KLASIFIKASI KEBUTUHAN NON-FUNGSIONAL MENGGUNAKAN FSKNN BERBASIS ISO/IEC 25010

Lukman Hakim[1)], Siti Rochimah[2)], dan Chastine Fatichah[3)]
[1, 2, 3)] Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Sukolilo, Surabaya
e-mail: lukman.if.its@gmail.com[1)], mailto:siti@if.its.ac.id[2)], chastine.fatichah@gmail.com[3)]

ABSTRAK

Aspek kualitas kebutuhan non-fungsional memiliki peran yang sangat besar dalam kesuksesan pengembangan perangkat lunak. Namun, mengidentifikasi aspek kualitas kebutuhan non-fungsional merupakan hal yang sulit untuk dilakukan. Karena aspek kualitas kebutuhan non-fungsional tersebut ditemukan tidak lengkap, tersembunyi, dan tercampur di dalam kalimat-kalimat kebutuhan lain pada dokumen. Penelitian yang ada mampu mengidentifikasi aspek kebutuhan non-fungsional dengan melakukan klasifikasi. Meskipun standar kualitas ISO/IEC 9126 telah digantikan oleh ISO/IEC 25010 pada tahun 2011. Namun, pada proses pelabelan masih menggunakan standar kualitas ISO / IEC 9126. Oleh karena itu, standar kualitas yang digunakan untuk melabeli kalimat kebutuhan pada penelitian ini adalah ISO/IEC 25010. Sedangkan metode klasifikasi yang digunakan adalah Fuzzy Similarity based K-Nearest Neighbor (FSKNN). Metode klasifikasi yang digunakan diuji dengan menggunakan nilai tetangga terdekat 10, 20 dan 30. Pada penelitian ini metode FSKNN berhasil memeroleh nilai tertinggi berdasarkan ground truth pakar yaitu accuracy sebesar 98.15%, precision sebesar 6.15 dan recall 21.73.

***Kata Kunci**: FSKNN, ISO/IEC 25010, Klasifikasi, Non-fungsional.*

ABSTRACT

The quality aspect of non-functional requirements play a very big role in the success of software development. However, identifying aspects of quality of non-functional requirements is a difficult thing to do. Because the quality aspect of non-functional requirements is often found incomplete, hidden, and mixed in the sentences of other requirements in the document. Existing research has been able to identify aspects of non-functional requirements by classifying. Although the ISO/IEC 9126 quality standard has been replaced by ISO/IEC 25010 in 2011. However, the labeling process still uses the ISO/IEC 9126 quality standard. Therefore, the quality standard used to label the sentence of need in this research is ISO/IEC 25010. While the classification method used is Fuzzy Similarity based K-Nearest Neighbor (FSKNN). The classification method used was tested using the nearest neighbor values 10, 20 and 30. In this research the FSKNN method succeeded in obtaining the highest score based on expert ground truth, namely accuracy by 98.15%, precision by 6.15% and recall by 21.73%.

***Keywords**: Classification, FSKNN, ISO/IEC 25010, Non-functional.*

I. PENDAHULUAN

KEBUTUHAN non-fungsional yang berkaitan erat dengan aspek kualitas perangkat lunak memiliki peran yang sangat besar untuk mendukung kesuksesan pengembangan perangkat lunak [1][2]. Non-fungsional merupakan kendala pada layanan atau fungsi yang ditawarkan oleh sistem secara keseluruhan seperti *usability*, *reliability*, dan *security* [3]. Non-fungsional berfokus pada produktivitas yang buruk, pemrosesan yang lambat, biaya yang relatif tinggi, kualitas yang rendah, dan ketidak puasan pelanggan [4].

Mengidentifikasi aspek kualitas kebutuhan non-fungsional merupakan hal yang sulit untuk dilakukan. Karena aspek kualitas kebutuhan non-fungsional tersebut ditemukan tidak lengkap, tersembunyi, dan tercampur di dalam kalimat-kalimat kebutuhan lain pada dokumen. Permasalahan tersebut menyebabkan aspek kualitas kebutuhan non-fungsional sering terlewatkan selama proses pengembangan sistem perangkat lunak [5][6]. Kalimat kebutuhan yang memiliki kaitan dengan aspek kualitas kebutuhan non-fungsional pada umumnya ditulis dalam bentuk bahasa alami. Kalimat kebutuhan yang ditulis dalam bahasa alami biasanya memiliki kerancuan. Kerancuan akan mempersulit analis dalam mengidentifikasi aspek kualitas kebutuhan non-fungsional yang terdapat di dalamnya [7]. Oleh karena itu dibutuhkan suatu cara untuk dapat mengidentifikasi aspek kualitas kebutuhan non-fungsional, salah satunya dengan cara melakukan klasifikasi dari kalimat-kalimat kebutuhan yang tertulis dalam dokumen kebutuhan tersebut [2].

Beberapa metode telah digunakan untuk memecahkan masalah identifikasi aspek kualitas. Diantaranya adalah algoritma Naive Bayes, algoritma klasifikasi model Probabilistik, penambangan term indikator menggunakan metode Cosine Similarity, dan klasifikasi menggunakan metode semantik-FSKNN. Metode pertama mengkombinasikan Naive Bayes dengan Expectation Maximization (EM) [6]. Penelitian ini hanya mampu mengklasifikasikan kalimat kebutuhan non-fungsional terbatas ke dalam satu aspek kualitas (*single-label*). Metode kedua terdiri dari dua tahap yaitu tahap awal dan tahap pengembangan. Tahap awal pada penelitian ini

terdiri dari dua proses. Proses pertama adalah menyediakan sekumpulan kata kunci baku. Kata kunci baku tersebut diambil dari katalog metode operasionalisasi softgoal keamanan dan kinerja pada *Softgoal interdependency graphs* (SIG). Sedangkan proses kedua adalah menggunakan kata kunci baku tersebut untuk mengklasifikasi setiap tipe non-fungsional dari kebutuhan yang tidak terklasifikasi ke dalam lebih dari satu aspek kualitas (*multi-label*) [8]. Tahap pengembangan pada penelitian ini terdiri dari tiga proses yaitu pelabelan data latih, penambangan *term* indikator, dan klasifikasi. Tahap pengembangan pada penelitian ini memiliki tingkat subjektivitas yang tinggi karena proses pelabelan data latih dilakukan secara manual. Disamping itu, banyaknya aspek kualitas yang ditawarkan oleh model-model kualitas seperti model *McCall*, *Boehm*, *Dromey*, *FURPS*, *BNN*, *Star*, dan *ISO* menyebabkan kebingungan dalam menentukan aspek kualitas [9].

Metode ketiga yaitu menggunakan standar kualitas ISO/IEC 9126 pada fase penambangan untuk mendapatkan kata kunci [10]. Namun, penelitian ini belum mempertimbangkan faktor hubungan semantik antara *term-term* relevan dengan setiap aspek kualitas yang diproses selama pelatihan. Faktanya faktor semantik dapat meningkatkan hasil klasifikasi menjadi lebih baik [11]. Metode keempat berusaha menambahkan pengukuran semantik ke dalam proses algoritma FSKNN[2]. Meskipun standar kualitas ISO/IEC 9126 telah digantikan oleh ISO/IEC 25010 pada tahun 2011. Namun, standar kualitas ISO/IEC 9126 masih digunakan oleh kedua penelitian diatas untuk melakukan proses pelabelan.

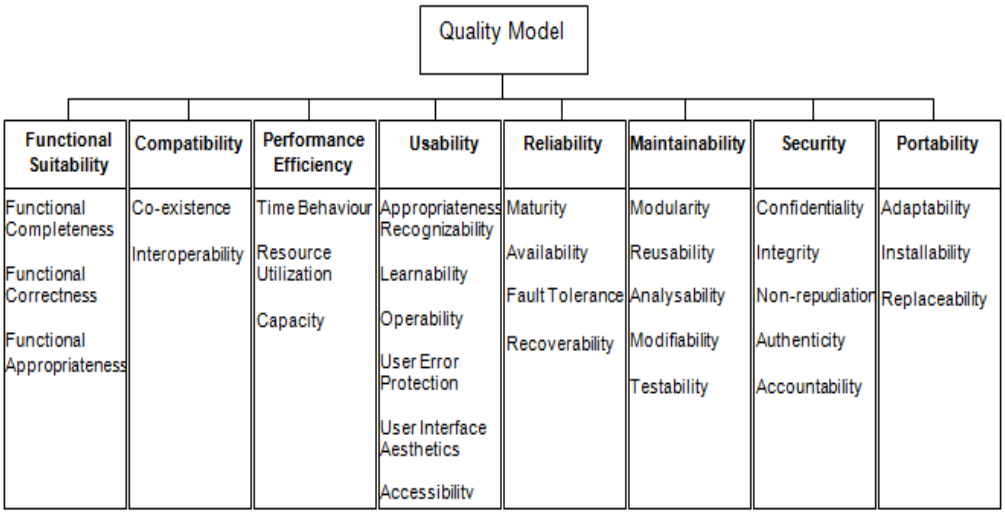
Evaluasi kritis dari penelitian yang dilakukan oleh Al-Kilidar [12] dan Al-Qutaish [13] mengungkapkan ambiguitas dalam enam sub-atribut dalam struktur hirarkis ISO / IEC 9126. Sejalan dengan evaluasi ini, Behkamal [9] hanya menggunakan 21 sub-atribut dari total 27 sub-atribut yang dimiliki oleh ISO/IEC 9126. Oleh karena itu, standar kualitas yang digunakan dalam penelitian ini adalah ISO/IEC 25010. Disamping itu, evaluasi kritis yang dilakukan oleh Ramadahani [2] mengungkapkan penambahan pengukuran semantik ke dalam proses algoritma FSKNN membuat peningkatan kinerja klasifikasi tidak signifikan. Hal ini dikarenakan data yang dibentuk untuk pola pelatihan tersaring lebih ketat. Terbukti pada evaluasi metrik yang digunakan, peningkatan nilai *precision* begitu kecil yaitu sebesar 5.8 % dan terjadi penurunan nilai recall sebesar 4 %. Oleh karena itu, metode klasifikasi yang digunakan dalam penelitian ini adalah FSKNN tanpa penambahan pengukuran semantik.

II. STUDI LITERATUR

A. *ISO/IEC 25010:2011*

ISO 25010:2011 menggantikan ISO/IEC 9126-1: 2001, yang telah direvisi secara teknis. ISO/IEC 25010:2011 adalah bagian dari rangkaian *Standart International SquaRE*, yang terdiri dari enam divisi yaitu *Quality Management*, *Quality Model*, *Quality Measurement*, *Quality Requirements*, *Quality Evaluation*, *dan SquaRE Extension*.

Mengukur kualitas sistem dengan cara seberapa baik suatu sistem dapat memenuhi berbagai kebutuhan pemangku kepentingan. Oleh karena itu, perlu untuk mempertimbangkan kualitas dari perspektif pemangku kepentingan yang berbeda. Model-model kualitas menyediakan kerangka untuk mengumpulkan kebutuhan pemangku kepentingan. Di antara dokumen dalam standar ISO / IEC, 25010:2011 "Model Kualitas" mendefinisikan model kualitas produk yang terdiri dari delapan atribut yang selanjutnya dibagi lagi menjadi sub-atribut. Model ini dipahami sebagai model struktural yang didefinisikan oleh SQuaRE sebagai; "Kualitas model: set



Gambar 1. ISO/IEC 25010:2011 untuk Quality Model

karakteristik yang ditetapkan, dan hubungan di antara mereka.". Standar ini juga memberikan definisi dari masing-masing atribut kualitas dan sub-atribut [14].

Strategi Dekomposisi Hierarki ini dimaksudkan untuk menyediakan sarana untuk menangani kompleksitas konseptual. Penanganan kompleksitas tersebut bertujuan untuk menggambarkan Kualitas sebagai properti multi-dimensi. Selain itu, Dekomposisi Hierarki ini juga menyediakan sarana untuk memperoleh atribut terukur sederhana yang cocok untuk kombinasi lebih lanjut ke dalam indeks kualitas [14].

Dalam penelitian ini yang digunakan sebagai dasar acuan dalam model aspek kualitas adalah dokumen ISO 25010:2011 yang berisi 1 model kualitas yaitu kualitas produk perangkat lunak. Model tersebut terdiri dari delapan atribut kualitas yang terbagi lagi menjadi 31 sub-atribut kualitas, detail sub-atribut tersebut dapat dilihat pada Gambar 1.

B. GROUND TRUTH

Ground truth adalah hasil kesepakatan antara satu pakar dengan pakar yang lain. Pada penelitian ini, pakar dilibatkan untuk melakukan pelabelan secara manual antara kalimat kebutuhan dengan aspek kualitas. Misalnya pakar pertama melabeli kalimat kebutuhan ke 1 ke dalam aspek kualitas *Capacity*, dan *Operability*. Sedangkan pakar kedua melabeli kalimat kebutuhan ke 1 ke dalam aspek kualitas *Capacity* dan *Learnability*. Hasilnya ditunjukkan pada Tabel I, Tabel II dan Tabel III.

Berdasarkan ketiga tabel diatas diperoleh hasil kesepakatan antara kedua pakar terhadap tiga aspek kualitas. Hasil kesepakatan tersebut akan digunakan sebagai Ground truth. Ground truth tersebut diperoleh dengan menggunakan Gwet's AC1.

C. GWET'S AC1

Gwet's AC1 adalah salah satu metode yang digunakan untuk mengukur indeks kesepakatan antara dua pengamat. Gwet's AC1 menunjukkan pendekatan yang lebih dapat diandalkan dibandingan dengan Cohen Kappa[15]. Penghitungan nilai AC1, hasil dari observasi pengamat dimasukkan dalam matrik 2x2 seperti yang ditunjukkan pada Tabel IV.

Pada Tabel IV terdapat dua pakar yang mengklasifikasikan N subjek ke dua kemungkinan kategori. Dua kategori dilabelkan sebagai "ya" dan "tidak". Variabel aa diklasifikasikan oleh dua pakar sebagai ya, Variabel bb diklasifikasikan oleh pakar 2 sebagai tidak, Variabel dd diklasifikasikan oleh dua pakar sebagai tidak, dan Variabel cc diklasifikasikan oleh pakar 1 sebagai tidak. Variabel B1 menunjukkan jumlah dari variabel aa dan variabel bb, Variabel B2 menunjukkan jumlah dari variabel cc dan variabel dd, Variabel A1 menunjukkan jumlah dari variabel aa dan variabel cc, dan variabel A2 menunjukkan jumlah dari variabel bb dan variabel dd.

Perhitungan AC1-statistic :

Kesepakatan yang terobservasi, $Po = \frac{a+d}{N}$ (1)

Dimana variabel aa adalah banyaknya data yang dikelompokkan ke dalam kategori "Ya" oleh kedua pakar sedangkan variabel dd adalah banyaknya data yang dikelompokkan ke dalam kategori "tidak" oleh kedua pakar.

Sedangkan untuk menghitung probabilitas acak antara kedua pakar yaitu :

TABEL I
HASIL KESEPAKATAN KEDUA PAKAR TERHADAP KALIMAT KEBUTUHAN KE 1 PADA ASPEK KUALITAS CAPACITY

Pakar 1	Pakar 2	
	Ya	Tidak
Ya	2	0
Tidak	0	0

TABEL II
HASIL KESEPAKATAN KEDUA PAKAR TERHADAP KALIMAT KEBUTUHAN KE 1 PADA ASPEK KUALITAS OPERABILITY

Pakar 1	Pakar 2	
	Ya	Tidak
Ya	1	1
Tidak	0	0

TABEL III
HASIL KESEPAKATAN KEDUA PAKAR TERHADAP KALIMAT KEBUTUHAN KE 1 PADA ASPEK KUALITAS LERANABILITY

Pakar 1	Pakar 2	
	Ya	Tidak
Ya	1	0
Tidak	1	0

TABEL IV
HASIL KESEPAKATAN KEDUA PAKAR TERHADAP KALIMAT KEBUTUHAN KE 1 PADA ASPEK KUALITAS LEARNABILITY

Pakar 1	Pakar 2		
	Ya	Tidak	Total
Ya	aa	bb	B1=aa+bb
Tidak	cc	dd	B2=cc+dd
Total	A1=aa+cc	A2=bb+dd	N

- Pakar 1 mengatakan Ya sebanyak 1 dan Tidak sebanyak 0.
- Pakar 2 mengatakan Ya sebanyak 1 dan Tidak sebanyak 0.

Diketahui bahwa kedua pengamat mengatakan Ya sebanyak 100 %. Jadi, probabilitas yang diharapkan dari kedua pengamat mengatakan Ya secara acak ditunjukkan pada persamaan 1, 2 dan untuk probabilitas keseluruhan ditunjukkan pada persamaan 3. Sedangkan untuk menghitung nilai indeks Kappa menggunakan Cohen Kappa statistik dengan persamaan 4.

$$\text{Pya}= \frac{aa+bb}{\text{aa+bb + cc+dd}} * \frac{aa+cc}{\text{aa+bb + cc+dd}} \quad (2)$$

$$\text{Ptidak}= \frac{cc+dd}{\text{aa+bb + cc+dd}} * \frac{bb+dd}{\text{aa+bb + cc+dd}} \quad (3)$$

$$\text{Pe} = \text{Pya} + \text{Ptidak} \quad (4)$$

$$\text{k}= \frac{Po+Pe}{1-\text{pe}} \quad (5)$$

Probabilitas *chance-agreement*, (6)

$$e(\gamma) = 2P_1(1 - P_1)$$

AC1-statistic, $AC1 = \frac{P-e(\gamma)}{1- e(\gamma)}$ (7)

Berikut adalah contoh perhitungan *AC1-static*:

Terdapat dua orang pengamat yang dimintai pendapatnya terhadap satu kalimat kebutuhan berdasarkan satu aspek kualitas ISO/IEC 25010:2011, apakah kalimat kebutuhan tersebut masuk ke dalam aspek kualitas *Functional Completeness* atau tidak. Hasil pengamatan terlihat seperti Tabel V di bawah ini.

Jika dihitung menggunakan indeks Cohen Kappa statistik maka didapat nilai indeks Kappa sesuai persamaan 1, 2, 3, 4 dan 5 sebagai berikut :

$\text{Po} = \frac{\text{aa+dd}}{N} = \frac{2+0}{2} = 1$

$\boldsymbol{Pya} = \left(\frac{\mathbf{2}}{\mathbf{2}}\right)\mathbf{x}\left(\frac{\mathbf{2}}{\mathbf{2}}\right) = 1$

$\boldsymbol{Ptidak} = \left(\frac{\mathbf{0}}{\mathbf{2}}\right)\mathbf{x}\left(\frac{\mathbf{0}}{\mathbf{2}}\right) = 0$

$\boldsymbol{Pe} = \mathbf{1} + \mathbf{0} = 1$

Kappa = (1 – 1) / (1-1) = 0

Nilai indeks Cohen Kappa yang dihasilkan pada data pengamatan Tabel V sangatlah rendah yaitu 0, hal ini dikarenakan pengamatan kedua pakar hanya berdasarkan satu aspek kualitas saja. Hasilnya akan tetap bernilai 0 meskipun kedua pengamat sudah menyatakan persetujuannya dengan memilih kategori Ya. Oleh karena itu, penelitian ini menggunakan Gwet's AC1. Gwet's AC1 terbukti mampu menghasilkan nilai yang benar sesuai dengan kesepakatan kedua pengamat. Berikut perhitungan dari Gwet's AC1sesuai persamaan 6, dan 7 :

$e(\gamma) = 2\left(\frac{2+2}{2\ x\ 2}\right)\left(1 - \frac{2+2}{2\ x\ 2}\right) = 2 * 0 = 0$

AC1 = (1 -0) / (1 - 0) = 1

Berdasarkan nilai AC1, maka dapat disimpulkan bahwa kalimat kebutuhan masuk ke dalam aspek kualitas *Functional Completeness.*

TABEL V
HASIL KESEPAKATAN KEDUA PAKAR TERHADAP KALIMAT KEBUTUHAN KE 1 PADA ASPEK KUALITAS FUNCTIONAL COMPLETENESS

Pakar 1	**Pakar 2**		
	Ya	**Tidak**	**Total**
Ya	2	0	2
Tidak	0	0	0
Total	2	0	2

TABEL VI
CONTOH HASIL PELABELAN

KK	Aspek Kualitas		
	Capacity	**Operability**	**Learnability**
1	√	√	√
2	√		√
3		√	
...			
225			√

Perhitungan AC1 ini diujikan terhadap hasil pelabelan manual yang dilakukan oleh tiga orang pakar dibidang requirement. Masing-masing pakar melabeli 225 kalimat kebutuhan non-fungsional ke dalam lebih dari satu aspek kualitas (*multi-label*). Contoh hasil pelabelan beberapa kalimat kebutuhan non-fungsional ditunjukkan pada Tabel VI.

III. METODOLOGI PENELITIAN

Secara umum, ada empat tahapan yang dilakukan untuk mengidentifikasi aspek kualitas kebutuhan non fungsional yaitu : praproses, pembobotan TF-IDF, pengukuran nilai kemiripan menggunakan metode cosine similarity, dan klasifikasi menggunakan metode FSKNN.

A. Praproses

Kalimat kebutuhan pada keseluruhan dataset terlebih dahulu melalui tahapan praproses. Pada tahap ini yang pertama dilakukan adalah *tokenization* dan *case folding*. *Tokenization* merupakan metode yang digunakan untuk mengubah kalimat menjadi kata-kata.Sedangkan *case folding* merupakan metode yang digunakan untuk merubah huruf menjadi huruf kecil. Contoh dari *tokenization dan case folding* misalnya salah satu kalimat kebutuhan yaitu "*The system must be maintainable"* akan ditokenisasi menjadi "*the", "system", "must", "be",* dan *"maintainable".*

Langkah selanjutnya dalam tahap ini adalah melakukan *stopword removal* dengan menghilangkan kata, tanda baca, simbol dan bilangan yang tidak berpotensi menjadi indikator topik pada kalimat kebutuhan. Contoh dari *stopword removal* yaitu "*a*", "*an*", "*and*", "*as*", "*be*", tanda baca koma dan sebagainya.
Setelah *stopword removal* selesai kemudian dilakukan *stemming* untuk menjadikan teks menjadi kata dasar.
Misal dari contoh kalimat kebutuhan pada tahap sebelumnya maka kata "*maintainable*" akan diubah menjadi "*maintan*". Tahapan praproses ini akan diterapkan untuk kalimat kebutuhan pada keseluruhan dataset maupun standard acuan model kualitas ISO/IEC 25010.

B. Pembobotan TF-IDF

Term frekuensi(tf) adalah menghitung jumlah kemunculan (*f*) *term* t_i dalam setiap kalimat kebutuhan d_j. Untuk proses perhitungan *tf* bisa dilihat pada persamaan (8).

$$TF = (t_i, D) \tag{8}$$

Inverse document frequency (idf) adalah perhitungan bobot *term* dari suatu kalimat kebutuhan terhadap kalimat kebutuhan lain. Dalam idf, term yang memiliki nilai yang sangat tinggi apabila term tersebut jarang sekali muncul. Log dari jumlah kalimat kebutuhan secara keseluruhan D dibagi dengan jumlah kalimat kebutuhan d yang mengandung term t_i menghasilkan sebuah bobot *term* terhadap kalimat kebutuhan yang ada. Untuk proses perhitungan *idf* bisa dilihat pada persamaan 9.

$$IDF = log \frac{N}{df(t_i, D)} + 1 \tag{9}$$

Proses selanjutnya adalah mengalikan tf dan idf. Proses ini perlu dilakukan normalisasi agar nilai bobot yang didapatkan berada dalam rentang nilai 0 hingga 1. Untuk proses perhitungan *tf*idf* bisa dilihat pada persamaan 10.

$$\omega_K(t_i) = \frac{tf(t_i, K) \times log \frac{N}{df(t_i, K)} + 1}{\sqrt{\sum_{t_i}(tf(t_i, K) \times log \frac{N}{df(t_i, K)} + 1)^2}} \tag{10}$$

C. Pengukuran Nilai Kemiripan menggunakan metode cosine similarity

Cosine Similarity digunakan untuk mengukur kemiripan antara kalimat kebutuhan dan definisi aspek kualitas menggunakan ISO/IEC 25010. Kalimat kebutuhan memiliki nilai kemiripan tertinggi apabila memperoleh nilai 1 dan memiliki nilai kemiripan terendah apabila memperoleh nilai 0. Cosine Similarity bisa dilihat pada persamaan 11.

$$Cos(Q,K) = \sum_{r=1}^{M} \omega_Q(t_i) \times \omega_K(t_i) \tag{11}$$

Dimana $\omega_Q(t_i)$ merupakan bobot dari term t_i yang berasal atribut aspek kualitas, sedangkan $\omega_K(t_i)$ merupakan bobot yang didapatkan dari term t_i yang berasal suatu kalimat kebutuhan *K*. Dengan menggunakan penentuan nilai *threshold* tertentu maka akan dapat dihasilkan kalimat - kalimat kebutuhan *K* yang terlabeli ke dalam satu atau lebih atribut aspek kualitas yang kemudian dapat digunakan sebagai data pelatihan dalam algoritma FSKNN.

D. Klasifikasi multi label menggunakan metode FSKNN

Proses pelatihan FSKNN terdiri dari dua tahap: pengelompokan pola training ke dalam kluster dan perhitungan prior probability dan nilai likelihoods berbasis KNN.

1) Pengelompokan Pola Training ke dalam kluster

Dokumen pelatihan $d_1, d_2, \dots, d_l$ dikelompokkan ke dalam sejumlah kluaster berbasis *fuzzy similarity measure*. Distribusi *term* t_i pada kategori c_j, dimana $dt(t_i, c_j)$ dan $dd(t_i, c_j)$ didefinisikan sebagai berikut:

$$dt(t_i, c_j) = \frac{\sum_{v=1}^{l} w_{iv} y_{jv}}{\sum_{v=1}^{l} w_{iv}} \tag{12}$$

$$dd(t_i, c_j) = \frac{\sum_{v=1}^{l} sgn(w_{iv}) y_{jv}}{\sum_{v=1}^{l} y_{iv}} \tag{13}$$

$$sgn(x) = \begin{cases} 1, & jika\ x > 0 \\ 0, & jika\ x = 0 \end{cases} \tag{14}$$

Sehingga tentunya akan didapatkan $0 \leq dt(t_i, c_j)$ dan $dd(t_i, c_j) \leq 1$. Tahapan selanjutnya mendefinisikan derajat keanggotaan t_i terhadap kategori c_j sebagai berikut :

$$\mu_R(t_i, c_j) = \frac{dt(t_i, c_j)}{\max\limits_{1 \leq u \leq m, 1 \leq v \leq p} dt(t_u, c_v)} \times \frac{dd(t_i, c_j)}{\max\limits_{1 \leq u \leq m, 1 \leq v \leq p} dd(t_u, c_v)} \tag{15}$$

Langkah berikutnya yaitu menentukan similaritas menggunakan *fuzzy similarity* dari setiap dokumen *d*, $d = \langle w_1, w_2, \dots, w_m \rangle$, terhadap kategori $c_j$sebagai berikut:

$$sim(d, c_j) = \frac{\sum_{i=1}^{m} \mu_R(t_i, c_j) \otimes \mu_d(t_i)}{\sum_{i=1}^{m} \mu_R(t_i, c_j) \oplus \mu_d(t_i)} \tag{16}$$

Dimana $\otimes$ dan $\oplus$ merupakan *fuzzy t-norm* dan *t-conorm* yang secara berurutan didefinisikan sebagai berikut :

$$x \otimes y = x \times y \tag{17}$$

$$x \oplus y = x + y - x \times y \tag{18}$$

$$\mu_d(t_i) = \frac{w_i}{\max\limits_{1 \leq v \leq m} w_v} \tag{19}$$

$\mu_d(t_i)$ adalah derajat keanggotaan dari *term* t_i terhadap *d*. Tahapan terakhir adalah mendefinisikan derajat keanggotaan dari dokumen *d* terhadap kategori c_j sebagai berikut :

$$\mu_{c_j}(d) = \frac{sim(d, c_j)}{\max\limits_{1 \leq v \leq p} sim(d, c_v)} \tag{20}$$

```
for untuk setiap dokumen pelatihan d_i, 1≤i≤l
    for untuk setiap kategori c_j, 1≤j≤p
        if( μ_{c_j}(d_i) ≥ α )
        then
for untuk setiap dokumen pelatihan d_u, 1≤u≤l
    for untuk setiap kluster S_v, 1≤v≤p
        if (d_u ∈ S_v )
        then G_u = G_u U S_v
```

Gambar 2. Pseodocode proses pengelompokan - search set G_i

Untuk mendefinisikan *p* kluster, $S_1, S_2, \dots, S_p$, adalah sebagai berikut :

$$S_v = \{d_u | \mu_{c_v}(d_u) \geq \alpha, \quad 1 \leq u \leq l\} \tag{21}$$

α adalah ambang batas yang didefinisikan oleh user untuk digunakan dalam proses pelatihan. *Pseudo-code* ditunjukkan pada Gambar 2, akan menghasilkan keluaran berupa *search set* $G_1, G_2, \dots, G_l$. *Search set* tersebut akan digunakan untuk menentukan data tetangga terdekat yang dapat membantu melakukan perhitungan nilai *prior probability* dan nilai *likelihood* pada tahapan berikutnya.

2) Perhitungan prior probability dan nilai likelihoods berbasis KNN

Diberikan $P(H_j)$ yang merupakan *prior probability* dan $P(E|H_j)$ yang merupakan suatu kelas *likelihood*. Perhitungan probabilitas ini dilakukan dari pola pelatihan yang didapatkan sebelumnya, sebagai berikut :

$$P(H_j = 1) = \frac{s + \sum_{i=1}^{l} y_{ji}}{2s + l} \tag{22}$$

$$P(H_j = 0) = 1 - P(H_j = 1) \tag{23}$$

Dimana *s* merupakan nilai *smoothing constant*. Kelas *likelihood* $P(E|H_j)$ memiliki nilai *E. Nilai E tersebut* dapat bernilai 0,1, ... , atau *k*. Setiap kalimat kebutuhan d_i, $1 \leq i \leq l$, dimana $N^i = \langle d_{v1}, d_{v2}, \dots, d_{vk} \rangle$ merupakan tetangga terdekat yang diambil dari *search set* G_i dan $n^i = \langle n_1^i, n_2^i, \dots, n_p^i \rangle$, yang merupakan vektor jumlah label yang didefinisikan :

$$n_j^i = \sum_{r=v_1}^{v_k} y_{jr} \tag{24}$$

Untuk $1 \leq j \leq p$. Tahap selanjutnya didefinisikan :

$$Z(e,j) = \sum_{i=1}^{l} y_{ji}\, \delta_{ei}(j) \tag{25}$$

$$\tilde{Z}(e,j) = \sum_{i=1}^{l} \tilde{y}_{ji}\, \delta_{ei}(j) \tag{26}$$

Dimana $\tilde{y}_{ji} = 1 - y_{ji}$dan

$$\delta_{ei}(j) = \begin{cases} 1, jika\ e = n_j^i \\ 0, jika\ e \neq n_j^i \end{cases} \tag{27}$$

Kemudian didefinisikan *likelihoods*, sebagai berikut :

$$P(E = e|H_j = 1) = \frac{s + Z(e,j)}{(k+1)s + \sum_{v=0}^{k} Z(v,j)} \tag{28}$$

$$P(E = e|H_j = 0) = \frac{s + \tilde{Z}(e,j)}{(k+1)s + \sum_{v=0}^{k} \tilde{Z}(v,j)} \tag{29}$$

3) Klasifikasi FSKNN

Maximum a posteriori (MAP) adalah estimasi yang digunakan oleh algoritma FSKNN pada proses testing, untuk menentukan aspek kualitas mana yang memiliki hubungan dengan d^t. Hubungan tersebut didapatkan dengan menghitung vektor label $y^t = \langle y_1^t, y_2^t, \dots, y_p^t \rangle$ dari kalimat kebutuhan d^t dengan menggunakan estimasi MAP sebagai berikut:

$$y_j^t \begin{cases} 1, jika\ P\left(H_j = 1 \middle| E = n_j^t\right) > P\left(H_j = 0 \middle| E = n_j^t\right) \\ 0, jika\ P\left(H_j = 0 \middle| E = n_j^t\right) > P\left(H_j = 1 \middle| E = n_j^t\right) \\ R[0,1], otherwise \end{cases} \quad (30)$$

H_j merupakan variabel acak untuk mengetahui kalimat kebutuhan tertentu masuk ke dalam aspek kualitas c_j $(H_j = 1)$ atau tidak $(H_j = 0)$. Sedangkan variabel E merupakan jumlah kalimat kebutuhan dalam N^t yang berhubungan dengan kategori c_j, dan $R[0,1]$ mengindikasikan 0 atau 1 dipilih secara acak. Dengan menggunakan *Bayes Rule* didapatkan :

$$P\left(H_j = b \middle| E = n_j^t\right) = \frac{P(H_j=b)P\left(E=n_j^t \middle| H_j=b\right)}{P(E=n_j^t)} \quad (31)$$

E. Evaluasi

Penelitian ini mengusulkan metode pengukuran menggunakan *accuracy*, *precision*, dan *recall*. Proses perhitungan *accuracy*, *precision*, dan *recall* ditentukan dari prediksi informasi pada nilai aktual yang diwakili oleh True Positive(TP), True Negative(TN), False Positive(FP), dan False Negative(FN). Persamaan ketiga dari metode evaluasi ditunjukkan dalam persamaan 32 untuk *accuracy*, persamaan 33 untuk *precision*, dan persamaan 34 untuk *recall*.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (32)$$

$$Precision = \frac{TP}{TP+FP} \quad (33)$$

$$Recall = \frac{TP}{TP+FN} \quad (34)$$

IV. UJI COBA

Pengujian dilakukan menggunakan dataset *voip* dan *geolocation* dengan jumlah 1366 kalimat kebutuhan. Kedua dataset tersebut terdiri dari 1141 kebutuhan fungsional dan 225 kebutuhan non-fungsional [8]. Dimana 1366 kalimat kebutuhan tersebut akan dilabeli secara otomatis dengan tahapan yang telah dijelaskan pada bab sebelumnya, dari 1366 kalimat kebutuhan akan diambil 60% (820 kalimat kebutuhan) untuk digunakan sebagai data latih dan 40% (546 kalimat kebutuhan) digunakan untuk data uji. Untuk mengetahui kinerja dari masing -

TABEL VII
PEMBENTUKAN KLUSTER SET

KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K	KK Uji	K
2	34	22	28	45	28	94	18	114	26	183	26	294	27	416	1	479	77	512	77
3	28	24	28	47	8	95	18	116	13	185	44	296	1	428	26	481	83	513	77
4	29	25	28	48	14	96	26	117	44	200	51	300	6	435	13	483	43	514	26
6	28	26	28	58	19	97	26	124	26	209	4	304	18	438	26	484	15	515	26
7	28	27	28	64	1	98	26	125	26	212	18	315	7	439	44	485	15	518	26
10	27	28	28	82	2	101	26	128	7	223	44	333	1	450	27	486	15	519	38
11	28	29	28	83	3	102	26	132	26	224	26	335	26	451	57	487	15	520	15
12	28	31	28	84	26	103	26	135	39	226	51	371	4	452	57	491	44	522	18
13	28	32	34	86	26	104	36	146	18	238	18	374	4	455	26	492	77	524	50
14	28	33	6	87	18	105	26	154	51	240	1	377	4	458	1	493	17	529	77
15	28	36	32	89	28	106	26	155	26	250	6	383	26	460	1	494	77	530	62
16	28	37	34	90	44	107	26	158	51	253	51	387	26	462	31	495	75	531	77
17	28	39	28	91	18	110	26	163	18	256	1	394	27	463	66	497	70	532	77
18	4	40	48	92	18	111	26	165	27	262	4	414	26	474	52	499	75	539	62
21	28	41	34	93	18	112	1	182	52	283	26	415	3	475	77	502	26	540	62

masing metode akan dilakukan proses evaluasi dengan pengukuran *precision, dan recall*. Berdasarkan hasil evaluasi tersebut nantinya akan diketahui kinerja klasifikasi terbaik dari ketiga nilai tetangga terdekat (k).

Skenario pengujian dilakukan dengan cara melakukan perbandingan kinerja klasifikasi menggunakan metode FSKNN. Metode klasifikasi yang digunakan menggunakan parameter jumlah tetangga terdekat (k) 10, 20, dan 30. Pengujian ini dilakukan untuk mengetahui pengaruh penentuan nilai (k) terbaik. Hasil klasifikasi dari ketiga nilai (k) akan diukur menggunakan dua evaluasi metrik yaitu *precision, dan recall*. Pengukuran dilakukan menggunakan *ground truth pakar*.

Penentuan nilai (k) maksimal 30 berdasarkan hasil tetangga terdekat yang diperoleh dari 546 data uji. Tabel VII merupakan jumlah kluster yang terbentuk. Terdapat 150 kalimat kebutuhan yang memiliki jumlah kluster. Jumlah kluster tersebut diambil nilai rata-rata. Nilai rata-rata tersebut dijadkan dasar untuk menentukan nilai tetangga terdekat maksimal. Rata-rata nilai K pada 150 data uji tersebut adalah 30.

Tabel VIII adalah *confusion matrix* untuk mengukur seberapa baik metode yang digunakan untuk melakukan klasifikasi berdasarkan *ground truth* pakar. Tabel IX adalah hasil pengukuran *accuracy, precision*, dan *recall* masing-masing aspek kualitas. Sedangkan Tabel X adalah rata-rata hasil pengukuran *accuracy, precision*, dan *recall* untuk semua aspek kualitas.

TABEL VIII
CONFUSION MATRIX

Aspek Kualitas	**Ground Truth**	***FSKNN***	***PENGUJIAN METRIK***		
	A	***K10 B***	***TP C***	***FP D:B-C***	***FN E:A-C***	***TN F:(1366-(TP+FN)-FP)***
Functional Correctness(Fc)	16	15	1	14	15	1336
Time Behaviour(Tb)	22	6	0	6	22	1338
Reource Utilization(Ru)	20	6	0	6	20	1340
Capacity(Cp)	8	26	0	26	8	1332
Appropriateness Recognizability(Ar)	2	19	0	19	2	1345
Operability(O)	7	7	5	2	2	1357
User Error Protection(Uep)	1	1	0	1	1	1364
Accessibility(Ac)	1	27	1	26	0	1339
Maturity(M)	5	27	0	27	5	1334
Availability(Av)	3	54	3	54	0	1312
Fault Tolerance(Ft)	5	36	1	36	4	1326
Integrity(Itg)	6	2	0	2	6	1358
Non-Repudiation(Nr)	5	5	0	5	5	1359
Reusability(Reu)	4	30	1	29	3	1333
Testability(Test)	10	33	0	33	10	1323
Adaptability(Adp)	4	20	1	19	3	1343

TABEL IX
TABULASI RATA-RATA KINERJA PRECSION, DAN RECALL MENGGUNAKAN METODE FSKNN MASING-MASING ASPEK KUALITAS

Aspek Kualitas	***PENGUJIAN METRIK***		
	Accuracy	***Precision***	***Recall***
Fc	98%	6.67%	6.25%
Tb	98%	0.00%	0.00%
Ru	98%	0.00%	0.00%
Cp	98%	0.00%	0.00%
Ar	98%	0.00%	0.00%
O	100%	71.43%	71.43%
Uep	100%	0.00%	0.00%
Ac	98%	3.70%	100%
M	98%	0.00%	0.00%
Av	96%	5.56%	100%
Ft	97%	2.78%	20.00%
Itg	99%	0.00%	0.00%
Nr	99%	0.00%	0.00%
Reu	98%	3.33%	25.00%
Test	97%	0.00%	0.00%
Adp	98%	0.00%	0.00%

TABEL X
TABULASI RATA-RATA KINERJA PRECSION, DAN RECALL MENGGUNAKAN METODE FSKNN

Nilai K	**Ground Truth**	***FSKNN***		
		Accuracy	***Precision***	***Recall***
10	Pakar	**98.15%**	**6.15%**	**21.73%**
20	Pakar	97.84%	5.65%	14.23%
30	Pakar	96.89%	2.12%	19.86%

V. KESIMPULAN

Dari hasil uji coba yang dilakukan, dapat diambil kesimpulan yaitu *accuracy*, *precision*, dan *recall* terbaik menggunakan metode FSKNN diperoleh dengan nilai K 10 sebesar 98.15%, 6.15% dan 21.73%.

Klasifikasi kebutuhan non-fungsional ini menggunakan dataset dengan jumlah kelas tidak seimbang yaitu 1141 fungsional dan 225 non-fungsional. Ketidakseimbangan tersebut berdampak pada kinerja klasifikasi yang sangat rendah. Penelitian dapat dikembangkan dengan menambahkan metode untuk mengatasi ketidakseimbangan data.

DAFTAR PUSTAKA

[1] S. Ullah, M. Iqbal, and A. M. Khan, "A survey on issues in non-functional requirements elicitation," *Proc. - Int. Conf. Comput. Networks Inf. Technol.*, pp. 333–340, 2011.

[2] D. A. Ramadhani, S. Rochimah, and U. L. Yuhana, "Classification of Non-Functional Requirements Using Semantic-FSKNN Based ISO/IEC 9126," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 13, no. 4, p. 1456, 2015.

[3] I. Sommerville, *Software Engineering*. 2010.

[4] L. Chung and J. D. P. Leite, "On Non-Functional Requirements in Software Engineering," *Concept. Model. Found. …*, pp. 363–379, 2009.

[5] M. Rahimi, M. Mirakhorli, and J. Cleland-Huang, "Automated extraction and visualization of quality concerns from requirements specifications," *2014 IEEE 22nd Int. Requir. Eng. Conf. RE 2014 - Proc.*, pp. 253–262, 2014.

[6] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Inf. Softw. Technol.*, vol. 52, no. 4, pp. 436–445, 2010.

[7] A. Rashwan, O. Ormandjieva, and R. Witte, "Ontology-based classification of non-functional requirements in software specifications: A new corpus and SVM-based classifier," *Proc. - Int. Comput. Softw. Appl. Conf.*, no. ii, pp. 381–386, 2013.

[8] J. Cleland-Huang, R. Settimi, X. Zou, and P. Sole, "The detection and classification of non-functional requirements with application to early aspects," *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 36–45, 2006.

[9] B. Behkamal, M. Kahani, and M. K. Akbari, "Customizing ISO 9126 quality model for evaluation of B2B applications," *Inf. Softw. Technol.*, vol. 51, no. 3, pp. 599–609, 2009.

[10] W. Suharso and S. Rochimah, "Sistem Penambang." 2012.

[11] L. S. Jensen and T. Martinez, "Improving Text Classification by Using Conceptual and Contextual Features," 2000.

[12] H. Al-Kilidar, K. Cox, and B. Kitchenham, "The use and usefulness of the ISO/IEC 9126 quality standard," *2005 Int. Symp. Empir. Softw. Eng. ISESE 2005*, pp. 126–132, 2005.

[13] R. E. Al-Qutaish, "Quality models in software engineering literature: an analytical and comparative study," *J. Am. Sci.*, vol. 6, no. 3, pp. 166–175, 2010.

[14] F. Febrero, C. Calero, and M. Á. Moraga, "Software reliability modeling based on ISO/IEC SQuaRE," *Inf. Softw. Technol.*, vol. 70, pp. 18–29, 2016.

[15] K. Gwet, "Self-care behaviors of Italian patients with heart failure," *Eur. J. Cardiovasc. Nurs.*, vol. 12, no. 1, p. S11, 2013.