Forum Statistika dan Komputasi, Oktoberl 2005, p: 8 - 11 Vol. 10 No. 2
ISSN : 0853-8115

PEMBANGKITAN BILANGAN ACAK UNTUK SIMULASI MONTE CARLO NON-PARAMETRIK

Bagus Sartono

Departemen Statistika FMIPA IPB

Ringkasan

Metode simulasi Monte Carlo banyak digunakan oleh para analis, misalnya oleh mereka yang menduga nilai resiko dari suatu portfolio bank. Penggunaan simulasi Monte Carlo seringkali terkendala keterbatasan pengetahuan mengenai bentuk sebaran data. Penentuan sebaran dengan didahului pengujian sebaran memiliki kelemahan dalam hal menentukan sebaran yang paling sesuai. Kernel merupakan metode non-parametrik untuk menduga bentuk sebaran yang tidak tergantung pada parameter tertentu. Tulisan ini memberikan usulan algoritma pembangkitan bilangan acak berdasarkan bentuk fungsi kepekatan yang diduga dengan metode kernel menggunaan teknik tabel look-up*. Penentuan banyaknya grid pada pembentukan tabel* look-up *diperkirakan mempengaruhi hasil bangkitan data.*

Kata kunci: *simulasi Monte Carlo, metode kernel, tabel* look-up

PENDAHULUAN

Dalam banyak kesempatan, teknik pemecahan masalah menggunakan simulasi banyak digunakan oleh peneliti maupun analis. Istilah simulasi merujuk pada metode analisis yang meniru (*imitate*) sistem pada kehidupan nyata, terutama ketika analisis yang rumit dan kompleks secara matematis sangat sulit untuk dihasilkan. (http://www.decisioneering.com/monte-carlo-simulation.html).

Salah satu jenis simulasi adalah teknik simulasi Monte Carlo, yang secara acak membangkitkan bilangan atau nilai-nilai dari suatu variabel dengan sebaran tertentu berulang-ulang. Sementara dalam *glossary* Wolfram dituliskan bahwa simulasi Monte Carlo merupakan suatu metode untuk menyelesaiakan masalah menggunakan pembangkitan bilangan acak yang sesuai dan selanjutnya mengamati sifat-sifat tertentu yang dihasilkan, dan metode ini berguna untuk mendapatkan solusi numerik pada masalah yang terlalu rumit diselesaikan secara analitik (http://mathworld.wolfram.com/MonteCarloMethod.html).

Teknik ini diperkenalkan oleh Stanslaw Ulam (1909 – 1986) pada tahun 1946. Pemberian nama Monte Carlo diberikan oleh rekannya Nicolas Constantine Metropolis (1915 – 1999) yang mengacu pada kota sebuah tempat di Monaco, yang mengandalkan daya tarik pada tempat-tempat permainan berpeluang, seperti roda rolet, dadu, dan mesin slot, yang memiliki perilaku acak.

Perilaku acak dari permainan-permainan tersebut mirip dengan cara kerja simulasi Monte Carlo yang memilih nilai variabel secara acak untuk mensimulasikan suatu model. Ketika kita melempar sebuah dadu, kita ketahui bahwa kemungkinan yang terjadi adalah munculnya mata dadu 1, 2, 3, 4, 5, atau 6, namun kita tidak dapat mengetahui mana yang akan muncul pada lemparan tertentu. Ini sama saja dengan nilai dari variabel yang kita bangkitkan kita ketahui kisaran nilainya, tapi ada ketidakpastian nilai yang dihasilkan (misalnya tingkat bunga, harga saham, dsb).

Pada penghitungan nilai resiko (VaR = *value at risk*) di berbagai lembaga keuangan termasuk bank, teknik simulasi Monte Carlo cukup banyak digunakan. Teknik ini digunakan pada penghitungan VaR berbagai resiko termasuk resiko pasar, resiko kredit, dan resiko operasional. Pendekatan simulasi ini dilakukan pada penghitungan resiko pasar ketika diperoleh kesulitan ketika portfolio yang dimiliki berupa kumpulan banyak instrument yang masing-masing saling berkorelasi. Sedangkan pada resiko kredit dan resiko operasional, teknik simulasi Monte Carlo digunakan pada saat menggabungkan distribusi banyaknya kejadian (*operational loss, default*) dalam rentang waktu tertentu dan besarnya kerugian (*loss, loss given default*) dari setiap kejadian (Cruz 2003, Crouhy *et al* 2001).

Persoalan kemudian muncul, ketika teknik simulasi Monte Carlo memerlukan pengetahuan mengenai bentuk sebaran data, misalnya bentuk sebaran nilai kerugian pada setiap kejadian dalam penghitungan resiko

ISSN : 0853-8115

operasional. Beberapa masalah yang muncul adalah:

- pada data tertentu dapat diidentifikasi sesuai dengan lebih dari satu bentuk sebaran
- pada data yang lain, ada kemungkinan tidak dapat diperoleh sebarannya (jika dilakukan pengujian sebaran-sebaran umum)

Kesulitan ini kemudian memunculkan ide penggunaan *historical simulation,* dengan tidak melakukan upaya menentukan sebaran apa yang sesuai. Kendala yang muncul adalah jika nilai yang didapatkan tentu saja tidak 'berubah' dalam pengertian tidak ada yang berbeda dengan nilai-nilai dari data historis.

Tulisan ini mencoba mengusulkan penggunaan teknik lain, yaitu menggunakan simulasi monte carlo yang didasarkan pada pendugaan fungsi kepekatan peluang univariat secara non-parametrik.

PENDUGAAN FUNGSI KEPEKATAN PELUANG NON-PARAMETRIK

Pendugaan fungsi kepekatan peluang kernel univariat terboboti melibatkan sebuah peubah X dan peubah pembobot W. Andaikan (X_i, W_i), i = 1, 2, ..., n adalah contoh X dan W yang berukuran n. Penduga kernel terboboti bagi fungsi kepekatan peluang X, $f(x)$, adalah

$$\hat{f}(x) = \frac{1}{\sum_{i=1}^{n} W_i} \sum_{i=1}^{n} W_i \varphi_h (x - X_i)$$

dengan h adalah lebar jendela dan

$$\varphi_h(x) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{x^2}{2h^2}\right)$$

adalah fungsi kepekatan normal baku yang diskalakan terhadap lebar jendela (SAS Institute Inc, 1999). Jika $h \rightarrow 0$ dan $nh \rightarrow \infty$ maka lebar jendela optimum adalah

$$h_{op} = \left[\frac{1}{2\sqrt{\pi}n\int (f'')^2}\right]^{1/5}$$

Nilai optimum ini tidak diketahui sehingga diperlukan metode pendekatan. Penurunan dari hasil di atas dapat dilihat pada Silverman (1986). Silverman (1986) mengusulkan penggunaan nilai

$$h_{op} = 0.9 \min\{s, (Q3 - Q1)/1.34\}\, n^{-1/5}$$

dengan s adalah simpangan baku contoh, dan Q1 dan Q3 adalah kuartil pertama dan ketiga.

PEMBANGKITAN BILANGAN ACAK UNTUK SIMULASI MONTECARLO NON-PARAMETRIK

Jika penduga dari $f(x)$ dapat ditemukan maka simulasi Monte Carlo didasarkan pada pembangkitan bilangan acak berdasarkan sebaran $\hat{f}(x)$. Persoalan berikutnya adalah bagaimana proses pembangkitan bilangan acak tersebut mengingat bentuk $\hat{f}(x)$ yang tidak bisa dituliskan dengan mudah. Bagian ini akan memberikan usulan algoritma teknik pembangkitan menggunakan *look-up table method*, seperti yang dibahas oleh Morgan (1984).

Dasar teori yang digunakan adalah sebagai berikut.

Teori.
Jika X adalah peubah acak dengan fungsi kepekatan peluang $f_X(x)$ dan fungsi sebaran kumulatif $F_X(x)$, maka $F_X(x)$ memiliki sebaran seragam (0, 1).

Bukti.
Andaikan $Y = F_X(x)$, maka $F_Y(y) = P(Y \leq y) = P(X \leq x^* \mid F_X(x^*) = y) = F_X(x^*) = y$, untuk $0 \leq y \leq 1$. Karena $F_Y(y) = y$ maka $Y = F(x)$ memiliki sebaran seragam (0, 1).

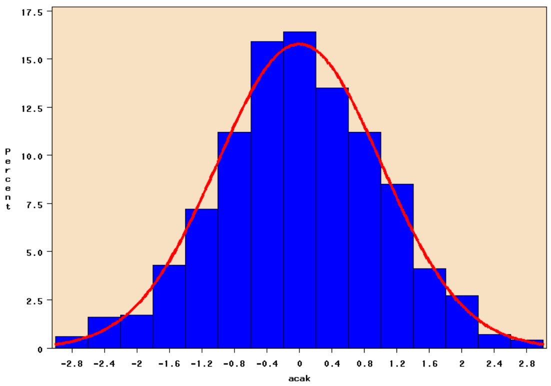
Berdasarkan teori di atas, maka dapat disusun algoritma untuk pembangkitan bilangan acak dengan sebaran $\hat{f}(x)$ sebagai berikut:

1. andaikan $x_1, x_2, \ldots, x_n$ adalah contoh acak berukuran n
2. lakukan pendugaan $\hat{f}(x)$ dengan metode kernel, gunakan h_{op} seperti yang disarankan Silverman (1986)
3. susun tabel berisi dua kolom yaitu nilai x dan $\hat{f}(x)$ untuk $x = a, a+d, a+2d, a+3d, \ldots, a+md$, dengan $a = \min\{x_1, x_2, \ldots, x_n\} - s$, $a+md = \max\{x_1, x_2, \ldots, x_n\} + s$, d sebuah konstanta yang kecil
4. hitung nilai $\hat{F}(x)$ dengan teknik integral Riemann Sum, dengan $\hat{F}(a+kd) = \sum_{i=0}^{k} \hat{f}(a+id)d$
5. susun sebuah tabel look-up antara x dan $\hat{F}(x)$
6. bangkitkan bilangan acak Seragam (0, 1), yang merupakan representasi F(x)
7. dapatkan nilai x yang bersesuaian dengan F(x) hasil bangkitan dari langkah nomor 6
8. ulangi tahap 5, 6, dan 7 sampai diperoleh ukuran sampel simulasi yang diinginkan

ISSN : 0853-8115

Proses pembangkitan bilangan acak menggunakan tabel look-up akan terkendala tidak adanya kesamaan nilai bilangan acak seragam yang dibangkitkan dengan nilai dugaan fungsi sebaran kumulatif $\hat{F}(x)$. Untuk mengatasi hal ini dapat digunakan pendekatan dengan memilih nilai x yang memiliki $\hat{F}(x)$ yang selisihnya paling kecil dengan bilangan acak seragamnya.

Lampiran tulisan ini menyajikan program yang disusun untuk pembangkitan bilangan acak yang didasarkan pada dugaan fungsi kepekatan peluangnya dengan metode kernel. Program disusun menggunakan SAS versi 9.1. Beberapa opsi perlu diubah ubah tujuan tertentu



Gambar 1. Histogram salah satu hasil bangkitan menggunakan algoritma yang diusulkan, berdasarkan sampel dari sebaran Normal (0, 1)

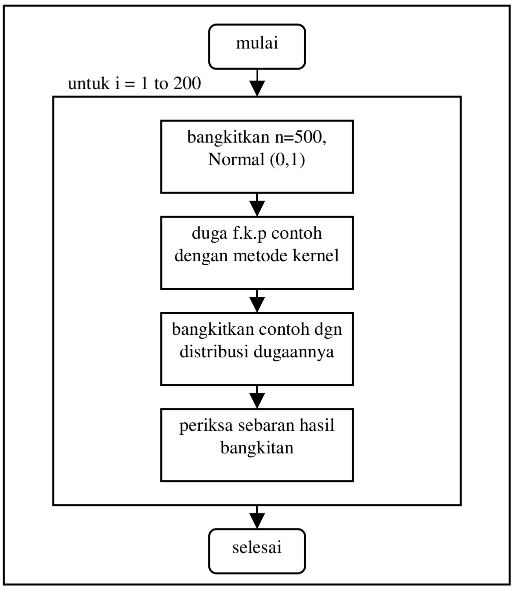
Gambar 1 menampilkan hitogram dan overlay kurva normal salah satu hasil pembangkitan data dari sampel berukuran n = 500 yang diambil dari sebaran Normal (0, 1). Sesuai dengan program pada Lampiran, digunakan dugaan kernel dengan fungsi pemulus Gauss dan grid sebanyak 1000. Bilangan acak yang dibangkitkan berukuran n = 500. Terlihat bahwa hasil pembangkitan berdasarkan dugaan fungsi kepekatannya cukup memuaskan.

DISKUSI

Tabel 1 menampilkan hasil simulasi pembangkitan bilangan acak berdasarkan dugaan f.k.p dari contoh berukuran 500 yang dibangkitkan dari sebaran Normal (0, 1). Pengujian dilakukan terhadap bentuk sebaran hasil bangkitan dengan ukuran 20, 50, dan 200. Algoritma yang digunakan seperti terlihat pada Gambar 2.

Tabel 1. Pemeriksaan sebaran contoh hasil bangkitan, menggunakan uji Shapiro-Wilk dengan taraf 5%.

ukuran contoh bangkitan	proporsi sesuai dengan sebaran asal
20	98.5%
50	94.0%
200	86.5%



Gambar 2. Algoritma simulasi pemeriksaan sebaran berdasarkan dugaan f.k.p contoh.

Hasil pada Tabel 1 memperlihatkan bahwa pada ukuran contoh 20 dan 50, hasil bangkitan data sangat memuaskan. Namun pada ukuran contoh 200 hanya diperoleh kesesuaian sebesar 86.5%.

Penulis memperkirakan bahwa penentuan banyaknya grid dalam penyusunan tabel look-up mempengaruhi hasil bangkitan. Dalam kata lain, penentuan nilai d pada langkah 3 algoritma pembangkitan bilangan acak perlu diperhatikan. Nilai d yang semakin kecil, atau grid yang semakin banyak mungkin akan memberikan hasil yang lebih baik. Tabel 1 menggunakan grid sebanyak 1000.

Ucapan Terima Kasih

Tulisan ini diilhami oleh diskusi mengenai penghitungan VaR resiko operasional dengan tim yang diketuai Dr. M. Muslich. Terima kasih disampaikan juga kepada Bapak Haryanto Tanudjaja atas ijin penggunaan SAS 9.1. Serta kepada Anang Kurnia, MSi dan Dr. I Made

Forum Statistika dan Komputasi, Oktoberl 2005, p: 8 - 11 Vol. 10 No. 2
ISSN : 0853-8115

Sumertajaya (Dept. Statistika FMIPA-IPB) atas kesediaan diskusinya. Makro SAS yang digunakan untuk simulasi dapat diperoleh dengan menghubungi penulis.

DAFTAR PUSTAKA

http://www.decisioneering.com/monte-carlo-simulation.html (26 Oktober 2005).

http://mathworld.wolfram.com/MonteCarloMethod.html (26 Oktober 2005).

Crouhy, M., D. Galay, & R. Mark. 2001. Risk Management. New York: McGraw-Hill.

Cruz, M. G. 2003. Modeling, Measuring and Hedging Operational Risk. Chichester: John Wiley & Sons Ltd.

Morgan, B.J.T. 1984. Element of Simulation. New York: Chapman and Hall, Ltd.

SAS Institute Inc. 1999. *SAS/STAT® User's Guide, Version 8*, Cary, NC: SAS Institute Inc.

Silverman, B.W. 1986. *Density Estimation for Statistics and Data Analysis,* London: Chapman and Hall.

Lampiran.
Program SAS untuk membangkitkan bilangan acak berdasarkan dugaan non-parametrik fungsi kepekatan peluangnya.

```
* pendugaan f.k.p dengan metode kernel ;
proc kde data=normal;
   univar x /levels out=bagus ngrid = 1000 gridl
= -4 gridu = 4;
run;

* penghitungan f.s. kumulatif --> F(x) duga ;
data bagus (keep = x density);
      set bagus;
      x = value;
      if density < 0 then density = 0;

proc iml;
      use bagus;
            read all into x;
      d = x[2,2] - x [1,2];
      y = x[1, 1]*d;
      n=nrow(x);
      print n;
      do i = 2 to n;
            l = x[i,1]*d;
            k = y[i-1,1] + l ;
            y = y//k;
      end;
      x = x||y;
      create data from x ;
            append from x;
quit;

data bagus (keep = x density dist);
set data;
density = col1;
x = col2;
dist = col3;
* pembangkitan bilangan acak dan penggunaan
look-up table ;
%macro generate;
%do i = 1 %to 500;
      data generate;
            x = ranuni(0);
            output; run;

      data _null_;
            set generate;
            if _n_ = 1 then pw=x;
            call sympput ('n',pw); run;

      data acak;
            set bagus;
            diff = abs(dist - &n); run;
      proc sort data=acak;
            by diff; run;

      data _acak1 (keep =acak);
            set acak;
            if _n_ = 1 then acak = x;
            if _n_ = 1; run;

      proc append base=acak1 data=_acak1
force;run;

%end;

%mend;
%generate;
run;

* tampilan histogram bilangan acak hasil
bangkitan ;
proc univariate data=acak1 noprint;
var acak;
histogram acak/caxes=BLACK
cframe=CXF7E1C2 waxis= 1
      cbarline=BLACK cfill=BLUE pfill=SOLID
      vscale=percent hminor=0 vminor=0
      name='HIST'
      normal( mu=est sigma=est w=3
color=RED);
run;
```

⊗