

Model Simulasi Rational Unified Process Pada Pengembangan Perangkat Lunak

Tuti Karen Tia^{*1}, Ilyas Nuryasin², Maskur³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

tiyakaren@gmail.com^{*1}, ilyas@umm.ac.id²

Abstract

Pada pengembangan perangkat lunak terdapat beberapa jenis metodologi yang dapat digunakan, salah satunya adalah Rational Unified Process (RUP). RUP merupakan salah satu jenis proses pengembangan perangkat lunak yang digunakan di berbagai perusahaan perangkat lunak. Terdapat berbagai jenis proyek pada pengembangan perangkat lunak RUP, yaitu proyek yang dibangun dari awal dan memiliki skala kecil, proyek yang memiliki skala besar dan proyek pengembangan dari sistem yang sudah dibangun sebelumnya. Setiap jenis proyek memiliki kebutuhan yang berbeda, terutama dalam penggunaan setiap peran. Pada RUP terdapat beberapa peran penting yang tidak dapat dilakukan oleh satu individu karena harus bekerja pada saat bersamaan. Pada dasarnya tujuan dari RUP adalah dapat menghasilkan perangkat lunak yang berkualitas tinggi. Hal tersebut tidak dapat tercapai apabila penggunaan peran yang kurang tepat. Penelitian ini mengusulkan jumlah peran yang dibutuhkan untuk pengembangan perangkat lunak menggunakan RUP dengan membuat model simulasi. Tujuan dari pembuatan model simulasi ini adalah untuk mengetahui peran yang sesuai berdasarkan setiap jenis proyek. Hasil dari model simulasi yaitu berupa persentase dari setiap peran yang dibutuhkan, project manager memiliki persentase terbesar dibandingkan dengan peran lainnya, yaitu sebesar 76% pada proyek skala kecil dan 60% pada proyek skala besar.

Kata Kunci: Rational Unified Process (RUP), Model Simulasi, Manajemen Proyek, Symphony.NET

Abstract

In software development there are several types of methodologies that can be used, one of which is Rational Unified Process (RUP). RUP is of software development process used in various software companies. There are various types of projects in RUP software development, small scale project, large-scale projects and re-engineering projects. Each type of project has different needs, especially in the use of each role. In RUP there are several important roles that cannot be done by an individual because they have to work at the same time. Basically, the purpose of the RUP is to be able to produce high quality software. This cannot be achieved if the inappropriate role is used. This study proposes the number of roles needed for software development using RUP by creating simulation models. The purpose of simulation model is to determine the appropriate role based on each type of project. The results of the simulation model are in the form of a percentage of each role needed, the project manager has the largest percentage compared to other roles, which is 76% in small-scale projects and 60% in large-scale projects.

Keywords: Rational Unified Process (RUP), Simulation Model, Project Management, Symphony.NET

1. Introduction

Pengembangan perangkat lunak dikaitkan dengan perancangan, penulisan, pengujian, implementasi dan pemeliharaan [1]. Hal tersebut membentuk proses pengembangan perangkat lunak sebagai siklus pengembangan atau yang disebut dengan 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 [2]. 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 dibahas pada penelitian ini adalah *Rational Unified Process* (RUP). RUP merupakan salah satu model pengembangan perangkat lunak berulang, yang memberikan tugas dan tanggung jawab dalam satu organisasi untuk memastikan produksi perangkat lunak berkualitas tinggi, yang berarti dapat memenuhi kebutuhan pengguna dengan jadwal dan anggaran yang dapat diprediksi [3]. Pada RUP anggota tim yang terlibat dalam proyek dipandu untuk mengikuti setiap langkah yang didasarkan pada setiap elemen. Terdapat tiga elemen dasar pada kerangka kerja RUP, yaitu kegiatan, peran dan artefak. Pemilihan ketiga elemen tersebut harus disesuaikan dengan proyek perangkat lunak. Setiap proyek berisi sekelompok anggota tim yang bertugas terhadap satu atau lebih peran. Setiap peran berpartisipasi terhadap satu atau lebih kegiatan dan setiap kegiatan menghasilkan satu atau lebih artefak [4].

RUP digunakan oleh berbagai perusahaan dan berbagai sektor industri, terutama industri perangkat lunak komputer dan industri pelayanan teknologi informasi. RUP juga diadopsi oleh industri kecil yang memiliki pegawai dibawah 50 orang [5]. Pada pengembangan industri perangkat lunak, RUP mendukung penggunaan tim kecil atau tim besar. Terdapat berbagai ukuran pengembangan proyek perangkat lunak, yaitu proyek dengan skala kecil, proyek skala menengah dan proyek skala besar [4]. Pengembangan perangkat lunak 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. Terutama pada penggunaan RUP, penerapan kerangka kerja dan peran yang dibutuhkan banyak yang terlewatkan karena diimplementasikan oleh tim kecil [6]. Terdapat beberapa kekurangan dari RUP terkait dengan manajemen sumber daya manusia, manajemen komunikasi dan manajemen kontrak [3].

Berdasarkan permasalahan tersebut diatas, dibutuhkan perencanaan proyek untuk menentukan jumlah dari setiap peran pada RUP. Oleh karena itu, penelitian ini mengusulkan sebuah model simulasi untuk pengembangan proyek menggunakan RUP. Simulasi dilakukan untuk mengetahui peran yang sesuai pada setiap jenis proyek tanpa mengabaikan fungsi penting dari RUP. Pembuatan model simulasi dilakukan pada komputer dengan menggunakan alat bantu simulator yaitu Symphony.NET [7].

2. Research Method

Rational Unified Process (RUP) merupakan salah satu produk pengembangan perangkat lunak yang ditawarkan oleh IBM *Rational*. RUP bersifat serial dalam skala besar, berulang dalam skala kecil dan menghasilkan rilis bertahap dari waktu ke waktu [1]. 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 [8].

Pada Tahap pengembangan RUP (Gambar 1) terdapat struktur dinamis dan struktur statis, Struktur dinamis pada dimensi horizontal mewakili dimensi waktu dari proses. Ini menunjukkan bagaimana proses dinyatakan dalam hal siklus, fase dan iterasi selama siklus hidup proyek [8]. Struktur dinamis terdiri dari beberapa fase, yaitu:

1. *Inception*

Inception adalah tahap untuk menentukan ruang lingkup dengan mengidentifikasi semua entitas eksternal yang akan berinteraksi dengan sistem. Hasil dari fase ini adalah dokumen persyaratan dan model *usecase*.

2. *Elaboration*

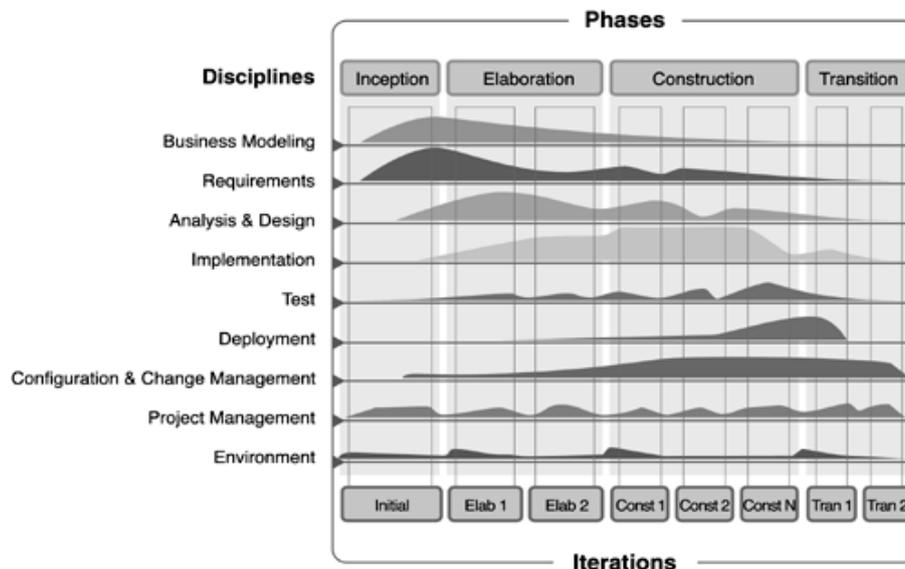
Elaboration adalah tahap untuk menganalisis dan memahami keseluruhan sistem, kemudian dibangun menjadi arsitektur. Hasil dari fase ini adalah model *use-case*, *prototype* arsitektur yang dapat dieksekusi.

3. Construction

Construction adalah tahap yang berisi implementasi perancangan yang dihasilkan pada fase sebelumnya (*elaboration*). Hasil dari fase ini adalah produk perangkat lunak beserta panduan pengguna.

4. Transition

Transition adalah fase untuk merilis perangkat lunak. Hasil dari fase ini adalah aktivitas yang diperlukan untuk menempatkan perangkat lunak ke pengguna.



Gambar 1. Tahap Pengembangan Pada Rational Unified Process (RUP) [8]

Fase tersebut dijalankan secara berurutan. Pekerjaan dikelompokkan ke dalam aktivitas logis yaitu disiplin yang dilakukan secara iteratif sepanjang fase [1], 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 Mangement, Project Management dan Environment. Kerangka kerja tersebut memungkinkan organisasi pengembangan perangkat lunak dari semua ukuran untuk bekerja. Pada RUP terdapat beberapa jenis proyek yaitu *Ganymede Project* yaitu proyek skala kecil dan *Mars Project* yaitu proyek skala besar.

2.1. Proyek Skala Kecil (*Ganymede Project*)

Disebut dengan proyek kecil karena perangkat lunak yang dihasilkan berukuran kecil dan membutuhkan waktu yang cukup singkat untuk pembangunannya. Siklus pengembangan awal dari perangkat lunak baru dimana segala sesuatu termasuk persyaratannya harus dirancang dari awal. Proyek *Ganymede* hanya membutuhkan satu iterasi. Proyek berskala kecil memiliki satu iterasi pada setiap fase seperti pada Tabel 1.

Tabel 1. Jumlah iterasi pada *Ganymede Project*

| Phase | Number of Iteration | Explanation |
|--------------|---------------------|---|
| Inception | 1 | Iterasi untuk menghasilkan prototipe <i>proof-of-concept</i> atau antarmuka pengguna <i>mock-up</i> |
| Elaboration | 1 | Iterasi untuk menghasilkan prototipe arsitektur |
| Construction | 1 | Iterasi untuk membangun produk (ke rilis beta) |
| Transition | 1 | Iterasi untuk menyelesaikan produk (rilis produk lengkap) |

2.2. Protek Skala Besar (*Mars Project*)

Tabel 2 disebut dengan proyek besar karena perangkat lunak yang dihasilkan lebih kompleks dan belum pernah dibangun sebelumnya. Sehingga membutuhkan waktu yang lebih lama untuk pembangunannya.

Tabel 2. Jumlah iterasi pada Mars Project

| Phase | Number of Iteration | Explanation |
|--------------|---------------------|--|
| Inception | 1 | Kemungkinan menghasilkan prototype |
| Elaboration | 2 | 1 iterasi untuk megembangkan prototype arsitektur awal |
| | | 1 iterasi untuk megembangkan prototype baseline awal |
| Construction | 2 | 1 iterasi untuk membuatnya matang untuk pengujian beta |
| | | 1 iterasi untuk memaparkan system parsial |
| Transition | 1 | Dimulai dari rilis beta hingga produk penuh |

2.3. Penelitian Terkait

Sonia Thind dan Karambir [4] 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 [2] 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 [9] 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.

3. Hasil dan Pembahasan

3.1. 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 [8].

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 [8]. Oleh karena itu penentuan perencanaan awal proyek harus dilakukan dengan tepat. Tujuan dari pembuatan model simulasi ini adalah untuk mengetahui peran yang sesuai pada setiap jenis proyek.

3.2. Model Simulasi

Model simulasi yang dibuat adalah model simulasi pengembangan perangkat lunak menggunakan *Rational Unified Process* (RUP). Pembuatan model simulasi menggunakan *tools* Symphony.NET, yang didalamnya terdapat beberapa *library* lengkap beserta elemen-elemen yang dibutuhkan untuk pembuatan model simulasi. Pada RUP terdapat empat fase dan sembilan workflow di dalam setiap fase, masing-masing workflow akan diletakkan pada elemen *Task*, kemudian dihubungkan dengan setiap *resource* sesuai dengan tugas masing-masing. Semua elemen yang dibuat akan dihubungkan satu sama lain kemudian dilakukan konfigurasi. Simulasi dijalankan selama lima kali dengan menggunakan 50 jumlah proyek yang datang. Setelah itu model dilakukan verifikasi dan dijalankan hingga mendapatkan hasil yang diinginkan. Hasil dari proses simulasi tersebut dapat berupa laporan statistik dan grafik yang dapat digunakan untuk perencanaan pembangunan proyek.

3.4 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 dua bagian berdasarkan kompleksitas dan skala, antara lain: proyek skala kecil dan proyek skala besar.

Asumsi selanjutnya untuk menentukan jumlah dari setiap proyek tersebut, yaitu 70% untuk proyek skala kecil dan 30% untuk proyek skala besar. 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 menentukan durasi pengerjaan pada setiap fase yang dirigrkas pada Tabel 3 berikut.

Tabel 3. Durasi Pengerjaan Setiap Workflow

| Fase | Business Modeling | Requirements | Analysis & Design | Implementation | Test | Deployment |
|--------------|-------------------|---------------|-------------------|-----------------|---------------|-----------------|
| Inception | Uniform (5,7) | Uniform (3,5) | Uniform (3,5) | Uniform (1,2) | Uniform (2,3) | Uniform (1,2) |
| Elaboration | Uniform (5,7) | Uniform (7,9) | Uniform (7,9) | Uniform (3,4) | Uniform (2,3) | Uniform (1,2) |
| Construction | Uniform (1,2) | Uniform (2,3) | Uniform (3,5) | Uniform (10,15) | Uniform (8,9) | Uniform (2,3) |
| Transition | Uniform (1,2) | Uniform (1,2) | Uniform (1,2) | Uniform (3,5) | Uniform (5,7) | Uniform (10,15) |

3.5. Model Simulasi *Rational Unified Process (RUP)*

Model simulasi yang dibangun menggunakan alat simulasi komputer yaitu *Simphony.NET* [7]. *Library* pada *Simphony* yang digunakan untuk membuat model simulasi yang diusulkan terdiri dari *resource*, *file*, *capture*, *release*, *task*, *destroy*, *create*, *counter*, dan *probabilistic branch*. Gambar 2 dan Gambar 3 menggambarkan elemen pemodelan sumber daya dengan jumlah antriannya, yang terdiri dari:

- A. Proyek skala kecil: *project manager*, *project reviewer*, *implementer*, *itegrator*, *process engineer*, *test manager*, *system tester* dan *system administrator*.
- B. Pryek skala besar: *project manager*, *integrator*, *project reviewer*, *system analyst*, *process engineer*, *implementer*, *software architect*, *system administrator*, *test manager*, *user-interface designer*, *course developer*, *database designer* dan *system btester*.

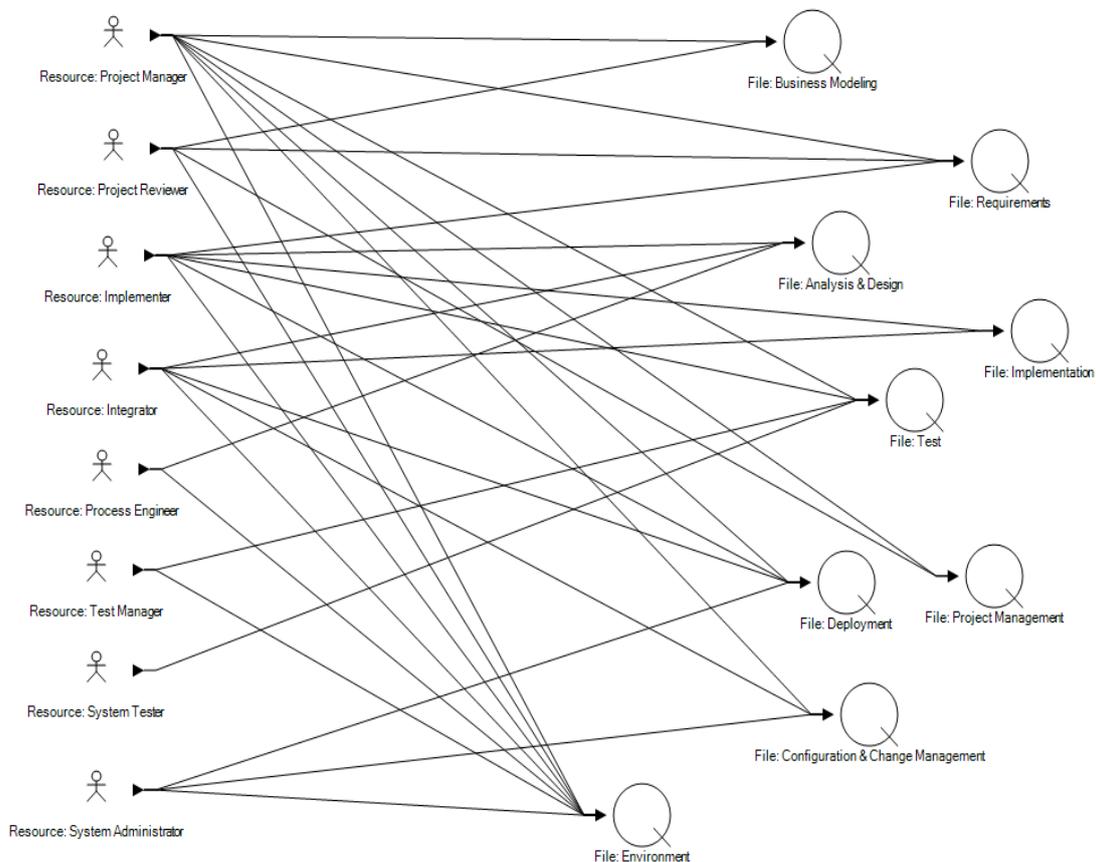
Gambar 2 merupakan model simulasi untuk *Resources* pada proyek skala kecil dan Gambar 3 untuk *Resources* pada proyek skala besar. Elemen yang digunakan pada model tersebut yaitu elemen *resource* yang berfungsi untuk mendefinisikan *resource* yang terlibat pada pengembangan perangkat lunak skala kecil. Kemudian elemen *file* pada model tersebut berfungsi sebagai tempat antrian *resource* apabila sedang menunggu giliran untuk dieksekusi pada saat model dijalankan. Elemen *file* mewakili setiap *workflow* pada RUP. Setiap *resources* dihubungkan dengan *file* yang sesuai dengan tugasnya. Pada proyek skala kecil elemen *resources* yang digunakan berjumlah delapan dan berjumlah tiga belas untuk proyek skala besar. Sedangkan Elemen *file* berjumlah sembilan sesuai dengan jumlah *workflow* pad RUP.

Model simulasi yang dibangun pada setiap jenis proyek dan tahap memiliki model yang sama, karena pada pengembangan perangkat lunak menggunakan RUP. Penerapan pada

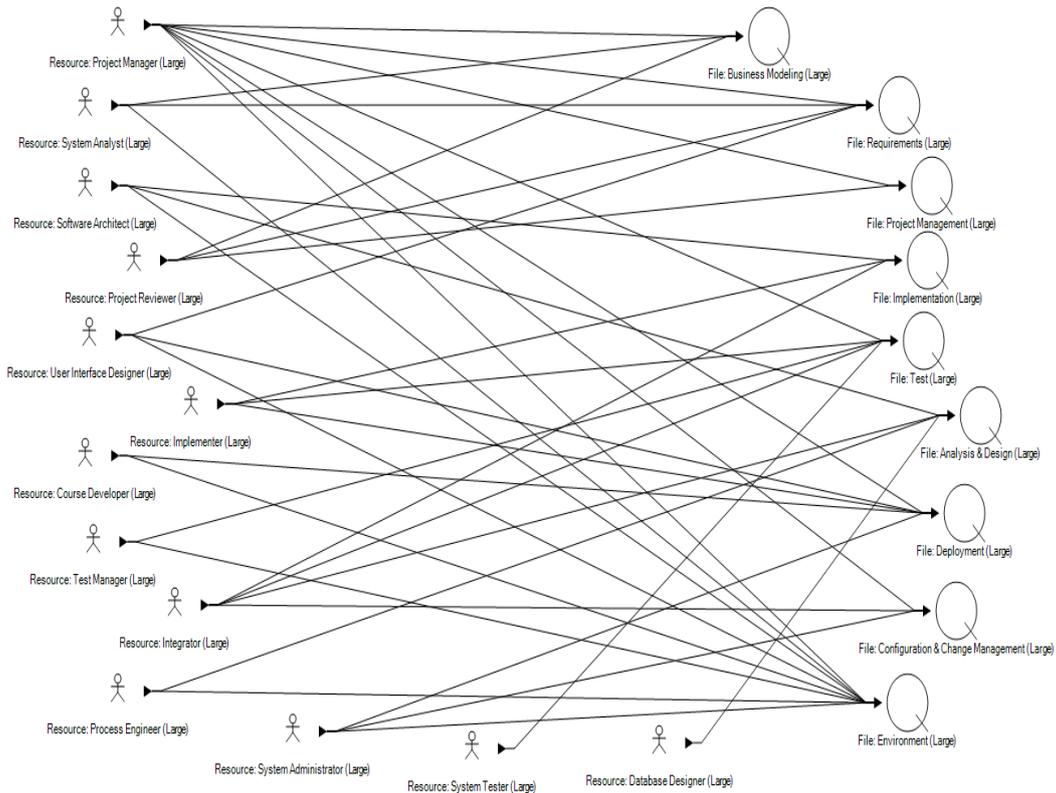
keempat tahap yaitu *inception*, *elaboration*, *construction* dan *transition* melibatkan semua workflow dan peran yang disediakan. Terdapat beberapa hal yang membedakan pada setiap tahap terhadap workflow dan peran. Perbedaan tersebut yaitu fokus utama pada setiap tahap yang sedang dikerjakan, sehingga waktu pengerjaan akan berbeda pada setiap tahap meskipun pada *workflow* yang sama.

Elemen yang digunakan pada model simulasi proyek skala kecil antara lain: elemen *create* berfungsi untuk membuat entitas (proyek yang masuk). Elemen *counter* berfungsi untuk mencatat setiap jumlah proyek yang melewatinya untuk masuk ke elemen selanjutnya. Elemen *probability branch* berfungsi untuk memasukkan jumlah persentase setiap jenis proyek. Elemen *capture* didefinisikan pada model berdasarkan *workflow*, yang berfungsi untuk memasukkan setiap peran yang bekerja pada masing-masing *workflow*. Elemen Release berfungsi untuk melepaskan server yang selesai digunakan oleh resource pada elemen *capture*. Elemen task yang berfungsi memasukkan waktu pengerjaan pada setiap *workflow* dengan menggunakan salah satu jenis distribusi. Setiap elemen memiliki konfigurasi untuk memasukkan data dari *design model*.

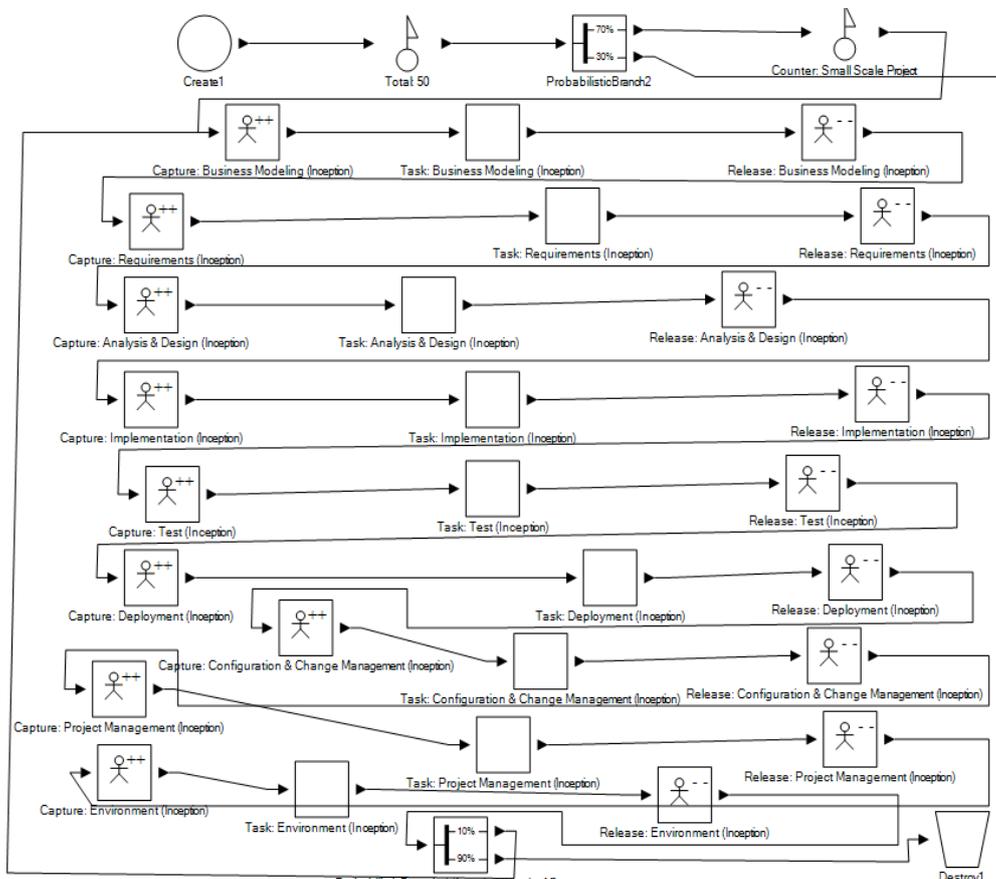
Beberapa elemen *probability branch* tersedia diantara setiap tugas sumber daya yang berbeda, tujuannya adalah untuk mensimulasikan kesalahan yang mungkin terjadi pada saat pengembangan. Unsur probabilitas memiliki dua cabang yaitu cabang di bagian atas dengan $Prob=0,1$ menunjukkan bahwa 10% tingkat kesalahan pada proyek sehingga harus mengulangi pada tahap sebelumnya untuk memperbaiki kesalahan tersebut. Kemudian cabang pada bagian bawah dengan $Prob=0,9$ menunjukkan bahwa 90% dari proyek tidak terdapat kesalahan setelah selesainya setiap tahap. Selain itu elemen *probability branch* terdapat pada awal model simulasi yang menggunakan dua cabang berdasarkan jenis proyek yang sudah ditentukan. Cabang pertama dengan $Prob=0,7$ menunjukkan bahwa 70% proyek yang masuk adalah proyek skala kecil. Cabang kedua dengan $Prob=0,3$ menunjukkan bahwa 30% proyek yang masuk adalah proyek skala besar. Gambar 4 menunjukkan elemen pemodelan untuk mensimulasikan model dengan konfigurasi probabilitas yang sudah disebutkan.



Gambar 2. Resource Modeling Proyek Skala Kecil



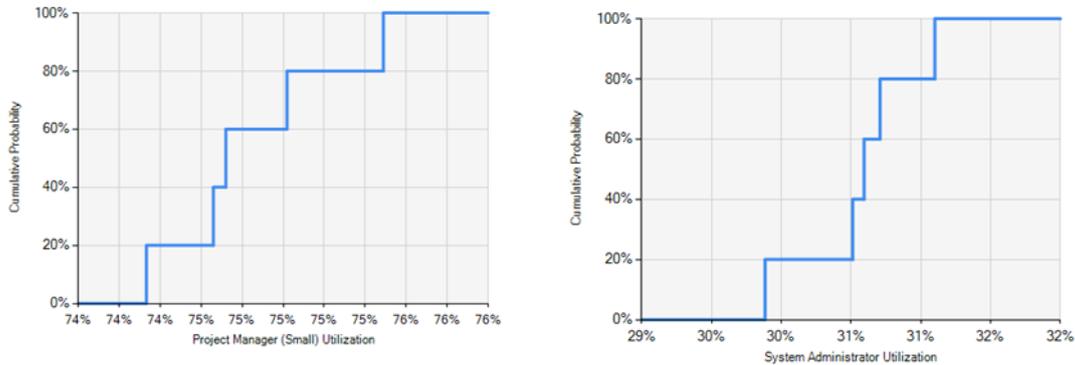
Gambar 3. Resource Modeling Proyek Skala Besar



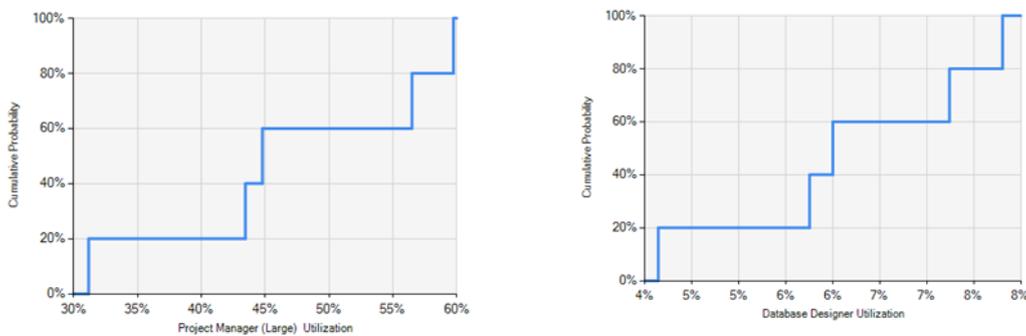
Gambar 4. Model Simulasi Rational Unified Process (RUP)

3.6. Analisis dan Hasil Simulasi

Setelah model dijalankan akan menghasilkan laporan statistic dan beberapa diagram. Hasil yang akan dilakukan Analisa yaitu diagram penggunaan *resource*. Diagram dari beberapa *resources* tersebut akan ditampilkan pada Gambar 5 dan Gambar 6.



Gambar 5. Pemanfaatan Project Manager dan System Administrator Proyek Skala Kecil



Gambar 6. Pemanfaatan Project Manager dan Database Designer Proyek Skala Besar

Setiap *resources* memiliki hasil dalam bentuk diagram yang berisi persentase penggunaan. Hasil keseluruhan dari penggunaan *resources* pada setiap jenis proyek dapat diringkas dalam Tabel 4 berikut.

Tabel 4. Penggunaan Resources Setiap Jenis Proyek

| Resource Name | Average Utilization for: | |
|-------------------------|--------------------------|--------------|
| | Ganymede Project | Mars Project |
| Project Manager | 76% | 60% |
| Integrator | 56,7% | 32,5% |
| Project Reviewer | 31,5% | 16,8% |
| System Analyst | - | 14,3% |
| Process Engineer | 18,7% | 10,4% |
| Implementer | 69,0% | 29,7% |
| Software Architect | - | 20,7% |
| System Administrator | 32% | 21,0% |
| Test Manager | 19,7% | 13,5% |
| User-Interface Designer | - | 19,6% |
| Course Developer | - | 14,3% |
| Database Designer | - | 8% |
| System Tester | 12,8% | 9,3% |

Berdasarkan Tabel 4 dapat disimpulkan bahwa dalam pengembangan perangkat lunak menggunakan RUP peran yang paling banyak dibutuhkan adalah *project manager*, pada proyek kecil maupun proyek besar. Hal tersebut dikarenakan *Peran project manager* merangkap banyak peran lainnya yang bekerja pada workflow berdurasi besar.

Tabel 5. Perolehan Statistik Hasil Simulasi

| Jenis Proyek | Received Project | Delivered Project | Average |
|--|------------------|-------------------|---------|
| Proyek Skala Kecil (Ganymede Project) | 39 | 39 | 41,702 |
| Proyek Skala Besar (Mars Project) | 11 | 11 | 26,841 |

Tabel 5 menunjukkan jumlah proyek yang diterima dan dikirimkan pada setiap jenis proyek. Hasil pada table tersebut menunjukkan bahwa jumlah proyek yang diterima dan dikirim memiliki jumlah yang sama, berarti dalam pengerjaan proyek dan pemanfaatan resources sudah sesuai. Total dari seluruh proyek yang dikirim juga sesuai dengan jumlah proyek yang dimasukkan pada saat awal model simulasi dibangun, yaitu proyek skala kecil dengan 39 proyek dan proyek skala besar 11 proyek sehingga total keseluruhan proyek yaitu 50. Hasil tersebut sesuai berdasarkan persentase yang sudah dimasukkan. Proyek skala kecil memiliki persentase yang lebih besar dibandingkan dengan proyek skala besar.

4. Kesimpulan

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 peran (*roles*), durasi, dan jenis proyek. Pada penelitian ini terdapat dua jenis proyek yaitu, proyek dengan skala kecil (*Ganymede Project*) dan skala besar (*Mars Project*). Peran (*roles*) yang digunakan yaitu *reduced model resources* untuk proyek skala kecil dan *based model resources* untuk proyek skala besar. Hasil dari pembuatan simulasi ini dapat membantu dan mempermudah pengembang perangkat lunak yang menggunakan RUP dalam perkiraan peran dan jumlah anggota tim yang sesuai dengan jenis proyek.

Referensi

- [1] S. W. Ambler, "A manager's introduction to the Rational Unified Process (RUP)," *Version: December*, vol. 4, p. 2005, 2005.
- [2] Y. Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle," *Int. J. Eng. Technol.*, vol. 2, no. 5, pp. 2049–3444, 2012.
- [3] P. Alexandra and F. Monteiro, "Universidade do Minho Escola de Engenharia," 2014.
- [4] S. T. ind, Karambir, "A Simulation Model for the Spiral Software Development Life Cycle," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 03, no. 05, pp. 3823–3830, 2015.
- [5] iDatatables, "iDatatables," 2018. .
- [6] I. Sommerville, *Software Engineering*. 2010.
- [7] University of Alberta, "Symphony.NET 4.6," *University of Alberta*, 2018. [Online]. Available: <https://www.ualberta.ca/engineering/research/groups/construction-simulation/symphony>.
- [8] P. Kroll and P. Kruchten, *The Rational Unified Process Made Easy*. 2003.
- [9] A. Singh and P. J. Kaur, "a Simulation Model for Incremental Software Development Life Cycle Model," vol. 8, no. 7, pp. 126–132, 2017.

