

# Penerapan Metode *Rayleigh* dalam Prediksi Keandalan pada Aplikasi Berbasis Web

Putu Agus Suardana  
S1 Teknik Informatika  
Fakultas Teknik  
Universitas Telkom  
Bandung, Indonesia  
agussuardana22@gmail.com

## Abstrak

*Reliability prediction* merupakan salah satu cara yang dapat kita lakukan untuk prediksi tingkat keandalan sebuah aplikasi. Dengan melakukan prediksi sejak tahap awal pembangunan aplikasi, output yang dihasilkan juga memiliki keandalan yang jauh lebih baik. Salah satu metode yang dapat digunakan untuk memprediksi keandalan sebuah sistem yaitu metode *Rayleigh*. Implementasi metode ini akan dilakukan sejak tahap awal pembangunan sistem. Sehingga sebelum aplikasi tersebut ke tangan pengguna, kita sudah dapat membayangkan keandalan yang dimiliki oleh aplikasi tersebut nantinya. Prediksi ini juga berguna dalam meningkatkan kualitas dalam proses desain dan pengambilan keputusan untuk proses selanjutnya.

**Kata kunci** : aplikasi berbasis web, keandalan sistem, reliability prediction, metode Rayleigh.

## 1. Pendahuluan

Teknologi *web services* didasarkan pada konsep komputasi *service-oriented*. *Web services* merupakan standar yang mengintegrasikan aplikasi berbasis web melalui koneksi dan pertukaran proses bisnis antar jaringannya. Aplikasi berbasis web ini mengacu pada aplikasi yang diakses melalui web browser pada jaringan. Aplikasi berbasis web ini banyak diperlukan pada lingkup interaksi bisnis antar jaringan, misalnya beberapa perusahaan luar negeri yang melakukan *outsourcing* proyek satu sama lain [1]. Desainer aplikasi berbasis web harus mengatasi banyaknya tantangan selama proses pengembangan desain dalam rangka memenuhi kualitas kebutuhan layanan, termasuk di dalamnya kecepatan, keandalan, dan keamanan. Dalam beberapa tahun terakhir banyak aplikasi berbasis non-web telah ditulis ulang dengan berbasis web karena kebutuhan bisnis yang terus berkembang [2].

Para pengembang aplikasi hingga saat ini masih berusaha untuk membangun sebuah aplikasi yang handal dengan tingkat kualitas yang dapat diterima oleh user [3]. Salah satu ukuran dalam mengukur kualitas aplikasi adalah *reliabilitas* atau keandalan dari sistem tersebut. Keandalan software dapat didefinisikan sebagai peluang suatu komponen atau operasi software yang dilakukan tanpa mengalami kesalahan atau *defect* selama periode waktu tertentu dalam lingkungan tertentu [4]. Software *defect* adalah *error*, kegagalan (*failure*), atau kesalahan (*fault*) dalam sebuah program komputer atau sistem yang menghasilkan hasil yang salah atau hasil yang tidak terduga, dan menurunkan kualitas dari software [5].

Dengan melakukan prediksi keandalan ini, kita dapat mengevaluasi tingkat keandalan sistem bahkan sejak tahap awal pelaksanaan proyek. Prediksi ini juga dapat membantu dalam hal mengevaluasi kelayakan dari *reliability requirement* yang diusulkan serta memberikan landasan yang rasional terhadap proses desain dan pengambilan keputusan. Metode yang digunakan dalam penulisan ini adalah metode *Rayleigh* yang masih merupakan anggota dari keluarga distribusi *Weibull* [6]. Metode ini melakukan prediksi kesalahan dari waktu ke waktu selama masa pembangunan aplikasi berbasis web. Data yang digunakan pada jurnal ini merupakan data defect pada tahap awal pembangunan aplikasi *Prosys (Project Information System)* milik PT Rekayasa Industri, Jakarta.

## 2. Related Work

Fungsi dari model *Rayleigh* ditemukan pertama kali oleh fisikawan Inggris Lord Rayleigh dalam karyanya yang berkaitan dengan hamburan gelombang akustik dan elektro-magnetik. Secara empiris model ini juga mampu digunakan dengan baik pada sejumlah proses desain dengan feedback yang sangat signifikan sebagai bagian dari proses solusi. Pada perkembangan selanjutnya telah dilakukan penelitian juga bahwa model kehandalan *Rayleigh* ini sangat mendekati data *defect* yang sebenarnya dari proyek yang dikumpulkan pada upaya pengembangan software [7]. Pada tahun 1982, Trachtenberg [8] memeriksa histori *defect* per bulan pada proyek software yang diujinya dan menemukan bahwa pola dari *defect* yang dihasilkan menyerupai kurva *Rayleigh*. Pada tahun 1984, Gaffney dari divisi *Federal System IBM* mampu memproyeksikan jumlah laten dari data *defect* yang diperkirakan muncul dengan memodelkan data yang dimilikinya menggunakan model *Rayleigh* [9]. Putnam juga melakukan pekerjaan yang di dalamnya terdapat penggunaan dari model *Rayleigh* untuk mengestimasi jumlah dari software *defect* [10].

### 3. Metode *Rayleigh*

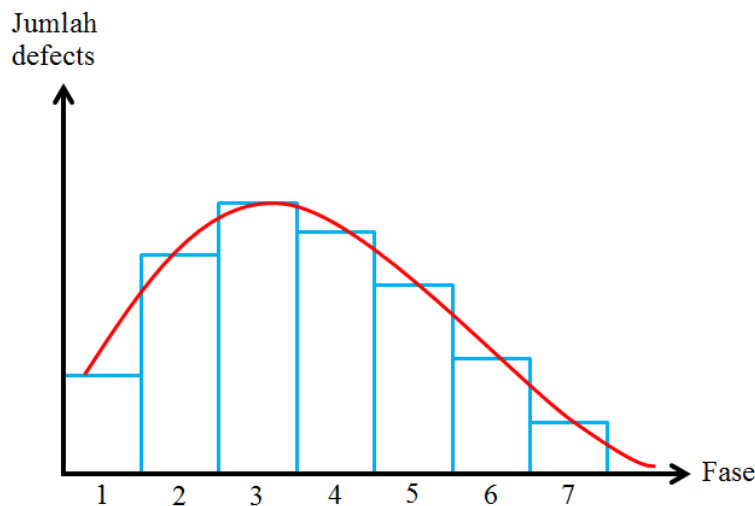
Metode yang digunakan pada penelitian ini adalah metode *Rayleigh*. Metode *Rayleigh* merupakan salah satu model dari distribusi *Weibull* [6]. Salah satu karakteristik yang khas dari metode ini adalah ujung ekor dari *PDF* (*Probability Density Function*) akan terus mendekati nol, tetapi tidak pernah menyentuh nilai tersebut. Berikut ini merupakan rumus *CDF* (*Cumulative Distribution Function*) dan *PDF* (*Probability Density Function*) dari metode *Rayleigh*:

$$CDF : F(t) = 1 - e^{-(t/\lambda)^k}$$

$$PDF : f(t) = \frac{k}{\lambda} \left(\frac{t}{\lambda}\right)^{k-1} e^{-(t/\lambda)^k}$$

Nilai  $k$  merupakan parameter *shape*,  $\lambda$  adalah parameter *scale*, dan  $t$  adalah *time*. Ketika diaplikasikan pada software, *PDF* juga sering diartikan sebagai tingkat kepadatan cacat (*defect density*) sepanjang waktu atau pola kemunculan *defect* sedangkan *CDF* berarti pola kemunculan *defect* secara kumulatif [6].

Model prediksi *Rayleigh* merupakan model yang paling cocok digunakan untuk melakukan prediksi keandalan suatu software diantara keluarga model distribusi *weibull* lainnya [11]. Model ini memprediksi jumlah kepadatan *defect* yang diharapkan pada berbagai fase proyek setelah parameter untuk grafik ditentukan. Parameter ini adalah jumlah total *defect* seluruh fase dan puncak grafik dalam satuan waktu. Jumlah *defect* yang ditemukan pada setiap fase akan mengikuti distribusi numerik yang dicari dengan menggunakan persamaan *Rayleigh*. Estimasi dari jumlah total *defect* secara keseluruhan diperoleh dengan menggunakan analisis persamaan regresi non-linier dengan data *defect* dari fase requirement dan desain. Estimasi jumlah *defect* dari fase yang lainnya dapat dicari dengan menggunakan persamaan rumus *PDF* model ini. Hasil perhitungan dengan menggunakan metode ini nantinya akan mengikuti grafik *Rayleigh* seperti Gambar 1.



Gambar 1. Grafik *Rayleigh*

Berdasarkan Gambar 1 di atas, grafik *Rayleigh* digambarkan dengan sumbu  $x$  adalah fase dari siklus pembangunan software dan sumbu  $y$  adalah jumlah atau banyaknya *defect* yang ditemukan untuk setiap fasenya. Daerah di bawah grafik menggambarkan total dari keseluruhan *defect* yang muncul selama siklus pembuatan software.

Kelebihan dari metode *Rayleigh* ini dalam melakukan prediksi tingkat keandalan adalah mampu menghitung prediksi kemunculan *defect* pada setiap fase sejak tahap awal pembangunan aplikasi yang diuji. Hal ini disebabkan karena metode *Rayleigh* melakukan prediksi pada setiap fase yang digunakan. Selain itu metode ini juga dapat digunakan bersamaan dengan model yang lain dalam memprediksi kehandalan sebuah aplikasi atau software. Namun kekurangan dari metode ini adalah jumlah *defect* harus dicari pada setiap fase pembangunannya yang kemudian dimodelkan dalam bentuk grafik untuk melihat perbandingan tingkat keandalan terhadap data prediksinya. Jadi penerapan metode *Rayleigh* ini akan lebih efektif diterapkan pada proyek yang memiliki skala medium ke atas dengan struktur pembangunan yang baik sesuai dengan siklus pengembangan proyek yang digunakan.

### 3.1 Prediksi Regresi Non-Linier

Dalam metode *Rayleigh* terdapat perhitungan untuk mencari parameter  $K$  menggunakan data *defect* aktual yang diberikan pada fase awal. Parameter  $K$  ini merupakan jumlah total *defect* dari keseluruhan fase yang diprediksi. Persamaan CDF (*Cumulative Distribution Function*) dari fungsi *Rayleigh*, yaitu:

$$F(y) = 1 - e^{-\frac{y^2}{K}}$$

Parameter *Peak* merupakan fase dengan jumlah *defect* tertinggi yang akan dijelaskan lebih lanjut pada sub bab selanjutnya. Model regresi non-linier yang digunakan untuk menghitung nilai  $K$  ini adalah dengan menggunakan model eksponen. Taksiran dari model eksponen ini dapat ditulis dengan rumus:

$$\hat{y} = a e^{bx} \quad \text{atau dalam bentuk logaritma,} \quad \log \hat{y} = \log a + (\log e) bx$$

Dengan  $a$  dan  $b$  adalah konstanta, sedangkan  $Y$  merupakan variabel terikatnya dan  $X$  sebagai variabel bebas. Dalam menghitung nilai  $K$  dari fungsi *Rayleigh*, yang menjadi variabel terikat adalah fase dari pembangunan aplikasi berbasis web. Sedangkan variabel bebasnya adalah jumlah *defect* pada masing-masing fase tersebut. Dalam bentuk logaritma, nilai  $a$  dan  $b$  dapat dicari dengan menggunakan rumus:

$$\log \hat{y} = \frac{\sum \log \hat{y}}{n} - (\log e) \frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n}$$

$$\log \hat{y} = \frac{(\sum x \log \hat{y}) - (\sum x) (\frac{\sum \log \hat{y}}{n})}{\sum x^2 - \frac{(\sum x)^2}{n}}$$

### 3.2 Persamaan Fungsi Rayleigh

Metode yang akan diimplementasikan untuk melakukan prediksi tingkat keandalan sebuah aplikasi berbasis web adalah metode *Rayleigh*. Metode ini akan melakukan prediksi jumlah *defect* yang muncul selama siklus pembangunan aplikasi berbasis web. Fungsi dalam metode *Rayleigh* ini mengikuti persamaan berikut [6]:

$$F(t) = 1 - e^{-\lambda t^2}$$

Parameter  $\lambda$  pada rumus di atas merupakan nilai kumulatif dari *defect density*,  $t$  merupakan waktu aktual, dan  $t_p$  merupakan waktu pada titik puncak grafik. Sebelum merepresentasikan grafik *Rayleigh*, parameter  $\lambda$  dan  $t_p$  harus diperkirakan dan ditentukan nilainya terlebih dahulu dengan menggunakan analisis regresi non-linier. Setelah semua parameter terdefinisi, perhitungan dapat dilakukan dan grafik dapat dibuat sesuai dengan interval waktu yang digunakan. Persamaan dari fungsi *Rayleigh* mengikuti formula sebagai berikut [6],

$$f(t) = \frac{1}{t_p^2} e^{-\lambda t^2} \cdot 2\lambda t$$

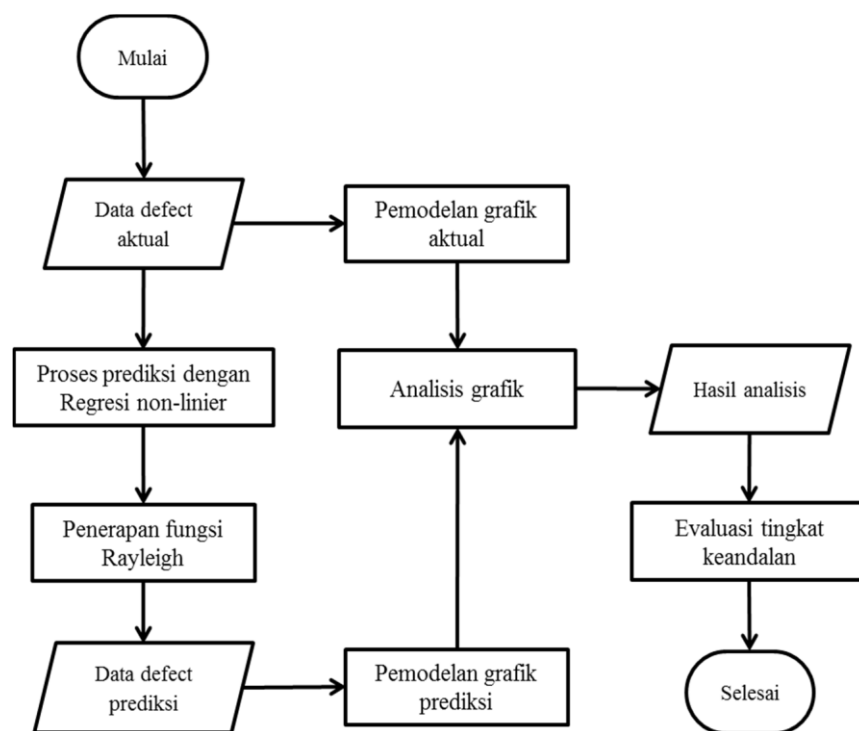
Parameter  $\lambda$  merepresentasikan jumlah *defect* secara keseluruhan, sedangkan *Peak* merupakan sebuah fungsi terhadap waktu, yaitu ketika grafik mencapai puncaknya, dan  $t_p$  merupakan waktu pada saat fase tertentu. Parameter *Peak* dapat dicari hasilnya dengan menggunakan rumus berikut [6]:

$$t_p = \frac{1}{\sqrt{\lambda}}$$

## 4. Pembahasan

### 4.1 Arsitektur Sistem

Sistem yang dibangun akan melakukan proses prediksi terhadap tingkat keandalan yang dimiliki oleh sebuah aplikasi berbasis web dengan menggunakan metode *Rayleigh*. Proses prediksi tingkat keandalan dilakukan dengan cara mengimplementasikan metode *Rayleigh* ke dalam sebuah aplikasi/tool. Input dari aplikasi ini adalah data *defect* pada tahap awal pembangunan aplikasi berbasis web yang diuji. Data *defect* ini selanjutnya akan lebih sering disebut dengan data *defect* aktual (data cacat yang sebenarnya). Secara umum, arsitektur sistem yang dibangun digambarkan dalam flowchart di bawah ini.

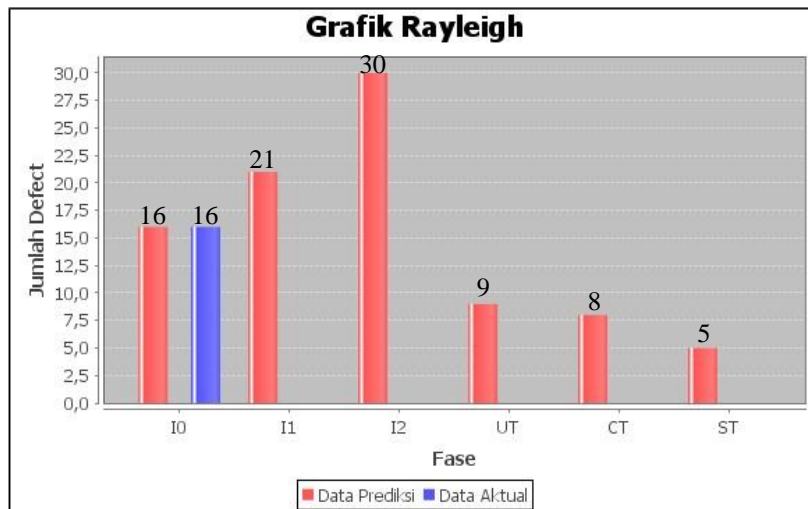


Gambar 2. Arsitektur sistem

### 4.2 Analisis Hasil Eksperimen

#### 4.2.1 Prediksi Data Defect

Pengujian aplikasi web *Prosys* ini dilakukan dengan menggunakan data *defect* aktual pada tahap awal pembangunan aplikasi web ini. Data *defect* ini didapatkan dengan melakukan review terhadap dokumen *SRS* (*Software Requirement Specification*). Berdasarkan hasil review yang sudah dilakukan didapatkan 16 buah *defect*. Data *defect* ini akan digunakan sebagai data input untuk menghitung jumlah data *defect* prediksi yang akan muncul pada fase selanjutnya hingga fase akhir pembangunan aplikasi web *Prosys*.



Gambar 3. Grafik prediksi aplikasi web Prosys

Setelah memasukkan data *defect* aktual aplikasi web Prosys, maka akan ditampilkan grafik seperti pada gambar di atas. Grafik akan menampilkan jumlah data prediksi yang dihitung dengan menggunakan persamaan Rayleigh. Data prediksi ini akan menjadi acuan jumlah *defect* aktual pada pembangunan aplikasi web Prosys untuk tahap selanjutnya. Jika ternyata pada tahap I1 jumlah *defect* aktual yang didapatkan nantinya melebihi nilai 21, maka dapat disimpulkan pada fase I1 pembangunan aplikasi web Prosys memiliki *defect* yang melebihi dari data prediksinya. Berdasarkan data tersebut tim pengembang aplikasi dapat menyimpulkan secara sementara bahwa pembangunan aplikasi web Prosys memiliki tingkat keandalan yang lebih rendah dibandingkan dengan data prediksinya. Dengan kesimpulan sementara ini tim pengembang aplikasi juga dapat menentukan usaha agar data *defect* yang diperoleh pada fase selanjutnya lebih sedikit dari jumlah data prediksinya.

#### 4.2.2 Pengukuran Tingkat Keandalan

Pengukuran tingkat keandalan dapat dilakukan setelah semua data *defect* aktual diberikan hingga fase keenam selama proses pembangunannya. Tingkat keandalan yang dimiliki oleh proyek A dapat dihitung dengan membandingkan jumlah total dari *defect* aktual dan *defect* prediksinya.

$$\frac{\text{Jumlah Defect Aktual}}{\text{Jumlah Defect Prediksi}} = \frac{\dots}{\dots} \times 100\%$$

## 5. Kesimpulan

Pengukuran tingkat keandalan sebuah software pada tahap awal pembangunannya dapat mengurangi total biaya proyek yang dikeluarkan serta meningkatkan efektifitas waktu yang dibutuhkan selama proses pembangunan software tersebut. Salah satu metode yang dapat diterapkan untuk memprediksi tingkat keandalan aplikasi adalah metode Rayleigh. Penerapan metode *Rayleigh* dalam memprediksi tingkat keandalan aplikasi berbasis web adalah dengan memprediksi jumlah total *defect* yang akan dihasilkan berdasarkan pada data *defect* yang ditemukan pada tahap awal pembangunan aplikasi berbasis web. Data prediksi ini akan menjadi acuan tingkat keandalan dari aplikasi web yang diuji.

## Referensi

- [1] A.-F. Sabah, "Deweloping Web Application," *International Journal of Software Engineering and it's Applications*, vol. 5, no. 2, pp. 57-68, 2011.
- [2] R. Sridaran, G. Padmavathi dan K. Iyakutti, "A Survey of Design Pattern Based Web Application," *Journal Of Object Technology*, vol. 8, no. 2, pp. 61-70, 2009.
- [3] A. K. Pandey dan N. K. Goyal, "Background: Software Quality and Reliability Prediction," dalam *Early Software Reliability Prediction, Studies in Fuzziness and Soft Computing*, India, Springer, 2013, pp. 17-33.
- [4] M. Razeef dan N. Mohsin, "Software Reliability Growth Models: Overview and Applications," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 9, pp. 1309-1320, 2012.
- [5] R. S. Wahono dan N. Suryana, "Combining Particle Swarm Optimzation based Feature Selection and Bagging Technique for Software Defect Prediction," *International Journal of Software Engineering and its Application*, vol. 7, no. 5, pp. 153-166, 2013.
- [6] S. H. Khan, *Metrics and Models in Software Quality Engineering*, Canada: Pearson Education, Inc., February 2007.
- [7] Quantitative Software Management, "<http://www.qsm.com/>," [Online]. Available: <http://www.qsm.com/reliability.pdf>. [Diakses 14 March 2015].
- [8] M. Trachtenberg, "Discovering How to Ensure Software Reliability," pp. 53-57, Jan-Feb 1982.
- [9] J. Gaffney dan E. John, "On Predicting Software Related Performance of Large-Scale Systems," dalam *Tenth International Computer Measurement Group Conference*, San Fransisco, 1984.
- [10] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," *Ieee Transactions On Software Engineering*, Vol. %1 dari %2SE-4, no. 4, pp. 345-361, 1978.
- [11] V. Alagappan, "Leveraging Defect Prediction Metrics in Software Program Management," *International Journal of Computer Applications* , vol. 50, no. 20, pp. 23-26, 2012.