Pengklasifikasian Dokumen Berbahasa Arab Menggunakan K-Nearest Neighbor

Fahrul Agus[1], Heliza Rahmania Hatta[2], Mahyudin[3]
Jl. Barong Tongkok Kampus Gn. Kelua Universitas Mulawarman, Samarinda 75123
[1,2,3]Program Studi Ilmu Komputer FMIPA Universitas Mulawarman
[1]fahrulagus@gmail.com, [2]heliza_rahmania@yahoo.com, [3]a.n.mahyudin@gmail.com

Abstrak

Algoritma k-Nearest Neighbor (kNN) adalah suatu algoritma supervised dimana hasil dari query akan diklasifikasikan berdasarkan mayoritas dari kategori. Algoritma ini bertujuan untuk mengklasifikasi objek baru berdasarkan atribut dan training sampel. Uji coba dilakukan pada dokumen teks berbahasa Arab diambil dari koleksi dokumen surat kabar Arab Al-Jazirah. Algoritma kNN dipilih karena lebih sederhana, efektif, dan dapat diaplikasikan pada jumlah training yang sedikit. Namun, dalam implementasinya kNN memerlukan waktu komputasi yang lama dalam pengklasifikasiannya, karena melakukan perhitungan jarak ke semua training sampel. Hasil uji coba membuktikan bahwa penggunaan algoritma kNN dapat melakukan klasifikasi dokumen berbahasa Arab dengan nilai lokal optimal F-Measure terbaik sebesar 0.86 dan tingkat akurasi 94%.

Kata kunci— *k-Nearest Neighbor, Klasifikasi dokumen berbahasa Arab*

Abstract

k-Nearest Neighbor (kNN) algorithm is a supervised algorithm where the results of the query will be classified based on the majority of categories. This algorithm aims to classify new objects based on attributes and training samples. Tests conducted on Arabic documents are taken from document collection Arab newspaper Al-Jazirah. kNN algorithm was chosen because it is more simple and more effective than others. It can be applied into a small size of training. However, kNN implementation requires a long time of computation in classification due to calculating the distance to all training samples. The test result proves that the use of kNN algorithms can perform classification of Arabic documents with the best F-Measure of 0.86 optimal local value and 94% accuracy level.

Keywords— *k-Nearest Neighbor, Arabic Documents Classification.*

1. PENDAHULUAN

Perkembangan teknologi informasi meningkatkan ketersediaan penyampaian dan penyimpanan informasi melalui internet, dimana internet menjadi media publikasi yang sangat populer. Banyak informasi digital yang tidak terstruktur sebagai akibat dari perkembangan teknologi informasi membutuhkan suatu cara pengorganisasian atau pengelompokan informasi untuk kemudahannya, oleh karena itu kategorisasi teks secara otomatis adalah merupakan solusi untuk masalah tersebut karena dengan signifikan dapat mereduksi biaya kategorisasi manual.

Pengklasifikasian dokumen didasarkan atas kesamaan fitur atau kesamaan isi dokumen. Klasifikasi dilakukan dengan cara memasukan dokumen-dokumen kedalam kategori-kategori yang sudah ditentukan sebelumnya. Dokumen baru yang akan masuk dalam kategori berdasarkan kesamaan fitur dengan kategori tersebut. Secara garis besar metode klasifikasi dibagi menjadi dua bagian yaitu *supervised learning* dan *unsupervised learning*. *Supervised learning* adalah pengelompokan dokumen yang sudah didefinisikan informasi-informasi pada kelas atau kategori sebelumnya sedangkan pada *unsupervised learning* adalah pengelompokan kategori secara otomatis tanpa didefinisikan informasi sebelumnya.

Menurut (Hatta, 2013) [1] bahwa klasifikasi teks dokumen juga dapat diterapkan dalam Bahasa Arab, ini dikarenakan Bahasa Arab memiliki morfologi yang lebih kaya dan kompleks dari pada bahasa Inggris ataupun bahasa Indonesia, dimana dalam teks Bahasa Arab kita dapat mencari bentuk

morfologi sebuah kata dari *stem* atau kata dasarnya. *Stemming* merupakan suatu proses menemukan kata dasar dari sebuah kata, dengan menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan kombinasi dari awalan dan akhiran (*confixes)* pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar.

Klasifikasi bertujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Sistem klasifikasi dapat membantu menaikkan ketelitian dan kepercayaan dari diagnosa dan meminimalkan kemungkinan kesalahan, maupun membuat waktu diagnosa lebih efisien[2]. Metode - metode klasifikasi yaitu *Decision/classification Trees*, *Rain Forest*, *Naïve Bayesian*, *Neural Network*, *Genetic Algorithm*, *Fuzzy*, *Case-Based Reasoning*, *K-Nearest Neighbor (KNN)* dan *Support Vector Machines* [3].

Penelitian tentang kategorisasi teks secara otomatis sering dilakukan pada beberapa bahasa diantaranya pada bahasa Inggris, bahasa Cina, dan bahasa Indonesia, namun sedikit sekali penelitian tentang kategorisasi teks untuk dokumen berbahasa Arab. Kategorisasi teks menggunakan algoritma *Naïve Bayes* untuk bahasa Arab sudah diimplementasikan dengan hasil bahwa *Naïve Bayes* dapat diaplikasikan dalam Bahasa Arab [4].

Metode pendekatan *machine learning* dengan tehnik *KNN* mampu untuk melakukan klasifikasi dokumen melalui sampel. Metode *KNN* dipilih karena lebih sederhana, efektif, dan juga dapat diaplikasikan pada jumlah training data yang sedikit dan metode ini mudah dalam mengelompokkan, dan mudah dimengerti [5]. Tugas akhir ini dilakukan penerapan metode *K-Nearest Neighbor (KNN)* untuk diimplementasikan melalui suatu perangkat lunak pengklasifikasi dokumen berbahasa Arab.

2. METODE PENELITIAN

A. Metode Pengumpulan Data

Penelitian ini dilakukan untuk memperoleh data dan informasi yang berhubungan dengan penelitian yang akan penulis lakukan, penulis menggunakan metode Studi Literatur. Tahap ini, dilakukan berbagai pengumpulan informasi terkait beberapa hal berikut

1. Pengumpulan informasi tentang bagaimana *indexing* atau *preprocessing* pada dokumen teks dilakukan. Informasi ini terkait dengan *stoplist*, pembobotan TF-IDF, *Arabic Stemming*.
2. Pengumpulan informasi dan sumber pembelajaran tentang algoritma *KNN*.
3. Pengumpulan informasi tentang *cosine distance* yang dipakai sebagai representasi nilai jarak dalam *KNN*.
4. Pengumpulan informasi tentang apa metode evaluasi hasil klasifikasi yang dipilih dan bagaimana metode itu diimplementasikan. Informasi ini berkaitan dengan metode yang digunakan yakni *recall*, *precision*, dan *F-measure*.
5. Studi literatur dari beberapa buku, prosiding dan jurnal-jurnal mengenai klasifikasi dan Metode *K-Nearest Neighbor* sebagai referensi dan bahan masukan dalam tinjauan pustaka serta sebagai penunjang pada proses penelitian agar tidak menyimpang dari ketentuan yang telah ada sebelumnya.
6. Jurnal yang menjadi refrensi utama yaitu :
   Jurnal Heliza "Metode hibridasi *Ant Colony Optimization* Dan *Information Gain* Untuk Seleksi Fitur Pada Kategorisasi Dokumen Teks arab".
   Jurnal Abdul Rosaq "Klasifikasi Dokumen Teks Berbahasa Arab Mengunakan Algoritma *Naïve Bayes".*

B. Analisis Kebutuhan Non-Fungsional

1. Kebutuhan Perangkat Keras
   Perangkat keras yang akan penulis gunakan pada penelitian ini, yaitu:
   a. Processor Intel Core i5 2.50 GHz
   b. Memori (RAM) 4 GB

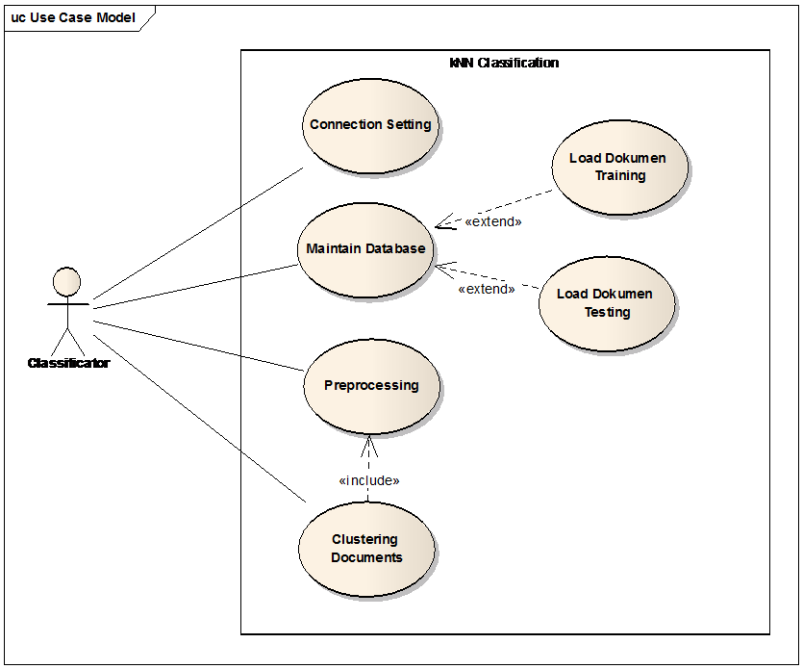
c. Kartu Grafis Nvidia Geforce 635M.

2. Kebutuhan Perangkat Lunak
   Perangkat lunak yang akan penulis gunakan pada penelitian ini, yaitu:
   a. Java
   b. Google Web Toolkit (GWT)
   c. Library Java Script (EXTJS)
   d. MySQL

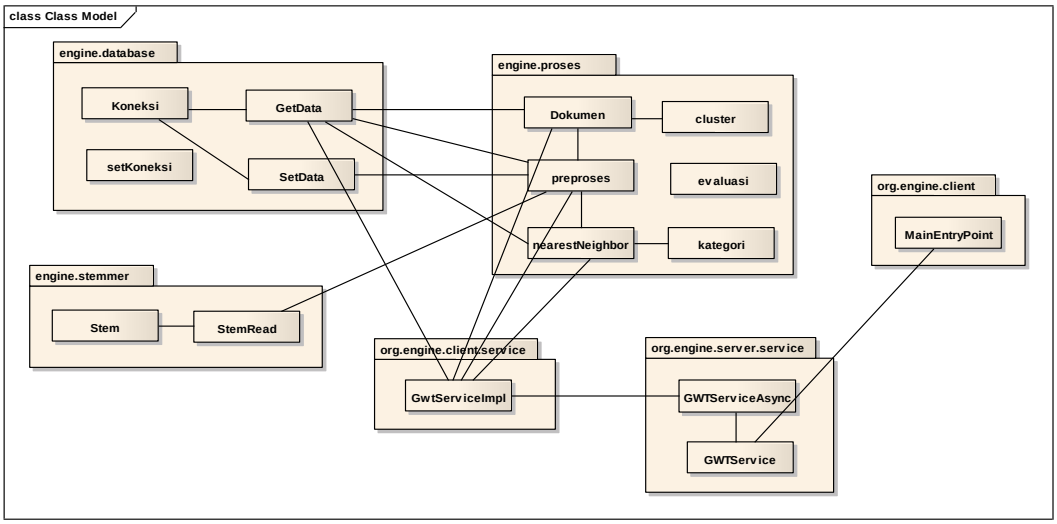
C. Analisis Kebutuhan Fungsional

Use Case Diagram

Desain use case model seperti pada Gambar 1 dalam diagram use case tersebut, aktor yang melakukan proses klasifikasi disebut sebagai classificator, berdasarkan use case tersebut, dibuat suatu class diagram seperti pada Gambar 2 yang melibatkan objek-objek utama pada aplikasi ini.



Gambar 1. Diagram *Use Case* Perangkat Lunak.



Gambar 2. Diagram Class Model perangkat lunak

D. Perancangan Data

Data yang digunakan dalam proses pengklasifikasian teks berbahasa Arab dibagi menjadi tiga bagian yaitu data masukan, data proses, dan data keluaran. Data masukan merupakan input dari pengguna perangkat lunak. Data proses adalah ketika tahap-tahap pengklasifikasian sedang dilakukan, sedangkan data keluaran adalah data yang ditampilkan kepada pengguna perangkat lunak. Penjelasan masing-masing jenis data

1. Data Masukan
   Data masukan adalah data-data yang digunakan sebagai input dari aplikasi, dalam penelitian ini, dataset yang menjadi masukan ke dalam sistem adalah kumpulan dokumen surat kabar Arab, seperti Al-Jazirah (http://www.al-jazirah.com/). Dokumen *training* digunakan sebagai objek pembelajaran untuk proses seleksi fitur. Data *training* ini yang nanti akan dibandingkan atribut-atributnya dengan dokumen *testing* yang ada.
2. Data Proses
   Data proses adalah data-data yang digunakan sebagai syarat suatu proses pada aplikasi. Sederhananya, data proses adalah data-data yang merupakan batasan, parameter, variabel, atau konstanta yang harus didefinisikan oleh pengguna, untuk mendapatkan data keluaran aplikasi nantinya. Data-data proses yang terdapat pada *preprocessing* tidak ada pilihan *stopword* atau *stemmer* karena pada *preprocessing* ini penggunaan *stopword* dan *stemmer* langsung digunakan dalam kode program.
3. Data Keluaran
   Data keluaran adalah data-data yang dihasilkan oleh aplikasi setelah proses-proses tertentu pada aplikasi selesai dijalankan, beberapa contoh data keluaran dari aplikasi ini adalah:
   a. Data dokumen ter-*stemming* dalam bentuk dokumen bahasa Arab sesuai dengan isi dokumen masing-masing.
   b. Data dokumen ter-*preprocessing* dalam bentuk *vektor term* dengan *frekuensi* dan bobot yang telah dihitung pada fase *preprocessing* dokumen.
   c. Data dokumen *testing* hasil kategori yang sudah diklasifikasikan sesuai dengan nilai parameter *k* tertentu.

E. Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri atas perancangan data masukan, proses, dan antarmuka aplikasi. Perancangan data masukan meliputi perancangan *corpus* dokumen berita berbahasa Arab. Perancangan data masukan lainnya adalah perancangan data *stoplist* dan kamus kata dasar dimana kamus pada bahasa arab terdiri dari kata dasar yang mempunyai tiga karakter dan empat karakter, dalam tahap perancangan data juga dilakukan perancangan skema *database* dalam *conseptual data model* dan *physical data model* menggunakan *Sybase Power Designer*, dengan target *database*-nya adalah MySQL.

Perancangan proses dilakukan identifikasi proses-proses utama yang akan diimplementasikan pada aplikasi Java yang akan dibuat, terdapat 4 proses utama yakni *maintenance database*, *preprocessing* dokumen, *classification* dokumen, dan evaluasi hasil *classification*.

Proses *preprocessing*, setiap dokumen berita diubah menjadi vektor dokumen dengan urutan proses sebagai berikut:

a. *Filtering*, yakni pengeliminasian karakter-karakter ilegal (angka dan simbol) dalam isi dokumen.
b. *Stoplist removal*, yakni penghilangan karakter-karakter yang termasuk dalam kategori *stopword* atau kata-kata yang memiliki frekuensi tinggi terdapat dalam *data stoplist*.
c. *Terms extraction*, yakni mengekstraksi *term-term* (kata) dari setiap dokumen yang akan diolah dan disusun ke dalam vektor *terms* yang merepresentasikan dokumen tersebut.
d. *Stemming*, yakni mengembalikan bentuk dasar dari setiap *term* yang ditemukan pada vektor dokumen dan mengelompokkannya berdasarkan *term-term* yang sejenis.

e. *TF-IDF weigthing,* yakni melakukan pembobotan TF-IDF pada setiap *term* yang ada pada vektor dokumen.

Proses klasifikasi dokumen yang akan diterapkan pada tugas akhir ini adalah akan menerapkan algoritma KNN pada dokumen teks berbahasa Arab. Proses evaluasi atas hasil klasifikasi (clustering) dibutuhkan untuk mengetahui kinerja algoritma yang diimplementasikan pada data uji. Metode evaluasi yang digunakan adalah metode F-measure, Recall, Precision. Metode ini dipilih karena adanya informasi kategori dari data uji yang digunakan.

F. Perancangan Proses

Perancangan proses dalam perangkat lunak ini dibuat dengan menggunakan konsep UML (*Unified Modelling Languange*), dengan menggunakan perangkat lunak Enterprise Architect 7.0. Perancangan berbasis UML minimal mengikuti dua hal, yakni *system structure* dan *system behavior*. *System structure* mendeskripsikan *objek-objek* dalam sistem dan relasinya. *System behavior* menunjukkan perilaku objek pada saat berinteraksi satu sama lain. UML mendefinisikan banyak tipe diagram untuk memodelkan kedua hal tersebut. Pada perancangan aplikasi ini, *system structure* dimodelkan dalam suatu *class diagram*, sedangkan *system behavior* dimodelkan dalam *activity diagram*. Pemodelan proses-proses utama dalam aplikasi ini dibuat dalam bentuk sederhana, tanpa bermaksud kehilangan detail perancangannya, sebelum membangun diagram-diagram tersebut, terlebih dahulu diperlukan analisa atas proses-proses utama dalam aplikasi. Secara garis besar, proses-proses tersebut adalah:

a. Mengatur koneksi ke *database*
   Proses ini berkaitan dengan mengatur variabel-variabel koneksi aplikasi ke database (*database URL*, *username*, *password*). Proses ini dilakukan saat eksekusi awal aplikasi, dimana variabel-variabel tersebut sudah dibuat *default* pada aplikasi, apabila proses koneksi gagal, maka user dihadapkan pada menu koneksi dan mengatur ulang variabel-variabel *default* tersebut.
b. *Maintenance database*
   Data-data masukan untuk aplikasi, data set dokumen sudah tersimpan terlebih dahulu dalam database dan beberapa sub proses adalah proses *load* dokumen *training*, *load* dokumen *testing*.
c. *Preprocessing* dokumen
   Proses ini merupakan proses *indexing* atau *preprocessing* pada dokumen yang telah disimpan pada *database*, beberapa sub-proses diantaranya adalah *filtering*, *stoplist/stopword removal*, *stemming*, dan *term weigthing*.
d. *Clustering* dokumen
   Proses *clustering* atau klasifikasi memiliki dependensi, yakni mensyaratkan proses *preprocessing* telah dilakukan sebelumnya.
e. Evaluasi Hasil *Clustering*
   Evaluasi Hasil *Clustering* bertujuan untuk melihat *precision*, *recall*, *F-Measure*, dan *accuration* dari hasil kategorisasi yang telah dilakukan oleh sistem.

G. Uji Coba Dan Evaluasi

Tahapan pembuatan perangkat lunak selesai, maka tahapan penelitian ini akan dilanjutkan dengan melakukan suatu uji coba terhadap sistem yang telah dibuat. Tahap ini dilakukan uji coba terhadap sistem yang telah dibuat, mengamati kinerja sistem yang baru dibuat, serta mengidentifikasi kendala yang mungkin timbul. Dokumen yang digunakan sebagai data ujicoba diambil dari koleksi dokumen surat kabar Arab online, Al-Jazirah (http://www.al-jazirah.com/). Pengujian tingkat keberhasilan klasifikasi dokumen dengan cara melihat nilai evaluasi F-measure yang dihasilkan.

3. HASIL DAN PEMBAHASAN

A. Klasifikasi Dokumen

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas

tertentu dari sejumlah kelas yang ada. Ada dua pekerjaan utama dalam klasifikasi yang dilakukan Prasetyo dkk [6], yaitu:

1. Pembangunan model sebagai prototipe untuk disimpan sebagai memori.
2. Penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui kelas mana objek data tersebut dalam model yang sudah disimpannya.

Klasifikasi pertama kali diterapkan pada bidang tanaman yang mengklasifikasikan suatu spesies tertentu, seperti yang dilakukan oleh Carolus von Line (dikenal dengan nama *Carolus Linnaeus*) yang pertama kali mengklasifikasikan spesies berdasarkan karakteristik fisik [7].

Klasifikasi bertujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Sistem klasifikasi dapat membantu menaikkan ketelitian dan kepercayaan dari diagnosa dan meminimalkan kemungkinan kesalahan, maupun membuat waktu diagnosa lebih efisien [2].

Klasifikasi dokumen adalah proses pengelompokan dokumen sesuai dengan kategori yang dimilikinya. Klasifikasi dokumen merupakan masalah yang mendasar namun sangat penting karena manfaatnya cukup besar mengingat jumlah dokumen yang ada setiap hari semakin bertambah, sebuah dokumen dapat dikelompokan ke dalam kategori tertentu berdasarkan kata-kata dan kalimat-kalimat yang ada didalam dokumen tersebut. Kata atau kalimat yang terdapat didalam sebuah dokumen memiliki makna tertentu dan dapat digunakan sebagai dasar untuk menentukan kategori dari dokumen [8].

Klasifikasi data terdiri dari 2 langkah proses. Pertama adalah *learning* (fase *training*), dimana algoritma klasifikasi dibuat untuk menganalisa data *training* lalu direpresentasikan dalam bentuk *rule* klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari *rule* klasifikasi. Proses klasifikasi didasarkan pada empat komponen [9] yaitu:

1. Kelas
   Variabel dependen yang berupa kategorikal yang merepresentasikan 'label' yang terdapat pada objek, contohnya: resiko penyakit jantung, resiko kredit, *customer loyalty*, jenis gempa.
2. *Predictor*
   Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.
3. *Training dataset*
   Satu set data yang berisi nilai dari kedua komponen diatas yang digunakan untuk menentukan kelas yang cocok berdasarkan *predictor.*
4. *Testing dataset*
   Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

Text mining klasifikasi mengacu kepada aktifitas menganalisis atau mempelajari himpunan dokumen teks *pre-classified* untuk memperoleh suatu model atau fungsi yang dapat digunakan untuk mengelompokkan dokumen teks lain yang belum diketahui kelasnya ke dalam satu atau lebih kelas kelas *predefined* tersebut.

Klasifikasi termasuk pembelajaran jenis *supervised learning*. Jenis lain adalah *unsupervised learning* atau dikenal sebagai *clustering,* pada *supervised learning*, data latihan mengandung pasangan data *input* (biasanya vektor) dan *output* yang diharapkan, sedangkan pada *unsupervised learning* belum ditentukan target *output* yang harus diperoleh.[10]

Text document clustering adalah *clustering* dengan spesialisasi pada dokumen berbasis teks. *Indexing* atau *preprocessing* juga berlaku dalam *text document clustering*. Teknik yang paling banyak dipakai adalah dengan merepresentasikan tiap dokumen teks dalam *vector-space model*. Dalam model ini, setiap dokumen *D*, direpresentasikan sebagai suatu vektor $c = \{t_1,t_2,...,t_n\}$ dimana t_n adalah frekuensi *term* ke-*n* pada dokumen bersangkutan [11]. Terkadang representasi frekuensi ini diganti menjadi format biner atau *boolean* (0 atau 1) yang menandakan ada-tidaknya *term* tersebut pada dokumen bersangkutan. Salton menyarankan untuk merepresentasikannya dalam bentuk yang sudah mengalami

pembobotan, seperti *TF-IDF*. Berdasarkan struktur hasil *clustering*-nya, maka teknik *clustering* dapat dibedakan menjadi dua tipe yakni *Hierarchical* dan *Non-hierarchical* [12].

Teknik hierarchical menghasilkan urutan partisi yang bersarang (nested) dengan satu cluster utama pada level atas, dan cluster-cluster kecil dibawah. Sebaliknya, teknik non-hierarchical menghasilkan partisi yang tidak bersarang (unnested) dengan membagi dokumen-dokumen dalam beberapa cluster awal yang ditentukan,kemudian mengubah posisi dokumen dalam cluster yang telah ada hingga solusi terakhir dicapai.

Proses klasifikasi teks dapat dibagi ke dalam dua fase, yaitu:

1. Fase *information retrieval* (IR) untuk mendapatkan data numerik dari dokumen teks. Langkah pertama yang dilakukan pada fase ini adalah *feature extraction*. Pendekatan yang umum digunakan adalah distribusi frekuensi kata. Nilai numerik yang diperoleh dapat berupa berapa kali suatu kata muncul didalam dokumen, 1 jika kata ada didalam dokumen atau 0 jika tidak ada (*biner*), atau jumlah kemunculan kata pada awal dokumen. Fitur yang diperoleh dapat direduksi agar dimensi vektor menjadi lebih kecil. Beberapa pendakatan *feature reduction* dapat diterapkan seperti menghapus *stopword* dan *stemming.*
2. Fase klasifikasi utama, suatu algoritma memproses data numerik diatas untuk memutuskan ke kategori mana teks baru (bukan contoh) ditempatkan, terdapat beberapa algoritma klasifikasi yang merupakan kajian dibidang statistika dan *machine learning* yang dapat diterapkan pada fase ini, diantaranya adalah *naïve Bayesian, Rocchio, Decision Tree, k-Nearest Neighbor* (k-NN), *Neural Network* (NN), dan *Support Vector Machines* (SVM). Teknik-teknik tersebut berbeda dalam mekanisme pembelajaran dan representasi model yang dipelajari.

Manfaat dari klasifikasi dokumen adalah untuk pengorganisasian dokumen, dengan jumlah dokumen yang sangat besar, untuk mencari sebuah dokumen akan lebih mudah apabila kumpulan dokumen yang dimiliki terorganisir dan telah dikelompokan sesuai kategorinya masing-masing. Contoh aplikasi penggunaan klasifikasi dokumen teks yang banyak digunakan adalah *email spam filtering*, pada aplikasi *spam filtering* sebuah email diklasifikasikan apakah *email* tersebut termasuk *spam* atau tidak dengan memperhatikan kata-kata yang yang terdapat dalam email tersebut. Aplikasi ini telah digunakan oleh banyak *provider* jasa layanan email.

B. K-Nearest Neighbor

Mengklasifikasikan sekumpulan data, sangatlah banyak cara yang bisa digunakan. Pengelompokkan dapat dilakukan dengan menggunakan *Naive Bayes* (*NB*), *K-Nearest Neighbor* (*KNN*), *Support Vector Machines* (*SVM*) dan *Neural Network* (*NN*). Metode pengklasifikasian *data mining* yang paling sering digunakan adalah *K-Nearest Neighbor*, hal ini disebabkan karena metode ini mudah dalam mengelompokkan, efektif dan mudah dimengerti[5].

K-Nearest Neighbor (KNN) adalah alat yang sangat sederhana, tapi kuat. Ini telah digunakan dalam berbagai aplikasi dan khususnya dalam klasifikasi tugas. Ide kunci dibalik *KNN* adalah bahwa pelatihan serupa sampel memiliki nilai *output* yang sama [13].

K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Teknik ini sangat sederhana dan mudah diimplementasikan. Mirip dengan teknik klastering, mengelompokkan suatu data baru berdasarkan jarak data baru itu ke beberapa data tetangga *(neighbor)* terdekat [14].

Metode *KNN* merupakan metode pengklasifikasian data yang bekerja relatif lebih sederhana dibandingkan dengan metode pengklasifikasian data lainnya. Metode ini berusaha mengklasifikasikan data baru yang belum diketahui *class*-nya dengan memilih data sejumlah k yang letaknya paling dekat dengan data baru tersebut. *Class* terbanyak dari data terdekat dipilih sebagai *class* yang diprediksi untuk data baru. Untuk nilai k, biasanya digunakan dalam jumlah ganjil untuk menghindari bila terjadi jumlah pemunculan yang sama dalam proses pengklasifikasian [5].

Algorima *KNN* adalah suatu algoritma yang menggunakan *supervised clustering* dimana hasil dari *query instance* yang baru diklasifikan berdasarkan mayoritas dari kategori pada *KNN*. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru berdasarkan atribut dan *training sample*. *Classifier*

tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada *memory*, diberikan titik *query*, akan ditemukan sejumlah *k* obyek atau (titik *training*) yang paling dekat dengan titik *query*. Klasifikasi menggunakan *voting* terbanyak diantara klasifikasi dari *k* obyek. Algoritma *KNN* menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru[15].

Algoritma *KNN* sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan *KNN*-nya. *Training sample* diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample, sebuah titik pada ruang ini ditandai kelac c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut, dekat atau jauhnya tetangga dihitung berdasarkan Cosine Distance.

Fase training algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor training sample dihitung dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai *k* yang terbaik untuk algoritma ini tergantung pada data, secara umum nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai *k* yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan *training data* yang paling dekat (dengan kata lain, *k = 1*) disebut algoritma *nearest neighbor*.

Ketepatan algoritma *KNN* sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

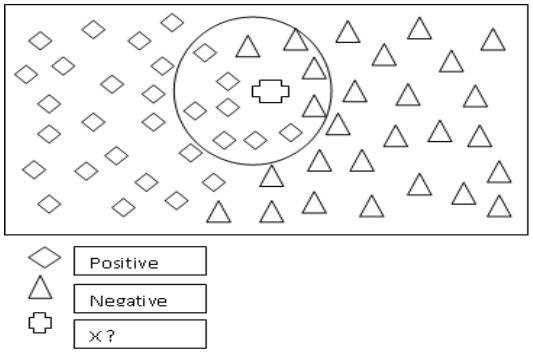
Algoritma *KNN* memiliki beberapa kelebihan yaitu ketangguhan terhadap *training data* yang memiliki banyak *noise* dan efektif apabila *training* data-nya besar dan juga dapat diterapkan untuk training data yang sedikit. Cara kerja algoritma k-NN adalah dengan memilih dokumen latih sejumlah k (k-values) yang letaknya terdekat dari dokumen uji. Kategori dokumen latih yang paling banyak muncul/ditemui pada sejumlah k terdekat, nantinya dipilih sebagai kategori yang diprediksikan untuk dokumen uji [16].

Kelemahan *KNN* adalah *KNN* perlu menentukan nilai dari parameter *k* (jumlah dari tetangga terdekat), *training* berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*.

Berikut ini adalah algoritma *KNN* :

1. Menghitung nilai kemiripan (*similarity*). Metode perhitungan yang sering digunakan adalah :
   a. *Euclidean distance*,
   b. *Manhatten Distance*,
   c. *Cosine Distance*
2. Mengurutkan hasil dari perhitungan nilai kemiripan/ketidakmiripan secara terurut.
3. Menentukan nilai *k* dan mengambil *k* jumlah tetangga dari hasil langkah 2.
4. Menentukan kelas dari data uji berdasarkan kelas yang paling banyak muncul dari hasil langkah 3.

Gambar 3, terdapat dua kelas yakni kelas *positive* dan kelas *negative*, dan terdapat sebuah data segitiga kuning sebagai *x* yang akan diklasifikasikan kedalam kelas *positive* atau kelas *negative*.

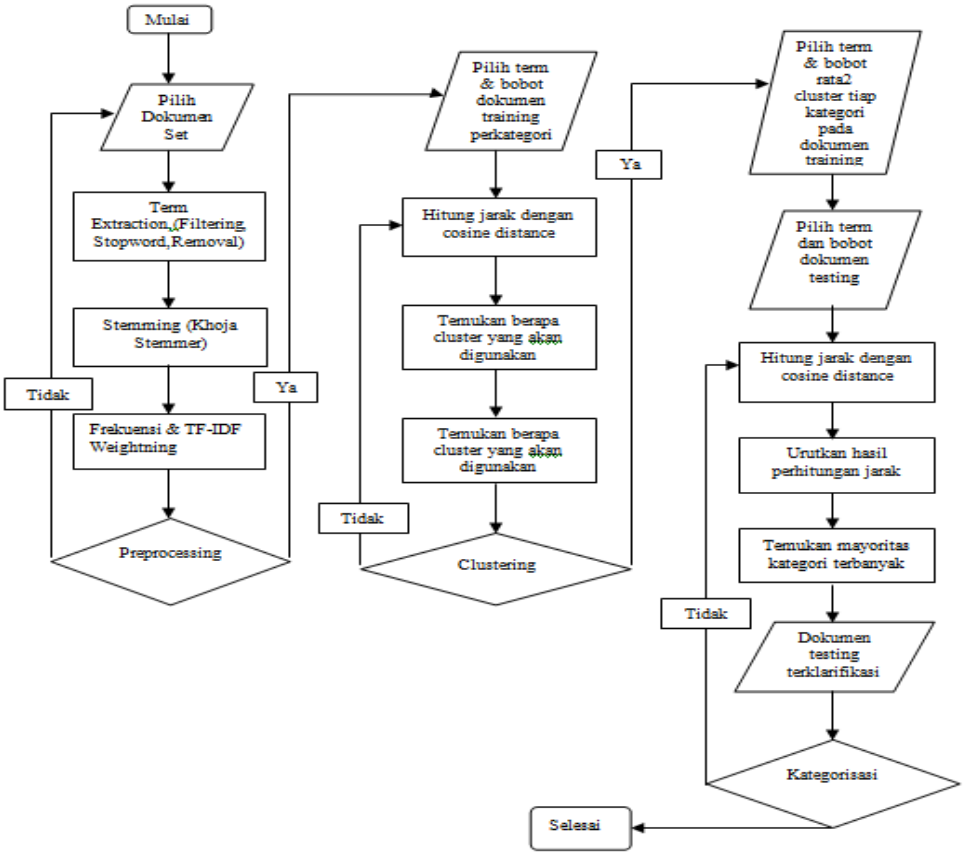


Gambar 3. Nearest Neighbour

Gambar 3 digunakan *k*=11 (ini dapat dilihat terdapat tujuh buah anggota kelas dalam lingkaran besar) dan setelah dihitung jarak *x* dengan semua anggota kelas ternyata ada dua kemungkinan untuk data *x*, yakni kelas *positive* atau kelas *negative*, karena data kelas *positive* berjumlah lebih banyak (frekuensi data lebih tinggi), yakni terdapat tujuh data sedangkan anggota kelas *negative* hanya empat data maka dapat disimpulkan bahwa kelas *x* dapat diklasifikasikan kedalam kelas *positive* [17].

Secara garis besar, skema perangkat lunak yang dibangun dapat dilihat pada Gambar 4. Terdapat beberapa hal yang perlu diperhatikan dalam mendesain perangkat lunak Tugas Akhir ini, yakni perancangan data, perancangan proses, dan perancangan antarmuka. Penjelasan atas proses-proses dalam skema perangkat lunak pada Gambar 4. ini dapat dilihat pada bagian perancangan proses.

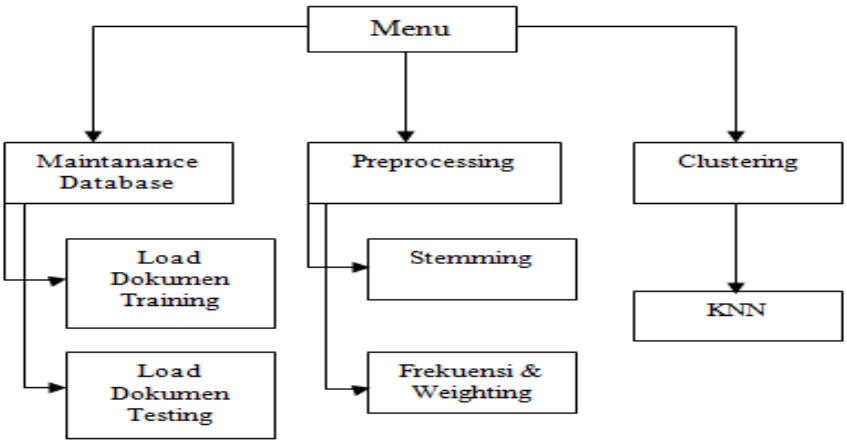
Gambar 4 merupakan skema perangkat lunak yang menggambarkan proses-proses utama dalam aplikasi yang akan dibuat, apabila diperinci dengan diagram alir maka akan terlihat jelas alur proses dari inisialisasi dokumen dari mulai tahapan preprocessing, clustering perkategori, klasifikasi dokumen sampai dokumen testing terklasifikasi. Diagram alir proses klasifikasi dokumen dapat dilihat pada Gambar 5.



Gambar 4. Diagram Alir Klasifikasi Dokumen

Gambar 5 menunjukkan desain conseptual data model application, terdapat beberapa entitas yang digunakan diantaranya adalah entitas dokumen untuk menyimpan field-field seperti id_dok, id_kategori, judul, isidecoded, isi, stem, status, tag dan label. Entitas dokumen ini berelasi dengan entitas kategori. Isidcoded untuk menyimpan hasil docede dari bahasa arab, karena teks bahasa arab belum di support oleh Java ke MySQL sehingga diperlukan decode pada proses update atau insert tabel dokumen. Field tag digunakan sebagai penanda kategori awal pada dokumen yang berstatus testing, sedangkan field stem adalah untuk menyimpan hasil stemming bahasa Arab dengan di decoded terlebih dahulu.

Perancangan antarmuka berguna untuk menentukan interaksi perinteraksi sesuai proses-proses yang telah didefinisikan sebelumnya. Secara garis besar, desain menu-menu utama untuk antarmuka aplikasi dapat dilihat pada Gambar 6. Menu "*Maintenance Database*" berkaitan dengan implementasi proses *maintenance* data, yakni *load* dokumen *training* dan *load* dokumen *testing*. Menu "*Preprocessing*" berkaitan dengan *preprocessing* dokumen, melalui menu ini dapat dilakukan proses *stemming* dan *pembobotan*. Menu "*Clustering*" merupakan implementasi proses clustering dari klasifikasi dokumen dengan kNN.



Gambar 5. Menu Antarmuka Aplikasi

C. Data Uji Coba

Data yang digunakan dalam uji coba ini merupakan kumpulan dokumen surat kabar Arab, seperti Al-Jazirah (http://www.al-jazirah.com/). Kategori yang dibahas dalam surat kabar Arab ini diantaranya diambil 5 kategori pembahasan yang menerangkan tentang: Seni, Budaya, Ekonomi, Internasional, dan Lokal.

Dokumen-dokumen ini dikelompokan berdasarkan kategorinya yang dipisah menjadi dokumen *training* dan dokumen *testing*. Dokumen training berjumlah 75 dokumen/halaman sedangkan pada dokumen *testing* berjumlah 35 dokumen/halaman. Data uji coba pada aplikasi ini dapat dilihat pada Tabel 1

Tabel 1. Data Uji Coba Aplikasi

No	Kategori	Jumlah file
1	Seni	15
2	Budaya	15
3	Ekonomi	15
4	Internasional	15
5	Lokal	15
Jumlah		75

Pengujian aplikasi yang dibangun dalam Tugas Akhir ini, adalah *clustering phase*. Pengujian *clustering phase* tujuannya adalah untuk melihat tingkat keberhasilan hasil klasifikasi kNN dengan

clustering pada tiap kategori. Pengujian ini utamanya adalah untuk mengetahui tingkat keberhasilan suatu sistem pengklasifikasi teks dalam dokumen berbahasa Arab dengan nilai *k* tertentu yang dimasukan oleh pengguna aplikasi.

Skenario uji coba ini untuk mengetahui tingkat keberhasilan dan waktu yang digunakan untuk mengklasifikasi dokumen *testing* dengan kNN dan *clustering* terlebih dahulu pada kategorinya dengan *clustering*, selain itu untuk mengetahui berapa nilai *k* yang paling optimal dalam proses klasifikasi kNN dan berapa nilai *cluster c* yang paling optimal dalam proses *clustering* kategori.

D. Hasil Uji Coba

Pelaksanaan uji coba terhadap aplikasi, dilakukan sesuai skenario-skenario yang telah ditentukan sebelumnya. Pelaksanaan uji coba menggunakan rumus *Precision*, *Recall*, *F-Measure* dan *Accuration* dengan pendekatan dokumen yang di *retrieve* dan relevan seperti pada Tabel 2 berikut

Tabel 2. Retrieve dan Relevant

	Relevant	Nonrelevant
Retrieved	TP	FP
Not Retrieved	FN	TN

- *Precision* P = tp/(tp + fp)
- *Recall* R = tp/(tp + fn)
- *F-Measure* F = 2*P*r/(P + R)
- *Accuration* A = (tp + tn) / (tp + fp+ fn + tn)

Hasil uji coba dengan *cluster c* pada proses clustering mulai dari *c*=2 sampai *c*=7 dan *k* pada kNN mulai dari *k*=2 sampai *k*=12 dapat disimpulkan bahwa nilai *F-Measure* Seni terbaik dengan nilai *F*=0.8611 terdapat pada c=5 dan k=11. Nilai masing-masing *recall*, *precision*, *F-Measure* dan *Accuration* dapat dilihat pada Tabel-Tabel berikut:

Tabel 3. Nilai *Recall*

	k										
Kategori	2	3	4	5	6	7	8	9	10	11	12
Seni	0,765	0,769	0,882	0,838	0,882	0,882	0,909	0,909	92,903	0,912	0,912
Budaya	0,750	-	-	0,667	-	-	-	-	-	-	-
Ekonomi	0,333	0,546	0,462	0,455	0,417	0,556	0,546	0,455	91,613	0,556	0,556
Internasional	0,583	0,500	0,500	0,500	0,800	0,750	0,800	0,750	88,387	0,750	0,800
Lokal	0,500	0,522	0,540	0,556	0,544	0,521	0,544	0,544	77,419	0,536	0,529

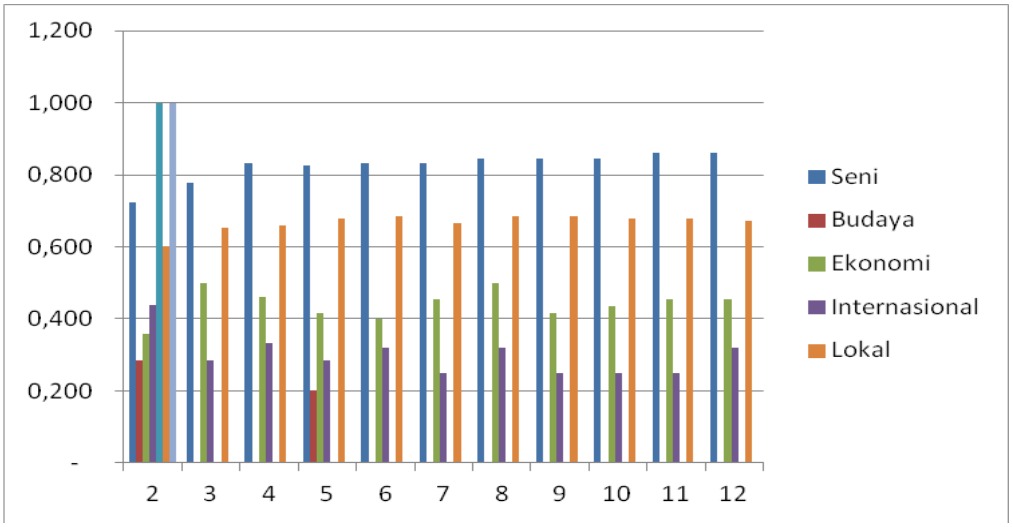
Tabel 4. Nilai *Precision*

	k										
Kategori	2	3	4	5	6	7	8	9	10	11	12
Seni	0,722	0,779	0,833	0,827	0,833	0,833	0,845	0,845	0,845	0,861	0,860
Budaya	0,286	-	-	0,200	-	-	-	-	-	-	-
Ekonomi	0,357	0,500	0,462	0,417	0,400	0,455	0,500	0,417	0,435	0,455	0,455
Internasional	0,438	0,286	0,333	0,286	0,320	0,250	0,320	0,250	0,250	0,250	0,320
Lokal	0,600	0,654	0,660	0,680	0,685	0,667	0,685	0,685	0,679	0,679	0,673

Nilai *precision* yang didapatkan pada Tabel 4, dari *cluster c*=5 dan tetangga terdekat dimulai dai *k*=2 sampai *k*=12. Menunjukan bahwa nilai *precision* yang kecil terdapat pada kategori budaya

Tabel 5. *Nilai F-Measure* c=5 dan k=11

Kategori	k										
	2	3	4	5	6	7	8	9	10	11	12
Seni	0,684	0,790	0,790	0,816	0,790	0,790	0,790	0,790	0,790	0,816	0,816
Budaya	0,177	0,118	0,177	0,118	0,059	0,118	0,059	0,059	0,059	0,059	0,059
Ekonomi	0,385	0,462	0,462	0,385	0,385	0,385	0,462	0,385	0,385	0,385	0,385
Internasional	0,350	0,200	0,250	0,200	0,200	0,150	0,200	0,150	0,150	0,150	0,200
Lokal	0,750	0,875	0,850	0,875	0,925	0,925	0,925	0,925	0,925	0,925	0,925

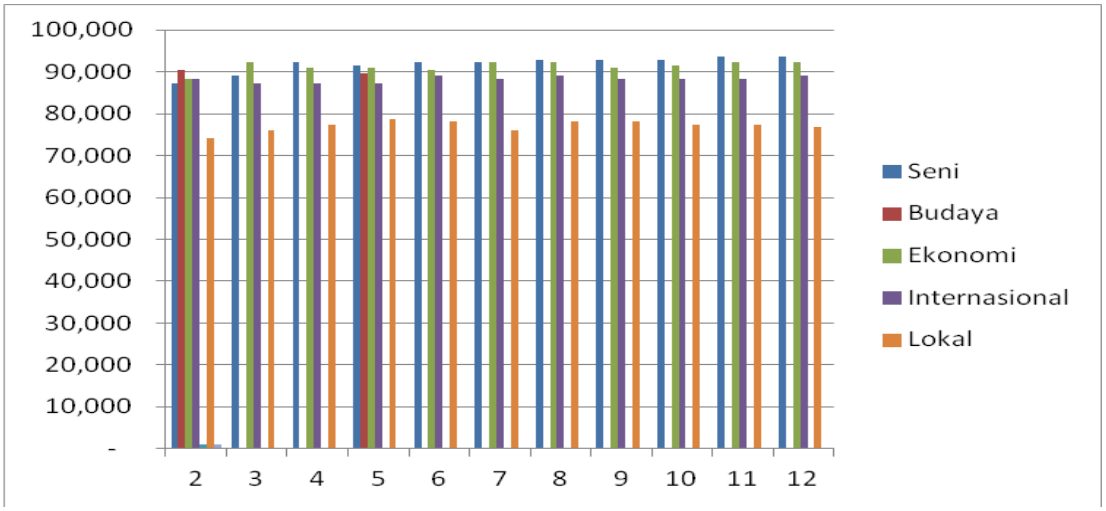


Gambar 6. Grafik *F-measure* pada c=5 dan k=11

Tabel 6. Nilai *accuration*

Kategori	2	3	4	5	6	7	8	9	10	11	12
Seni	87,097	89,032	92,258	91,613	92,258	92,258	92,903	92,903	92,903	93,548	93,541
Budaya	90,323	-	-	89,677	-	-	-	-	-	-	-
Ekonomi	88,387	92,258	90,968	90,968	90,323	92,258	92,258	90,968	91,613	92,258	92,258
Internasional	88,387	87,097	87,097	87,097	89,032	88,387	89,032	88,387	88,387	88,387	89,032
Lokal	74,194	76,129	77,419	78,710	78,065	76,129	78,065	78,065	77,419	77,419	76,774

Nilai F-Measure dapat dilihat pada Tabel 5 dan grafiknya pada Gambar 6, dari data yang diperoleh, tingkat keberhasilan suatu sistem untuk mengklasifikasi dokumen dimana k=11 dan c=5 didapatkan nilai lokal optomal F-Measure terbaik sebesar 0.862 dan tingkat akurasi dapat dilihat pada Tabel 6 dan grafiknya pada Gambar 7 dengan akurasi sebesar 93.548% atau 94 %



Gambar 7. Grafik Accuration pada c=5 dan k=11

4. KESIMPULAN

Berdasarkan aplikasi yang telah dibuat dan hasil uji coba yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma kNN dapat diaplikasikan pada teks berbahasa Arab dengan *F-Measure* terbaik sebesar 0.8611 dengan tingkat akurasi 94%.
2. Nilai *k* lokal optimal pada kNN ketika *k* bernilai 11 dan *c* lokal optimal pada *Clustering* ketika *c* bernilai 5.
3. Aplikasi yang dibangun berbasis algoritma *KNN* dapat digunakan untuk mengklasifikasikan dokumen berbahasa Arab dan menghasilkan tingkat akurasi yang tinggi.

4. SARAN

1. Pemilihan nilai *k* secara otomatis pada kNN agar menghasilkan *k* yang optimal.
2. Pemilihan *cluster c* secara otomatis pada *clustering* agar menghasilkan *c* yang optimal.
3. Perlunya dikembangkan suatu sistem untuk perbaikan hasil pembacaan secara otomatis yang dapat diintegrasikan dengan sistem ini sehingga dapat menghasilkan tingkat akurasi klasifikasi yang lebih baik.
4. Pada pemilihan fitur penelitian ini menggunakan metode term frequency, untuk penelitian selanjutnya dapat menggunakan metode lainnya seperti metode chi-square, expected cross entropy, odds ratio, the weight of evidence of text dan sebagainya

DAFTAR PUSTAKA

[1] Hatta, H.R. 2013, *Metode Hibridasi Ant Colony Optimization dan Information Gain untuk seleksi fitur pada kategorisasi dokument text arab*, Teknik Informatika, Institut Teknologi Sepuluh Nopember Surabaya. vol. 8 2014 ISSN : 1978-0087.

[2] Sinha, K.B, Sinhal, A. & Verma, B. 2013. Software Measurement Using Artificial Neural Network and Support Vector Machine. *International Journal of Software Engineering & Applications (IJSEA), Vol.4, No.4.*

[3] Krisandi, H. d. 2013. *Algoritma k-Nearest Neighbor dalam Klasifikasi Data Hasil Produksi Kelapa Sawit Pada PT.MINAMAS.* Pontianak : Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster) Volume 02, No.1(2013), hal. 33-38, Pontianak.

[4] Rozaq, A,. 2012. Klasifikasi Dokumen Teks Berbahasa Arab Menggunakan Algoritma Naïve Bayes, *Skripsi,* Teknik Informatika, Institut Teknologi Sepuluh Nopember, Surabaya.

[5] Hardjianto, M., & Winarko, E. 2013. Penentuan Kesehatan Lansia Berdasarkan Multi Variabel Dengan Algoritma K-NN Pada Rumah Cerdas. *Prosiding Seminar Nasional Informatika, Vol. 7, Hal. 214-218.*
[6] Prasetyo, E. 2012. Data Mining konsep dan aplikasi menggunakan matlab. Yogyakarta: Andi.
[7] Widodo, P.P., dkk. 2013. *Penerapan Data Mining dengan MATLAB*. Bandung: Rekayasa Sains.
[8] Efendi, R., Malik, R.F,. Mila, J,. 2012, Klasifikasi Dokumen Berbahasa Indonesia Menggunakan Naive Bayes Classifier, *Journal of Research in Computer Science and Applications–Vol. I, No. I.*
[9] Leidiyana, H,. 2013. Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bemotor. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic (J Piksel), Vol. 1, No. 1.*
[10] Wijaya, A.P,. & Agus Santoso, H.A. 2016. Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-. *Journal of Applied Intelligent System, Vol.1, No. 1, Februari 2016: 48-55.*
[11] Yerpude, P., Jakhotiya, R. & Chandak, M. 2015. Algorithm For Text To Graph Conversion And Summarizing Using. *International Journal on Natural Language Computing (IJNLC) Vol. 4, No.4,*
[12] Shah , N., & Mahajan, S. 2012. "Document Clustering: A Detailed Review". *International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868Foundation of Computer Science FCS, New York, USAVolume 4 No.5*
[13] Yu, Q. 2013. *Machine Learning for Corporate Bankruptcy Prediction.* Finland: School of Science Aalto University.
[14] Zainuddin, S., dkk. (2013). Penerapan Algoritma Modified K-Nearest Neighbour (M-KNN) Pada Pengklasifikasian Penyakit Tanaman Kedelai. *Skripsi*, Malang:Universitas Brawijaya
[15] Ndaumanu, R.I,. Kusrini, M. Rudyanto, A,. 2014, Analisis Prediksi Tingkat Pengunduran Diri Mahasiswa dengan Metode K-Nearest Neighbor, *Jatisi, Vol. 1 No. 1*
[16] Ridok, A. 2011. Penerapan Algoritma Improved K-Nearest Neighbors Untuk Pengkategorian Dokumen Teks Berita Berbahasa Indonesia. *Skripsi*, Malang: Universitas Brawijaya.
[17] Samuel, Y., Delima, R,. Rachmat, A,. 2014, Implementasi Metode K-Nearest Neighbor dengan Decision Rule untuk Klasifikasi Subtopik Berita, Program Studi Teknik Informatika Universitas Kristen Duta Wacana, Yogyakarta *Jurnal Informatika, Vol. 10 No. 1.*