BAB III
METODE SIMULASI

3.1 Metode Simulasi

3.1.1 Pengertian

Untuk merumuskan model stokastik pada sebuah sistem yang kompleks, perlu adanya pertimbangan yang baik dalam menentukan model tiruan sistem nyata dan analisis matematika mana yang dapat dikerjakan. Oleh karena itu, tidak akan ada hasil apapun yang diperoleh dalam memilih model yang sangat sesuai dengan sistem yang diteliti jika model tersebut tidak dapat dianalisis secara matematis. Dewasa ini, metode yang digunakan dalam memilih model yang bersesuian dengan sistem nyata dengan teknik analisis matematis yang mumpuni adalah simulasi. Dalam kacamata statistikawan, simulasi merupakan metode kuantitatif dengan kunci utamanya adalah keacakan, di mana dalam menganalisis suatu sistem, pendekatannya menggunakan sebuah algoritma. (Thompson, 2000:1)

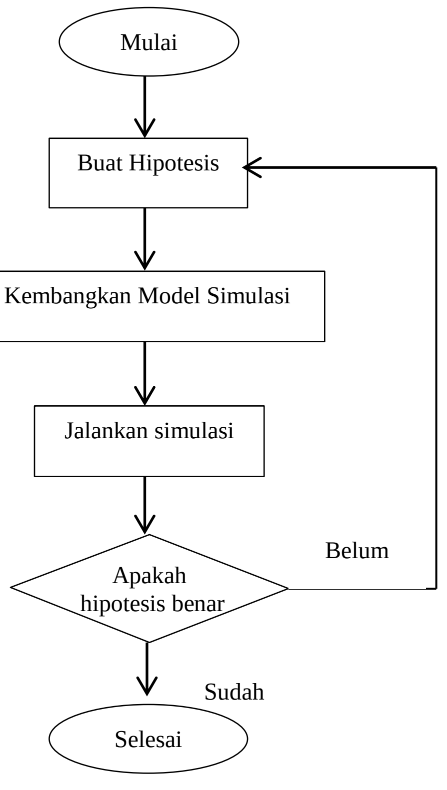
Selain itu, Harrel dkk (2004:5) mengatakan bahwa simulasi didefinisikan sebagai sebuah sistem dinamik yang menggunakan model komputer dengan tujuan untuk mengevaluasi dan meningkatkan kinerja sistem.

Selanjutnya, Harrel (2004:6) mengutip dari Schriber (1987) dengan mengungkapkan bahwa simulasi adalah pemodelan sebuah proses atau sistem sedemikian rupa model yang dibentuk dapat menyerupai bentuk sistem nyata berdasarkan kejadian-kejadian yang berlangsung dari waktu ke waktu.

Simulasi hampir selalu dilakukan sebagai bagian dari proses yang lebih besar dari desain sistem atau pengembangannya. Simulasi pada dasarnya merupakan sebuah alat eksperimen di mana model komputer dari sistem baru atau dari yang sudah ada dibuat dengan tujuan melakukan sebuah penelitian.

Prosedur melakukan simulasi mengikuti tahapan-tahapan dari metode ilmiah, antara lain : (1) susun sebuah hipotesis (2) atur sebuah penelitian (3) uji hipotesisnya dan (4) ambil kesimpulan tentang validitas hipotesis.

Dalam simulasi, sebuah hipotesis disusun untuk menentukan desain atau aturan operasi manakah yang paling bagus. Kemudian sebuah penelitian dilakukan dalam bentuk model simulasi untuk menguji hipotesis. Dan pada tahap akhir, hasil simulasi dianalisis dan selanjutnya kesimpulan dibuat berdasarkan kepada penelitian yang sudah dilakukan. Jika hipotesis benar maka desain sistem dapat dilanjutkan untuk membuat desain perubahan operasi. Proses melakukan peneltian dengan simulasi dapat dilihat pada gambar berikut ini:



Gambar 3.1 Proses Penelitian dengan Metode Simulasi

3.1.2 Karakteristik Simulasi

Simulasi merupakan metode dengan menghindari teknik trial dan proses mahal yang memakan waktu. Dengan menggunakan bantuan komputer untuk memodelkan, maka model imitasi dapat menyerupai sistem nyata.

Kekuatan simulasi terletak dalam hal menyediakan metode analisis yang tidak hanya bersifat formal dan prediktif, tapi juga dapat memprediksi secara akurat bagaimana kinerja sebuah sistem yang kompleks sekalipun.

Adapun karakteristik simulasi yang menjadikannya sebagai alat pembuat keputusan yang kuat dapat dijelaskan di bawah ini:

1. Mengidentifikasi ketergantungan dalam sistem
2. Bersifat fleksibel untuk jenis sistem manapun
3. Menunjukkan perilaku terhadap waktu
4. Tidak memakan banyak biaya, banyak waktu dan error yang lebih besar dibandingkan dengan metode kuantitatif lain
5. Menghasilkan informasi pada pengukuran beberapa kinerja sistem
6. Menghasilkan analisis yang mudah dipahami dan dijelaskan

3.2 Langkah-Langkah Simulasi

Berdasarkan Harrel (2004), tahapan-tahapan simulasi yang harus dilakukan adalah sebagai berikut:

1. Pengidentifikasian sistem dan entitas-entitasnya
2. Pengumpulan Data
3. Identifikasi Jenis-jenis data

Menurut Harrel dkk (2004:128), untuk membangun sebuah model, hal pertama kali yang harus dilakukan adalah menentukan jenis data yang dikumpulkan. Adapun jenis-jenis data dapat dikategorikan sebagai berikut:

a. Data Struktural

Data struktural terdiri dari semua elemen sistem yang dimodelkan. Termasuk ke dalamnya yaitu entitas, sumber daya dan lokasi. Pada sistem pelayanan permintaan darah di BDRS RSUP Dr. Hasan Sadikin

Bandung entitas, sumber daya dan lokasi masing-masing adalah komponen darah, UTD PMI dan bank darah itu sendiri.

b. Data Operasional

Adapun data operasional menjelaskan tentang bagaimana sebuah sistem bekerja. Seperti kapan, di mana dan bagaimana kegiatan dapat berlangsung. Data operasional mengandung semua informasi tentang sistem seperti *routings* (urutan perencanaan), penjadwalan, penghentian waktu kerja serta alokasi sumber daya.

c. Data Numerik

Data numerik ini memberikan informasi kuantitatif mengenai sistem. Yang termasuk ke dalam data numerik diantaranya:

1. Data banyaknya persediaan *packed red cells*, *fresh frozem plasma*, *whole blood* dan *trombosit concentrate* dari masing-masing golongan darah
2. Data banyaknya permintaan *packed red cells* dan *trombosit concentrate* dari masing-masing golongan darah Data waktu kadatangan permintaan dari dokter untuk masing-masing komponen darah
3. Data banyaknya komponen darah yang kadaluarsa dari masing-masing golongan darah
4. Data banyaknya komponen darah dari masing-masing golongan darah yang diuji *crossmatch*
5. Data banyaknya komponen darah dari masing-masing golongan darah yang diuji *crossmatch* tetapi tidak dipakai dalam kegiatan transfusi bulan pada tahun 2014

4. Identifikasi sumber data

Data yang baik dapat diperoleh dari sumber-sumber seperti berikut ini:

a. Data historis Bank Darah RSUP Dr. Hasan Sadikin Bandung

b. Dokumentasi sistem seperti rencana proses, prosedur kerja;

c. Observasi personal;
d. Perbandingan dengan sistem yang serupa;
e. Estimasi desain seperti waktu proses;
f. Studi pustaka.

5. Definisikan aliran entitas

Mendefinisikan aliran ensitas di sini adalah menggambarkan diagram aliran entitas pada sebuah sistem. Diagram aliran entitas sebagaimana yang dikemukakan Harrel, dkk (2004:131) sedikit berbeda dengan *flowchart* pada umumnya. *Flowchart* lebih menekankan apa yang terjadi pada entitas dengan menunjukkan logika *what if* yang paling sesuai dengan entitas itu sendiri. Sedangkan diagram aliran entitas ini hanya menjelaskan di mana entitas singgah dari tempat satu ke tempat lainnya pada sebuah sistem. Akan tetapi untuk sistem yang lebih kompleks, aliran entitas akan lebih menyerupai *flowchart*.

6. Kembangkan deskripsi operasi

Setelah menggambarkan diagram aliran entitas pada sebuah sistem, akan lebih baik jika ditambahkan dengan penjelasan lebih lanjut mengenai waktu kedatangan entitas, waktu pendistribusian entitas dan sebagainya.

7. Buat asumsi

Peneliti membuat asumsi sebagai salah satu langkah dalam simulasi dengan tujuan agar model yang terbentuk bisa lebih valid dibandingkan dengan metode lain. Katsaliaki,dkk (2007) menyebutkan bahwa banyaknya permintaan darah yang datang dari dokter mempunyai model yang berbentuk Proses Poisson Nonhomogen. Oleh karena itu, penulis mengasumsikan bahwa kedatangan permintaan darah di Bank Darah RSUP Dr. Hasan Sadikin Bandung merupakan proses Poisson Non-homogen. Yang artinya, proses tersebut bergantung pada waktu.

Asumsi yang telah dibuat ini selanjutnya akan divalidasi dengan menggunakan metode selain metode *Goodness-of-it* dan Kolmogorov-Smirnov. Karena menurut Ross (2004:237), metode pendekatan ini lebih efektif dibandingkan dengan dua metode lainnya. Metode pendekatan ini didasarkan pada fakta bahwa rerata dan varians dari distribusi Poisson sama ($\mu = \sigma^2 = \lambda$).

Misalkan $N_i, i = 1, \ldots, r$ menotasikan banyaknya permintaan darah yang datang di Bank Darah RSUP dr. Hasan Sadikin Bandung pada hari ke-i dan jika proses permintaan darah merupakan Proses Poisson Non-Homogen maka jumlah banyaknya permintaan darah merupakan variabel-variabel acak berdistribusi Poisson dengan rerata yang sama. Oleh sebab itu, jika N_i merupakan sebuah sampel dari distribusi Poisson, rerata sampelnya adalah

$$\bar{N} = \sum_{i=1}^{r} \frac{N_i}{r} \tag{3.1}$$

dan variansi sampelnya adalah

$$S^2 = \sum_{i=1}^{r} \frac{(N_i - \bar{N})^2}{r - 1} \tag{3.2}$$

haruslah sama. Untuk itu, hipotesis yang diuji adalah

$H_0 = N_i$ adalah variabel-variabel acak berdistribusi Poisson yang independen dengan rerata yang sama pada uji

$$T = \frac{S^2}{\bar{N}} \tag{3.3}$$

nilai p-value untuk $T = t$ akan menjadi,

$$p - value = 2\, minimum \left(P_{H_0}\{T \leq t, P_{H_0} \geq t\}\right) \tag{3.4}$$

Hal tersebut dikarenakan nilai T yang sangat kecil maupun sangat besar akan tidak konsisten dengan H_0.

Di sini H_0 tidak memberikan gambaran jelas mengenai rerata dari distribusi Poisson, oleh karenanya persamaan (3.4) tidak dapat langsung dihitung dan yang harus dilakukan adalah mengestimasi rerata dari data yang diobservasi. Dengan menggunakan estimator $\bar{N}$, dan jika nilai $\bar{N}$ terobservasi adalah $\bar{N} = m$, nilai p-value dapat dihampiri dengan

$$p - value \approx 2\ Minimum\ (P_m\{T \leq t\}, P_m\{T \geq t\})$$

di mana T didefinisikan oleh (3.3) dengan $N_1, \dots, N_r$ adalah variabel-variabel acak berdistribusi Poisson dengan rerata m. $P_m\{T \leq t\}$ dan $P_m\{T \geq t\}$ dapat dihampiri dengan simulasi. Maka, variabel-variabel acak berdistribusi Poisson dan rerata m akan dibangkitkan dan nilai T akan dihitung. Dengan $T \leq t$ merupakan estimasi dari $P\{T \leq t\}$ dan $\{T \geq t\}$ merupakan estimasi dari $P\{T \geq t\}$.

Jika nilai p-value nya kecil, maka H_0 akan ditolak. Sedangkan jika sebaliknya, maka H_0 diterima yang berarti banyaknya permintaan darah berdistribusi Poisson dan asumsi yang telah dibuat merupakan asumsi yang dapat digunakan. Agar asumsi yang dibuat lebih valid, misalkan waktu kedatangan permintaan darah pada hari $j, j = 1, \dots, r$ adalah $X_{j.1}, \dots, X_{j.N_j}$. Jika benar waktu kedatangan permintaan darah merupakan proses Poisson Nonhomogen, maka hal tersebut dapat ditunjukkan dari masing-masing r himpunan waktu kedatangan permintaan darah yang mana merupakan sampel dari distribusi yang sama. Dengan H_0 diterima, maka semua r himpunan dari $X_{j.1}, \dots, X_{j.N_j}$ merupakan variabel-variabel acak berdistribusi identik dan independen.

Penjelasan di atas, dapat juga diuji dengan *multisample rank test.*

Yakni,

$$R = \frac{12}{N(N+1)} \sum_{j=1}^{r} \frac{\left(R_j - N_j \frac{(N+1)}{2}\right)^2}{N_j} \qquad (3.5)$$

dengan $N \equiv \sum_{j=i}^{r} N_j$ dan R_j merupakan penjumlahan *ranks* dari nilai-nilai data N_j. Ketika H_0 benar, R akan mempunyai distribusi Chi Kuadrat dengan derajat kebebasan $r - 1$. Oleh karena itu, jika nilai R yang terobservasi adalah $R = y$, nilai p $-$value dapat diaproksimasi dengan

$$p - value = 2Minimum\ (P_{H_0}\{R \le y\}, P_{H_0}\{R \ge y\})$$
$$\approx 2\ Minimum\ (P\{\chi^2_{r-1} \le y\}, 1 - P\{\chi^2_{r-1} \le y\})$$

di mana χ^2_{r-1} merupakan variabel acak berdistribusi Chi kuadrat dengan derajat kebebasan $r - 1$. Jika nilai p-value nya tidak terlalu kecil, maka asumsi dipenuhi.

8. Membangkitkan Bilangan Acak dengan Pembangkit Bilangan Acak

Misalkan $X = (X_1, X_2, \dots, X_n)$ adalah sebuah vektor acak yang memiliki fungsi kepadatan peluang $f(x_1, \dots, x_n)$ dan misalkan akan dihitung,

$$E[g(x)] = \iint \dots \int g(x_1, \dots, x_n) f(x_1, \dots, x_n) dx_1 dx_2 \dots dx_n$$

dengan g adalah fungsi n-dimensional.

Untuk menghampiri nilai $E[g(X)]$, hal pertama yang dilakukan adalah dengan membangkitkan sebuah vektor acak $X^{(1)} = (X_1^{(1)}, \dots, X_n^{(1)}$ dengan fungsi kepadatan peluang gabungan $f(x_1, \dots, x_n)$ dan kemudian hitung $Y^{(1)} = g(X^{(1)})$. Langkah selanjutnya yang dilakukan adalah membangkitkan vektor acak lainnya yakni $X^{(2)}$ dan hitung $Y^{(2)} = g(X^{(2)})$. Lakukan langkah ini sampai ke-r, di mana r merupakan bilangan tetap dari variabel acak yang i.i.d dengan $Y^{(i)} = g\left(X^{(i)}\right); i = 1, \dots, r$ telah dibangkitkan. Dengan menggunakan *strong law of large number* dapat diketahui bahwa

$$\lim_{r \to \infty} \frac{Y^{(1)} + \dots + Y^{(r)}}{r} = E\left[Y^{(i)}\right] = E[g(X)].$$

dan dari sini rerata dari nilai-nilai Y yang dibangkitkan merupakan estimasi dari $E[g(X)]$. Pendekatan untuk menaksir $E[g(X)]$ disebut pendekatan simulasi Monte Carlo.

Jelasnya, terdapat beberapa masalah tentang bagaiman membangkitkan vektor-vktor acak yang mempunyai distribusi gabungan tertentu. Metode simulasi memiliki kelebihan diantara metode kuantitatif lainnya seperti survey, eksperimen yang mana memungkinkan untuk meniru perilaku acak yang merupakan karakteristik sistem stokastik (Terzi dan Cavalieri, 2004: 4). Sebagai contohnya perilaku acak dalam sistem yang bersifat stokastik adalah waktu kedatangan entitas ke dalam sistem, banyaknya entitas yang diproses oleh sistem dan rerata waktu yang dibutuhkan untuk mengolah entitas dalam sistem.

Adapun cara untuk meniru perilaku acak adalah dengan menggunakan bilangan acak. Bilangan acak seperti yang dituturkan oleh Munir (2010) yakni suatu bilangan yang tidak dapat diprediksi.

Bilangan acak ini merupakan bilangan unik, berdistribusi seragam dengan nilai $0 \leq x \leq 1$ dan dibangkitkan oleh pembangkit bilangan acaknya (Harrel, dkk, 2004: 51).

Akan tetapi, perlu ditegaskan bahwa sebenarnya tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna. Karena bisa saja, bilangan acak yang dibangkitkan merupakan pengulangan dari yang sebelumnya. Oleh karena itu, pembangkit bilangan acak yang demikian disebut dengan *Pseudo-random number generator (PRNG)*.

Adapun salah satu jenis pembangkit bilangan acak dalam metode simulasi lain adalah *Linear Congruential Generator* (*LCG*). Pembangkit bilangan acak kongruen linier (*linear congruential generator* atau *LCG*) adalah *PRNG* yang berbentuk:

$$X_i = (aX_{i-1} + b) mod\ m \qquad (3.6)$$

di mana,

X_i = bilangan bulat acak ke-*i* dari deretnya

X_{I-1} = bilangan acak sebelumnya

a = faktor pengali

b = *increment*

m = modulus (Munir, 2009).

Baik untuk kasus diskrit maupun kasus kontinu, X_0 harus ditetapkan terlebih dahulu dengan cara memilih bilangan $0 \leq X_0 \leq m - 1$. Di mana X_0, a, b, m merupakan bilangan bulat nonnegatif. Nilai-nilai X_i, diperoleh dengan membagi $(aX_{i-1} + b)$ dengan m. Atau $X_i = (aX_{i-1} + b)$ mod m. Oleh karena itu nilai-nilai X_i terbatas di $0 \leq X_i \leq m - 1$ dan berdistribusi seragam. Untuk kasus yang kontinu, nilai-nilai nya pun akan berada diantara 0 dengan 1 yang dinotasikan dengan U_i. Oleh karena itu, nilai U_i diperoleh dengan membagi X_i oleh m.

Contoh: Misalkan $a = 21$, $b = 5$, dan $m = 16$. Dengan demikian,

$$Xi = (aXi - 1 + b) \, mod \, m$$

$$Xi = (21Xi - 1 + 5) \, mod \, (16)$$

Sebuah bilangan bulat X_0 dipilih secara sebarang dengan interval $0 \leq X_0 \leq m - 1$, $0 \leq X_0 \leq 15$. Bilangan bulat yang terpilih yakni $X_0 = 12$. Adapun bilangan-bilangan acak yang berhasil dibangkitkan dengan menggunakan metode LCG dapat dilihat pada tabel berikut ini:

Tabel 3.1

Bilangan Acak yang dibangkitkan dengan LCG

i	$(21X_{i-1} + 5)$ mod (16)	X_i	$U_i = \frac{X_i}{16}$
0		12	
1	257	1	0,0625
2	26	10	0,625
3	215	7	0,4375
4	152	8	0,5
5	173	13	0,8125
6	278	6	0,375
7	131	3	0,1875
8	68	4	0,25

9	89	9	0,5625
10	194	2	0,125
11	47	15	0,9375
12	320	0	0
13	5	5	0,3125
14	110	14	0,875
15	299	11	0,6875
16	236	12	0,75
17	257	1	0,0625
18	26	10	0,625
19	215	7	0,4375
20	152	8	0,5
21	173	13	0,8125
22	278	6	0,375
23	131	3	0,1875

Dengan memperhatikan tabel 3.1, dapat dilihat bahwa bilangan acak yang dihasilkan dengan menggunakan metode LCG, berulang pada $i = 17$. Hal tersebut dikarenakan adanya kendala komputasi. Tabel 3.1 menunjukkan bahwa metode LCG hanya akan membangkitkan deret bilangan acak sebanyak 16 bilangan dan nilai bilangan acak berulang kembali pada deret ke-17. Menurut Harrel, dkk (2004:53), disebutkan bahwa nilai bilangan acak ke-16 ini merupakan panjang siklus pembangkit bilangan acak. Yang untuk kasus di atas, panjang siklusnya dapat dikatakan pendek. Hal tersebut dikarenakan nilai yang dihasilkan pembangkit bilangan acak berulang pada deret ke-17. Sebagai contoh, untuk menjalankan sebuah simulasi dengan satu kali percobaan ulangan, setidaknya membutuhkan 1000 buah bilangan acak yang dibangkitkan. Dan jika ingin melakukan lima buah percobaan ulangan tentu saja, dibutuhkan 5000 bilangan acak yang dibangkitkan. Oleh karena itu, Harrel, dkk (2004:53) menambahkan bahwa panjang maksimum siklus metode

LCG yang dapat dicapai adalah m. Oleh karena itu a, b, m harus dipilih secara baik dan benar. Adapun langkah-langkahnya adalah sebagai berikut:

1. Tetapkan m , di mana $m = 2^c$, c merupakan bilangan bulat nonnegatif yang didasarkan pada banyaknya bit yang digunakan komputer. Komputer biasanya menggunakan 32 bit per kata. Oleh karena itu, $c = 31$.
2. b dan m harus memiliki faktor persekutuan terbesarnya 1.
3. $a = 1 + 4k$, di mana k merupakan sebuah bilangan bulat.

Pada dasarnya, terdapat dua macam LCG. Antara lain *Mixed Congruential Generator* dan *Mulplicative Congruential Generator. Mixed Congruential Generator* menggunakan $c > 0$. Sedangkan *Multiplicative Congruential Generator* menggunakan $c = 0$. Dan pada umumnya, software komputer menggunakan Multiplicative Congruential Generator dengan

$$X_i = (630360016X_{i-1}) mod\ (2^{31} - 1).$$

Sebelumnya telah dijelaskan mengenai proses pembangkitan bilangan acak yang berdistribusi seragam. Sedangkan metode untuk membangkitkan *random variates* yang berasal dari distribusi selain distribusi seragam $U(0,1)$ akan dijelaskan seperti berikut ini.

Sebagai contohnya, waktu kedatangan mobil ke restauran yang memiliki *drive-thru* diasumsikan berdistribusi eksponensial dan waktu yang diperlukan pengendara untuk sampai di jendela *drive thru* dapat diasumsikan berdistribusi log-normal. Nilai-nilai (*random variates*) dari ditribusi-distribusi seperti ini biasanya diperoleh dengan cara mentransformasikan *random variates*, yang terlebih dahulu dibangkitkan dengan pembangkit bilangan acak yang sesuai dengan distribusi. Salah satu metode umum yang digunakan dalam hal ini adalah metode invers.

Proposisi 3.1: Misalkan U berdistribusi seragam (0,1). Untuk sebarang fungsi distribusi F kontinu, variabel acak X didefinisikan sebagai:

$$X = F^{-1}(U) \quad (3.7)$$

Maka variabel acak X akan memiliki disrribusi F. Dengan $(F^{-1}(u)$ didefinisikan sama dengan nilai x untuk setiap $F(x) = u)$ (Ross, 2009:672).

Bukti 3.1:

$$F_x(a) = P(X \leq a) \quad (3.8)$$
$$= P(F^{-1}(U) \leq a)$$

Karena $F(x)$ merupakan fungsi monoton, $F^{-1}(U) \leq a$ berlaku jika dan hanya jika $U \leq F(a)$. Persamaan (3.3) menjadi:

$$F_x(a) = P(F^{-1}(U) \leq a)$$
$$= F(a) \quad \blacksquare$$

a. Membangkitkan Bilangan Acak Distribusi Diskrit (Distribusi Poisson)

Pada bagian ini, akan dijelaskan langkah-langkah untuk membangkitkan *random variates* yang diasumsikan mempunyai distribusi Poisson. Adapun langkah-langkah tersebut adalah sebuah algoritma sederhana yang mana algoritma ini digunakan juga oleh sebagian besar software simulasi.

Langkah 1: bangkitkan bilangan acak U

Langkah 2: $i = 0, p = e^{-\lambda}, F = p$

Langkah 3: Jika $U < F$, tetapkan $X = i$ dan berhenti

Langkah 4: $p = \frac{\lambda p}{i+1}, F = F + p; i = i + 1$

Langkah 5: kembali ke langkah 3

b. Pembangkit Bilangan Acak Distribusi Kontinu (Distribusi Eksponensial)

Berikut akan dijelaskan mengenai metode untuk membangkitkan *random variates* yang berasal dari distribusi selain distribusi seragam $U(0{,}1)$. Sebagai contohnya, waktu kedatangan mobil ke restauran yang memiliki *drive-thru* diasumsikan berdistribusi eksponensial dan waktu yang diperlukan pengendara untuk sampai di jendela *drive thru* dapat diasumsikan berdistribusi log-normal. Nilai-nilai (*random variates*) dari ditribusi-distribusi seperti ini biasanya diperoleh dengan cara mentransformasikan *random variates*, yang terlebih dahulu dibangkitkan dengan pembangkit bilangan acak yang sesuai dengan distribusi. Salah satu metode umum yang digunakan dalam hal ini adalah metode invers.

Jika diberikan fungsi densitas peluang $f(x)$, maka

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt \qquad (3.9)$$

dan dengan menetapkan

$$U = F(x)$$

di mana U berdistribusi seragam (0,1). x dapat dicari dengan metode invers.

$$x = F^{-1}(U)$$

untuk distribusi eksponensial,

diketahui:

$$f(x) = \begin{cases} \frac{1}{\beta} e^{-\frac{x}{\beta}} & ; x > 0 \\ 0 \; ; & x \; lainnya \end{cases}$$

maka,

$$F(x) = \begin{cases} 1 - e^{-\frac{x}{\beta}} & ; x > 0 \\ 0 \; ; & x \; lainnya \end{cases}$$

misalkan $U = F(x)$

$$U = 1 - e^{-\frac{x}{\beta}}$$

$$e^{-\frac{x}{\beta}} = 1 - U$$

$$\ln\left(e^{-\frac{x}{\beta}}\right) = \ln 1 - U$$

$$-\frac{x}{\beta} = \ln(1 - U)$$

$$x = -\beta \ln(1 - U)\,. \qquad (3.10)$$

Random variates yang dihasilkan dari metode invers, $x = -\beta \ln(1 - U)$ berdistribusi eksponensial dengan rerata β (Harrel dkk, 2004:56).

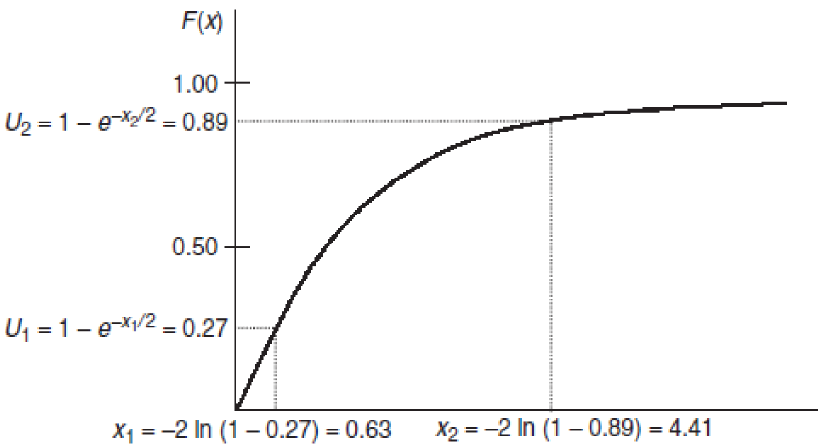
Contoh 2: Misalkan ada tiga buah observasi dengan variabel-variabel acaknya berdistribusi Eksponensial dan memiliki rerata $\beta = 2$. Tiga buah bilangan acak yang dibangkitkan antara lain $U_1 = 0{,}27$, $U_2 = 0{,}89$, dan $U_3 = 0{,}13$. Kemudian bilangan-bilangan acak tersebut ditransformasikan ke dalam bentuk random variates x_1, x_2, x_3. Dengan demikian diperoleh:

$$x_i = -\beta \ln(1 - U_i),\ i = 1{,}2{,}3$$

$$x_1 = -2\ln(1 - U_1) = -\ 2\ln(1 - 0{,}27) = -2(0{,}315) = 0{,}63$$

$$x_2 = -2\ln(1 - U_2) = -2\ln(1 - 0{,}89) = -2(-2{,}21) = 4{,}42$$

$$x_3 = -2\ln(1 - U_3) = -2\ln(1 - 0{,}13) = -2(0.139) = 0{,}28$$



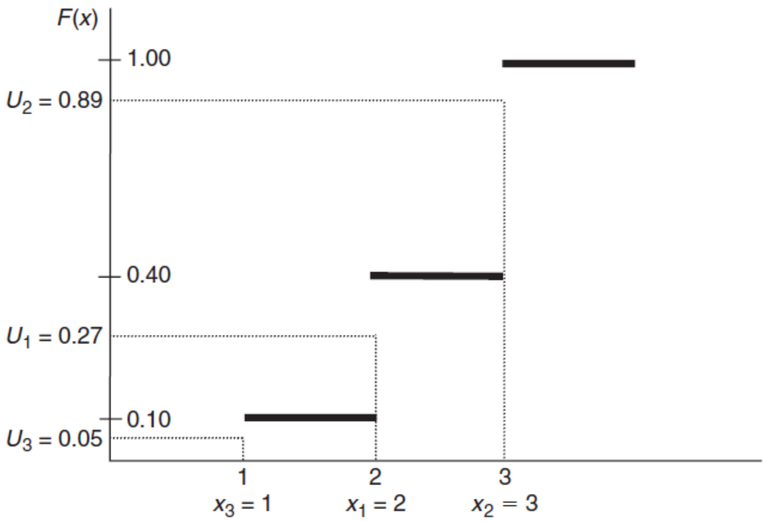
Gambar 3.2 Grafik Penerapan Metode Invers
untuk random variates dengan distribusi kontinu
(eksponensial)
(Harrel dkk, 2004:57)

Contoh 3: Pembangkit Bilangan Acak Distribusi Diskrit

Misalkan,

$$p(x) = P(X = x) = \begin{cases} 0.1 & x = 1 \\ 0.3 & x = 2 \\ 0.6 & x = 3 \end{cases}$$

Dengan demikian, random variate x memiliki tiga nilai kemungkinan, antara lain $P(X = 1) = 0.1$; $P(X = 2) = 0.3$; dan $P(X = 3) = 0.6$. Dengan diketahui nilai-nilai $U_1, U_2,$ dan U_3 masing-masing diberikan 0,05; 0,27 dan 0,89. Grafik fungsi distribusi kumulatif nya dapat dilihat pada gambar 3.3 berikut ini:



Gambar 3.3 Grafik Penerapan Metode Invers untuk random variates dengan distribusi diskrit (Harrel dkk, 2004:58)

Bilangan acak yang dibangkitkan haruslah independen, berdistribusi seragam dengan nilai berdistribusi seragam dengan nilai $0 \leq x \leq 1$. Agar pembangkit bilangan acak dapat menghasilkan(membangkitkan) bilangan dengan kriteria di atas, maka langkah-langkah yang harus dilakukan adalah sebagai berikut:

a. membangkitkan barisan bilangan acak $Z_1, Z_2, Z_3, \dots$ dengan pembangkit bilangan acak

b. kemudian uji hipotesis, di mana:

   H_0: nilai-nilai Z_i yang dihasilkan dari pembangkit bilangan acak bersifat independen

   H_1: nilai-nilai Z_i yang dihasilkan dari pembangkit bilangan acak bersifat independen

   dengan uji-uji statistik yang digunakan adalah scatter plot, runs test.

c. uji hipotesis selanjutnya yang dilakukan adalah

   H_0: nilai-nilai Z_i berdistribusi seragam (0,1)

   H_1: nilai-nilai Z_i tidak berdistribusi seragam (0,1)

Uji statistik yang digunakan dalam hal ini adalah Kolmogorov-Smirnov test, uji Chi-kuadrat.

c. Mensimulasikan Proses Poisson

Perhatikan sebuah proses Poisson $N(t)$ dengan intensitas λ, di mana

$$P(N(t) = r) = \frac{(\lambda t)^r e^{-\lambda t}}{r!}$$

dengan banyaknya kedatangan dalam interval-interval (tidak tumpang tindih) adalah independen dan distribusi dari banyaknya kedatangan dalam interval-interval hanya bergantung pada panjang interval itu sendiri.

Misalkan jika waktu antar-kedatangan ke- i yakni X_i didefinisikan sebagai interval di antara kedatangan ke-$(i-1)$ dan ke-i dari proses Poisson maka dapat dikatakan bahwa X_i ini akan berdistribusi $i.i.d \sim Exp(\lambda)$.

Dari sini, sebuah Proses Poisson dapat disimulasikan hanya dengan membangkitkan waktu-waktu antar kedatangan X_i, yang mana $X_i \sim Exp(\lambda)$.

Jika n bilangan-bilangan acak $U_1, U_2, \dots, U_n$ dibangkitkan dan ditetapkannya $X_i = -\frac{1}{\lambda}\log U_i$ maka X_i adalah interval di antara kedatangan ke-$(i-1)$ dan ke-i dari proses Poisson. Karena waktu sebenarnya dari peristiwa ke-j akan sama dengan penjumlahan j waktu-waktu antar kedatangan yang pertama, maka nilai-nilai yang dibangkitkan dari n waktu kejadian adalah

$$\sum_{i=1}^{j} X_i\,, j = 1, \dots, n \qquad (3.11)$$

Proses mensimulasikan unit-unit T waktu pertama dari Proses Poisson sama halnya dengan membangkitkan waktu antar kedatangan, hanya saja proses pembangkitan proses Poisson akan berhenti ketika waktu kedatangan pertama dari proses Poisson yakni (3.4) melebihi T.

Algoritma berikut ini akan digunakan untuk membangkitkan semua waktu-waktu dari peristiwa yang terjadi di $(0, T)$ pada proses Poisson yang mempunyai laju λ.

t = waktu

I= banyaknya peristiwa yang terjadi pada t

$S(I)$ = waktu peristiwa yang terakhir terjadi

Langkah 1: $t = 0, I = 0$
Langkah 2: bangkitkan bilangan acak U
Langkah 3: $t = t - \frac{1}{\lambda}\log U$, jika $t > T$ maka algoritma dihentikan
Langkah 4: $I = I + 1, S(I) = t$
Langkah 5: kembali ke langkah 2

(Ross, 2009:82)

d. Mensimulasikan Proses Poisson Non Homogen

Langkah-langkah untuk membangkitkan proses Poisson non-Homogen hampir sama dengan proses Poisson pada umumnya.

Misalkan, T unit waktu dari Proses Poisson Non homogen yang mempunyai fungsi laju/intensitas $\lambda(t)$ akan dibangkitkan. Hal pertama yang dilakukan adalah dengan memilih nilai laju λ sehingga

$$\lambda(t) \leq \lambda \text{ untuk semua } T$$

Seperti yang telah dijelaskan sebelumnya, proses Poisson Non Homogen bisa dibangkitkan dengan pemilihan waktu kejadian secara acak yang memiliki laju λ.

Dengan demikian, jika sebuah peristiwa dari Proses Poisson dengan laju λ terjadi pada saat t terhitung dan memiliki peluang $\frac{\lambda(t)}{\lambda}$, maka proses dari peristiwa-peristiwa yang terhitung adalah proses Poisson nonhomogen dengan laju $\lambda(t), 0 \leq t \leq T$. Singkatnya, proses Poisson nonhomogen dapat dibangkitkan dengan terlebih dahulu membangkitkan proses Poisson dan kemudian menghitung secara acak berapa banyak peristiwa yang terjadi.

Metode yang dijelaskan di atas disebut metode *thinning* atau pendekatan *random sampling*.

Langkah 1: $t = 0, I = 0$

Langkah 2: bangkitkan bilangan acak U

Langkah 3: $t = t - \frac{1}{\lambda}\log U$, jika $t > T$ maka algoritma dihentikan

Langkah 4: bangkitkan kembali bilangan acak U

Langkah 5: Jika $U \leq \lambda(t)/\lambda$ maka $I = I + 1, S(I) = t$

Langkah 6: kembali ke langkah 2

Ross (2009:84).

9. Uji Kecocokan Distribusi

Sampel data numerik yang telah dikumpulkan berupa waktu kegiatan, interval kedatangan, jumlah *batch*, dan sebagainya dapat diwakili dalam model simulasi dengan satu dari dua cara. Pertama, distribusi empiris dapat digunakan yang mencirikan data. Metode kedua yang digunakan adalah dengan memilih distribusi teoritis yang paling sesuai dengan data.

Uji kecocokan distribusi ini adalah proses trial (mencoba-coba) dan error. Prosedur dasarnya terdiri dari tiga tahapan yaitu (1) satu atau lebih distribusi dipilih sebagai salah satu kandidat untuk dijadikan distribusi yang paling sesuai dengan data sampel, (2) hitung estimasi parameter-parameter untuk setiap distribusi dan (3) kemudian, pengujian *goodness-of-fit* dilakukan untuk mengetahui dengan pasti sejauh mana kesesuaian distribusi dengan data yang ada.

Goodness-of-fit test dilakukan untuk menentukan kesesuaian distribusi dengan data, Tes ini dilakukan jika sebuah distribusi beserta parameter-parameternya telah didefiniskan. Harrel, dkk (2004:154) mengatakan bahwa:

A goodness-of-fit test measures the deviation of the sample distribution from the inferred theoretical distribution.

Uji *goodness-of-it* akan memberikan gambaran tentang deviasi atau penyimpangan data sampel dari distribusi teoritis yang ddduga. Uji *Goodness-of-fit* yang sering digunakan adalah Uji Chi Kuadrat dan Kolmogorov-Smirnov (see Breiman 1973; Banks et al. 2001; Law and Kelton 2000; Stuart and Ord 1991).

3.3 Model Simulasi

Model simulasi merupakan representasi dari output yang dihasilkan oleh komputer tentang bagaimana perilaku dan interaksi elemen-elemen dalam sistem.

3.4 Verifikasi model

Verifikasi merupakan proses penentuan apakah model tersebut beroperasi seperti yang dimaksudkan atau tidak (Harrel dkk, 2004:206). Selama proses verifikasi, pembuat model mencoba untuk mendeteksi kesalahan atau error dalam model data dan logikanya dan kemudian menghapusnya. Model yang telah diverifikasi disebut dengan model *bug-free.* Kesalahan dalam pemodelan simulasi dibagi menjadi dua macam, yakni:

1. Syntac error

   Kesalahan jenis ini merupakan kesalahan yang disebabkan oleh penambahan yang tidak disengaja, kesalahan atau penempatan lokasi yang menyebabkan model tidak berjalan dengan baik.
2. Semantic error

   Kesalahan ini merupakan jenis kesalahan yang berhubungan dengan maksud atau tujuan dari si pembuat model.

Teknik verifikasi model telah banyak dikembangkan oleh para peneliti. Berdasarkan kepada Law dan Kelton (2000), salah satunya adalah dengan membandingkan rerata dan varians hasil simulasi dengan data historisnya.

3.5 Validitas model

Validitas merupakan proses dalam menentukan apakah model dapat direpresentasikan seperti sistem nyata atau tidak (Hoover dan Perry, 1990). Sedangkan menurut Law dan Kelton (2000) validitas merupakan proses dalam menentukan apakah model simualsi merepresentasikan sistem secara akurat untuk tujuan tertentu atau tidak. Model dikatakan valid jika model tersebut dibangun berdasarkan informasi yang akurat dan diverifikasi seperti yang diharapkan (Harrel dkk, 2000). Adapun beberapa teknik yang dapat digunakan dalam validitas model adalah sebagai berikut:

1. Melihat animasi model

   Animasi tentang bagaimana visual model bekerja dapat dibandingkan dengan bagaimana sistem nyata berjalan dengan bantuan dari ahlinya.
2. Membandingkan dengan sistem nyata.

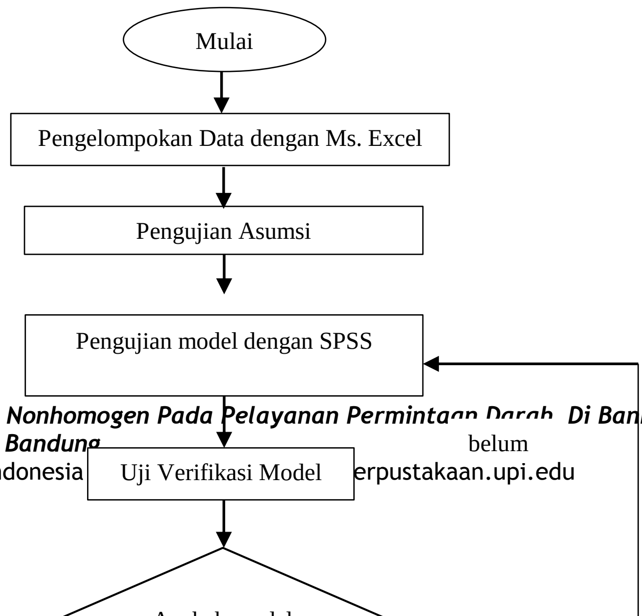
Validitas dilakukan dengan membandingkan model simulasi dengan sistem nyata dengan cara memberi input dan kondisi yang sama.

3. Membandingkan dengan model lain

   Jika model valid lain telah dibentuk dari proses seperti dari teknik analytics models, dan model simulasi valid lainnya, hasil simulasi dapat dibandingkan.

4. Memeriksa validitas
5. Menguji dengan data historis

   Jika terdapat data historis operasi dan kinerja sistem, maka model dapat diuji dengan menggunakan data historis kinerja sistem.



Gambar 3.4 Skema Alur Penelitian dengan Simulasi untuk Analisis Sistem Pelayanan Permintaan Darah