

# Identifikasi Karakteristik Kualitas Menggunakan Pendekatan TCM Berbasis Kebutuhan Fungsional

Denni Aldi Ramadhani dan Umi Laili Yuhana

**Abstract**—Defining quality requirements is an important activity in the development of software because quality requirements describe how well the software can be implemented. In this study will propose an approach to make improvements the specification of quality characteristics software using a spectrum analysis by utilizing the TCM (Term-Characteristic Mapping) method, in this method also utilize algorithms RabinKarp used to perform mapping between the list of term with functional requirements written in the software requirement specification documents. for testing phase its used 167 term of quality and 22 subcategories quality characteristics and uses two software requirement specification documents that will appear the quality characteristics. Results from this study is a list of quality characteristics that appear along with the value of its appearance in the functional requirements are analyzed.

**Keywords**—quality requirements, TCM, RabinKarp, quality characteristics

**Abstrak**—Mendefinisikan kebutuhan kualitas merupakan suatu aktivitas penting dalam pengembangan perangkat lunak karena kebutuhan kualitas menggambarkan seberapa baik fungsional perangkat lunak dapat diimplementasikan. Dalam penelitian ini akan mengajukan sebuah pendekatan untuk melakukan perbaikan tahap spesifikasi karakteristik kualitas perangkat lunak dengan menggunakan pendekatan analisis spektrum dengan memanfaatkan metode TCM (Term-Characteristic Mapping), dalam metode tersebut juga akan memanfaatkan algoritma RabinKarp yang digunakan untuk melakukan mapping antara daftar *term* dengan kebutuhan fungsional yang tertulis dalam dokumen spesifikasi kebutuhan perangkat lunak. Untuk pengujian digunakan 167 *term kualitas* dari 22 subkategori karakteristik kualitas dan menggunakan 2 dokumen SKPL yang akan diidentifikasi karakteristik kualitasnya. Hasil dari penelitian ini adalah daftar karakteristik kualitas yang muncul beserta nilai kemunculannya dalam kebutuhan fungsional yang dianalisis.

**Kata kunci**—kebutuhan kualitas, TCM, RabinKarp, karakteristik kualitas

Denni Aldi Ramadhani adalah dosen Teknik Informatika Universitas Dian Nuswantoro Semarang, Indonesia (email : denni.aldi@gmail.com)

Umi Laili Yuhana adalah dosen Teknik Informatika Institut Teknologi Sepuluh November, Surabaya, Indonesia (email : yuhana@if.its.ac.id)

## I. PENDAHULUAN

**M**ENDEFINISIKAN kebutuhan kualitas suatu sistem perangkat lunak merupakan suatu aktivitas yang tergolong sulit karena pemangku kepentingan biasanya tidak menyatakan kebutuhan kualitas secara eksplisit. J. D. Blaine dan J. Cleland-Huang dalam [1] menjelaskan perbedaan mendasar antara kebutuhan fungsional dengan kebutuhan kualitas, bahwa kebutuhan fungsional menentukan yang akan sistem lakukan sedangkan kebutuhan kualitas menggambarkan seberapa baik fungsi – fungsi tersebut dilakukan. Sehingga mendefinisikan kebutuhan kualitas lebih sulit jika dibandingkan dengan mendefinisikan kebutuhan fungsional. Terpenuhinya kebutuhan kualitas sangat penting karena merupakan salah satu faktor untuk mengukur kualitas dari suatu perangkat lunak. Dalam siklus hidup suatu perangkat lunak, dengan diketahuinya kebutuhan kualitas maka dapat membantu dalam proses pemeliharaan dan manajemen perangkat lunak tersebut.

Karena beberapa alasan diatas banyak penelitian yang telah mengajukan metode untuk mengidentifikasi kebutuhan kualitas. Seperti pada penelitian yang dilakukan oleh Vatesh Pasrija *et all* [2] mengajukan sebuah pendekatan metode untuk memprediksi kualitas software dengan menggunakan Choquet Integral Approach, dan pendekatan fuzzy yang digunakan untuk mengukur parameter kualitas perangkat lunak. Kelemahan pada pendekatan ini adalah sub-kriteria yang muncul tidak dapat ikut dipertimbangkan dalam melakukan prediksi padahal sub-kriteria juga memiliki pengaruh yang signifikan. Pada Penelitian oleh Rosaria Bittencourt *et all* [3] berusaha memunculkan atau mengidentifikasi kebutuhan kualitas suatu sistem informasi berdasarkan pada suatu model bisnis. Kelemahan pada pendekatan ini adalah tidak mempertimbangkan batasan, prioritas, operasionalisasi, dan interdependensi.

Reinhold Plösch *et all* [4] telah melakukan penelitian untuk mendefinisikan kebutuhan kualitas dengan pendekatan berorientasi tujuan (*Goals*) dengan model kualitas (*Quality Models*) dan dengan pendekatan hambatan (*Obstacles*). Ada 2 kelemahan pada metode ini yaitu yang pertama adalah pada pendekatan hambatan (*Obstacles*) dimana penggunaan waktu yang lama karena perumusan ide untuk menentukan

kebutuhan kualitas berdasarkan pemikiran langsung dari peneliti yang kemudian harus dibandingkan dengan para pemangku kepentingan dan para pengembang, kelemahan kedua adalah pada pendekatan dengan menggunakan *Quality Models* di tahap spesifikasi karakteristik kualitas, dimana pada tahap ini hanya bisa memunculkan satu karakteristik kualitas yang akan diproses pada tahap selanjutnya sehingga tidak semua karakteristik kualitas yang berhubungan dengan *Goal* dapat diproses untuk memunculkan kebutuhan kualitasnya. Kelemahan ini juga akan berdampak pada ketidaklengkapan kebutuhan kualitas yang didefinisikan karena hanya melihat dari satu karakteristik kualitas saja.

Pada paper ini berusaha memperbaiki kelemahan tahapan spesifikasi karakteristik kualitas pada *Quality Models* dalam penelitian [4] dengan metode analisis spektrum menggunakan TCM (*Term-Characteristic Map*) [5], [6]. Pada paper ini juga akan merubah tahapan awal *Quality Models* yang berbasis *Goal* menjadi berbasis kebutuhan fungsional yang berasal dari dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak). Perubahan ini bertujuan agar karakteristik kualitas yang dimunculkan sesuai dengan fungsional perangkat lunak yang dibangun [1]. Domain sistem yang akan diuji untuk dimunculkan karakteristik kualitasnya adalah sistem manajemen informasi akademik sehingga SKPL yang digunakan dalam paper ini adalah SKPL pengembangan sistem manajemen informasi akademik berbahasa Inggris dan karakteristik kualitas yang dimunculkan sesuai dengan standar ISO 9126. Domain sistem manajemen informasi akademik dipilih agar hasil uji coba dalam paper ini dapat dibandingkan dengan hasil pada penelitian [4]. Hasil uji coba yang berupa karakteristik kualitas yang muncul nantinya akan dirangkingkan sesuai dengan nilai dari pengukuran pada dokumen SKPL.

II. DASAR TEORI

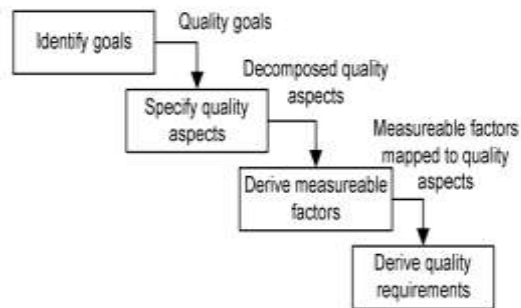
A. Model Kualitas

Model kualitas dalam penelitian yang dilakukan oleh Florian Deissenboeck *et al* [7] didefinisikan sebagai sebuah model yang memiliki tujuan untuk mendeskripsikan, menilai dan / atau memprediksi kualitas. Dalam penelitian yang dilakukan oleh Reinhold Plösch *et al* [4] memanfaatkan model kualitas untuk mendefinisikan kebutuhan kualitas dengan pendekatan berorientasi tujuan (*Goals*). Model kualitas yang digunakan dalam penelitian tersebut terbagi menjadi 4 tahapan seperti yang terlihat pada Gambar 1, yaitu terdiri dari tahap mengidentifikasi tujuan (*Goals*), tahap spesifikasi aspek kualitas, tahap mendapatkan faktor yang terukur, dan tahap mendapatkan kebutuhan kualitas.

Tahapan pertama adalah Menentukan dan mendefinisikan *goal* / tujuan bisnis dari para stakeholder yang nantinya mampu ditangani oleh perangkat lunak. Setelah mendefinisikan *goal* tersebut maka harus dikonsolidasikan dan diperiksa, mungkin

*goal* tersebut masuk dalam strategi yang ada, rencana kualitas atau roadmaps dari perangkat lunak tersebut, atau mungkin *goal* tersebut merupakan *goal* yang didefinisikan untuk menangani masalah – masalah khas mengenai kualitas produk. Pada tahapan ini terdapat kelemahan yaitu hanya satu *goal* / tujuan yang dapat diproses untuk dimunculkan aspek kualitasnya pada tahap selanjutnya.

Tahapan kedua adalah Berdasarkan *goal* yang sudah didefinisikan pada tahap sebelumnya, maka langkah selanjutnya adalah mengidentifikasi hirarki aspek kualitas secara menyeluruh sehingga dapat ditemukan aspek kualitas yang dapat berkontribusi dalam mencapai tujuan yang telah didefinisikan sebelumnya, dan membahas aspek kualitas tersebut dengan *stakeholder*. Tahap berusaha menghubungkan antara tujuan dengan aspek kualitas yang spesifik. Pada tahapan ini terdapat kelemahan yaitu aspek kualitas yang dimunculkan untuk diproses pada tahap selanjutnya hanya satu aspek kualitas tanpa mempertimbangkan aspek kualitas lainnya.



Gambar 1. Bagan Model Kualitas[4]

Tahapan ketiga adalah Model kualitas yang mendasari harus menyediakan ukuran yang dapat mempengaruhi aspek kualitas masing-masing baik secara langsung atau tidak langsung melalui faktor. Faktor-faktor atau langkah-langkah yang jelas tidak memberikan kontribusi terhadap pencapaian tujuan tidak dipertimbangkan. Selain itu, hubungan yang sesuai antara faktor / langkah-langkah dan aspek kualitas yang diperlukan. Oleh karena itu, pendekatan ini menyerukan untuk memeriksa dampak dari faktor / langkah-langkah untuk setiap aspek kualitas yang diidentifikasi. Baik dampak positif atau negatif.

Tahapan keempat adalah Sebagai sebuah model kualitas yang sudah menyediakan hubungan antara faktor dan aspek kualitas, tahap ini dilakukan untuk memperoleh kebutuhan kualitas dari faktor dengan mempertimbangkan dampak pada aspek kualitas di tahap sebelumnya. Biasanya kebutuhan kualitas dijelaskan secara tekstual dan berisi informasi seperti deskripsi, kriteria yang sesuai, dll. Deskripsi biasanya menentukan mana properti entitas sistem perangkat lunak yang harus dipenuhi, sedangkan kriteria yang sesuai, menyediakan langkah – langkah dan threshold proyek mengenai cara memverifikasi kebutuhan.

B. Analisis Spektrum dan TCM

Analisis spektrum untuk kebutuhan kualitas diperkenalkan pada penelitian yang dilakukan oleh

Haruhiko Kaiya *et all* [5] dalam penelitian tersebut diajukan suatu konsep metode yang disebut dengan “*software quality spectrum*”. Seperti yang tertulis dalam [5], *quality spectrum* menunjukkan sebuah rasio dari setiap karakteristik kualitas yang terdapat dalam artefak rekayasa perangkat lunak, seperti misalnya spesifikasi kebutuhan, sebuah manual, sebuah dokumen desain dll. Pada dasarnya ide dari analisis spektrum kualitas software ini seperti dalam teori optik, dimana kita dapat mengidentifikasi panjang gelombang dan kekuatan dari masing – masing panjang gelombang dalam suatu gelombang menggunakan analisis spektrum gelombang. Ide ini sangat berguna untuk fitur yang tersebar seperti panjang gelombang atau sebuah rasio dari spesifik material. Dari teori tersebut Haruhiko Kaiya *et all* menerapkan ide tersebut ke dalam kebutuhan kualitas, karena keinginan utamanya adalah untuk mengidentifikasi karakteristik kualitas dan tingkat kepentingan dari setiap karakteristik kualitas yang tersebar dalam artefak rekayasa perangkat lunak.

Prosedur dalam analisis spektrum kualitas yaitu pertama fokus pada artefak rekayasa perangkat lunaknya karena analisis spektrum kualitas ini menggunakan artefak tersebut, maka harus ditentukan terlebih dahulu artefak apa yang akan digunakan (dalam paper ini artefak rekayasa perangkat lunak yang digunakan adalah dokumen SKPL). Prosedur kedua adalah mempersiapkan analiser spektrum kualitas untuk artefak yang telah dipilih pada prosedur pertama, artefak tersebut nantinya menjadi sebuah input. Pada prosedur ketiga adalah menghasilkan sebuah spektrum kualitas untuk artefak yang diinputkan dengan menggunakan spektrum analiser pada prosedur 2. Prosedur terakhir adalah mendapatkan spektrum kualitas tambahan dari domain sistem.

Dalam penelitian keduanya Haruhiko Kaiya *et all* [6] melakukan perbaikan dalam metode tersebut dengan menambahkan suatu pendekatan metode yang disebut dengan TCM (*Term-Characteristics Map*) yaitu suatu pendekatan yang berfungsi sebagai domain pengetahuan untuk membentuk suatu hubungan antara artefak perangkat lunak (dokumen SKPL) dengan karakteristik kualitas. Dalam membentuk hubungan tersebut yaitu dengan melakukan mapping antara term dalam TCM sesuai ISO 9126 dengan kebutuhan fungsional yang dijelaskan dalam dokumen SKPL.

Spesifikasi dalam penerapan spektrum analisis dengan TCM yaitu yang pertama adalah gelombang input, yang dimaksud dalam gelombang input adalah artefak dari rekayasa perangkat lunak dalam hal ini adalah dokumen SKPL. Spesifikasi kedua adalah panjang gelombang, yang dimaksud panjang gelombang adalah karakteristik kualitas berdasarkan ISO 9126 yang akan dijelaskan pada bagian berikutnya. Spesifikasi ketiga adalah kekuatan, yang dimaksud kekuatan adalah nilai bobot dari setiap *term* yang berhubungan pada setiap masing – masing karakteristik kualitas. Pemberian bobot dengan range antara 0 hingga 1. Penentuan bobot tiap *term* berdasarkan tingkat kepentingan *term* tersebut terhadap karakteristik

kualitas yang berhubungan dengannya. Bobot ini yang nantinya akan digunakan untuk perhitungan perangkingan karakteristik kualitas yang didapatkan.

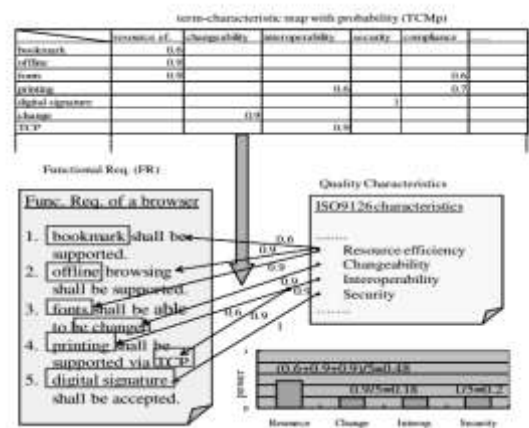
Implementasi dari metode analisis spektrum dengan TCM ini adalah tahap pertama dekomposisi dokumen SKPL kedalam potongan kalimat sederhana. Tahap kedua kenali bagian kebutuhan fungsional dalam dokumen tersebut. Tahap ketiga adalah untuk setiap *term* yang berhubungan dengan karakteristik kualitas cari hubungan yang mungkin ada, yaitu dengan cara memappingkan setiap *term* dengan kebutuhan fungsional yang didapatkan dari dokumen SKPL. Misalnya *term* “*password*” terdapat dalam kebutuhan fungsional, yang berhubungan dengan karakteristik kualitas “*security*”. Tahap keempat adalah menghitung nilai setiap karakteristik kualitas berdasarkan bobot yang telah diberikan pada setiap *term*, dengan persamaan :

$$T = \frac{\sum \omega}{n} \quad (1)$$

Keterangan :

$\omega$  = bobot tiap *term* yang muncul pada SKPL

$n$  = jumlah seluruh kebutuhan fungsional dalam SKPL



Gambar 2. Contoh dari metode TCM [6]

Pada Gambar 2 merupakan contoh sederhana penerapan metode TCM, yaitu tabel paling atas dalam gambar merupakan domain pengetahuan dari TCM, yang berisi *term – term* yang berhubungan pada setiap karakteristik kualitas, tidak lupa juga bobot pada masing – masing *term* tersebut. Misalnya *term* “*bookmark*” merupakan *term* yang berhubungan dengan karakteristik kualitas *resource efficiency* dengan bobot 0.6. Pada gambar sebelah kiri merupakan contoh sederhana kebutuhan fungsional yang didapatkan dari dokumen SKPL yang terdiri dari 5 contoh kebutuhan fungsional. Dan gambar sebelah kanan adalah daftar dari karakteristik kualitas yang ingin dicari nilai kemunculannya. Misalnya dalam gambar tersebut untuk karakteristik kualitas *resource efficiency*, setiap *term* yang berhubungan dengannya dimappingkan dengan kebutuhan fungsional, dan dari hasil mapping terlihat bahwa ketiga *term* yang berhubungan dengan *resource efficiency* muncul dalam kebutuhan fungsional tersebut. Sehingga perhitungan nilai untuk karakteristik kualitas

resource efficiency adalah  $0.6+0.9+0.9/5=0.48$ , menggunakan rumus sesuai dengan persamaan (1).

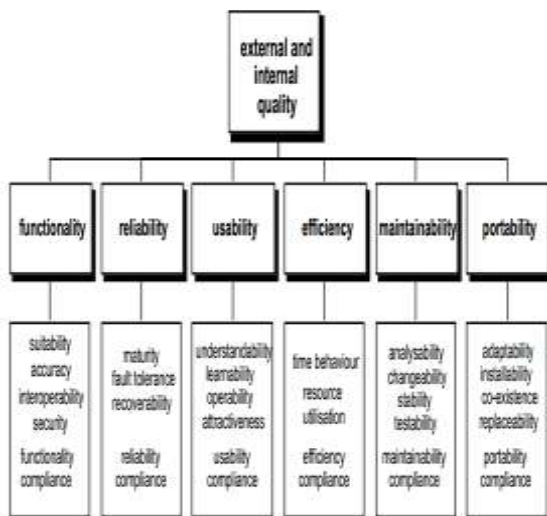
C. Algoritma Rabin Karp String Matching

Algoritma Rabin Karp diciptakan pada tahun 1987 oleh Richard M. Karp dan Michael O. Rabin. Algoritma ini memanfaatkan fungsi hashing untuk menemukan pola dalam suatu teks. Beberapa Karakteristik dari Algoritma Rabin Karp sesuai dalam penelitian [8] adalah dalam algoritma Rabin Karp menggunakan suatu fungsi hashing. Karakteristik kedua adalah fase preprocessing menggunakan kompleksitas waktu  $O(m)$ . Karakteristik ketiga dalam fase pencarian kompleksitasnya adalah  $O(mn)$  dan waktu yang diperlukan dalam algoritma ini adalah  $O(n+m)$ .

Waktu  $O(n+m)$  yang dibutuhkan dalam algoritma ini jauh lebih efisien jika dibandingkan dengan algoritma lain seperti misalnya Brute Force yaitu  $O(nm)$ . Algoritma ini menjadi pilihan bila digunakan untuk mencari string dengan pola yang banyak (multiple pattern search) yang sesuai dengan penelitian yang dilakukan dalam paper ini.

Dalam metode ini juga menggunakan metode k-gram. K-gram adalah rangkaian term dengan panjang K. Yang biasanya digunakan sebagai suatu term adalah kata. K-gram merupakan suatu metode yang digunakan untuk membangkitkan kata atau karakter. Metode ini digunakan untuk mengambil potongan – potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinu dibaca dari sumber hingga akhir.

Dalam algoritma ini juga menggunakan fungsi hashing dalam algoritma digunakan untuk menghindari perbandingan jumlah karakter yang kuadratik. Hashing sendiri merupakan suatu cara untuk mentransformasikan sebuah string menjadi suatu nilai yang unik dengan panjang tertentu yang berfungsi sebagai penanda pada string yang ditransformasikan tersebut. Fungsi yang digunakan untuk menghasilkan ini adalah fungsi hash dan nilai yang dihasilkan disebut dengan nilai hash. Algoritma Rabin Karp ini didasarkan jika nilai hash-nya sama maka kedua string yang dicocokkan pasti sama.



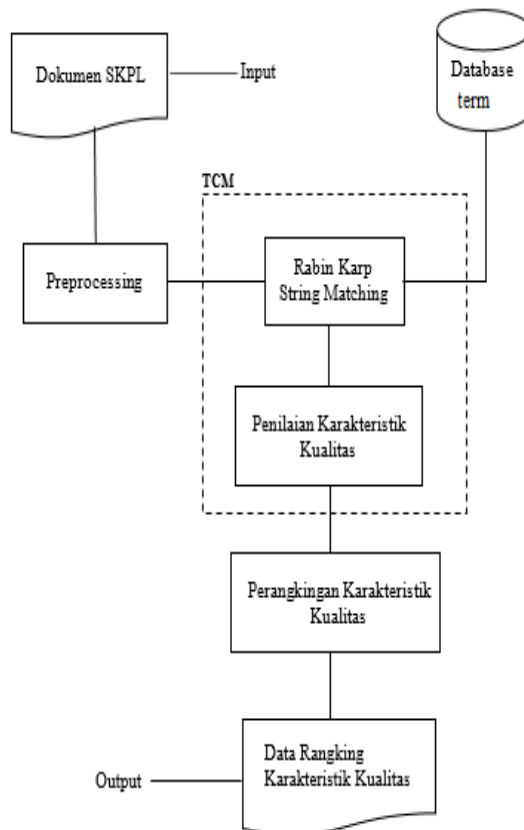
Gambar 3. Kategori karakteristik kualitas dalam ISO 9126 [9]

Algoritma Rabin Karp ini digunakan untuk

pencocokan string dengan memanfaatkan fungsi hash sebagai pembanding antara string yang dicari (a) dengan substring pada suatu teks (b). Apabila nilai hash pada perbandingan kedua string a dan b sama maka akan dilakukan perbandingan kembali terhadap karakter yang ada dalam kedua string tersebut. Jika karakter pada kedua string sama maka proses akan selesai atau mungkin dilanjutkan untuk pencocokan dengan pola string yang berbeda, tetapi jika tidak maka substring b akan bergeser ke kanan dan pergeseran tersebut akan dilakukan maksimal sebanyak (b-a) kali.

D. ISO 9126

Suatu kualitas dalam perangkat lunak telah memiliki standar yang tertulis dalam ISO 9126, dalam ISO 9126 disebutkan dalam bagian quality model for external and internal quality terdapat enam kategori karakteristik kualitas yaitu functionality, reliability, usability, efficiency, maintainability dan portability. Masing – masing kategori memiliki beberapa subkategori. semua subkategori yang terdapat pada keenam kategori tersebut yang digunakan dalam penelitian ini untuk melakukan pengidentifikasian pada dokumen SKPL kecuali subkategori compliance dalam reliability, usability, efficiency, maintainability dan portability. Bagan karakteristik kualitas tersebut dapat dilihat pada Gambar 3.



Gambar 4. Alur Metodologi Penelitian

III. METODOLOGI PENELITIAN

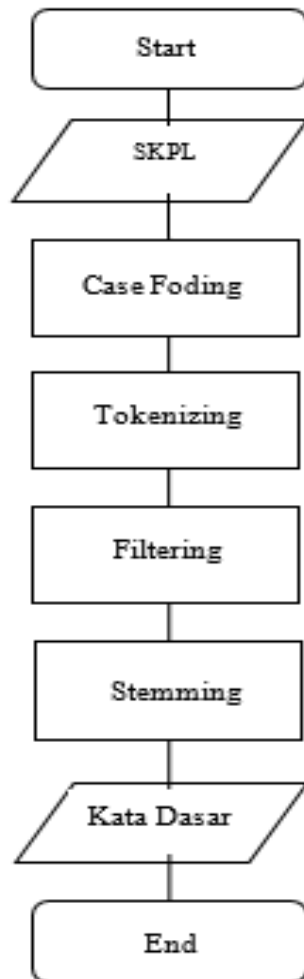
Alur penelitian yang dilakukan dalam paper ini digambarkan dalam Gambar 4. Penelitian ini terbagi

menjadi 3 tahapan yaitu tahap preprocessing, tahap TCM, dan tahap perancangan karakteristik kualitas yang dimunculkan.

Sesuai pada Gambar 4 diatas ketiga tahapan alur penelitian dapat dapat dijabarkan sebagai berikut :

#### A. Preprocessing

Tahap pertama dalam penelitian ini adalah tahap *preprocessing*, dimana seperti dalam Gambar 5 dalam tahap *preprocessing* ini terbagi menjadi empat tahap yaitu *case foding*, *tokenizing*, *filtering*, dan *stemming*.



Gambar 5. Tahap Preprocessing

Tahap pertama adalah *case foding*, dalam tahapan ini dokumen SKPL yang diinputkan dalam sistem akan diproses untuk diubah semua hurufnya menjadi huruf kecil. Huruf yang diterima dalam proses *case foding* adalah huruf “a” hingga “z”, karakter selain huruf yang diterima akan dihilangkan dan dianggap delimiter. Sebagai contoh dokumen kebutuhan fungsional dalam Gambar 2 semua huruf telah diubah menjadi huruf kecil

Tahapan kedua adalah proses *Tokenizing/parsing* yaitu tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Seperti misal salah satu kebutuhan fungsional yang digambarkan pada Gambar 2 yaitu “*bookmark shall be supported*” akan ditokenisasi menjadi “*bookmark*”, “*shall*”, “*be*”, dan “*supported*”.

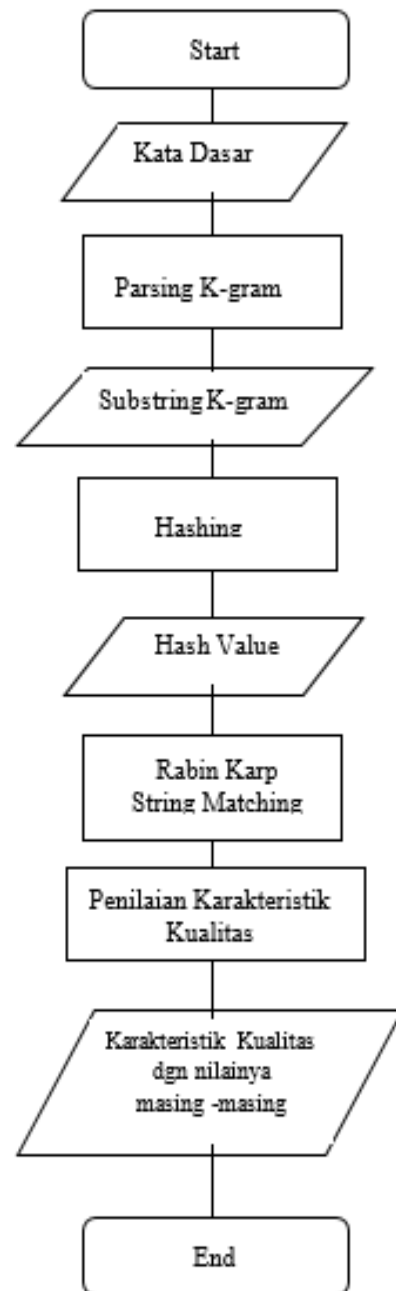
Tahapan berikutnya adalah *filtering*, proses *filtering* adalah proses penghilangan kata – kata dan tanda baca

seperti “*a*”, “*an*”, “*and*”, “*as*”, “*be*”, tanda baca koma dan sebagainya. Misal dari hasil tokenisasi sebelumnya maka kata “*be*” akan dihilangkan.

Tahapan terakhir adalah *stemming*, proses *stemming* adalah suatu proses pemotongan terhadap imbuhan seperti prefix (awalan) dan suffiks (akhiran) dan confix (awalan dan akhiran) sehingga akan didapatkan kata dasarnya. Misal dari contoh ditahap sebelumnya maka kata “*supported*” akan diubah menjadi “*support*”.

#### B. TCM (Term Characteristics Mapping)

Tahap kedua pada penelitian ini adalah tahap TCM, dimana dalam tahap TCM ini terbagi menjadi beberapa proses seperti pada Gambar 6.



Gambar 6. Tahap TCM

Didalam tahap inilah algoritma Rabin Karp digunakan untuk melakukan *string matching*. Sebelum melakukan *string matching*, proses pertama adalah kata

dasar yang didapatkan dalam proses *preprocessing* diproses dalam tahap *parsing* k-gram, dalam proses *parsing* k-gram ini panjang k-gram berubah – ubah sesuai dengan panjang karakter dalam *term* yang akan digunakan untuk melakukan mapping nantinya, misal jika *term*-nya adalah “*applicable*” maka panjang k-gramnya adalah 10 sesuai dengan jumlah karakter dalam *term* “*applicable*” tersebut, dalam proses ini nantinya akan menghasilkan data *substring* k-gram yang telah terpotong sesuai panjang k-gramnya.

Pada proses kedua data *substring* k-gram tersebut kemudian akan dilanjutkan pada proses *hashing*, dalam proses *hashing* ini data *substring* k-gram akan diubah menjadi suatu hash value, perubahan ke dalam hash value yaitu memanfaatkan nilai ASCII, misal karakter huruf “a” yaitu akan diubah menjadi nilai 97. *Hash value* ini yang akan digunakan untuk proses *string matching* sehingga akan diketahui berapa k-gram yang sama atau similar.

Tahap ketiga adalah *String matching* dilakukan antara string yang didapatkan dari dokumen SKPL dengan *string term* yang berhubungan dengan masing – masing karakteristik sesuai dengan ISO 9126 yang telah disimpan dalam database. Dalam tahap *string matching* juga dilakukan pengukuran similaritas dari string yang dicocokkan tersebut.

Setelah tahapan *string matching*, didapatkan data string dokumen SKPL yang cocok dengan string TCM. Dari data tersebut dilakukan proses penilaian karakteristik kualitas sesuai dengan persamaan (1) yang dijelaskan pada bab sebelumnya.

C. Perangkingan Karakteristik Kualitas

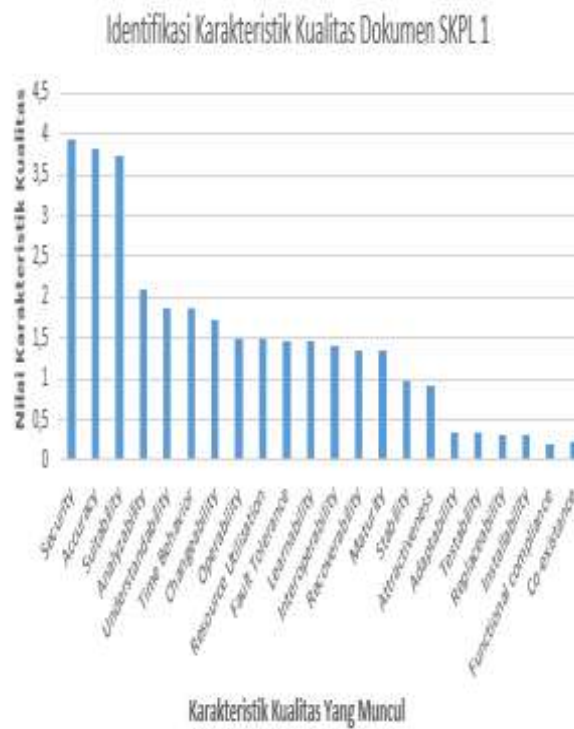
Tahap terakhir adalah tahap perangkingan karakteristik kualitas berdasarkan dengan nilai tiap karakteristik kualitas yang dihasilkan dari proses sebelumnya. Nilai pada tiap – tiap karakteristik ini dihitung berdasarkan persamaan (1). Sehingga nantinya akan didapatkan output berupa rangking karakteristik kualitas dari nilai yang terbesar hingga yang terkecil. Tujuan dari perangkingan ini adalah untuk mengetahui urutan karakteristik kualitas yang diutamakan untuk dilanjutkan pada tahapan selanjutnya dalam model kualitas yang diperbaiki dalam penelitian ini.

IV. HASIL DAN PEMBAHASAN

A. Dataset

Dataset yang digunakan dalam penelitian ini ada dua jenis, yaitu yang pertama adalah dataset daftar *term* yang didapatkan dari pendapat ahli di bidang pengembangan perangkat lunak dari DinusTech Semarang. Dari 22 subkarakteristik kualitas, data *term* yang didapatkan untuk keseluruhannya berjumlah 167 *term*. Dataset yang kedua adalah dokumen SKPL sistem manajemen informasi akademik sebanyak dua dokumen yang akan diambil kebutuhan fungsionalnya untuk diidentifikasi karakteristik kualitas didalamnya. Kebutuhan fungsional yang diambil dari dokumen SKPL tersebut kemudian akan diproses seperti dalam tahapan metodologi.

B. Hasil Pengujian



Gambar 5. Grafik hasil pemunculan karakteristik kualitas pada SKPL pertama

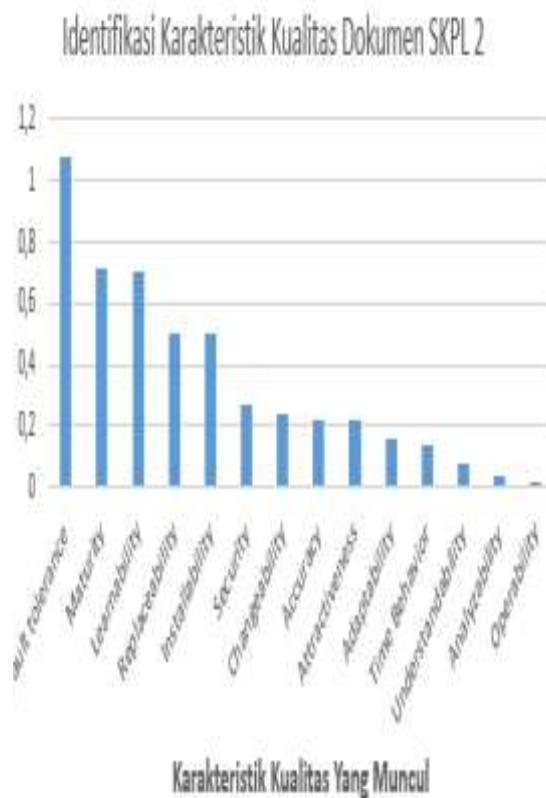
TABEL I  
RANGKING HASIL PEMUNCULAN KARAKTERISTIK KUALITAS SKPL 1

No.	Karakteristik Kualitas	Nilai
1	Security	3.936
2	Accuracy	3.828
3	Suitability	3.740
4	Analyzability	2.1
5	Understandability	1.877
6	Time Behavior	1.873
7	Changeability	1.735
8	Operability	1.502
9	Resource Utilization	1.482
10	Fault Tolerance	1.465
11	Learnability	1.455
12	Interoperability	1.415
13	Recoverability	1.360
14	Maturity	1.342
15	Stability	0.975
16	Attractiveness	0.916
17	Adaptability	0.342
18	Testability	0.34
19	Replaceability	0.305
20	Installability	0.305
21	Functional compliance	0.205
22	Co-existence	0.237

Dapat dilihat pada Gambar 7, hasil analisis spektrum dengan menggunakan TCM yaitu karakteristik kualitas yang muncul dari dokumen SKPL yang pertama adalah sebanyak 22 karakteristik kualitas dengan detail nilai untuk tiap – tiap karakteristik kualitas dapat dilihat pada Tabel I. Dari hasil tersebut dapat dilihat bahwa karakteristik kualitas dengan tiga nilai tertinggi yaitu *security* dengan nilai 3.936, *accuracy* dengan nilai



3.828, dan *suitability* dengan nilai 3.740 merupakan karakteristik kualitas yang termasuk dalam kategori *functional characteristics* sehingga dapat dianalisis bahwa dokumen SKPL pertama mengutamakan kualitas dari fungsional sistem yang akan dibangun, terutama terhadap kualitas dari sisi keamanan, akurasi, dan kesesuaian sistem. Dengan 22 karakteristik kualitas yang dimunculkan dengan analisis spektrum ini nantinya juga akan mempermudah pada tahap untuk mendapatkan kebutuhan kualitas yang sesuai dengan dokumen SKPL yang pertama ini.



Gambar 6. Grafik hasil pemunculan karakteristik kualitas pada SKPL kedua

Hasil analisis spektrum untuk dokumen SKPL yang kedua dapat dilihat pada grafik yang ada dalam Gambar 8, dalam grafik tersebut dapat dilihat bahwa karakteristik kualitas yang dimunculkan sebanyak 14 karakteristik kualitas dengan masing – masing detail nilai dapat dilihat pada Tabel II. Hasil ini lebih sedikit dibandingkan dengan hasil yang muncul dari dokumen SKPL yang pertama, karena kedetailan dan keterperincian dalam menjelaskan kebutuhan fungsional dalam dokumen SKPL juga akan berpengaruh ketika dilakukan proses analisis spektrum dengan TCM. Berbeda halnya dengan dokumen SKPL pertama yang mengutamakan kualitas dari karakteristik fungsional, dalam dokumen SKPL kedua ini lebih mengutamakan pada karakteristik kualitas *reliability* dan *usability*. Terbukti dari tiga karakteristik kualitas dengan nilai tertinggi adalah *fault tolerance*, *maturity* dan *learnability*, dimana *fault tolerance* dan *maturity* masuk ke dalam kategori *reliability characteristics* dan

*learnability* masuk ke dalam kategori *usability characteristics*. Dengan masing – masing detail nilai secara terurut adalah 1.076, 0,715, dan 0,699.

TABEL II  
RANGKING HASIL PEMUNCULAN KARAKTERISTIK KUALITAS SKPL 2

No.	Karakteristik Kualitas	Nilai
1	Fault tolerance	1.076
2	Maturity	0.715
3	Learnability	0.699
4	Replaceability	0.5
5	Installability	0.5
6	Security	0.269
7	Changeability	0.242
8	Accuracy	0.215
9	Attractiveness	0.215
10	Adaptability	0.161
11	Time Behavior	0.138
12	Understandability	0.080
13	Analyzability	0.038
14	Operability	0.015

Dengan melihat hasil munculnya karakteristik kualitas dari masing – masing pengujian yaitu 22 karakteristik kualitas untuk dokumen SKPL pertama, dan 14 karakteristik kualitas untuk dokumen SKPL kedua, maka terbukti bahwa perbaikan pada tahap spesifikasi karakteristik kualitas pada *Quality Models* dalam penelitian [4] dengan menggunakan pendekatan metode TCM berhasil dan lebih baik dalam memunculkan karakteristik kualitas yang terdapat dalam suatu kebutuhan fungsional. Jika hasil pengujian ini dibandingkan dengan hasil sebelum dilakukan perbaikan maka hasil yang didapatkan dari uji coba ini terbukti lebih lengkap karena dengan satu kali proses semua karakteristik kualitas dalam suatu dokumen SKPL dapat dimunculkan. Hasil uji coba yang didapatkan diharapkan dapat digunakan untuk tahapan berikutnya sesuai pada model kualitas yang telah dijelaskan pada bagian dasar teori.

## V. KESIMPULAN DAN SARAN

Dengan melihat hasil uji coba menggunakan pendekatan analisis spektrum dengan memanfaatkan metode TCM, maka pendekatan ini telah berhasil memperbaiki kelemahan pada tahap identifikasi dan tahap spesifikasi aspek / karakteristik kualitas. Kelemahan pada tahap identifikasi telah berhasil diperbaiki dengan mengubah identifikasi berbasis *Goal / tujuan* menjadi berbasis kebutuhan fungsional yang terdapat dalam dokumen SKPL sehingga untuk masuk ke dalam tahap spesifikasi aspek kualitas tidak hanya berdasarkan satu tujuan / *goal* saja. Kelemahan kedua pada tahap spesifikasi aspek / karakteristik kualitas berhasil diperbaiki dengan menggunakan pendekatan analisis spektrum dengan memanfaatkan metode TCM maka aspek kualitas yang dimunculkan tidak hanya satu aspek saja seperti sebelumnya tetapi dapat sekaligus dimunculkan dalam satu kali proses.

Metode TCM yang digunakan sangat bergantung pada daftar term yang saling berkaitan dengan tiap – tiap karakteristik kualitas sehingga secara tidak

langsung juga sangat bergantung kepada pendapat ahli untuk mendapatkan daftar term tersebut. Sehingga hasil identifikasi mungkin akan berbeda jika daftar term didapatkan dari pendapat ahli yang berbeda pula.

Dengan adanya perbaikan dalam tahap identifikasi dan tahap spesifikasi aspek kualitas, maka juga akan mendorong perubahan pada dua tahap berikutnya, yang akan menjadi fokus pada penelitian berikutnya.

#### DAFTAR PUSTAKA

- [1] J. D. Blaine and J. Cleland-Huang, "Software Quality Requirements: How to Balance Competing Priorities," *IEEE Software*, vol. 25, pp. 22-24, 2008.
- [2] V. Pasrija, S. Kumar and P. R. Srivastava, "Assessment of Software Quality: Choquet Integral Approach," *Procedia Technology* 6, vol. 6, p. 153 – 162, 2012. R. Bittencourt, R. Araujo and C. Cappelli, "Experiences on the use of business models for identifying quality requirements for information systems," in 2010 Seventh International Conference on the Quality of Information and Communications Technology, 2010.
- [3] R. Plösch, A. Mayr and C. Körner, "Collecting Quality Requirements Using Quality Models and Goals," in 2010 Seventh International Conference on the Quality of Information and Communications Technology, 2010.
- [4] H. Kaiya, T. Sato, A. Osada, N. Kitazawa and K. Kaijiri, "Toward Quality Requirements Analysis based on Domain Specific Quality Spectrum," in 23rd Annual ACM Symposium on Applied Computing, Fortaleza, Ceara, Brazil, 2008.
- [5] H. Kaiya, M. Tanigawa, S. Suzuki, T. Sato, A. Osada and K. Kaijiri, "Improving Realiability of Spectrum Analysis for Software Quality Requirements Using TCM," *IEICE TRANSACTIONS on Information and Systems*, Vols. E93-D, pp. 702-712, 2010.
- [6] F. Deissenboeck, E. Juergens, K. Lochmann and S. Wagner, "Software Quality Models: Purposes, Usage Scenarios and Requirements," in *Software Quality, 2009. WOSQ '09. ICSE Workshop on*, Vancouver, BC, 2009.
- [7] R. M. Karp and M. O.Rabin, "Efficient Randomized pattern-matching algorithms," *IBM J. RES. DEVELOP*, vol. 31, pp. 249 - 260, 1987.
- [8] ISO/IEC, "Information technology—Software product quality". Geneva 20 3 2000.