

Kehandalan *Software* Berdasarkan Data Sekunder Menggunakan Distribusi Poisson dan Kualifikasi Cronbach's Alpha

Bambang Pudjoatmodjo¹, Mokhamad Hendayun²

Abstract--In the last 17 years, the usage of computer software, such as operating system software and application software, (after the appearance of Windows 95) have increased rapidly. This is shown by the increasing number of choices for user to use the software to facilitate their work, such as word processing software, digital image processing, and financial application. A software must have a good reliability. It means the software can work according to the needs of the user. In addition, the software can perform fast recovery when subjected to disruption caused by mistakes made by users of the application. Therefore, we need a method for determining the software reliability. The assessment is based on the observation of failures occur in a certain interval range. Based on the observation we will determine the specific characteristics of data that will be used to select the probability distribution to determine the value of reliability. The type of data is secondary data (discrete). This aims to conduct a preliminary assessment before they are applied in real cases. The determination of reliability is used to draw conclusions on observations. Cronbach's Alpha scale is used to see how well the conclusions are drawn.

Intisari--Pemakaian perangkat lunak, baik perangkat lunak sistem operasi maupun perangkat lunak aplikasi, pada 17 tahun terakhir ini mengalami peningkatan. Hal ini ditandai dengan semakin banyaknya pilihan bagi pengguna untuk memakai perangkat lunak yang akan digunakan untuk memudahkan pekerjaan. Perangkat lunak yang dibuat tentunya harus memiliki kehandalan yang baik. Kehandalan yang baik memiliki arti yaitu dapat bekerja sesuai dengan kebutuhan pihak pengguna. Selain itu, perangkat lunak dapat melakukan pemulihan dengan cepat ketika mengalami gangguan yang diakibatkan oleh kesalahan yang dilakukan oleh pengguna aplikasi. Untuk itu diperlukan suatu metode penentuan nilai kehandalan perangkat lunak. Penilaian tersebut didasarkan pada pengamatan banyaknya kegagalan yang terjadi dalam suatu rentang interval tertentu. Dari pengamatan banyaknya kegagalan yang terjadi akan ditentukan ciri-ciri khusus data yang akan dijadikan dasar pemilihan distribusi peluang untuk menentukan nilai kehandalan. Data yang digunakan adalah data sekunder (bersifat diskrit). Penggunaan data sekunder ditujukan untuk melakukan kajian pendahuluan sebelum diterapkan ke kasus nyata. Penentuan nilai kehandalan digunakan untuk menarik kesimpulan terhadap pengamatan yang dilakukan. Skala Cronbach's Alpha digunakan untuk melihat seberapa baik kesimpulan yang diambil.

Kata kunci : Perangkat Lunak, Kehandalan Perangkat Lunak, Distribusi Peluang, Cronbach's Alpha.

¹ Dosen, Program Studi D3 Teknik Informatika, Universitas Telkom, Telekomunikasi Terusan Buah Batukota Bandung 40257, INDONESIA (telp:022-7564108; e-mail: b.pudjoatmodjo@gmail.com)

² Dosen, Program Studi S1 Teknik Informatika, Universitas Langlangbuana, Karapitan 116 Bandung, INDONESIA (hendayun@gmail.com)

I. PENDAHULUAN

Perkembangan perangkat lunak untuk mendukung aktivitas kegiatan manusia, dari tahun 1995 sampai dengan tahun 2009 (setelah sistem operasi Windows 95 diluncurkan) meningkat tajam. Tampilan sistem operasi Windows yang ramah terhadap pengguna mengakibatkan banyak produsen perangkat lunak yang membuat aplikasi yang berjalan di atas sistem operasi ini.

Pembangunan perangkat lunak untuk sistem operasi dan aplikasi pendukungnya, pada dasarnya, melewati tahapan [1] analisis kebutuhan perangkat lunak, perancangan perangkat lunak, pembangunan (*development*) perangkat lunak, dan pengujian perangkat lunak. Tujuan tahapan dalam pembangunan perangkat lunak adalah untuk tercapainya perangkat lunak yang mempunyai performa yang baik. Performa perangkat lunak yang baik adalah perangkat lunak yang mempunyai tingkat kesalahan (*error*), kekurangan (*fault*), dan kegagalan (*failure*) yang minimal. Berdasarkan [2] *error* merupakan kekeliruan yang diakibatkan oleh *programmer*, *fault* [2] merupakan perwujudan *error* dalam kode program, dan *failure* [2] merupakan sebuah peristiwa yang terjadi ketika pengguna mendapatkan bahwa program berhenti memberikan layanan yang diharapkan. *Failure* biasanya terjadi akibat masukan yang tidak sesuai dari pengguna.

Pada dasarnya *error* dan *fault* dapat diminimalkan dari tahapan-tahapan yang ada di pembangunan perangkat lunak, sedangkan *failure* mempunyai kemungkinan terjadi sebagai akibat masukan dari pengguna yang tidak sesuai. Berdasarkan data *failure* yang terjadi akan dilakukan analisis apakah dengan catatan data *failure* yang ada, perangkat lunak yang dibangun layak untuk diluncurkan kepada pengguna.

Untuk mengetahui layak atau tidaknya, analisis yang dilakukan adalah menentukan nilai kehandalan perangkat lunak. Jadi kajian yang dilakukan adalah bukan membuat model ataupun menentukan berapa lama (dalam jam) perangkat lunak diuji sebelum diberikan kepada pengguna.

Untuk menentukan tingkat kehandalan perangkat lunak berdasarkan data kegagalan (bersifat diskrit) yang dicatat selama selang waktu tertentu akan digunakan distribusi Poisson. Data yang digunakan untuk menentukan nilai kehandalan adalah data sekunder (data diperoleh dari sumber lain).

Data sekunder dipilih dengan karakteristik data diskrit. Pemilihan data sekunder ditujukan untuk melakukan kajian pendahuluan sebelum diterapkan pada kasus nyata. Pada kajian pendahuluan ini, akan ditunjukkan bahwa dengan data kegagalan yang tercatat dapat ditentukan nilai kehandalan perangkat lunak.

II. KEHANDALAN PERANGKAT LUNAK

Pembuatan sebuah sistem didasarkan pada kebutuhan yang diberikan oleh pihak pengguna. Pada saat sistem tersebut selesai dibuat, dilakukan proses pengujian agar pihak pembuat yakin bahwa sistem yang dibuat telah sesuai dengan keinginan pihak pengguna.

Pada saat melakukan pengujian tentunya ditemukan data kegagalan yang terjadi. Data kegagalan ini dianalisis untuk menentukan kehandalan sistem tersebut sehingga pihak pembuat dapat memutuskan untuk melepas produk atau tidak.

Kehandalan adalah peluang dari keberhasilan atau peluang bahwa sistem akan memberikan performa yang sesuai fungsinya berdasarkan batasan desain yang telah dibuat [2],[3],[4].

Dari definisi di atas, konsekuensi dari performa yang baik adalah sistem yang dibuat mempunyai tingkat kesalahan, kekurangan, dan kegagalan yang minimal.

Secara matematis, Kehandalan $R(t)$ adalah peluang bahwa sistem akan bekerja dengan baik pada interval waktu 0 sampai dengan t , dinotasikan sebagai berikut [1].

$$R(t) = P(T > t) \quad t \geq 0 \quad (1)$$

dengan T adalah peubah acak yang menunjukkan waktu kegagalan. Nilai kehandalan perangkat lunak akan ditentukan berdasarkan hasil pencatatan kegagalan yang terjadi.

Setelah data peubah diskrit (tentang kegagalan) didapat, dilakukan analisis terhadap data untuk menentukan distribusi peluang.

Adapun distribusi peluang yang dipilih adalah distribusi Poisson. Pemilihan distribusi Poisson digunakan karena kegagalan yang jarang terjadi dan data kegagalan yang diambil mempunyai ciri-ciri yang sesuai distribusi Poisson, yaitu nilai peubah acaknya diskrit dan memilih parameter yang menyatakan rata-rata dari suatu kejadian dalam suatu interval waktu tertentu (λ). Berdasarkan parameter λ , yang menyatakan rata-rata dari suatu kejadian, terdapat dua kondisi yaitu λt konstan (*Homogenous Poisson Model*) dan $\lambda(t)$ tidak konstan (*Non Homogenous Poisson Model*). Parameter λ merupakan fungsi dari waktu.

III. METODE YANG DIGUNAKAN

Distribusi Poisson merupakan distribusi teoretis yang merupakan alat untuk menentukan apa yang dapat kita harapkan bila asumsi-asumsi yang dibuat benar. Distribusi teoretis diperlukan karena menggunakan distribusi sebenarnya harus diperoleh melalui eksperimen yang biasanya memakan biaya yang tinggi dan kadang karena suatu hal yang tidak dapat dilakukan.

A. Homogenous Poisson Model

Distribusi Poisson mempunyai karakteristik yaitu [5] peubah acaknya diskrit dan informasi mengenai besarnya nilai rata-rata dari suatu kejadian dalam suatu interval waktu tertentu (disimbolkan dengan λ). Persamaan dari Distribusi Poisson ditunjukkan pada (2) [2].

$$P(X = x) = \frac{(\lambda t)^x e^{-\lambda t}}{x!}, \text{ untuk } x = 0, 1, 2, \dots \quad (2)$$

dengan

λt = laju kegagalan konstan

x = banyaknya kegagalan dalam interval waktu t .

Dengan kata lain, $P(X = x)$ adalah peluang terjadinya tepat x kegagalan dalam waktu t .

Kehandalan dinyatakan dalam (3) [2] :

$$R(x) = \sum_{x=0}^k \frac{(\lambda t)^x e^{-\lambda t}}{x!} \quad (3)$$

B. Non Homogenous Poisson Model

Non Homogenous Poisson model (NHPM) merupakan pengembangan *Homogenous Poisson Model* (HPM). Jika pada HPM laju kegagalan, λt , diasumsikan konstan maka NHPM laju kegagalan merupakan fungsi dari waktu, $\lambda(t)$.

Dengan laju kegagalan tidak diasumsikan konstan, ($\lambda(t)$), maka banyaknya kejadian pada suatu selang berdasar pada panjang interval dan waktu terjadinya peristiwa yang diamati. Persamaan dari Distribusi Poisson dengan $\lambda(t)$ merupakan fungsi laju kegagalan adalah (2):

$$P(X = x) = \frac{(\lambda(t))^x e^{-\lambda(t)}}{x!}, \text{ untuk } x = 0, 1, 2, \dots$$

dengan

$\lambda(t)$ = fungsi laju kegagalan

x = banyaknya kegagalan dalam interval waktu t .

Persamaan kehandalan dengan $\lambda(t)$ merupakan fungsi laju kegagalan dinyatakan dalam (4):

$$R(x|t) = \sum_{x=0}^k \frac{(\lambda(t))^x e^{-\lambda(t)}}{x!} \quad (4)$$

C. Statistika Inferensi

Statistika inferensi adalah statistika dengan semua informasi dari sampel yang digunakan untuk menarik kesimpulan mengenai karakteristik populasi dari mana sampel tersebut diambil [5]

Dalam statistika inferensi, penarikan kesimpulan dilakukan melalui dua cara yaitu penaksiran parameter dan pengujian hipotesis.

1) *Penaksiran Parameter*: Penaksiran merupakan suatu alat dalam statistika yang digunakan untuk memperkirakan nilai sesungguhnya dari suatu parameter. Dalam statistika dikenal dua penaksiran yaitu [5] penaksiran titik dan penaksiran selang. Ciri-ciri penaksir yang baik, yaitu tak bias, varians minimum (efisien), dan konsisten.

2) *Pengujian Hipotesis*: Hipotesis adalah suatu pernyataan yang akan diuji nilai kebenarannya [5]. Terdapat dua buah hipotesis yaitu hipotesis nol (H_0) dan hipotesis tandingan (H_1).

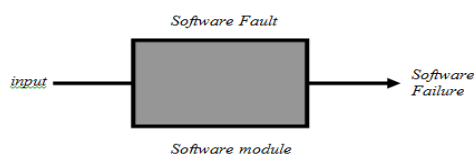
3) *Kualifikasi Kehandalan*: Salah satu alat yang digunakan untuk melakukan kualifikasi kehandalan adalah kualifikasi

Cronbach's Alpha[6] dengan kriteria *Excellent* (91%-100%), *Good* (81%-90%), *Acceptable* (71%-80%), *Questionable* (61%-70%), *Poor* (51%-60%), *Unacceptable* (0%-50%), dan angka 80% merupakan *Reasonable goal*. Berdasarkan kualifikasi tersebut, nilai kehandalan yang diharapkan minimal berada pada kualifikasi *Acceptable* atau bernilai 71%-80%.

4) *Penentuan Nilai Kehandalan dengan Menggunakan Model Goel dan Okumoto*: Nilai kehandalan yang diperoleh dengan menggunakan Distribusi Poisson akan dievaluasi dengan model lain yang sudah dikenal. Hal ini dilakukan untuk mengetahui apakah hasil analisis yang dilakukan mempunyai nilai kehandalan yang lebih baik atau sama atau saling melengkapi atau berada di bawah model yang sudah dikenal. Model Goel dan Okumoto[2] merupakan model yang masuk ke dalam keluarga model eksponensial dan proses pendeteksian banyaknya kesalahan secara kumulatif pada waktu mengikuti distribusi Poisson. Hal inilah yang mendasari dipilihnya model Goel dan Okumoto sebagai model acuan.

IV. EKSPLORASI DATA

Gbr. 1 menunjukkan bagaimana kesalahan perangkat lunak (*software fault*) dipicu oleh masukan tertentu yang menyebabkan terjadinya kegagalan perangkat lunak (*software failure*).



Gbr. 1 Kaitan antara kesalahan dan kegagalan perangkat lunak [2].

Dari ilustrasi tersebut, yang akan dijadikan kajian untuk dianalisis adalah data kegagalan perangkat lunak. Data kegagalan yang terjadi dicatat dan dilakukan analisis statistik deskriptif untuk mengetahui perilaku data yang dicatat. Data yang dihimpun adalah hari pengujian dan jumlah kegagalan yang terjadi per hari dalam rentang waktu tertentu.

Data kegagalan perangkat lunak yang dijadikan kajian analisis adalah data sekunder [2]. Data sekunder digunakan untuk memverifikasi model yang diambil. Data yang dicatat adalah hari pengujian dan catatan terjadinya kegagalan perangkat lunak per hari dalam rentang waktu 21 hari.

Berikut ini adalah keterangan tentang data sekunder yang dijadikan bahan kajian dan Tabel I adalah hasil pengujian yang diambil dari buku tersebut.

Program kecil *online* untuk paket entri pengujian data (*The small online software data entry package test data*), ada sejak tahun 1980 di Jepang [2]. Banyaknya baris kode pada perangkat lunak tersebut mendekati 40.000 baris kode. Pengujian diukur oleh sejumlah *shift tester* yang melakukan uji kasus dan melakukan analisis terhadap hasil pengujian. Hasil pengujian yang disajikan adalah *testing day*, *failure* dan *cumulative failure*.

Proses pencatatan data kegagalan dihentikan pada hari ke 21. Penghentian pencatatan kegagalan didasarkan pada

persentase kumulatif kegagalan yang mendekati 10% atau berada di bawah 10%. Adapun perhitungan persentase kumulatif kegagalan sebagai berikut:

$$y = \frac{\text{kumulatif failure}}{\text{jumlah total kumulatif failure}} \times 100\% = \frac{46}{475} \times 100\% = 9,6\%$$

Pada hari ke-21 persentase kegagalan yang ada adalah 9,6% sehingga pencatatan kegagalan dihentikan.

TABEL I
HASIL PENGUJIAN [2]

Testing day	Failure	Cumulative Failure
1	2	2
2	1	3
3	1	4
4	1	5
5	2	7
6	2	9
7	2	11
8	1	12
9	7	19
10	3	22
11	1	23
12	2	25
13	2	27
14	4	31
15	1	32
16	6	38
17	1	39
18	3	42
19	1	43
20	3	46
21	1	47

A. Analisis Deskriptif

Ukuran pemusatan data merupakan suatu besaran yang menyatakan posisi di mana data akan mengumpul atau terkonsentrasi [5]. Adapun ukuran pemusatan data adalah sebagai berikut.

1) *Mean (rata-rata)*: Dari data pengamatan di atas didapatkan rata-rata = 2,238. Nilai rata-rata ini menunjukkan rata-rata terjadi kegagalan adalah sebanyak dua kegagalan.

2) *Kuartil*. Q_1 (Kuartil Bawah)=1 dan Q_3 (Kuartil Atas)=3.

3) *Data kegagalan*: Data kegagalan setelah diurutkan berdasarkan data hari pengujian dan kegagalan ditunjukkan pada Tabel II. Data terurut ke-11 adalah 2 dan median = 2

B. Pencilan

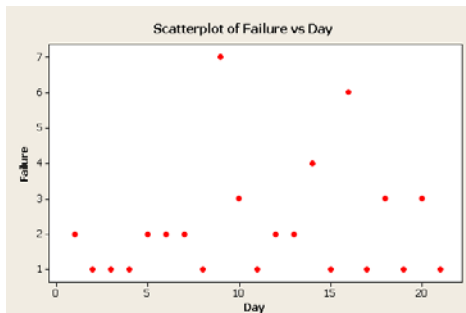
Pencilan memberikan informasi mengenai data yang nilainya jauh berbeda dari nilai data lainnya. Informasi ini diperlukan karena data yang masuk dalam pencilan akan mengganggu hasil analisis data[5]. Dari data didapat batas bawah pencilan (BBP = -2) dan batas atas pencilan (BAP = 6).

C. Analisis Data Hari Pengujian dan Kegagalan

Hasil plot kegagalan terhadap *testing day* ditujukan untuk mengetahui sebaran data pengujian selama 21 hari. Gbr. 2 menunjukkan grafik kegagalan terhadap hari pengujian.

TABEL II
DATA KEGAGALAN SETELAH DIURUTKAN

Data Kegagalan
1
1
1
1
1
1
1
1
1
1
1
2
2
2
2
2
2
2
2
3
3
3
4
6
7



Gbr 2. Grafik Kegagalan terhadap Hari Pengujian.

Dari sebaran data grafik, dapat dilihat bahwa kegagalan yang terjadi sesuai dengan analisis deskriptif. Rata-rata kegagalan yang terjadi adalah dua kegagalan per hari $\lambda t = 2.1 = 2$, dengan t konstan. Berdasarkan informasi tersebut dapat diputuskan bahwa data berdistribusi *Homogenous Poisson Process* dengan parameter $\lambda = 2$. Kegagalan yang terjadi masih dalam batas toleransi kegagalan yang diijinkan yaitu empat kegagalan.

Pada hari ke-9 terjadi lonjakan kegagalan yaitu sebanyak tujuh kegagalan dan pada hari ke-16 terjadi enam kegagalan. Data kegagalan yang dicatat pada hari ke-9 dan ke-16 merupakan data pencilan, sehingga data kegagalan pada hari ke-9 dan ke-16 tidak akan dimasukkan ke dalam analisis data.

D. Analisis Data Kegagalan dan Peluang Kegagalan

Distribusi Poisson dirumuskan dalam sebuah persamaan seperti ditunjukkan pada (5).

$$p_r = \frac{\lambda^x e^{-\lambda}}{x!}, x = 0,1,2,3, \dots \quad (5)$$

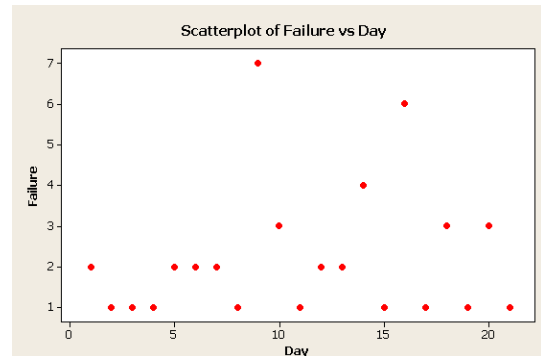
Berdasarkan (5) akan dihitung peluang kegagalan yang terjadi. Diharapkan peluang kegagalan yang terjadi kecil karena pada dasarnya pembuatan perangkat lunak dirancang untuk meminimalkan terjadinya kegagalan, bahkan bila memungkinkan tidak terjadi kegagalan.

Tabel III berikut ini adalah tabel perhitungan peluang kegagalan.

TABEL III
PERHITUNGAN PELUANG KEGAGALAN

Day	x	$p_r = \frac{\lambda^x e^{-\lambda}}{x!}$
1	2	0.267083
2	1	0.238467
3	1	0.238467
4	1	0.238467
5	2	0.267083
6	2	0.267083
7	2	0.267083
8	1	0.238467
9	7	0.005977
10	3	0.199422
11	1	0.238467
12	2	0.267083
13	2	0.267083
14	4	0.090224
15	1	0.238467
16	6	0.018678
17	1	0.238467
18	3	0.199422
19	1	0.238467
20	3	0.199422
21	1	0.238467

Grafik hubungan kegagalan terhadap hari pengujian ditunjukkan pada Gbr. 3.

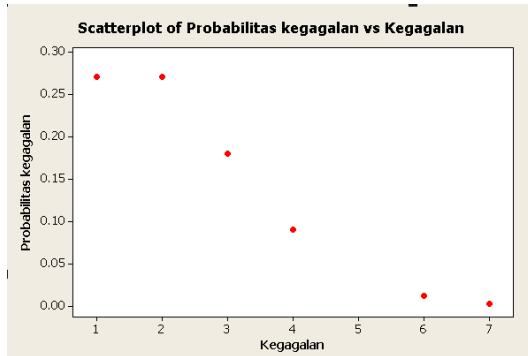


Gbr 3. Grafik Kegagalan terhadap hari pengujian.

Pada hari ke-9 dan ke-16 terjadi lonjakan kegagalan. Disebutkan pada pembahasan sebelumnya bahwa lonjakan kegagalan tersebut tidak memengaruhi pembuatan perangkat lunak. Terlihat pada tabel perhitungan peluang kegagalan pada hari ke-9 adalah 0,005977(0,6%), dan peluang kegagalan pada hari ke-16 adalah 0,018678 (1,8%). Ini menunjukkan bahwa memang kejadian terjadi tujuh kegagalan atau enam kegagalan per hari kecil dan dapat diabaikan. Kemudian, perhitungan peluang di atas akan digunakan untuk

menampilkan grafik probabilitas kegagalan (*failure probability*) dengan kegagalan yang terjadi.

Gbr. 4 menunjukkan bahwa peluang kegagalan semakin mengecil. Artinya perangkat lunak yang dibuat akan mempunyai kehandalan yang baik.



Gbr 4. Grafik probabilitas kegagalan terhadap kegagalan.

V. PENENTUAN NILAI KEHANDALAN

Setelah dilakukan analisis deskriptif untuk analisis perilaku data, hasil analisis tersebut akan digunakan untuk menentukan kehandalan perangkat lunak dengan menggunakan statistika inferensi dan nilai kehandalan dengan menggunakan kualifikasi Cronbach's Alpha[6].

Langkah-langkah untuk melakukan penentuan nilai kehandalan perangkat lunak adalah sebagai berikut.

- Menentukan $\hat{\lambda}$ menggunakan *Maximum Likelihood Estimator* (MLH), diperoleh

$$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x} = 2 \tag{6}$$

- Menunjukkan bahwa $\hat{\lambda} = \bar{x}$ adalah penaksir yang baik karena memenuhi kriteria tak bias, efisien dan konsisten[7].
- Menguji penaksir yang digunakan, merupakan alat yang baik atau tidak. Setelah penaksir terbukti merupakan alat yang baik maka kehandalan perangkat lunak dapat ditentukan menggunakan pengujian hipotesis.
- Menentukan besarnya nilai kehandalan ($R(x)$).

A. Pengujian Hipotesis

Pengujian hipotesis digunakan untuk menunjukkan bahwa rata-rata banyaknya kegagalan maksimum dua dapat diterima ($H_0 : \mu \leq 2$), dengan kata lain perangkat lunak handal, dan sebagai uji tandingan diambil rata-rata banyaknya kegagalan lebih dari dua ($H_1 : \mu > 2$), atau perangkat lunak tidak handal.

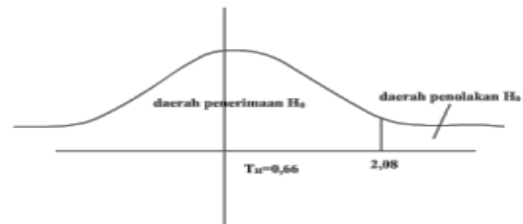
Untuk melakukan pengujian hipotesis dilakukan langkah-langkah sebagai berikut.

- Menentukan hipotesis uji, yaitu $H_0 : \mu \leq 2$ dan $H_1 : \mu > 2$
 $\alpha = 5\% = 0.05$
- Menentukan statistik uji, yaitu $S_x = 1.67$

$$T_H = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{2.24 - 2}{1.67/\sqrt{21}} = 0.66$$

- Pengambilan keputusan,

$$t_{0,05:v} = t_{0,05:20} = 2,0859 \quad \text{dimana } v = 20$$



Karena $T_H < 2,08$ (untuk $\alpha = 0,05$) maka H_0 diterima. Jadi rata-rata kegagalan maksimum dua dapat diterima. Atau dengan kata lain perangkat lunak handal.

B. Penentuan Kehandalan Perangkat Lunak dengan Model Goel dan Okumoto

Persamaan kehandalan untuk model Goel dan Okumoto dituliskan pada (7).

$$\hat{R}(x|t) = e^{-\hat{a}[e^{-bt} - e^{-b(t+x)}]} \tag{7}$$

Penaksir \hat{a} dan \hat{b} diperoleh dengan menggunakan metode *least square* yaitu:

$$\hat{a} = 1.81 \quad \text{dan} \quad \hat{b} = 0.0390$$

Diperoleh kehandalan seperti ditunjukkan pada Tabel IV.

TABEL IV
PERHITUNGAN KEHANDALAN DENGAN GOEL DAN OKUMOTO

Testing day	Failure	Reliability(%)
1	2	87.76
2	1	93.80
3	1	94.03
4	1	94.25
5	2	89.43
6	2	89.81
7	2	90.18
8	1	95.06
9	7	73.76
10	3	87.34
11	1	95.59
12	2	92.85
13	2	92.15
14	4	85.95
15	1	96.22
16	6	81.68
17	1	96.50
18	3	90.57
19	1	96.75
20	3	91.25
21	1	96.69

Dari Tabel IV, didapat nilai kehandalan berturut-turut mulai dari hari ke-1 sampai dengan hari ke-21 adalah 87,76%, 93,90%, ..., 96,99%.

Jadi kehandalan yang didapat dengan menggunakan model Goel dan Okumoto adalah 96,99%. Berdasarkan kualifikasi

Cronbach's Alpha[6] nilai kehandalan ini termasuk kategori *excellent*, dan nilai kehandalan ini menyatakan bahwa perangkat lunak tersebut tangguh dan layak untuk diserahkan kepada pihak pengguna.

C. *Distribusi Poisson*

Untuk menentukan besarnya nilai kehandalan, digunakan data pada Tabel V.

TABEL V
DATA TESTING DAY DAN FAILURE [2]

Testing day	Failure
1	2
2	1
3	1
4	1
5	2
6	2
7	2
8	1
9	7
10	3
11	1
12	2
13	2
14	4
15	1
16	6
17	1
18	3
19	1
20	3
21	1

Dari hasil analisis deskriptif terdapat informasi tentang batas atas pencilan yang bernilai 6. Pada Tabel *testing day* dan *failure*, terdapat dua data pencilan pada hari ke-9 dengan tujuh kegagalan dan hari ke-16 dengan enam kegagalan. Kedua data tersebut dapat kita abaikan dan tidak dimasukkan ke dalam perhitungan kehandalan. Pengabaian data ini juga diperkuat dengan kenyataan bahwa peluang kegagalan keduanya juga kecil yaitu 0,6% (probalitas kegagalan hari ke-9) dan 1,9% (probalitas kegagalan hari ke-16), sehingga tidak berpengaruh kepada kehandalan perangkat lunak.

Batas bawah pencilan menunjukkan apakah ada data yang bernilai -2 atau berada di bawah -2. Pada Tabel V tidak terdapat data batas bawah pencilan. Perhitungan kehandalan menggunakan (3), yaitu :

$$R(x) = \sum_{x=0}^k \frac{(\lambda t)^x e^{-\lambda t}}{x!}$$

dengan x menyatakan banyaknya kegagalan yang terjadi selama 1 hari, λt menyatakan rata-rata banyaknya kegagalan yang terjadi selama 1 hari, dan λ dihitung dengan

menentukan terlebih dahulu penaksir λ dengan menggunakan *Maximum Likelihood Estimation*.

$$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}$$

dan didapat

$$\hat{\lambda} = 2$$

Kehandalan yang diperoleh ditunjukkan pada Tabel VI.

TABEL VI
PERHITUNGAN KEHANDALAN

x	P(X = x)
0	0.107
1	0.239
2	0.267
3	0.199
4	0.112
5	0.050
Kehandalan	97.32%

Kehandalan yang diperoleh adalah $R(x) = 97,32\%$ (*excellent*) berdasarkan kualifikasi *Cronbach's Alpha*[6]. Nilai kehandalan ini menyatakan bahwa perangkat lunak tersebut tangguh dan layak untuk diserahkan kepada pihak pengguna.

VI. KESIMPULAN

Model Goel dan Okumoto memiliki kelebihan yaitu dapat memberikan informasi tentang kehandalan perangkat lunak untuk tiap hari pengamatan. Tetapi penentuan nilai parameter penaksir a dan b dengan menggunakan metode *Least Square* atau *Maximum Likelihood Estimator* (MLE) memiliki kesukaran dan membutuhkan pemecahan secara grafis (*graphical*) atau menggunakan metode numerik.

Distribusi Poisson memiliki kelebihan dalam hal proses perhitungan nilai kehandalan berdasarkan data kegagalan yang terjadi dan memberikan informasi nilai kehandalan yang memadai. Proses perhitungan nilai kehandalan dapat menggunakan analisis statistik biasa.

REFERENSI

- [1] Ian, Sommerville, "Software Engineering", Jilid 1, Edisi 6, Penerbit Erlangga, Jakarta, 2003.
- [2] Pham, Hoang, System Software Reliability, Springer, Verlag – London, 2006.
- [3] Tsong-Lyu, Michael R., Software Reliability Theory, Encyclopedia of Software Engineering, 2002, John Wiley & Sons, Article Online Update Posting Date: January 15, 2002.
- [4] Valido-Cabrera, Eduardo, Software Reliability Methods, Report, Technical University of Madrid, Agustus 2006.
- [5] Soleh, Achmad Zanbar, Ilmu Statistika, Cetakan Pertama, Penerbit Rekayasa Sains, Bandung, 2005Chillarege, Ram, What Is Software Failure, IEEE Transactions on Reliability, Vol.45, No.3, 1996 September.
- [6] George, D., and Mallery, P., SPSS for Windows step by step: A simple guide andreference. 11.0 update (4th ed.). Boston: Allyn & Bacon, 2003.
- [7] Herrhyanto, Nar., dan Gantini Tuti, Pengantar Statistika Matematis, Cetakan Pertama, Yrama Widya, Bandung, 2009.