BAB 3

PEMBANGUNAN MODEL SIMULASI MONTE CARLO

3.1 Simulasi Monte Carlo

Simulasi Monte Carlo merupakan salah satu metode simulasi sederhana yang dapat dibangun secara cepat menggunakan *spreadsheet*. Penggunaan simulasi ini didasarkan pada probabilitas yang diperoleh dari data historis sebuah kejadian dan frekuensinya.

3.1.1 Sejarah Monte Carlo

Sebelumnya harus diketahui pengertian simulasi itu sendiri. Simulasi adalah suatu solusi analitis dari sebuah sistem yang digunakan untuk memecahkan berbagai masalah atau menguraikan persoalan-persoalan dalam kehidupan nyata yang penuh dengan ketidakpastian ketika solusi matematis tidak memadai, dengan menggunakan model atau metode tertentu dan lebih ditekankan pada pemakaian komputer untuk mendapatkan solusinya.

Penggunaan nama Monte Carlo, yang dipopulerkan oleh para pioner bidang tersebut (termasuk Stanislaw Marcin Ulam, Enrico Fermi, John von Neumann dan Nicholas Metropolis), merupakan nama kasino terkemuka di Monako. Penggunaan

keacakan dan sifat pengulangan proses mirip dengan aktivitas yang dilakukan pada sebuah kasino. Dalam autobiografinya *Adventures of a Mathematician*, Stanislaw Marcin Ulam menyatakan bahwa metode tersebut dinamakan untuk menghormati pamannya yang seorang penjudi, atas saran Metropolis.

Penggunaannya yang cukup dikenal adalah oleh Enrico Fermi pada tahun 1930, ketika ia menggunakan metode acak untuk menghitung sifat-sifat neutron yang waktu itu baru saja ditemukan. Metode Monte Carlo merupakan simulasi inti yang digunakan dalam proyek Manhattan, meski waktu itu masih digunakan peralatan komputasi yang sangat sederhana. Sejak digunakannya komputer elektronik pada tahun 1945, Monte Carlo mulai dipelajari secara mendalam. Pada tahun 1950-an, metode ini digunakan di Laboratorium Nasional Los Alamos untuk penelitian awal pengembangan bom hidrogen, dan kemudian sangat populer dalam bidang fisika dan riset operasi. *Rand Corporation* dan Angkatan Udara AS merupakan dua institusi utama yang bertanggung jawab dalam pendanaan dan penyebaran informasi mengenai Monte Carlo waktu itu, dan mereka mulai menemukan aplikasinya dalam berbagai bidang.

Penggunaan metode Monte Carlo memerlukan sejumlah besar bilangan acak, dan hal tersebut semakin mudah dengan perkembangan pembangkit bilangan pseudoacak, yang jauh lebih cepat dan praktis dibandingkan dengan metode sebelumnya yang menggunakan tabel bilangan acak untuk sampling statistik.

3.1.2 Pembangunan Model Simulasi *Monte Carlo*

Jika suatu sistem mengandung elemen yang mengikutsertakan faktor kemungkinan, maka model yang digunakan adalah model Monte Carlo. Dasar dari penggunaan simulasi Monte Carlo adalah percobaan elemen kemungkinan dengan menggunakan sampel random (acak).

Metode yang dilakukan sebelum proses simulasi Monte Carlo terbagi menjadi beberapa tahapan sebagai berikut:

1. Membuat distribusi kemungkinan untuk variabel tertentu.

Gagasan dasar dari simulasi Monte Carlo adalah membuat nilai dari tiap variabel yang merupakan bagian dari model yang dipelajari. Banyak variabel dalam dunia nyata yang secara alami mengandung berbagai kemungkinan yang dapat disimulasikan.

Salah satu cara umum untuk membuat distribusi kemungkinan untuk suatu variabel adalah dengan memperhitungkan data hasil masa lalu. Kemungkinan atau frekuensi relatif untuk tiap kemungkinan hasil dari tiap variabel ditentukan dengan membagi frekuensi dengan jumlah total observasi. Variabel acak yang digunakan dalam tugas akhir ini adalah variabel acak diskrit.

Telah dijelaskan pada bab sebelumnya bahwa jika $f(x)$ adalah suatu fungsi peluang atau distribusi peluang suatu peubah acak diskrit X bila untuk setiap hasil x yang mungkin, memenuhi :

1. $f(x) \geq 0$

2. $\sum_{x} f(x) = 1$

3. $P(X = x) = f(x)$

Contoh kasus : Permintaan akan buku di toko “merapi” selama 250 hari ke belakang adalah terlihat dalam tabel berikut:

Tabel 3.1 Permintaan Buku

Jumlah permintaan	Frekuensi permintaan
0	10
1	20
2	30
3	40
4	50
5	60
6	40
jumlah	250 hari

Keadaan tersebut dapat diubah menjadi distribusi peluang (dengan asumsi tingkat penjualan masa lalu akan tetap bertahan sampai ke masa depan) dengan membagi frekuensi permintaan dengan total frekuensi permintaan. Tabel distribusinya adalah sebagai berikut :

Tabel 3.2 Distibusi Peluang Permintaan Buku

Jumlah permintaan	Fungsi peluang $f(x)$
0	$10/250 = 0,04$
1	$20/250 = 0,08$
2	$30/250 = 0,12$
3	$40/250 = 0,16$
4	$50/250 = 0,20$
5	$60/250 = 0,24$
6	$40/250 = 0,16$
Jumlah	$250/250 = 1$

2. Membangun distribusi kumulatif untuk tiap variabel yang penting di tahap pertama.

Distribusi kumulatif $F(x)$ suatu peubah acak X dengan distribusi peluang $f(x)$ dinyatakan oleh $F(x) = P(X = x) = \sum_{1 \le x} f(t)$. Untuk contoh sebelumnya, distribusi kumulatifnya dapat disajikan dalam tabel berikut :

Tabel 3.3 Distibusi Kumulatif Permintaan Buku

Jumlah permintaan	Fungsi peluang	Fungsi peluang komulatif
0	0,04	0,04
1	0,08	0,12
2	0,12	0,24
3	0,16	0,40
4	0,20	0,60
5	0,24	0,84
6	0,16	1

3. Menentukan angka acak untuk tiap variabel.

Setelah menentukan distribusi probabilitas kumulatif untuk tiap variabel dalam simulasi, selanjutnya tentukan batas atau interval angka acak yang mewakili tiap kemungkinan hasil. Penentuan interval didasarkan oleh kemungkinan kumulatifnya. Interval angka acak untuk contoh permintaan buku toko "Merapi" sebagai berikut :

Tabel 3.4. Interval Angka Acak Permintaan Buku

Jumlah permintaan	Fungsi peluang	Fungsi peluang komulatif	Interval angka acak
0	0,04	0,04	01 s/d 04
1	0,08	0,12	05 s/d 12
2	0,12	0,24	13 s/d 24
3	0,16	0,40	24 s/d 40
4	0,20	0,60	41 s/d 60
5	0,24	0,84	61 s/d 84
6	0,16	1,00	85 s/d 100

Berikut adalah proses simulasi Monte Carlo, yaitu :

1. Membuat angka acak

Pembangkitan angka acak untuk tugas akhir ini menggunakan program MATLAB. Misalkan untuk contoh sebelumnya, angka random yang diperoleh adalah

Tabel 3.5. Contoh Angka Acak

Angka acak
28
50
78
8
16
61
98
51
45
21

2. Membuat simulasi dari rangkaian percobaan

Berdasarkan angka acak yang telah diperoleh seperti tabel di atas, maka dapat dibuat simulasi untuk 10 hari. Contohnya untuk angka acak 78, angka itu terletak pada interval angka acak 61 s/d 84 yang berati permintaan 5 buah buku. Tabel 3.10 dapat disimulasikan menjadi tabel yang bernilai jumlah permintaan sebagai berikut:

Tabel 3.6. Hasil Simulasi dari Angka Acak

Hari	Angka acak	Permintaan (simulasi)
1	28	3
2	50	4
3	78	5
4	8	1
5	16	2
6	61	5
7	98	6
8	51	4
9	45	4
10	21	2
Jumlah		36

Total permintaan untuk 10 hari adalah 36 buah buku, dengan rata-rata permintaan per hari adalah 3,6 buku.

3.1.3 Pembangkitan Bilangan Acak pada Simulasi Monte Carlo Menggunakan Program MATLAB

Belakangan ini, istilah Monte Carlo telah menjadi sinonim dengan simulasi probabilitas. Namun secara khusus teknik Monte Carlo dapat didefinisikan sebagai suatu teknik untuk memilih angka-angka secara acak dari suatu distribusi probabilitas untuk digunakan dalam suatu percobaan dari suatu simulasi. Simulasi ini adalah proses repetitif sederhana yang membangkitkan *(generate)* solusi deterministik untuk kasus yang diberikan. Setiap solusi mewakili suatu set nilai deterministik dari variabel acak yang digunakan. Elemen utama dari proses Monte Carlo adalah membangkitkan angka acak dari distribusi probabilitas yang telah ditentukan.

Sebelum memulai proses simulasi, terlebih dahulu ditentukan berapa jumlah minimum simulasi yang harus dilakukan. Rumus yang digunakan untuk menghitungnya adalah:

$$n = \frac{Z_{(\alpha/2)}{}^{2}.s^{2}}{B^{2}}$$

dimana:

n = jumlah simulasi yang diperlukan

Z = nilai invers dari distribusi normal

s = standar deviasi sampel

B = interval kesalahan *(margin of error)*

Beberapa rumus dan asumsi yang harus dihitung terlebih dahulu adalah:

1. Standar deviasi untuk sampel (s)
2. Tingkat kepercayaan = diambil sebesar 95%
3. Tingkat signifikansi = 100% - tingkat kepercayaan
4. Interval kesalahan $B = s/\sqrt{N}$
5. Perhitungan nilai Z

Proses simulasi dimulai dengan pembangkitan bilangan acak. Untuk mendapatkan bilangan yang benar-benar acak, secara manual dapat dilakukan dengan menggunakan undian, arisan, atau pemakaian mesin *roullete*. Tetapi secara komputasi, hal ini sulit dilakukan. Hal ini disebabkan bahwa komputer merupakan mesin deterministik, sedangkan bilangan acak muncul sebagai kejadian yang probabilistik. Satu-satunya cara untuk mendapatkan bilangan acak adalah dengan menggunakan *pseudo random generator* (pembangkit bilangan acak semu), di mana bilangan acak diperoleh secara deterministik (aritmatik).

Untuk pembangkitan bilangan acak pada simulasi Monte Carlo bisa digunakan beberapa macam program atau *software*. Pada tugas akhir ini,

pembangkitan bilangan acak pada simulasi Monte Carlo menggunakan program MATLAB 5.3. Metode pembangkitan bilangan acak pada MATLAB juga beragam, namun hanya akan dibahas pembangkitan bilangan acak menggunakan *Linear Congruent Method (LCM)*.

Metode *Linear Congruent* digunakan untuk membangkitkan bilangan acak $r_1, r_2, ..., r_n$ yang bernilai $[0, m]$ dengan memanfaatkan nilai sebelumnya. Untuk membangkitkan bilangan acak ke $n+1$ atau (r_{n+1}) dengan metode *Linear Congruent*, didefinisikan:

$$r_{n+1} = (ar_n + c) \bmod m$$

dimana :

r_{n+1} = bilangan acak ke $n+1$

a, c, m = nilai pembangkit

Untuk pembangkitan bilangan acak menggunakan LCM, kita harus mendefinisikan nilai r_0 terlebih dahulu, r_0 disebut nilai awal, biasanya nilai ini yang digunakan dalam proses *randomize* (mengacak di awal atau state awal).

Contoh kasus : Misalkan ditentukan *a=4, c=1* dan $r_1 = 3$, maka bilangan acak 0 s/d 8 (*m=9*) dapat dihitung:

$$r_2 = ((4)(3)+1) \bmod 9 = 4$$
$$r_3 = ((4)(4)+1) \bmod 9 = 8$$
$$r_4 = ((4)(8)+1) \bmod 9 = 6$$

dan seterusnya.

Algoritma simulasi Monte Carlo menggunakan metode *linear congruent* (LCM) adalah :

1. Masukkan *a, c, m* dan r_1
2. Masukkan berapa bilangan acak yang akan dibangkitkan (*n*)
3. Untuk *i=1 s/d n* : hitung $r_{i+1} = (a.r_i + c) \bmod m$

Implementasi algoritma simulasi Monte Carlo menggunakan LCM adalah sebagai berikut :

Misalkan kita definisikan a = 4, c = 1, dan m =9 sehingga program MATLAB untuk pembangkitan bilangan acak menjadi sebagai berikut :

```
% Mendefinisikan nilai pembangkit
a=4; c=1; m=9;
% Mendefinisikan nilai state awal
r(1)=5;
%Proses pembangkitan 20 bilangan acak
for k=1:20
   r(k+1)=mod(a*r(k)+c,m);
end
%Menampilkan bilangan acak
disp(r)
```

Gambar 3.1. Program Matlab dengan *a = 4, c = 1, dan m =9*

sehingga output untuk program di atas adalah

```
» Columns 1 through 12

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

Columns 13 through 21

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

Gambar 3.2. Output dari Program Matlab dengan *a = 4, c = 1, dan m =9*

Berdasarkan contoh tersebut terlihat bahwa terjadi pengulangan bilangan setiap 9 bilangan, ini terlihat bahwa bilangan acak yang dihasilkan tidak benar-benar acak. Agar bilangan acak yang dikeluarkan tidak terjadi pengulangan dapat dilakukan dengan mengganti nilai pembangkitnya. Misalkan a, c, m, r_1 = (327,3,500,11), sehingga programnya menjadi seperti berikut :

```
% Mendefinisikan nilai pembangkit
a=327; c=3; m=500;
% Mendefinisikan nilai state awal
r(1)=11;
%Proses pembangkitan 20 bilangan acak
for k=1:20
   r(k+1)=mod(a*r(k)+c,m);
end
%Menampilkan bilangan acak
disp(r)
```

Gambar 3.3. Algoritma dengan a, c, m, r_1 = (327,3,500,11

Dan outputnya menjadi :

```
Columns 1 through 12

   11   100   203   384    71   220   443   364    31   140   283    44

Columns 13 through 21

  391   360   223   424   151   380   263     4   311
```

Gambar 3.4. Output dari program Matlab dengan a, c, m, r_1 = (327,3,500,11)

Jika program di atas dijalankan secara berulang, maka hasil yang didapat akan selalu sama. Untuk mengatasi hal tersebut maka nilai r_1 harus diubah dengan :

```
t=fix(clock);
r(1)=floor(mod(t(6),m));
```

sehingga programnya berubah menjadi :

```
% Mendefinisikan nilai pembangkit
a=327; c=3; m=500;
% Mendefinisikan nilai state awal
t=fix(clock);
        r(1)=floor(mod(t(6),m));
%Proses pembangkitan 20 bilangan acak
for k=1:20
   r(k+1)=mod(a*r(k)+c,m);
end
%Menampilkan bilangan acak
disp(r)
```

Gambar 3.5. Program Matlab dengan *t=fix(clock)* dan *r(1)=floor(mod(t(6),m))*

dan output pertama untuk program Matlab tersebut adalah :

Columns 1 through 12

55 488 79 336 375 128 359 396 495 368 339 356

Columns 13 through 21

415 208 19 216 135 148 399 476 155

Gambar 3.6. Output 1 dari Program Matlab dengan *t=fix(clock)* dan *r(1)=floor(mod(t(6),m))*

output berikutnya masih dari program yang sama adalah :

Columns 1 through 12

5 138 129 186 325 278 409 246 445 18 389 206

Columns 13 through 21

365 358 69 66 85 298 449 326 105

Gambar 3.7. Output 2 dari Program Matlab dengan *t=fix(clock)* dan *r(1)=floor(mod(t(6),m))*

Terlihat output untuk program di atas telah berbeda untuk pembangkitan bilangan acak tersebut. Maka untuk tugas akhir ini, dipilhlah jenis program yang ketiga untuk pembangkitan bilangan acak.

3.2 Implementasi Simulasi Monte Carlo pada Pengambilan Keputusan

Pada permasalahan pengambilan keputusan, alternatif yang menjadi pilihan dapat ditentukan dengan mencari nilai harapan yang terbesar dari alternatif yang ada. Tetapi dalam proses pengambilan keputusan disajikan dalam diagram keputusan dengan nilai *outcome* yang tidak dapat ditentukan secara pasti, nilai *outcome*nya ditentukan terlebih dahulu dengan simulasi Monte Carlo.

Nilai *outcome* dapat ditentukan dengan mencari rata-rata dari hasil simulasi. Dalamsimulasi Monte Carlo, nilai hasil simulasi itu bisa berupa nilai rata-rata, nilai modus, median dan sebagainya. Namun pada tugas akhir ini digunakan nilai rata-rata sebagai nilai *outcome*nya.

Setelah dilakukan simulasi sebanyak jumlah yang telah ditentukan diperoleh nilai rata-rata *outcome*nya. Nilai rata-rata ini yang akan digunakan sebagai nilai *outcome* yang digunakan untuk proses perhitungan nilai harapan. Nilai harapan ditentukan dengan melakukan perhitungan seperti biasa, yaitu jumlah dari perkalian probabilitas masing-masing keadaan tidak pastinya dengan nilai *outcome* yang telah ditentukan.