt{\frac{\chi^2_\nu}{\nu}}}$$

maka $Y \sim t_\nu$.

Algoritma 3.5. *Membangkitkan data dari distribusi* t_ν *adalah*

1. *bangkitkan* X_1 *dari* $\chi^2_{\nu_1}$

2. *bangkitkan* X_2 *dari* $\chi^2_{\nu_2}$

3. *transformasikan*

$$Y = \frac{X_1/\nu_1}{X_2/\nu_2}$$

maka $Y \sim F_{\nu_1,\nu_2}$.

3.6 Membangkitkan Data Distribusi Normal Bivariat dan Multivariat

Distribusi bivariat normal dapat dibangkitkan dari distribusi uniform yang saling independen melalui transformasi seperti pada Hasil 3.6Data dengan Distribusi Normalsection.3.6. Grafik sebaran dari data X, Y yang berdistribusi $BVN(30, 40, 16, 25, 0.5)$ diambil untuk ukuran sampel sebanyak 40 pasang adalah seperti pada Gambar 3.3Data dengan Distribusi Normalfigure.3.3

Hasil 3.9. *Jika* X_1, X_2 *iid* $N(0,1)$, *maka*

$$Y_1 = \mu_1 + \sigma_1 X_1$$
$$Y_2 = \mu_2 + \rho\sigma_2 X_1 + \sigma_2\sqrt{1-\rho^2}X_2$$

berdistribusi $BVN(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$.

Algoritma 3.6. *Membangkitkan data dari* $BVN(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$

1. *bangkitkan* u_1 *dan* u_2 *dari* $N(0,1)$
2. *lakukan transformasi*

$$x = \mu_1 + \sigma_1 u_1$$
$$y = \mu_2 + \rho\sigma_2 u_1 + \sigma_2\sqrt{1-\rho^2}u_2$$

maka (x, y) *adalah data dari* $BVN(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$.

Implementasi dengan S-Plus atau R dapat dilihat pada progam berikut.

Program 3.6.

```
n<-100

mux<-30

muy<-40

sdx<-4

sdy<-5
```

```
r<-0.5
dmat<-matrix(0,n,2)

for(i in 1:n){
    u1<-runif(1,0,1)
    u2<-runif(1,0,1)
    dmat[i,1]<-mux+sdx*u1
   dmat[i,2]<-muy+r*sdy*u1+sdy*sqrt(1-r^2)*u2
  }
x<-dmat[,1]
y<-dmat[,2]
plot(x,y,type='p')
```

Untuk distribusi normal multivariat kita meemiliki hasil berikut

---

Hasil 3.10. *Jika* $\mathbf{X}$ *p sampel yasng berdistribusi identik dan independen* $N(0,1)$, *maka* $\mathbf{Y} = \boldsymbol{\mu} + AX$ *berdistribusi multivariat normal dengan rataan* $\boldsymbol{\mu}$ *dan ragam -koragam* $\boldsymbol{\Sigma} = \mathbf{AA^T}$ *dan fungsi kepadatan*

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \ \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

Sebaliknya untuk menghasilkan data dari $\mathbf{Y}$ yang berdistribusi multivariat dengan rataan $\boldsymbol{\mu}$ dan ragam-koragam Σ, maka kita harus menemukan matriks $\mathbf{A}$ sedemikian sehingga $\Sigma = \mathbf{AA}^T$.Dapat ditunjukkan bahwa (Lihat Rubinstein &

Gambar 3.3: Grafik Sebaran Data dengan Distribusi Bivariat Normal

Melamed (Melamed 1998, hal 30-32))

$$a_{ij} = \frac{\sigma_{ij} - \sum_{k=1}^{j-1} a_{ik} a_{jk}}{\left(\sigma_{jj} - \sum_{k=1}^{j-1} a_{jk}^2 \right)^{1/2}} \tag{3.1}$$

Selanjutnya kita peroleh $\mathbf{Y} = \mathbf{AX} + \boldsymbol{\mu}$.

3.7 Daftar Bacaan

Sebagian besar materi pada bab ini dibahas pada Tirta (Tirta 2003)[Bab 8](Materi kuliah Statistika Matematika I dan II). Sebagian algorithma untuk membangkitkan data peubah acak dapat dilihat pada Rubinstein & Melamed (Melamed 1998).

3.8 Latihan Soal-soal

1. Buatlah program lengkap dengan S-Plus atau R untuk membangkitkan data dari $N(50,4)$ Selanjutnya periksa data tersebut dengan membuat grafik densitasnya.

2. Buatlah program lengkap dengan S-Plus atau R untuk membangkitkan data dari $t(10)$ Selanjutnya periksa data tersebut dengan membuat grafik densitasnya.

3. Buatlah program lengkap dengan S-Plus atau R untuk membangkitkan data dari $F(5,4)$ Selanjutnya periksa data tersebut dengan membuat grafik densitasnya.

4. Buatlah program lengkap dengan S-Plus atau R untuk membangkitkan data dari $N(10,15,9,16,r)$ untuk berbagai nilai $0,3 \leq r \leq 0,9$. Selanjutnya periksa data tersebut dengan membuat diagram pencarnya.

5. Berdasarkan Hasil 3.6Data dengan Distribusi Normalfigure.3.3

   (a) tunjukkan bahwa persamaan $(3.1$Data dengan Distribusi Normal$\text{equation}.3.1)$ adalah benar;

(b) buat algoritma untuk membangkitkan data dari $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$;

(c) buat program S-Plus atau R untuk membangkitkan data di atas.

6. Buatlah program lengkap dengan S-Plus atau R untuk membangkitkan 50 data dari $\mathbf{Y}$ dari distribusi $MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ dengan

$$\boldsymbol{\mu} = \begin{pmatrix} 10 \\ 15 \\ 20 \end{pmatrix} \text{ dan } \Sigma = \begin{pmatrix} 9 & 5 & 4 \\ 5 & 16 & 8 \\ 4 & 8 & 25 \end{pmatrix}$$

BAB 4

MEMBANGKITKAN DATA DENGAN TRAFORMASI TAK LANGSUNG

Tujuan Umum

Dapat membangkitkan data peubah acak dengan tehnik transformasi invers atau dengan tehnik Monte Carlo

Tujuan Khusus

Setelah selesai membahas materi pada bab ini diharapkan agar mahasiswa dapat:

1. menjelaskan tehnik transformasi invers
2. menerapkan tehnik transformasi invers untuk membangkitkan beberapa jenis distribusi yang tidak bisa dibangkitkan dengan cara transformasi langsung

3. menjelaskan tehnik Monte Carlo
4. menggunakan tehnik Monte Carlo untuk membangkitkan data dari distribusi yang tidak bisa dibangkitkan secara langsung maupun dengan tehnik transformasi invers
5. menghitung pendekatan nilai π dengan tehnik Monte Carlo
6. dapat menggunakan Monte Carlo untuk menghitung integral tertentu

Materi

1. Metode Transformasi invers
2. Tehnik Monte Carlo
3. Menghitung pendekatan nilai π

4.1 Metode Transformasi Invers

Perhatikan bahwa fungsi distribusi komulatif (f.d.k.), $F(.)$, memetakan nilai peubah acak x ke interval $[0, 1]$ secara seragam, karena fungsi ini merupakan fungsi bijektif (Lihat Gambar 4.1Metode Transformasi Inversfigure.4.1). Oleh karena itu manakala cara membangkitkan data dari distribusi seragam $U(0, 1)$ telah diturunkan atau diketahui, maka pembangkitan data dari peubah acak yang diketahui fungsi kepadatan peluangnya (f.k.p.) dapat dilakukan dengan menggunakan invers dari fungsi distribusi komulatifnya.

Gambar 4.1: Grafik Fungsi Kepadatan dan Fungsi Kumulatif

Hasil 4.1. *Jika X mempunyai f.k.p. $f(x)$ dan f.d.k. $F(x)$, maka ada korespondensi satu- satu antara $F(x)$ dengan $(0,1)$. Dengan kata lain $F(X) \sim U(0,1)$*

Apabila suatu distribusi dapat ditentukan invers dari fungsi kumulatifnya, maka kita dapat mentransformasikan $U(0,1)$ ke X dengan fungsi kumulatif $F(x)$.

Hasil 4.2. *jika $X \sim U(0,1)$, maka $Y = F^{-1}(x)$, dengan $F()$, adalah fungsi kumulatif, berdistribusi dengan fungsi kepadatan peluang $f(x)$.*

Hasil 4.3. *Jika X mempunyai f.k.p. $f(x)$ dan f.d.k. $F(x)$, maka ada korespondensi satu- satu antara $F(x)$ dengan $(0,1)$. Dengan kata lain $F(X) \sim U(0,1)$*

Akibat 4.1. *jika $X \sim U(0,1)$, maka $Y = F^{-1}(X)$ berdistribusi dengan f.k.p. $f(x)$.*

Algoritma 4.1 (Metode Inverse-Transform)**.** *Aturan untuk membangkitkan data dari X dengan f.k.p. $f(x)$ adalah*

1. *turunkan fungsi kumulatif $F(x)$;*
2. *turunkan invers fungsi komulatif $F^{-1}(x)$;*
3. *ambil data $u \in U(0,1)$;*
4. *buat transformasi $x = F^{-1}(u)$; maka x berasal dari f.k.p. $f(x)$.*

Contoh 4.1.

Misalkan X adalah suatu peubah acak seragam pada selang [0,1], yaitu $X \sim U(0,1)$. Tetukan transformasi $Y = \Phi(X)$ sedemikian sehingga $Y = \Phi(X)$ memiliki suatu distribusi eksponensial dengan rataan β.

Jawab:

Jika X memiliki distribusi seragam pada selang [0,1], maka fungsi distribusi dari X adalah

$$F_X(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1. \end{cases}$$

Sementara itu jika Y berdistribusi eksponensial dengan rataan β, maka

$$f(y) = \frac{1}{\beta}\exp(-y/\beta)$$

$$F_Y(y) = \begin{cases} 0, & y < 0 \\ 1 - e^{-y/\beta}, & y \geq 0. \end{cases}$$

Perlu dicatat bahwa $F_Y(y)$ adalah monoton naik pada selang $[0, \infty]$, yang dipetakan satu-satu ke $0 < x < 1$. Untuk sembarang x sedemikian sehingga $0 < x < 1$, terdapat satu nilai y sedemikian hingga $F_Y(y) = x$. Karenanya $F_Y^{-1}(x) = y,\ 0 < x < 1$ terdefinisikan secara wajar. Dalam kasus ini $F_Y(y) = 1 - e^{-y/\beta} = y$ jika dan hanya jika $x = -\beta \ln(1-y) = F_X^{-1}(Y)$. Perhatikan bahwa peubah acak $F_X^{-1}(Y) = -\beta\ln(1-Y)$, dan amati bahwa jika $x > 0$,

$$\begin{aligned} P(F_X^{-1}(Y) \leq x) &= P(-\beta\ln(1-Y) \leq x) \\ &= P(\ln(1-Y) \geq -x/\beta) \\ &= 1 - e^{-x/\beta}. \end{aligned}$$

Karenanya $\Phi(YX) = F^{-1}(X) = -\beta\ln(1-X)$ memiliki distribusi eksponensial dengan rataan β sebagaimana diharapkan. Prinsip ini diaplikasikan dalam metode simulasi untuk membangkitkan data dari suatu distribusi dengan mentransformasikan data dari peubah acak seragam $U(0,1)$. Sebagai ilustrasi lihat Gambar 4.2Metode Transformasi Inversfigure.4.2.

Skrip S-Plus atau R untuk membangkitkan data dari distribusi eksponensial dengan metode invers transform adalah

```
n<-1000
bet<-5
x<-runif(n,0,1)
y<--bet*log(1-x)
```

Untuk membandingkan hasil dengan prosedur yang telah ada pada S-Plus atau R, maka kita dapat melengkapi skrip dengan perintah S-Plus atau R. Selanjutnya kita dapat menghitung rataan dan ragam dari data yang dibangkitkan dengan masing-masing prosedur. Namun perlu dicatat S-Plus atau R menggunakan notasi yang terbalik untuk parameter β pada distribusi Gamma.

Program 4.1.

```
n<-1000
bet<-5
x<-runif(n,0,1)
y<--bet*log(1-u)
teta<-1/bet
y1<-rexp(n,teta)
   mean(y)
   mean(y1)
   var(y)
```

Gambar 4.2: Grafik Densitas dan Kumulatif Distribusi Eksponensial

```
var(y1)
```

Contoh 4.2.

Misalkan X adalah peubah acak dengan fungsi kepadatan

$$f(x) = \begin{cases} 2/25x & \text{untuk } 0 < x < 5, \\ 0 & \text{untuk yang lain} \end{cases}$$

Contoh 4.3.

Jika m adalah bilangan bulat positif. Tentukan prosedur membangkitkan data dari peubah acak $Y \sim G(m, \beta)$.

Jawab:

Distribusi eksponensial pada dasarnya adalah distribusi Gamma dengan parameter *shape* $\alpha = 1$. Pada distribusi Gamma berlaku sifat reproduktif, yaitu jumlah distribusi Gamma yang saling bebas dengan skala yang sama menghasilkan distribusi Gamma dengan skala yang sama dan parameter bentuk yang merupakan jumlah dari parameter bentuk masing-masing, yaitu seperti Hasil 4.1Metode Transformasi Inversch.4.3

Hasil 4.4 (Sifat reproduktif distribusi gamma). *Jika* X_i, $i = 1, 2, \cdots, n$ *masing- masing berdistribusi Gamma saling bebas dengan parameter* (α_i, β) *maka* $Y = \sum X_i$ *berdistribusi Gamma dengan parameter* $(\sum \alpha_i, \beta)$.

Jadi jika kita bisa membangkitkan $X_i \sim Exp(1/\beta) = G(1, \beta)$ maka kita dapat membangkitkan $Y = \sum_{i=1}^{m} X_i \ \sim G(m, \beta)$.

Algoritma 4.2. *Prosedur membangkitkan data dari peubah acak* Y *dengan distribusi* $G(m, \beta)$ *adalah::*

1. *bangkitkan* X_i; $i = 1, 2, 3, \cdots, m$ *dari* $U(0, 1)$;
2. *tentukan* $Y = -\frac{1}{\beta} \sum_{i=1}^{m} \log(X_i)$ *atau* $Y = -\frac{1}{\beta} \log(\prod_{i=1}^{m} X_i)$.

4.2 Metode Monte Carlo

Metode Monte Carlo memberikan solusi pendekatan untuk berbagai masalah matematika dengan melakukan 'eksperimen' sampling statistik pada komputer. Walaupun pendekatannya stokastik, metode Monte Carlo dapat dipergunakan untuk mencari solusi pendekatan dari persoalan-persoalan yang bersifat deterministik.

4.2.1 Sejarah Singkat

Nama Monte Carlo diambil dari nama sebuah kota di Monaco yang terkenal sebagai pusat kasino. Di sana pada umumnya judi menggunakan bilangan yang dibangkitkan secara acak melalui berbagai alat judi. Lalu apa yang sama antara tehnik simulasi komputer dengan casino Monte Carlo? Unsur peluang berperan pada keduanya dan dalam waktu yang panjang hasil yang diharapkan akan muncul. Pemilik kasino ingin agar dalam jangka panjang, dia memperoleh keuntungan, sementara dalam setiap permainan para penjudi memperoleh kesempatan yang masuk akal untuk menang. Metode Monte Carlo menggunakan pembangkit bilangan acak untuk membangkitkan kejadian (lihat Nelson (Nelson 1997).)

Secara sistematik metode Monte Carlo mulai berkembang tahun 1944, walaupun sebelumnya yaitu pada paruh ke dua abad 19 banyak orang melakukan percobaan menjatuhkan jarum diantara dua garis sejajar untuk menghitung pendekatan π. Percobaan tersebut asal mulanya dimulai oleh George Buffon. Tahun 1931 Kolmogorov menunjukkan hubungan antara proses stokastik Markov dengan persamaan differensial.Tahun 1908 Mahasiswa (*Student,* W.S. Gosset) menggunakan percobaan untuk membantunya menemukan distribusi koefisien korelasi. Pada tahun yang bersamaan Mahasiswa (student) menggunakan metode sampling untuk memantapkan keyakinannya pada distribusi yang disebutnya distribusi t.

Penggunaaan riil dari metode Monte Carlo berasal dari penelitian pada bom atom selama perang dunia kedua. Pekerjaan ini menyangkut simulasi langsung dari persoalan probabilistik berkaitan dengan difusi acak neutron pada material fissile. Tetapi perkembangan sistematik ide ini harus menunggu hasil karya Harris and Herman Kahn tahun 1948. Sekitar tahun 1948 Fermi, Metropolis, and Ulam menemukan estimasi Monte Carlo untuk nilai eigen dari persamaan Schrodinger.

Sekitar tahun 1970, perkembangan teori baru dalam kompleksitas komputasi menyebabkan adanya alasan yang lebih tepat dan menjanjikan penerapan metode Monte Carlo. Teori ini mengidentifikasi sekumpulan masalah dimana saat itu orang masih berkonsentrasi mendapatkan solusi eksak. Pertanyaannya apakah metode Monte Carlo dapat menduga solusi persoalan tersebut (Lihat juga Pllana (Pllana n.d.)).

4.2.2 Metode Penerimaan dan Penolakan (*Acceptance-Rejection*)

Penerimaan dan Penolakan Untuk Peubah Acak Kontinu Terbatas

Penerapan metode transformasi invers membutuhkan inves fungsi kumulatif yang tidak selalu mudah diturunkan. Untuk peubah acak kontinu peluang suatu interval tertentu didefinisikan sebagai perbandingan antara perbandingan antara nilai integral yang dibatasi oleh interval bersangkutan dengan nilai integral pada selutuh rentang peuban acak X. Secara geometris perbandingan tersebut ekuivalen dengan perbandingan luas daerah yang dibatasi interval tersebut dengan luas daerah seluruhnya (Lihat Gambar 4.3Penerimaan dan Penolakan Untuk Peubah Acak Kontinu Terbatasfigure.4.3. Perbandingan luas daerah tersebut dapat dihitung dengan cara berikut:

$$\begin{aligned} P(c < X < d) &= \frac{\int_c^d f(x)dx}{\int_R f(x)dx} \\ &= \int_c^d f(x)dx \end{aligned}$$

karena $\int_R f(x)dx = 1$. Dengan demikian pembangkitan data dengan fungsi kepadatan f dapat dilakukan jika kita dapat membangkitkan data secara acak dan merata pada persegi panjang yang dibatasi oleh batas-batas nilai X, 0 dan maksimum $f(x)$, seperti pada Gambar 4.4Penerimaan dan Penolakan Untuk Peubah Acak Kontinu Terbatasfigure.4.4. Ini artinya kita harus bisa membangkitkan titik-titik (x, y) pada daerah bujur sangkar tersebut. Selanjutnya kita menerima atau menolak x dengan ketentuan:

1. terima x, jika (x, y) berada di bawah kurva dan
2. tolak x, jika (x, y) berada di atas kurva.

Prosedur di atas dapat diuraikan secara lebih formal pada algoritma berikut

---

Algoritma 4.3 (Metode Penerimaan-Penolakan)**.** *Membangkitkan data dari peubah acak yang memiliki fungsi kepadatan f untuk $x_0 < x < x_1$ dengan menggunakan metode penerimaan dan penolakan adalah:*

1. *bangkitkan data x dari $U(x_0, x_1)$;*
2. *tentukan F sedemikian sehingga $f(x) \leq F$ untuk semua $x_0 < x < x_1$;*
3. *bangkitkan data y_1 dari $U(0, F)$;*
4. *hitung $y_2 = f(x)$;*
5. *bandingkan y_1 dan y_2*
   (a) *terima x jika $y_1 \leq y_2$;*
   (b) *tolak x jika sebaliknya.*

Gambar 4.3: Ilustrasi Peluang dengan Luas Daerah

Menghitung Integral Tertentu

Menghitung integral dengan menggunakan Monte Carlo pada dasarnya menggunakan prinsip *ekspektasi* pada statistika, yaitu jika X peubah acak dengan fungsi kepadatan $f(x)$, maka:

$$E(u(X)) = \int_{R_X} u(x)f(x)\, dx.$$

Dengan demikian jika kita ingin menghitung integral

$$I = \int_{x_0}^{x_1} g(x)\, dx \tag{4.1}$$

Gambar 4.4: Ilustrasi Prinsip Penerimaan dan Penolakan

dengan $g(x)$ pada umumnyabukan merupakan fungsi kepadatan. Oleh karena itu kita harus memodifikasi I sehingga mengandung fungsi kepadatan. Dari banyak pilihan, maka pilihan yang paling sederhana adalah dengan mengambil distribusi seragam pada interval (x_0, x_1) yaitu $U(x_0, x_1)$, yaitu

$$f(x) = \frac{1}{x_1 - x_0} \text{ untuk } x_0 < x < x_1$$

Jika fungsi ini digabungkan ke bentuk integral yang akan dihitung maka diperoleh

$$
\begin{aligned}
I &= (x_1 - x_0) \int_{x_0}^{x_1} g(x) f(x)\, dx \\
&= (x_1 - x_0)\, E(g(X) \\
&\approx \frac{(x_1 - x_0)}{N} \sum_{i=1}^{N} g(x_i) \text{ dengan } x_i \in U(x_0, x_1)
\end{aligned} \tag{4.2}
$$

Algoritma 4.4. *Prosedur menghitung integral* $I = \int_{x_0}^{x_1} g(x)\, dx$ *adalah:*

1. *bangkitkan* $x_i \sim U(x_0, x_1)$ *untuk* $i = 1, 2, \cdots, N$ *dan* N *yang reletif besar.*
2. *hitung rata-rata* $g(x)$, *yaitu* $\overline{g(x)} = \frac{1}{N} \sum_{i=1}^{N} g(x_i)$
3. *hitung* $I = (x_1 - x_0)\overline{g(x)}$

Contoh 4.4. Misalkan kita ingin menghitung integral berikut:

$$I = \int_0^2 x^2\, dx.$$

Penyelesaian eksak dari integral di atas menghasilkan

$$
\begin{aligned}
I &= \int_0^2 x^2\, dx \\
&= \left.\frac{x^3}{3}\right]_0^2 \\
&= \frac{8}{3}
\end{aligned}
$$

Dengan metode Monte Carlo maka integral tersebut dapat dibuat dengan Program berikut

Program 4.2.

```
n<-500000
a<-0
b<-2
fs<-function(x){
    x^2
}
x<-runif(n,a,b)
y<-mean(fs(x))
int<-(b-a)*y
print(int)
```

Hasil yang diperoleh untuk berbagai n menghasilkan keluaran sebagai berikut

n	I
10	2.19145305188085
100	2.29927627738072
1000	2.61699531342553
10000	2.67844866946301
100000	2.66017086964216

Prosedur menghitung integral tertentu dapat dengan mudah diperluas untuk integral multivariat.

Algoritma 4.5. *Prosedur menghitung integral* $I = \int_{z_0}^{z_1} \int_{y_0}^{y_1} \int_{x_0}^{x_1} g(x, x, y)\, dx\, dy\, dz$ *adalah:*

1. *bangkitkan* $x_i \sim U(x_0, x_1)$, $y_i \sim U(y_0, y_1)$,$z_i \sim U(z_0, z_1)$ *secara independen untuk* $i = 1, 2, \cdots, N$ *dan* N *yang reletif besar.*

2. *hitung rata-rata* $g(x,y,z)$*, yaitu* $\overline{g(x,y,z)} = \frac{1}{N}\sum_{i=1}^{N} g(x_i, y_i, z_i)$

3. *hitung* $I = (x_1 - x_0)(y_1 - y_0)(z_1 - z_0)\overline{g(x,y,z)}$

Contoh 4.5. Hitung

$$I = \int_{10}^{11}\int_{0}^{2\pi}\int_{0}^{\pi}\left(\frac{9}{2\pi}\right) z^2 sin^3(x) dx\ dy\ dz$$

Solusi eksak integral di atas menghasilkan I=1324 (Modifikasi dari Munem & Foulis (Foulis 1978, hal.957). Penghitungan dengan Monte Carlo untuk $N = 10^6$ menghasilkan hasil yang sama.

Program 4.3.

```
#Program contoh menghitung integral lipat tiga
a1<-0
b1<-pi
a2<-0
b2<-2*pi
a3<-10
b3<-11

n<-1000000
x<-runif(n,a1,b1)
y<-runif(n,a2,b2)
z<-runif(n,a3,b3)
fs.int<-function(x,y,z){
```

```
    k<-9/(2*pi)
    (sin(x))^3*k*z^2
}
fs<-fs.int(x,y,z)
int<-(b3-a3)*(b2-a2)*(b1-a1)*mean(fs)
```

4.3 Menghitung Pendekatan Pi

4.3.1 Sejarah Perhitungan Pi

Sejak dulu diketahui bahwa rasio luas lingkaran terhadap kuadrat hjaraknya dan rasio keliling lingkaran dengan diameternya adalah konstan. Namun, pada awalnya belum diketahui bahwa kedua konstanta tersebut adalah sama. Buku-buku kuno menggunakan konstanta yang berbeda untuk kedua rasio tersebut.

Perhitungan π menarik perhatian sejak zaman sebelum masehi (sekitar 1650 SM, di Mesir Kuno). Sejak itu sampai sekarang banyak sekali para matematisi yang melakukan perhitungan baik secara analitik maupun dengan menggunakan komputer. Pada zaman modern sekarang akurasi perhitungan π sempat dijadikan salah satu tes untuk mengukur kecanggihan komputer maupun suatu algorithma. Beberapa hasil perhitungan π diberiikan pada Tabel 4.1Sejarah Perhitungan Pitable.4.1 dan Tabel 4.2Sejarah Perhitungan Pitable.4.2

Tabel 4.1: Perhitungan π secara Analitik

Matematisi	Waktu	Desimal	Nilai
Rhind papyrus	2000 SM	1	3.16045 ($= 4(8/9)^2$)
Archimedes	250 SM	3	3.1418
Aryabhata	499	4	3.1416 (= 62832/2000)
Brahmagupta	640	1	3.1622 ($= \sqrt{10}$)
Fibonacci	1220	3	3.141818
Madhava	1400	11	3.14159265359
Newton	1665	16	3.1415926535897932
Rutherford	1824	208	hanya 152 benar
Shanks	1874	707	hanya 527 benar

4.3.2 Percobaan Jarum Buffon

Buffon melakukan percobaan dengan menjatuhkan secara acak dan seragam jarum (dengan panjang L) diantara dua garis sejajar yang berjarak D, dengan $D > L$ sehingga jarum hanya bisa menyentuh salah satu garis. Selanjutnya Buffon menghitung banyaknya jarum dijatuhnya dan banyaknya jarum menyentuh garis sejajar. Misalkan jarum jatuh dengan titik tengah pada jarak y dari garis bawah dan dengan sudut $x = \theta$. Maka apakah jarum menyentuh garis atau tidak sangat bergantung pada besarnya y dan x, dimana $0 < y < D$ dan $0 < x < \pi$. Sebagai contoh jika $y = 1/2D$ maka berapapun sudutnya jarum tidak akan menyentuh garis karena $L < D$. Demikian juga jika sudut yang dibentuk 0 atau π tetapi y tidak tepat pada garis, maka jarum juga tidak menyentuh garis. Kondisi agar jarum menyentuh garis adalah seperti berikut ini.

1. Jarum akan menyentuh garis bagian bawah jika $y < L/2 \sin x$, yaitu jika

Tabel 4.2: Perhitungan π dengan Mesin

Matematisi	Waktu	Desimal	Mesin
Ferguson	1947	710	Kalkulator
Ferguson, Wrench	1947	808	Kalkulator
Smith, Wrench	1949	1120	Kalkulator
Reitwiesner dkk.	1949	2037	ENIAC
Nicholson, Jeenel	1954	3092	NORAC
Felton	1957	7480	PEGASUS
Genuys	1958	10000	IBM 704
Felton	1958	10021	PEGASUS
Guilloud	1959	16167	IBM 704
Shanks, Wrench	1961	100265	IBM 7090
Guilloud, Filliatre	1966	250000	IBM 7030
Guilloud, Dichampt	1967	500000	CDC 6600
Guilloud, Bouyer	1973	1001250	CDC 7600
Miyoshi, Kanada	1981	2000036	FACOM M-200
Guilloud	1982	2000050	
Kanada, Yoshino, Tamura	1982	16777206	HITACHI M-280H
Ushiro, Kanada	1983	10013395	HITACHI S-810/20
Gosper	1985	17526200	SYMBOLICS 3670
Bailey	1986	29360111	CRAY-2
Kanada, Tamura, Kubo	1987	134217700	NEC SX-2
Kanada, Tamura	1988	201326551	HITACHI S-820/80
Chudnovskys	1989	525229270	
Kanada, Tamura	1989	536870898	
Chudnovskys	1989	1011196691	
Kanada, Tamura	1989	1073741799	
Chudnovskys	1994	4044000000	
Kanada, Tamura	1995	3221225466	
Kanada	1995	6442450938	
Kanada, Takahashi	1999	206158430000	HITACHI SR8000

proyeksi separuh batang jarum, $L/2\sin x$ lebih panjang dari y, diukur dari garis. Dengan kata lain

$$y < L/2\sin x$$

atau

2. Jarum akan menyentuh garis bagian atas jika jika proyeksi setengah panjang jarum, $L/2\sin x$ lebih berar dari jarak titik tengah jarum dengan garis bagian atas, $D - y$. Jadi $D - y < (L/2\sin x)$ atau

$$y > (D - L/2\sin x).$$

Jadi daerah A yang termasuk dalam daerah penyelesaian adalah

$$A = \{(x, y) : 0 < x < \pi; y < (L/2\sin x)\}$$

$$\cup\{(x, y) : 0 < x < \pi; y > (D - L/2\sin x)\}.$$

Dengan menggunakan tehnik kalkulus dapat dihitung bahwa luas $l(A) = 2DL$. Sementara itu luas seluruh daerah yang dibatasi $(0, \pi)$ dan $(0, D)$ adalah πD. Selanjutnya pendekatan π dihitung sebagai berikut:

1. Besarnya peluang jarum menyentuh garis adalah sama dengan banyaknya jarum menyentuh garis dibagi banyaknya jarum jatuh.

$$P = \frac{R}{N}$$

dengan $R =$ banyaknya jarum menyentuh garis dan N banyaknya jarum dijatuhkan.

Gambar 4.5: Ilustrasi Percobaan Jarum dari Buffon

2. dilain pihak secara geometris peluang jarum menyentuh garis sama dengan perbandingan antara

$$P = \frac{\text{luas } A}{\text{luas } S} = \frac{2DL}{D\pi} \tag{4.3}$$

$$= \frac{R}{N} \tag{4.4}$$

3. Dari persamaan di atas dapat diperoleh bahwa

$$\pi \approx \frac{2LN}{RD}$$

Ide percobaan jarum Buffon dapat dimodifikasi dengan menggunakan benda lainnya. Ardana (Ardana 2003) memodifikasi percobaan jarum Buffon dengan per-

Gambar 4.6: $Hit\text{-}Miss$ Monte Carlo meniru Percobaan Buffon

cobaan batang korek a$\mathbf{pi}$ untuk menghitung nilai $\mathbf{pi}$ (π) dan menyederhanakan persoalan dengan mengambil $D = 1$. Lihat Gambar 4.6Percobaan Jarum Buffonfigure.4.6.

4.3.3 Menghitung π dengan Monte Carlo

Metode Penolakan dan Penerimaan atau *hit and miss* Monte Carlo dapat diaplikasikan untuk menghitung pendekatan nilai π. Dengan mengambil $D = 1, L = 0.8$ maka untuk berbagai nilai N diperoleh pendekatan pi seperti berikut ini.

Menghitung π dengan pendekatan lingkaran

Secara geometris nilai π adalah sama dengan luas lingkaran dengan radius satu ($L = \pi r^2 = \pi 1^2 = \pi$). Dengan metode penerimaan dan penolakan kita dapat menghitung pendekatan π dengan cara berikut:

1. menggambar kurva 1/4 lingkaran dengan radius satu pada kuadran pertama. Dengan kata lain kita menggambar kurva $y = \sqrt{1 - x^2}$.
2. membuat daerah bujur sangkar dengan sisi 1 ‘pada kuadran pertama dengan salah satu titik pada $O(0, 0)$;
3. membuat secara random titik-titik dengan koordinat yang berada pada daerah bujur sangkar tadi;
4. menghitung jumlah titik yang berada pada juring lingkaran.
5. maka $1/4\pi$ mendekati perbandingan antara titik yang jatuh pada daerah juring lingkaran dengan keselurtuhan titik pada daerah bujur sangkar tadi(lihat Gambar 4.7Menghitung π dengan pendekatan lingkaranfigure.4.7).

Dalam bahasa S-Plus atau R langkah langkah tadi dapat dituliskan sebagai berikut:

Program 4.4.

```
#Program inti menghitung pendekatan pi dengan cara hit-miss Monte
#Carlo


x<-runif(n,0,1)
y<-runif(n,0,1)
```

```
z<-x^2+y^2
    k<-floor(z)
    c<-n-sum(k)
    pi<-4*c/n
```

n	$\approx \pi$	n	$\approx \pi$
100	3,160000	19600	3,141429
400	3,030000	22500	3,142933
900	3,146667	25600	3,132812
1600	3,145000	28900	3,131903
2500	3,088000	32400	3,134074
3600	3,172222	36100	3,149917
4900	3,095510	40000	3,123400
6400	3,133750	44100	3,148299
8100	3,184691	48400	3,138512
10000	3,146000	52900	3,155085
12100	3,131240	57600	3,144306
14400	3,138611	62500	3,149952
16900	3,124734		

Program Monte Carlo percobaan Buffon

Versi *hit-miss* Monte Carlo dari percobaan Buffon dapat dibuat dengan mengganti jatuhnya jarum dengan panjang L dengan dua peubah yang berdistribusi seragam yaitu $U_1 \sim U(0, \pi)$ mewakili sudut dan $U_2 \sim U(0, D)$ mewakili jarak y, dengan

Gambar 4.7: Ilustrasi Penerimaan dan Penolakan untuk Menghitung π dengan Metode Monte Carlo

$L < D$ (Lihat Gambar 4.6Percobaan Jarum Buffonfigure.4.6). Aturan penerimaan dan penolakan adalah sebagai berikut:

$$(x,y)\begin{cases} \text{diterima jika } y < L/2\sin x \text{ atau } y > (D-L/2)\sin x; \\ \text{ditolak untuk yang lain, } L/2\sin x \;<\; y \;< (D-L/2)\sin x. \end{cases}$$

Sedangkan nilai π ditentukan dengan

$$\pi \approx \frac{2 \times L \times \text{jumlah } (x,y)}{D \times \text{jumlah } (x,y) \text{ yang diterima}}$$

Besarnya nilai pendekatan π untuk berbagai ukuran sampel n adalah

n	$\approx \pi$
100	2.75862068965517
2500	3.07455803228286
10000	3.17082837891399
40000	3.12881935956979
62500	3.15845993493573

Cara lain pendekatan Monte Carlo untuk menghitung π adalah tidak menggunakan penolakan dan penerimaan tetapi langsung menghitung luas daerah yang diwakili oleh integral

$$I = 2\int_{-1}^{1} \sqrt{1-x^2}\;\; dx$$

yang merupakan luas setengah lingkaran dengan jari-jari 1. Dengan cara ini untuk $N = 10^6$ diperoleh $\pi \approx 3,143$.

4.4 Daftar Bacaan

Aplikasi metode Monte Carlo sangat luas. Buku-buku tentang metode Monte Carlo belum banyak tersedia di perpustakaan, lebih-lebih dalam bahasa Indonesia.

Namun beberapa informasi dapat dilihat di beberapa situs internet antara lain

1. `http://www.chem.unl.edu/zeng/joy/mclab/mcintro.html`
2. `http://www.npac.syr.edu/users/paulc/lectures/montecarlo/p_montecarlo.html`
3. `http://csep1.phy.ornl.gov/mc/mc.html`
4. `http://csep1.phy.ornl.gov/guidry/phys594/lectures/monte_carlo/mc.html`
5. `http://www.circle4.com/pww/mc/`

4.5 Latihan Soal-soal

1. Diketahui X dengan fungsi kepadatan

$$f(x) = kx^2 \text{ untuk } 1 < x < 5,$$

   (a) tentukan k;

   (b) selidiki apakah X dapat dibangkitkan dengan transformasi invers;

   (c) buat program untuk membangkitkan 50 data dari X selanjutnya buat grafik densitasnya.

2. Diketahui X dengan fungsi kepadatan,

$$f(x) = k_1x^2 + k_2x \text{ untuk } 0 < x < 4,$$

   (a) tentukan k;

(b) selidiki apakah X dibangkitkan dengan transformasi invers;

(c) buat program untuk membangkitkan 50 data dari X selanjutnya buat grafik densitasnya.

3. Metode penerimaan dan penolakan berlaku untuk peubah acak dengan batas berhingga, selidiki untuk tak berhingga pendekatan apa yang bisa dilakukan.

   (a) Buat program untuk membangkitkan X yang memiliki fungsi kepadatan $f(x) = k/x$ untuk $0 < x < \infty$ dengan menggunakan metode penerimaan dan penolakan.

   (b) Buat program untuk membangkitkan X yang berdistribusi eksponensial dengan menggunakan metode penerimaan dan penolakan.

4. Tentukan pendekatan yang bisa dilakukan untuk menghitung

$$\int_{\infty}^{a} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}\, dx,$$

untuk suatu nilai a yang ditentukan, misalnya $a = -1, 0, 1$ dan seterusnya.

DAFTAR PUSTAKA

Ardana, N. 2003, Korek api dan pi. Bahan Pelatihan Pemodelan Matematika Jurusan Matematika FMIPA IPB.

Banks, J. 1998, Principles of simulations, *in* J. Banks, ed., 'Handbook of Simulations', John Wiley & Sons, New York, chapter I, pp. 3–30.

Borwein, E. B. . J. 1989, *Collins Dictionary Mathematics*, Collins, Great Britain.

Coddington, P. n.d., *CPS 713: Monte Carlo Simulation for Statistical Physiscs*, http://www.npac.syr.edu/users/paulc/lectures/montecarlo/p_montecarlo.html, Syracuse University.

Foulis, M. M. . D. 1978, *Calcullus*, Worth Publishers Inc.

Melamed, R. Y. R. . B. 1998, *Modern Simulation and Modeling*, John Wiley & Sons Inc, New York.

Nelson, P. 1997, *Monte Carlo Simulation*, http://www.circle4.com/pww/mc/.

Pllana, S. n.d., *History of Monte Carlo Method*, http://stud4.tuwien.ac.at/∼e9527412/.

Tirta, I. M. 2001, *Pemrograman Statistika dengan S-Plus 4.5*, FMIPA Universitas Jember, Jember. Diktat Kuliah.

Tirta, I. M. 2003, *Pengantar Statistika Matematika*, FMIPA Universitas Jember, Jember. Diktat Kuliah.

INDEKS

animasi, 8

computer
- intensive statistics, 3

distribusi
- χ^2, 30
- F, 33
- gamma, 30
- t, 32

generator
- bilangan acak, 10
- Linier Congruential Generator (LCG), 10
- Multiplicative Linier Congruential Generator (MLCG), 10

karakteristik, 4

kongruen, 12

konvergen, 18

model, 4
- matematika, 4

Monte Carlo, 46
- hit-miss, 50
- integrasi, 50, 52
- pi, 55
- sejarah, 46

normal
- bivariat, 34
- multivariat, 35
- standar, 26
- univariat, 28

numerik, 4

parameter, 2

Penerimaan-Penolakan, 50

random number
    psudo, 10
reproduktif
    gamma, 46
resampling method, 3

sampling
    ulang, 3
seed, 10
simulasi, 43

transformasi
    bivariat, 34
    Box-Muller, 26
    invers, 39, 42
    multivariat, 34
    peubah acak, 17, 25