Penaksiran Jumlah Kerusakan dalam Laptop Menggunakan Model Reliabilitas *Software* Hipergeometrik

Achmad Syarif Widyanto*, Aceng Komaruding Mutaqin

Prodi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Bandung, Indonesia.

*boocah84@gmail.com, aceng.k.mutaqin@gmail.com

Abstract. The probabilistic nature of software reliability modeling views that the testing of a software that is carried out to determine the amount of damage in the software is a random experiment. One of the models used to model the observations from the random experiment is the model hypergeometric software reliability (Padberg, 2002). In the reliability model hypergeometric software, software testing is a gradual process and done several times. A damage is said to be detected only if the damage has not been detected in previous tests. Each test result is modeled using a hypergeometric probability distribution that is depending on the number of m crashes in the software and the number of ck-1 different defects detected in the previous k-1 test. The hypergeometric software reliability model does not assume that faults are detected repaired instantly. Software reliability models often use the maximum likelihood estimation method to calculate the estimated number of defects (m) based on test data. In a hypergeometric software reliability model, the probability function contains many factorials. One way to calculate the maximum likelihood estimate of hypergeometric reliability model parameters by using the growth quotient of the probability function (Padberg, 2002). The data used for this study is primary data about the amount of damage found on the MSI GL62M laptop. The results show that the estimated number of damage to the operating system and software contained in the MSI GL62M laptop is 4,105 damage.

Keywords: Maximum Likelihood Estimation, Hypergeometric Software Reliability Model, Growth Quotient.

Abstrak. Pemodelan reliabilitas software yang sifatnya probabilistik memandang bahwa pengujian suatu software yang dilakukan untuk mengetahui jumlah kerusakan dalam software tersebut sebagai suatu eksperimen acak. Salah satu model yang digunakan untuk memodelkan hasil pengamatan dari eksperimen acak tersebut adalah model reliabilitas software hipergeometrik (Padberg, 2002). Dalam model reliabilitas software hipergeometrik, pengujian software adalah suatu proses yang bertahap dan dilakukan beberapa kali. Suatu kerusakan dikatakan baru terdeteksi apabila kerusakan tersebut belum terdeteksi pada pengujian-pengujian sebelumnya. Masing-masing hasil pengujian dimodelkan dengan menggunakan distribusi peluang hipergeometrik yang tergantung pada jumlah m kerusakan yang terdapat pada software dan jumlah ck-1 kerusakan berbeda yang terdeteksi pada k-1 pengujian sebelumnya. Model reliabilitas software hipergeometrik tidak mengasumsikan bahwa kerusakan yang terdeteksi diperbaiki seketika. Model reliabilitas software sering menggunakan metode penaksiran kemungkinan maksimum untuk menghitung nilai taksiran jumlah kerusakan (m) berdasarkan pada data hasil pengujian. Dalam model reliabilitas software hipergeometrik, fungsi kemungkinannya mengandung banyak sekali faktorial. Salah satu cara untuk menghitung taksiran kemungkinan maksimum dari parameter model reliabilitas hipergeometrik yaitu dengan menggunakan hasil bagi pertumbuhan dari fungsi kemungkinan (Padberg, 2002). Data yang digunakan untuk penelitian ini merupakan data primer tentang banyaknya kerusakan yang terdapat pada laptop MSI GL62M. Hasilnya menunjukan bahwa taksiran banyaknya kerusakan pada sistem operasi dan software yang terdapat pada laptop MSI GL62M sebanyak 4.105 kerusakan.

Kata Kunci: Penaksiran Kemungkinan Maksimum, Model Reliabilitas Software Hipergeometrik, Hasil Bagi Pertumbuhan.

1. Pendahuluan

Banyaknya kerusakan yang terdapat dalam software adalah suatu atribut kualitas yang tidak bisa ditentukan secara tepat (Padberg, 2002). Oleh karena itu, dalam praktek banyaknya kerusakan tersebut harus ditaksir. Pemodelan reliabilitas software yang sifatnya probabilistik memandang bahwa pengujian suatu software yang dilakukan untuk mengetahui jumlah kerusakan dalam software tersebut sebagai suatu eksperimen acak. Salah satu model yang digunakan untuk memodelkan hasil pengamatan dari eksperimen acak tersebut adalah model reliabilitas software hipergeometrik (Padberg, 2002).

Salah satu cara untuk menghitung taksiran kemungkinan maksimum dari parameter model reliabilitas hipergeometrik yaitu dengan menggunakan hasil bagi pertumbuhan dari fungsi kemungkinan (Padberg, 2002). Cara ini sangat sederhana dan cepat dalam memperoleh nilai taksirannya.

Berdasarkan latar belakang yang telah diuraikanm maka perumusan masalah dalam penelitian ini sebagai berikut: “Salah satu cara untuk menghitung taksiran kemungkinan maksimum dari parameter model reliabilitas hipergeometrik yaitu dengan menggunakan hasil bagi pertumbuhan dari fungsi kemungkinan (Padberg, 2002). Cara ini sangat sederhana dan cepat dalam memperoleh nilai taksirannya”. Selanjutnya, tujuan dalam penelitian ini yaitu untuk mengetahui penerapan metode penaksiran kemungkinan maksimum untuk model reliabilitas software hipergeometrik menggunakan hasil bagi pertumbuhan untuk menaksir banyaknya kerusakan pada laptop MSI GL62M menggunakan aplikasi *Auslogics Registry Cleaner*.

2. Metodologi

Distribusi Hipergeometrik

Pengujian banyaknya kerusakan dalam sistem menggunakan software ke-k terdiri dari jumlah wk kerusakan yang terdeteksi dan mengandung jumlah xk kerusakan yang baru terdeteksi. Pertimbangkan hasil pengujian ke-k sebagai jumlah kumulatif kerusakan berbeda yang terdeteksi dalam k pengujian pertama yaitu,

$$ck= x1+ x2+\cdots+ xk \qquad ...(2.1)$$

Hasil dari pengujian ke-k di atas dapat dimodelkan oleh distribusi hipergeometrik berikut,

$$P_{m,}w_k\left(c_{k,}c_{k-1}\right) = \frac{\left(\frac{m- c_{k-1}}{c_k- c_{k-1}}\right).\left(\frac{c_{k-1}}{w_k-(c_k-c_{k-1})}\right)}{\left(\frac{m}{w_k}\right)} \qquad ...(2.2)$$

Untuk $\boldsymbol{c_k \leq m, c_k - c_{k-1} \leq w_k}$, dan $\boldsymbol{w_k \leq c_k}$. Parameter $\boldsymbol{m}$ menunjukan jumlah kerusakan yang terdapat pada *sofware.*

Hasil Bagi Pertumbuhan

Didefinisikan hasil bagi pertumbuhan dari fungsi kemungkinan maksimum sebagai berikut,

$$Q(m) = \frac{L(m)}{L(m-1)} \qquad ...(2.5)$$

Hasil bagi pertumbuhan $Q(m)$ lebih besar dari satu ketika fungsi kemungkinan $L(m)$ menaik dan sebaliknya hasil bagi pertumbuhan $Q(m) \leq 1$ ketika fungsi kemungkinan $L(m)$ menurun. Hasil bagi pertumbuhan dapat diuraikan sebagai berikut,

$$Q(m) = \frac{(m-w_1)\cdot(m-w_2)\cdot\ldots\cdot(m-w_n)}{m^{n-1}\cdot(m-c_n)} \qquad ...(2.6)$$

Di bawah ini adalah beberapa hasil penting (untuk n≥2) yang diperoleh dari Padberg (2002) berkaitan dengan nilai taksiran kemungkinan maksimum dari parameter model reliabilitas software hipergeometrik:

1. Kasus A. Jika $\boldsymbol{c_n = w_k}$ untuk beberapa $\boldsymbol{k}$ dan $\boldsymbol{w_j > 0}$ untuk paling sedikit satu $\boldsymbol{j \neq k}$, maka penaksir kemungkinan maksimum $\boldsymbol{\widehat{m} = c_n}$
2. Kasus B. Jika $\boldsymbol{c_n = w_k}$ untuk beberapa $\boldsymbol{k}$ dan $\boldsymbol{w_j = 0}$ untuk semua $\boldsymbol{j \neq k}$, maka setiap nilai yang lebih besar sama dengan $\boldsymbol{c_n}$ adalah suatu penaksir kemungkinan maksimum dari $\boldsymbol{m}$.
3. Kasus C. Misalkan bahwa $\boldsymbol{c_n > w_j}$ untuk semua $\boldsymbol{j}$, jika $\boldsymbol{w_k = c_k - c_{k-1}}$ untuk semua $\boldsymbol{k}$, maka tidak ada taksiran kemungkinan maksimum.

3. Pembahasan dan Diskusi

Penaksiran Kemungkinan Maksimum Untuk Model Reliabilitas *Software* Hipergeometrik Menggunkan Hasil Bagi Pertumbuhan

Pemerikasaan Kasus

Tetapkan terlebih dahulu nilai c_n menggunakan Persamaan (2.1), diperoleh $c_n = 304 + 3 + 3 + \cdots + 285 = 3.701$. Kemudian lakukan pemeriksaan kasus:

Kasus A ditolak, karena tidak ada $c_n = w_k$ untuk beberapa k dan $w_j > 0$ untuk paling sedikit satu $j \neq k$, maka penaksiran kemungkinan $\widehat{m} \neq 3.701$

Kasus B ditolak, karena tidak ada $c_n = w_k$ untuk beberapa k dan $w_j = 0$ untuk semua $j \neq k$, maka setiap nilai yang lebih besar sama dengan c_n bukanlah suatu penaksir kemungkinan maksimum dari m.

Kasus C ditolak, karena tidak ada $w_k = c_k - c_{k-1}$ untuk semua k, maka ada taksiran kemungkinan maksimum.

Setelah dilakukan pemeriksaan kasus, diketahui bahwa adanya taksiran kemungkinan maksimum. Maka dilakukan langkah selanjutnya yaitu hitung penaksiran kemungkinan maksimum $\widehat{m}$ secara iteratif.

Menghitung Penaksiran Kemungkinan Maksimum $\widehat{m}$ Secara Iteratif

Menghitung penaksiran kemungkinan maksimum $\widehat{m}$ secara iteratif dapat dilakukan dengan menggunakan langkah-langkah perhitungan sebagai berikut:

1. Misalkan $m = x$, dan tetapkan $x = cn + 1$.
   Diketahui nilai $C_n = 3.701$, maka $x = 3.701 + 1 = 3.702$
2. Menggunakan hasil bagi pertumbuhan sebagaimana Persamaan (2.5) dan Persamaan (2.6), dilakukan perhitungan sebagai berikut:
   Diketahui nilai $C_n = 3.701$ dan $x = 3.702$, akan dicari nilai $Q(x)$:
   $$Q(x) = \frac{L(x)}{L(x-1)}$$
   $$Q(3.702) = \frac{(3.702 - 304) \cdot (3.702 - 302) \cdot \ldots \cdot (3.702 - 297)}{3.702^{31-1} \cdot (3.702 - 3.701)}$$
   $$= 280{,}3893$$

Karena $Q(3.702) > 1$, maka lakukan iterasi dengan mengulangi langkah pertama sampai memperoleh hasil $Q(x) \leq 1$.

Iterasi 1

Tetapkan $x = 3.702 + 1 = 3.703$

$$Q(3.703) = \frac{L(3.703)}{L(3.703 - 1)}$$
$$Q(3.703) = \frac{(3.703 - 304) \cdot (3.703 - 302) \cdot \ldots \cdot (3.703 - 297)}{3.703^{31-1} \cdot (3.703 - 3.701)}$$
$$= 140{,}3344$$

Iterasi 2

Tetapkan $x = 3.703 + 1 = 3.704$

$$Q(3.704) = \frac{L(3.704)}{L(3.704-1)}$$

$$Q(3.704) = \frac{(3.704-304)\cdot(3.704-302)\cdot\ldots\cdot(3.704-297)}{3.704^{31-1}\cdot(3.704-3.701)}$$

$$= 93{,}6495$$

Iterai 3

Tetapkan $x = 3.704 + 1 = 3.705$

$$Q(3.705) = \frac{L(3.705)}{L(3.705-1)}$$

$$Q(3.705) = \frac{(3.705-304)\cdot(3.705-302)\cdot\ldots\cdot(3.705-297)}{3.703^{31-1}\cdot(3.703-3.701)}$$

$$= 70{,}3071$$

⋮

⋮

Iterasi 404

Tetapkan $x = 4.105 + 1 = 4.106$

$$Q(4.106) = \frac{L(4.106)}{L(4.106-1)}$$

$$Q(4.106) = \frac{(4.106-304)\cdot(4.106-302)\cdot\ldots\cdot(4.106-297)}{4.106^{31-1}\cdot(4.106-3.701)}$$

$$= 0{,}9995$$

Perhitungan tersebut dapat disajikin ke dalam bentuk tabel sebagai berikut,

Tabel 1. Penaksir Kemungkinan Maksimum $\hat{m}$ Secara Iteratif

$i(x)$	x	$L(x)$	$L(x-cn)$	$Q(x)$
0	3702	3,168501E+109	1,130036E+107	280,3892536
1	3703	3,197464E+109	2,278460E+107	140,3344371
2	3704	3,226683E+109	3,445487E+107	93,64954603
3	3705	3,256160E+109	4,631337E+107	70,30713633
⋮	:	⋮	⋮	⋮
404	4106	1,022845E+111	1,0233481E+111	0,999508664

Berdasarkan **Tabel 1**, dapat dilihat bahwa iterasi berhenti pada iterasi ke-404, karena $Q(4.106) = 0{,}9995 < 1$. Maka penaksiran kemungkinan maksimumnya adalah $\hat{m} = 4.106 - 1 = 4.105$. Dengan demikian taksiran banyaknya kerusakan pada sistem operasi dan *software* yang terdapat pada laptop MSI GL62M ada sebanyak 4.105 kerusakan.

4. Kesimpulan

Dalam skripsi ini telah diterapkan model reliabilitas *software* hipergeometrik menggunakan hasil bagi pertumbuhan untuk menaksir banyaknya kerusakan sistem operai dan *software* yang terdapat pada laptop MSI GL62M menggunakan *Auslogics Registry Cleaner* sebagai alat pendeteksi kerusakan. Hasil perhitungan menunjukkan bahwa taksiran banyaknya kerusakan pada sistem operasi dan *software* yang terdapat pada laptop MSI GL62M ada sebanyak 4.105 kerusakan.

5. Acknowledge

Penelitian ini merupakan bentuk syukur, juga salah satu bentuk dari bakti kepada kedua orang tua tercinta, Mamah (Cucu Hernawati) dan Papah (Pitoyo), tidak lupa adik (Fuad Hasan) dan kakak (Nani Purwaty) tersayang yang selalu memberikan *support* baik dalam bentuk do'a, waktu, pikiran, dan keleluasaan mulai dari awal pendaftaran kuliah hingga kelulusannya. Sekali lagi terimakasih sebesar-besarnya.

Daftar Pustaka

[1]Auslogics.com. Registry Cleaner Free, 2008, (Online), (https://www.auslogics.com/en/software/registry-cleaner/) diakses pada tanggal 07 Juli 2021

[2] Nesabamedia.com. Download Auslogics Registry Cleaner 8.2.0.4, 8 Januari 2021, (Online), (https://www.nesabamedia.com/download-auslogics-registry-cleaner/) di akses pada tanggal 07 Juli 2021.

[3] Nifontov, N., dan Berezovsky, A. (2005). Advanced Registry Doctor (Professional Version). (www.http:\antispy4you.com).

[4] Padberg, Frank. (2002). Maximum Likelihood Estimasi For The Hypergeometric Software Reliability Model. Fakultat Fur Informatik, Universitat Karlsruhe.

[5] R.-H. Hou, S.-Y. Kuo, Y.-P. Chang. (1994)."Applying Various Learning Curves to Hyper-Geometric Distribution Software Reliability GrowthModel", Proceedings International Symposium on Software Reliability Engineering.

[6] Sutoyo. (2012). Pemodelan Data Statistik Melalui Pendekatan Distribusi Diskrit. UIN Suska Riau.

[7] Shofwani Sheila Ghazia, Kudus Abdul. (2021). *Penentuan Kriteria Pengunjung dalam Pemilihan Green Hotel di Kota Bandung Menggunakan Metode Discrete Choice Experiment dengan Desain Choice Sets Kombinatorial.* Jurnal Riset Statistika, 1(1), 1-9.