

OPTIMALISASI PROSES KLASIFIKASI DENGAN MENAMBAHKAN SEMANTIK PADA KEBUTUHAN NON-FUNGSIONAL BERBASIS ISO/IEC 25010

Lukman Hakim^{*1}, Choirul Huda²

^{1,2} Politeknik Negeri Jember, Jember
Email: ¹lukman.hakim@polije.ac.id, ²chuda@polije.ac.id

(Naskah masuk: 16 Februari 2023, diterima untuk diterbitkan: 03 Oktober 2023)

Abstrak

Kebutuhan non-fungsional (NFRs) memiliki peranan yang sangat penting untuk merancang sebuah sistem. Tetapi, mengidentifikasi NFRs masih menjadi tantangan nyata. NFRs dapat digunakan untuk memastikan apakah sistem sesuai dengan kebutuhan pengguna. Oleh karena itu, dibutuhkan suatu sistem untuk dapat mengidentifikasi NFRs. Penelitian sebelumnya menggunakan *dataset* dengan jumlah NFRs lebih banyak dibanding kebutuhan fungsional (FRs) memperoleh nilai tinggi. Akan tetapi, faktanya adalah NFRs lebih sedikit dari pada FRs. Penelitian selanjutnya menggunakan *dataset* dengan jumlah NFRs lebih sedikit dibandingkan (FRs) memperoleh nilai rendah terutama *precision* dan *recall*. Berdasarkan permasalahan pada penelitian sebelumnya, perlu dilakukan optimalisasi dengan cara menambahkan semantik pada metode FSKNN. Sistem yang digunakan untuk mengidentifikasi NFRs terdiri dari dua tahap. Pertama, sistem akan menghasilkan data latih secara otomatis tanpa pengukuran semantik. Setelah itu, data latih tersebut ditambahkan pengukuran semantik dan menghasilkan data latih baru. Kedua, proses klasifikasi menggunakan FSKNN. Pengujian dilakukan berdasarkan nilai ketetangaan tertinggi. Hasil dari pengujian menggunakan *dataset* dengan penambahan pengukuran semantik memperoleh nilai *accuracy*, *precision*, dan *recall* lebih baik yaitu sebesar 98,04%, 25,05%, dan 21,35. Data latih baru dengan penambahan semantik memperoleh hasil lebih tinggi dari data latih tanpa penambahan semantik. Nilai *precision* rendah karena *dataset* yang digunakan tidak seimbang.

Kata kunci: NFRs, SEMANTIK, FSKNN

OPTIMIZATION OF CLASSIFICATION PROCESS BY ADDING SEMANTIC OF NON-FUNCTIONAL REQUIREMENTS BASED ON ISO/IEC 25010

Abstract

Non-functional requirements (NFRs) are very important to design a system. However, identifying NFRs remains a real challenge. NFRs can be used to determine whether the system meets user requirements. Therefore, we need a system to be able to identify NFRs. Previous studies using a dataset with a higher number of NFRs than functional requirements (FRs) obtained high scores. However, the fact is that there are fewer NFRs than FRs. Subsequent studies using datasets with fewer NFRs than FRs obtain low values, especially precision, and recall. Based on the problems in previous studies, it is necessary to optimize by adding semantics to the FSKNN method. The system used to identify NFRs consists of two stages. First, the system will generate training data automatically without semantic measurements. After that, the training data is added with semantic measurements and generates new training data. Second, the classification process uses FSKNN. Testing is done based on the highest neighbor value. The results of testing using a dataset with the addition of semantic measurements obtained better accuracy, precision, and recall values of 98,04%, 25,05%, and 21,35. New training data with semantic additions obtains higher results than training data without semantic additions. The precision value is low because the dataset used is unbalanced.

Keywords: NFRs, HSO, FSKNN

1. PENDAHULUAN

Kebutuhan non-fungsional (NFRs) dan aspek kualitas perangkat lunak merupakan dua hal yang tidak bisa dipisahkan. Aspek kualitas NFRs memiliki peranan yang sangat penting untuk

merancang sebuah sistem. Tetapi, mengidentifikasi aspek kualitas NFRs masih menjadi tantangan nyata (Sotelo & Esteban, 2018). Pada proses pengembangan perangkat lunak, NFRs tidak terlalu diperhatikan karena dianggap tidak terlalu berguna.

Padahal, NFRs bisa digunakan untuk menentukan prioritas kebutuhan perangkat lunak, khususnya kebutuhan fungsional (FRs). Penentuan proses tersebut dapat mengurangi konflik yang muncul di antara pemangku kepentingan (Raharjo, Kurniawan & Rusdianto, 2019).

NFRs adalah permasalahan yang ditawarkan oleh sistem berupa fungsi layanan (Melinda, Borman & Susanto, 2018). Penelitian untuk mengidentifikasi NFRs lebih banyak dilakukan dibandingkan untuk mengidentifikasi FRs. Hal tersebut dikarenakan NFRs sulit diidentifikasi karena jumlahnya sedikit dibandingkan FRs. Disamping itu, FRs memiliki *term-term* yang menyerupai *term-term* pada NFRs sehingga sudah dibedakan (Aziiza & Fadhilah, 2020). Kualitas perangkat lunak akan lebih baik diketahui di awal fase pengembangan, karena dapat meminimalisir biaya. Salah satu caranya adalah menganalisa aspek kualitas NFRs.

Berbagai cara telah dilakukan untuk menyempurnakan sistem identifikasi aspek kualitas NFRs. Pertama, melakukan klasifikasi NFRs untuk meningkatkan kualitas produk perangkat lunak dan mengurangi upaya para analis dalam mengidentifikasi kalimat kebutuhan secara manual dari dokumen (SRS), dengan bantuan *Thematic Role* menggunakan hubungan linguistik (Singh, Singh & Sharma, 2016). *Dataset* yang digunakan pada penelitian tersebut 635 kalimat kebutuhan dengan 265 FRs dan 370 NFRs. Hasil sangat bagus karena kalimat kebutuhan hanya sedikit dan NFRs lebih banyak. Disamping itu, standard ISO yang digunakan menggunakan ISO/IEC 9126 dimana ISO tersebut merupakan standart lama.

Kedua, klasifikasi dengan mengkombinasikan hipernim dan sinonim dengan membandingkan dua metode klasifikasi yaitu KNN dan SVM. Hasilnya menggunakan 1 level hipernim & 20 sinonim memperoleh nilai *accuracy*, *precision*, dan *recall* lebih baik dibandingkan 2 level hipernim & 9 sinonim pada kedua metode klasifikasi. Hal tersebut karena term akan kehilangan makna asli apabila menggunakan 2 level (Hakim, Rochimah & Faticah, 2019). Ketiga, melakukan klasifikasi NFRs menggunakan metode FSKNN. Ketiga nilai evaluasi matriks yang digunakan memperoleh nilai tertinggi menggunakan metode FSKNN dengan nilai $k = 10$ (Hakim, Rochimah & Faticah, 2019). Penelitian kedua dan ketiga menggunakan *dataset* sama yaitu 1366 kalimat kebutuhan dengan 1141 FRs dan 225 NFRs. Nilai yang diperoleh pada kedua penelitian tersebut rendah terutama pada nilai *precision* dan *recall*. Hal tersebut dikarenakan NFRs lebih sedikit dibandingkan FRs.

Penelitian ini mengusulkan teknik klasifikasi menggunakan metode FSKNN dengan penambahan semantik menggunakan standard ISO/IEC 25010. Klasifikasi yang dilakukan menggunakan *dataset* yang sama dengan penelitian kedua dan ketiga. Oleh

karena itu, hasil dari penelitian tersebut dapat dijadikan parameter, apakah nilai *accuracy*, *precision*, dan *recall* lebih tinggi atau lebih rendah dari penelitian sebelumnya. Pengukuran semantik ditambahkan pada data latih yang diperoleh dari fase sebelumnya yaitu otomatisasi data pelatihan. Metode yang digunakan untuk mengukur semantik adalah HSO (Ramadhani, Rochimah & Yuhana, 2015).

Pada fase otomatisasi data pelatihan ditambahkan pengukuran semantik. Proses tersebut akan menghasilkan data latih baru. Kedua data latih tersebut akan diuji menggunakan metode FSKNN. Berdasarkan data uji coba maka metode klasifikasi tersebut bertujuan untuk mengukur performa dari data latih yang diperoleh.

2. METODE PENELITIAN

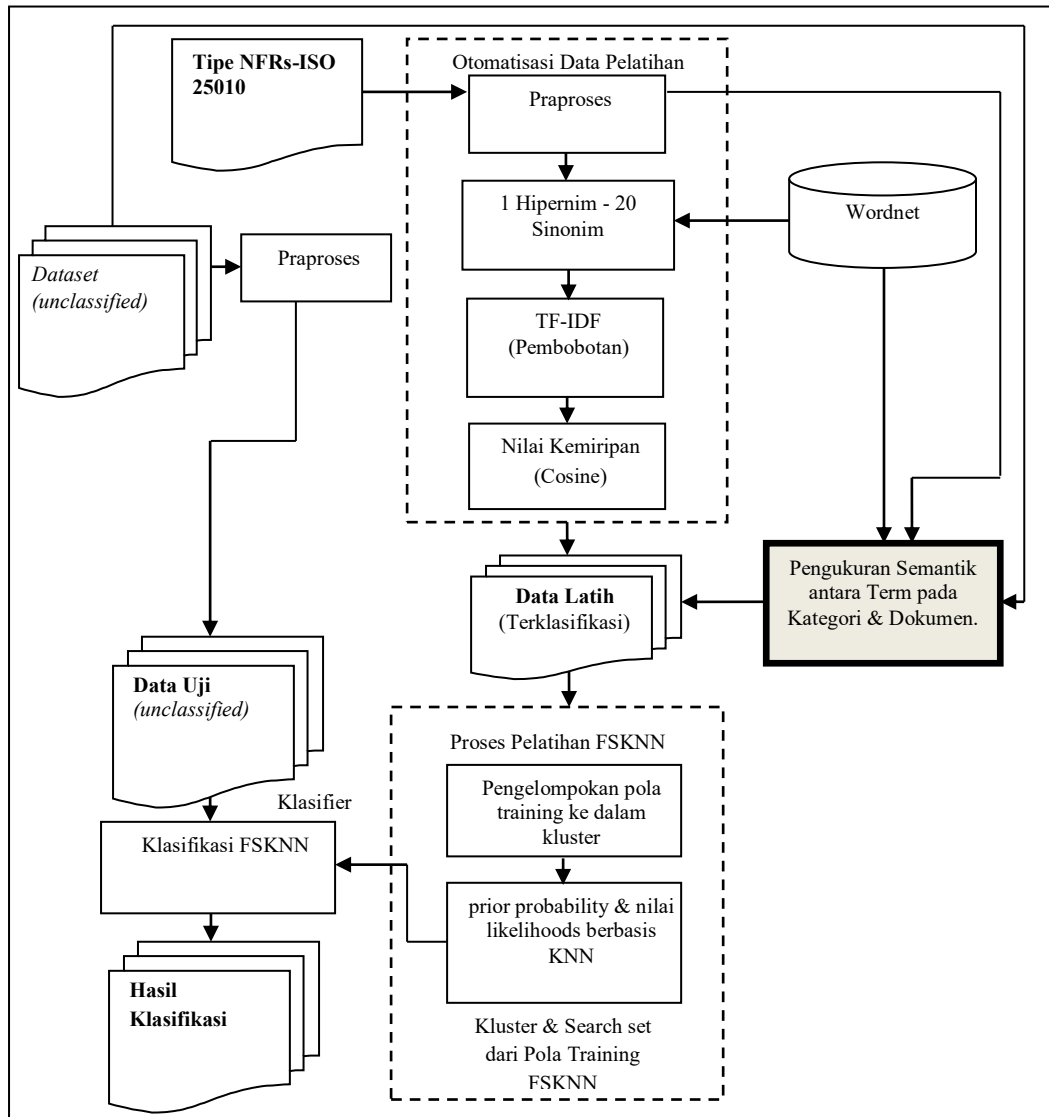
Mekanisme identifikasi NFRs yang diusulkan ditunjukkan pada gambar 1. Pada mekanisme yang diusulkan terdapat dua proses utama yaitu otomatisasi data pelatihan dan proses klasifikasi. Kontribusi utama pada penelitian ini terdapat pada penambahan pengukuran semantik pada data latih yang diperoleh dari proses otomatisasi data pelatihan. Pengukuran semantik tersebut akan menghasilkan data latih baru. Hasil klasifikasi menggunakan FSKNN dari kedua data latih tersebut akan dibandingkan untuk menguji pengaruh dari penambahan pengukuran semantik tersebut.

2.1. Data

Berdasarkan hasil penelitian sebelumnya data yang digunakan pada penelitian ini terdiri dari 1141 FRs dan 225 NFRs (Hakim, Rochimah & Faticah, 2019). Kalimat kebutuhan tersebut sejumlah 1300 diperoleh dari *dataset VoIP* dan 66 diperoleh dari *geolocation*. Jadi total data yang digunakan pada penelitian ini sejumlah 1366.

2.2. Praproses

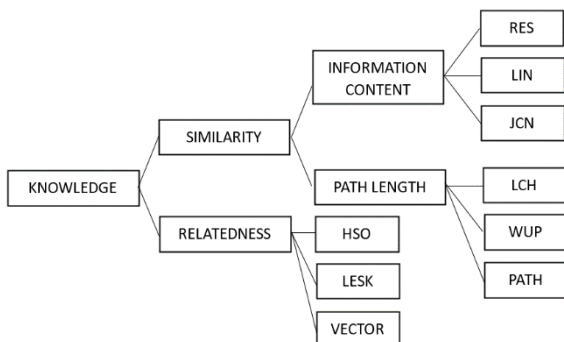
Dataset yang digunakan akan melalui tahapan praproses terlebih dahulu sebelum masuk ke tahap selanjutnya. Praproses memiliki empat tahap, diantaranya : 1. Tokenisasi yaitu mengubah kalimat menjadi potongan kata, 2. Case folding yaitu merubah huruf menjadi huruf kecil. Contoh: “*The data should be secure*” akan ditokenisasi menjadi “*the*”, “*data*”, “*should*”, “*be*”, dan “*secure*”. 3. Stopwords removal yaitu menghilangkan simbol, tanda baca, bilangan dan kata yang memiliki prosentase rendah untuk menjadi indikator. Contoh: “*a*”, “*an*”, “*and*”, “*as*”, “*be*”, dan sebagainya. 4. Stemming yaitu merubah teks menjadi kata dasar (*stemming*). Contoh: “*secure*” menjadi “*secur*”, “*placed*” menjadi “*place*”, “*granted*” menjadi “*grant*” dan “*available*” menjadi “*avail*”.



Gambar 1. Model deteksi dan klasifikasi NFRs

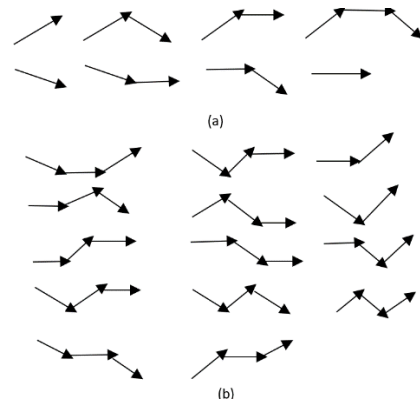
2.3. Pengukuran Semantik

Pengukuran semantik yang digunakan pada penelitian ini adalah *Knowledge-based*. Pengukuran semantik tersebut digunakan karena berhubungan dengan pemanfaatan informasi yang terdapat dalam wordnet, detail pembagian kelompok dapat dilihat pada gambar 2 (Ramadhani, Rochimah & Yuhana, 2015).



Gambar 2. Knowledge-based Similarity Measures

Relasi Kohesi digunakan oleh metode HSO untuk menghitung jalur yang diizinkan antara dua kata. HSO sendiri memiliki tipe relasi yang bervariasi, diantaranya *Medium Strong Relation*, *Strong Relation*, dan *Extra Strong Relation*. Detail pembagian jalur dapat dilihat pada gambar 3.

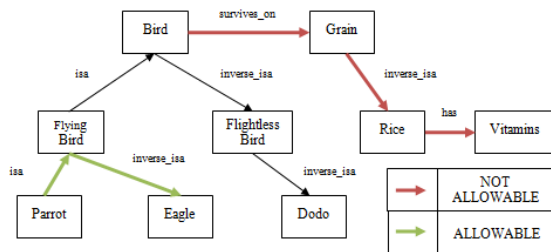


Gambar 3. Pola Jalur Diizinkan (a) dan Pola Jalur yang Tidak Diizinkan (b)

Hubungan relasi *strong relation* antara satu dengan yang lainnya dapat dilihat pada gambar 4. Contohnya kata *parrot* dan *eagle*, kedua kata tersebut terhubung dengan salah satu jalur yang diizinkan. Sedangkan kata *vitamins* dan *bird* secara semantik tidak memiliki hubungan antara satu dengan yang lainnya dan kedua kata tersebut juga terhubung dengan jalur yang tidak diizinkan. Sehingga kedua kata tersebut dapat dikatakan sangat berbeda. Sedangkan untuk memastikan suatu jalur sudah sesuai, HSO memiliki dua kondisi yaitu jika ada kata diturunkan menggunakan *downward link*, maka kata tersebut tidak boleh digeneralisasi menggunakan *upward link*.

Pengukuran keterkaitan semantik antara dua kata pada *medium strong relation* dengan menghitung nilai bobot. Perhitungan nilai bobot bisa dilihat pada persamaan 1.

$$weight = C - path\ length - k * number\ of\ changes\ of\ direction \quad (1)$$



Gambar 4. Contoh Relasi Dengan Jalur yang Diizinkan dan Tidak Diizinkan

2.4. Wordnet

Wordnet digunakan untuk melakukan pengembangan *term* dari data latih dengan dari relasi hipernim (Hakim, Rochimah & Fatichah, 2019a).

2.5. Pembobotan TF-IDF

Untuk menghitung jumlah kemunculan (*f*) term *t_i* dalam setiap kalimat kebutuhan *d_j* menggunakan persamaan 2.

$$TF = (t_i, D) \quad (2)$$

Sedangkan untuk menghitung bobot term antar kalimat kebutuhan menggunakan persamaan 3. Term yang memiliki nilai tinggi jika term tersebut jarang muncul.

$$IDF = \log \frac{N}{df(t_i, D)} + 1 \quad (3)$$

Proses selanjutnya adalah melakukan normalisasi agar nilai bobot yang didapatkan berada dalam rentang nilai 0 hingga 1. Untuk proses perhitungan *tf*idf* bisa dilihat pada persamaan 4.

$$\omega_K(t_i) = \frac{tf(t_i, K) \times \log \frac{N}{df(t_i, K)} + 1}{\sqrt{\sum t_i (tf(t_i, K) \times \log \frac{N}{df(t_i, K)} + 1)^2}} \quad (4)$$

2.6. Cosine Similarity

Untuk mengukur nilai kemiripan antara kalimat kebutuhan dengan definisi aspek kualitas menggunakan ISO/IEC 25010 menggunakan Cosine Similarity. Query dan Dokumen dianggap sangat mirip jika bernilai 1, dan tidak mirip tidak bernilai 0. Cosine Similarity bisa dilihat pada persamaan 5.

$$Cos(Q, K) = \sum_{r=1}^M \omega_Q(t_i) \times \omega_K(t_i) \quad (5)$$

2.7. Fuzzy Similarity based K-Nearest Neighbor (FSKNN)

FSKNN merupakan metode klasifikasi yang dapat digunakan untuk melakukan pengelompokan kalimat kebutuhan multilabel. FSKNN merupakan kombinasi antara fuzzy similarity measure dan k-nearest neighborhood (Lubis, Sari & Fauzi, 2019).

Proses pelatihan FSKNN terdiri dari 2 tahap yaitu pengelompokan pola training ke dalam kluaster dan perhitungan prior probability dan nilai likelihoods berbasis KNN. Selanjutnya proses klasifikasi dapat dilakukan menggunakan Maximum a posteriori (MAP) (Hakim, Rochimah & Fatichah, 2019b).

2.8. Evaluasi

Penelitian ini menggunakan tiga evaluasi matrik diantaranya *accuracy*, *precision*, dan *recall*. Proses perhitungan ketiganya ditentukan dari prediksi informasi pada nilai aktual yang diwakili oleh TP, FP, FN, dan TN. Persamaan dari ketiga metode evaluasi ditunjukkan dalam persamaan 6, 7, dan 8 (Zakia, 2020).

Sistem dikatakan efektif apabila rasio *precision* dan *recall* sama besarnya. Sistem juga dikatakan efektif apabila nilai *precision* lebih tinggi dari pada *recall*. Hal tersebut dikarenakan semakin tinggi nilai *precision* itu berarti sistem mampu menemukan banyak kalimat kebutuhan relevan dari pada kalimat kebutuhan tidak relevan. (Nilawati *et al.*, 2019).

Accuracy digunakan untuk mengukur kedekatan antara nilai prediksi dan nilai aktual. Contoh :

Aspek Kualitas *time behaviour* sejumlah 9 kalimat kebutuhan dicampurkan dengan aspek kualitas lain sejumlah 1357 kalimat kebutuhan. Sistem dapat mendeteksi 25 kalimat kebutuhan sebagai aspek kualitas *time behaviour*. Setelah dilakukan pengecekan kembali, ternyata dari 25 kalimat kebutuhan tersebut hanya 5 yang merupakan aspek kualitas *time behaviour* (*Correct Result*) sedangkan 4 kalimat kebutuhan *time behaviour* yang tidak terdeteksi disebut hasil yang benar tetapi tidak terdeteksi (*Missing Result*). Selanjutnya, hasil yang tidak diharapkan sejumlah 20 (*Unexpected Result*). Terakhir, kalimat kebutuhan yang bukan termasuk *time behaviour* dan tidak terdeteksi oleh sistem

sejumlah 1337 (Correct absence of result). Hasilnya ditunjukkan pada tabel 1.

Precision digunakan untuk mengukur kecocokan antara dokumen yang diminta dengan dokumen yang ditawarkan oleh sistem. Ketika sistem menawarkan beberapa dokumen, maka precision bisa menjadi alat ukur yang tepat untuk mencari dokumen yang paling cocok. Sedangkan recall proporsi jumlah dokumen yang ditemukan kembali oleh sistem.

Tabel 1 Contoh Evaluasi Matriks

Nilai prediksi	Nilai Sebenarnya		
		TRUE	FALSE
TRUE	(TP) 5	(FP) 20	
FALSE	(FN) 4	(TN)1337	

Berdasarkan pada tabel 1 diperoleh nilai *precision* sebesar 0,20, *recall* 0,56, dan *accuracy* sebesar 0,98. Ketiga nilai ini akan semakin tinggi apabila jumlah TP semakin tinggi khusus *precision*.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (6)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (8)$$

3. HASIL DAN PEMBAHASAN

Pengujian pada penelitian ini dibagi menggunakan dua data latih berbeda. Data latih pertama, diperoleh dari *dataset* yang sudah melalui proses otomatisasi data pelatihan. Data latih kedua, diperoleh dari data latih pertama ditambahkan pengukuran semantik dan menghasilkan data latih baru. Data latih tersebut sebelumnya sudah diuji dan dievaluasi pada fase otomatisasi data pelatihan dari hasil *cosine similitaty*. Mekanisme pengujian dan evaluasi tersebut menggunakan rentang *threshold* 0,1 hingga 0,7. Nilai *threshold* yang memiliki nilai seimbang pada ketiga metode evaluasi akan digunakan pada proses klasifikasi.

3.1. Proses Klasifikasi

Pengujian dilakukan menggunakan dua *dataset* berbeda sejumlah 1366 kalimat kebutuhan. *Dataset* tersebut akan dipecah menjadi dua bagian yaitu 820 digunakan sebagai data latih dan 546 sebagai data uji. Setelah melalui tahapan otomatisasi data pelatihan maka akan menghasilkan data latih pertama. Data latih tersebut ditambahkan pengukuran semantik menggunakan metode HSO. Proses penambahan semantik tersebut akan menghasilkan data latih kedua. Selanjutnya, metode FSKNN digunakan untuk membandingkan kinerja klasifikasi berdasarkan kedua data latih tersebut. Proses klasifikasi menggunakan metode FSKNN menggunakan Metode klasifikasi yang digunakan

menggunakan *threshold* 0.5 dan parameter jumlah tetangga terdekat (*k*) 10, 20, dan 30.

Pada penelitian ini menggunakan tetangga terdekat (*k*) 10 karena pada penelitian Zakia dkk dapat memperoleh rata-rata tertinggi untuk nilai *precision* sebesar 70,3 (Zakia, 2020). Selanjutnya, sebagai bentuk perbandingan penelitian ini juga menggunakan tetangga terdekat (*k*) 20 dan 30.

Hasil yang ditunjukkan pada tabel 2 dan tabel 3 diambil dari nilai tertinggi dari jumlah tetangga terdekat (*k*). Pengujian ini dilakukan untuk mengetahui pengaruh penambahan semantik pada data latih pertama. Hasil klasifikasi dari kedua data latih tersebut akan diukur menggunakan tiga evaluasi metrik yaitu *accuracy*, *precision*, dan *recall*. Tabel 2 menggunakan data latih yang sama dengan penelitian sebelumnya sedangkan tabel 3 menggunakan data latih dengan penambahan semantik sehingga menjadi data latih baru. Penambahan semantik pada data latih tersebut merupakan kontribusi utama pada penelitian ini.

3.2. Hasil Pengujian

Pada subbab ini akan membahas hasil pengujian dari setiap kedua data latih tanpa penambahan semantik dan data latih dengan penambahan semantik yang telah dibahas sebelumnya.

a. Hasil Pengujian Data Latih Pertama

Nilai evaluasi pengujian yang akan dilakukan menggunakan *threshold* 0,5. Nilai *threshold* tersebut dipilih berdasarkan evaluasi kritis dari penelitian sebelumnya bahwa yang dilakukan oleh hakim dkk (2019) (Hakim, Rochimah & Faticah, 2019) dengan menggunakan *dataset* yang sama. Evaluasi hasil pengujian pada data latih pertama ditunjukkan pada tabel 2.

Tabel 2 Evaluasi matriks menggunakan metode FSKNN pada data latih 1

K	Threshold	Data Latih 1 : FSKNN		
		Accuracy	Precision	Recall
10	Unsupervised	98,00%	22,55%	22,65%
20	Unsupervised	97,84%	15,10%	25,90%
30	Unsupervised	96,89%	7,55%	20,77%

b. Hasil Pengujian Data Latih Kedua

Hasil evaluasi pada data latih dengan penambahan pengukuran semantik ditunjukkan pada tabel 3.

Tabel 3 Evaluasi matriks menggunakan metode FSKNN pada data latih 2

K	Threshold	Data Latih 2 : FSKNN		
		Accuracy	Precision	Recall
10	Unsupervised	96,33%	13,42%	17,09%
20	Unsupervised	94,55%	4,18%	18,15%
30	Unsupervised	98,04%	25,05%	21,35%

