

Identifikasi Kebutuhan Pengguna menggunakan *Design Thinking* yang digabungkan dengan *Framework Scrum* dalam Pembangunan *Website* “Kerjayuk” untuk Mahasiswa Universitas Telkom

Hema Ditania¹, Dana Sulisty Kusumo², Asristyo Hadikusuma³

^{1,2,3} Universitas Telkom, Bandung

¹hemaditania@student.telkomuniversity.ac.id,

²danakusumo@telkomuniversity.ac.id,

³tyokusuma@telkomuniversity.ac.id,

Abstrak

Kebutuhan akan pekerjaan sudah menjadi salah satu yang paling dicari di Indonesia. Selain itu pekerjaan menjadi salah satu tren di Indonesia dengan ciri khasnya yang bersifat fleksibel. Namun dalam pengaplikasiannya terdapat permasalahan yaitu sulitnya menemukan informasi mengenai pekerjaan yang sesuai dengan keahlian dan pendidikan seorang pencari kerja atau pelamar. Maka dari itu pada penelitian ini akan dilakukan pembangunan sebuah platform yang menyediakan lapangan pekerjaan yaitu Kerjayuk yang berbasis *Website*. Kerjayuk ini menggabungkan *framework Scrum* dan metode *Design Thinking* dalam pengaplikasiannya. Namun terdapat permasalahan yaitu *Scrum* hanya berfokus pada proses pengembangan perangkat lunak. Tim pengembang hanya berfokus pada bagaimana proses pengimplementasian perangkat lunak tanpa mengetahui seluk beluk masalah yang dijadikan sebagai acuan pengembangan. Maka dari itu digunakan *Design Thinking* yang bertujuan untuk mengetahui kebutuhan pengguna, seperti apa kebutuhan pengguna yang diteliti, serta bagaimana mengidentifikasi kebutuhan pengguna yang digunakan sebagai acuan dalam pengembangan perangkat lunak. Kerjayuk menggunakan penggabungan antara *Design Thinking* dan *Scrum* yang bertujuan untuk melakukan pengembangan dengan mengidentifikasi permasalahan lalu dapat fokus kepada pengimplementasiannya.

Kata kunci: *Scrum, Design Thinking, Kerjayuk, Website.*

Abstract

The need for jobs has become one of the most sought after in Indonesia. In addition, work is one of the trends in Indonesia with its characteristic being flexible. However, in its application there are problems, namely the difficulty of finding information about jobs that match the skills and education of a job seeker or applicant. Therefore, this research will conduct the development of a platform that provides employment, namely Web-based Kerjayuk. This work combines the Scrum framework and Design Thinking methods in its application. However, there is a problem that Scrum only focuses on the software development process. The development team only focuses on how the software implementation process is without knowing the details of the problem which is used as a development reference. Therefore, Design Thinking is used which aims to determine user needs, such as what the needs of the user being studied are, and how to identify user needs that are used as a reference in software development. Kerjayuk uses a combination of Design Thinking and Scrum which aims to develop by identifying problems and then can focus on their implementation.

Keywords: *Scrum, Design Thinking, Kerjayuk, Website.*

1. Pendahuluan

Pada bagian ini akan dibahas mengenai latar belakang penelitian, rumusan masalah, tujuan, rencana kegiatan dan jadwal kegiatan yang akan dilakukan.

1.1 Latar Belakang

Design Thinking umumnya didefinisikan sebagai proses analitik dan kreatif yang melibatkan seseorang dalam peluang untuk bereksperimen, membuat dan membuat prototipe model, mengumpulkan umpan balik, dan mendesain ulang [7]. Kerjayuk merupakan sebuah produk berbentuk jasa yang akan dibangun, dimana produk ini akan membangun sebuah platform berbasis *Website* sebagai sarana pencarian informasi lowongan kerja. Kerjayuk akan menghubungkan perusahaan dengan para pencari kerja melalui *Website* dimana pengguna dapat berperan sebagai pencari kerja dan penyedia kerja.

Kerjayuk dibuat untuk mempermudah mahasiswa Universitas Telkom dalam mencari pekerjaan. Dikarenakan sudah banyak platform lain untuk mencari pekerjaan, tetapi platform yang sudah ada terbuka untuk umum, tetapi Kerjayuk mengutamakan mahasiswa Universitas Telkom dalam mencari pekerjaan. Kerjayuk dapat diakses oleh semua orang, tetapi platform ini mengutamakan lulusan atau mahasiswa Universitas Telkom dalam mencari pekerjaan. Permasalahan yang dihadapi dalam Kerjayuk ini adalah bagaimana melakukan pengembangan sesuai dengan kebutuhan pengguna dan menghasilkan produk yang

lebih unggul daripada platform yang sudah ada sehingga dapat menjadi alternatif mahasiswa Universitas Telkom dalam melakukan pencarian pekerjaan. Maka dari itu dibuatlah Kerjayuk yang bertujuan untuk memfasilitasi mahasiswa Universitas Telkom dalam mencari pekerjaan. Kerjayuk juga menggunakan mahasiswa Universitas Telkom dalam proses pengembangan agar dapat menjadi produk yang sesuai dengan kebutuhan pengguna. Dalam proses pembangunan *Website* Kerjayuk, digunakan metode pengembangan perangkat lunak *Scrum* dari *Agile Software Development* atau sering disebut dengan *Framework Scrum*.

Scrum digunakan karena bersifat cepat, mudah beradaptasi, meningkatkan kualitas produk, implementasi menggunakan iterasi yang *Incremental* dan mudah beradaptasi serta menerima perubahan yang sewaktu waktu dapat berubah. Selain itu, *Framework Scrum* juga memberikan kepuasan kepada pengguna karena mempunyai waktu pengerjaan yang jelas serta respon yang beradaptasi pada permintaan. *Framework Scrum* juga tidak akan merubah jadwal serta fitur fitur yang akan ditentukan atau dikerjakan, karena jadwal sudah dibuat pada tahap awal pengembangan. Selain itu *Scrum* juga digunakan karena mempunyai estimasi pekerjaan yang sudah jelas dan terstruktur serta pengerjaan pekerjaan dapat diselesaikan dengan lebih terstruktur dan jelas [10]. Terdapat alasan mengapa mengintegrasikan *Design Thinking* dan *Scrum* dalam pengembangan Kerjayuk, karena *Design Thinking* dan *Scrum* memiliki tujuan yang sama, yaitu untuk membuat produk yang berfokus kepada pengguna dan berfokus kepada pengembangan produk. Selain itu juga *Scrum* digabungkan dengan *Design Thinking* karena *Scrum* tidak mendefinisikan pencarian kebutuhan pengguna, sedangkan *Design Thinking* berisi tahapan tahapan yang mendefinisikan serta melakukan pencarian kebutuhan pengguna secara mendetail. Meskipun memiliki tujuan yang sama, *Design Thinking* mempunyai kelemahan yaitu tidak menghasilkan produk yang dapat digunakan, hanya menghasilkan produk yang berupa prototipe [1].

Tetapi *Scrum* mempunyai kelemahan dalam mengidentifikasi kebutuhan pengguna, yaitu saat proses pembangunan produk *scrum* hanya berfokus kepada proses serta solusi masalah dalam pengembangan perangkat lunak yang berakibat tidak diperuntukkan untuk kebutuhan pengguna [4]. Oleh karena itu, tim pengembangan perangkat lunak menggabungkan metode *Scrum* dengan *Design Thinking*, *Design Thinking* berfokus pada kebutuhan pengguna, mengidentifikasi serta menganalisis masalah yang ada. hal ini karena cara kerja *Design Thinking* adalah dengan mengamati serta menganalisis suatu masalah yang sedang dihadapi oleh pengguna, lalu dapat menciptakan suatu produk perangkat lunak yang dapat digunakan [2]. Dengan menggabungkan kedua metode yaitu *Scrum* dan *Design Thinking*, diharapkan dapat melengkapi kekurangan *scrum* yang hanya berfokus pada pemecahan atau solusi suatu masalah saat dilakukan pengembangan perangkat lunak [2]. Karena tujuan dari *design thinking* adalah dengan mengamati serta menganalisis suatu masalah yang sedang dihadapi oleh pengguna, lalu dapat menciptakan suatu produk perangkat lunak yang dapat digunakan [2].

1.2 Topik dan Batasannya

Berdasarkan latar belakang masalah yang sudah diuraikan, dapat dirumuskan perumusan masalah dalam penelitian sebagai berikut:

1. Bagaimana tahapan tahapan penggabungan antara metode *Design Thinking* dengan *Framework Scrum* dalam membangun platform penyedia informasi lowongan pekerjaan Kerjayuk?
2. Bagaimana pengaruh penggabungan *Framework Scrum* dengan metode *Design Thinking* dalam membangun platform penyedia informasi lowongan pekerjaan Kerjayuk?

Batasan masalah yang akan dibahas dalam penelitian ini antara lain:

1. Kebutuhan pengguna yang akan diteliti adalah mahasiswa/i Telkom *University*.
2. Menggunakan sistem *Sprint* dalam proses membangun platform penyedia pekerjaan Kerjayuk.
3. Peneliti sebagai *Development Team* dalam proses pembangunan platform Kerjayuk.
4. Target penelitian merupakan mahasiswa Universitas Telkom yang sedang mencari pekerjaan.
5. Metode penelitian yang digunakan dalam pembangunan ini adalah *Framework Scrum* yang digabungkan dengan metode *Design Thinking*.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disebutkan, maka dapat diketahui tujuan dari penelitian ini adalah:

1. Mengetahui tahapan yang dibutuhkan dalam pembangunan *website* Kerjayuk yang menggunakan menggunakan *Framework Scrum* yang digabungkan dengan metode *Design Thinking*.
2. Mengetahui pengaruh penggabungan metode *Design Thinking* dan *Framework Scrum* dalam pembangunan aplikasi Kerjayuk.

2. Organisasi Tulisan

2.1 Design Thinking Scrum

Design Thinking mengajarkan bagaimana mengidentifikasi kebutuhan pengguna secara mendetail sehingga produk yang dibangun dapat sesuai dengan kebutuhan pengguna [1]. Mirip dengan *Design Thinking*, *Scrum* juga berorientasi kepada pengguna tetapi lebih berfokus atau memfokuskan pada pembuatan atau pengembangan solusi [5]. Dengan melibatkan pengguna dalam proses pengembangan pengguna dalam proses pengembangan, telah terbukti tidak hanya meningkatkan kesuksesan produk tetapi juga mengurangi biaya pengembangan dengan memfokuskan pengembangan pada masalah yang paling penting sejak awal proses pengembangan [9]. Selain metode *Design Thinking*, *Framework Scrum* juga digunakan karena pendekatan *Scrum* yang dapat dilakukan secara berulang memastikan fleksibilitas dalam proses pengembangan serta dapat terus mengintegrasikan *feedback* pengguna (Schwaber and Sutherland, 2017). Dalam pelaksanaan iterasi serta *user stories* dapat menempatkan pengguna dan kebutuhan pengguna pada pusat proses pengembangan dan memastikan produk yang dihasilkan sesuai dengan target pengguna (Schwaber, 1995). Berikut merupakan alur metode penggabungan *scrum* dan *design thinking* yang dibagi menjadi 3 tahap [1]. Beberapa tahapan dalam metode *Design Thinking Scrum* antara lain:

1. Design Thinking Mode

Design Thinking Mode dibagi menjadi empat bagian yaitu *Understand*, *Observe*, *Synthesize* dan *Ideate* yang masing masing tahap bertujuan untuk mendekati diri kepada pengguna dengan cara memahami perumusan masalah, pemahaman masalah, pencetusan ide serta pengembangan ide atau solusi dari permasalahan. Pada tahap ini peran pengguna sangat penting, karena pengguna sebagai pusat dari penelitian dan juga tahap ini yaitu *Understand*, *Observe*, *Synthesize* dan *Ideate* merupakan penggalian masalah dari pengguna dan *Product Owner*. Walaupun sudah terdapat *Product Owner* sebagai pengguna, tetapi pengguna selain *Product Owner* juga diperlukan dengan tujuan untuk mengetahui berbagai sisi agar dapat melakukan pengamatan dengan mendalam dan tidak hanya melihat dari satu pengguna atau dari *Product Owner* saja.

2. Initial Development Mode

Initial Development Mode dibagi menjadi dua bagian yaitu *Prototype* dan *Testing*. Tahap ini bertujuan untuk menggambarkan atau pengaplikasian dari ide yang sudah ditemukan. Penggambaran solusi ini bersifat produk yang belum siap diluncurkan atau produk belum siap jadi. Pada tahap ini *Product Owner* dan pengguna melakukan pengujian *Prototype* agar menghasilkan sudut pandang yang beragam serta dapat mengetahui kelebihan serta kelemahan produk dengan lebih mendetail.

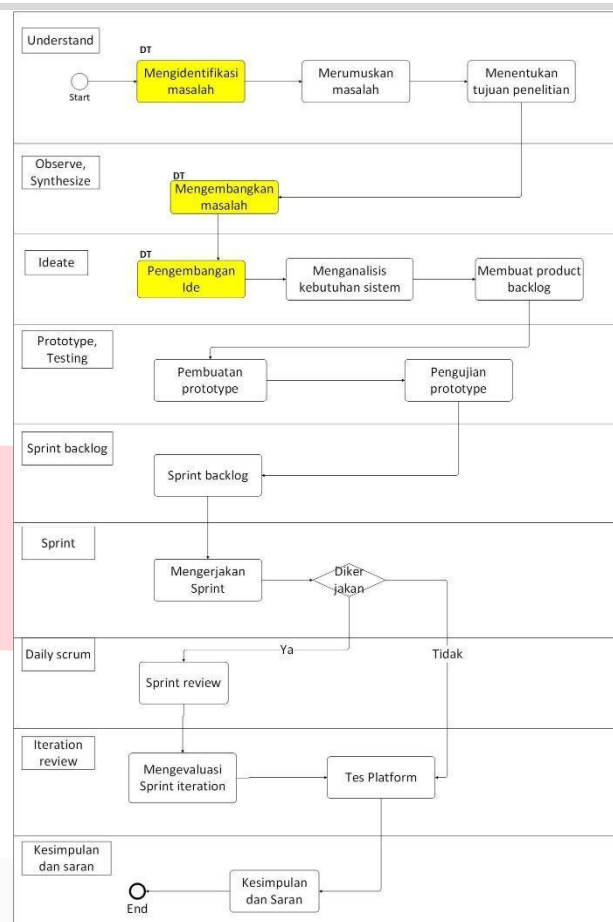
3. Fully Integrated Mode

Fully Integrated Mode dibagi menjadi empat tahap yaitu *Sprint*, *Sprint Planning Meeting*, *Daily Scrum* dan *Iteration Review*. Tahap ini merupakan tahap pengembangan perangkat lunak yang menggunakan *Sprint* dalam pengaplikasiannya. Pada tahap ini *Product Owner* dibutuhkan sebagai orang yang menyetujui produk yang akan diluncurkan, apakah produk sesuai dengan kebutuhan atau tidak dan apakah produk siap diluncurkan atau tidak.

3. Alur Penelitian

3.1 Design Thinking Scrum

Berikut merupakan tata cara, metode, urutan atau langkah-langkah yang dilakukan selama proses pengembangan perangkat lunak serta untuk menyelesaikan sebuah penelitian atau riset yang sedang dilakukan. Dalam pengembangan perangkat lunak Kerjayuk ini menggunakan metode *Scrum* dan metode *Design Thinking*.



Gambar I. Alur kerja Penelitian *Design Thinking Scrum*

Berikut beberapa tahapan yang akan dilakukan, antara lain:

3.1.1 *Design Thinking Mode*

3.1.1.1 *Understand.*

Pada tahap ini melakukan pengidentifikasian masalah kepada pengguna yang direpresentasikan menggunakan *Empathy Map*. *Empathy Map* digunakan dengan tujuan untuk memahami pengguna mengenai permasalahan yang dihadapi. *Empathy Map* didapat setelah melakukan observasi kepada calon pengguna yang dituju. Pada tahap ini dilakukan pemahaman kepada pengguna serta *Product Owner*, dengan tujuan mengetahui apa berbagai sudut pandang dalam melakukan pemahaman terhadap permasalahan yang dihadapi.

3.1.1.2 *Observe dan Synthesize*

Setelah mengetahui permasalahan yang ada, maka selanjutnya akan membuat *User Persona*. Peneliti membuat *User Persona* dari pemahaman serta permasalahan yang telah dibuat. *User Persona* didapatkan dari melakukan wawancara serta pemahaman kepada pengguna dan *Product Owner*. Pengguna juga dibutuhkan karena saat menentukan *User Persona* pengguna yang diamati tidak hanya satu saja melainkan lebih dari satu pengguna agar dapat menghasilkan *User Persona*.

3.1.1.3 *Ideation*

Setelah memahami lebih lanjut mengenai kebutuhan pengguna dengan dibuatnya *User Persona*, maka selanjutnya adalah mengembangkan ide atau solusi dari permasalahan yang ada. Tahap ini berisi *Requirement* yang bertujuan untuk mengetahui batasan serta gambaran yang ada dalam perangkat lunak. Pada tahap ini dibutuhkan *Product Owner* untuk menyetujui serta Menyusun *Requirement* yang akan digunakan. *Requirement* terdiri dari dua jenis, yaitu *Functional Requirement* dan *Non Functional Requirement*. *Functional Requirement* mencakup beberapa hal, antara lain:

1. Sistem dapat membuat akun (untuk *Seeker* dan *Company*).
2. Sistem dapat mencari pekerjaan berdasarkan kata kunci.
3. Sistem dapat menampilkan daftar pekerjaan yang telah dilamar.
4. Sistem dapat menerima laporan lowongan yang sudah tidak aktif.
5. Sistem dapat menerima laporan penyalahgunaan aplikasi.
6. Sistem dapat menampilkan informasi pribadi (untuk *Seeker* dan *Company*).
7. Sistem dapat memperbaharui informasi pribadi (untuk *Seeker* dan *Company*).
8. Sistem dapat menambah lowongan pekerjaan.
9. Sistem dapat menampilkan daftar calon pekerja atau kandidat.
10. Sistem dapat menampilkan daftar laporan penyalahgunaan aplikasi dari *Seeker* maupun *Company*.

Selain *Functional Requirement*, terdapat *Non Functional Requirement*. *Non Functional Requirement* terdiri dari beberapa hal, antara lain:

1. Sistem memiliki *User Interface* yang menarik.
2. Sistem memiliki *User Interface* yang mudah dipahami dan digunakan.
3. Sistem dapat diakses melalui semua *Operating System*.
4. Sistem berbentuk *Website*.
5. Sistem dapat diakses menggunakan berbagai gawai seperti *Handphone* dan *Laptop*.

Berdasarkan *Functional Requirement* dan *Non Functional Requirement*, terbentuklah *User Story* yang bertujuan untuk menggambarkan serta memperjelas alur kerja dari suatu sistem. *User Story* terdiri dari beberapa bagian, yaitu *Seeker*, *Company* dan admin. Berikut beberapa bagian dari *User Story* antara lain:

1. Dapat membuat akun untuk *Seeker* dan *Company*.
2. Dapat melihat lowongan yang tersedia untuk *Seeker*.
3. Dapat melakukan pendaftaran atau dapat melamar pada lowongan pekerjaan yang tersedia.
4. Dapat melihat lowongan pekerjaan yang telah didaftarkan atau dilamar.
5. Dapat melakukan laporan lowongan pekerjaan yang sudah tidak aktif.
6. Dapat melakukan laporan penyalahgunaan yang dilakukan oleh suatu perusahaan.
7. Dapat mengelola informasi pribadi.

1.1.2 *Initial Development Mode*

1.1.2.1 *Prototype dan Testing*

Setelah menemukan solusi dari perumusan masalah yang sudah terdefinisi teridentifikasi tahap tahap yang dibutuhkan untuk mengetahui kebutuhan pengguna, selanjutnya akan dilakukan pembuatan *prototype*. *Prototype* yang dibuat bersifat *high fidelity* yang menggunakan figma dalam pengaplikasiannya. Setelah pembuatan *Prototype* maka dilanjutkan dengan melakukan pengujian. Pengujian ini dilakukan oleh *Product Owner* serta pengguna.

1.1.3 *Fully Integrated Mode*

1.1.3.1 *Sprint Backlog*

Setelah *Prototype* dan *Testing* dilakukan maka disusunlah *Sprint Backlog*. *Sprint Backlog* didapatkan melalui *Product Backlog* atau *Requirement* yang sudah ditentukan pada tahap sebelumnya. Penyusunan *Sprint Backlog* ini dilakukan oleh tim Kerjayuk dan *Product Owner* setelah melakukan pencarian kebutuhan yang mendalam atau secara mendetail. Setelah terdapat *Product Backlog*, tim Kerjayuk membuat *Sprint Backlog* dengan tujuan agar pengerjaan menjadi lebih terdefinisi serta terstruktur. Terdapat dua versi *Sprint Backlog*, yaitu *Sprint Backlog Design Thinking + Scrum* dan *Sprint Backlog Scrum*. Pada penelitian kali ini berencana memakai lima *Sprint* yang masing masing mempunyai interval selama satu minggu. Rencana *Sprint Backlog* antara lain:

Tabel I. Perbedaan *Sprint Backlog DT + Scrum* dengan *Scrum*

No.	<i>Sprint Backlog DT + Scrum</i>	<i>Sprint Backlog Scrum</i>
1.	Membuat akun (untuk <i>seeker</i> dan <i>company</i>).	Membuat akun.
2.	Fitur untuk mencari pekerjaan berdasarkan kata kunci.	Fitur untuk mendaftar pekerjaan.
3.	Fitur untuk menampilkan daftar pekerjaan yang telah dilamar.	Fitur untuk menambah lowongan pekerjaan.
4.	Fitur untuk menerima laporan lowongan yang sudah tidak aktif.	Fitur untuk memperbarui informasi pribadi.
5.	Fitur untuk menerima laporan penyalahgunaan aplikasi	Fitur untuk menampilkan informasi pribadi.
6.	Fitur untuk menampilkan serta memperbarui informasi pribadi (untuk <i>seeker</i> dan <i>company</i>).	Fitur untuk melakukan pelaporan lowongan pekerjaan.
7.	Fitur untuk menambah lowongan pekerjaan.	
8.	Fitur untuk menampilkan daftar calon pekerja atau kandidat.	
9.	Fitur untuk menampilkan daftar laporan penyalahgunaan aplikasi dari <i>seeker</i> maupun <i>company</i>	

Dari hasil tabel mengenai perbedaan *Sprint Backlog DT + Scrum* dengan *Scrum*, menunjukkan bahwa hasil identifikasi pengguna lebih detail serta jelas jika menggunakan penggabungan metode *Design Thinking* dengan *Framework Scrum*. *Sprint Backlog* yang akan digunakan adalah yang penggabungan antara metode *Design Thinking* dengan *Framework Scrum*. *Sprint Backlog* ini bersifat tetap atau tidak berubah, karena salah satu sifat *Framework Scrum* adalah fitur atau jadwal yang telah ditetapkan tidak berubah [10]. Dibandingkan dengan menggunakan *Framework Scrum* tidak dapat mengetahui atau mengidentifikasi kebutuhan pengguna secara detail, maka dari itu dilakukan penggabungan antara metode *Design Thinking* dengan *Framework Scrum* untuk mengetahui kebutuhan pengguna lebih mendetail serta terperinci.

1.1.3.1 *Sprint*

Setelah membuat *Sprint Backlog*, maka selanjutnya adalah pengerjaan *Sprint*. Pengerjaan *Sprint* yang direncanakan menjadi lima *sprint* dengan interval masing masing *Sprint* adalah satu minggu. *Sprint* ini dilakukan dengan bantuan *tools* Trello dalam mendefinisikan serta menjadwalkan *Sprint* yang akan, sedang dan telah dilakukan.

1.1.3.2 *Daily Scrum*

Daily Scrum dilakukan setiap hari selama pengerjaan *Sprint*. Pada penelitian ini *Daily Scrum* dilakukan pada malam hari dengan tim *developer* masing masing mengatakan apa yang telah dikerjakan kemarin, hari ini serta apa kesulitan yang dihadapi saat sedang melakukan pengembangan.

1.1.2.4 Iteration Review

Setelah dilakukan *Daily Scrum* pada akhir minggu setelah satu *Sprint* selesai dilakukan maka akan melakukan *Sprint Retrospective* yang ditujukan untuk mengevaluasi kinerja pengembangan serta kinerja tim yang terjadi dalam satu minggu terakhir. Pada tahap ini *Product Owner* digunakan untuk menyetujui produk akhir yang telah dikembangkan, serta memberi tanggapan mengenai produk yang dibuat apakah produk sudah sesuai kebutuhan dan dapat diluncurkan.

4. Evaluasi

Pada evaluasi penelitian ini melakukan pengumpulan data menggunakan metode kuantitatif dan metode kualitatif. Metode kuantitatif dilakukan dengan cara menyebarkan kuesioner yang berisi lima pertanyaan. Metode kualitatif dilakukan dengan cara melakukan wawancara dengan enam pertanyaan. Hasil dari dua metode ini akan dianalisis dan dibuat kesimpulan untuk mengetahui pengaruh dari penggabungan metode Design Thinking dengan *Framework Scrum*.

4.1 Hasil Pengumpulan Data

Pengumpulan data menggunakan dua metode, yaitu metode kuantitatif dan metode kualitatif. Pada metode kuantitatif dilakukan dengan melakukan pengisian kuisisioner. Lalu dilanjutkan dengan metode kualitatif yang dilakukan dengan wawancara. Terdapat empat responden yang menjadi narasumber dalam melakukan metode kuantitatif dan kualitatif dalam penelitian ini. Empat responden tersebut merupakan tim Kerjayuk yang membangun perangkat lunak Kerjayuk. Pengumpulan data hanya dilakukan kepada empat orang tersebut karena empat orang ini selaku tim Kerjayuk yang membangun perangkat lunak Kerjayuk serta telah menggunakan penggabungan metode Design Thinking dengan *Framework Scrum* sehingga cukup paham dalam menjawab pertanyaan pada tahap evaluasi ini.

Tabel II. Nama dan Peran Anggota Tim Kerjayuk

Nama	Peran dalam Tim Kerjayuk
Arya Wahyu Ramadhan	Front End Designer
Muhammad Fachrian	UI Designer dan Quality Assurance Tester
Naqliya Arum Permata	Requirement Analyst dan Front End Designer
Nurul Zakiyah	Requirement Analyst dan UI Designer

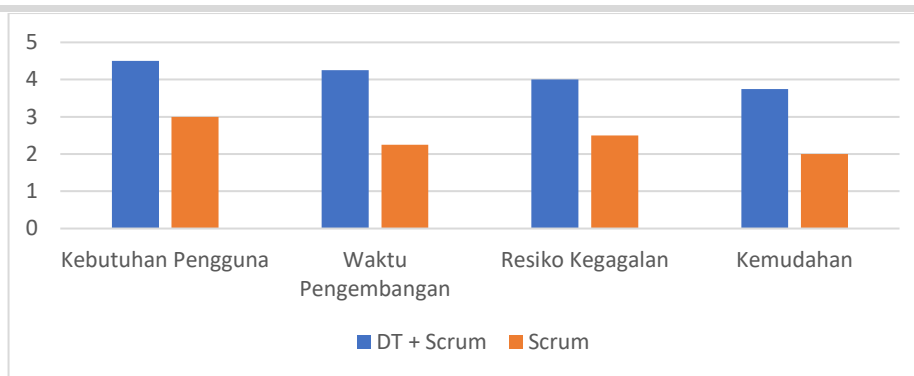
Walaupun peran dalam *Framework Scrum* tidak didefinisikan atau hanya dikenal sebagai *Development Team*, tetapi terdapat pembagian kerja yang berbeda beda satu sama lain, maka dari itu peran anggota tim ditulis secara spesifik.

4.1.1 Data Kuantitatif

Terdapat empat faktor yang akan diuji pada metode kuantitatif untuk mengetahui bagaimana pengaruh penggabungan metode *Design Thinking* dengan *Framework Scrum*. Faktor yang diujikan antara lain bagaimana pengaruh penggabungan metode Design Thinking dengan *Framework Scrum* terhadap pencarian kebutuhan pengguna, waktu yang dibutuhkan saat proses pengembangan Kerjayuk, meminimalisir resiko kegagalan Kerjayuk dan kemudahan penggunaan metode. Kebutuhan pengguna menjadi salah satu faktor yang diukur karena proses *Design Thinking* berasal serta ditujukan untuk pengguna. Waktu pengembangan juga menjadi faktor yang diukur untuk mengetahui apakah dengan dilakukannya penggabungan antara metode *Design Thinking* dengan *Framework Scrum* dapat meminimalisir waktu pengembangan Kerjayuk. Resiko kegagalan diukur karena keuntungan dari *Framework Scrum* adalah selalu cepat beradaptasi terhadap perubahan. Dan salah satu faktor yang diukur adalah kemudahan untuk mengetahui apakah penggabungan kedua metode mudah dilakukan saat dilakukan pengembangan.

Tabel III. Data Kuantitatif Penelitian *Design Thinkig Scrum*

Question	Question Description	Averages	
		DT + Scrum	Scrum
Q1	Kebutuhan Pengguna	4.5	3
Q2	Waktu Pengembangan	4.25	2.25
Q3	Resiko Kegagalan	4	2.5
Q4	Kemudahan	3.75	2
	General Averages	4.13	2.44

Gambar II. Data Kuantitatif Penelitian *Design Thinking Scrum*

Pada pengumpulan data metode kuantitatif, menggunakan responden atau partisipan yang sama untuk melakukan perbandingan antara DT + Scrum dengan Scrum, maka kemungkinan akan terjadinya bias. Bias hasil penelitian karena perbandingan DT + Scrum dengan Scrum yang dilakukan oleh partisipan yang sama diatasi dengan cara memberikan penjelasan yang mendetail kepada partisipan mengenai penggabungan antara DT + Scrum, lalu menggunakan kata-kata yang jelas dan tidak membingungkan agar responden mengerti dan menjawab dengan benar.

4.1.2 Data Kualitatif

Terdapat enam pertanyaan tentang bagaimana pengaruh penggabungan metode *Design Thinking* dengan *Framework Scrum* terhadap pencarian kebutuhan pengguna, waktu pengembangan, meminimalisir resiko kegagalan perangkat lunak Kerjayuk dan kemudahan penggunaan penggabungan metode *Design Thinking* dengan *Framework Scrum*. Tujuan dilakukannya metode ini adalah untuk mengetahui alasan dari penggabungan metode kuantitatif. Berikut beberapa pertanyaan yang ditanyakan pada metode kualitatif ini antara lain:

- Q1: Tahapan pada Scrum hanya mengenai pengembangan saja, apakah perlu ditambahkan metode lain lagi untuk mendefinisikan kebutuhan? Mengapa?
- Q2: Apakah menggunakan *Framework Scrum* saja sudah cukup dalam mengetahui kebutuhan pengguna? Mengapa?
- Q3: Apakah dengan melakukan penggabungan antara metode *Design Thinking* dan *Framework Scrum* dapat meminimalisir waktu pengembangan? Mengapa?
- Q4: Apakah dengan menggabungkan metode *Design Thinking* dengan *Framework Scrum* dapat meminimalisir resiko kegagalan perangkat lunak? Mengapa?
- Q5: Apakah penggabungan dari metode *Design Thinking* dengan *Framework Scrum* mudah untuk dilakukan? Mengapa?
- Q6: Apakah tahapan *Framework Scrum* yang dilakukan mudah untuk dilakukan? Mengapa?

4.2 Hasil Analisis

Hasil dari pengukuran menggunakan metode kuantitatif dan metode kualitatif kemudian dianalisis menggunakan Teknik Analisis Data Kuantitatif dan Teknik Analisis Data Kualitatif untuk mengetahui bagaimana pengaruh penggabungan metode *Design Thinking* dengan *Framework Scrum*.

Tabel IV. Hasil Analisis Data Kuantitatif dan Data Kualitatif Penelitian

No.	Question Description	DT + Scrum	Hasil Analisis	Scrum	Hasil Analisis
1.	Kebutuhan Pengguna	Dapat memenuhi kebutuhan pengguna	Menurut hasil yang didapat, disimpulkan bahwa dengan menggunakan penggabungan metode <i>Design Thinking</i> dengan <i>Framework Scrum</i> dapat memenuhi kebutuhan pengguna dengan nilai 4.5 karena dengan melakukan penggabungan maka kebutuhan pengguna karena pada tahapan <i>Design Thinking</i> memiliki keuntungan yaitu kebutuhan pengguna diperjelas	Kurang dapat memenuhi kebutuhan pengguna	Dengan menggunakan Scrum mendapat nilai kuantitatif sebesar 3. Walaupun dalam Scrum terdapat tahapan mengidentifikasi pengguna, tetapi itu saja dirasa belum cukup untuk mengetahui kebutuhan pengguna secara mendetail.

			untuk mengetahui kebutuhan pengguna secara mendetail yang lalu dilanjutkan dengan pengembangan perangkat lunak.		
2.	Waktu Pengembangan	Meminimalisir waktu pengembangan	Menurut hasil yang didapat, disimpulkan bahwa dengan menggunakan penggabungan metode <i>Design Thinking</i> dengan <i>Framework Scrum</i> dapat memenuhi kebutuhan pengguna dengan nilai 4.25 karena dapat meminimalisir waktu pengembangan karena saat proses pengembangan kebutuhan pengguna telah terdefinisi atau telah diidentifikasi dengan jelas maka saat melakukan pengembangan tidak perlu melakukan identifikasi kebutuhan pengguna karena telah ditetapkan pada metode <i>Design Thinking</i> .	Tidak meminimalisir waktu pengembangan	Dengan menggunakan Scrum mendapat nilai kuantitatif sebesar 2.25 dinilai tidak meminimalisir waktu pengembangan karena dalam melakukan <i>Sprint</i> mungkin terdapat hal hal yang sifatnya tidak direncanakan yang ditambahkan dalam <i>Sprint</i> .
3.	Resiko Kegagalan	Meminimalisir resiko kegagalan	Menurut hasil yang didapat, disimpulkan bahwa dengan menggunakan penggabungan metode <i>Design Thinking</i> dengan <i>Framework Scrum</i> dapat meminimalisir resiko kegagalan dengan nilai 4 karena keuntungan <i>Framework Scrum</i> adalah selalu melakukan pengujian produk. Maka dari itu dengan diadakannya pengujian produk dapat menilai apakah produk cocok untuk dirilis lalu digabungkan dengan <i>Design Thinking</i> agar mengetahui bagaimana reaksi	Meminimalisir resiko kegagalan	Dengan menggunakan Scrum mendapat nilai kuantitatif sebesar 2.5 dinilai meminimalisir resiko kegagalan tetapi tidak besar karena pengujian produk yang dilakukan setiap iterasi dilakukan oleh <i>Product Owner</i> saja, tidak dengan pengguna juga.

			pengguna terhadap produk yang dibuat.		
4.	Kemudahan	Mudah digunakan	Menurut hasil yang didapat, disimpulkan bahwa dengan menggunakan penggabungan metode <i>Design Thinking</i> dengan <i>Framework Scrum</i> mudah digunakan dengan nilai 3.75 karena metode <i>Design Thinking</i> dan <i>Framework Scrum</i> mempunyai alur masing masing yang jelas serta terstruktur maka dari itu walaupun baru menggunakan dinilai mudah digunakan.	Mudah digunakan	Dengan menggunakan <i>Scrum</i> mendapat nilai kuantitatif sebesar 2 dinilai mudah digunakan karena terdapat struktur, alur serta posisi masing masing yang rinci dan jelas.

Pengaruh penggabungan yang mempunyai nilai terbesar dihasilkan dari penggabungan metode *Design Thinking* dengan *Framework Scrum* adalah memenuhi kebutuhan pengguna dengan nilai 4.5. Hal tersebut dikarenakan adanya tahapan *Design Thinking* yang merupakan metode dari pengguna dan untuk pengguna. Pada tahapan tahapan *Design Thinking* ini juga berorientasi kepada pengguna seperti *Understand*, *Observe*, *Synthesize* dan *Ideate*. Meskipun pada *Framework Scrum* juga terdapat mendefinisikan kebutuhan, tetapi pada *Framework Scrum* kebutuhan tidak didefinisikan secara saksama, melainkan hanya sebatas mengetahui saja tidak didefinisikan secara mendetail. Dengan ditambahkannya tahapan tahapan yang ada pada metode *Design Thinking*, kebutuhan pengguna menjadi lebih terdefinisi serta dapat memahami kebutuhan pengguna dengan baik. Selain pengaruh terbesar, terdapat juga pengaruh terkecil yang dihasilkan dalam penggabungan metode *Design Thinking* dengan *Framework Scrum* yaitu kemudahan penggabungan metode dengan nilai 3.75. Penggunaan penggabungan metode ini mendapat nilai terkecil karena teori penggabungan dua metode belum banyak dipakai dan merupakan metode yang mempunyai aturan serta tahapannya masing masing. Tetapi walaupun mempunyai aturan serta tahapannya masing masing, kedua metode ini mempunyai kemiripan yaitu sama sama berorientasi kepada pengguna, serta sama sama melakukan uji coba kepada pengguna dalam prosesnya. Selain itu juga kedua metode ini juga dapat melengkapi satu sama lain karena metode *Design Thinking* menghasilkan produk yang berbentuk *Prototype* lalu *Framework Scrum* menghasilkan produk yang siap diluncurkan. Maka dari itu metode *Design Thinking* dengan *Framework Scrum* dapat melengkapi satu sama lain dalam pengaplikasiannya.

5. Kesimpulan

Penggunaan penggabungan metode *Design Thinking* dan *Framework Scrum* lebih baik digunakan dibandingkan dengan menggunakan *Framework Scrum*. Dengan menggunakan penggabungan metode *Design Thinking* dan *Framework Scrum* mendapatkan manfaat yang lebih beragam daripada menggunakannya secara terpisah. Saat melakukan metode *Design Thinking* dapat menggali serta memahami pola pikir, tingkah laku serta kebiasaan pengguna yang sifatnya sebuah kebiasaan. Lalu dengan menggunakan metode *Design Thinking* juga menghasilkan beberapa *Requirement*, *Persona*, *User Story* dan *Prototype* yang dapat menjadi pembantu dalam melakukan tahap tahap selanjutnya. Hasil akhir dari metode *Design Thinking* adalah sebuah *Prototype* yang belum merupakan produk yang siap diluncurkan. Lalu digabungkan dengan *Framework Scrum* yang mempunyai hasil akhir berupa *Sprint* yang berisikan mengenai proses proses pengembangan perangkat lunak yang bertujuan untuk menghasilkan produk yang siap diluncurkan. Maka dari itu digabungkannya metode *Design Thinking* yang dapat membantu dalam proses pengembangan, *Framework Scrum* ini juga hanya fokus kepada pengembangan saja sehingga tidak dilakukan observasi serta pemahaman terhadap pengguna. Dengan melakukan penggabungan metode *Design Thinking* dan *Framework Scrum* dapat menghasilkan sebuah produk yang berorientasi kepada pengguna, tidak adanya fase dimana melakukan hal yang tidak terstruktur, pengerjaan sudah terstruktur karena pada awal fase telah ditetapkan pembagian serta porsi kerja masing masing dan tidak membuang buang waktu serta dapat menghasilkan produk yang berkualitas dengan mengutamakan kebutuhan pengguna.

REFERENSI

- [1] F. Häger, T. Kowark, J. Krüger, C. Vetterli, F. Übernickel, and M. Uflacker, "DT@Scrum: Integrating Design Thinking with Software Development Processes," pp. 263–289, 2015, doi: 10.1007/978-3-319-06823-7_14.

- [2] R. dos S. Braz, J. R. Merlin, D. Freitas Guilhermino Trindade, C. Eduardo Ribeiro, E. M. Sgarbi, and F. de S. Junior, "Design Thinking and Scrum in Software Requirements Elicitation: A Case Study," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11583 LNCS, pp. 179–194, 2019, doi: 10.1007/978-3-030-23570-3_14.
- [3] F. - and A. Yulianto, "Kolaborasi Scrum dan Design Sprint Dalam Pengembangan Aplikasi Laboratorium Medis," *REMIK (Riset dan E-Jurnal Manaj. Inform. Komputer)*, vol. 4, no. 2, p. 47, 2020, doi: 10.33395/remik.v4i2.10558.
- [4] M. Ota, "Scrum in Research," pp. 109–116, 2010.
- [5] C. Vetterli et al., "Jumpstarting Scrum with Design Thinking," *Des. Think. Res. Build. Innov.*, vol. 2, no. 2, pp. 1–60, 2013, [Online]. Available: <https://www.alexandria.unisg.ch/publications/224403>.
- [6] M. M. Higuchi and D. N. Nakano, "Agile Design: A Combined Model Based on Design Thinking and Agile Methodologies for Digital Games Projects," *Rev. Gestão e Proj.*, vol. 08, no. 02, pp. 109–126, 2017, doi: 10.5585/gep.v8i2.528.
- [7] R. Razzouk and V. Shute, "What Is Design Thinking and Why Is It Important?," *Rev. Educ. Res.*, vol. 82, no. 3, pp. 330–348, 2012, doi: 10.3102/0034654312457429.
- [8] V. Temitayo, "Adopting Scrum as an Agile Approach in Distributed Software Development : A Review of Literature," 2017.
- [9] L. Augustin and M. Schabacker, "Combining scrum and design thinking for a highly iterative and user-centric hardware development project," *Proc. Int. Conf. Eng. Des. ICED*, vol. 2019-August, no. August, pp. 2189–2198, 2019, doi: 10.1017/dsi.2019.225.
- [10] M. Mahalakshmi and M. Sundararajan, "Traditional SDLC Vs Scrum Methodology – A Comparative Study," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 6, pp. 2–6, 2013

Lampiran

