Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 2, No. 2, Februari 2018, hlm. 486-492 http://j-ptiik.ub.ac.id

Klasifikasi Dokumen Tumbuhan Obat Menggunakan Metode *Improved k-Nearest Neighbor*

Arinda Ayu Puspitasari[1], Edy Santoso[2] , Indriati[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]arindayu91@gmail.com, [2]edy144@ub.ac.id, [3]indriati.tif@ub.ac.id

Abstrak

Tingkat pemanfaatan tumbuhan obat yang semakin tinggi memicu banyaknya penelitian terhadap tumbuhan obat. Penelitian-penelitian tersebut tentu memerlukan dokumentasi yang berisi informasi tentang tumbuhan obat. Dokumentasi yang banyak dan tersebar menyebabkan kesulitan dalam pencarian informasi seputar tumbuhan obat. Untuk mengatasi permasalahan tersebut dibutuhkan sebuah sistem yang dapat mengklasifikasikan dokumen tumbuhan obat secara otomatis agar pencarian informasi terkait dapat dilakukan dengan lebih efektif dan efisien. Salah satu metode yang sering digunakan dalam klasifikasi teks adalah *k-Nearest Neighbor,* tetapi memiliki kelemahan pada akurasi karena penggunaan nilai *k* yang sama pada semua kategori. Nilai *k* adalah jumlah data latih terdekat terhadap data uji. Untuk mengatasi permasalahan tersebut digunakan metode *Improved k-Nearest Neighbour* di mana nilai *k* akan disesuaikan dengan jumlah data latih yang dimiliki setiap kategori. Dari hasil pengujian pengaruh pertambahan nilai *k* diperoleh rata-rata *F1-measure* sebesar 70,99%. Pengujian variasi data latih menunjukkan bahwa semakin besar jumlah data latih maka semakin tinggi nilai rata-rata akurasinya, sedangkan untuk pengujian data latih tidak seimbang diperoleh nilai *F1-measure* data latih seimbang 1,9% lebih baik dari data latih tidak seimbang.

Kata kunci: *k-Nearest Neighbor, Improved k-Nearest Neighbor, text mining, tanaman obat, klasifikasi dokumen*

Abstract

The high utilization rates of medicinal plants is leading to increase the studies on it. Those studies certainly require documentation that contains information about medicinal plants. The large and scattered documentation cause difficulties in searching for information about medicinal plants. To overcome these problems a system that can classify the document automatically is needed to make the information search work more effective and efficient. K-Nearest Neighbor is the algorithm often used to classify text, but has a weakness in accuracy because of the fixed k values for each category. K values is the amount of the closest training data to the test data. Improved k-Nearest Neighbour is the algorithm used in this study to overcome the problem where the different k values will be applied based on the amount of the training data for each category. The average accuracy for the k values testing is 70,99%. The training data variation testing shows that the bigger amount of training data the higher average accuracy will be. The unbalanced data testing showed that the balance data training category has 1,9% better accuracy than the unbalanced category.

Keywords: *k-Nearest Neighbor, Improved k-Nearest Neighbor, text mining, medicinal plants, document classification*

1. PENDAHULUAN

Tumbuhan obat merupakan tumbuhan yang memiliki senyawa bermanfaat untuk mencegah maupun mengobati suatu penyakit. Tumbuhan obat telah banyak digunakan sebagai bahan baku pembuatan obat kimiawi maupun herbal. Namun, kesadaran terhadap bahaya bahan-bahan kimiawi yang terkandung dalam obat-obatan modern semakin membuka mata akan betapa penting dan bernilainya obat-obatan tradisional yang telah disediakan secara berlimpah oleh alam Indonesia. Berdasarkan survei yang dilakukan Kementerian Kesehatan terhadap 20 persen dari 1.168 suku etnis, diperoleh 1.500 formula yang diramu dari 24.927 jenis tumbuhan obat di Indonesia.

Tingkat pemanfaatan yang tinggi memicu

banyaknya penelitian tentang tumbuhan obat. Dokumentasi yang berisi informasi tentang tumbuhan obat tentu menjadi kebutuhan dasar dalam penelitian tersebut. Dokumentasi terkait tumbuhan obat telah banyak tersebar ke dalam bentuk media cetak dan elektronik. Namun, jumlah dokumen yang banyak dan tersebar menyebabkan kesulitan dalam pencarian informasi tentang tumbuhan obat. Hal tersebut tentu dirasa kurang memberi manfaat bagi para peneliti tumbuhan obat. Untuk itu, diperlukan sistem yang dapat mengelompokkan informasi dalam jumlah banyak agar dapat diperoleh informasi tentang tumbuhan obat yang sesuai secara lebih mudah dan terorganisir.

Salah satu penelitian terkait dengan klasifikasi dokumen tumbuhan obat dilakukan oleh Kristiana Paskianti dengan judul Klasifikasi Dokumen Tumbuhan Obat Menggunakan Algoritma KNN-Fuzzy. Metode Fuzzy dilakukan dengan memberi derajat keanggotaan pada setiap kelas guna mengatasi data yang tidak seimbang serta karakteristik dokumen yang cenderung seragam. Dalam penelitian ini terbukti bahwa metode KNN-Fuzzy dapat mengklasifikasi data yang tidak seimbang (Paskianti, 2011).

k-Nearest Neighbor dikenal sebagai metode klasifikasi yang sederhana dan mudah diimplementasikan. Metode ini melakukan klasifikasi terhadap suatu objek berdasarkan jarak terdekatnya terhadap data latih. Metode *k-Nearest Neighbor* dinilai cukup efektif dalam proses pengelompokan namun memiliki kelemahan pada akurasi (Tan, 2006). Kelemahan akurasi pada metode *k-Nearest Neighbor* disebabkan karena penggunaan nilai *k* yang sama pada semua kategori tanpa mempertimbangkan jumlah data latih yang belum tentu sama pada setiap kategori.

Kelemahan pada *k-Nearest Neighbor* dapat diatasi dengan menggunakan metode *Improved k-Nearest Neighbor* yang diperkenalkan oleh Baoli, Shiwen, dan Qin pada penelitian berjudul *An Improved k-Nearest Neigbors for Text Categorization* (Baoli, 2003). Penelitian ini dilakukan terhadap dokumen berbahasa Cina. Metode *Improve k-Nearest Neighbor* melakukan modifikasi pada penetapan nilai *k* yang digunakan. Nilai *k* yang digunakan akan disesuaikan dengan jumlah data latih yang dimiliki oleh masing-masing kategori. Dalam penelitian yang dilakukan oleh Baoli, Shiwen, dan Qin, metode *Improved k-Nearest Neighbor* terbukti dapat menunjukkan kestabilan dalam akurasi. Berdasarkan latar belakang di atas maka penulis mengangkat topik "**Klasifikasi Dokumen Tumbuhan Obat Menggunakan Metode *Improved k-Nearest Neighbor*".**

2. TINJAUAN PUSTAKA

2.1. *Preprocessing*

Text preprocessing merupakan proses awal dalam *text mining*. *Text preprocessing* bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan pada proses selanjutnya.

a. *Case Folding*

Case folding merupakan tahap mengubah semua karakter huruf pada dokumen menjadi huruf kecil. Huruf yang diterima adalah huruf "a" sampai dengan "z". Karakter selain huruf tersebut seperti emotikon dan simbol-simbol akan dihilangkan dan dianggap sebagai *delimiter*.

b. *Tokenizing*

Tokenizing merupakan tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Dalam tahap ini spasi digunakan sebagai pemisah antar kata.

c. *Filtering*

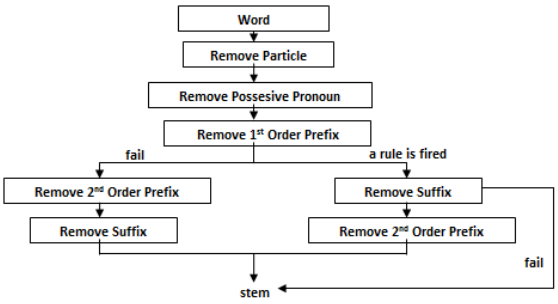
Filtering merupakan tahap mengambil kata-kata penting dari hasil *tokenizing* yang digunakan untuk mewakili isi dari suatu dokumen dan membedakannya dari dokumen lain dalam koleksi. Proses yang dilakukan adalah menghapus *stopword* yaitu kata-kata yang dianggap tidak bermakna atau tidak layak untuk dijadikan sebagai pembeda atau sebagai kata kunci dalam klasifikasi dokumen, contohnya adalah kata penghubung.

d. *Stemming*

Stemming merupakan tahap mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan atau mengubah kata kerja menjadi kata benda. *Stemming* pada teks Bahasa Indonesia berbeda dengan Bahasa Inggris. Untuk teks Bahasa Inggris yang perlu dihilangkan hanya akhiran (*suffixes)*, sedangkan dalam teks Bahasa Indonesia jenis imbuhan yang dapat dihilangkan selain akhiran (*suffixes)* adalah awalan (*prefixes*), sisipan (*infixes*), dan kombinasi awalan dan akhiran (*confixes*).

Tahap *stemming* dalam penelitian ini menggunakan algoritma *Porter Stemmer* untuk

Bahasa Indonesia yang dikembangkan pertama kali oleh Fadillah Z. Tala pada tahun 2003. Algoritma *Porter Stemmer* untuk Bahasa Indonesia merupakan modifikasi dari algoritma *Porter Stemmer* Bahasa Inggris dikarenakan struktur Bahasa Indonesia yang berbeda dengan Bahasa Inggris. Gambar 1 menunjukkan proses *stemming* menggunakan *Porter Stemmer* untuk Bahasa Indonesia (Tala, 2003).



Gambar 1. Algoritma *Porter Stemmer* Untuk Bahasa Indonesia

2.2. Pembobotan

Pembobotan atau *term weighting* merupakan proses mendapatkan nilai dari *term* yang berhasil diekstrak dari proses sebelumnya. Metode yang digunakan dalam proses pembobotan ini adalah *Term Frequency-Inverse Document* (TF-IDF).

Term Frequency (TF) adalah jumlah kemunculan sebuah kata pada suatu dokumen. Jika sebuah kata sering muncul dan sama dengan *term* dalam sebuah dokumen, maka nilai TF *term* akan bertambah. Semakin tinggi nilai frekuensi kemunculan kata dalam suatu dokumen maka semakin besar pengaruh *term* pada dokumen tersebut.

Inverse Document Frequency (IDF) merupakan jumlah dokumen yang mengandung sebuah *term* yang dicari dari kumpulan dokumentasi yang ada. IDF dihitung dengan Persamaan (1) (Yiming, 1999).

$$w(i,j) = tf(i,j) * idf \quad (1)$$

Dimana:
w(i,j) : bobot term ke-*i* dalam dokumen *j*
tf(t,d) : frekuensi kata *i* pada dokumen *j*
idf : nilai *idf* term ke-*i*

2.3. *Cosine Similarity*

Metode *cosine similarity* merupakan salah satu model ruang vektor yang tujuan utamanya adalah untuk mencari kemiripan antara vektor dokumen dengan vektor *query*. Semakin sama suatu vektor dokumen dengan vektor *query* maka dokumen dapat dipandang semakin sesuai dengan *query*. Persamaan (2) digunakan untuk menghitung *cosine similarity* (Yong, 2009).

$$\text{CosSim}(q, d_j) = \frac{\sum_{i=1}^{m} w_{i,q} . w_{i,j}}{\sqrt{\sum_{i=1}^{m} w_{i,q}^2} . \sqrt{\sum_{i=1}^{m} w_{i,j}^2}} \quad (2)$$

Dimana:
q : dokumen uji
d_j : dokumen *training* ke-*j*
$w_{i,q}$: bobot term *i* pada dokumen uji *q*
$w_{i,j}$: bobot term *i* pada dokumen latih *j*
m : batas atas jumlah term
i=1 : batas bawah jumlah term

2.4. *Improved k-Nearest Neighbor*

Hasil klasifikasi sangat dipengaruhi oleh nilai *k* yang dipilih. Penentuan nilai *k* yang tepat diperlukan agar diperoleh akurasi yang tinggi dalam proses kategorisasi dokumen uji. Pada *k-Nearest Neighbor* nilai *k* yang digunakan untuk tiap kategori sama, hal ini kurang efektif karena setiap kategori dapat memiliki jumlah data latih yang berbeda. Untuk mengatasi masalah tersebut digunakan metode *Improved k-Nearest Neighbor* di mana pada metode ini digunakan nilai *k* yang berbeda untuk setiap kategori disesuaikan dengan jumlah data latih yang dimiliki (Baoli, 2003).

Proses klasifikasi diawali dengan mengurutkan nilai *cosine similarity* dari terbesar ke terkecil. Selanjutnya dilakukan proses penghitungan untuk mendapatkan nilai *k* baru dengan Persamaan (3) (Baoli, 2003).

$$n = \left[\frac{k * N(c_m)}{Maks\{N(c_m)|j=1...Nc\}}\right] \quad (3)$$

Dimana:
n : nilai *k* baru
k : nilai *k* yang ditetapkan
N(Cm) : jumlah dokumen latih kategori m
Maks{N(Cm)|j=1...Nc} : jumlah dokumen latih terbanyak pada semua kategori

Sejumlah *n* dokumen yang dipilih adalah top *n* dokumen yang memiliki kemiripan paling besar di setiap kategori. Proses selanjutnya adalah menghitung peluang dokumen uji X termasuk ke dalam kategori m dengan Persamaan (4).

$$p(x, c_m) = argMaks_m \frac{\sum d_j \in top\ n\ k\ NN\ (C_m) sim(x, d_j) y(d_j . c_m)}{\sum d_j \in top\ n\ k\ NN\ (C_m) sim(x, d_j)} \quad (4)$$

Dimana:
$p(X,C_m)$: probabilitas dokumen X anggota C_m
$sim(x,d_j)$: kemiripan antara dokumen X dengan dokumen latih d_j
top n kNN : top n tetangga
$y(d_j.C_m)$: fungsi atribut yang memenuhi dari sebuah kategori, akan bernilai 1 apabila dokumen latih d_j masuk dalam anggota C_m jika tidak maka bernilai 0.

Dokumen uji *X* akan diklasifikasikan ke dalam kategori yang nilai probabilitasnya paling besar (Baoli, 2003).

2.5. *Confusion Matrix*

Evaluasi dilakukan untuk mengetahui kinerja dan akurasi dari metode klasifikasi yang telah diterapkan. *Confusion matrix* merupakan tabel yang terdiri dari banyaknya baris data uji yang diprediksi benar dan salah oleh model klasifikasi yang kemudian digunakan untuk menentukan kinerja dan akurasi dari model klasifikasi seperti ditunjukkan pada Tabel 1 (David, 2011).

Tabel 1. *Confusion Matrix*

Kategori x	***Predicted***	
Actual	*True positive*	*False positive*
	False negative	*True negative*

TP (*True Positive)* menunjukkan jumlah data uji yang diklasifikasikan sistem ke dalam kategori *x*, dan semua data tersebut memang benar termasuk kategori *x*.

FP *(False Positive)* menunjukkan jumlah data uji yang tidak diklasifikasikan sistem ke dalam kategori *x*, tetapi seharusnya semua data tersebut termasuk kategori *x*.

FN *(False Negative)* menunjukkan jumlah data uji yang diklasifikasikan sistem ke dalam kategori *x*, tetapi seharusnya semua data tersebut bukan termasuk kategori *x*.TN *(True Negative)* menunjukkan jumlah data uji yang tidak diklasifikasikan sistem ke dalam ketegori *x*, dan semua data tersebut memang bukan termasuk kategori *x*.

2.6. *Precision, Recall,* dan *F1-measure*

Dalam penelitian kasus klasifikasi, evaluasi yang sering digunakan untuk mengukur akurasi sistem adalah dengan menghitung parameter *precision*, *recall*, dan *F1-measure* dari hasil ringkasan aplikasi.

Precision adalah keakuratan hasil klasifikasi dari seluruh dokumen oleh sistem, sehingga dapat diketahui apakah kategori data yang diklasifikasi sesuai dengan kategori sebenarnya. *Precision* dihitung dari jumlah pengenalan data yang bernilai benar oleh sistem dibagi dengan jumlah keseluruhan pengenalan data yang dilakukan pada sistem seperti pada Persamaan (5) (David, 2011).

$$Precision = TP / (TP + FP) \quad (5)$$

Dimana:
TP : *True positive*
FP : *False positive*

Parameter *recall* menunjukkan tingkat keberhasilan sistem dalam mengenali suatu kategori. Secara matematis *Recall* dihitung dari jumlah pengenalan data yang bernilai benar oleh sistem dibagi dengan jumlah data yang seharusnya dapat dikenali sistem. *Recall* dihitung dengan Persamaan (6) (David, 2011).

$$Recall = TP / (TP + FN) \quad (6)$$

Dimana:
TP : *True positive*
FN : *False negative*

F1-measure merupakan gambaran pengaruh relatif antara *precision* dan *recall* atau disebut *harmonic mean.* Performa metode yang digunakan dapat disimpulkan dari nilai *F1-measure. F1-measure* dihitung dengan Persamaan (7) (David, 2011).

$$F1\text{-}measure = (2*P)*R / (P + R) \quad (7)$$

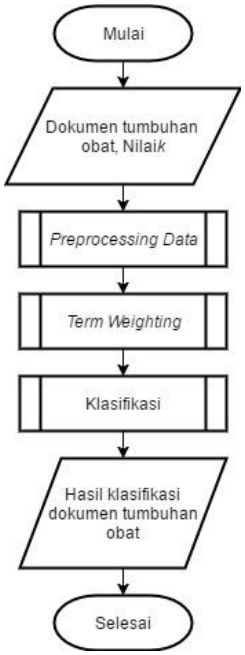
P : *Precision*
R : *Recall*

3. METODOLOGI & PERANCANGAN

Metodologi berisi penjelasan tentang gambaran sistem secara umum serta metode-metode yang digunakan dalam penelitian klasifikasi dokumen tumbuhan obat dengan metode *Improved k-Nearest Neighbor.* Gambar 2 menunjukkan tahapan di dalam penelitian yang akan dilakukan.

Gambar 2. Tahapan Penelitian

Selanjutnya adalah tahap perancangan sistem. Alur sistem secara umum terdiri dari tiga tahap yaitu *preprocessing*, *term weighting*, dan klasifikasi seperti pada Gambar 3.



Gambar 3. *Flowchart* Sistem

Preprocessing, *term weighting*, dan klasifikasi merupakan tahapan yang memiliki sub-proses. Pada *preprocessing* terdapat sub-proses *case folding*, *tokenizing*, *filtering*, dan *stemming*. Pada *term weighting* terdapat sub-proses menghitung *term frequency*, *document frequency*, *inverse document frequency*, dan TFIDF, sedangkan pada klasifikasi terdapat sub-proses menghitung *cossim*, *k* baru, dan probabilitas.

4. IMPLEMENTASI

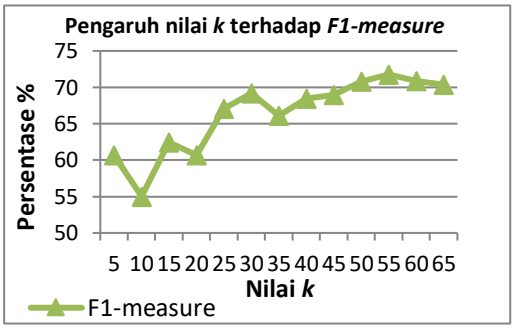
Implementasi sistem menggunakan bahasa pemrograman *Java* dengan *software* NetBeans IDE 8.1. Data *input* yang akan digunakan merupakan dokumentasi tanaman obat yang berisi deskripsi berupa *file* berformat *.txt*. Deskripsi tumbuhan mencakup habitus, jenis batang, daun, bunga, buah, biji, dan akar tumbuhan.

5. PENGUJIAN & ANALISIS

Berikut adalah hasil pengujian dan analisis dari implementasi klasifikasi dokumen tanaman obat menggunakan metode *Improved k-Nearest Neighbor*.

5.1. Pengujian Pengaruh Nilai *k*

Skenario pengujian pertama dilakukan untuk mengetahui pengaruh pertambahan nilai *k* awal terhadap akurasi. Gambar 4 merupakan grafik yang menunjukkan nilai rata-rata *F1-measure* untuk nilai *k*=5 sampai dengan *k*=65 dengan interval 5.

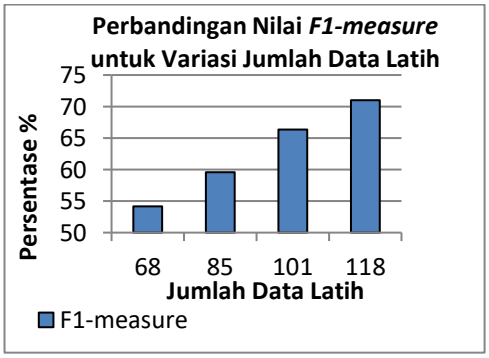


Gambar 4. Pengaruh Nilai *k* Terhadap Persentase *F1-measure*

Hasil *F1-measure* menunjukkan bahwa performa sistem telah cukup baik dalam mengklasifikan data uji. Dari Gambar 4 terlihat bahwa seiring bertambahnya nilai *k*, nilai *F1-measure* cenderung mengalami peningkatan. Nilai *F1-measure* tertinggi yaitu 71,77% diperoleh saat *k*=55 dan untuk nilai *k* selanjutnya menghasilkan nilai *F1-measure* yang stabil. Nilai *F1-measure* yang mengalami perubahan naik dan turun dipengaruhi oleh jumlah dan kategori dari tetangga terdekat atau *k* yang digunakan. Semakin besar jumlah *k* maka semakin tinggi kemungkinan data uji terklasifikasi dengan benar, namun kategori tetangga terdekat yang digunakan juga belum tentu sama dengan kategori yang dicari sehingga menyebabkan hasil penghitungan peluang yang berubah-ubah.

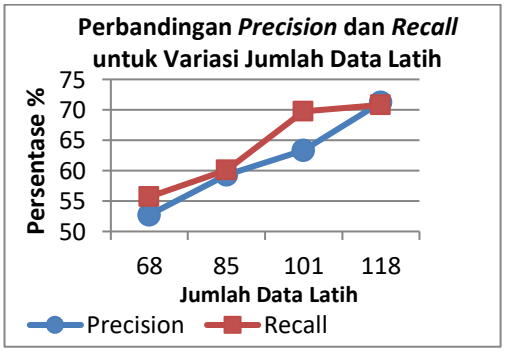
5.2. Pengujian Pengaruh Variasi Data Latih

Skenario pengujian kedua bertujuan untuk mengetahui pengaruh variasi jumlah data latih terhadap akurasi. Pengujian dilakukan menggunakan 169 data dengan proporsi data latih sebesar 70%, 60%, 50%, dan 40% atau sejumlah 118, 101, 85 serta 68 data. Gambar 5 merupakan diagram batang yang menunjukkan perbandingan nilai *F1-measure* untuk masing-masing proporsi data latih.



Gambar 5. Pengaruh Variasi Data Latih Terhadap *F1-measure*

Berdasarkan Gambar 5 diperoleh nilai *F1-measure* tertinggi terdapat pada data latih yang berjumlah 118 data yaitu 70,99% dan terendah pada data latih berjumlah 68 data yaitu 54,14%. Seiring bertambahnya jumlah data latih yang diujikan nilai *F1-measure* juga mengalami peningkatan. Peningkatan nilai *F1-measure* ini dapat terjadi karena semakin bertambahnya jumlah data latih maka jumlah variasi *term* unik juga semakin banyak sehingga peluang data uji terklasifikasi secara benar menjadi bertambah. Perbandingan nilai *precision* dan *recall* untuk pengujian kedua dapat dilihat pada Gambar 6.



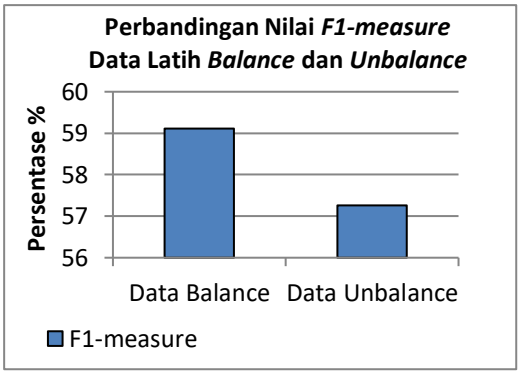
Gambar 6. Pengaruh Variasi Data Latih Terhadap *Precision* dan *Recall*

Precision merupakan persentase keakuratan hasil klasifikasi oleh sistem. Nilai rata-rata *precision* tertinggi diperoleh saat data latih berjumlah 118 data yaitu sebesar 71,20%, sedangkan nilai terendah pada saat data latih berjumlah 68 data yaitu sebesar 52,73%. Dari grafik tersebut terlihat bahwa seiring bertambahnya jumlah data latih maka nilai *precision* juga mengalami peningkatan. Hal ini dapat disebabkan oleh bertambahnya jumlah *term* unik yang dimiliki oleh data latih. Semakin banyak jumlah *term* unik pada data latih maka semakin tinggi potensi data uji terklasifikasi secara benar oleh sistem.

Recall menunjukkan tingkat keberhasilan sistem dalam mengenali suatu kategori. Tingkat keberhasilan tersebut ditunjukkan melalui perbandingan dari jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji sebenarnya dari suatu kategori. Dari nilai rata-rata *recall* yang dihasilkan, dapat disimpulkan bahwa sistem dapat mengenali kategori data uji dengan cukup baik.

5.3. Pengujian Pengaruh Data Latih Tidak Seimbang

Skenario pengujian ketiga bertujuan untuk mengetahui pengaruh data latih tidak seimbang terhadap performa metode *Improved k-Nearest Neigbors.* Gambar 7 merupakan diagram batang yang menunjukkan pengaruh data latih *balance* dan *unbalance* terhadap nilai *F1-measure.*



Gambar 7. Pengaruh Data Latih *Balance* dan *Unbalance* Terhadap *F1-measure*

Berdasarkan Gambar 7 terlihat bahwa nilai rata-rata *F1-measure* dari data *balance* ke *unbalance* mengalami penurunan sebesar 1,8%. Saat nilai *F1-measure* pada kategori *unbalance* mengalami penurunan artinya hasil klasifikasi kurang akurat. Hal tersebut disebabkan oleh jumlah data latih yang sedikit pada kategori *unbalance.* Jumlah data latih yang sedikit menyebabkan *term* unik yang dihasilkan juga berjumlah sedikit, padahal jumlah *term* unik pada setiap kategori berpengaruh pada peluang penentuan kategori dokumen uji. Selain itu, jumlah *term* unik yang sedikit pada kategori *unbalance* juga menyebabkan data uji memiliki

kecenderungan untuk terklasifikasi ke dalam kategori dengan *term* unik lebih banyak.

6. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang sudah dilakukan, maka dapat diambil kesimpulan bahwa nilai *k* berpengaruh terhadap akurasi sistem klasifikasi yaitu semakin besar nilai *k* maka semakin tinggi peluang data uji terklasifikasi dengan benar dengan nilai *F1-measure* tertinggi sebesar 71,77% pada saat nilai *k* awal=55 sedangkan nilai *F1-measure* terendah sebesar 55,00% saat nilai *k* awal=10.

Selain itu, dari pengujian pengaruh variasi jumlah data latih serta data latih *unbalanced* dapat disimpulkan bahwa jumlah data latih juga berpengaruh terhadap performa sistem klasifikasi yaitu semakin banyak jumlah data latih maka semakin besar peluang data uji terklasifikasi secara benar dan hasil akurasi data latih *balance* lebih baik 1,9% dari data latih *unbalance*.

7. DAFTAR PUSTAKA

Baoli, Siwen, Qin. 2003. *An Improved k-Nearest Neighbour Algorithm for Text Categorization. To Appear in the Proceedings of the 20th International Conference of Computer Processing of Oriental language.* Shenyang, China. 16 Juni 2003.

David M.W. 2007. *Evaluation: From Precision, Recall And F-Measure To ROC, Informedness, Markedness & Correlation*, [e-journal]. 2(1), 37-63.

Paskianti, Kristina. 2011. Klasifikasi Dokumen Tumbuhan Obat Menggunakan Algoritma *KNN Fuzzy*. S1. Institut Pertanian Bogor.

Tala, Fadillah Z. 2003. A *Study of Stemming Effects on Information Retrieval in* Bahasa Indonesia. Universiteit van Amsterdam, The Netherlands.

Tan, Pang Ning. Michael. Steinbach, dan Vipin. Kumar. 2006. *Introduction to Data Mining*. *1st penyunt. Boston: Pearson Addison Wesley*.

Yiming, Jaime G.C, Rulf D Brown, Thomas Pierce, Brian T Achibald, Xin Liu. 1999. *Learning Approach for Detecting and Tracking News Event. Journal of* IEEE *Intelegent Sistem.* Carbegie Mellon University.

Yong Z, Youwen L, Xhixion X. 2009. *An Improved kNN Text Classification Algorithm based on Clustering. Journal of Computers,* 4(3). *Conference on Neural Information Processing.*