

## MODEL SIMULASI PENGEMBANGAN PERANGKAT LUNAK MENGGUNAKAN *RATIONAL UNIFIED PROCESS (RUP)*

Tuti Karen Tia<sup>1</sup>, Wahyu Andhyka K<sup>2</sup>

Teknik Informatika, Fakultas Teknik  
Universitas Muhammadiyah Malang, Indonesia  
e-mail : tiyakaraen@gmail.com, kusuma.wahyu.a@gmail.com

Diterima: 18 April 2018. Disetujui : 15 Juni 2018. Dipublikasikan : 29 Juni 2018



©2018 –TESJ Fakultas Teknik Universitas Maarif Hasyim Latif. Ini adalah artikel dengan akses terbuka di bawah lisensi CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

### ABSTRAK

Rekayasa Perangkat Lunak menyajikan prosedur yang digunakan untuk pengembangan proyek perangkat lunak. *Software Development Life Cycle (SDLC)* adalah proses atau metodologi untuk pengembangan proyek yang memiliki tahapan tertentu sesuai dengan tujuan dan sasaran proyek. Setiap SDLC memiliki tahapan yang berbeda-beda. Salah satu model SDLC yang akan dibahas pada penelitian ini adalah *Rational Unified Process (RUP)*. Terdapat empat fase dalam RUP, yaitu *inception*, *elaboration*, *construction* dan *transition*. Seringkali pengembangan perangkat lunak tidak sesuai dengan perkiraan awal yang sudah ditentukan. Sehingga hal tersebut berpengaruh pada hasil dan kualitas dari perangkat lunak yang dihasilkan. Oleh karena itu perencanaan pada awal pengembangan proyek sangat penting untuk dilakukan, mulai dari penentuan jumlah tim, biaya dan waktu pengembangan proyek. Karena terdapat berbagai jenis proyek pada pengembangan perangkat lunak yaitu, proyek skala kecil, proyek skala menengah dan proyek skala besar. Setiap jenis proyek memiliki kebutuhan sumber daya yang berbeda. Pada penelitian ini mengusulkan sebuah model simulasi untuk pengembangan proyek menggunakan *Rational Unified Process*. Simulasi dilakukan untuk membantu manajer proyek dalam menentukan jumlah sumber daya yang optimal dengan waktu minimum agar produktivitas perusahaan dapat meningkat. Pembuatan model simulasi dilakukan pada komputer dengan menggunakan alat bantu simulator yaitu *Simphony.NET*.

**Kata kunci:** estimasi sumber daya, manajemen proyek, *rational unified process*, *rup*, *simphony.net*, simulasi model, *software development life cycle*, *sdlc*

### PENDAHULUAN

Pengembangan perangkat lunak dikaitkan dengan perancangan, penulisan, pengujian, implementasi dan pemeliharaan. Hal tersebut membentuk proses pengembangan perangkat lunak sebagai siklus pengembangan atau yang disebut dengan *Software Development Life Cycle (SDLC)*. SDLC adalah proses atau metodologi untuk pengembangan proyek yang memiliki tahapan tertentu sesuai dengan tujuan dan sasaran proyek. Tujuan utama dari SDLC adalah menyediakan produk berkualitas tinggi (Singh & Kaur, 2017). Terdapat beberapa model pengembangan, setiap model menggambarkan pendekatan terhadap aktivitas yang berlangsung selama pengembangan. Pemilihan model SDLC memiliki dampak yang sangat tinggi terhadap perangkat lunak yang dihasilkan. Oleh karena itu penting untuk memilih model SDLC yang sesuai dengan kebutuhan proyek.

Salah satu model SDLC yang akan dibahas pada penelitian ini adalah *Rational Unified Process*

(RUP). RUP adalah proses pengembangan perangkat lunak menggunakan pendekatan *usecase-driven* yang berarti mengacu pada *usecase* pada saat penentuan kebutuhan perangkat lunak, dan *iterative* yang berarti pengembangan berulang dan *incremental* sehingga dapat meningkatkan kualitas perangkat lunak yang dihasilkan. RUP digunakan untuk mengembangkan perangkat lunak berbasis komponen dan berfokus pada UML.

Terdapat berbagai ukuran pengembangan proyek perangkat lunak, yaitu proyek dengan skala kecil, proyek skala menengah dan proyek skala besar (Thind & Karambir, 2015). Di dalam sebuah perusahaan, pengembangan proyek tidak selalu memiliki skala yang sama. Sehingga apabila dalam perencanaan pengembangan proyek tidak dilakukan dengan tepat, maka akan berakibat pada kegagalan perangkat lunak yang dihasilkan. Selain itu, perusahaan dapat mengalami kerugian apabila perkiraan sumber daya tidak sesuai dengan kebutuhan pengembangan proyek perangkat lunak. Untuk mengatasi masalah tersebut, penelitian ini

mengusulkan sebuah model simulasi untuk pengembangan proyek menggunakan *Rational Unified Process*. Simulasi dilakukan untuk membantu manajer proyek dalam menentukan jumlah sumber daya yang optimal dengan waktu minimum agar produktivitas perusahaan dapat meningkat. Pembuatan model simulasi dilakukan pada komputer dengan menggunakan alat bantu simulator yaitu Symphony.NET.

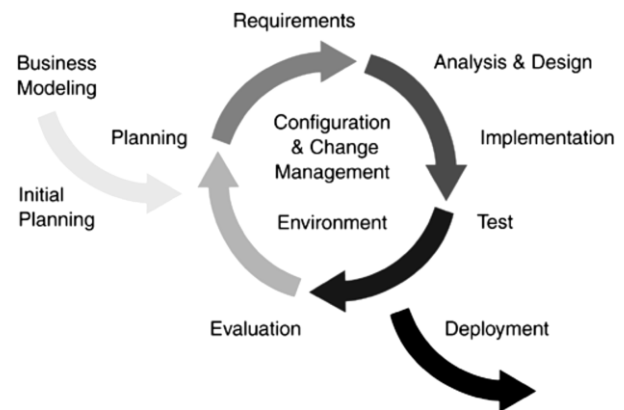
**METODE PENELITIAN**

*Rational Unified Proess (RUP)* merupakan salah satu produk pengembangan perangkat lunak yang ditawarkan oleh IBM *Rational*. RUP menyediakan kerangka kerja yang dapat disesuaikan untuk memenuhi kebutuhan individu organisasi. RUP bersifat serial dalam skala besar, berulang dalam skala kecil dan menghasilkan rilis bertahap dari waktu ke waktu (Ambler, 2005). Model pengembangan ini digunakan oleh berbagai perusahaan dan berbagai sektor industri. Pada pengembangan industri perangkat lunak yang mendukung penggunaan tim kecil atau tim besar.

RUP menggunakan pendekatan *iterative* atau berulang dengan urutan langkah-langkah tambahan. Setiap iterasi mencakup sebagian atau sebagian besar fase pembangunan. Setiap iterasi yang berurutan dibangun diatas dari hasil iterasi sebelumnya untuk memperbaiki sistem sampai produk akhir selesai. Proses berulang ini memungkinkan anggota tim untuk mempermainkan banyak

peran, selain itu manajer proyek dapat menggunakan tim yang tersedia dengan lebih baik sehingga dapat memperluas lingkup keahlian anggota tim (Kroll & Kruchten, 2003).

Tabel 1 dapat digunakan sebagai acuan untuk menentukan jumlah iterasi yang harus dilakukan. Iterasi membangun pekerjaan yang dilakukan oleh iterasi sebelumnya dan merakit sistem final secara bertahap. Ketika iterasi berakhir, khususnya selama fase Konstruksi, sebagian kecil dari sistem telah selesai yang mungkin bisa digunakan untuk pengguna sebagai rilis, meskipun hanya sebagai versi alfa atau beta (Ambler, 2005).

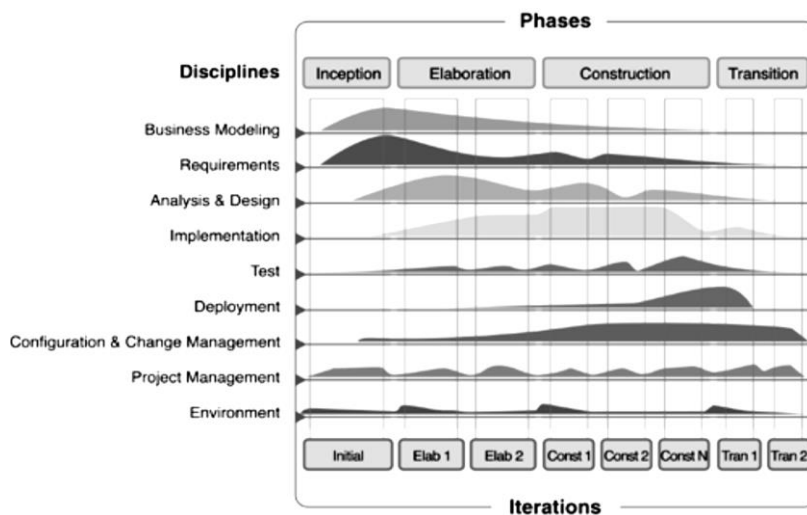


Gambar 1. Pengembangan iterative pada *Rational Unified Proess (RUP)*

Pada tahap pengembangan RUP terdapat struktur dinamis dan struktur statis. Struktur dinamis pada dimensi horizontal mewakili dimensi waktu dari proses. Ini menunjukkan bagaimana proses, dinyaa-

Tabel 1. Jumlah Iterasi berdasarkan Ukuran Proyek

Project Size (People)	Project Length (Months)	Iteration Length (Weeks)	Number Of Iteration			
			Inception	Elaboration	Construction	Transition
3	4	2-3	1	1	3	1
10	8	4	1	2	3	2
80	20	7-8	2	3	4	2



Gambar 2. Tahap Pengembangan pada *Rational Unified Process (RUP)*

takan dalam hal siklus, fase dan iterasi, selama siklus hidup proyek (Kroll & Kruchten, 2003). Struktur dinamis terdiri dari beberapa fase, yaitu:

**a. Inception**

Merupakan tahap untuk menentukan ruang lingkup dan *business case*. Menentukan seberapa penting kontribusi perancangan perangkat lunak dalam membantu proses bisnis yang ada.

**b. Elaboration**

Bertujuan untuk meningkatkan pemahaman tentang permasalahan utama. Pada tahap ini pengembang melakukan analisis persyaratan dan resiko, menentukan *base line*.

**c. Construction**

Melakukan iterasi yang meliputi analisis, desain, implementasi dan pengujian. Pada fase ini harus terbentuk perangkat lunak dan dokumentasi untuk pengguna.

**d. Transition**

Menghasilkan sebuah produk yang akan diuji dengan *beta* dan *performance testing*. Membuat dokumentasi untuk kebutuhan *training* dan *user guides*. Merencanakan *launching* produk kepada pengguna (Sommerville, 2004).

Fase tersebut dijalankan secara berurutan. Pekerjaan dikelompokkan ke dalam aktivitas logis yaitu disiplin yang dilakukan secara iteratif sepanjang fase (Ambler, 2005), Disiplin tersebut digambarkan pada struktur statis atau dimensi vertikal. Struktur statis terdiri dari alur kerja antara lain, *Business Modeling, Requirements, Analysis and Design, Implementation, Testing, Deployment, Configuration and Change Management, Project Management dan Environment*. Kerangka kerja tersebut memungkinkan organisasi pengembangan perangkat lunak dari semua ukuran untuk bekerja.

## HASIL DAN PEMBAHASAN

### Penelitian Terkait

Sonia Thind dan Karambir (Thind & Karambir, 2015) mengusulkan model simulasi untuk pengembangan perangkat lunak menggunakan Spiral dengan menggunakan alat simulasi komputer yaitu Symphony.NET, penelitian tersebut menghasilkan model simulasi yang dapat membantu manajer proyek dalam meningkatkan produktivitas perusahaan dengan menggunakan jumlah minimum sumber daya. Hal tersebut telah dilakukan dengan membuat semua anggota tim sibuk sepanjang waktu sehingga tidak ada waktu yang terbuang oleh tim siaga atau tim yang menunggu tahap lainnya selesai. Youssef Bassil (Bassil, 2012) membuat model simulasi untuk pengembangan perangkat lunak menggunakan Waterfall, yang bertujuan untuk mempermudah manajer proyek dalam menentukan bagaimana untuk mencapai produktivitas optimal dengan penggunaan sumber

daya minimum. Amninder Singh dan Puneet Jai Kaur (Singh & Kaur, 2017) membuat model simulasi dengan tujuan untuk menyediakan sumber daya pada proyek bersama dengan batasan waktu. Hasil model simulasi yang dibuat menggunakan 10 perusahaan dengan berbagai jenis proyek menunjukkan bahwa sejumlah anggota tim bekerja pada beberapa proyek dengan cara asosiatif.

### Definisi Masalah

Model pengembangan perangkat lunak yang digunakan oleh sebagian besar tim yaitu model Waterfall, dimana setiap fase diselesaikan dengan urutan yang ketat, analisis dan desain, kemudian implementasi dan pengujian. Pendekatan tersebut membuat anggota tim utama tidak bekerja karena harus menunggu tim pada tahap lainnya menyelesaikan tugasnya (Kroll & Kruchten, 2003).

Pengembangan perangkat lunak setiap proyek berbeda dan proses yang diikuti harus disesuaikan untuk memenuhi kebutuhannya. RUP menggunakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab pada pengembangan perangkat lunak yang bertujuan untuk memastikan produk yang dihasilkan merupakan produk berkualitas tinggi, selain itu produk yang dihasilkan menggunakan RUP harus dapat memenuhi kebutuhan pengguna akhir dalam jadwal dan anggaran yang diprediksi (Kroll & Kruchten, 2003). Oleh karena itu penentuan perencanaan awal proyek harus dilakukan dengan tepat. Tujuan dari pembuatan model simulasi ini adalah untuk membantu manajer proyek dalam menentukan perencanaan proyek yang sesuai.

### Model Simulasi

RUP merekomendasikan untuk menentukan tim yang diperlukan proyek menurut jenisnya. RUP memastikan bahwa semua anggota tim bekerja dengan baik, sehingga pada saat proyek mulai berjalan anggota tim yang berbeda diperlukan pada posisi yang berbeda. Pada tahap *Inception* fokusnya adalah fungsi manajemen dan membangun Visi. Dalam tahap *Elaboration* fokusnya adalah pada arsitektur sehingga tim akan terdiri dari lebih banyak desainer. Dalam tahap *Construction* fokusnya adalah pada pengembangan, sehingga tim akan terdiri dari lebih banyak pengembang. Dalam tahap *Transition* fokusnya adalah pada Jaminan Kualitas Penilaian Perangkat Lunak.

### Asumsi dan Spesifikasi

Simulasi model pada *Rational Unified Process* (RUP) menggunakan beberapa asumsi salah satunya adalah membuat asumsi bahwa proyek datang secara acak pada sebuah perusahaan perangkat lunak. Proyek tersebut dapat digolongkan menjadi 3 bagian berdasarkan kompleksitas

tas dan skala, antara lain: *Small scale project*, *Medium scale project*, dan *Large scale project*.

Asumsi selanjutnya untuk menentukan jumlah dari setiap proyek tersebut, yaitu 60% untuk *small scale project*, 30% untuk *medium scale project* dan 10% untuk *large scale project*. Waktu kedatangan proyek dapat ditetapkan dengan menggunakan rumus distribusi *triangular*. Dengan menentukan batas waktu minimum kedatangan proyek adalah 10 hari, batas waktu maksimum kedatangan proyek adalah 50 hari dan rata-rata waktu kedatangan proyek adalah 30 hari. Dapat ditulis dengan persamaan (1).

$$f(x | a, b, c) = \begin{cases} 0 & \text{for } x < 10 \\ \frac{2(x-a)}{(b-a)(c-a)} & \text{for } 10 \leq x < 30 \\ \frac{2}{b-a} & \text{for } x = 30 \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{for } 30 < x \leq 50 \\ 0 & \text{for } 50 < x \end{cases} \quad (1)$$

Selanjutnya adalah mengasumsikan jumlah anggota tim yang tersedia dalam perusahaan perangkat lunak:

5 BA, 5 D, 10 P, 20 T dan 5 MM.

Keterangan:

*Business Analyst* : BA

*Designer* : D

*Programmer* : P

*Tester* : T

*Maintenance Man* : MM

Sehingga dapat ditentukan pembagian tim untuk pengerjaan proyek perangkat lunak sebagai berikut:

### 1. Small Scale Project

- a. *Business Analyst* : Menggunakan distribusi uniform dengan lower limit 2 hari dan upper limit 3 hari.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 2 \leq x \leq 3 \\ 0 & \text{for } x < 2 \text{ or } x > 3 \end{cases} \quad (2)$$

- b. *Designer*: Menggunakan distribusi eksponensial dengan mean = 2  
 c. *Programming*: Menggunakan distribusi eksponensial dengan mean = 4  
 d. *Testing*: Menggunakan distribus uniform dengan lower limit 6 hari dan upper limit 10 hari

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 6 \leq x \leq 10 \\ 0 & \text{for } x < 6 \text{ or } x > 10 \end{cases} \quad (3)$$

- e. *Maintenance*: Menggunakan distribusi uniform dengan lower limit 2 hari dan upper limit 3 hari. Persamaan sama dengan (2)

### 2. Medium Scale Project

- a. *Business Analyst* : Menggunakan distribusi uniform dengan lower limit 5 hari dan upper limit 10 hari.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 5 \leq x \leq 10 \\ 0 & \text{for } x < 5 \text{ or } x > 10 \end{cases} \quad (4)$$

- b. *Designer*: Menggunakan distribusi eksponensial dengan mean = 8  
 c. *Programming*: Menggunakan distribusi eksponensial dengan mean = 15  
 d. *Testing*: Menggunakan distribus uniform dengan lower limit 10 hari dan upper limit 20 hari

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 10 \leq x \leq 20 \\ 0 & \text{for } x < 10 \text{ or } x > 20 \end{cases} \quad (5)$$

- e. *Maintenance*: Menggunakan distribusi uniform dengan lower limit 7 hari dan upper limit 10 hari.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 7 \leq x \leq 10 \\ 0 & \text{for } x < 7 \text{ or } x > 10 \end{cases} \quad (6)$$

### 3. Large Scale Project

- a. *Business Analyst* : Menggunakan distribusi uniform dengan lower limit 10 hari dan upper limit 20 hari. Persamaan sama dengan (5)  
 b. *Designer*: Menggunakan distribusi eksponensial dengan mean = 25  
 c. *Programming*: Menggunakan distribusi eksponensial dengan mean = 35  
 d. *Testing*: Menggunakan distribus uniform dengan lower limit 30 hari dan upper limit 35 hari

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 30 \leq x \leq 35 \\ 0 & \text{for } x < 30 \text{ or } x > 35 \end{cases} \quad (7)$$

- e. *Maintenance*: Menggunakan distribusi uniform dengan lower limit 15 hari dan upper limit 20 hari.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } 15 \leq x \leq 20 \\ 0 & \text{for } x < 15 \text{ or } x > 20 \end{cases} \quad (8)$$

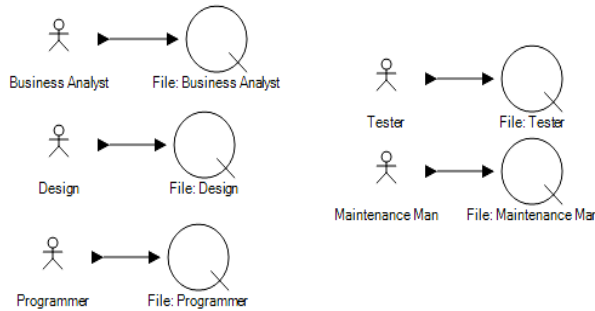
Kemudian mengasumsikan tingkat kegagalan atau error yang dapat terjadi pada setiap proyek:

1. Small scale project: 10% error = Probabilitas: 0,1 & 90% berhasil = Probabilitas: 0,9
2. Medium scale project: 20% error = Probabilitas: 0,2 & 80% berhasil = Probabilitas: 0,8
3. Large scale project: 30% error = Probabilitas: 0,3 & 70% berhasil = Probabilitas: 0,7

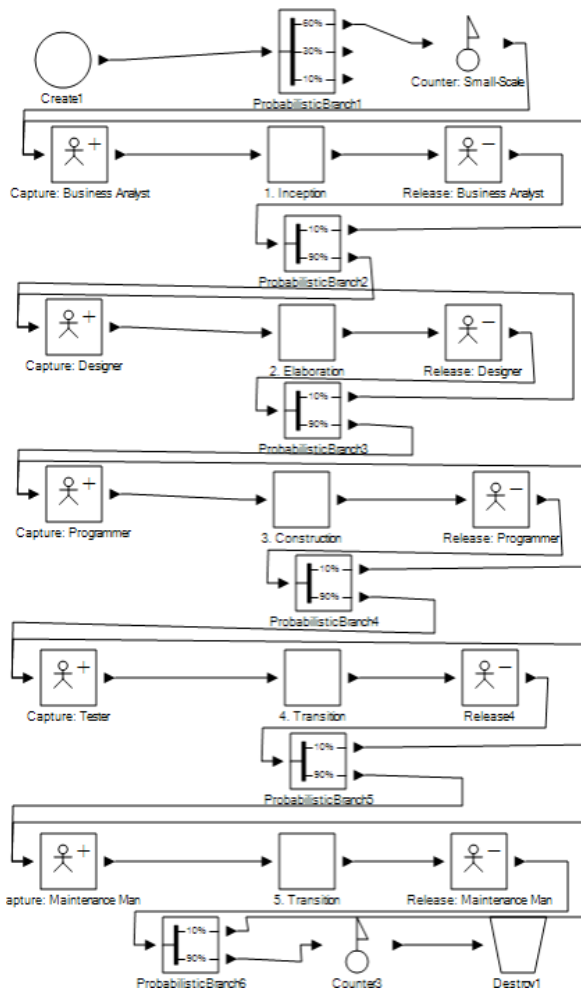
### Model Simulasi Rational Unified Process (RUP)

Model simulasi yang diusulkan menggunakan alat simulasi computer yaitu Symphony.NET (University of Alberta, 2018). *Library* pada Symphony yang digunakan untuk membuat model simulasi yang diusulkan terdiri dari *resource*,

file, capture, release, task, destroy, create, counter, dan probabilistic branch. Gambar 3 menggambarkan elemen pemodelan sumber daya dengan jumlah antriannya, yang terdiri dari bisnis analisis, desainer, programmer, tester dan maintenance man.



Gambar 3 . Elemen Pemodelan Sumber Daya



Gambar 4. Model Simulasi Proyek Small-Scale

Beberapa elemen *probability branch* tersedia diantara setiap tugas sumber daya yang berbeda, tujuannya adalah untuk mensimulasikan kesalahan yang mungkin terjadi pada saat pengembangan. Unsur probabilitas memiliki dua cabang yaitu cabang di bagian atas dengan Prob=0,1 menunjuk-

kan bahwa 10% tingkat kesalahan pada proyek small-scale sehingga harus mengulangi pada tahap sebelumnya untuk memperbaiki kesalahan tersebut. Kemudian cabang pada bagian bawah dengan Prob=0,9 menunjukkan bahwa 90% dari proyek small-scale tidak terdapat kesalahan setelah selesainya setiap tahap. Selain itu elemen *probability branch* terdapat pada awal model simulasi yang menggunakan tiga cabang berdasarkan jenis proyek yang sudah ditentukan. Cabang pertama dengan Prob=0,6 menunjukkan bahwa 60% proyek yang masuk adalah proyek small-scale. Cabang kedua dengan Prob=0,3 menunjukkan bahwa 30% proyek yang masuk adalah proyek medium-scale. Terakhir yaitu cabang ketiga dengan Prob=0,1 menunjukkan bahwa 10% proyek yang masuk merupakan proyek large-scale. Gambar 4 menunjukkan elemen pemodelan untuk mensimulasikan model small-scale dengan konfigurasi probabilitas yang sudah disebutkan diatas.

**Analisis dan Hasil Simulasi**

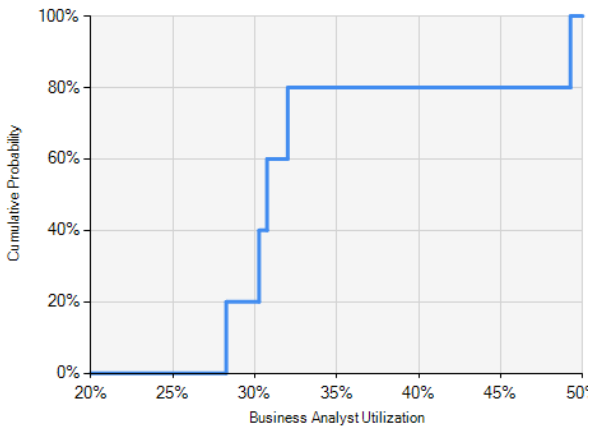
Simulasi model dieksekusi sebanyak 5 kali dengan waktu 1,234 detik untuk 60 proyek yang masuk menggunakan Symphony.NET. Berikut Tabel 2 menunjukkan pemanfaatan rata-rata setiap sumber daya setelah selesainya simulasi.

Tabel 2. Hasil Rata-Rata Sumber Daya

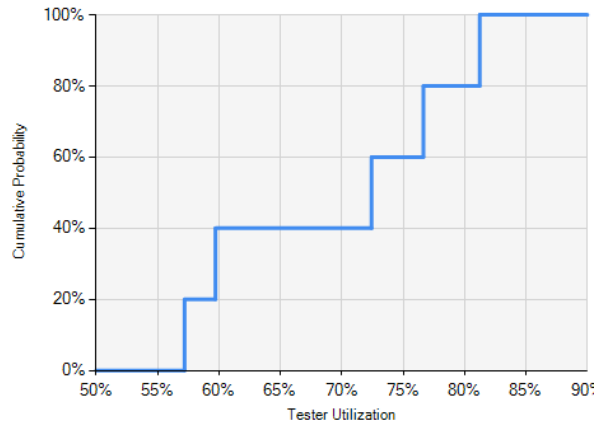
Resource	Average Utilization
Business Analyst	34,1%
Designer	34,7%
Programmer	34,7%
Tester	58,2%
Maintenance Man	69,4%

Setelah berhasil menjalankan model simulasi 5 kali, untuk proyek small-scale, menghasilkan grafik yang berbeda-beda untuk setiap pemanfaatan sumber daya yang digunakan. *Cumulative Probability* pada sumbu y relatif terhadap pemanfaatan berbagai sumber daya pada sumbu x yang digunakan untuk pemanfaatan sumber daya maksimum dari business analyst, designer, programmer, tester dan maintenance man untuk proyek small-scale.

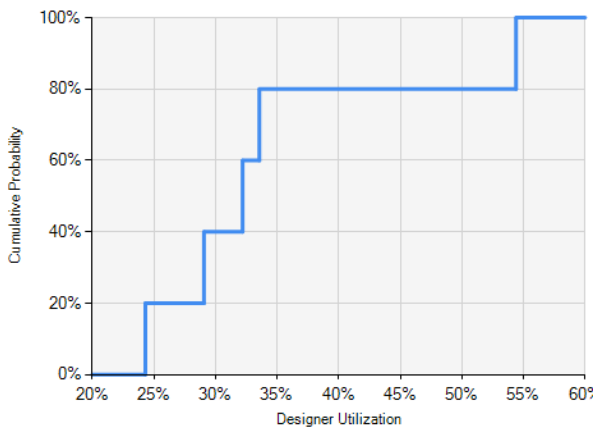
Berikut Gambar 5 menunjukkan pemanfaatan business analyst dengan nilai rata-rata 0,305 dan standar deviasi 0,048. Gambar 6 menunjukkan pemanfaatan *designer* dengan nilai rata-rata 0,370 dan standar deviasi 0,166. Gambar 7 menunjukkan pemanfaatan programmer dengan nilai rata-rata 0,647 dan standar deviasi 0,085. Gambar 8 menunjukkan pemanfaatan tester dengan nilai rata-rata 0,658 dan standar deviasi 0,084. Gambar 9 menunjukkan pemanfaatan maintenance man dengan nilai rata-rata 0,023 dan standar deviasi 0,066.



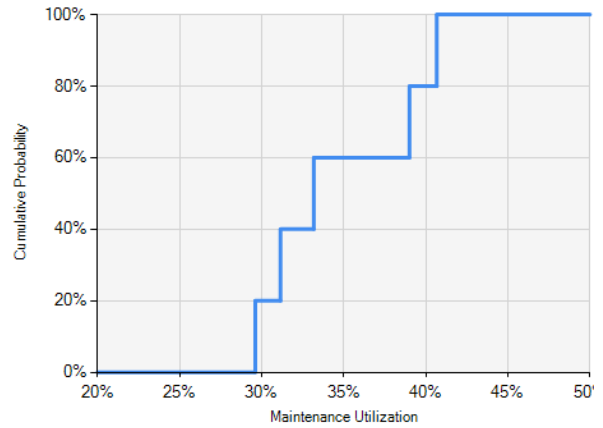
Gambar 5. Pemanfaatan *Business Analyst*



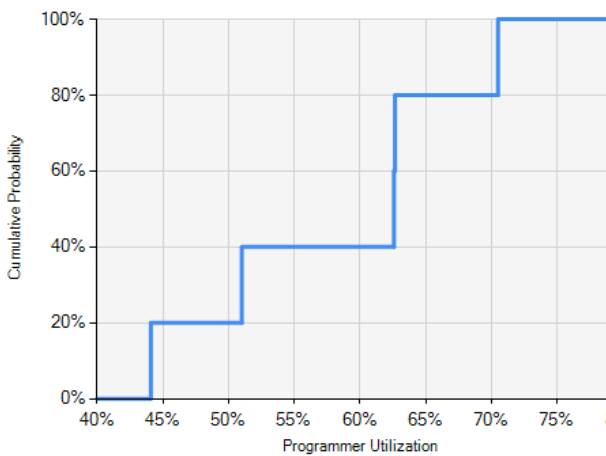
Gambar 8. Pemanfaatan *Tester*



Gambar 6. Pemanfaatan *Designer*



Gambar 9. Pemanfaatan *Maintenance Man*



Gambar 7. Pemanfaatan *Programmer*

## PENUTUP

Penelitian ini menghasilkan sebuah model simulasi untuk pengembangan perangkat lunak. Pembuatan model simulasi dilakukan dengan menggunakan Symphony.NET. Model simulasi yang dibuat terdiri dari semua fase pada *Rational Unified Process* (RUP) termasuk jumlah tim, waktu, dan jenis proyek. Pada penelitian ini terdapat tiga jenis proyek yaitu, proyek dengan skala kecil, skala menengah dan skala besar. Hasil dari pembuatan simulasi ini diharapkan dapat membantu manajer proyek dalam menentukan jumlah sumber daya yang optimal dengan waktu minimum agar produktivitas perusahaan dapat meningkat.

## DAFTAR PUSTAKA

- Ambler, S. W. (2005). A manager's introduction to the Rational Unified Process (RUP). *Version: December, 4, 2005*.
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology*, 2(5), 2049-3444. <https://doi.org/10.15680/ijircce.2015.0305>

013

- Kroll, P., & Kruchten, P. (2003). *The Rational Unified Process Made Easy. Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*.
- Singh, A., & Kaur, P. J. (2017). a Simulation Model for Incremental Software Development Life Cycle Model, *8*(7), 126–132.
- Sommerville, I. (2004). *Software Engineering* (7th ed.). Addison Wesley.
- Thind, S., & Karambir. (2015). A Simulation Model for the Spiral Software Development Life Cycle. *International Journal of Innovative Research in Computer and Communication Engineering*, *3*(5), 3823–3830. <https://doi.org/10.15680/ijircce.2015.0305013>
- University of Alberta. (2018). Symphony.NET 4.6.

Halaman ini sengaja dikosongkan