

## PEMBANGKITAN SAMPEL RANDOM MENGGUNAKAN ALGORITMA METROPOLIS-HASTINGS

Lies Kurnia Irwanti<sup>1</sup>, Moch. Abdul Mukid<sup>2</sup>, Rita Rahmawati<sup>3</sup>

<sup>1</sup>Mahasiswa Jurusan Statistika FSM Universitas Diponegoro

<sup>2,3</sup>Staf Pengajar Jurusan Statistika FSM UNDIP

### ABSTRACT

Generating random samples can be done directly and indirectly using simulation techniques. This final project will discuss the process of generating random samples and estimate the parameters using an indirect simulation. Indirect simulation techniques used if the target distribution has a complicated shape and high dimension of density functions. *Markov Chain Monte Carlo* (MCMC) simulation is a solution to do it. One of the algorithms that is commonly used is *Metropolis-Hastings*. This algorithm uses the mechanism of acceptance and rejection to generate a sequence of random samples. In the example to be discussed, Metropolis-Hastings algorithm is applied to generate random samples of Beta distribution and also estimate the parameter value of the Poisson distribution using a proposal distribution *random-walk Metropolis*.

**Keywords:** *Markov Chain Monte Carlo*, *Metropolis-Hastings* algorithm, proposal distribution

### 1. PENDAHULUAN

Sejarah perolehan sampel random cukup panjang dan menarik, mulai dari pelemparan dadu, pembagian kartu atau cara sejenis lainnya sampai penggunaan algoritma dengan bantuan *software* (perangkat lunak). Salah satu peran statistisi (ahli statistika) adalah mempelajari berbagai prosedur pengambilan keputusan. Untuk mendapatkan keputusan yang terbaik haruslah dilakukan uji validitas untuk prosedur atau metode statistika, salah satunya dengan mencobanya pada data yang parameternya diketahui. Namun, adakalanya bentuk distribusi posterior tidak dapat ditelusuri bentuk fungsi densitasnya karena memiliki dimensi tinggi, sehingga pendugaan parameternya dilakukan menggunakan algoritma *Markov Chain Monte Carlo* (MCMC). Pada algoritma MCMC ini distribusi posterior diketahui sebagai hasil kali antara fungsi likelihood (data sampel) dengan fungsi prior (informasi awal). Oleh karena itu, perlu dikembangkan teknik yang bertujuan membangkitkan data dengan sifat-sifat atau parameter yang diinginkan.

Simulasi sebagai salah satu teknik pembangkitan data mengalami perkembangan yang cukup pesat seiring dengan perkembangan dan penggunaan komputer selama beberapa dekade yang lalu. Salah satu metode simulasi tidak langsung, yaitu algoritma Metropolis-Hastings. Algoritma Metropolis-Hastings menggunakan mekanisme penerimaan dan penolakan (*accept-reject*) untuk membangkitkan barisan sampel dari suatu distribusi yang sulit untuk dilakukan penarikan sampel secara langsung. Penarikan sampel menggunakan algoritma Metropolis-Hastings dapat dilakukan pada distribusi probabilitas  $\pi(x)$  apapun dengan syarat bahwa densitas  $X = x$  dapat dihitung. Sampel random yang dibangkitkan menggunakan algoritma ini banyak digunakan misalnya pada teori antrian mengenai waktu antar kedatangan dan analisis finansial mengenai estimasi volatilitas dan estimasi *return* dalam konteks Bayesian yang distribusi posteriornya rumit. Dalam tulisan ini akan dijelaskan penggunaan algoritma Metropolis-Hastings untuk membangkitkan sampel random berdistribusi beta dan mengestimasi parameter distribusi poisson menggunakan metode Bayes.

### 2. TINJAUAN PUSTAKA

Tinjauan pustaka yang digunakan dalam tulisan ini adalah sebagai berikut:

#### 2.1. Variabel Random

Variabel merupakan karakteristik suatu objek yang diamati. Dalam sampel dapat dibuat beberapa variabel sesuai karakteristik objek yang diamati. Pengamatan terhadap objek bersifat random, oleh karenanya disebut variabel random.

### Definisi 2.1.1

Variabel random ( $X$ ) adalah fungsi yang bernilai real yang didefinisikan pada ruang sampel  $S$ .  $X$  dapat dituliskan sebagai  $X(c) = x$ , dengan  $c$  merupakan setiap hasil yang mungkin pada  $S$ .

(Bain dan Engelhardt, 1992)

### Definisi 2.1.2

Variabel random  $X$  disebut variabel random diskrit jika himpunan semua nilai yang mungkin muncul dari  $X$  merupakan himpunan terhingga (*countable*).

(Bain dan Engelhardt, 1992)

### Definisi 2.1.3

Variabel random  $X$  disebut variabel random kontinu jika nilai  $X$  dapat mencakup semua nilai dari suatu interval atau dikatakan himpunan semua nilai yang mungkin muncul dari  $X$  dan merupakan himpunan tak terhingga (*uncountable*).

(Bain dan Engelhardt, 1992)

## 2.2. Fungsi Likelihood

Fungsi densitas bersama (*joint density function*) dari  $n$  variabel random  $X_1, X_2, \dots, X_n$  yang dinyatakan dalam bentuk  $f(x_1, x_2, \dots, x_n | \theta)$  disebut sebagai fungsi likelihood. Jika  $x_1, x_2, \dots, x_n$  tetap, fungsi likelihood adalah fungsi dari  $\theta$  dan sering dinotasikan sebagai  $L(\theta)$ . Jika  $x_1, x_2, \dots, x_n$  mewakili sampel random dari  $f(x|\theta)$  maka:

$$L(\theta) = f(x_1|\theta)f(x_2|\theta)\dots f(x_n|\theta)$$

(Bain dan Engelhardt, 1992)

## 2.3. Distribusi Prior

Distribusi prior adalah distribusi awal yang harus ditentukan terlebih dahulu sebelum merumuskan distribusi posteriornya.

### Definisi 2.3.1

Misal dalam mengestimasi parameter  $\theta$  tersedia informasi bahwa  $\theta$  berubah sesuai dengan distribusi peluang tertentu. Jadi,  $\theta$  adalah variabel random yang menjalani harga-harga  $\theta_1, \theta_2, \theta_3, \dots, \theta_k$  dengan distribusi peluang  $f(\theta_1), \dots, f(\theta_k)$ .  $f(\tilde{\theta} = \theta_j)$ ,  $j = 1, 2, \dots, k$  disebut distribusi prior untuk  $\tilde{\theta}$ , dimana  $\tilde{\theta} = (\theta_1, \theta_2, \dots, \theta_k)$ .

(Soejoeti dan Soebanar, 1988)

Menurut Box dan Tiao (1973) distribusi prior dikelompokkan menjadi dua kelompok berdasarkan bentuk fungsi likelihoodnya, yaitu:

1. Berkaitan dengan bentuk distribusi hasil identifikasi pola datanya dapat dibedakan menjadi:
  - a. Distribusi prior sekawan (*conjugate*), merupakan pemberian bentuk distribusi prior yang sekawan atau sepola dengan bentuk distribusi dari hasil identifikasi pola datanya.
  - b. Distribusi prior tidak sekawan (*non-conjugate*), merupakan pemberian bentuk distribusi prior yang tidak sekawan dengan bentuk hasil identifikasi dari datanya.
2. Berkaitan dengan penentuan masing-masing parameter pada pola distribusi prior tersebut, dapat dibedakan menjadi dua bentuk yang berbeda, yaitu:
  - a. Distribusi prior informatif mengacu pada pemberian parameter dari distribusi prior yang telah dipilih baik distribusi prior sekawan atau tidak, pemberian nilai parameter pada distribusi prior ini didasarkan pada informasi yang diperoleh dari datanya.
  - b. Distribusi prior non-informatif, jika penentuan parameter distribusi prior tidak didasarkan pada data yang ada.

### Definisi 2.3.2

Salah satu bentuk pendekatan dari prior non-informatif adalah dengan menggunakan metode Jeffrey. Metode ini menyatakan bahwa distribusi prior  $f(\theta)$  merupakan akar kuadrat dari informasi Fisher yang dinyatakan dalam:

$$f(\theta) = [I(\theta)]^{1/2} \quad (1)$$

dimana  $I(\theta)$  merupakan nilai harapan informasi Fisher.

$$I(\theta) = -E_{\theta} \left[ \frac{\partial^2 \ln f(x|\theta)}{\partial^2 \theta} \right] \quad (2)$$

(Berger, 1980)

### 2.4. Distribusi Posterior

Distribusi posterior adalah fungsi densitas bersyarat  $\theta$  jika diketahui nilai observasi  $x$  dan dapat dituliskan sebagai berikut:

$$f(\theta|x) = \frac{f(\theta, x)}{f(x)} \quad (3)$$

Persamaan (3) menunjukkan bahwa fungsi densitas bersyarat satu variabel random ( $\theta$ ) jika diketahui nilai variabel random kedua ( $x$ ) adalah fungsi densitas bersama dua variabel random itu ( $f(\theta, x)$ ) dibagi dengan fungsi densitas marginal variabel random kedua. Tetapi fungsi densitas bersama  $f(\theta, x)$  dan fungsi densitas marginal  $f(x)$  pada umumnya tidak diketahui, hanya distribusi prior dan fungsi likelihood yang biasanya dinyatakan.

Fungsi densitas bersama dan marginal yang diperlukan dapat ditulis dalam bentuk distribusi prior dan fungsi likelihood:

$$f(\theta, x) = f(x|\theta) \cdot f(\theta) \quad (4)$$

dimana  $f(x|\theta)$  merupakan fungsi likelihood dan  $f(\theta)$  merupakan distribusi prior. Selanjutnya diketahui bahwa:

$$f(x) = \int_{-\infty}^{\infty} f(\theta, x) d\theta = \int_{-\infty}^{\infty} f(\theta) f(x|\theta) d\theta \quad (5)$$

Sehingga dari persamaan (3), (4) dan (5), fungsi densitas posterior untuk variabel random kontinu dapat ditulis sebagai berikut:

$$f(\theta|x) = \frac{f(\theta) f(x|\theta)}{\int_{-\infty}^{\infty} f(\theta) f(x|\theta) d\theta}$$

(Soejoeti dan Soebanar, 1988)

### 2.5. Metode Bayesian

Pada metode Bayesian, inferensi didasarkan pada distribusi posterior  $f_{\theta|x}(\theta)$ , yang dapat juga dituliskan sebagai  $\pi(\theta|x_1, \dots, x_n)$ . Dari persamaan (5) maka:

$$\pi(\theta|x_1, \dots, x_n) = \frac{f(x_1, \dots, x_n|\theta) \pi(\theta)}{\int_{-\infty}^{\infty} f(x_1, \dots, x_n|\theta) \pi(\theta) d\theta}$$

$$\pi(\theta|x_1, \dots, x_n) = \frac{f(x_1, \dots, x_n|\theta)\pi(\theta)}{m(x_1, \dots, x_n)}$$

Faktor penyebut, yaitu  $m(x_1, \dots, x_n)$  merupakan fungsi likelihood marginal dari data, sehingga dapat dirumuskan bahwa:

$$m(x_1, \dots, x_n) = \int_{-\infty}^{\infty} f(x_1, \dots, x_n|\theta)\pi(\theta)d\theta$$

Dalam metode Bayesian dikenal suatu faktor kesebandingan untuk menentukan distribusi posterior  $\pi(\theta|x_1, \dots, x_n)$ , yaitu:

$$\pi(\theta|x_1, \dots, x_n) \propto f(x_1, \dots, x_n|\theta)\pi(\theta) \quad (6)$$

Lambang  $\propto$  menyatakan sifat proporsional atau sebanding. Pada perspektif Bayesian fungsi likelihood merupakan fungsi dari  $\theta$  pada data  $x_1, \dots, x_n$  sehingga mengakibatkan elemen-elemen likelihood yang tidak mengandung fungsi  $\theta$  menjadi bagian dari kesebandingan pada persamaan (6). Dengan kata lain, melalui sifat kesebandingan diperoleh bahwa densitas posterior hanya mengandung fungsi yang memuat  $\theta$ .

Penduga Bayes untuk  $\theta$  diperoleh melalui nilai harapan dari  $(\theta|x_1, \dots, x_n)$  dapat ditulis sebagai:

$$\hat{\theta} = E(\theta|x_1, \dots, x_n)$$

(Congdon, 2003)

## 2.6. Metode *Markov Chain Monte Carlo* (MCMC)

*Markov Chain Monte Carlo* (MCMC) adalah sebuah rangkaian metode untuk menciptakan barisan sampel random yang berasal dari distribusi peluang, dengan membangun rantai Markov sesuai dengan distribusi tertentu yang diinginkan. Metode *Markov Chain Monte Carlo* dewasa ini telah banyak diaplikasikan di berbagai bidang untuk menyelesaikan bermacam-macam permasalahan. Permasalahan yang dibahas khususnya yang terkait dengan inferensi Bayesian, dimana seringkali ditemui persoalan dalam mendapatkan suatu distribusi posterior dan juga distribusi prior pada beberapa studi kasus. Algoritma yang sering menggunakan metode MCMC, yaitu *Metropolis-Hastings* dan *Gibbs Sampling*. Dalam tulisan ini hanya akan membahas algoritma *Metropolis-Hastings* (Walsh, 2004).

### 2.6.1. Algoritma *Metropolis-Hastings* (M-H)

Pada awalnya Metropolis dkk. memformulasikan algoritma *Metropolis* dengan mengenalkan rantai Markov berdasarkan metode-metode simulasi yang digunakan dalam ilmu pengetahuan alam. Kemudian Hastings menggeneralisasi metode tersebut yang terkenal dengan nama algoritma *Metropolis-Hastings*. Akhirnya, dipertimbangkan menjadi formula umum untuk semua metode MCMC. Green lebih lanjut menggeneralisasi algoritma *Metropolis-Hastings* dengan mengenalkan algoritma *Metropolis-Hastings reversible jump* untuk sampling dari ruang parameter dengan dimensi yang berbeda (Ntzoufras, 2009).

Asumsikan distribusi targetnya adalah  $f(\mathbf{x})$  yang mana sampel yang diinginkan akan dibangkitkan dengan ukuran  $T$ . Algoritma *Metropolis-Hastings* dapat dijelaskan dengan mengikuti langkah-langkah iterasi, dimana  $\mathbf{x}^{(t)}$  adalah vektor dari nilai-nilai yang dibangkitkan di iterasi ke- $t$  dari algoritma:

1. Atur nilai awal  $\mathbf{x}^{(0)}$ .
2. Untuk  $t = 1, \dots, T$  lakukan langkah-langkah berikut:
  - a. Mengatur nilai  $\mathbf{x} = \mathbf{x}^{(t-1)}$
  - b. Membangkitkan calon nilai yang baru  $\mathbf{x}'$  dari distribusi proposal  $q(\mathbf{x} \rightarrow \mathbf{x}') = q(\mathbf{x}'|\mathbf{x})$ .

- c. Menghitung  $\alpha = \min\left(1, \frac{f(\mathbf{x}')q(\mathbf{x}|\mathbf{x}')}{f(\mathbf{x})q(\mathbf{x}'|\mathbf{x})}\right)$  dan membangkitkan sampel random  $u \sim U(0,1)$ .
- d. Memperbarui  $\mathbf{x}^{(t)} = \mathbf{x}'$  dengan peluang penerimaan  $\alpha$  dan  $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$  dengan peluang  $1 - \alpha$ . Jika  $u \leq \alpha$  maka  $\mathbf{x}'$  diterima sebagai anggota sampel dan jika  $u > \alpha$  maka nilai sebelumnya ( $\mathbf{x}$ ) yang diterima sebagai anggota sampel.

Algoritma Metropolis-Hastings akan konvergen ke distribusi target tanpa memperhatikan apapun distribusi proposal  $q$  yang dipilih.

Bagan algoritma tersebut dapat diimplementasikan secara langsung pada kerangka Bayesian dengan mensubstitusikan  $\mathbf{x}$  dengan parameter  $\boldsymbol{\theta}$  dan distribusi target  $f(\mathbf{x})$  dengan distribusi posterior  $f(\boldsymbol{\theta}|\mathbf{x})$ . Pada inferensi Bayesian, ringkasan algoritmanya adalah sebagai berikut:

1. Mengatur nilai awal  $\boldsymbol{\theta}^{(0)}$
2. Untuk  $t = 1, \dots, T$  lakukan langkah berikut ini:
  - a. Mengatur  $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t-1)}$ .
  - b. Membangkitkan nilai calon parameter  $\boldsymbol{\theta}'$  dari distribusi proposal  $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ .
  - c. Menghitung  $\alpha = \min\left(1, \frac{f(\boldsymbol{\theta}'|\mathbf{x})q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{f(\boldsymbol{\theta}|\mathbf{x})q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right)$  dan membangkitkan sampel random  $u \sim U(0,1)$ .
  - d. Memperbarui  $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}'$  dengan peluang  $\alpha$  dan  $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$  dengan peluang  $1 - \alpha$ . Jika  $u \leq \alpha$  maka  $\boldsymbol{\theta}'$  diterima dan jika  $u > \alpha$  nilai sebelumnya ( $\boldsymbol{\theta}$ ) yang diterima sebagai sampel.

### 2.6.2. Distribusi Proposal *Random-walk Metropolis*

Dalam algoritma Metropolis yang asli, hanya distribusi proposal yang simetris dengan tipe  $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}|\mathbf{x}')$  yang diperhatikan. *Random-walk Metropolis* merupakan kasus khusus dengan  $q(\mathbf{x}'|\mathbf{x}) = q(|\mathbf{x}' - \mathbf{x}|)$ . Kedua kasus menghasilkan probabilitas penerimaan yang hanya tergantung pada distribusi target, yaitu:

$$\alpha = \min\left(1, \frac{f(\mathbf{x}'|\boldsymbol{\theta})}{f(\mathbf{x}|\boldsymbol{\theta})}\right) \quad (7)$$

Biasanya distribusi proposal ini adalah normal multivariat  $q(\mathbf{x}'|\mathbf{x}) \sim N_d(\mathbf{x}, \mathbf{S}_x)$  dengan  $d$  adalah dimensi dari  $\mathbf{x}$ . Selain distribusi normal multivariat, distribusi proposal yang dapat digunakan adalah uniform  $q(x'|x) \sim U(x - \delta, x + \delta)$ ,  $\delta$  merupakan sebuah konstanta yang ditetapkan peneliti dimana nilai  $\delta \leq 1$ .

### 2.6.3. Konvergensi Algoritma

Ada beberapa cara untuk memantau konvergensi algoritma. Metode yang sering digunakan, yaitu:

#### 1. Trace Plot

Salah satu cara pendugaan *burn-in period* adalah dengan memeriksa *trace plot* nilai simulasi dari komponen (atau beberapa fungsi lainnya)  $\mathbf{x}$  terhadap jumlah iterasi. *Trace plot* merupakan gambaran sebuah plot dari iterasi versus nilai yang telah dibangkitkan. Sebuah tren naik atau turun pada nilai parameter pada *trace plot* menunjukkan bahwa *burn-in period* belum selesai. Jika tren-tren seperti ini muncul, maka penting untuk menghilangkan bagian awal dari rantai, karena nilai-nilai awal ini tidak menunjukkan perkiraan sampel yang benar dari distribusi target. Dengan kata lain, jika semua nilai-nilai berada dalam sebuah daerah tanpa keperiodikan yang kuat dan (secara khusus) kecenderungan, maka dapat diasumsikan telah tercapai konvergensi.

## 2. Autokorelasi

Perhatian yang kedua pada analisis output algoritma MCMC adalah tingkat autokorelasi pada nilai-nilai sampel. Untuk kedua algoritma Metropolis-Hastings dan Gibbs sampling, nilai simulasi  $\mathbf{x}$  pada iterasi ke- $(t+1)$  bergantung pada nilai simulasi pada iterasi ke- $t$ . Jika pada rantai terdapat korelasi yang kuat di antara nilai-nilai yang beruntun, maka kedua nilai beruntun tersebut memberikan informasi hanya secara marginal mengenai distribusi target dan bukan nilai dari sebuah simulasi tunggal. Korelasi yang kuat di antara iterasi yang beruntun menunjukkan bahwa algoritma masih berada pada daerah tertentu dari ruang parameter dan mungkin membutuhkan waktu yang lama untuk penyampelan dari keseluruhan daerah distribusi.

Statistik yang umum digunakan untuk mengukur tingkat ketergantungan diantara pengambilan berurutan pada rantai adalah autokorelasi. Autokorelasi mengukur korelasi diantara kumpulan nilai-nilai simulasi  $\{x_i^{(t)}\}$  dan  $\{x_i^{(t+L)}\}$ , dimana  $L$  merupakan jumlah selisih sampling dari iterasi terpisah pada dua kumpulan nilai-nilai. Untuk komponen tertentu, fungsi autokorelasi dapat dihitung sebagai fungsi nilai-nilai yang berbeda dari selisih sampling. Untuk komponen  $i$ , autokorelasi  $L$  dapat diduga dengan:

$$r_{iL} = \frac{T' - L}{T' - 1} \frac{\sum_{i=1}^{T'-L} (x_i - \bar{x})(x_{i+L} - \bar{x})}{\sum_{i=1}^{T'} (x_i - \bar{x})^2}$$

dimana  $\bar{x}$  merupakan rata-rata dari nilai-nilai simulasi. Nilai autokorelasi untuk selisih sampling 1 akan hampir selalu menjadi positif untuk algoritma M-H dan Gibbs sampling. Bagaimanapun, jika rantai sedang bercampur dengan cukup, nilai-nilai autokorelasi akan berkurang menuju nol selama nilai selisih sampling ditingkatkan.

## 3. Ergodic Mean Plot

Istilah *ergodic mean* menunjukkan nilai mean sampai dengan *current iteration*. *Ergodic mean plot* adalah plot antara nilai mean di atas dengan iterasinya. Jika *ergodic mean* stabil setelah beberapa iterasi, maka ini merupakan sebuah indikasi konvergensi dari algoritma.

### 2.6.4. Pendugaan Parameter

Pendugaan parameter biasanya terjadi untuk kasus-kasus inferensi Bayesian. Misalkan  $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_p)$  adalah sebuah vektor parameter yang akan diduga nilainya. Dengan menjalankan sebuah algoritma MCMC, nilai-nilai simulasi  $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(T')}$  masing-masing terdistribusi secara kira-kira ke distribusi posterior  $f(\boldsymbol{\theta}|\mathbf{x})$ .

Penduga dari parameter  $\theta_i$  diperoleh dari nilai rata-rata dari nilai-nilai sampel yang tersimulasi, yaitu:

$$\hat{\theta}_i = \frac{\sum_{t=1}^{T'} \theta_i^{(t)}}{T'} \quad i = 1, 2, \dots, p \quad (8)$$

dengan  $\theta_i^{(t)}$  adalah nilai-nilai tersimulasi.

Setelah diperoleh dugaan parameter, perhitungan penting lainnya pada analisis output adalah mengenai *standard error*. Salah satu metode yang sederhana dan mudah diterapkan untuk menghitung *standard error* yaitu metode *batch mean*.

Metode *batch mean* dilakukan dengan membagi lagi urutan nilai-nilai simulasi  $\theta_i^{(1)}, \dots, \theta_i^{(T')}$  menjadi  $h$  kelompok yang berukuran  $w$ , sehingga  $T' = h \cdot w$ . Untuk setiap kelompok dihitung rata-rata sampel, misal rata-rata kelompok sampel adalah  $\bar{\theta}_i^{(1)}, \dots, \bar{\theta}_i^{(h)}$ . Misalkan bahwa, ukuran sampel  $w$  telah dipilih cukup besar sehingga autokorelasi pada rangkaian *batch means* kecil, katakan di bawah 0,1 maka estimasi *standard error*  $\hat{\theta}_i$  dapat diduga dengan standard deviasi dari *batch means*, yaitu:

$$S_{\hat{\theta}_i}^B = \sqrt{\frac{\sum_{l=1}^h (\hat{\theta}_i^l - \bar{\theta}_i)^2}{(h-1)h}} \quad (9)$$

*Standard error* ini sangat berguna untuk menentukan ketelitian dari rata-rata distribusi target yang dihitung pada simulasi yang dijalankan. Pada kejadian tersebut, jika *standard error* terlalu besar, maka algoritma MCMC sebaiknya dijalankan kembali menggunakan jumlah iterasi yang lebih besar.

(Johnson dan Albert, 1999)

### 3. HASIL DAN PEMBAHASAN

Dua contoh kasus sebagai ilustrasi penerapan algoritma M-H yang efektif akan dibahas berikut ini. Contoh pertama diambil kasus umum univariat dan contoh kedua diambil kasus Bayesian univariat.

#### 3.1. Penerapan Algoritma M-H untuk Pembangkitan Sampel Random Berdistribusi Beta

Pada contoh ini akan dibangkitkan sebuah sampel random dari populasi yang berdistribusi  $Beta(a,b)$  dengan fungsi densitas, sebagai berikut:

$$f(x|a,b) = \frac{1}{B(a,b)} x^{a-1} (1-x)^{b-1}, \quad 0 < x < 1 \quad (10)$$

Pembangkitan sampel random menggunakan algoritma Metropolis-Hastings perlu menggunakan sebuah distribusi proposal. Distribusi proposal yang digunakan adalah normal dengan fungsi densitas, berikut ini:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Bentuk fungsi densitas distribusi target berdasarkan persamaan (10) untuk  $f(x) \sim Beta(2,6)$  adalah:

$$\begin{aligned} f(x|2,6) &= \frac{1}{B(2,6)} x^{2-1} (1-x)^{6-1} \\ &= \frac{1}{\Gamma(2)\Gamma(6)} x \cdot (1-x)^5 \\ &= \frac{1}{1!5!} x \cdot (1-x)^5 \\ &= \frac{1}{7!} x \cdot (1-x)^5 \\ f(x|2,6) &= \frac{1}{42} x \cdot (1-x)^5 \end{aligned}$$

$$f(x) = 42x \cdot (1-x)^5$$

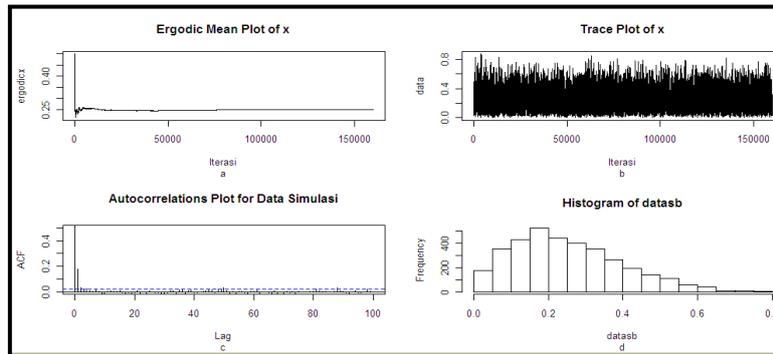
Probabilitas penerimaan sesuai persamaan (7), yaitu:

$$\alpha = \min\left(1, \frac{f(x')}{f(x)}\right)$$

$$\alpha = \min\left(1, \frac{(42x' \cdot (1-x')^5)}{(42x \cdot (1-x)^5)}\right)$$

$$\alpha = \min\left(1, \frac{x' \cdot (1-x')^5}{x \cdot (1-x)^5}\right)$$

Pembangkitan sampel random dijalankan dengan bantuan *software* R menggunakan nilai awal  $x^{(0)} = 0,5$ , selisih sampling 20 dan iterasi sebanyak 160000. Berikut ini analisis konvergensi dari *output* yang dihasilkan:



**Gambar 1** (a) *Ergodic mean plot* dari sampel yang dibangkitkan, (b) *Trace plot* dari sampel yang dibangkitkan, (c) Plot autokorelasi dari sampel yang dibangkitkan, (d) Histogram dari sampel yang dibangkitkan

Pada Gambar 1 terlihat bahwa *ergodic mean plot* sampel yang dibangkitkan telah stabil mulai iterasi ke-90000. Pada *trace plot*-nya terlihat tidak membentuk suatu pola yang menunjukkan proses *burn-in* telah selesai. Berikutnya pada plot autokorelasi terlihat bahwa nilai-nilai autokorelasi pada lag pertama mendekati satu dan untuk selanjutnya berkurang menuju 0. Berdasarkan ketiga macam plot tersebut diketahui bahwa algoritma telah konvergen. Gambar terakhir merupakan histogram dari sampel yang dibangkitkan mulai dari iterasi 90001 sampai 160000. Sampel random setelah konvergen tersebut diuji menggunakan uji Kolmogorov-Smirnov dan diperoleh nilai statistik ujinya  $D=0,0132$ . Jadi, sampel random yang dibangkitkan berasal dari distribusi target, yaitu distribusi beta.

### 3.2. Penerapan Algoritma M-H untuk Pendugaan Parameter pada Kasus Bayesian

Untuk mengilustrasikan penggunaan algoritma Metropolis-Hastings dalam pendugaan parameter sebuah distribusi pada kasus Bayesian, maka pada contoh ini akan diduga parameter dari populasi jumlah bus jurusan Jepara yang masuk terminal Terboyo, Semarang. Populasi tersebut diketahui berdistribusi  $Poisson(\mu)$  yang memiliki rata-rata 4,55 dan ukuran sampel yang digunakan 29. Fungsi densitas distribusi poisson, sebagai berikut:

$$f(x|\mu) = \frac{e^{-\mu} \mu^x}{x!} \quad x = 0,1,2,\dots \text{ dan } \mu > 0$$

Dalam inferensi Bayesian, perlu diketahui fungsi likelihood sampel dan distribusi prior dari parameternya. Fungsi likelihood untuk distribusi poisson, adalah:

$$\begin{aligned} f(x_1, \dots, x_n | \mu) &= \frac{e^{-\mu} \cdot \mu^{x_1}}{x_1!} \cdot \frac{e^{-\mu} \cdot \mu^{x_2}}{x_2!} \cdot \dots \cdot \frac{e^{-\mu} \cdot \mu^{x_n}}{x_n!} \\ &= \prod_{i=1}^n \frac{e^{-\mu} \cdot \mu^{x_i}}{x_i!} \end{aligned}$$

Jika  $y = \sum_{i=1}^n x_i$ , maka fungsi likelihoodnya:

$$f(y|\mu) = \frac{e^{-n\mu} \cdot \mu^y}{\prod_{i=1}^n x_i!}$$

Selanjutnya, distribusi prior dari  $\mu$  dapat ditentukan menggunakan prior konjugat dan prior non konjugat. Prior non konjugat dapat diperoleh dengan menggunakan metode Jeffrey.

#### 3.2.1. Prior Konjugat untuk Distribusi Poisson

Prior konjugat merupakan distribusi prior informatif untuk memberikan nilai parameter yang didasarkan pada informasi yang diperoleh dari datanya. Prior konjugat dari distribusi poisson adalah distribusi gamma, yaitu:

$$f(\mu; r, v) = \frac{v^r}{\Gamma(r)} \mu^{r-1} \cdot e^{-v\mu} \quad ; r > 0, v > 0$$

Distribusi posterior untuk  $\mu$  diperoleh sebagai berikut:

$$f(\mu|y) = \frac{f(\mu; r, v) \cdot f(y|\mu)}{\int_0^{\infty} f(\mu; r, v) \cdot f(y|\mu) d\mu}$$

$$f(\mu|x_1, \dots, x_n) = \frac{(v+n)^{r+\sum x_i}}{\Gamma(\sum x_i + r)} \cdot \mu^{r-1+\sum x_i} \cdot e^{-(v+n)\mu} \quad ; \mu > 0 \quad (13)$$

Distribusi proposal yang digunakan dalam contoh kasus ini yaitu *random-walk Metropolis* dengan distribusi  $U(\mu - \delta, \mu + \delta)$  dimana  $\delta = 0,1$ . Prior konjugat yang digunakan yaitu

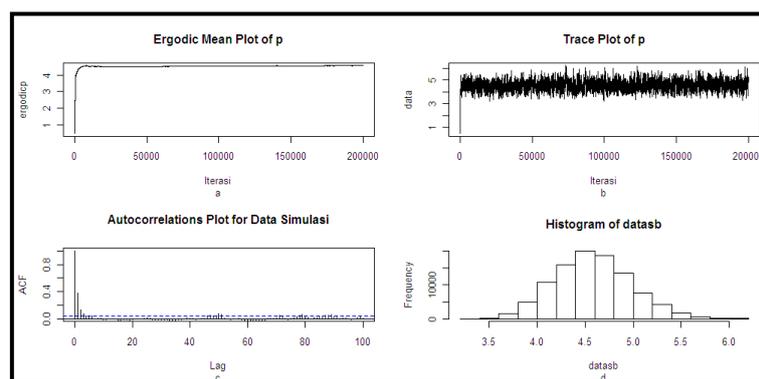
$\text{Gamma}\left(1, \frac{1}{2}\right)$ . Probabilitas penerimaan algoritmanya berdasarkan persamaan (7), yaitu:

$$\alpha = \min\left(1, \frac{f(\mu'|x_1, \dots, x_n)}{f(\mu|x_1, \dots, x_n)}\right)$$

$$\alpha = \min\left(1, \frac{\frac{(v+n)^{r+\sum x_i}}{\Gamma(\sum x_i + r)} \cdot \mu'^{r-1+\sum x_i} \cdot e^{-(v+n)\mu'}}{\frac{(v+n)^{r+\sum x_i}}{\Gamma(\sum x_i + r)} \cdot \mu^{r-1+\sum x_i} \cdot e^{-(v+n)\mu}}\right)$$

$$\alpha = \min\left(1, \frac{\mu'^{r-1+\sum x_i} \cdot e^{-(v+n)\mu'}}{\mu^{r-1+\sum x_i} \cdot e^{-(v+n)\mu}}\right)$$

Pembangkitan sampel random dijalankan dengan bantuan *software* R menggunakan nilai awal  $x^{(0)} = 0,5$ , selisih sampling 100 dan iterasi sebanyak 200000. Berikut ini analisis konvergensi dari *output* yang dihasilkan:



**Gambar 2** (a) *Ergodic mean plot* dari sampel yang dibangkitkan, (b) *Trace plot* dari sampel yang dibangkitkan, (c) Plot autokorelasi dari sampel yang dibangkitkan, (d) Histogram dari sampel yang dibangkitkan

Pada Gambar 2 terlihat bahwa *ergodic mean plot* sampel yang dibangkitkan telah stabil mulai iterasi ke-100000. Pada *trace plot*-nya terlihat tidak membentuk suatu pola yang menunjukkan proses *burn-in* telah selesai. Berikutnya pada plot autokorelasi terlihat bahwa nilai-nilai autokorelasi pada lag pertama mendekati satu dan untuk selanjutnya berkurang menuju 0. Berdasarkan ketiga macam plot

tersebut diketahui bahwa algoritma telah konvergen. Gambar terakhir merupakan histogram dari sampel yang dibangkitkan mulai dari iterasi 100001 sampai 200000.

Penduga Bayes untuk parameter  $\mu$  dari distribusi poisson  $\hat{\mu} = 4,5211$  dan secara teoritis nilai mean dari distribusi gamma  $E(\mu|x_1, \dots, x_n) = 4,5085$ . Jadi, nilai dugaan  $\mu$  secara komputasi dan teoritis diperoleh hasil yang hampir sama dengan nilai *standard error* yang dihasilkan sebesar 0,0198.

### 3.2.2. Prior Jeffrey untuk Distribusi Poisson

Prior Jeffrey merupakan salah satu bentuk pendekatan dari prior non-informatif. Langkah-langkah menentukan prior untuk  $\mu$ , sebagai berikut:

$$f(x|\mu) = \frac{e^{-\mu} \mu^x}{x!}$$

$$\ln f(x|\mu) = -\mu + x \ln \mu - \ln x!$$

$$f'(x|\mu) = \frac{\partial \ln f(x|\mu)}{\partial \mu} = -1 + \frac{x}{\mu}$$

$$f''(x|\mu) = \frac{\partial^2 \ln f(x|\mu)}{\partial^2 \mu} = -\frac{x}{\mu^2}$$

selanjutnya nilai harapan informasi Fisher dihitung berdasarkan persamaan (2), sebagai berikut:

$$\begin{aligned} I(\mu) &= -E\left(\frac{\partial^2 \ln f(x|\mu)}{\partial^2 \mu}\right) \\ &= \frac{1}{\mu} \end{aligned}$$

Berdasarkan persamaan (1) prior Jeffrey untuk distribusi poisson, yaitu:

$$\begin{aligned} \pi(\mu) &\propto [I(\mu)]^{\frac{1}{2}} \\ &\propto \frac{1}{\sqrt{\mu}} \end{aligned}$$

Distribusi posterior untuk  $\mu$  diperoleh dari hasil kali antara fungsi likelihood dengan distribusi priornya dan dapat dituliskan sebagai berikut:

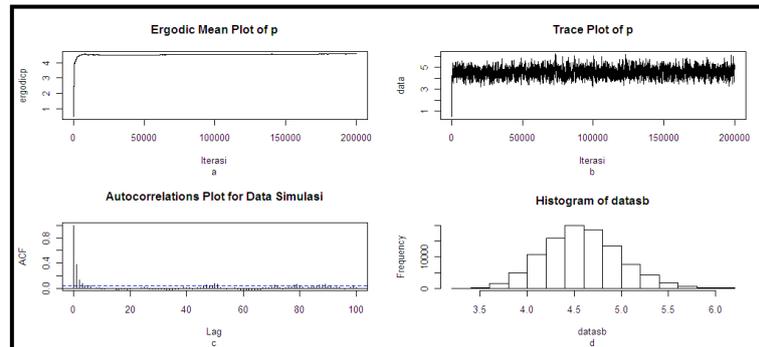
$$f(\mu|x_1, \dots, x_n) \propto e^{-n\mu} \cdot \mu^{\sum x_i} \cdot \frac{1}{\sqrt{\mu}}$$

$$f(\mu|x_1, \dots, x_n) \propto e^{-n\mu} \cdot \mu^{\sum x_i - \frac{1}{2}} \quad (14)$$

Distribusi proposal yang digunakan dalam contoh kasus ini yaitu *random-walk Metropolis* dengan distribusi  $U(\mu - \delta, \mu + \delta)$  dimana  $\delta = 0,1$ . Probabilitas penerimaan algoritmanya berdasarkan persamaan (7), yaitu:

$$\begin{aligned} \alpha &= \min\left(1, \frac{f(\mu'|x_1, \dots, x_n)}{f(\mu|x_1, \dots, x_n)}\right) \\ \alpha &= \min\left(1, \frac{e^{-n\mu'} \cdot \mu'^{\sum x_i - \frac{1}{2}}}{e^{-n\mu} \cdot \mu^{\sum x_i - \frac{1}{2}}}\right) \end{aligned}$$

Pembangkitan sampel random dijalankan dengan bantuan *software* R menggunakan nilai awal  $x^{(0)} = 0,5$ , selisih sampling 100 dan iterasi sebanyak 200000. Berikut ini analisis konvergensi dari *output* yang dihasilkan:



**Gambar 3** (a) *Ergodic mean plot* dari sampel yang dibangkitkan, (b) *Trace plot* dari sampel yang dibangkitkan, (c) Plot autokorelasi dari sampel yang dibangkitkan, (d) Histogram dari sampel yang dibangkitkan

Pada Gambar 3 terlihat bahwa *ergodic mean plot* sampel yang dibangkitkan telah stabil mulai iterasi ke-100000. Pada *trace plot*-nya terlihat tidak membentuk suatu pola yang menunjukkan proses *burn-in* telah selesai. Berikutnya pada plot autokorelasi terlihat bahwa nilai-nilai autokorelasi pada lag pertama mendekati satu dan untuk selanjutnya berkurang menuju 0. Berdasarkan ketiga macam plot tersebut diketahui bahwa algoritma telah konvergen. Gambar terakhir merupakan histogram dari sampel yang dibangkitkan mulai dari iterasi 100001 sampai 200000. Penduga Bayes untuk parameter  $\mu$  dari distribusi poisson  $\hat{\mu} = 4,5812$  dengan nilai *standard error* yang dihasilkan sebesar 0,0208.

#### 4. KESIMPULAN

Algoritma Metropolis-Hastings merupakan salah satu algoritma dari metode *Markov Chain Monte Carlo* (MCMC) yang sering digunakan. Dalam penggunaan algoritma Metropolis-Hastings diperlukan sebuah distribusi proposal yang dapat dipilih salah satunya menggunakan *random-walk Metropolis*. Sebagai ilustrasi penggunaan algoritma M-H, diberikan contoh pembangkitan sampel random yang berdistribusi beta dan pendugaan parameter untuk distribusi poisson.

Pembangkitan sampel random yang berdistribusi beta dijalankan menggunakan jumlah iterasi 160000. Berdasarkan analisis *output* yang dilakukan diketahui bahwa sampel random yang dihasilkan sesuai dengan distribusi targetnya. Pada contoh pendugaan parameter dari distribusi poisson dengan inferensi Bayesian digunakan prior konjugat dan prior Jeffrey. Hasil dugaan  $\mu$  yang menggunakan prior konjugat yaitu 4,5211 dengan *standard error* sebesar 0,0198 dan yang menggunakan prior Jeffrey yaitu 4,5812 dengan *standard error* 0,0208

#### DAFTAR PUSTAKA

- Bain, L. J. dan Engelhardt, M. 1992. *Introduction to Probability and Mathematical Statistics Second Edition*. Duxbury Press: California.
- Berger, J.O. 1980. *Statistical Decision Theory and Bayesian Analysis Second Edition*. Springer-Verlag inc.: New York.
- Box, G.E.P and Tiao, G.C. 1973. *Bayesian Inference In Statistical Analysis*. Addison-Wesley Publishing Company, Inc: Philippines.
- Congdon, P. 2003. *Bayesian Statistical Modelling*. Chichester, UK: John Wiley.
- Johnson, V. E. dan Albert, J. H. 1999. *Ordinal Data Modeling*. Springer-Verlag Inc.: New York.
- Ntzoufras, I. 2009. *Bayesian Modeling Using WinBUGS*. John Wiley & Sons, Inc: New Jersey.
- Nurrستاني, P.R. 2008. Peningkatan Pelayanan di Terminal Terboyo Semarang dengan Teori Antrian. Undip: Semarang.
- Soejoeti, Z. dan Soebanar.1988. *Inferensi Bayesian*. Karunika Universitas Terbuka: Jakarta.
- Walsh, B. 2004. Markov Chain Monte Carlo and Gibbs Sampling. Lecture Notes for EEB 581. [www.stat.columbia.edu](http://www.stat.columbia.edu) [diakses pada 8 Februari 2012]

