Otomatisasi Klasifikasi Buku Perpustakaan dengan Menggabungkan Metode *K-NN* dengan *K-Medoids*

Ni Nyoman Emang Smrti
Sistem Informasi, STMIK Bandung, Bali
e-mail:smrti_nyoman@yahoo.com

Abstrak

Klasifikasi buku perpustakaan sangatlah penting untuk memudahkan pengunjung dalam pencarian buku. Dengan memanfaatkan metode yang ada pada data mining khususnya text mining, maka dalam penelitian ini akan dibangun program aplikasi untuk otomatisasi klasifikasi buku perpustakaan. Metode yang akan digunakan untuk mengklasifikasi buku perpustakaan adalah metode k-nearest neighborhood (*K-NN*) *digabungkan dengan metodek-medoids*. *Program aplikasi otomatisasi klasifikasi buku perpustakaan ini dibangun dengan data latih dari buku perpustakaan STMIK Bandung Bali dan data uji berasal dari beberapa toko buku online.Aplikasi yang dibuat mampu mengklasifikasi buku perpustakaan dengan prosentase keberhasilan 84% dengan jumlah data latih 507 dan 50 data uji.*

Kata kunci: *klasifikasi*, *text mining, k-nearest neighborhood, k-medoids.*

Abstract

Classification oflibrary'sbooksis an important effort tofacilitate visitorsin searching ofthe books. By using theexisting methodsindata mining, text miningin particular, it was constructedan automaticclassificationapplicationof library's books. The methodswereutilizedto classifylibrarybooksarek-nearest neighborhood(K-NN) by combining withk-medoids. This applicationwas constructedwith training datafrom library of STMIKBandung Bali. Testing datacome from severalonlinebookstores. The results showed that the applicationiscapable ofclassifyingthe library'sbooksby84%of successusing 507 trainingdata and 50testingdata.

Keywords:*classification, text mining, k-nearst neighbor, k-medoids*

1. Pendahuluan

Perpustakaan adalah institusi yang menyediakan koleksi bahan pustaka tertulis, tercetak dan terekam sebagai pusat sumber informasi yang diatur menurut sistem aturan dan didayagunakan untuk keperluan pendidikan, penelitian serta rekreasi intelektual bagi masyarakat.Perpustakaan berperan melakukan layanan informasi literal kepada masyarakat.Karena tujuannya memberikan layanan informasi literal kepada masyarakat maka tugas pokoknya adalah: (1) menghimpun bahan pustaka yang meliputi buku dan nonbuku sebagai sumber informasi, (2) mengelola dan merawat pustaka, (3) memberikan layanan bahan pustaka [1].

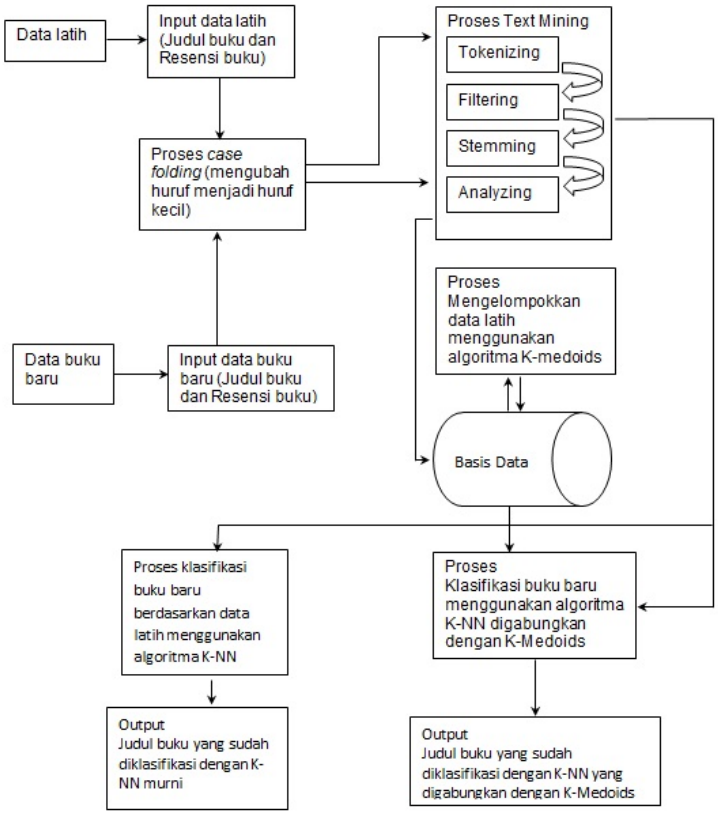
Klasifikasi adalah pengelompokan yang sistematis mengenai objek, gagasan, buku atau benda-benda lain ke dalam kelas atau golongan tertentu berdasarkan ciri-ciri yang sama. Klasifikasi buku perpustakaan yang paling banyak dipakai adalah penggolongan berdasarkan isi atau subjek buku dengan menggunakan metode klasifikasi peresepuluh *dewey*. Aturan klasifikasi buku perpustakaan *DDC* (*dewey decimal classification*) atau disebut dengan persepuluh dewey, pertama-tama membagi ilmu pengetahuan ke dalam 10 kelas utama. Kemudian masing-masing kelas utama itu dibagi lagi ke dalam 10 divisi dan selanjutnya masing-masing divisi dibagi lagi ke dalam 10 seksi, sehingga dengan demikian *DDC* (*dewey decimal classification*) terdiri dari 10 kelas utama, 100 divisi dan 1000 seksi. Meskipun demikian, *DDC* masih memungkinkan diadakannya pembagian lebih lanjut dari seksi menjadi sub-seksi, dari sub-seksi menjadi sub-sub-seksi dan seterusnya.Pola perincian ilmu pengetahuan yang berdasarkan kelipatan sepuluh inilah maka *DDC* disebut klasifikasi persepuluh atau klasifikasi decimal [2].Banyak metode yang

mendukung *text mining* salah satunya adalah algoritma *k-nearest neighbor* (*K-NN*).Algoritma *K-NN* berdasarkan *survey paper* tahun 2006 termasuk dalam 10 algoritma terpopuler dalam *data mining* [3].

Penelitian untuk proses klasifikasi dengan menggunakan algoritma *K-NN* tradisional dan dioptimalkan metode *k-means* telah dilakukan oleh Zhou Yong, dkk, yang pada intinya proses klasifikasi dengan metode *K-NN* yang besarnya jumlah sampel pelatihan akan meningkatkan kompleksitas perhitungan dan sementara satu klasifikasi memiliki kemiripan ciri, maka dengan menggunakan algoritma *clustering*, pengujian tidak dilakukan pada keseluruhan data latih. Dari masalah tersebut klasifikasi teks dengan menggunakan *K-NN*akan ditingkatkan dengan menggunakan algoritma *clusteringk-means* [4]. *K-medoid*s lebih kuat terhadap *noise* dibandingkan dengan *k-means* karena meminimalkan jumlah dari ketidaksamaan bukannya meminimalkan jumlah kuadrat jarak *Euclidean* [5].Berdasarkan penelitian terdahulu tentang *text mining* yang telah dipublikasikan, serta mempertimbangkan kelemahan dan kelebihan dari metode *text mining* yang telah digunakan oleh para peneliti terdahulu, maka dalam penelitian ini akan menggunakan metode *K-NN* dan digabungkan dengan menggunakan metode *clusteringk-medoids*.

2. Metodelogi Penelitian

Penelitian ini dilaksanakan di Perpustakaan STMIK Bandung Bali dengan jumlah buku yang berbahasa Indonesia adalah 507 buah judul. Buku-buku yang telah menjadi koleksi perpustakaan STMIK Bandung Bali akan dijadikan sebagai data latih.



Gambar 1. Gambaran umum sistem

2.1 Data

Koleksi buku pada perpustakaan STMIK Bandung Bali diklasifikasikan dengan menggunakan DCC (*deweydecimal classification*). Data uji diperoleh dari toko buku *online* yaitu

gramediaonline.com, bukukita.com dan belbuk.com.Tahapan secara langkap program aplikasi otomatisasi klasifikasi buku perpustakaan dapat dilihat pada Gambar 1 tentang gambaran umum sistem.

2.2 Tahapan Penelitian

Sesuai dengan gambaran umum dari sistem yang akan dibuat dalam penelitian ini, tahapannya dapat dirinci sebagai berikut:

1. Masukkan data latih yaitu judul dan sinopsis buku perpustakaan yang telah diklasifikasikan ke kategori tertentu sesuai dengan isi buku.
2. *Case folding* adalah mengubah semua huruf dalam dokumen menjadi huruf kecil.Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima.Karakter selain huruf dihilangkan dan dianggap *delimiter*.
3. Tahap *text mining* terdiri dari
   a. *Tokenizing/parsing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya.
   b. Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *token*.Algoritma yang digunakan bisanya adalah *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).
   c. *Tagging* adalah tahap mencari bentuk awal/*root* dari tiap kata hasil *stemming* berdasarkan hasil dari tahap *filtering.*
   d. Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterkaitan antar kata-kata dari dokumen yang ada.Tahap ini menghitung keterkaitankata-kata yang terdapat dalam judul danringkasan dibandingkan dengan kata kunci.Kata kunci disini adalah kata-kata yangsering muncul dalam satu kategori buku.

      Berikut akan disajikan proses *text mining* yang diawali dengan menyajikan data buku dapat dilihat pada Tabel 1.

Tabel 1. Data buku

Dokumen	Term yang mewakili dokumen
D1	Kamus Umum Lengkap
D2	Kamus Indonesia Inggris
D3	Kamus Lengkap Inggris-Indonesia &Indonesia Inggris
D4	Kamus Besar Bahasa Indonesia Edisi 3
D5	Apelatif Cara Praktis Temukan 1100 Entri Istilah Pengetahuan

Data buku yang disajikan pada Tabel 1 akan dilakukan proses perhitungan *tf* (*term frequency*) banyaknya kata yang muncul di masing-masing dokumen (D1 sampai dengan D5). Hasil perhitungan *tf* disajikan pada Tabel 2 di bawah ini.

Tabel 2. Hasil perhitungan *tf*

Kata	Tf				
	D1	D2	D3	D4	D5
Apelatif	0	0	0	0	1
Bahasa	0	0	0	1	0
Besar	0	0	0	1	0
Cara	0	0	0	0	1
Entri	0	0	0	0	1
Indonesia	0	1	2	1	0
Inggris	0	1	2	0	0
Istilah	0	0	0	0	1
Kamus	1	1	1	1	0
Lengkap	1	0	1	0	0
Tahu	0	0	0	0	1
Praktis	0	0	0	0	1
Temu	0	0	0	0	1
Umum	1	0	0	0	0

Dari Tabel 2 dapat dilihat bahwa kata “apelatif” hanya muncul pada dokumen 5 (D5) saja, “bahasa” hanya muncul pada dokumen 4 (D4) sampai dengan kata “umum” hanya muncul pada dokumen 1 (D1) saja. Perhitungan selanjutnya adalah *df*(*documentfrequency*) diperoleh dari menghitung total kata yang muncul pada seluruh dokumen. Lihat Tabel 3 berikut ini kata “Apelatif” hanya terdapat pada dokumen 5, jadi nilai *df* = 1.Hasil perhitungan *df*secara lengkap dapat dilihat pada Tabel 3 berikut ini.

Tabel 3. Hasil perhitungan *df* dan *idf*

Kata	*Tf*						*idf*
	D1	*D2*	*D3*	*D4*	*D5*	*Df*	= log(n/df)
apelatif	0	0	0	0	1	1	0,69897
bahasa	0	0	0	1	0	1	0,69897
besar	0	0	0	1	0	1	0,69897
cara	0	0	0	0	1	1	0,69897
entri	0	0	0	0	1	1	0,69897
indonesia	0	1	2	1	0	4	0,09691
inggris	0	1	2	0	0	3	0,22185
istilah	0	0	0	0	1	1	0,69897
kamus	1	1	1	1	0	4	0,09691
lengkap	1	0	1	0	0	2	0,39794
tahu	0	0	0	0	1	1	0,69897
praktis	0	0	0	0	1	1	0,69897
temu	0	0	0	0	1	1	0,69897
umum	1	0	0	0	0	1	0,69897

Dari Tabel 3 dapat dilihat hasil perhitungan *df* dari setiap kata dan pada kolom terakhir merupakan perhitungan *idf* . Contoh perhitungan *idf*dapat dilihat dari persamaan berikut ini. Kata “aplatif” hanya terdapat pada dokumen 5 maka:

nilai*df = 1 maka nilai idf = log(n/df) = log(5/1) = 0,69897*

Setelah didapatkan nilai *df*, perhitungan selanjutnya adalah menghitung bobot. Kata “apelatif” pada masing-masing dokumen dapat dihitung sebagai berikut:

W untuk dokumen 5 = 1 x *0,69897* = *0,69897*

Untuk hasil perhitungan secara lengkap dapat dilihat pada Tabel 4 berikut ini.

Tabel 4. Hasil perhitungan bobot (*W*)

Kata	*idf* =	*W= tf x idf*				
	log(n/df)	*D1*	*D2*	*D3*	*D4*	*D5*
Apelatif	0,69897	0	0	0	0	0,69897
Bahasa	0,69897	0	0	0	0,69897	0
Besar	0,69897	0	0	0	0,69897	0
Cara	0,69897	0	0	0	0	0,69897
Entri	0,69897	0	0	0	0	0,69897
indonesia	0,09691	0	0,09691	0,19382	0,09691	0
Inggris	0,22185	0	0,221849	0,443697	0	0
Istilah	0,69897	0	0	0	0	0,69897
kamus	0,09691	0,09691	0,09691	0,09691	0,09691	0
lengkap	0,39794	0,39794	0	0,39794	0	0
tahu	0,69897	0	0	0	0	0,69897
praktis	0,69897	0	0	0	0	0,69897
temu	0,69897	0	0	0	0	0,69897
umum	0,69897	0,69897	0	0	0	0

4. Buku perpustakaan telah diklasifikasi secara manual akan dijadikan data latih. Data latih yang telah melalui tiga tahap di atas, disetiapklasifikasinya akan dikelompokan dengan menggunakan metode *k-medoids*. *Medoids* yang didapatkan akan disimpan di dalam basis data. *Medoids* ini nantinya akan dibandingkan dengan data uji.

5. Langkah berikutnya adalah masukkan data buku baru sebagai data uji.Data buku baru juga harus melalui tahap *case folding* dan *text mining* seperti pada data latih yaitu di tahap ke-2 dan ke-3.
6. Langkah berikutnya adalah menentukan klasifikasi buku baru yang akan menjadi koleksi perpustakaan dengan menggunakan algoritma *K-NN*. Ada sebuah uji coba yang menarik dari penggunaan algoritma *K-NN* yang biasanya harus membandingkan semua data latih dengan data baru, namun disini berdasarkan hasil dari langkah ke-5, maka perbandingan hanya dilakukan pada *medoids* yang dihasilkan dari algoritma *clustering*.Penjelasan mengenai algoritma *K-NN* adalah sebagai berikut.

   Misalkan terdapat *j* kategori latih *C1, C2,...,Cj* dan jumlah sampel latih *N*. Setelah*preprosesing*, masing-masing dokumen akan menjadi vektor fitur berdimensi *m*. Selanjutnya langkah-langkah untuk penerapan metode ini adalah sebagai berikut :

   a. Membuat dokumen *X* dari semua sampel latih menjadi bentuk vektor fitur yang sama ($X_1, X_2, \ldots X_m$).
   b. Hitung kesamaan antara semua sampel latih dan dokumen *X*. Ambil dokumen ke *i* di ($di_1, di_2, \ldots, di_m$) sebagai contoh, kesamaan SIM (*X,di*) adalah sebagai berikut:

$$sim(X_i, d_i) = \frac{\sum_{j=1}^{m} X_j d_y}{\sqrt{\left(\sum_{j=1}^{m} X_j\right)^2}\sqrt{\left(\sum_{j=1}^{m} d_{ij}\right)^2}} \quad (1)$$

   c. Memilih k sampel yang lebih besar dari kesamaan N dari SIM (*X,di*), (*i = 1,2,...,N*). Dan memperlakukannya sebagai kumpulan *K-NN* dari *X*. Kemudian hitung probabilitas *X* ke masing-masing kategori menggunakan Persamaan 2 berikut:

$$P(X, C_j) = \sum_{d_i \in KNN} SIM(X, d_i) \cdot y(d_i, C_j) \quad (2)$$

   Dimana, $y(d_i, Cj)$ adalah fungsi *attribute* kategori yang memenuhi Persamaan 1.

$$y(d_i, C_j) = \begin{cases} 1, d_i \in C_j \\ 0, d_i \notin C_j \end{cases} \quad (3)$$

   d. Uji dokumen *X* untuk mengetahui kategorinya dengan melihat *P(X,Cj)* terbesar.
7. Tahap terakhir adalah tahap pengujian yang akan memberikan kategori pada data tes dengan menggunakan model yang telah dibangun pada tahap memasukkan data latih. Tahap pengujian ini dilakukan dua kali yang pertama pengujian data tes menggunakan metode *K-NN* murni dan yang kedua menggunakan metode *K-NN* yang digabungkan dengan metode *k-medoids*.

3. Kajian Pustaka

3.1 Preprosesing Dokumen

Sebelum proses klasifikasi dilakukan dengan menggunakan metode *K-NN* digabungkan dengan metode *k-medoids*, maka data latih maupun data uji yang berupa judul buku diolah terlebih dahulu menjadi data numerik. Tahapan *preprocessing* ini merupakan tahapan dari *text mining* yang harus dilakukan, bila akan menambang informasi berupa teks.*Text mining* merupakan menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen [6].

Text mining merupakan proses mengesktrak *petterns* dan *knowledge* yang bersifat menarik dan penting dari dokumen-dokumen teks. Pada intinya proses kerja *text mining* sama dengan proses kerja *data mining* pada umumnya hanya saja data yang di-*mining* merupakan *text databases* [7].Di dalam *knowledge discovery* terdapat tahap *data mining* seperti yang telah

disebutkan diatas sebenarnya pada tahap *data mining* inilah *text mining* dijalankan.Jadi pada intinya *text mining* adalah istilah yang dipakai oleh *data mining* yang mengekstrak data berupa teks.Tahap-tahap *text mining* secara umum adalah:

1. *Tahap tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya.
2. Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *token*.Algoritma yang digunakan adalah algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).
3. Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering.* Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia.
4. Tahap *tagging* adalah tahap mencari bentuk awal/*root* dari tiap kata hasil *stemming*.
5. Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antara kata-kata antar dokumen yang ada. Tahap ini menggunakan algoritma *termfrequency*(*tf*), *invers document frequency* (*idf*) dan kombinasi perkalian antara keduanya (*tfxidf*).

3.2 Algoritma *Porter*

Algoritma *Porter* adalah algoritma *stemming* untuk Bahasa Inggris yang ditemukan oleh *Martin Porter* 1980. Cara kerja algoritma ini adalah dengan membuang imbuhan (dalam Bahasa Inggris akhiran).Berdasarkan algoritma *Porter*, pada penelitian *Fadillah Tala* yang berjudul "*A Study of Stemming Effects on Information Retrieval in* Bahasa Indonesia*Stemming*"mengadopsi cara kerja algoritma *Porter* yang disesuaikan dengan karakteristik Bahasa Indonesia. Langkah-langkah algoritma *Porter*.adalah sebagai berikut[8]:

1. Hapus *Particle*.
2. Hapus *Possesive Pronoun*.
3. Hapus awalan pertama.Jika tidak ada lanjutkan ke langkah 4a, jika ada cari maka lanjutkan ke langkah 4b.
4. (a) Hapus awalan kedua, lanjutkan ke langkah 5,
   (b) Hapus akhiran, jika tidak ditemukan maka kata tersebut diasumsikan sebagai *root word*. Jika ditemukan maka lanjutkan ke langkah 5b.
5. (a) Hapus akhiran. Kemudian kata akhir diasumsikan sebagai *root word*,
   (b) hapus awalan kedua. Kemudian kata akhir diasumsikan sebagai *root word*.

3.3 *K-Nearest Neighborhood (K-NN)*

Algoritma *K-NN* merupakan algoritma *supervised learning* di mana hasil klasifikasi data baru berdasar kepada kategori mayoritas tetangga terdekat ke-*k*. Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan data *training*.Algoritma *K-NN* menggunakan klasifikasi ketetanggaan sebagai prediksi terhadap data baru.Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data tes (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran 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, pada umumnya nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai *k* yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma

nearest neighbor.Ketepatan algoritma *K-NN* ini 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.Langkah-langkah algoritma *K-NN*:

1. Tentukan parameter*k*= jumlahtetanggaterdekat.
2. Hitungjarak antaradata yang akan ditentukan klasifikasinya dengansemuasampelpelatihan.
3. Urutkanjarakdan tentukantetangga terdekatberdasarkanjarak minimum*k.*
4. Kumpulkankategoritetanggaterdekat.
5. Gunakanmayoritassederhana darikategoritetangga terdekatsebagai nilaiprediksidari data yang ditentutukan klasifikasinya.

3.4 *K-Medoids*

K-medoids adalah teknik partisi klasik untuk *clustering* yang melakukan *clustering* data dari *n* objek ke dalam *cluster* dikenal dengan *apriori*. *K-medoid*s lebih kuat terhadap *noise* dan *outliner* dibandingkan dengan *k-means* karena meminimalkan jumlah dari ketidaksamaan bukannya meminimalkan jumlah kuadrat jarak *Euclidean*. *Medoids* dapat didefinisikan sebagai *objek cluster*, yang rata-rata perbedaan untuk semua objek dalam suatu *cluster* minimal yaitu merupakan titik paling pusat dari data yang diberikan.

Realisasi yang paling umum dari *clustering k-medoids* adalah *partition around medoids* (PAM) dan algoritma adalah sebagai berikut:

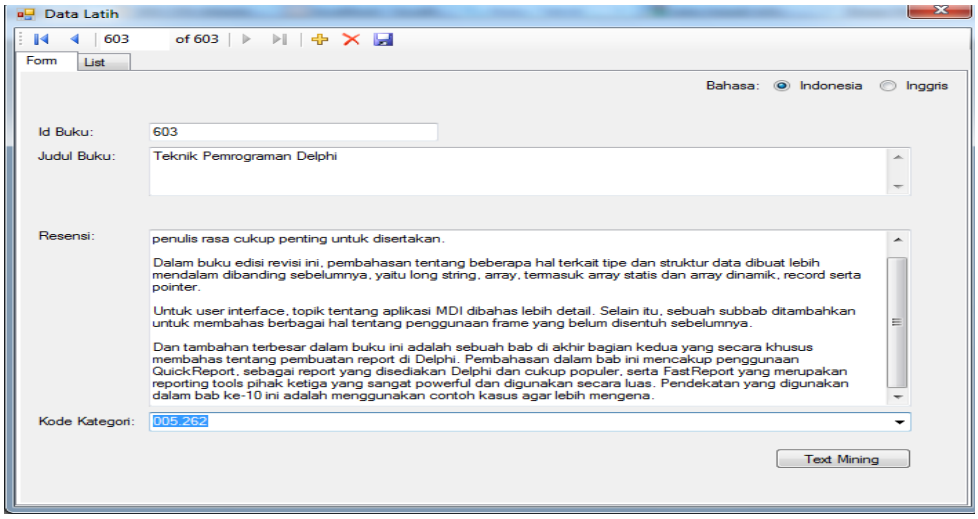
1. Inisialisasi: pilih secara acak *k* dari *n* data *point* sebagai*medoids.*
2. Asosiasikan setiap data *point* ke *medoids* yang terdekat (terdekat berarti menggunakan perhitungan jarak yang biasa digunakan adalah *Euclidean distance, Manhattan distance* atau *Minkowski distance*)
3. Untuk setiap *medoidsm* dan untuk setiap data non *medoidso*
   Tukarkan m dan *o* dan hitung berapa total*cost* dari setiap konfigurasi (penukaran *m* dan *o*)
4. Pilih konfigurasi dengan *cost* paling sedikit.
5. Ulangi langkah 2 sampai 5 dan hentikan jika sudah tidak terdapat perubahan *medoids*.

4. Hasil dan Pembahasan

4.1 Uji Coba

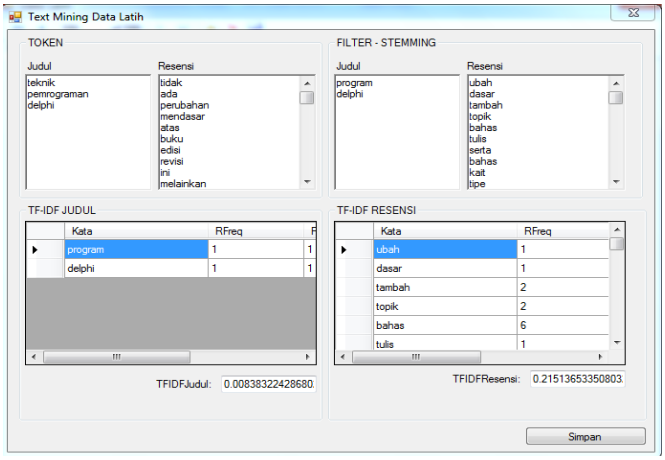
Tahapan uji coba aplikasi otomatisasi klasifikasi buku perpustakaan ini, seperti yang telihat pada Gambar 1 yaitu gambaran umum sistem terdiri dari 13 tahapan. Tahapan uji coba tersebut akan dijelaskan berikut ini:

1. *Input* data latih
   Tahap ini adalah memasukkan data buku koleksi perpustakaan STMIK Bandung Bali yang telah diklasifikasi sesuai dengan judul buku tersebut. Implementasi dari tahap *input* data latih dapat dilihat pada Gambar 2.Antramuka input data latih yang terlihat pada Gambar 3 di atas memasukkan judul buku “Teknik Pemrograman Delphi”. Setelah seluruh *field* terisi pada pojok kanan bawah terdapat tombol “*Text Mining*” yang berfungsi untuk melanjutkan tahapan *text mining* dari data latih.



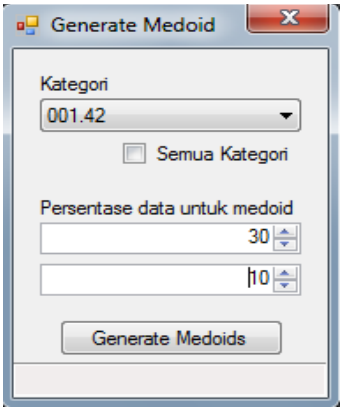
Gambar 2. Input data latih

2. *Case folding*
   Tahapan yang kedua yaitu merubah *field* judul dan resensi yang telah dimasukkan menjadi huruf kecil. Tahapan ini pada implementasi digabungkan dengan tahapan "*Text Mining*" pada proses *token.*
3. *Text mining*
   Proses *text mining* dari *token* sampai dengan *Analyzing* dapat dilihat implementasinya pada Gambar 3 berikut ini:



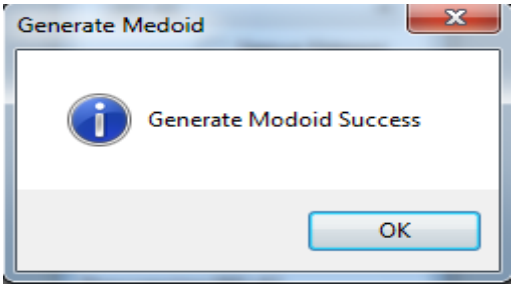
Gambar 3. Hasil *text mining*data uji

4. Tahap ke 4 adalah menyimpan hasil proses *text mining* ke dalam basis data, pada Gambar 3 dapat dilihat terdapat fasilitas untuk menyimpan dengan meng-klik *button* "Simpan".
5. Tahap ke 5 mengambil data latih yang telah tersimpan di dalam basis data, kemudian setiap klasifikasi dari data buku tersebut dilakukan proses *clustering*. Antar muka proses *clustering* dapat dilihat pada Gambar 4.
   Proses *clustering* seperti terlihat pada Gambar 4 terdapat *field* "Kategori", disini dilakukan pemilihan kategori yang akan dilakukan proses *clustering*. Pada Gambar 4 terlihat proses *clustering* untuk kode kategori "001.42". Dibawah field "Kategori" terdapat *check list* "semua kategori", apabila ini dipilih, maka proses klasifikasi dilakukan pada seluruh kategori yang telah dimasukkan ke dalam basis data. *Field* "presentase untuk *medoids*" digunakan untuk menetukan berapa persen dari data latih digunakan sebagai *medoids*.



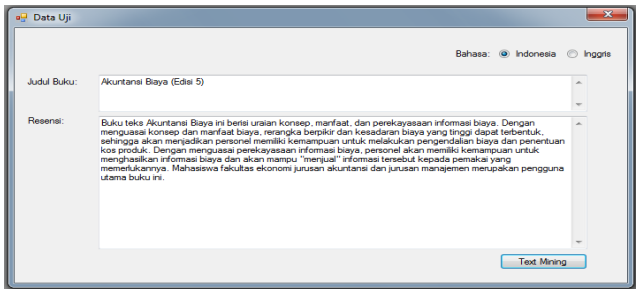
Gambar 4. Proses *clustering*

6. Tahap ke 6 adalah menyimpan hasil proses *clustering*. Pada Gambar 4 terlihat *button* "*Generate Medoids*" yang berfungsi melakukan proses *clustering* sekaligus menyimpan ke dalam basis data. Apabila proses *clustering* telah selesai, maka akan tampil pesan bahwa proses *clustering* telah sukses dilakukan seperti terlihat pada Gambar 5 berikut ini:
7.



Gambar 5. Proses *clustering* telah sukses dilakukan

8. Tahap ke 7 merupakan proses uji coba klasifikasi buku terhadap data latih yang telah dimasukkan ke dalam basis data. Tahap uji coba ini diawali dengan memasukkan data buku yang akan diklasifikasi. Antramuka untuk memasukkan data buku untuk uji coba dapat dilihat pada Gambar 6 berikut ini.



Gambar 6. Input data uji

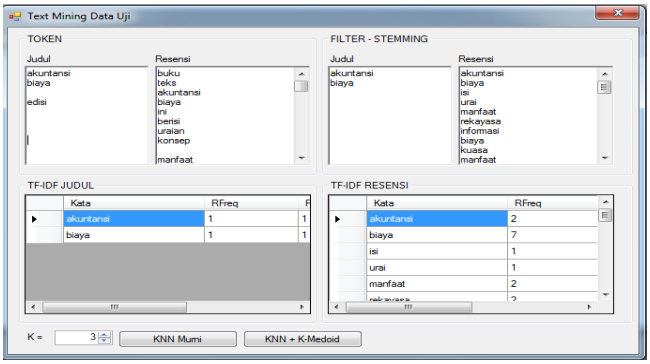
Pada Gambar 6 terlihat telah dimasukkan data buku yang berjudul "Akuntansi Biaya (Edisi 5)" dan untuk melanjutkan ke tahap *case folding*, maka langkah yang dilakukan adalah dengan mengklik *button* "Text Mining" yang terdapat pada pojok kanan bawah.

9. *Case folding*
   Tahapan kedelapan sama dengan tahap kedua, hanya saja tahapan ini merubah *field* judul dan resensi yang telah dimasukkan menjadi huruf kecil untuk data buku sebagai data uji.

Tahapan ini pada implementasi digabungkan dengan tahapan "*Text Mining*" pada proses *token.*

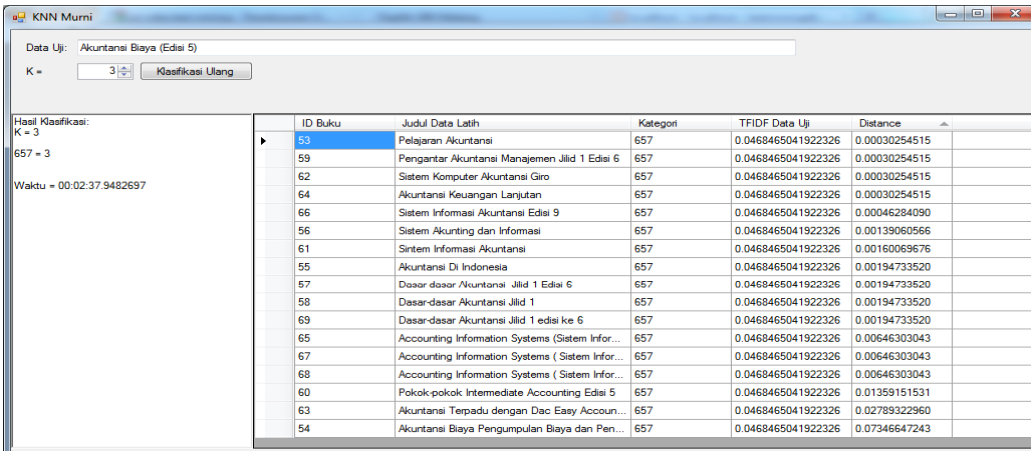
10. *Text mining*

    Proses *text mining* pada tahap ini sama dengan proses pada tahap ketiga, hanya saja proses *text mining* ini digunakan untuk data buku sebagai data uji. Proses *text mining* mulai dari *token* sampai dengan *Analyzing* dapat dilihat implementasinya pada Gambar 7 berikut ini.



Gambar 7. Proses *text mining* untuk data latih

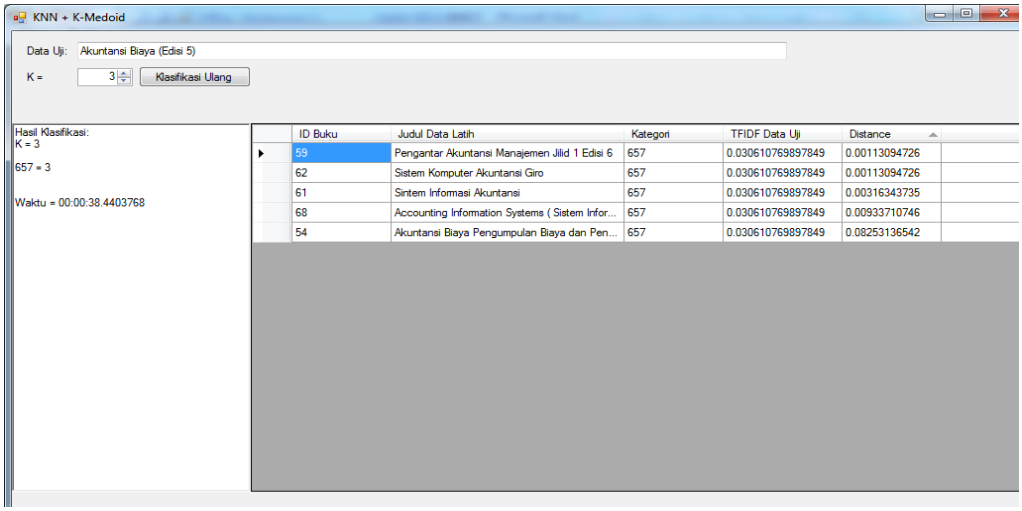
11. Tahap berikutnya adalah mengambil data latih yang tersimpan di dalam basis data, disini ada 2 tahapan yang sedikit berbeda yang pertama adalah mengambil data latih secara keseluruhan dan yang kedua adalah mengambil data latih yang telah di-*cluster*.
12. Tahap ke 11 ini adalah proses klasifikasi. Seperti yang telihat pada Gambar 7 pada bagian bawah terdapat 2 *button* yaitu "*K-NN* Murni" dan "*K-NN+K-medoids*". Apabila *button* "*K-NN* Murni" dipilih, maka proses klasifikasi dengan menggunakan metode *K-NN* sedangkan button "*K-NN+K-medoids*", maka proses klasifikasi dengan menggunakan metode *K-NN* digabungkan dengan *k-medoids*.
13. Tahap ke 12 adalah tahap untuk menampilkan hasil klasifikasi dengan menggunakan metode *K-NN*, implementasinya dilihat pada Gambar 8.



Gambar 8. Hasil klasifikasi dengan metode *K-NN*

Pada Gambar 8 terlihat bahwa judul "Akuntansi Biaya (edisi 5)" diklasifikasi dengan kode 567 yaitu kategori akuntasi dan *k* = 3. Waktu yang diperlukan 2 menit 37 detik. Apabila ingin mengetahui hasil klasifikasi dengan *k* = 4, maka langkah yang dilakukan dengan merubah varibel *k* pada pojok kiri atas dilanjutkan dengan menekan button "Klasifikasi Ulang".

14. Tahap terakhir menampilkan hasil klasifikasi dengan menggunakan metode *K-NN* digabungkan dengan *k-medoids*. Implementasi hasil klasifikasinya dapat dilihat pada Gambar 9 berikut ini.



Gambar 9. Hasil klasifikasi dengan metode *K-NN* digabung dengan *k-medoids*

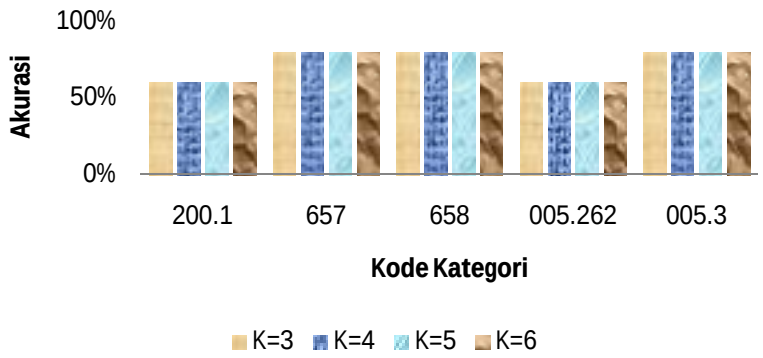
Pada Gambar 9 terlihat hasil klasifikasi dengan menggunakan metode *K-NN* digabung dengan *k-medoids* dengan hasil klasifikasi kode 657 yaitu kategori akuntansi dengan waktu yang diperlukan untuk proses klasifikasi adalah 38 detik.

4.2 Evaluasi

Hasil uji coba pada sub bab 4.2 akan dihitung tingkat akurasinya, guna mengetahui seberapa kedekatan nilai hasil uji dengan nilai sebenarnya. Hasil perhitungan akurasi data uji dengan menggunakan metode*K-NN* dapat dilihat pada Tabel 5 dan Gambar 10 berikut ini.

Tabel 5. Akurasi hasil uji coba dengan metode *K-NN*

Kode Kategori	Akurasi K = 3	Akurasi K = 4	Akurasi K = 5	Akurasi K = 6	Akurasi K = 13
200.1	60 %	60 %	60 %	60 %	60 %
657	80 %	80 %	80 %	80 %	80 %
658	80 %	80 %	80 %	80 %	80 %
005.262	60 %	60 %	60 %	60 %	80 %
005.3	80 %	80 %	80 %	80 %	80 %
Rata-rata	72%	72%	72%	72%	76%



Gambar 10. Grafik akurasi hasil uji dengan menggunakan metode *K-NN*

Dari Gambar 10 dapat dilihat tingkat akurasi dari hasil klasifikasi dengan menggunakan metode *K-NN*, untuk setiap kategori dengan k = 3, k = 4, k = 5 dan k = 6, hasilnya adalah sama. Jadi dapat ditarik kesimpulan nilai *k*sampai dengan*k = 6*, tidak mempengaruhi akurasi.Hasil perhitungan akurasi hasil uji coba dengan menggunakan metoda *K-NN* digabungkan dengan metode *k-medoids* dengan jumlah *medoids* 10% dari data latih dapat dilihat pada Tabel 6 berikut ini.

Tabel 6. Akurasi hasil uji coba metode *K-NN* digabung dengan *k-medoids* dengan *medoids* 30% dari data

Kode Kategori	Akurasi K = 3	Akurasi K = 4	Akurasi K = 5	Akurasi K = 6	Akurasi K = 13
200.1	60 %	60 %	60 %	60 %	50 %
657	20 %	20 %	20 %	20 %	20 %
658	100 %	90 %	90 %	90 %	90 %
005.262	80 %	80 %	80 %	70 %	70 %
005.3	70 %	70 %	70 %	70 %	80 %
Rata-rata	66%	64%	64%	62%	62%

Hasil perhitungan akurasi hasil uji coba dengan menggunakan metoda *K-NN* digabungkan dengan metode *k-medoids* dengan jumlah *medoids* 30% dari data latih ditambah dengan 1 anggota *medoids* yang terjauh dapat dilihat pada Tabel 7 berikut ini.

Tabel 7.Akurasi hasil uji coba metode *K-NN* digabung dengan *k-medoids* dengan*medoids*30% *plus*

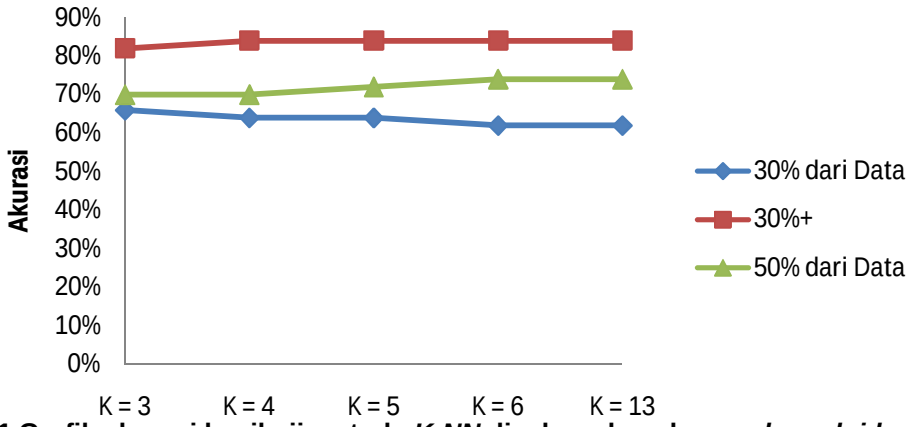
Kode Kategori	Akurasi K = 3	Akurasi K = 4	Akurasi K = 5	Akurasi K = 6	Akurasi K = 13
200.1	60 %	60 %	60 %	60 %	60 %
657	90 %	90 %	90 %	90 %	90 %
658	100 %	100 %	100 %	100 %	100 %
005.262	90 %	90 %	90 %	90 %	90 %
005.3	70 %	80 %	80 %	80 %	80 %
Rata-rata	82%	84%	84%	84%	84%

Hasil perhitungan akurasi hasil uji coba dengan menggunakan metoda *K-NN* digabungkan dengan metode *k-medoids* dengan jumlah *medoids* 50% dari data latih dapat dilihat pada Tabel 10 berikut ini.

Tabel 8. Akurasi hasil uji coba metode *K-NN* digabung dengan *k-medoids* dengan*medoids* 50% dari data

Kode Kategori	Akurasi K = 3	Akurasi K = 4	Akurasi K = 5	Akurasi K = 6	Akurasi K = 13
200.1	50 %	50 %	50 %	50 %	50 %
657	50 %	50 %	50 %	60 %	60 %
658	100 %	100 %	100 %	100 %	100 %
005.262	90 %	90 %	90 %	90 %	90 %
005.3	60 %	60 %	70 %	70 %	70 %
Rata-rata	70%	70%	72%	74%	74%

Hasil perhitungan akurasi pada Tabel 6,7 dan 8 dapat dilihat dengan menggunakan grafik pada Gambar 11 berikut ini.



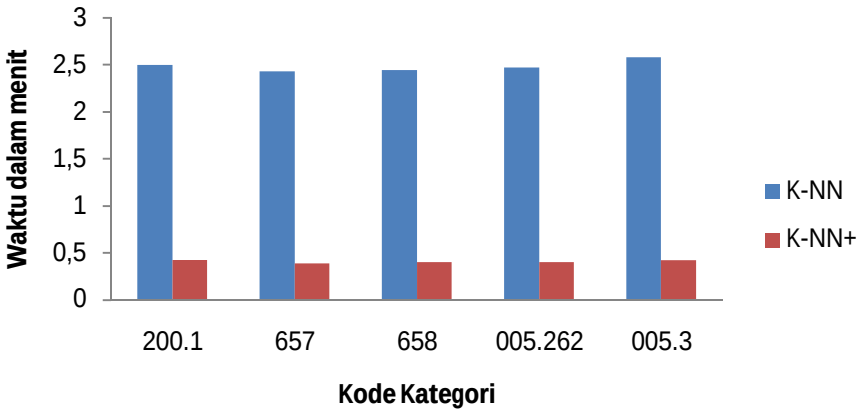
Gambar 11.Grafik akurasi hasil uji metode *K-NN* digabungkan dengan *k-medoids*

Perbandingan waktu klasifikasi untuk metode *K-NN* dengan metode *k-medoids* dapat dilihat pada Tabel 9 dan grafikpada Gambar 12 berikut ini.

Tabel 9. Rata-rata waktu untuk proses klasifikasi

Kode Kategori	Rata-rata Waktu	
	Metode *K-NN*	Metode *K-NN* + *k-medoids*
200.1	2 menit 50 detik	39 detik
657	2 menit 43 detik	38 detik
658	2 menit 44 detik	39 detik
005.262	2 menit 47 detik	40 detik
005.3	2 menit 58 detik	41 detik

Rata-rata waktu untuk proses klasifikasi dengan menggunakan metode *K-NN* lebih lama karena semua data uji harus dibandingkan dengan data latih yang akan diklasifikasi, sedangkan untuk rata-rata waktu proses klasifikasi dari hasil gabungan dua metode yaitu *K-NN* dan *k-medoids* memerlukan waktu 2 menit lebih cepat dari metode *K-NN*, hal ini disebabkan karena data uji hanya dibandingkan dengan data latih yang menjadi *medoids*.



Gambar 12. Grafik Rata-rata Waktu Klasifikasi

5. Simpulan

Berdasarkan hasil uji coba yang telah dilakukan dapat disimpulkan beberapa hal, yaitu: program aplikasi otomatisasi klasifikasi buku perpustakaan berbahasa Indonesia dengan menggunakan metode *K-NN* rata-rata akurasinya 72% dengan jumlah data uji 50 buah dan rata-rata waktu yang diperlukan untuk proses klasifikasi 2 menit 48 detik, bila menggunakan metode *K-*

*NN*digabungkan dengan *k-medoids* rata-rata akurasinya 84% dengan 50 data uji dan waktu yang diperlukan untuk proses klasifikasi 39,4 detik.Klasifikasi dengan menggunakan metode *K-NN* digabungkan dengan *k-medoids* menghasilkan akurasi yang lebih tinggi dan waktu yang lebih singkat dibandingkan hanya dengan menggunakan metode *K-NN*.

Daftar Pustaka

[1] Wahyu Supriyanto,"Ahmad Muhsin, Informasi Perpustakaan", Yogyakarta, Kansius (Anggota IKAPI),2008.
[2] Tawa P. Hamakonda, Mls & J. N. B Tairas, "Pengantar Klasifikasi Persepuluhan Dewey", Cetakan ke – 18. Jakarta,2008.
[3] Xindong Wu, dkk, "Top 10 algorithms in data mining", London, Springer-Verlag,2007.
[4] Zhou Yong,"An Improved K-NN Text Classification Algorithm Based on Clustering",2009. www.academypublisher.com/jcp/vol04/no03/ jcp0403230237.pdf[diunduh: tanggal 5 Mei 2011]
[5] Helmi Harniawati, "Image Clustering Berdasarkan Warna untuk Identifikasi Buah dengan Metode Valley Tracing",Proyek Akhir, Surabaya: Institut Teknologi Sepuluh Nopember, 2007.
[6] Milkha Harlian Ch, Text Mining,2006.http://kesehatankerja.depkes.go.id/downloads/ 6Text%20Mining.pdf[diunduh: tanggal 30 Nopember 2011]
[7] Kusrini, Emha Taufiq Luthfi, "Algoritma Data Mining",Yogyakarta, Andi, 2009.
[8] Fadillah Z. Tala, "A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia, Netherland, Universiteit van Amsterdam, http://ucrel.lancs.ac.uk/acl/P/P00/P00-1075.pdf[diakses: tanggal 25 Juli 2009]