

# Mesin Parsing *User Requirement* dari Proyek berbasis *Software Menggunakan Design Thinking*

Michael Vieri Alfa Loppies<sup>1</sup>  
<sup>1</sup>Fakultas Informatika,  
 Universitas Telkom, Bandung<sup>2</sup>  
 michaelvieri@students.telkom

Dana Sulisty Kusumo<sup>1</sup>  
<sup>2</sup>Fakultas Informatika,  
 Universitas Telkom, Bandung<sup>2</sup>  
 danakusumo@telkomuniversity.ac.id,

**Abstrak**—Proses pengembangan sebuah *software* diawali dengan *Requirement Engineering* atau RE yang merupakan proses pengumpulan *requirements* atau syarat-syarat dalam pembangunan sebuah *software*. Akan tetapi dalam penerapannya juga proses RE mengalami kekurangan, contohnya adalah dalam proses pengumpulan *requirement* dari pengguna, yang mana didapati bahwa proses RE masih mengalami permasalahan dalam memenuhi keinginan pengguna, dan *requirement* yang didapatkan terkadang masih tidak sesuai. Sehingga dilakukan upaya untuk menanggulangi permasalahan tersebut, salah satunya adalah dengan mengimplementasikan penggunaan proses *design thinking*. Proses *design thinking* atau DT merupakan sebuah proses menyelesaikan masalah secara kreatif dan inovatif. DT juga merupakan proses yang berfokus pada pengguna. Namun dalam penggunaan DT untuk membantu proses RE didapatkan juga kekurangan-kekurangan, beberapa kekurangan tersebut adalah masih belum ada pemetaan dari bagian DT mana saja yang dapat digunakan dalam membantu menyelesaikan RE dan juga kurangnya dokumentasi dari proses DT tersebut. Sehingga pada penelitian ini membangun mesin parsing *user requirement* untuk menanggulangi kekurangan dokumentasi dalam penggunaan DT dalam membantu dalam proses RE dan juga memetakan artefak DT yang dapat digunakan untuk membantu proses RE tersebut. Melalui penelitian ditemukan bahwa artefak DT dapat digunakan dalam pembentukan *user requirement*, artefak tersebut yaitu; *Point Of View (POV)*, *How Might We (HMW)*, *Empathy Map*, *User Persona*, dan *Site Map*. Kemudian, dokumen *user requirement* yang dihasilkan dapat menggunakan *Affinity Map* dalam pembangunannya agar mendapatkan hasil yang lebih baik sesuai dengan hasil validasi.

**Kata Kunci**—*requirement engineering, design thinking, user requirement*

**Abstract**—*The software development process begins with Requirements Engineering or RE, which is the process of collecting requirements or requirements for developing software. However, in its implementation the RE process also experiences shortcomings, for example in the process of collecting requirements from users, where it is found that the RE process still experiences problems in fulfilling user desires, and the requirements obtained are sometimes still not appropriate. So efforts are made to overcome these problems, one of which is by implementing the use of the design thinking process. The design thinking or DT process is a process of solving problems creatively and innovatively. DT is also a user-*

*focused process. However, in using DT to assist the RE process, there are also shortcomings, some of these shortcomings are that there is still no mapping of which parts of the DT can be used to help complete the RE and there is also a lack of documentation of the DT process. So this research will build a user requirements parsing machine to overcome the lack of documentation in using DT to assist in the RE process and also map DT artifacts that can be used to assist the RE process. Through research it was found that DT artifacts can be used in forming user requirements, these artifacts are; Point Of View (POV), How Might We (HMW), Empathy Map, User Persona, and Site Map. Then, the resulting user requirements document can use Affinity Map in its development to get better results in accordance with the validation results.*

**Keywords**—*requirement engineering, design thinking, user requirement*

## I. PENDAHULUAN

Proses awal dari sebuah perancangan *software* adalah *Requirement engineering* atau RE [1][2]. Dalam penerapannya, proses RE memiliki tujuan untuk menentukan *requirement* (persyaratan) untuk membentuk sebuah *software* [3]. Temuan *requirement* tersebut kemudian akan digunakan sebagai basis dalam membangun sebuah *software* [3]. Proses RE memiliki peran penting dalam suatu perancangan *software* [4]. Hal ini dikarenakan. berhasil tidaknya sebuah perancangan *software* dipengaruhi dengan seberapa baik proses RE dapat dilakukan dan dipenuhi pada proses *software engineering* [1][2][5][6].

Namun, proses RE juga memiliki batasan dalam penerapannya [4][7]. Salah satu kekurangan tersebut adalah dalam tahapan untuk mengumpulkan *requirement* dari pihak pengguna [2][3]. Permasalahan tersebut berkaitan dengan *requirement* yang terkumpulkan tidak sesuai atau memenuhi masukan dari pengguna [5] dan menimbulkan permasalahan dalam pengembangan *software* [2]. Hal tersebut dijelaskan pada penelitian yang sama [2] bahwa 33% dari kesalahan dalam sebuah pengembangan *software* bermula pada proses RE yang tidak memadai untuk menanggulangi keterbatasan dari proses RE, terdapat beberapa penelitian [2][5][6] yang

membahas dan meneliti penggabungan metode *design thinking* dalam melakukan proses RE.

*Design thinking* atau disingkat DT sendiri merupakan sebuah proses kreatif dan inovatif untuk memecahkan sebuah permasalahan [6][8]. *Design thinking* menggunakan pendekatan *human-centered approach* yang berarti *design thinking* lebih terpusat kepada pengguna [2][5]. Hal ini menjadikan penggunaan proses *design thinking* dalam pengembangan sebuah *software* semakin pesat, contohnya pada perusahaan; IBM, Deutsche Bank, SAP [7], dan juga SISDOT dan SISBOL di Brazil [9].

Dalam penelitian [4][7] ditemukan bahwa penggunaan proses DT dalam integrasi dengan proses RE dapat membantu dalam mengumpulkan *requirement* dari pengguna. Proses DT juga dapat membantu dalam mendefinisikan permasalahan yang dialami oleh pengguna secara lebih detail dengan menggunakan *tools* seperti; *User persona*, *user journeys*[7]. Proses DT juga memungkinkan inovasi dalam pengembangan *software* yang tetap terhubung dengan pengguna [5].

Akan tetapi, menurut penelitian ditemukan juga bahwa proses DT dalam membantu proses RE memiliki beberapa permasalahan, beberapa permasalahan tersebut ialah masih belum ada pemetaan dari bagian DT mana saja yang dapat digunakan dalam membantu menyelesaikan RE [4] dan kurangnya dokumentasi yang menyebabkan kesulitan untuk *tracing back* [7]. Maka dari itu, dalam penelitian ini, akan dilakukan pembangunan mesin parsing yang akan digunakan untuk membangun dokumen *user requirement* dari artefak hasil dari proses *design thinking*. Hal ini diharapkan untuk dapat menambahkan dokumentasi yang lebih baik dari hasil proses *design thinking* yang nantinya digunakan dalam membangun *user requirement*.

Penelitian ini melakukan analisis artefak *design thinking* apa saja yang dapat digunakan untuk menghasilkan *user requirement*. Kemudian dilakukan perancangan mesin parsing *user requirement* yang dibentuk menggunakan artefak dari proses *design thinking*.

Penelitian ini memiliki batasan pengambilan data yaitu mahasiswa S1 Informatika Universitas Telkom Bandung sebagai pengguna dalam melakukan proses DT. Setelah itu, Penelitian kali ini menggunakan satu studi kasus DT yaitu proses DT yang dilakukan. Kemudian dalam perancangan mesin parsing didapatkan masukan berupa format artefak *design thinking*,

Penelitian ini bertujuan untuk mencari artefak proses DT yang dapat digunakan sebagai bahan untuk merancang sebuah *user requirement*. Kemudian juga untuk merancang mesin parsing *user requirement* dengan masukan dari artefak DT. Kemudian yang terakhir yaitu untuk memvalidasi dokumen *user requirement* yang telah terbentuk dari artefak DT.

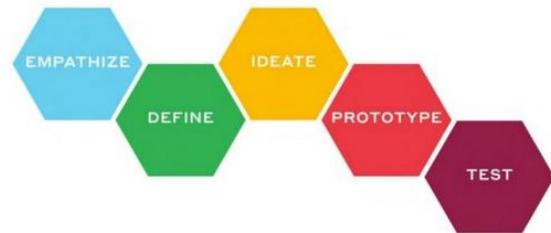
## II. STUDI TERKAIT

### A. *Design thinking*

*Design thinking* merupakan sebuah metode yang digunakan untuk memecahkan sebuah permasalahan untuk

mendapatkan solusi yang inovatif dan kreatif [2][6][8]. Untuk mendapatkan solusi tersebut, *design thinking* mengintegrasikan sisi pengguna atau manusia yang menjadi perspektif utama dalam pengembangan pencarian solusi (*user-centered approach*) [2][6].

*Design thinking* merupakan metode *iteratif*, hal ini berarti jika sekiranya ada proses yang perlu dilakukan kembali oleh karena suatu hal dan lainnya, maka akan dilakukan kembali proses sebelumnya [8][10]. Gambar 1 Menunjukkan proses dari *design thinking* [10]



GAMBAR 1  
METODE DESIGN THINKING

#### 1. *Empathize*

Proses *Empathize* merupakan proses utama dalam *design thinking*, proses ini memiliki tujuan untuk mendapatkan *insight* mengenai keinginan dan juga perspektif dari seorang pengguna sehingga kita bisa menemukan permasalahan yang dihadapi oleh pengguna. Untuk dapat berempati dengan pengguna dapat dilakukan dengan cara observasi dan juga interaksi secara langsung (interview). Proses *empathize* dapat dibantu dengan menghasilkan artefak seperti *Empathy Map*

#### 2. *Define*

Setelah melakukan proses *Empathize* maka akan dilakukan proses *Define* yang bertujuan untuk menentukan problem statement dari pemahaman yang telah didapatkan setelah menganalisis hasil dari proses *Empathize* pengguna, setelah menemukan pola dan juga keperluan yang perlu dipenuhi maka kita bisa membuat sebuah *Point-of-View* dari seorang pengguna.

#### 3. *Ideate*

Proses *Ideate* dilakukan setelah terbentuknya sebuah *Point of view*, dalam proses ini penciptaan ide-ide mengenai solusi permasalahan yang dihadapi oleh pengguna. Dalam penerapannya dilakukan dengan cara seperti: *sketching*, *mind mapping*, dan *brainstorming*. Semakin banyak ide yang muncul, maka akan semakin baik, dan oleh karena itu proses ini merupakan proses "kreatif" dalam metode *Design thinking*.

#### 4. *Prototype*

Setelah mendapatkan ide mengenai solusi yang akan diterapkan menjadi sebuah prototipe yang bersifat ringan, misalnya dengan rancangan berbentuk *post-it note*, *storyboard*, atau rancangan lainnya yang dapat digunakan oleh pengguna dengan tujuan untuk merealisasikan solusi dari permasalahan pengguna dengan biaya yang lebih terjangkau dan juga agar adanya komunikasi antara pengguna dan *design thinker*.

### 5. Test

Kemudian setelah prototipe telah terbuat, maka akan dilakukan testing atau pengujian sebuah *Prototype* kepada pengguna, hal ini dilakukan agar *design thinker* dapat mengevaluasi hasil *Prototype* yang telah dibuat dengan feedback dari pengguna sehingga *Prototype* yang telah dibuat dapat diperbaharui.

### B. Requirement Engineering

*Requirement engineering* merupakan salah satu proses dalam *software engineering* yang berfokus dalam pengembangan *software* yang memenuhi kebutuhan dari pengguna [11][12]. Dalam *Requirement engineering* terdapat lima tahapan[6][12] yang memiliki sifat iteratif, [12]:

#### 1. Requirement Elicitation

Merupakan proses mendapatkan *Requirement* sistem dari pengguna, pada intinya, tahapan ini merupakan tahap yang menciptakan relasi antara pengembang dan juga konsumen,

#### 2. Requirements Analisis

Merupakan proses penemuan dan analisis *Requirement* dari sebuah *software* yang akan dirancang dengan tujuan untuk mengidentifikasi stakeholder proyek, Batasan dan cakupan proyek dan juga mengklasifikasi dan mendokumentasi *Requirement*

#### 3. Requirements Modeling

Merupakan proses representasi dari *Requirement* yang didapatkan dari tahap *Requirement elicitation* dalam sebuah model salah satu contoh dari model tersebut adalah UML

#### 4. Requirements Specification

Merupakan tahapan yang mana SRS terbentuk, tahapan ini merupakan tahapan sentral dalam aktivitas *software development*, proses ini berisi semua *input* dan *output* dari sebuah sistem

#### 5. Requirements Validation

Merupakan tahapan terakhir dalam proses RE, pada tahapan ini semua *Requirement* yang telah dibuat akan divalidasi kembali agar tidak terjadi kerancuan, error, dan juga inkonsistensi, jika ada maka akan dilakukan perbaikan

### C. Relasi Antara Design Thinking (DT) dengan Requirement Engineering

DT merupakan sebuah proses kreatif dan inovatif dalam menyelesaikan sebuah permasalahan [5]. Proses DT sendiri telah dilakukan dengan upaya untuk menyelesaikan permasalahan rumit dan memberikan solusi yang inovatif di beberapa perusahaan [7] dan juga di badan pertahanan [9]. Dalam penelitian yang sama [7][9] juga dinyatakan bahwa, penggunaan dari DT dapat membantu dalam proses *elicitation requirement* dari *use*. Hal ini terhubung dengan *requirement engineering*, yang merupakan proses awal dalam sebuah pembentukan *software* yang berfungsi untuk mengumpulkan semua syarat dalam pembangunan sebuah

*software* [3].

### D. Software Requirement Specification (SRS)

*Software Requirement Specification* atau SRS merupakan suatu dokumen yang berisi ketentuan yang harus dipenuhi pada sebuah *software* yang nantinya akan dibangun[12]. SRS memiliki beberapa kualitas yang perlu untuk dipenuhi, kualitas tersebut adalah[11][13]: *clearness or unambiguously, completeness, correctness, undertannability, verifiability or valadidy, consistency and feasibility* [11].SRS dapat dibangun secara berbeda-beda sesuai dengan pembuatnya[13], namun secara umum, dokumen SRS dianjurkan untuk memenuhi kerangka dokumen PADA TABAEL 1[14].

TABEL 1 BAGIAN DALAM SRS MENURUT IEEE 830

Bagian SRS	Bagian
Introduction	Purpose
	Scoope
	Definitions, Acr, Abb
	Reference
	Overview
Overall Description	Product Perspective
	Product Function
	User Characteristic
	Constraint
Spesific Requirements	External Requirements
	Functional Requirements
	Performance Requirements
	Logical Database Requirements
	Design Constraint
	Software System Identities

## III. METODOLOGI PENELITIAN

Dalam perancangan mesin parsing *user requirement*, pertama-tama dilakukan proses *design thinking*. Kemudian setelah proses *design thinking* dilaksanakan dan menghasilkan artefak. Kemudian dilakukan pemetaan dari kedua tahapan dari *Design thinking* dan SRS (yang tidak sempurna, namun memenuhi syarat untuk *user requirement*), untuk mencari bagian mana yang cocok bagi artefak *design thinking* untuk bisa menjadi bahan input dalam membangun *user requirement*. Kemudian dilakukan perancangan mesin parsing *user requirement*. Dokumen yang dihasilkan oleh mesin parsing kemudian divalidasi oleh narasumber. Jika masih terdapat kekurangan atau masukan dari narasumber, maka dilakukan perbaikan sistem.

Dalam penelitian ini, menggunakan *template* dari SRS,

namun pada kenyataannya pemetaan dari SRS sendiri tidak dapat dilakukan secara maksimal, mengingat SRS yang lengkap memerlukan *system requirement* sedangkan proses dari DT hanya berfokus pada penghasilan solusi untuk memenuhi problem utama yang dialami oleh pengguna, sehingga hasil yang nantinya dibangun memenuhi syarat *user requirement*.

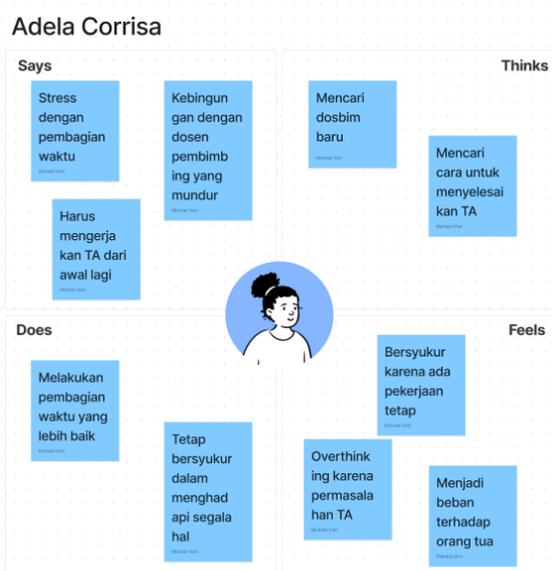
A. Pelaksanaan *design thinking*

Penerapan *design thinking* pada penelitian yang dilakukan adalah mengenai pembangunan aplikasi *mental health* mahasiswa, Pelaksanaan dari *design thinking* dilakukan dengan tahapan *empathize*, *define*, dan *ideate*. Hal ini dikarenakan fokus dalam penelitian yang bertujuan untuk memenuhi pengumpulan *user requirement*.

Pelaksanaan dari proses *design thinking* menghasilkan beberapa artefak yaitu *Empathy Map*, *Point-of-View (POV)*, *How Might We (HMW)*, *Site Map*, dan juga *User Flow*. Gambar 1 menunjukkan hasil dari *Design Thinking*

1. *Empathize*

*Empathize* merupakan tahapan yang dilakukan untuk mencari permasalahan yang dialami oleh pengguna. Maka dari itu dilakukan metode *interview* dan juga pengambilan data melalui kuesioner terhadap mahasiswa S1 Informatika Universitas Telkom. Yang kemudian dibentuk *empathy map* untuk lebih dapat memperlihatkan masalah yang dialami.



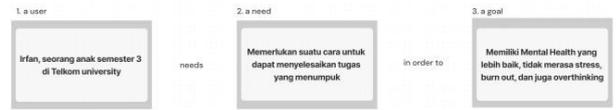
GAMBAR 1  
CONTOH EMPATHY MAP

Gambar 1 menunjukkan hasil dari fase pertama yang dilakukan dalam penelitian ini yaitu tahapan *empathize*. Setelah ditemukan permasalahan yang dialami oleh mahasiswa, maka dilakukan tahapan pendefinisian permasalahan, hal ini dilakukan agar *problem statement* dari pengguna dapat ditentukan dengan baik. Hal ini dapat didapatkan dengan *Point of View (POV)* dan *How Might We (HMW)*.

2. *Define*

Setelah ditemukan permasalahan yang dialami oleh

mahasiswa, maka dilakukan tahapan pendefinisian permasalahan. Hal ini dapat didapatkan dengan *Point of View (POV)* dan *How Might We (HMW)*



GAMBAR 2  
POINT OF VIEW (POV)

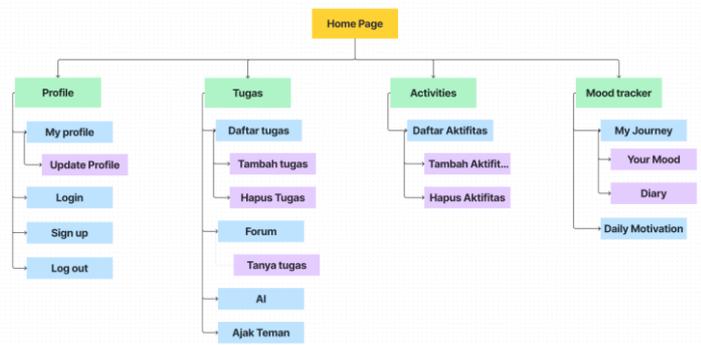


GAMBAR 3  
HOW MIGHT WE (HMW)

Gambar 2 dan 3 menunjukkan *Point Of View (POV)* dan *How Might We (HMW)* yang dilakukan agar *problem statement* dari pengguna dapat ditentukan dengan baik

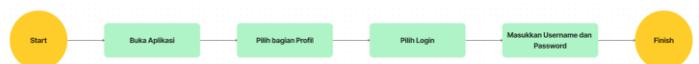
3. *Ideate*

Setelah kita mendapatkan *problem statement* dari fase sebelumnya, maka dilakukan pemuatan ide-ide terbaru.



GAMBAR 4  
SITE MAP/ SKETCHES

Gambar 4 menunjukkan sketsa *site map* yang dibentuk, agar solusi yang dilakukan dapat ter visualisasikan dengan baik.



GAMBAR 5  
CONTOH USER FLOW

Gambar 5 menunjukkan *user flow* yang dibentuk, sebagai lajur bagi pengguna yang menggunakan *software* tersebut

B. Pemetaan SRS dan *Design thinking*

Kemudian Setelah proses *design thinking*, dilakukan proses pemetaan antara bagian dalam SRS dengan artefak *design thinking*. Tujuan dari pemetaan untuk membantu perumusan *user requirement*, SRS sendiri bukan target dari pembangunan mesin parsing dalam penelitian ini. Namun

SRS memiliki bagian-bagian yang bersinggungan dengan penemuan *user requirement*. Berikut hasil pemetaan artefak *design thinking* dan bagian dalam SRS tersebut.

TABEL 2 PEMETAAN SRS DENGAN PHASE DAN ARTEFAK DESIGN THINKING

Bagian SRS	Bagian	Penjelasan	Fase Design thinking	Artefak Design thinking yang digunakan	Alasan
Introduction	Purpose	Menggambarkan tujuan SRS	Define	POV (Point of view)	POV Menggambarkan Tujuan dari pembentukan sebuah <i>software</i> dari sudut pandang Pengguna dalam memecahkan permasalahan yang sedang mereka alami
	Scope	Identifikasi produk <i>software</i> yang akan diproduksi dengan nama Menjelaskan apa yang perlu dan tidak perlu dilakukan oleh perangkat lunak Menjelaskan penerapan <i>software</i> termasuk manfaat dan tujuan	Define	How Might We	How Might We memberikan penjelasan mengenai permasalahan yang dialami oleh pengguna secara terjabar
	Definitions, Acr, Abb		--		
	Reference		--		
	Overview		--		
Overall Description	Product Perspective		--		
	Product Function	Merangkum kapabilitas fungsional utama yang ditampilkan secara singkat	Ideate	Site map	Site map menampilkan fungsionalitas utama dari <i>software</i>

				yang akan dibuat	
	User Characteristic	Mendeskripsikan karakteristik umum dari pengguna yang dituju, hal ini meliputi tingkat pendidikan, pengalaman, dan kemampuan	Empathize	Empathy map dan Persona	Empathy Map dan User Persona menggambarkan segmen pengguna yang akan menggunakan <i>software</i> yang dibuat
	Constraint		--		
Specific Requirements	External Requirements		--		
	Functional Requirements	Mengidentifikasi tindakan dasar yang harus dilakukan <i>Software</i>	Ideate	User Flow	User flow menggambarkan arah yang harus dilalui oleh pengguna dalam menjalankan <i>software</i>
	Performance Requirements		--		
	Logical Database Requirements		--		
	Design Constraint		--		
	Software System Identities		--		

Didapatkan bahwa tidak terlalu banyak bagian dari SRS yang tidak dapat diisi oleh artefak *design thinking*. Hal ini dikarenakan *design thinking* yang lebih mengutamakan kebutuhan dari pengguna [2][6] dan SRS yang juga memiliki fokus lebih ke dalam kebutuhan sistem [14].

C. Perancangan mesin parsing *user requirement*

Kemudian setelah melakukan pemetaan, dilakukan perancangan pembangkit otomatis SRS. Berikut merupakan bagan alur perancangan mesin parsing. Pertama-tama dilakukan pemformatan dari artefak *design thinking* menjadi *file text*, Kemudian setelah melakukan format dilakukan perancangan mesin parsing yang kemudian membentuk dokumen *user requirement*



GAMBAR 6 BAGAN ALUR MESIN PARSING

Pertama, dilakukan pemformatan dari Artefak *design thinking* menjadi format *text*. Tabel menunjukkan pemformatan setiap artefak *design thinking* menjadi format *text*

TABEL 3

FORMAT TEXT DARI ARTEFAK DESIGN THINKING

Artefak design thinking	Format text setiap artefak DT
POV	[Pengguna] perlu [keperluan] karena [wawasan yang didapatkan]
HMW	Bagaimana kita bisa [Kata kerja] untuk [mencapai tujuan]
Empathy map	Nama: [...] Umur: [...] Pekerjaan: [...]  User Says: [-..... (berbentuk poin)] User Thinks: [-..... (berbentuk poin)] User Does: [-..... (berbentuk poin)] User Feels: [-..... (berbentuk poin)]
Site Map	[Level 1] [Level 2] [Level 3]
User Flow	[Tindakan User ] [Tahapan] (dalam bentuk urutan 1...seterusnya)

Setelah dilakukan pemformatan, dilakukan *input* ke dalam mesin parsing.



GAMBAR 8 ALUR BAGAN MESIN PARSING

Gambar 8 menunjukkan alur bagan Mesin parsing. Mesin parsing membaca *file text* yang telah dibuat sebagai bahan *input*, kemudian dilakukan parsing data dari *file text*, yang kemudian dimasukkan ke dalam dokumen dengan format *.word*. Sehingga nantinya keluar produk yaitu dokumen *user requirement*.

D. Validasi SRS

Validasi oleh empat narasumber yang memiliki pengalaman 1-4 tahun dengan *design thinking*, dan juga dengan *Software Development Life Cycle*. Hal ini dilakukan agar didapatkannya penilaian dan juga masukan berdasarkan hasil dari SRS yang telah terbentuk, kesesuaian artefak *design thinking* dengan Bagian SRS, dan juga masukan secara menyeluruh dengan penelitian yang telah dilakukan.

Validasi yang diterapkan adalah dengan memberikan pertanyaan-pertanyaan mengenai mesin parsing yang telah dibentuk dan juga kesesuaiannya, berikut merupakan hasil dari validasi:

1) **Pertanyaan pertama:** Menurut Anda, Sudah tepatkah hasil *generate* bagian-bagian dari SRS yang dihasilkan dari artefak-artefak dan atau tahapan *design thinking*.

**Rangkuman Jawaban:** Dari respons narasumber didapatkan bahwa, bagian SRS dari hasil pemetaan sudah tepat dengan artefak *design thinking* yang dipet,

2) **Pertanyaan kedua:** Menurut Anda, Apakah artefak dari *design thinking* lain yang dapat digunakan sebagai bahan *input* untuk membangun SRS

**Rangkuman Jawaban:** terdapat dua narasumber

yang memberikan masukan untuk menggunakan *affinity map* untuk memperjelas permasalahan utama yang dialami dan juga memprioritaskan permasalahan. Kemudian juga mendapati masukan dari narasumber untuk menggunakan *user journey maps* untuk mengisi bagian *non-functional requirement*

3) **Pertanyaan ketiga:** Berikan masukan terhadap bagian-bagian dari SRS yang tidak dipetakan dari *design thinking*

**Rangkuman Jawaban:** Disarankan untuk dapat mengisi bagian *non-functional requirement* SRS

4) **Pertanyaan keempat:** Berikan masukan terhadap format artefak *design thinking* yang digunakan sebagai bahan masukan dari SRS

**Rangkuman Jawaban:** Pemformatan artefak *design thinking* sudah cukup baik dan akurat dengan bagian dari SRS yang dapat diisikan

Hasil dari Validasi didapatkan bahwa artefak dari *design thinking* dapat digunakan dalam membangun bagian dari SRS yang berkaitan dengan *user requirement*, ditemukan bahwa artefak seperti: POV, HMW, *Empathy Map*, *User Persona*, *User flow*, dan *Site Map* dapat memberikan kemudahan dalam proses mendapatkan *user requirement*.

E. Iterasi

Setelah dilakukan validasi dari narasumber, dilakukan perbaikan, salah satunya adalah dengan menambahkan artefak *Affinity Map* dalam mesin parsing dan merapikan pemformatan.

TABEL 4 FORMAT AFFINITY MAP

Artefak design thinking	Format text
<i>Affinity Map</i>	[Kategori / Permasalahan Utama] -[Daftar Kategori/ Permasalahan Utama] .....

penggunaan dari *Affinity Map* dapat memperjelas permasalahan utama dan juga mengategorikan permasalahan tersebut, sehingga dapat membantu dalam mengisi bagian dari *Purpose*



GAMBAR 9 AFFINITY MAP

IV. EVALUASI.

Evaluasi penelitian ini bertujuan untuk menampilkan hasil dari penelitian yang telah dilakukan, mulai dari tahapan awal, proses pengerjaan dan juga hasil yang telah didapatkan. Evaluasi juga dilakukan untuk menjawab rumusan masalah dari penelitian yang telah dilakukan. Melalui validasi yang dilakukan didapatkan beberapa temuan sebagai berikut Artefak *design thinking* yang dapat digunakan adalah; POV, HMW, *Empathy Map*, *User Persona*, *User Flow*, *Site Map*. Artefak ini digunakan untuk mengisi bagian SRS; *Purpose*, *Scope*, *Product Function*, *User Characteristic*, dan juga *Functional Requirement*.

Lalu, didapatkan juga temuan dari hasil validasi yang menyarankan untuk menggunakan *Affinity Map* agar solusi dari permasalahan dapat terlihat lebih jelas dan lebih terurut sesuai dengan tema permasalahan. Kemudian juga disarankan untuk menggunakan *user journey* agar dapat memenuhi kebutuhan fungsional dan juga kebutuhan non-fungsional. kemudian untuk bagian pemformatan dan pemataan dari format artefak *design thinking*. Narasumber memberikan masukan untuk lebih dirapikan dan juga dibuat lebih sederhana.

Dengan melakukan perbaikan dalam pembentukan mesin parsing, maka berikut merupakan hasil dari menjalankan mesin parsing

#### User Requirement

##### 1. Introduction

###### 1.1 Purpose

Sebagai mahasiswa Informatika, saya memerlukan cara untuk mengurangi rasa stress saya agar saya dapat menjalankan perkuliahan saya dengan baik dan memiliki kesehatan mental yang baik

Sebagai mahasiswa Informatika tingkat akhir, saya memerlukan motivasi mengerjakan TA untuk menyelesaikan studi saya di perkuliahan

Sebagai mahasiswa Informatika, saya memerlukan bantuan teman dalam mengerjakan tugas untuk membantu saya dalam memahami tugas

###### 1.2 Scope

Permasalahan Mahasiswa:

1. Permasalahan Kuliah
2. Permasalahan Harian
3. Perasaan Mahasiswa

Bagaimana kita bisa memfasilitasi mahasiswa dalam mengerjakan tugas yang kurang dipahami untuk mengurangi stress

Bagaimana kita bisa membantu mahasiswa mengatur waktu untuk menjalankan kesibukan mereka dengan baik

Bagaimana kita bisa membangun fitur untuk mengingatkan mahasiswa terhadap tugas mereka untuk mengurangi overthinking terhadap tugas yang menumpuk

Bagaimana kita bisa memotivasi mahasiswa dalam menjalani perkuliahan untuk menjadikan kesehatan mental yang baik

## V. KESIMPULAN DAN SARAN

Hasil dari penelitian ini yakni, pemetaan SRS menggunakan artefak DT tidak dapat dilakukan dengan sempurna, hal ini dikarenakan proses DT yang memiliki fokus utama pada pengguna, hal ini menyebabkan artefak dari DT tidak dapat mengisi bagian *requirement* sistem yang merupakan bagian dalam SRS. Perumusan *user requirement* dapat dibentuk dengan menggun artefak dari *design thinking* yang telah dipetakan dan diformat terlebih dahulu. Artefak *design thinking* dapat digunakan untuk membentuk *User Requirement* yaitu *Empathy Map*, *User Persona*, POV, HMW, *Site Map*, dan *User Flow*. Setelah divalidasi, penggunaan *Affinity Map* dapat dibangun dalam membentuk *user requirement*

Penelitian selanjutnya dapat melakukan proses *design thinking* secara lebih lengkap. Pemformatan artefak *design thinking* dapat diperbaiki sehingga hasil lebih rapi. Validasi dapat dilakukan oleh orang/ narasumber yang lebih banyak agar hasil yang didapatkan lebih akurat

## DAFTAR PUSTAKA

- [1] T. ur Rehman, M. N. A. Khan, and N. Riaz, "Analysis of Requirement Engineering Processes, Tools/Techniques and Methodologies," *Int. J. Inf. Technol. Comput. Sci.*, vol. 5, no. 3, pp. 40–48, 2013, doi: 10.5815/ijitcs.2013.03.05.
- [2] J. Hehn and D. Mendez, "Combining Design Thinking and Software Requirements Engineering to Create Human-Centered Software-Intensive Systems," pp. 11–60, 2022, doi: 10.1007/978-3-030-90594-1\_2.
- [3] T. Shah and S. V Patel, "A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods," *Int. J. Comput. Appl.*, vol. 99, no. 15, pp. 36–45, 2014, doi: 10.5120/17451-8370.
- [4] Kahan, E., Genero, M. and Oliveros, A. (2019a) 'Challenges in requirement engineering: Could design thinking help?', *Communications in Computer and Information Science*, pp. 79–86. doi:10.1007/978-3-030-29238-6\_6.
- [5] R. Sharma and J. N. Singh, "Design-Thinking and User's Requirement Engineering: Human Centred Development," *Int. J. Mech. Eng.*, vol. 6, pp. 108–114, 2021.
- [6] C. Vetterli, W. Brenner, F. Uebernickel, and C. Petrie, "From palaces to yurts: Why requirements engineering needs design thinking," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 91–94, 2013, doi: 10.1109/MIC.2013.32.
- [7] J. Hehn and F. Uebernickel, "The use of design thinking for requirements engineering: An ongoing case study in the field of innovative software-intensive systems," *Proc. - 2018 IEEE 26th Int. Requir. Eng. Conf. RE 2018*, pp. 400–405, 2018, doi: 10.1109/RE.2018.00-18.
- [8] J. Liu, M. Zhang, and X. Hu, "Understanding Design Thinking : a Process," *Thinking*, no. September, pp. 44–48, 2011.
- [9] A. Marcus and W. Wang, *Design, User Experience, and Usability - Theory and Practice - 7th International Conference, DUXU 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part 1*. 2018.
- [10] D. Kelley and T. Brown, "An introduction to Design Thinking," *Institute Des. Stanford*, p. 6, 2018.
- [11] F. Belfo, "People, Organizational and Technological Dimensions of Software Requirements Specification," *Procedia Technol.*, vol. 5, pp. 310–318, 2012, doi: 10.1016/j.protcy.2012.09.034.
- [12] A. O. J. ale Sabriye and W. M. N. W. Zainon, "A framework for detecting ambiguity in software requirement specification," *ICIT 2017 - 8th Int. Conf. Inf. Technol. Proc.*, no. May 2017, pp. 209–213, 2017, doi: 10.1109/ICITECH.2017.8080002.
- [13] A. J. Doe, "IEEE recommended practice for software requirements specifications," *Softw. Requir. Eng.*, pp. 207–244, 2011, doi: 10.1109/9781118156674.ch3.
- [14] ISO/IEC/IEEE, "ISO 29148 Systems and software engineering—Life cycle processes—Requirements engineering," vol. 2011, 2017.