KLASIFIKASI *GENRE* FILM BERDASARKAN JUDUL DAN SINOPSIS MENGGUNAKAN *FUZZY K-NEAREST NEIGHBOUR* (*FUZZY K-NN*)

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Disusun Oleh :
DIMAS PRASETYO ADI SASONO
NIM. 0810960041

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2014

LEMBAR PERSETUJUAN

KLASIFIKASI *GENRE* FILM BERDASARKAN JUDUL DAN SINOPSIS MENGGUNAKAN *FUZZY K-NEAREST NEIGHBOUR* (*FUZZY K-NN*)

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Disusun Oleh :

DIMAS PRASETYO ADI SASONO

NIM. 0810960041

Telah diperiksa dan disetujui oleh :

Pembimbing I,	Pembimbing II,
Lailil Muflikhah, S.Kom., M.Sc	**Drs. Achmad Ridok, M.Kom**
NIP. 19741113 200501 2 001	**NIP. 19680825 199403 1 002**

LEMBAR PENGESAHAN

KLASIFIKASI *GENRE* FILM BERDASARKAN JUDUL DAN SINOPSIS MENGGUNAKAN *FUZZY K-NEAREST NEIGHBOUR* (*FUZZY K-NN*)

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Disusun Oleh:

DIMAS PRASETYO ADI SASONO

NIM. 0810960041

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 7 Januari 2014

Penguji I,

Rekyan Regasari MP, ST., MT.
NIK. 770414 06 1 2 0253

Penguji II,

Edy Santoso, S.Si., M.Kom
NIP. 19740414 200312 1 004

Penguji III,

Muhammad Tanzil Furqon, S.Kom., M.Sc
NIP. 19820930 200801 1 004

Mengetahui,
Ketua Program Studi Informatika / Ilmu Komputer,

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Dimas Prasetyo Adi Sasono

NIM : 0810960041

Program Studi : Informatika / Ilmu Komputer

Penulis skripsi berjudul : Klasifikasi *Genre* Film Berdasarkan Judul Dan Sinopsis Menggunakan *Fuzzy k-Nearest Neighbour* (*Fuzzy k-NN*)

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, Januari 2014

Dimas Prasetyo Adi Sasono

NIM. 0810960041

KATA PENGANTAR

Segala puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya, sehingga penulis dapat menyelesaikan laporan skripsi dengan judul **"Klasifikasi *Genre* Film Berdasarkan Judul Dan Sinopsis Menggunakan *Fuzzy k-Nearest Neighbour* (*Fuzzy k-NN*)"**.

Skripsi ini disusun guna memenuhi syarat menyelesaikan pendidikan program studi Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Sekaligus memperoleh gelar sarjana (strata 1) dalam bidang Ilmu Komputer. Terselesaikannya laporan skripsi ini tentu tidak lepas dari bantuan beberapa pihak, oleh karena itu penulis menyampaikan ucapan terima kasih kepada :

1. Lailil Muflikhah, S.Kom., M.Sc., selaku Dosen Pembimbing I yang telah membimbing, dan menyalurkan ilmu dengan sabar dalam proses pengerjaan dan penyelesaian skripsi ini.
2. Drs. Achmad Ridok, M.Kom., selaku Dosen Pembimbing II yang telah bersedia memberi tambahan pengetahuan, dan arahan yang berkaitan dengan proses pengerjaan dan penyelesaian skripsi ini.
3. Drs. Marji, M.T., selaku Ketua Prodi Teknik Informatika / Ilmu Komputer Universitas Brawijaya yang selalu memberi semangat dan motivasi pada penulis dalam proses pengerjaan dan penyelesaian skripsi ini.
4. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Universitas Brawijaya.
5. Jajaran staf dan karyawan Program Teknologi Informasi dan Ilmu Komputer.
6. Keluarga besar penulis dikota Solo dan sekitarnya. Terkhusus untuk Mama Murtantina, adikku Novia Putri Bertina, eyang uti Siti Nuraini, Pakdhe Sunaryanto, S.Pd yang telah memberikan kasih sayang, doa, motivasi, serta support yang luar biasa selama penulis menempuh pendidikan di Universitas Brawijaya.
7. Keluarga besar Prof. DR. dr. H. Djanggan Sargowo, SpPD, SpJP, Mbak Dyaning Wahyu Primasari, Mbak Hapsari Retno Dewanti. Yang telah

memberikan support yang luar biasa baik materi maupun non-materi selama penulis menempuh pendidikan di Universitas Brawijaya.

8. Keluarga Bapak Udi Sampurno dan Ibu Gung Endah, Mas Sapto Bagus, Mbak Arini yang sudah memberikan nasehat, petuah hidup, motivasi yang berguna untuk penulis.
9. Eko Alfiyanto, Gumilang Ajie Hendicempata, Moh. Shohib Habibi, Ardhy Wisdarianto, Alfian Ardhi, Ardhiyan Syahrullah, dan Nasrul Akhmad Hidayat. Terima kasih menjadi sahabat baik, setia, dan selalu bersama-sama.
10. Mas Moch. Lutfi, Mas M. Dedi Rudianto, dan Mas Fais Al Huda yang telah bersedia meluangkan waktunya untuk memberikan bantuan dan arahan yang berkaitan tentang *coding* program skripsi.
11. Teman-teman program studi Ilmu Komputer angkatan 2008 dan 2009 yang selalu memberikan bantuan dan motivasinya demi kelancaran skripsi ini.
12. Semua pihak yang telah membantu terselesaikannya skripsi ini yang tidak dapat disebutkan satu per satu.

Semoga skripsi ini bermanfaat bagi pembaca sekalian, Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, maka kritik dan saran yang membangun dari semua pihak sangat diharapkan demi penyempurnaan skripsi ini. Kritik dan saran dapat dikirimkan ke email penulis di dimas.sasono@gmail.com.

Malang, Januari 2014

Penulis

ABSTRAK

Dimas Prasetyo Adi Sasono. 2014. Klasifikasi *Genre* Film Berdasarkan Judul Dan Sinopsis Menggunakan *Fuzzy k-Nearest Neighbour* (*Fuzzy k-NN*). Skripsi Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing: Lailil Muflikhah, S.Kom,.M.Sc, dan Drs. Achmad Ridok, M.Kom.

Film merupakan salah satu media hiburan yang digemari oleh masyarakat luas. Makin berkembangnya industri perfilman berdampak pada banyaknya sajian tayangan film dengan berbagai macam *genre* yang bisa dinikmati oleh masyarakat. Diperlukan klasifikasi terhadap tayangan film tersebut supaya masyarakat bisa memilih tayangan film sesuai dengan kebutuhannya. Proses klasifikasi film ditentukan berdasarkan teks judul dan sinopsis dari suatu film. Pada penelitian ini digunakan metode *Fuzzy k-Nearest Neighbour*. Data *testing* yang diklasifikasikan diberikan nilai keanggotaan untuk semua kelas *genre* berdasarkan sejumlah *k* data *training* yang memiliki jarak terdekat dengan data *testing*. Proses klasifikasi dilakukan dengan memilih nilai keanggotaan kelas *genre* tertinggi pada data *testing* tersebut. Pada penelitian ini dilakukan beberapa pengujian untuk melihat nilai tingkat akurasi yang dihasilkan. Untuk pengujian pengaruh jumlah data *training* yang bervariasi beserta nilai *k* diketahui bahwa penggunaan nilai *k* yang optimal rata-rata berada pada rentang $k = 10$ hingga $k = 20$. Sedangkan pada pengujian pengaruh penggunaan metode pencarian jarak, yaitu metode *cosine similarity* dan *euclidean distance* diperoleh hasil bahwa nilai *F-measure* yang dihasilkan pada penggunaan metode *cosine similarity* lebih baik jika dibandingkan metode *euclidean distance*.

Kata Kunci : *Genre*, Klasikasi teks, *Fuzzy k-Nearest Neighbour*.

ABSTRACT

Dimas Prasetyo Adi Sasono. 2014. Movie Genre Classification Based On Title And Synopsis Using Fuzzy k-Nearest Neighbour (Fuzzy k-NN). Minor Thesis Program of Study Information Technology / Computer Science, Program of Technology Information and Computer Science University of Brawijaya.
Advisor: Lailil Muflikhah, S.Kom,.M.Sc, and Drs. Achmad Ridok, M.Kom.

Movie is one of many entertainments media that is liked by public mass. Due to the development of scene industry, it's affected on the numerous offers of film publication with various genres that can be consumed by public. Classification is required to the publication of films so that public can choose which one of films that appropriate with their needs. The process of movie classification is determined based on title and synopsis from a movie. This research is using Fuzzy k-Nearest Neighbour algorithm. The testing data that have been classified is given a membership value to the whole genre classes based on how many k training data which have the closest range with the testing data. The process of classification is done by choosing the highest genre classes of membership value on those testing data. In this research is done with some experiments to look for the accurate value levels that have made. For the experiments of the influence of total training data which have many variations including k value is known that, the use of optimal k value is ranged between k = 10 until k = 20. In the other hand in the influence experiment that use distance measure method, that are cosine similarity method and euclidean distance method is gotten F-measure value that is produced in the use of cosine similarity method which is more better than the euclidean distance method.

Keywords : Genre, Text Classification, Fuzzy k-Nearest Neighbour.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SOURCECODE

BAB I
PENDAHULUAN

1.1 Latar Belakang

Film merupakan salah satu media hiburan bagi masyarakat luas. Film sendiri dapat juga berarti sebuah industri, yang mengutamakan eksistensi dan ketertarikan cerita yang dapat mengajak banyak orang terlibat. Semakin berkembangnya industri perfilman dalam negeri maupun luar negeri dari berbagai rumah produksi, berbanding lurus dengan persaingan dalam menghasilkan karya film yang menarik dan berkualitas untuk para penikmat film diseluruh dunia. Makin bermunculan pula film-film dengan berbagai *genre*, ide cerita, dan berbagai teknologi pembuatannya termasuk proses *editing*. Para penikmat film, di era sekarang tidak harus pergi ke bioskop untuk menikmati tayangan dari suatu film. Film bisa dinikmati melalui media lain seperti televisi. Film yang disimpan dalam bentuk file berekstensi *audio visual*, dapat dinikmati dalam media komputer.

Dengan begitu mudahnya masyarakat mendapatkan sajian tayangan film, perlu adanya klasifikasi terhadap film tersebut, apakah *genre* film tersebut sesuai dengan kebutuhan dari seorang penikmat film. Ketidaksesuian *genre* film yang ditonton oleh seorang penikmat film akan menimbulkan ketidakpuasan, karena disebabkan kegemaran setiap individu dalam menikmati tayangan film tidaklah sama.

Supaya penikmat film tidak salah menonton film, disusunlah penelitian tentang bagaimana membantu penikmat film untuk bisa menentukan *genre* dari suatu film. Proses penentuan menggunakan metode klasifikasi *genre* film berdasarkan judul dan sinopsis ceritanya tanpa harus melihat tayangan sajian suatu film termasuk *preview* suatu film. Dalam kenyataannya, film dikategorikan kedalam berbagai macam *genre*. Satu judul film dapat memiliki sedikitnya dua macam *genre*. Agar diperoleh hasil klasifikasi yang lebih baik diperlukan metode klasifikasi yang dapat digunakan untuk mengklasifikasi film kedalam berbagai *genre* yang sesuai. Salah satu alternatif dalam menyelesaikan kasus klasifikasi ini adalah dengan *text mining* dengan metode *Fuzzy k-Nearest Neighbour*.

Metode *Fuzzy k-Nearest Neighbour* dikembangkan pertama kali oleh James M. Keller. Metode *Fuzzy k-Nearest Neighbour*, sesuai namanya merupakan penggabungan antara teori *Fuzzy* dan teori *k-Nearest Neighbour*. Dasar dari algoritma ini adalah pemberian nilai *membership* sebagai fungsi pola jarak / kesamaan dari sejumlah himpunan *k-Nearest Neighbour* dan pemberian nilai keanggotaan *neighbour* pada kelas tertentu. Sehingga pada algoritma ini, data *testing* yang akan diklasifikasikan akan memiliki nilai keanggotaan pada semua kelas. Klasifikasi algoritma ini nantinya akan memilih nilai keanggotaan kelas pada data *testing* yang paling tinggi [ZHA-09].

Penelitian sebelumnya yang berhubungan dengan kasus ini yaitu berjudul *Web Document Classification Based on Fuzzy k-NN Algorithm* yang dilakukan oleh Juan Zhang, dkk. Kelebihan dari penelitian yang dilakukan menggunakan metode *Fuzzy k-NN* ini adalah tingkat keakurasian yang dihasilkan lebih baik jika dibandingkan dengan metode klasifikasi yang lain seperti *k-Nearest Neighbour* (*k-NN*), dan *Support Vector Machine* (*SVM*). Dari hasil penelitian yang dilakukan didapatkan nilai presentasi akhir *Fuzzy k-Nearest Neighbour* (*Fuzzy k-NN*) 64,12%, *k-Nearest Neighbour* (*k-NN*) 58,23%, dan *Support Vector Machine* (SVM) 58,45%. Metode *Fuzzy k-NN* dapat digunakan untuk menentukan klasifikasi kedalam lebih dari satu jenis kategori. Tetapi dalam penelitian ini juga terdapat kelemahan yaitu dari segi kecepatan pengklasifikasian yang lebih lambat jika dibandingkan dengan metode pengklasifikasian yang lain seperti *k-Nearest Neighbour* (*k-NN*), dan *Support Vector Machine* (*SVM*) [ZHA-09].

Jika pada penelitian sebelumnya Juan Zhang, dkk menggunakan metode *euclidean distance* sebagai metode pencarian jarak, maka pada penelitian ini akan digunakan metode *cosine similarity* sebagai metode pengukuran kedekatan antar vektor dokumen. Diharapkan dengan modifikasi ini dapat diketahui perbedaan pengaruh penggunaan salah satu dari metode *cosine similarity* atau *euclidean distance* terhadap tingkat akurasi hasil klasifikasi yang dihasilkan.

Berdasarkan latar belakang yang telah dikemukakan, maka judul yang diambil dalam penelitian ini adalah "**Klasifikasi *Genre* Film Berdasarkan Judul dan Sinopsis Menggunakan *Fuzzy k-Nearest Neighbour* (*Fuzzy k-NN*)**".

1.2 Rumusan Masalah

Dengan adanya latar belakang di atas, maka dapat dirumuskan permasalahan yang akan dijadikan objek penelitian ini, yaitu :

1. Bagaimana implementasi metode *Fuzzy k-NN* untuk melakukan klasifikasi *genre* film berdasarkan judul dan sinopsisnya ?
2. Bagaimana pengaruh jumlah data *training*, dan besarnya nilai *k* (jumlah tetangga terdekat) terhadap tingkat akurasi hasil klasifikasi penelitian pada setiap kategori dengan metode *Fuzzy k-NN* ?
3. Bagaimana pengaruh penggunaan metode *cosine similarity* dan *euclidean distance* terhadap tingkat akurasi hasil klasifikasi penelitian dengan metode *Fuzzy k-NN* ?

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan ruang lingkup sebagai berikut :

1. Hal yang dibahas dalam penelitian ini adalah untuk dapat mengetahui hasil klasifikasi data *testing* yang berupa file teks judul dan sinopsis dari suatu film.
2. Judul dan sinopsis film berbahasa Inggris.
3. Algoritma *stemming* yang digunakan adalah *Porter Stemmer* yang telah dikembangkan oleh M.F Porter sebelumnya.
4. Kategori yang digunakan dalam penelitian ini berupa *genre* primer film yang berjumlah dua belas buah, yaitu *action*, *drama*, *history*, *fantasy*, *sci-fi*, *horror*, *comedy*, *crime*, *musical*, *adventure*, *war*, dan *western*.
5. Sumber data penelitian (data *training* maupun data *testing*) diperoleh dari www.imdb.com, dan disimpan dalam format *file text* dengan ekstensi txt (*.*txt*).
6. Format data *training* maupun data *testing* yang digunakan pada penelitian ini adalah baris pertama file berisi judul sinopsis, baris kedua berisi *genre* sinopsis, dan baris ketiga hingga baris terakhir berisi teks sinopsis keseluruhan.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini, antara lain :

1. Mengetahui implementasi metode *Fuzzy k-NN* untuk melakukan klasifikasi *genre* film berdasarkan judul dan sinopsisnya.
2. Mengetahui pengaruh jumlah data *training*, dan besarnya nilai *k* (jumlah tetangga terdekat) terhadap tingkat akurasi hasil klasifikasi penelitian pada setiap kategori dengan metode *Fuzzy k-NN*
3. Mengetahui pengaruh penggunaan metode *cosine similarity* dan *euclidean distance* terhadap tingkat akurasi hasil klasifikasi penelitian dengan metode *Fuzzy k-NN*.

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah :

1. Memahami penerapan metode *Fuzzy k-Nearest Neighbour* dalam pengklasifikasian *genre* film secara otomatis.
2. Mempermudah pengklasifikasian *genre* film sesuai dengan judul dan sinopsisnya. Tanpa harus melihat film secara keseluruhan, maupun *trailer* atau *preview* dari suatu film.

1.6 Metodologi Penelitian

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah berikut:

1. **Studi Literatur**

   Pada tahap ini dilakukan pemahaman kepustakaan beberapa literatur (buku, jurnal ilmiah dan artikel dari *website*) mengenai sinopsis film, *genre* film, *text mining* beserta tahapannya, *stemming*, pembobotan, metode pencarian jarak dan metode *Fuzzy k-NN*.

2. **Analisis Kebutuhan**

   Pada tahap ini dilakukan analisis kebutuhan data dari sistem (baik data *training* maupun data *testing*) dan mengidentifikasi kebutuhan dari pengguna terhadap sistem yang akan dibuat.

3. **Perancangan Perangkat Lunak**

   Pada tahap ini dilakukan perancangan dari hasil analisis kebutuhan yang telah dilakukan, meliputi proses *text preprocessing*, sampai dengan implementasi algoritma *Fuzzy k-NN* dalam proses klasifikasi teks.

4. **Implementasi Perangkat Lunak**

   Pada tahap ini dilakukan implementasi rancangan sistem yang telah dibuat. Dalam tahap ini akan direalisasikan apa yang sudah menjadi rancangan sistem sehingga menjadi aplikasi yang sesuai dengan apa yang sudah direncanakan dan dibutuhkan pihak yang terkait.

5. **Uji Coba dan Evaluasi**

   Pada tahap ini, dilakukan uji coba terhadap sistem yang telah dibuat kemudian dilakukan perbaikan apabila terdapat kesalahan sehingga dapat dilakukan evaluasi terhadap hasil uji coba tersebut.

6. **Penyusunan Laporan Penelitian**

   Pada tahap ini merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan sistem serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan

Pembuatan tugas akhir ini dilakukan dengan sistematika penulisan sebagai berikut :

1. **BAB I PENDAHULUAN**

   Bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

2. **BAB II TINJAUAN PUSTAKA**

   Bab ini berisi teori dari berbagai pustaka yang menunjang dalam penelitian ini. Teori yang terdapat pada bab ini antara lain berisi mengenai definisi film dan *genre* beserta karakteristiknya, konsep *data mining* dan teks *mining*, proses *preprocessing*, pembobotan, metode pencarian jarak, dan metode *Fuzzy k-NN*.

3. **BAB III METODOLOGI DAN PERANCANGAN**

   Bab ini berisi mengenai perancangan sistem perangkat lunak yang digunakan untuk penelitian ini, meliputi analisis data, analisis sistem, rancangan sistem, contoh perhitungan manual, rancangan pengujian, dan rancangan antar muka sistem.

4. **BAB IV IMPLEMENTASI**

   Bab ini berisi segala hal yang berkaitan dengan implementasi sistem perangkat lunak yang digunakan untuk penelitian. Meliputi implementasi *sourcecode*, dan antar muka sistem.

5. **BAB V HASIL DAN PEMBAHASAN**

   Bab ini berisi hasil dari implementasi sistem perangkat lunak yang digunakan untuk mengukur akurasi hasil klasifikasi, pembahasan analisis hasil uji coba dan evaluasi uji coba.

6. **BAB VI PENUTUP**

   Bab ini berisi kesimpulan dari hasil penelitian dan saran-saran yang bermanfaat untuk pengembangan penelitian ini selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Film

Definisi Film Menurut UU 8/1992, adalah karya cipta seni dan budaya yang merupakan media komunikasi massa pandang-dengar yang dibuat berdasarkan asas sinematografi dengan direkam pada pita seluloid, pita video, piringan video, dan/atau bahan hasil penemuan teknologi lainnya dalam segala bentuk, jenis, dan ukuran melalui proses kimiawi, proses elektronik, atau proses lainnya, dengan atau tanpa suara, yang dapat dipertunjukkan dan/atau ditayangkan dengan sistem proyeksi mekanik, eletronik, dan/atau lainnya [ANO-92].

2.2 Sinopsis

Sinopsis adalah merupakan istilah yang digunakan dalam pembuatan ringkasan dari sebuah karya sastra yang berbentuk prosa dan drama. Sinopsis bisa diartikan ringkasan cerita dari alur yang panjang menjadi cerita singkat namun dapat menjelaskan secara keseluruhan cerita tersebut [ANO-12].

2.3 *Genre*

2.3.1 Definisi *Genre*

Genre berasal dari bahasa Perancis yang bermakna “bentuk” atau “tipe”. Didalam film, *genre* dapat didefinisikan sebagai jenis atau klasifikasi dari sekelompok film yang memiliki karakter atau pola yang sama (khas) seperti *setting*, isi, dan subyek cerita, tema, struktur cerita, aksi, atau peristiwa, periode, gaya, situasi, ikon, mood, serta karakter. Dari klasifikasi tersebut dapat dihasilkan *genre-genre* film popular seperti aksi, petualangan, drama, komedi, horor, *western*, *film noir*, roman, dan sebagainya [PRA-08].

2.3.2 Fungsi *Genre*

Fungsi utama dan *genre* adalah untuk memudahkan klasifikasi sebuah film. *Genre* juga dapat membantu dalam memilih film-film tersebut sesuai dengan spesifikasinya. Industri film sendiri sering menggunakan *genre* sebagai strategi

marketing. *Genre* apa yang saat ini menjadi tren, menjadi tolok ukur film yang akan diproduksi. Selain untuk klasifikasi, *genre* juga dapat berfungsi sebagai antisipasi penonton terhadap film yang akan ditonton. Jika seorang penonton telah memutuskan untuk melihat sebuah film ber-*genre* tertentu, maka sebelumnya ia telah mendapatkan gambaran umum dikepalanya tentang film yang akan ia tonton. Misalnya jika ingin mendapatkan tayangan yang sifatnya menghibur, umumnya memilih film ber-*genre* komedi atau aksi [PRA-08].

2.3.3 Jenis *Genre* dan Ciri-Cirinya

Dalam setiap film cerita setidaknya memiliki satu *genre* induk. Dan *genre* induk tersebut terdiri dan dua kelompok, yaitu *genre* induk primer dan *genre* induk sekunder.

Genre induk primer merupakan *genre* pokok yang telah ada sejak awal perkembangan sinema era 1900-an hingga 1930-an. Berikut ini adalah jenis-jenis *genre* induk primer, antara lain *action*, *drama*, *history*, *fantasy*, *sci-fi*, *horror*, *comedy*, *crime*, *musical*, *adventure*, *war*, dan *western* [PRA-08].

2.3.3.1 *Action*

Film-film aksi berhubungan dengan adegan-adegan aksi fisik seru, menegangkan, berbahaya, nonstop dengan tempo cerita yang cepat. Umumnya berisi adegan aksi kejar-mengejar, perkelahian, tembak-menembak, balapan, berpacu dengan waktu, ledakan, serta aksi-aksi fisik lainnya. Aksi kejar mengejar sering kali menggunakan berbagai cara dan alat transportasi.

Film-film aksi juga umumnya memiliki karakter protagonis dan antagonis yang jelas serta konflik berupa konfrontasi fisik. Tokoh protagonis biasanya adalah seorang penegak hukum seperti polisi, detektif, agen pemerintah, tentara, veteran perang, dan sebagainya. Dan dalam cerita film biasanya tokoh protagonis selalu terancam jiwanya dan selalu berada dalam tekanan pihak antagonis.

2.3.3.2 *Drama*

Film drama bisa jadi merupakan *genre* yang paling banyak diproduksi karena jangkauan ceritanya yang sangat luas. Film-film drama biasanya

berhubungan dengan tema, cerita, *setting*, karakter serta suasana yang memotret kehidupan nyata. Konflik bisa dipicu oleh lingkungan, diri sendiri, maupun alam. Kisahnya sering kali menggugah emosi, dramatik, dan mampu menguras air mata penontonnya. Tema umumnya mengangkat isu-isu sosial baik skala besar (masyarakat) maupun skala kecil (keluarga) seperti ketidakadilan, kekerasan, diskriminasi, rasialisme, ketidakharmonisan, masalah kejiwaan, penyakit, kemiskinan, politik, kekuasaan, dan sebagainya.

2.3.3.3 *History*

Genre ini pada umumnya mengambil tema periode masa silam (sejarah) dengan latar sebuah kerajaan, peristiwa atau tokoh besar yang menjadi mitos, legenda, atau kisah biblikal. Film berskala besar (kolosal) ini sering kali menggunakan *setting* mewah dan megah, ratusan hingga ribuan figuran. variasi kostum dengan aksesoris yang unik, serta variasi perlengkapan perang seperti pedang, tombak, helm, kereta kuda, panah, dan sebagainya.

Film epik sejarah juga sering menyajikan aksi pertempuran skala besar yang berlangsung lama. Tokoh utama biasanya merupakan sosok heroik yang gagah berani dan disegani oleh semua lawannya. *Genre* biografi merupakan pengembangan dan *genre* epik sejarah. Namun tidak seperti biografi, tingkat keakuratan cerita dalam epik sejarah sering kali dikorbankan.

2.3.3.4 *Fantasy*

Film fantasi berhubungan dengan tempat. peristiwa, serta karakter yang tidak nyata. Film fantasi berhubungan dengan unsur magis, mitos, negeri dongeng, imajinasi, halusinasi, serta alam mimpi.

Film fantasi berhubungan dengan pedang dan mantera gaib, naga, kuda terbang, karpet terbang, dewa-dewi, penyihir, jin, serta peri. Film fantasi terkadang juga berhubungan dengan aspek religi, seperti Tuhan atau malaikat yang turun ke bumi, campur tangan kekuatan Ilahi, surga dan neraka, dan lain sebagainya.

2.3.3.5 *Sci-Fi*

Fiksi ilmiah berhubungan dengan masa depan, perjalanan angkasa luar, percobaan ilmiah, penjelajahan waktu, invasi atau kehancuran bumi. *Genre* ini sering berhubungan dengan teknologi serta kekuatan yang berada diluar jangkauan teknologi masa kini.

Fiksi ilmiah biasanya berhubungan dengan karakter non-manusia atau *artificial*, seperti makhluk asing, robot, monster, hewan purba, dan sebagainya. Film fiksi ilmiah mengalami masa emas pada era 1950-an dan hingga kini pun masih popular.

2.3.3.6 *Horror*

Film horor memiliki tujuan utama memberikan efek rasa takut, kejutan, serta teror bagi penontonnya. Plot film horor, umumnya sederhana, yakni bagaimana usaha manusia untuk melawan kekuatan jahat dan biasanya berhubungan dengan dimensi supernatural atau sisi gelap manusia.

Film horor umumnya menggunakan karakter-karakter antagonis non manusia yang berwujud fisik menyeramkan. Pelaku teror bisa berwujud manusia, makhluk gaib, monster, hingga makhluk asing. Film horor biasanya berkombinasi dengan *genre* supernatural (melibatkan makhluk supernatural atau gaib, seperti hantu, *vampire*, atau *werewolf*), fiksi-ilmiah (melibaikan makhluk luar angkasa luar atau hasil uji coba ilmiah, seperti *alien*, *zombie*, atau mutan), serta *thriller* (melibatkan seorang psikopat atau pembunuh serial).

2.3.3.7 *Comedy*

Komedi boleh jadi merupakan *genre* yang paling popular diantara semua *genre* lainnya sejak era silam. Komedi adalah jenis film yang tujuan utamanya memancing tawa penontonnya.

Film komedi biasanya berupa film drama ringan yang melebih-lebihkan aksi, siluasi, bahasa, hingga karakternya. Film komedi juga hiasanya berakhir dengan penyelesaian cerita yang memuaskan penontonnya (*happy ending*). Film komedi secara umum dibagi menjadi dua jenis, yaitu komedi situasi (unsur

komedi menyatu dengan cerita) serta komedi lawakan (unsur komedi bergantung pada figur komedian). Namun keduanya juga sering berkombinasi.

2.3.3.8 *Crime*

Dalam film kriminal dan gangster berhubungan dengan aksi-aksi kriminal seperti perampokan bank. pencurian, pemerasan, perjudian, pembunuhan, persaingan antar kelompok, serta aksi kelompok bawah tanah yang bekerja diluar sistem hukum.

Sering kali film jenis ini mengambil kisah kehidupan tokoh kriminal yang diinspirasi dan kisah nyata. *Genre* ini juga sening menampilkan perseteruan antara pelaku kriminal dengan penegak hukum seperti detektif swasta, polisi, pengacara, atau agen rahasia. Ciri khas adegan aksi dan *genre* tersebut adalah menggunakan tongkat pemukul, senapan mesin, serta bom mobil.

2.3.3.9 *Musical*

Genre musikal adalah film yang mengkombinasi unsur musik. lagu, tari (dansa), serta gerak (koreografi). Lagu-lagu dan tarian biasanya mendominasi sepanjang film dan biasanya menyatu dengan cerita. Penggunaan musik dan lagu bersama liriknya biasanya mendukung jalannya alur cerita.

Cerita film musikal umumnya berkisah ringan seperti percintaan, kesuksesan, serta popularitas. Sasaran dalam film musikal lebih ditujukan untuk penonton keluarga, remaja, dan anak-anak.

2.3.3.10 *Adventure*

Film petualangan berkisah tentang perjalanan, eksplorasi, atau ekspedisi ke suatu wilayah asing yang belum pernah tersentuh. Film-film petualangan selalu menyajikan panorama alam eksotis seperti hutan rimba, pegunungan, savana, gurun pasir, lautan, serta pulau terpencil.

Plot film umumnya seputar pencarian sesuatu yang bernilai seperti harta karun, artefak, kota yang hilang, mineral (emas dan berlian), dan sebagainya. Atau usaha penyelamatan diri dari suatu wilayah.

2.3.3.11 *War*

Genre perang mengangkat tema kengerian serta teror yang ditimbulkan oleh aksi perang. Umumnya film jenis ini menampilkan adegan pertempuran seru baik di darat, laut, maupun udara.

Biasanya film-film perang memperlihatkan kegigihan, perjuangan, dan pengorbanan para tentara dalam melawan musuh-musuh mereka. Film perang umumnya menampilkan adegan pertempuran dengan kostum, peralatan, perlengkapan, serta strategi yang relatif modern, mulai dari seragam, topi, sepatu bot, pistol, senapan mesin, granat, meriam, tank, helikopter, rudal, torpedo, pesawat jet, kapal tempur, kapal selam, dan lain sebagainya.

2.3.3.12 *Western*

Western adalah sebuah *genre* orisinil milik Amerika. *Western* memiliki beberapa ciri karakter tema serta fisik yang sangat spesifik. Umumnya tema dalam film ini adalah seputar konflik antara pihak baik dan pihak jahat. *Setting* sering kali menampilkan kota kecil, bar, padang gersang, sungai, rel kereta api. pohon kaktus, *ranch* atau peternakan, serta perkampungan suku Indian. *Western* memiliki karakter-karakter yang khas yakni koboi, Indian, kavaleri, *sheriff* deputi, juga binatang seperti kuda, sapi, keledai, ular derik, burung bangkai, dan sebagainya.

Film ini umumnya berisi aksi berkuda, lempar tali (laso), tembak menembak. serta yang menjadi *trademark* yaitu aksi duel. Karakter karakternya memiliki perlengkapan serta kostum yang khas seperti pistol, senapan, jaket kulit, sabuk, topi, sepatu bot, hingga aksen (dialog) yang khas.

2.4 *Data Mining*

Dikutip dari Kusrini dan Luthfi tahun 2009, Turban dkk menyatakan Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam database. Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar [KUS-09].

Data mining adalah proses menemukan pola yang menarik dan pengetahuan dari sejumlah data skala besar. Sumber data dapat mencakup *database*, *data warehouse*, *web*, repositori informasi lainnya, atau data yang mengalir ke sistem secara dinamis [HAN-12].

2.5 *Text Mining*

Teks mining adalah proses menemukan sesuatu yang baru dengan serangkaian proses komputasi dari informasi yang sebelumnya tidak diketahui kegunaanya, dengan melakukan ekstraksi secara otomatis informasi dari dokumen yang berbeda. Elemen kunci dari proses teks mining adalah menghubungkan bersama informasi yang telah diekstraksi untuk membentuk fakta-fakta atau hipotesis baru yang akan digali lebih lanjut dengan metode eksperimen yang lebih konvensional.

Teks mining berbeda dengan *web search* (pencarian web) yang sudah dikenal. Dalam *web search*, pengguna akan mencari suatu informasi yang sudah dikenal dan telah didokumentasikan oleh orang lain. Permasalahannya adalah membuang semua bahan yang saat ini tidak relevan dengan kebutuhan, supaya dapat menemukan informasi yang relevan. Dalam teks mining, tujuannya adalah untuk menemukan informasi yang hingga periode tertentu belum diketahui, sesuatu yang belum diketahui oleh siapapun dan sehingga belum bisa dituliskan [HEA-03].

2.6 Klasifikasi Teks

Teks kategorisasi atau klasifikasi teks adalah cara untuk menetapkan kategori standar pada dokumen teks. Dengan adanya klasifikasi teks, maka dapat memberikan pandangan konseptual mengenai cara pengelompokkan dokumen yang sebenarnya memiliki peranan penting dalam dunia nyata. Misalnya, berita yang biasanya dikelola berdasarkan topik atau kode geografis, makalah (*paper*) akademis sering diklasifikasikan berdasarkan bidang ilmunya, laporan pasien di rumah sakit diklasifikasi dari berbagai aspek yaitu menggunakan taksonomi kategori penyakit, jenis prosedur penanganan, kode penggantian asuransi dan sebagainya. Aplikasi lain teks kategorisasi yang lebih luas adalah *spam filtering*,

dimana tiap pesan *email* diklasifikasikan ke dalam dua kategori *spam* dan non−*spam* [YAN-08].

Tujuan dari pengkategorian teks adalah mengklasifikasi dokumen kedalam sejumlah kategori yang umum. Tiap dokumen dapat diklasifikasikan dalam beberapa kategori, hanya satu, atau tidak berkategori sama sekali. Dengan menggunakan *machine learning*, yang berguna untuk pembelajaran aturan klasifikasi dari contoh berupa data latih (*training set*) sehingga dapat melakukan proses pengklasifikasian kedalam kategori secara otomatis. Ini merupakan suatu *supervised learning*. Karena kategori mungkin tumpang tindih satu sama lain, masing-masing kategori diperlakukan sebagai *binary classification* yang terpisah [JOA-98].

2.7 *Text Preprocessing*

Pada tahap *text preprocessing* dilakukan beberapa proses untuk menyiapkan judul dan sinopsis film untuk menjadi dokumen teks yang siap diolah pada tahap selanjutnya. Pada tahap ini pada umumnya terdapat beberapa proses, antara lain *tokenizing, filtering, stemming,* dan *term weighting* [GAR-05].

2.7.1 *Tokenizing*

Selama proses *tokenizing* berlangsung semua *string input* akan diuraikan sesuai dengan tiap kata yang menyusunnya. Setiap huruf *input* akan diubah menjadi huruf kecil. Semua tanda baca dan tanda hubung akan dihapuskan, termasuk semua karakter selain huruf alfabet [GAR-05].

2.7.2 *Filtering*

Proses *filtering* adalah proses menentukan istilah yang mewakili isi dari dokumen tersebut sehingga dapat digunakan untuk menggambarkan isi dari dokumen tersebut dan membedakan dokumen dari dokumen lain dalam koleksi [GAR-05]. Dalam proses ini bisa menggunakan algoritma *stoplist* (membuang kata yang tidak penting) atau *wordlist* (menyimpan kata yang penting). *Stoplist* adalah daftar kata yang sering digunakan dan tidak menjelaskan isi dari dokumen, atau *stopword*. Contoh *stopword* adalah '*the*', '*and*', '*what*',' *usually*'.

2.7.3 *Stemming*

Stemming adalah proses yang dilakukan untuk menguraikan bentuk kata menjadi kata dasarnya (*stem*). Tidak semua sistem bahasa menggunakan metode *stemming* yang sama. Untuk kata dalam bahasa Inggris, metode *stemming* yang paling populer digunakan adalah algoritma Martin Porter's *stemmer* atau biasa disebut *Porter Stemmer* [GAR-05].

Bentuk dasar (*stem*) dapat dianggap sebagai bentuk yang biasanya akan ditemukan sebagai entri dalam kamus. Sebagai contoh, 'go', 'goes', 'going', 'gone', dan 'went' akan dihubungkan dan diubah bentuknya menjadi bentuk dasarnya (*stem*) yaitu 'go' [JAC-02].

2.7.3.1 Algoritma *Porter Stemmer*

Banyak algoritma yang digunakan untuk *stemming* bahasa Inggris. Salah satu diantaranya adalah algoritma *Porter Stemmer*. Pada dasarnya algoritma *Porter Stemmer* merupakan algoritma penghilangan akhiran morfologi dan *infleksional* yang umum dari bahasa Inggris.

Langkah-langkah algoritma *Porter Stemmer* ini adalah [POR-80] :

- *Step* 1a : Menghapus atau mengganti akhiran pada kata yang berbentuk jamak. Kata yang berakhir dengan akhiran 'sses' diubah menjadi akhiran 'ss'. Kata yang berakhir dengan akhiran 'ies' diubah menjadi akhiran 'i'. Kata yang berakhir dengan akhiran 'ss' tidak mengalami perubahan. Terakhir, kata yang berakhir dengan akhiran 's' akan dihapus akhirannya hingga didapatkan bentuk dasar (*stem*). *Step* 1a ditunjukkan pada tabel 2.1.
- *Step* 1b : Menghapus atau mengganti akhiran pada kata yang mengalami modulasi lisan / pengucapan, berupa akhiran 'eed' menjadi 'ee' jika terdapat paling sedikit satu huruf vokal-konsonan berurutan. Akhiran 'ed', dan 'ing' tidak mengalami perubahan untuk kata yang memiliki hanya sebuah huruf vokal, namun akan dihapus untuk yang memiliki lebih dari satu huruf vokal. *Step* 1b ditunjukkan pada tabel 2.2.
- *Step* 1b1 : Merupakan tahap lanjutan untuk *rule* aturan 'ed' dan 'ing'. Hasil *stemming* pada akhiran 'ed' dan 'ing' pada *step* sebelumnya akan di-

stemming kembali. Kata yang berakhir dengan akhiran ‘at’ diubah menjadi akhiran ‘ate’. Kata yang berakhir dengan akhiran ‘bl’ diubah menjadi akhiran ‘ble’. Kata yang berakhir dengan akhiran ‘iz’ diubah menjadi akhiran ‘ize’. Untuk kata yang berakhir dengan dobel huruf konsonan dan tidak berakhir dengan akhiran huruf ‘l’, ‘s’, atau ‘z’ akan diubah menjadi kata yang berakhir dengan akhiran satu huruf konsonan saja, tetapi jika kata berakhiran dengan huruf ‘l’, ‘s’, atau ‘z’, maka tidak diganti. Untuk kata yang berakhir dengan huruf konsonan-vokal-konsonan berurutan dimana konsonan akhirnya bukan ‘w’, ‘x’, atau ‘y’ dan hanya terdapat satu urutan huruf vokal-konsonan didalamnya maka ditambahkan akhiran ‘e’. *Step* 1b1 ditunjukkan pada tabel 2.3.

- *Step* 1c : Jika terdapat kata yang memiliki sebuah huruf vokal dan berakhir dengan akhiran ‘y’, maka akan diganti dengan akhiran ‘i’. *Step* 1c ditunjukkan pada tabel 2.4.
- *Step* 2 : Jika terdapat kata yang memiliki sedikitnya satu huruf vokal-konsonan berurutan, maka kata yang berakhir dengan akhiran ‘ational’, ‘ation’, atau, ‘ator’ diubah menjadi akhiran ‘ate’. Kata yang berakhir dengan akhiran ‘tional’ diubah menjadi akhiran ‘tion’. Kata yang berakhir dengan akhiran ‘enci’ diubah menjadi akhiran ‘ence’. Kata yang berakhir dengan akhiran ‘anci’ diubah menjadi akhiran ‘ance’. Kata yang berakhir dengan akhiran ‘izer’, atau ‘ization’ diubah menjadi akhiran ‘ize’. Kata yang berakhir dengan akhiran ‘iviti’, atau ‘iveness’ menjadi akhiran ‘ive’. Kata yang berakhir dengan akhiran ‘ality’, ‘alism’, atau ‘alli’ diubah menjadi akhiran ‘al’. Kata yang berakhir dengan akhiran ‘biliti’ diubah menjadi akhiran ‘ble’. Kata yang berakhiran dengan akhiran ‘abli’ diubah menjadi akhiran ‘able’. Kata yang berakhir dengan akhiran ‘ently’ diubah menjadi akhiran ‘ent’. Kata yang berakhir dengan akhiran ‘eli’ diubah menjadi akhiran ‘e’. Kata yang berakhir dengan akhiran ‘ousli’, atau ‘ousness’ diubah menjadi akhiran ‘ous’. Kata yang berakhir dengan akhiran ‘fulness’ diubah menjadi akhiran ‘ful’. *Step* 2 ditunjukkan pada tabel 2.5.

- *Step* 3 : Jika terdapat kata yang memiliki sedikitnya satu huruf vokal-konsonan berurutan, maka kata yang berakhir dengan akhiran ‘ative’, ‘ful’, atau ‘ness’ akan dihapus akhirannya. Sedangkan kata yang berakhir dengan akhiran ‘icate’, ‘iciti’ atau ‘ical’ akan diubah menjadi akhiran ‘ic’, dan kata yang berakhir dengan akhiran ‘alize’ akan diubah menjadi akhiran ‘al’. *Step* 3 ditunjukkan pada tabel 2.6.
- *Step* 4 : Jika terdapat kata yang memiliki sedikitnya dua huruf vokal-konsonan berurutan, maka kata yang berakhir dengan akhiran ‘al’, ‘ance’, ‘ence’, ‘er’, ‘ic’, ‘able’, ‘ible’, ‘ant’, ‘ement’, ‘ment’, ‘ent’, ‘ion’, ‘ou’, ‘ism’, ‘ate’, ‘iti’, ‘ous’, ‘ive’, atau ‘ize’ akan dihapus akhirannya. *Step* 4 ditunjukkan pada tabel 2.7.
- *Step* 5a : Menghapus akhiran ‘e’, jika kata tersebut sedikitnya memiliki dua huruf vokal-konsonan yang berurutan atau memiliki sebuah huruf vokal-konsonan berurutan dan tidak diakhiri dengan akhiran huruf konsonan-vokal-konsonan berurutan, dimana konsonan akhir bukan ‘w’, ‘x’, atau ‘y’. *Step* 5a ditunjukkan pada tabel 2.8.
- *Step* 5b : Jika kata hanya memiliki sebuah huruf vokal-konsonan berurutan dan tidak berakhir dengan akhiran dobel huruf konsonan dan huruf ‘l’ maka diganti dengan akhiran satu huruf konsonan saja. *Step* 5b ditunjukkan pada tabel 2.9.

Tabel 2.1 *Stemming Step* 1a

Conditions	***Suffix***	***Replacement***	***Examples***
NULL	*sses*	*ss*	*caresses → caress*
NULL	*ies*	*i*	*ponies → poni*
			ties → ti
NULL	*ss*	*ss*	*caress → caress*
NULL	*s*	*NULL*	*cats → cat*

Tabel 2.2 *Stemming Step* 1b

Conditions	***Suffix***	***Replacement***	***Examples***
(m>0)	*eed*	*ee*	*feed → feed*
			agreed → agree
*(*v*)*	*ed*	*NULL*	*plastered → plaster*
			bled → bled
*(*v*)*	*ing*	*NULL*	*motoring → motor*
			sing → sing

Tabel 2.3 *Stemming Step* 1b1

Conditions	***Suffix***	***Replacement***	***Examples***
NULL	*at*	*ate*	*conflat(ed) → conflate*
NULL	*bl*	*ble*	*troubl(ing)→ trouble*
NULL	*iz*	*ize*	*siz(ed) → size*
*(*d and not (*<L> or *<S> or *<Z>))*	*NULL*	*single letter*	*hopp(ing) → hop*
			tann(ed) → tan
			fall(ing) → fall
			hiss(ing) → hiss
			fizz(ed) → fizz
*(m=1 and *o)*	*NULL*	*e*	*fail(ing) → fail*
			fil(ing) → file

Tabel 2.4 *Stemming Step* 1c

Conditions	***Suffix***	***Replacement***	***Examples***
*(*v*)*	*y*	*i*	*happy → happi*
			sky → sky

Tabel 2.5 *Stemming Step* 2

Conditions	***Suffix***	***Replacement***	***Examples***
(m>0)	*ational*	*ate*	*relational → relate*
(m>0)	*tional*	*tion*	*conditional → condition*

(m>0)			*rational → rational*
(m>0)	*enci*	*ence*	*valenci → valence*
(m>0)	*anci*	*ance*	*hesitanci → hesitance*
(m>0)	*izer*	*ize*	*digitizer → digitize*
(m>0)	*abli*	*able*	*conformabli → conformable*
(m>0)	*alli*	*al*	*radicalli → radical*
(m>0)	*entli*	*ent*	*differentli → different*
(m>0)	*eli*	*e*	*vileli → vile*
(m>0)	*ousli*	*ous*	*analogousli → analogous*
(m>0)	*ization*	*ize*	*vietnamization → vietnamize*
(m>0)	*ation*	*ate*	*predication → predicate*
(m>0)	*ator*	*ate*	*operator → operator*
(m>0)	*alism*	*al*	*feudalism → feudal*
(m>0)	*iveness*	*ive*	*decisiveness → decisive*
(m>0)	*fulness*	*ful*	*hopefullness → hopeful*
(m>0)	*ousness*	*ous*	*callousness → callous*
(m>0)	*aliti*	*al*	*formaliti → formal*
(m>0)	*iviti*	*ive*	*sensitiviti → sensitive*
(m>0)	*biliti*	*ble*	*sensibiliti → sensible*

Tabel 2.6 *Stemming Step* 3

Conditions	***Suffix***	***Replacement***	***Examples***
(m>0)	*icate*	*ic*	*triplicate → triplic*
(m>0)	*ative*	*NULL*	*formative → form*
(m>0)	*alize*	*al*	*formalize → formal*
(m>0)	*iciti*	*ic*	*electriciti → electric*
(m>0)	*ical*	*ic*	*electrical → electric*
(m>0)	*ful*	*NULL*	*hopeful → hope*
(m>0)	*ness*	*NULL*	*goodness → good*

Tabel 2.7 *Stemming Step* 4

Conditions	***Suffix***	***Replacement***	***Examples***
(m>1)	*al*	*NULL*	*revival → reviv*
(m>1)	*ence*	*NULL*	*allowence → allow*
(m>1)	*ance*	*NULL*	*inference → infer*
(m>1)	*er*	*NULL*	*airliner → airlin*
(m>1)	*ic*	*NULL*	*gyroscopic → gyroscop*
(m>1)	*able*	*NULL*	*adjustable → adjust*
(m>1)	*ible*	*NULL*	*defensible → defens*
(m>1)	*ant*	*NULL*	*irritant → irrit*
(m>1)	*ement*	*NULL*	*replacement → replac*
(m>1)	*ment*	*NULL*	*adjustment → adjust*
(m>1)	*ent*	*NULL*	*dependent → depend*
(m>1)	*ion*	*NULL*	*adoption → adopt*
(m>1)	*ou*	*NULL*	*homologou → homolog*
(m>1)	*ism*	*NULL*	*communism → commun*
(m>1)	*ate*	*NULL*	*activate → activ*
(m>1)	*iti*	*NULL*	*angulariti → angular*
(m>1)	*ous*	*NULL*	*homologous → homolog*
(m>1)	*ive*	*NULL*	*effective → effect*
(m>1)	*ize*	*NULL*	*bowdlerize → bowdler*

Tabel 2.8 *Stemming Step* 5a

Conditions	***Suffix***	***Replacement***	***Examples***
(m>1)	*e*	*NULL*	*probate → probat*
			rate → rate
*(m=1 and not *o)*	*e*	*NULL*	*cease → ceas*

Tabel 2.9 *Stemming Step* 5b

Conditions	*Suffix*	*Replacement*	*Examples*
*(m=1 and not *d and *<L>)*	*NULL*	*single letter*	*controll → control*
			roll → roll

Keterangan:

m : ukuran (*measure*) dari sebuah stem berdasarkan urutan vokal-konsonan.

**<X>* : berarti *stem* berakhir dengan huruf X.

v : berarti *stem* mengandung sebuah vokal.

**d* : berarti *stem* diakhiri dengan konsonan dobel.

**o* : berarti *stem* diakhiri dengan konsonan – vokal – konsonan, berurutan, di mana konsonan akhir bukan w, x, atau y.

2.7.4 *Weighting*

Setelah serangkaian *tokenizing*, *filtering*, serta *stemming* langkah utama selanjutnya dari proses klasifikasi dokumen adalah pemilihan fitur (*feature selection*) dari dokumen data *training*. Sudah tersedia beberapa metode untuk melakukan *feature selection*, sebagai contoh *document frequency, word frequency, mutual information, information gain, odds ratio, X^2 statistic* dan *term strength* [ZHA-09].

Metode yang akan digunakan dalam penelitian ini adalah *TF-IDF* (*Term Frequency* / *Inverse Document Frequency*).

Metode *TF-IDF* merupakan metode pembobotan *term* yang banyak digunakan sebagai metode pembanding terhadap metode pembobotan baru. Pada metode ini, perhitungan bobot *term* dalam sebuah dokumen dilakukan dengan mengalikan nilai *Term Frequency* dengan *Inverse Document Frequency*.

Term Frequency (*TF*) adalah faktor yang menentukan bobot *term* pada suatu dokumen berdasarkan jumlah kemunculannya dalam dokumen tersebut. Nilai jumlah kemunculan suatu kata (*term frequency*) diperhitungkan dalam pemberian bobot terhadap suatu kata. Semakin besar jumlah kemunculan suatu *term* dalam dokumen, semakin besar pula bobotnya dalam dokumen atau akan memberikan nilai kesesuian yang semakin besar.

Inverse Document Frequency (*IDF*) adalah pengurangan dominansi *term* yang sering muncul di berbagai dokumen. Hal ini diperlukan karena *term* yang banyak muncul di berbagai dokumen, dapat dianggap sebagai *term* umum (*common term*) sehingga tidak penting nilainya. Sebaliknya faktor kejarangmunculan kata (*term scarcity*) dalam koleksi dokumen harus diperhatikan dalam pemberian bobot. Kata yang muncul pada sedikit dokumen harus dipandang sebagai kata yang lebih penting (*uncommon term*) daripada kata yang muncul pada banyak dokumen. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kata (*inverse document frequency*). Hal ini merupakan usulan dari George Zipf. Zipf mengamati bahwa frekuensi dari sesuatu cenderung kebalikan secara proposional dengan urutannya [ZAF-08].

TF-IDF dihitung menggunakan persamaan 2.1 :

$$IDF(t_i) = \log(\frac{|D|}{DF(t_i)}) \tag{2.1}$$

Keterangan :

$IDF(t_i)$: *inverse document frequency* dari kata (*term*) t_i

D : jumlah dokumen *training* keseluruhan

$DF(t_i)$: jumlah dokumen *training* yang memiliki kata (*term*) t_i

Kemudian rumus untuk menghitung bobot kata (w_i) dalam dokumen dihitung menggunakan persamaan 2.2 :

$$w_i = TF(t_i, d) \times IDF(t_i) \tag{2.2}$$

Keterangan :

w_i : bobot kata (*term*) dalam dokumen *d*

$TF(t_i, d)$: banyaknya kata (*term*) t_i yang muncul dalam dokumen *d*

$IDF(t_i)$: *inverse document frequency* dari kata (*term*) t_i

[ZHA-09]

2.8 Pembentukan *Vector Space Model*

Vector Space Model adalah suatu model yang digunakan setelah proses pembobotan, dimana kumpulan dokumen direpresentasikan ke dalam sebuah vektor matrik. Dari matrik tersebut akan didapatkan titik koordinat tertentu.

Kasus klasifikasi teks untuk suatu dokumen biasanya tersusun dari kumpulan kata-kata (*string*), oleh karena itu harus ditransformasi kedalam suatu representasi yang sesuai untuk proses pembelajaran (*learning*) dan klasifikasi sistem. Untuk kasus ini digunakan fitur representasi vektor dokumen, dimana dokumen akan diset sebagai suatu rangkaian kata. Kemudian dokumen akan dibentuk menjadi pasangan dalam bentuk $< t, w >$. *Term* $t_1, t_2, t_3, \dots, t_n$ menyatakan fitur yang menunjukkan konten atau isi dari dokumen, bobot (*weight*) $w_1, w_2, w_3, \dots, w_n$, menyatakan nilai yang relevan dengan $t_1, t_2, t_3, \dots, t_n$ masing-masing. Setiap dokumen akan dipetakan sebagai fitur vektor :

$$V(d) = (t_1, w_1, t_2, w_2, t_3, w_3 \dots, t_n, w_n)$$

[LIP-02]

2.9 Logika *Fuzzy*

Logika *Fuzzy* merupakan salah satu komponen pembentuk *soft computing*. Logika *Fuzzy* pertama kali diperkenalkan Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut. Himpunan *fuzzy* memiliki 2 atribut, yaitu :

a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA
b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti : 40, 25, 50, dsb [KUS-10].

2.10 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering

juga disebut derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi [KUS-10].

2.11 Algoritma *k-Nearest Neighbour*

Algoritma *Nearest Neighbour* sangat bergantung pada proses pembelajarannya. Konsepnya, ketika proses *training* algoritma ini akan mempelajari dan bahkan "mengingat" semua data *training* beserta fitur yang ada didalamnya. Selanjutnya, ketika akan melakukan klasifikasi terhadap suatu data *testing*, maka algoritma ini akan memilih sejumlah *k* data *training* yang terdekat dengan data *testing*. Kemudian, mengambil satu atau lebih kategori sebagai acuan klasifikasi data *testing*, berdasarkan kategori yang muncul pada sejumlah *k* data *training* yang terdekat.

Untuk dapat menentukan hasil dari klasifikasi menggunakan algoritma *k-Nearest Neighbour*, hal pertama yang harus dilakukan adalah menentukan jarak untuk mengukur seberapa dekat jarak antara data *training* dengan data *testing*. Metode *euclidean distance* dapat digunakan untuk menentukan jarak antara data *training* dengan data *testing*. Selain itu dapat digunakan pula metode pengukuran kesamaan (*similarity*) untuk mengukur kemiripan antara data *training* dengan data *testing*. Adapun rumus *cosine similarity* dihitung menggunakan persamaan 2.3.

$$sim(q,d) = \frac{\sum_t w_{t,d} \cdot w_{t,q}}{\sqrt{\sum_t w_{t,d}^2} \cdot \sqrt{\sum_t w_{t,q}^2}} \tag{2.3}$$

Keterangan :

$sim(q,d)$: kemiripan (*similarity*) antara data *testing* dengan data *training*

q : data *testing*

d : data *training*

$w_{t,d}$: nilai bobot *term* data *training*

$w_{t,q}$: nilai bobot *term* data *testing*

t : indeks *term*

Berdasarkan percobaan yang telah dilakukan, algoritma *k-Nearest Neighbour* adalah algoritma klasifikasi yang paling efektif. Keefektifan *k-Nearest Neighbour* dipandang dari proses *training* yang cepat, karena yang harus dilakukan hanya menyimpan sejumlah data *training* sebagai suatu vektor. Namun dipandang dari sisi lain, proses klasifikasinya tidak sebegitu cepat karena banyaknya jumlah komputasi yang harus dilakukan pada algoritma ini [JAC-02].

2.12 *Fuzzy k*-Nearest *Neighbour*

Jika *k-NN* melakukan prediksi secara tegas pada data uji berdasarkan perbandingan *k* tetangga terdekat, maka ada pendekatan lain yang dalam melakukan prediksi juga berdasarkan *k* tetangga terdekat tapi tidak secara tegas memprediksi kelas yang harus diikuti oleh data uji, pemberian label kelas data uji pada setiap kelas dengan memberikan nilai keanggotaan seperti halnya teori himpunan *fuzzy*. Algoritma *Fuzzy k-Nearest Neighbour* diperkenalkan oleh Keller dengan mengembangkan *k-NN* yang digabungkan dengan teori *fuzzy* dalam memberikan definisi pemberian label kelas pada data uji yang diprediksi [PRA-12].

Dasar dari algoritma ini adalah pemberian nilai *membership* sebagai fungsi pola jarak / kesamaan dari sejumlah himpunan *k-NN* dan pemberian nilai keanggotaan *neighbour* pada kelas tertentu. Sehingga pada algoritma ini, data *testing* yang akan diklasifikasikan akan memiliki nilai keanggotaan pada semua kelas. Klasifikasi algoritma ini nantinya akan memilih nilai keanggotaan kelas pada data *testing* x $(\mu_i(x))$ yang paling tinggi [ZHA-09]. Berikut ini merupakan persamaan untuk memberikan nilai keanggotaan pada data *testing* x, dengan menggunakan metode *euclidean distance*.

Jarak *euclidean* antara vektor data *testing* x, dan vektor data *training* x_j dihitung menggunakan persamaan 2.4.

$$\|x - x_j\| = (\sum_{l=1}^{n} |N_l - N_l^j|^2)^{1/2} \quad (2.4)$$

Keterangan :

$\|x - x_j\|$: jarak *euclidean* vektor data *testing* dan dokumen *training* ke-*j*

N_l : bobot *term* dalam dokumen *testing*

N_l^j : bobot *term* dalam dokumen *training* ke-*j*

l : index *term* ke-l

n : jumlah *term* keseluruhan hasil *text preprocessing*

Kemudian penentuan nilai keanggotaan kelas ke- i pada tetangga terdekat ke- j dari himpunan *k-NN* dihitung menggunakan persamaan 2.5. Jika tetangga milik kelas ke- i maka perhitungan nilai keanggotaan memakai aturan $j = i$. Namun jika tetangga bukan milik kelas ke- i maka perhitungan memakai aturan $j \neq i$.

$$\mu_{ij} = \begin{cases} 0{,}51 + (\frac{n_j}{k}) * 0{,}49 & , j = i \\ (\frac{n_j}{k}) * 0{,}49 & , j \neq i \end{cases} \tag{2.5}$$

Keterangan :

μ_{ij} : nilai keanggotaan kelas ke- i pada tetangga ke- j

n_j : jumlah tetangga yang termasuk kelas-*j*

i : kelas *genre*

k : jumlah *k- nearest neighbour*

Kemudian penentuan nilai *membership* kelas pada data *testing* x dengan metode *euclidean distance* dihitung menggunakan persamaan 2.6.

$$\mu_i(x) = \frac{\sum_{j=1}^{k} \mu_{ij} \left(1 \Big/ \|x - x_j\|^{2/(m-1)}\right)}{\sum_{j=1}^{k} \left(1 \Big/ \|x - x_j\|^{2/(m-1)}\right)} \tag{2.6}$$

Keterangan :

$\mu_i(x)$: nilai keanggotaan kelas ke-*i* pada data *testing* x

m : bobot pangkat (*weight exponent*)

Sedangkan untuk memberikan nilai keanggotaan pada data *testing* x, dengan menggunakan metode *cosine similarity* dihitung menggunakan persamaan 2.7.

$$\mu_i(x) = \frac{\sum_{j=1}^{k} \mu_{ij}(sim(q,d))}{\sum_{j=1}^{k} (sim(q,d))} \quad (2.7)$$

Dimana $\mu_i(x)$ adalah nilai keanggotaan kelas pada data *testing*, μ_{ij} adalah nilai keanggotaan kelas ke- i pada tetangga ke- j yang telah dihitung menggunakan persamaan 2.5, dan $sim(q,d)$ adalah kemiripan (*similarity*) antara data *testing* q dengan data *training* d yang telah dihitung menggunakan persamaan 2.3.

Penentuan nilai keanggotaan data *testing* dipengaruhi oleh jarak atau kemiripan vektor data *testing* dengan vektor data *training* pada himpunan *k-NN* yang terbentuk, dan nilai keanggotaan kelasnya. Nilai jarak atau kemiripan akan menghasilkan nilai keanggotaan yang lebih besar jika nilai jaraknya makin kecil atau nilai kemiripannya makin besar. Sebaliknya nilai keanggotaan menjadi lebih kecil jika nilai jaraknya makin besar atau nilai kemiripannya makin kecil.

Algoritma FK-NN ini masih mempunyai beberapa kelemahan, salah satunya yaitu keakuratannya tergantung pada pemilihan nilai *k*. Ketika pemilihan nilai *k* terlalu kecil akan menghasilkan akurasi yang rendah karena hasil dari klasifikasi hanya tergantung pada sejumlah *k* dan tidak memperhatikan besar data latih yang tersedia. Sedangkan pemilihan nilai *k* yang terlalu tinggi juga akan menyebabkan akurasi rendah karena semakin banyak data yang tidak relevan (*noise*), sehingga hasil kategorisasi dokumen baru lebih terpengaruh dengan kelas yang lebih besar [WIS-13].

2.13 Algortima Klasifikasi Sinopsis dengan *Fuzzy k-Nearest Neighbour*

Tabel 2.10 Algoritma *Fuzzy k-NN*

Algorithm : FK-NN_Classification (w, x)
Input: sekumpulan data *training* D = {d_1, d_2 ,.., $d_{|D|}$} label kelas yang berasosiasi C(dj) = {c_1, c_2 ,.., $c_{|C|}$}, data *testing* d ;
1. Set x dan k , | | 1 ≤ k ≤D | |
2. Preprocessing D
3. Hitung *vocabulary / term set* {t_1, t_2 ,.., t_n},yang mengandung n kata yang berbeda yang muncul pada *training* dokumen.
4. Set x_j sebagai vektor fitur dari *dokumen* yang mengandung dokumen latih, x_j dapat direpresentasikan sebagai : $x_j = (N_1^j, N_2^j, \dots, N_n^j)$.
5. Set x sebagai vektor fitur dari data *testing* d, yang diekspresikan sebagai $x = (N_1, N_2, \dots, N_n)$.
6. Cari *K-Nearest Neighbour* dari x menurut jarak dari x ke vector fitur dari document latih, set x_1, x_2 ,.., x_k sebagai *K-Nearest Neighbour* dari x ;
7. Inisialisaisi i = 1
8. *Do until* (data testing x mempunyai *membership* di semua kelas)
9. Hitung $\mu_i(x)$ menggunakan rumus 2.6 dan/atau 2.7
10. i ← i + 1
11. *End do*
12. Klasifikasi x pada kelas dengan nilai terbesar $\mu_i(x)$

2.14 Evaluasi

Tujuan evaluasi percobaan pada klasifikasi (*classifier*) yaitu untuk mengukur keefektifan apakah sistem mengklasifikasi secara benar. Evaluasi biasanya membutuhkan sebuah matriks yang disebut berupa *matriks confusion*. *Matriks confusion* adalah matriks yang berisi tentang informasi mengenai hasil klasifikasi oleh sistem pengklasifikasi dan klasifikasi yang sebenarnya. Evaluasi standar yang biasa dilakukan adalah *precision* dan *recall*, sedangkan kombinasi dari kedua evaluasi tersebut adalah *F-measure*. *Matriks confusion* ditunjukkan pada tabel 2.11.

Tabel 2.11 *Matriks Confusion*

Ck	*Classifier positive label*	*Classifier negative label*
True positive label	A	B
True negative label	C	D

Dari matriks di atas (tabel 2.11) menunjukkan bahwa jika diberikan kategori Ck, parameter A adalah jumlah dokumen yang berhasil dikategorikan oleh sistem ke dalam kategori Ck, parameter B adalah jumlah dokumen yang mempunyai kategori Ck namun sistem tidak mengklasifikasikannya ke dalam kategori Ck, parameter C adalah jumlah dokumen yang bukan kategori Ck namun sistem mengklasifikasikannya ke dalam kategori Ck, dan parameter D adalah jumlah dokumen yang tidak temasuk kategori Ck dan sistem juga tidak mengklasifikasikannya ke dalam kategori Ck.

Recall adalah ukuran keberhasilan sistem dalam mengenali dokumen pada setiap kategori tanpa melihat ketepatan klasifikasi yang dilakukan. *Recall* dihitung menggunakan persamaan 2.8.

$$Recall = \frac{A}{A + B} \tag{2.8}$$

Precision adalah ukuran keberhasilan sistem dalam melakukan ketepatan klasifikasi tanpa melihat seberapa banyak dokumen yang berhasil dikenali. *Precision* dihitung menggunakan persamaan 2.9.

$$Precision = \frac{A}{A + C} \tag{2.9}$$

F-measure mewakili pengaruh relatif antara *precision* dan *recall*, yang dihitung dengan persamaan berikut. *F-measure* dihitung menggunakan persamaan 2.10.

$$F\text{-}measure = \frac{2 \,.\, recall \,.\, precision}{(recall + precision)} \quad (2.10)$$ [SOU-05]

BAB III
METODOLOGI DAN PERANCANGAN

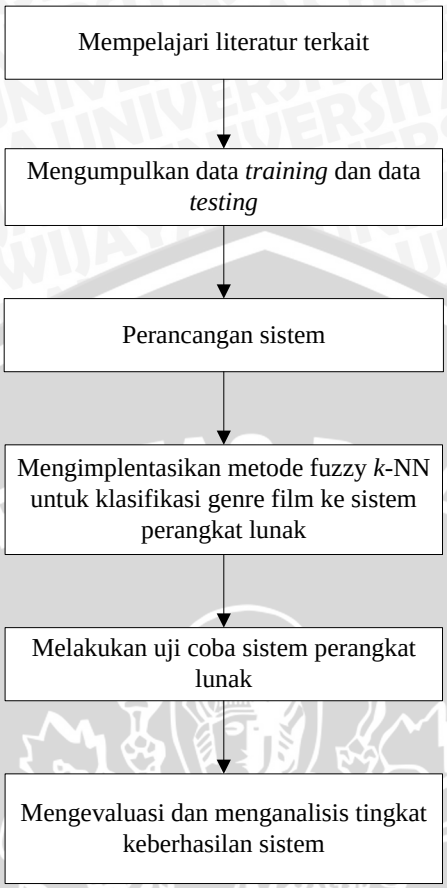
Pada bab metodologi dan perancangan ini akan diberikan penjelasan mengenai metode dan langkah-langkah perancangan yang dilakukan untuk pembuatan sistem klasifikasi *genre* film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN*.

3.1 Metodologi

Adapun metodologi dalam penelitian ini adalah sebagai berikut :

1. Melakukan studi literatur yang berkaitan dengan penelitian.
2. Mengumpulkan data penelitian, yaitu berupa judul dan sinopsis dari film untuk data *training* dan data *testing*.
3. Melakukan perancangan sistem klasifikasi *genre* film menggunakan metode *Fuzzy k-NN*.
4. Mengimplementasikan hasil perancangan yang telah dilakukan sebelumnya menjadi sebuah sistem klasifikasi *genre* film otomatis dengan menggunakan data *training* yang telah disiapkan dalam pengenalan ke sebuah sistem perangkat lunak.
5. Melakukan pengujian terhadap sistem perangkat lunak menggunakan data *testing* berupa judul dan sinopsis dari film yang baru. Hasil yang diperoleh adalah mengetahui termasuk dalam *genre* apakah film yang judul dan sinopsisnya telah di-*testing* oleh sistem perangkat lunak.
6. Mengevaluasi dan menganalisis tingkat keberhasilan sistem yaitu dengan membandingkan hasil pengklasifikasian yang dilakukan oleh sistem dengan hasil pengkategorian dari sumber data. Tingkat keberhasilan sistem diukur dari nilai *recall*, *precision*, dan *F-measure.*

Alur penelitian ditunjukkan pada gambar 3.1.



Gambar 3.1 Gambar Alur Penelitian

3.1.1 Studi Literatur

Pada penelitian ini dibutuhkan studi literatur untuk merealisasikan tujuan dan penyelesaian masalah. Teori-teori mengenai definisi film beserta *genre*, konsep *text mining*, klasifikasi teks, *text preprocessing*, *weighting*, dan metode *Fuzzy k-NN* yang digunakan sebagai dasar penelitian, diperoleh dari sumber – sumber atau buku - buku referensi yang berkaitan dengan skripsi, jurnal ataupun *browsing* dari internet.

3.1.2 Analisis Data

Pada penelitian ini data yang digunakan berupa judul dan sinopsis sejumlah film dari berbagai jenis *genre* berbahasa Inggris yang diambil dari sebuah situs yang memuat informasi mengenai film dalam skala internasional, yaitu http://www.imdb.com. Sumber tersebut merupakan salah satu dari situs

penyedia informasi mengenai film secara mendetail yang dapat dipercaya, hingga saat ini situs tersebut masih aktif dalam memberi informasi mengenai film secara *online*. Oleh karena itu, situs tersebut dapat dijadikan referensi dalam penelitian ini. Data kemudian dikumpulkan secara random dan merata sesuai dengan *genre* yang disediakan pada penelitian ini. Hal ini dimaksudkan untuk memperoleh data *training* yang tepat dan untuk mempermudah pengujian dan keakuratan pada data *testing*.

Data kemudian disimpan pada file dokumen berformat *file text* dengan ekstensi txt (*.txt). Untuk format penyimpanan teks data, baris pertama merupakan judul data sinopsis, baris kedua merupakan *genre* dari data sinopsis, dan baris ketiga hingga baris terakhir merupakan bagian dari isi teks sinopsis secara utuh.

3.2 Analisis dan Perancangan Sistem

Pada sub-bab ini akan dibahas mengenai semua hal yang diperlukan dalam proses pembuatan sistem pengklasifikasian film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN*.

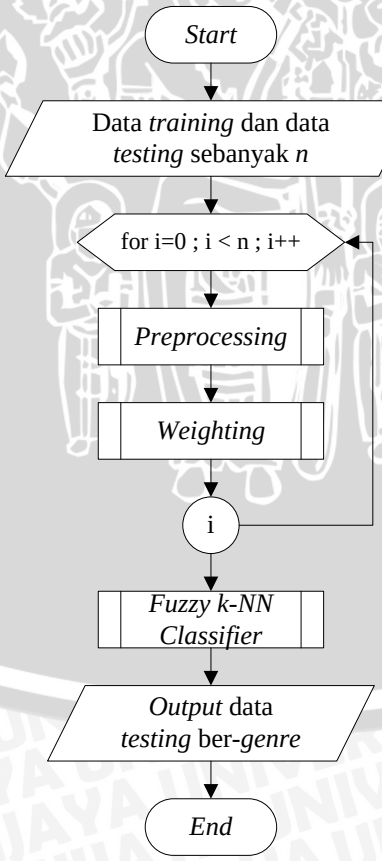
3.2.1 Deskripsi Sistem

Sistem yang akan dibuat merupakan sistem untuk mengklasifikasi film berdasarkan judul dan sinopsis berbahasa Inggris kedalam satu atau lebih jenis genre dengan menggunakan metode *Fuzzy k-NN* secara otomatis pada file dokumen berformat *file text* dengan ekstensi txt (*.*txt*). Pengklasifikasian dibuat berdasarkan unit terkecil dari dokumen atau disebut dengan kata. Data yang dibutuhkan dalam sistem ini dibagi menjadi dua jenis, yaitu data *training* dan data *testing*. Data *training* merupakan judul dan sinopsis dari film yang akan dijadikan pembanding data *testing* sehingga dapat ditentukan jenis *genre* filmnya, sedangkan data *testing* merupakan judul dan sinopsis dari film yang akan dikategorikan kedalam *genre* yang tersedia. Proses yang dilakukan adalah :

1. *User* memasukkan sejumlah data *training* dan data *testing*.

2. Sistem akan menjalankan proses *Preprocessing* dan *Weighting*, dimana semua data *training* disiapkan sehingga menjadi vektor data *training* dan siap diolah untuk proses selanjutnya.
3. Data *testing* berupa judul dan sinopsis dari film yang akan diklasifikasikan menjalani proses *Fuzzy k-NN Classifier*, namun sebelumnya data *testing* menjalani proses *Preprocessing* dan *Weighting* sehingga menjadi vektor data *testing*.
4. Berdasarkan hasil yang diperoleh pada proses *Preprocessing*, dan *Weighting* selanjutnya akan dilakukan proses *Fuzzy k-NN Classifier*. Proses ini bertujuan untuk mengklasifikasi sejumlah sinopsis film kedalam kategori. Algoritma yang digunakan yaitu *Fuzzy k-NN*. Hasil akhir adalah film yang diklasifikasikan kedalam *genre* yang ada sesuai teori pada bab sebelumnya.

Secara garis besar, deskripsi sistem ditunjukkan pada gambar 3.2



Gambar 3.2 *Flowchart* Deskripsi Sistem

3.3 Perancangan Sistem

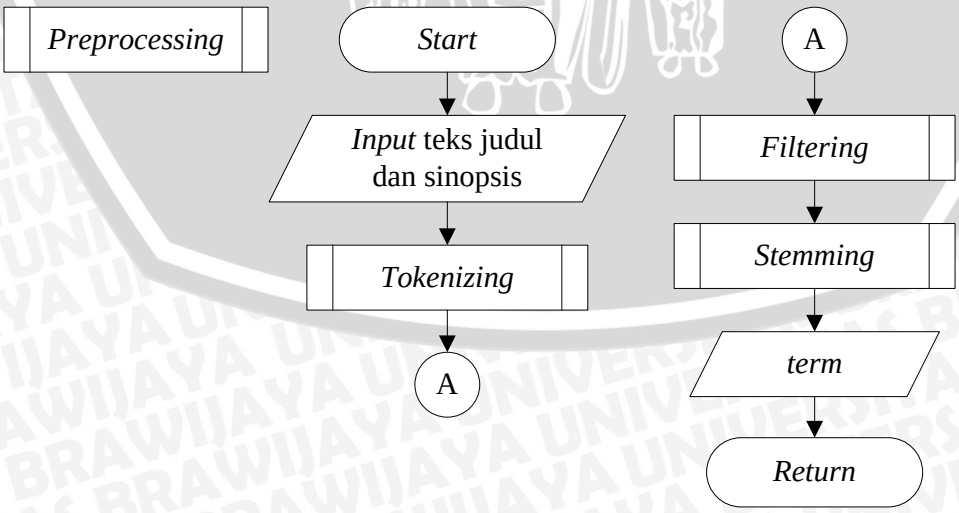
Pada sub bab perancangan sistem akan dijelaskan mengenai tahapan atau proses-proses dalam membangun sistem antara lain proses *Preprocessing*, *Weighting*, dan *Fuzzy k-NN Classifier*.

3.3.1 Perancangan Proses *Preprocessing*

Pada tahap ini dilakukan *preprocessing* terhadap seluruh dokumen yang berisi judul beserta sinopsis film, baik data *training* maupun data *testing*. Proses yang dilakukan adalah sebagai berikut :

1. Data *training* dan data *testing* yang berisi judul dan sinopsis film diambil dari direktori.
2. Tahap *tokenizing*, meliputi mengubah semua huruf menjadi huruf kecil (*case folding*), menghilangkan semua karakter spesial dan angka, serta pemecahan teks menjadi *token*.
3. Tahap *filtering*, yaitu penghapusan kata yang sering muncul namun tidak merepresentasikan isi dalam teks (*stopword*) yang terdapat pada *token*.
4. Tahap *stemming*, yaitu pembentukan kata dasar dari kata yang telah melalui proses *filtering*.
5. Hasil akhir dari proses *preprocessing* berupa kata dasar.

Tahapan perancangan proses *preprocessing* ditunjukkan pada gambar 3.3.



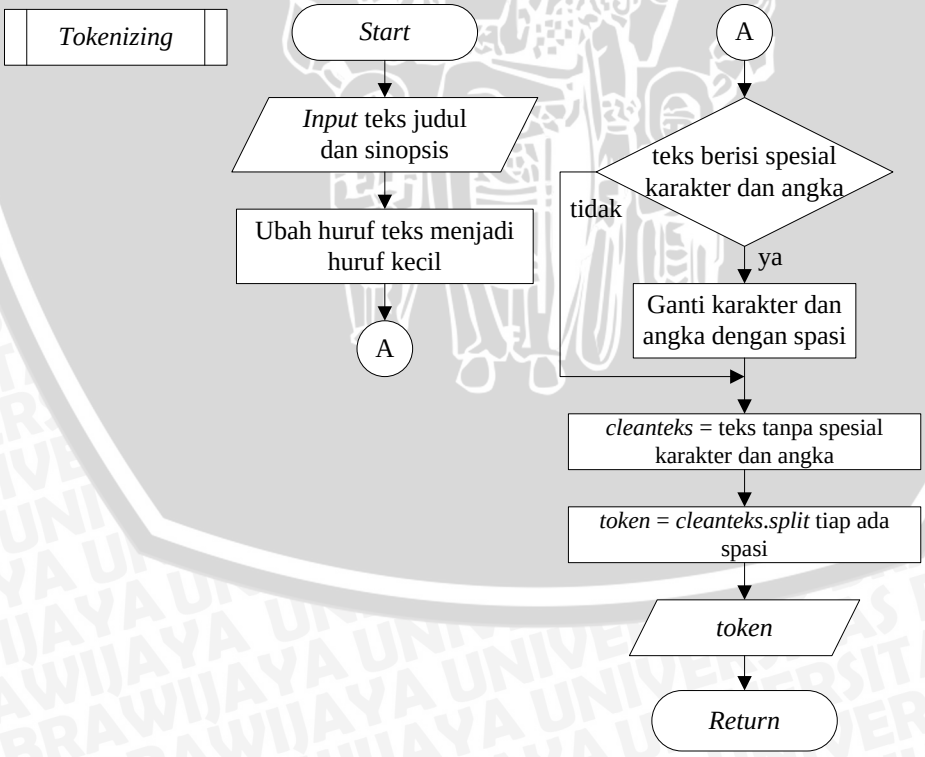
Gambar 3.3 *Flowchart* Proses *Preprocessing*

3.3.1.1 Perancangan Proses *Tokenizing*

Tokenizing adalah proses untuk menguraikan dokumen utuh menjadi daftar kata (*token*). Input yang diperlukan dalam proses ini adalah dokumen yang berisi judul beserta sinopsis film. Hasil akhir dari proses ini adalah daftar kata (*token*) yang sudah dipisahkan dari spesial karakter dan karakter angka. Secara spesifik, proses yang dilakukan adalah sebagai berikut :

1. Dokumen yang berisi judul dan sinopsis film diambil dari direktori.
2. Huruf pada isi dokumen diubah menjadi bentuk huruf kecil.
3. Dilakukan pengecekan jika huruf merupakan spesial karakter atau angka maka huruf tersebut akan diganti dengan spasi, hingga didapatkan teks tanpa spesial karakter dan angka yang ditampung pada variabel cleanteks.
4. Selanjutnya proses *split* pada *cleanteks* setiap menemukan spasi untuk memisahkan kata.
5. Hasil akhir adalah *token* hasil proses *tokenizing*.

Tahapan perancangan proses *tokenizing* ditunjukkan pada gambar 3.4.



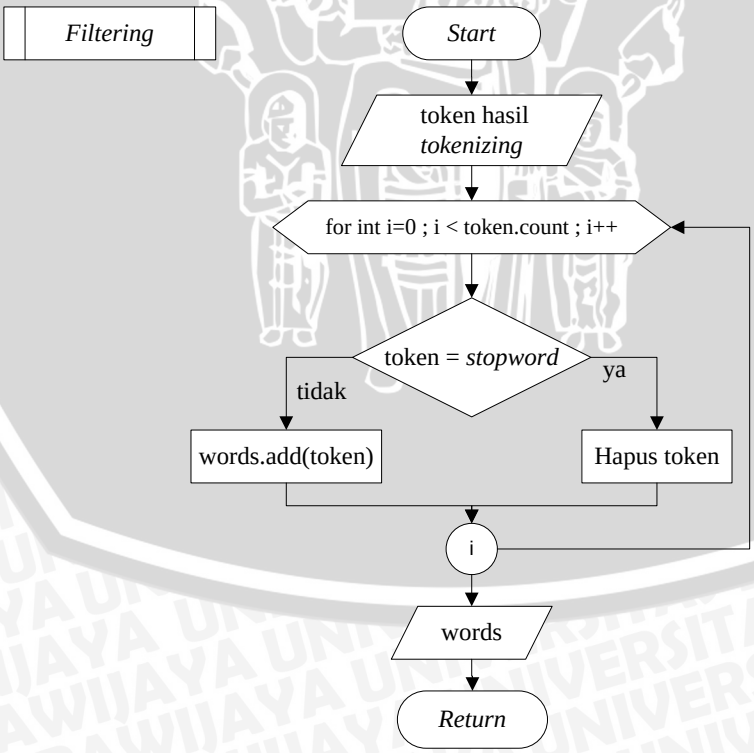
Gambar 3.4 *Flowchart* Proses *Tokenizing*

3.3.1.2 Perancangan Proses *Filtering*

Filtering adalah proses yang berfungsi untuk memilih kata yang merepresentasikan dokumen. Jika kata termasuk salah satu dari kata dalam daftar *stopword*, maka kata tersebut akan dihapus karena dianggap tidak merepresentasikan isi dokumen. Input yang diperlukan dalam proses ini adalah *token* yang dihasilkan dari proses *tokenizing*. Hasil akhir dari proses ini adalah *token* tanpa *stopword*. Secara spesifik, proses yang dilakukan adalah sebagai berikut :

1. *Token* hasil *tokenizing* diambil sebagai inputan proses *filtering*.
2. Dilakukan perulangan sebanyak *token* hasil *tokenizing*.
3. Dilakukan pengecekan jika token termasuk *stopword*, maka kata tersebut akan dihapus. Dan jika *token* bukan *stopword* maka ditampung kedalam variabel words.
4. Hasil akhir adalah token hasil *filtering*, yang disimpan pada variabel words.

Tahapan perancangan proses *filtering* ditunjukkan pada gambar 3.5.



Gambar 3.5 *Flowchart* Proses *Filtering*

3.3.1.3 Perancangan Proses *Stemming*

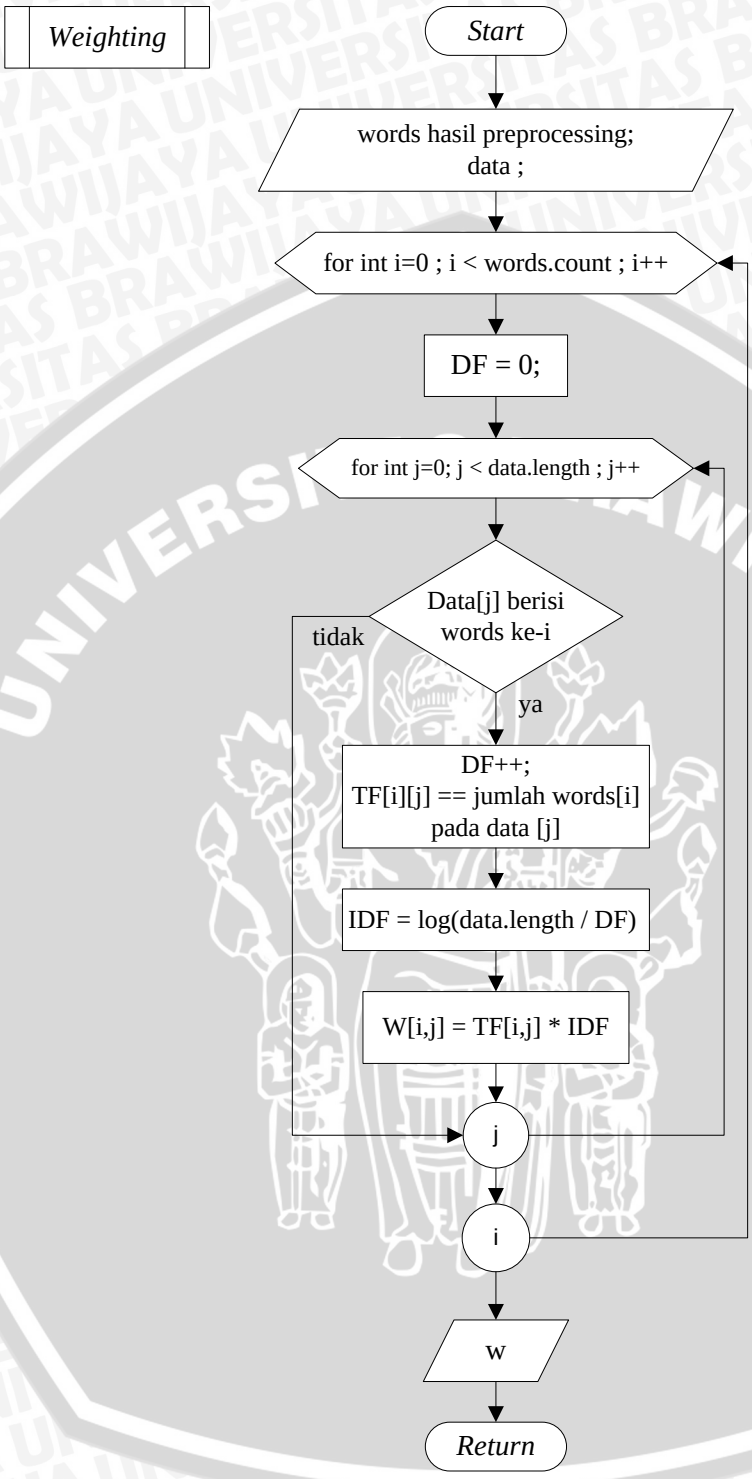
Stemming merupakan proses yang dilakukan untuk mengubah semua *termlist* yang telah diproses pada tahap sebelumnya menjadi bentuk kata dasarnya. Proses *stemming* dilakukan menggunakan algoritma *Porter stemmer* yang telah dikembangkan Martin F. Porter. Tahapan proses *stemming* dapat dilihat pada subbab 2.7.3.1.

3.3.2 Perancangan Proses *Weighting*

Proses *weighting*, adalah proses untuk menentukan nilai bobot *term* yang telah dihasilkan pada proses *preprocessing*. Data yang semula berupa *string* akan dikonversi menjadi data numerik sehingga nilai bobot tersebut bisa digunakan pada proses komputasi selanjutnya. Input yang dibutuhkan adalah words hasil preprocessing, dan sejumlah data penelitian. Hasil akhir dari proses ini adalah bobot *term* pada setiap data. Secara spesifik, proses yang dilakukan adalah sebagai berikut :

1. Inisialisasi awal input proses, yaitu words hasil *preprocessing*, dan sejumlah data.
2. Dilakukan proses perulangan sebanyak words hasil preprocessing, dan perulangan sebanyak jumlah data. Nilai DF diinisialisasi = 0.
3. Dilakukan proses pengecekan, jika data ke-j berisi words ke-i maka nilai DF akan ditambah 1. TF adalah frekuensi words yang ada pada tiap data.
4. Dilakukan proses perhitungan *inverse document frequency* sesuai persamaan 2.1.
5. Dilakukan proses perhitungan bobot *term* sesuai dengan persamaan 2.2.
6. Proses diulangi hingga indeks words yang paling akhir.
7. Hasil akhir didapatkan bobot *term* pada setiap data.

Tahapan perancangan proses *weighting* ditunjukkan pada gambar 3.6.



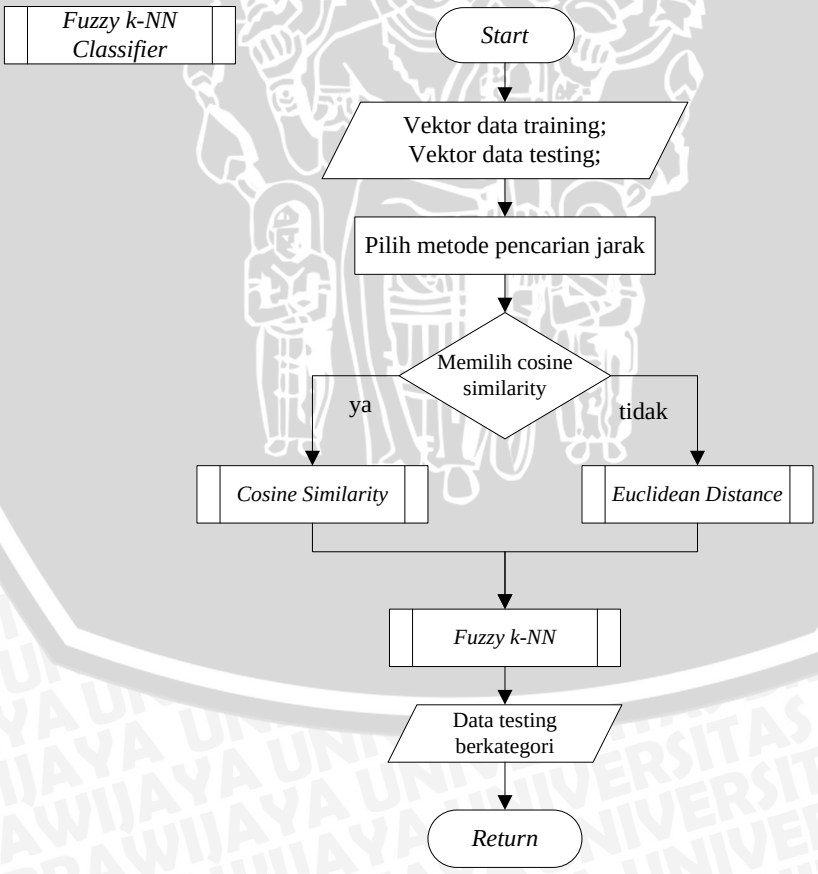
Gambar 3.6 *Flowchart* Proses *Weighting*

3.3.3 Perancangan Proses *Fuzzy k-NN Classifier*

Proses *fuzzy k-NN classifier*, adalah proses untuk menentukan klasifikasi untuk data *testing* dengan menggunakan algoritma *Fuzzy k-Nearest Neighbour*. Secara spesifik, proses yang dilakukan adalah sebagai berikut :

1. Inisialisasi awal inputan proses yaitu vektor data *training*, dan vektor data *testing*.
2. Memilih metode pencarian jarak yang digunakan pada penelitian ini yaitu *cosine similarity* atau *euclidean distance*.
3. Dilakukan proses komputasi *Fuzzy k-Nearest Neighbour* sesuai dengan rumus dan algoritmanya
4. Hasil akhir dari proses *fuzzy k-NN classifier* adalah data testing yang telah diklasifikasi kedalam kategori *genre* yang tersedia.

Tahapan perancangan proses *fuzzy k-NN classifier* ditunjukkan pada gambar 3.7.



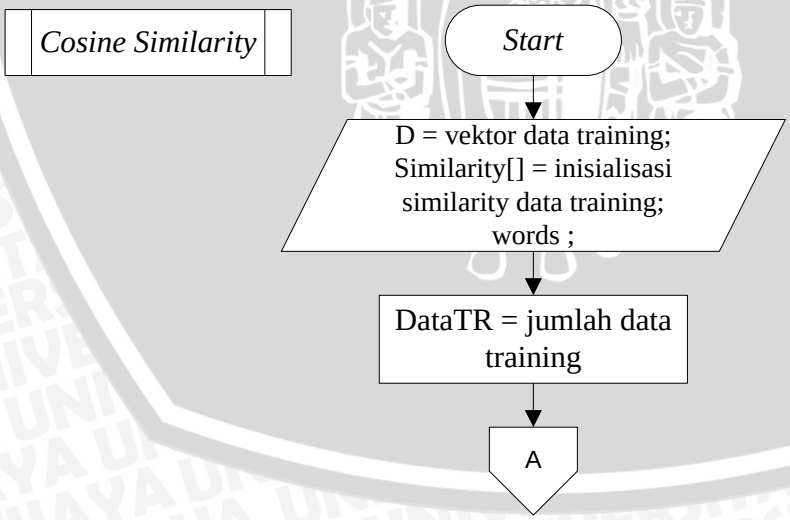
Gambar 3.7 *Flowchart* Proses *Fuzzy k-NN Classifier*

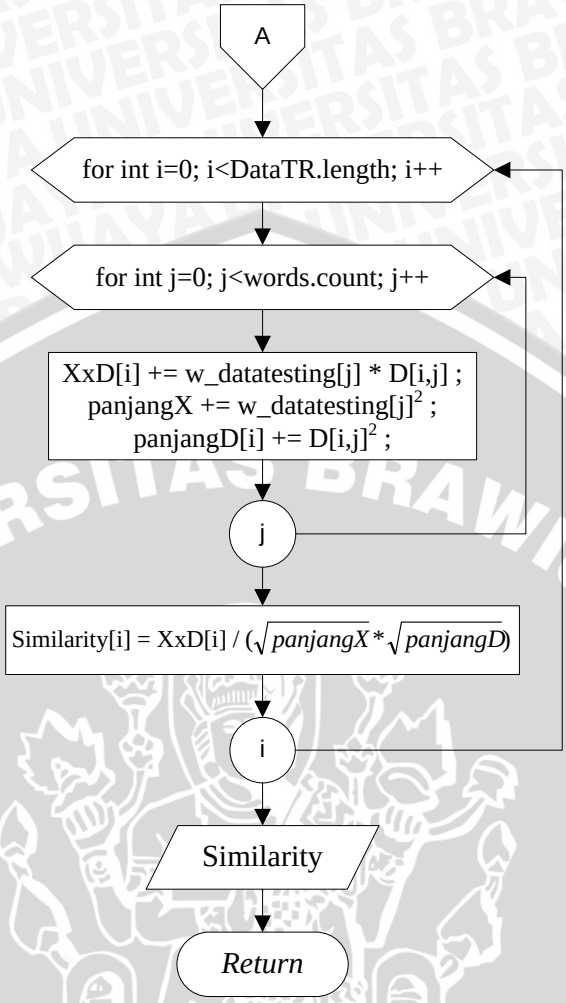
3.3.3.1 Perancangan *Cosine Similarity*

Proses *cosine similarity* adalah proses untuk menghitung kemiripan vektor antara data *testing* dengan keseluruhan data *training* dengan memanfaatkan metode *cosine similarity*. Kemiripan vektor dihitung menggunakan persamaan 2.3. Input yang diperlukan untuk proses ini adalah nilai bobot *term* pada tiap dokumen (w) dan words hasil *preprocessing*. Hasil akhir proses ini adalah nilai kemiripan antara vektor data *testing* dan vektor data *training*. Secara spesifik, proses yang dilakukan sebagai berikut :

1. Inisialisasi awal inputan proses yaitu vektor data *training*, words hasil *preprocessing*, dan variabel Similarity untuk menampung hasil komputasi.
2. Dilakukan perulangan sebanyak jumlah data *training*, dan sebanyak jumlah words hasil *preprocessing*.
3. Dilakukan proses perhitungan *cosine similarity* sesuai dengan persamaan 2.3.
4. Hasil akhir dari proses adalah adalah nilai kemiripan antara vektor data *testing* dan vektor data *training* yang disimpan pada variabel Similarity.

Tahapan perancangan proses *cosine similarity* ditunjukkan pada gambar 3.8.





Gambar 3.8 *Flowchart* Proses *Cosine Similarity*

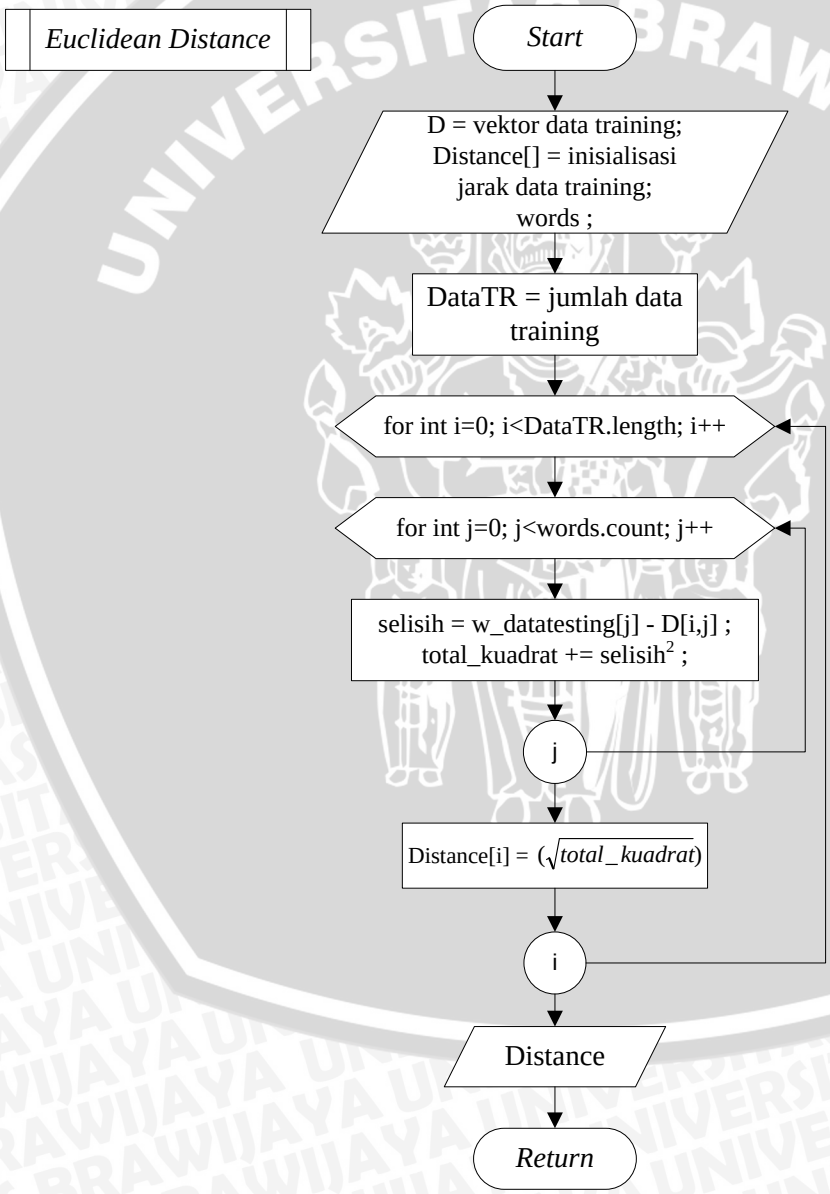
3.3.3.2 Perancangan *Euclidean Distance*

Proses *euclidean distance* adalah proses untuk menghitung jarak vektor antara data *testing* dengan keseluruhan data *training* dengan memanfaatkan metode *euclidean distance*. Jarak vektor dihitung menggunakan persamaan 2.4. Input yang diperlukan untuk proses ini adalah nilai bobot *term* pada tiap dokumen (w) dan words hasil *preprocessing*. Hasil akhir proses ini adalah nilai jarak antara vektor data *testing* dan vektor data *training*. Secara spesifik, proses yang dilakukan sebagai berikut :

1. Inisialisasi awal inputan proses yaitu vektor data *training*, words hasil *preprocessing*, dan variabel Distance untuk menampung hasil komputasi.

2. Dilakukan perulangan sebanyak jumlah data *training*, dan sebanyak jumlah words hasil *preprocessing*.
3. Dilakukan proses perhitungan *euclidean distance* sesuai dengan persamaan 2.4.
4. Hasil akhir dari proses ini adalah adalah nilai jarak antara vektor data *testing* dan vektor data *training* yang disimpan pada variabel Distance.

Tahapan perancangan proses *euclidean distance* ditunjukkan pada gambar 3.9.



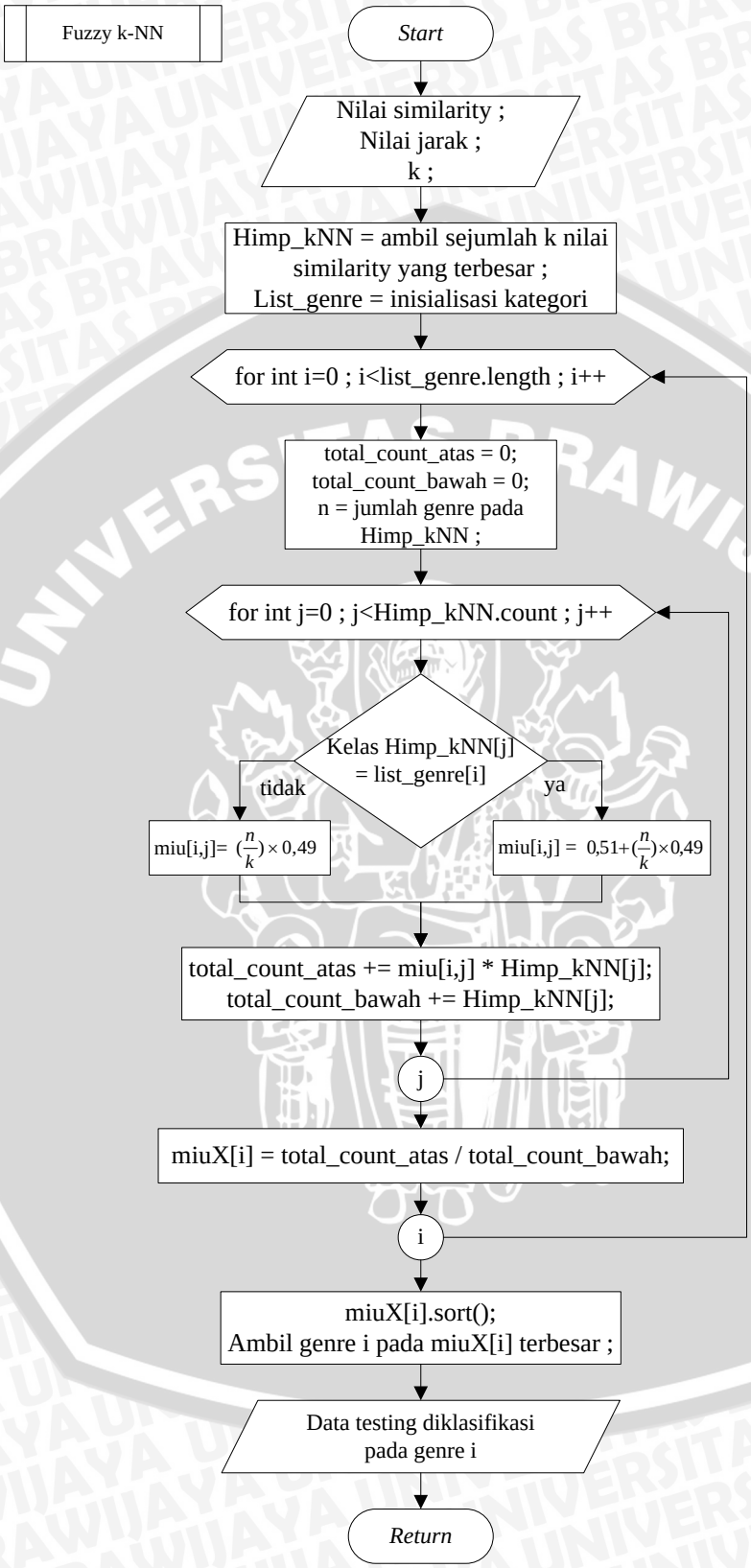
Gambar 3.9 *Flowchart* Proses *Euclidean Distance*

3.3.3.3 Perancangan Proses *Fuzzy k-NN*

Proses *Fuzzy k-NN* adalah proses pemberian nilai keanggotaan vektor data *testing* dengan menggunakan metode *Fuzzy k-Nearest Neighbour*, dimana pada data *testing* nantinya akan memiliki nilai keanggotaan di semua kelas kategori. Input yang diperlukan untuk proses ini adalah nilai kemiripan (*similarity*) antara vektor data *testing* dan data *training* jika metode *cosine similarity* yang digunakan, atau nilai jarak antara vektor data *testing* dan data *training* jika metode *euclidean distance* yang digunakan. Selain itu nilai *k* yang merupakan parameter jumlah tetangga terdekat juga digunakan sebagai inputan pada proses ini. Hasil akhir proses ini adalah diperoleh nilai keanggotaan kelas kategori yang memiliki nilai tertinggi. Nilai keanggotaan kelas kategori tertinggi pada data *testing* tersebut merepresentasikan bahwa data tersebut diklasifikasikan sebagai bagian dari kelas kategori tersebut. Secara spesifik, proses yang dilakukan sebagai berikut :

1. Inisialisasi awal inputan proses yaitu nilai similarity, nilai jarak, dan nilai *k* (jumlah tetangga terdekat).
2. Mengambil himpunan *k-NN*, yaitu sejumlah *k* tetangga terdekat yang mempunyai nilai similarity terbesar.
3. Dilakukan perulangan sebanyak kategori yang digunakan pada penelitian ini, dan sebanyak himpunan *k-NN*. Nilai variabel total_count_atas, dan total_count_bawah diinisialisasi = 0.
4. Dilakukan proses komputasi nilai keanggotaan kelas ke-i pada tetangga terdekat sesuai dengan persamaan 2.5.
5. Dilakukan proses komputasi nilai keanggotaan data *testing* pada kelas ke-i sesuai dengan persamaan 2.6 jika yang digunakan adalah metode *euclidean distance*, atau persamaan 2.7 jika yang digunakan adalah metode *cosine similarity*.
6. Hasil akhir dari proses ini adalah data testing diklasifikasi pada kelas kategori, yang mempunyai nilai keanggotaan data *testing* yang paling besar.

Tahapan perancangan proses *fuzzy k-NN* ditunjukkan pada gambar 3.10.



Gambar 3.10 *Flowchart* Proses *Fuzzy k-NN*

3.4 Perhitungan Manual

3.4.1 Sumber Data Perhitungan Manual

Data yang digunakan dalam perhitungan manual berjumlah 11 yang terdiri dari 10 data *training* dan 1 data *testing*. Dokumen yang digunakan pada perhitungan manual sudah melalui proses *text preprocessing*. Dokumen teks yang digunakan pada perhitungan manual ini terdapat pada lampiran 2.

3.4.2 Perhitungan *Term Weighting* (*TF-IDF*)

Setelah didapatkan frekuensi kemunculan kata pada data *training* dan data *testing*, proses selanjutnya adalah menentukan nilai frekuensi kemunculan dokumen yang mengandung kata yang bersangkutan untuk semua data *training* (DF). Nilai tersebut nantinya digunakan untuk menghitung *inverse document frequency* ($IDF(t_i)$) menggunakan persamaan 2.1.

Proses perhitungan dimulai untuk kata pada index kata i = 1 yaitu kata "agora". Data *training* berjumlah 10, oleh karena itu nilai D = 10. Kata "agora" muncul sebanyak 1 kali, masing-masing pada data *training* d1. Oleh karena itu nilai $DF(t_1)$ untuk kata "block" adalah 1. Contoh perhitungannya adalah sebagai berikut.

$$IDF(t_1) = \log(\frac{10}{1}) = 1$$

Daftar frekuensi kemunculan kata beserta hasil perhitungan *inverse document frequency* ($IDF(t_i)$) ditampilkan pada tabel 3.1.

Langkah selanjutnya adalah menghitung bobot dari masing-masing kata pada tiap-tiap dokumen menggunakan persamaan 2.2. Proses perhitungan dimulai untuk kata pada index kata i = 1 yaitu kata "agora". Bobot kata "agora" akan dihitung pada keseluruhan dokumen. Contoh perhitungannya adalah sebagai berikut.

$$w_{1,d1} = TF(t_1, d1) \times IDF(t_1) = 1 \times 1 = 1$$

Hasil perhitungan bobot kata ditampilkan pada tabel 3.2.

Tabel 3.1 Data Frekuensi Kemunculan *Term* dan Hasil Perhitungan *Inverse Document Frequency* (*IDF*)

No	*Term*	*TF*											*D*	*DF*	*IDF*
		d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	X			
1	agora	1	0	0	0	0	0	0	0	0	0	0	10	1	1
2	alexandria	1	0	0	0	0	0	0	0	0	0	0	10	1	1
3	ad	1	0	0	0	0	0	0	0	0	0	0	10	1	1
4	hypatia	2	0	0	0	0	0	0	0	0	0	0	10	1	1
5	teach	1	0	0	0	0	0	0	0	0	0	0	10	1	1
6	astronomi	1	0	0	0	0	0	0	0	0	0	0	10	1	1
7	mathemat	1	0	0	0	0	0	0	0	0	0	0	10	1	1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
443	spacecraft	0	0	0	0	0	0	0	0	0	0	1	10	0	-
444	aid	0	0	0	0	0	0	0	0	0	0	1	10	0	-
445	robot	0	0	0	0	0	0	0	0	0	0	1	10	0	-
446	villain	0	0	0	0	0	0	0	0	0	0	1	10	0	-
447	shockwav	0	0	0	0	0	0	0	0	0	0	1	10	0	-
448	scene	0	0	0	0	0	0	0	0	0	0	1	10	0	-
449	battl	0	0	0	0	0	0	0	0	0	0	1	10	0	-

Tabel 3.2 Hasil Perhitungan Bobot *Term* Dalam Dokumen

No	*Term*	w_i										
		d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	X
1	agora	1	0	0	0	0	0	0	0	0	0	0
2	alexandria	1	0	0	0	0	0	0	0	0	0	0
3	ad	1	0	0	0	0	0	0	0	0	0	0
4	hypatia	2	0	0	0	0	0	0	0	0	0	0
5	teach	1	0	0	0	0	0	0	0	0	0	0
6	astronomi	1	0	0	0	0	0	0	0	0	0	0
7	mathemat	1	0	0	0	0	0	0	0	0	0	0
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
443	spacecraft	0	0	0	0	0	0	0	0	0	0	0
444	aid	0	0	0	0	0	0	0	0	0	0	0
445	robot	0	0	0	0	0	0	0	0	0	0	0
446	villain	0	0	0	0	0	0	0	0	0	0	0
447	shockwav	0	0	0	0	0	0	0	0	0	0	0
448	scene	0	0	0	0	0	0	0	0	0	0	0
449	battl	0	0	0	0	0	0	0	0	0	0	0

3.4.3 Perhitungan *Fuzzy k-NN Classifier*

Proses perhitungan *fuzzy k-NN classifier* terdiri atas 2 tahap, yaitu perhitungan *cosine similarity*, dan perhitungan *Fuzzy k-NN*. Setiap tahap akan dijelaskan perhitungannya sebagai berikut :

3.4.3.1 Perhitungan *Cosine Similarity*

Langkah selanjutnya setelah didapatkan nilai bobot kata tiap dokumen adalah menghitung *similarity* vektor data *testing* dan seluruh data *training* menggunakan persamaan 2.3. Nilai inputan vektor berasal dari nilai bobot *term* yang telah dihitung pada proses perhitungan bobot *term*.

Proses perhitungan *cosine similarity* antara vektor data *testing* dan data *training* ditunjukkan pada proses berikut, dimana t adalah indeks *term*, $w_{t,d}$ adalah bobot *term* data *training*, dan $w_{t,q}$ adalah bobot *term* data *testing*.

Tabel 3.3 Perhitungan *Cosine Similarity* Vektor Data *Testing* dan Data *Training* ke-1

t	$w_{t,d}$	$w_{t,q}$	$w_{t,d} \cdot w_{t,q}$	$w_{t,d}^2$	$w_{t,q}^2$
1	1	0	0	1	0
2	1	0	0	1	0
3	1	0	0	1	0
4	2	0	0	4	0
5	1	0	0	1	0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
445	0	0	0	0	0
446	0	0	0	0	0
447	0	0	0	0	0
448	0	0	0	0	0
449	0	0	0	0	0
	$\sum_t w_{t,d} \cdot w_{t,q} = 1{,}2506$			$\sum_t w_{t,d}^2 = 71{,}4557$	$\sum_t w_{t,q}^2 = 29{,}2506$

$$sim(q,d) = \frac{\sum_t w_{t,d} \cdot w_{t,q}}{\sqrt{\sum_t w_{t,d}^2} \cdot \sqrt{\sum_t w_{t,q}^2}}$$

$$sim(q,d) = \frac{1{,}2506}{\sqrt{71{,}4557} \cdot \sqrt{29{,}2506}} = \frac{1{,}2506}{45{,}7179} = 0{,}0274$$

Diperoleh hasil perhitungan *cosine similarity* antara vektor data *testing* dan data *training* ke-1 adalah 0,0274. Proses perhitungan *cosine similarity* vektor data *testing* dengan data *training* yang lain dilakukan dengan cara yang sama. Proses akhir perhitungan *cosine similarity* vektor data *testing* dan data *training*, ditunjukkan pada tabel 3.4

Tabel 3.4 Hasil Perhitungan *Cosine Similarity*

Data *Training*	Kategori	*Cosine Similarity*
d1	*Adventure, Drama, History*	0,0274
d2	*Action, Crime*	0,0147
d3	*Western*	0
d4	*Fantasy, Adventure*	0,0189
d5	*Adventure, Drama, Fantasy*	0
d6	*Drama, War*	0
d7	*Horror*	0,0241
d8	*Comedy*	0,01
d9	*Drama, Musical*	0,0176
d10	*Action, Sci-Fi, Adventure*	0,6542

3.4.3.2 Perhitungan *Fuzzy k-NN*

Proses selanjutnya setelah didapatkan nilai *similarity* vektor data *testing* dan keseluruhan data *training* adalah menentukan himpunan *k-NN*. Misal ditentukan *k* = 3, maka himpunan *k-NN* yang terbentuk ditunjukkan pada tabel 3.5

Tabel 3.5 Penentuan Himpunan *k-NN*

Data *Training*	Kategori	*Cosine Similarity*
d10	*Action, Adventure, Sci-Fi*	0,6542
d1	*Adventure, Drama, History*	0,0274
d7	*Horror*	0,0241

Pada algoritma *Fuzzy k-NN*, data *testing* yang akan diklasifikasikan akan diberi nilai keanggotaan pada semua kelas. Pada penelitian ini, jumlah kelas kategori yang digunakan adalah 12 yaitu kelas *action, drama, history, fantasy, sci-fi, horror, comedy, musical, adventure, war,* dan *western*. Sehingga nantinya data *testing* x akan memiliki 12 nilai keanggotaan. Perhitungan nilai keanggotaan x ($\mu i\ (x)$) dihitung menggunakan persamaan 2.5 dan 2.7. Berikut perhitungan manual pemberian nilai keanggotaan kelas data *testing* :

Tabel 3.6 Perhitungan Nilai Keanggotaan Tetangga

Kelas	μ_{ij}	
	$j = i$	$j \neq i$
Action	0,673	0,163
Drama	0,673	0,163
History	0,673	0,163
Fantasy	-	-
Sci-fi	0,673	0,163
Horror	0,673	0,163
Comedy	-	-
Crime	-	-
Musical	-	-
Adventure	0,837	0,327
War	-	-
Western	-	-

Contoh perhitungan nilai keanggotaan kelas *action* pada data *testing x*

$$\mu_{\text{action}}(x) = \frac{\sum_{j=1}^{3} \mu_{\text{action},j}(sim(q,d))}{\sum_{j=1}^{3}(sim(q,d))}$$

$$\mu_{\text{action}}(x) = \frac{(0{,}673 \times 0{,}6542) + (0{,}163 \times 0{,}0274) + (0{,}163 \times 0{,}0241)}{0{,}6542 + 0{,}0274 + 0{,}0241}$$

$$\mu_{\text{action}}(x) = \frac{0{,}4489}{0{,}7057} = 0{,}6361$$

Proses perhitungan nilai keanggotaan pada kelas kategori yang lain dilakukan dengan cara yang sama. Hasil perhitungan nilai keanggotaan semua kelas pada data *testing x* ditunjukkan pada tabel 3.7.

Tabel 3.7 Hasil Perhitungan Nilai Keanggotaan Kelas pada Data *Testing* X

Kelas (i = *genre*)	$\mu_i(x)$
Action	0,6361
Drama	0,0261
History	0,0261
Fantasy	0
Sci-fi	0,6242
Horror	0,0230
Comedy	0
Crime	0
Musical	0
Adventure	0,8081
War	0
Western	0

Sesuai dengan hasil perhitungan nilai keanggotaan kelas untuk data *testing* x, didapatkan nilai $\mu_{adventure}(x)$ adalah yang terbesar. Oleh karena itu, dapat disimpulkan bahwa data *testing* x diklasifikasikan kedalam kelas *adventure*.

3.5 Rancangan Pengujian

Sistem pengujian pada sistem sistem klasifikasi *genre* film berdasarkan judul dan sinopsis dilakukan untuk mengetahui ketepatan hasil klasifikasi yang dihasilkan oleh sistem. Tingkat akurasi sistem diukur berdasarkan *recall* sesuai dengan persamaan 2.8, *precision* sesuai dengan persamaan 2.9, dan *F-measure* sesuai dengan persamaan 2.10.

Pengujian pertama dilakukan untuk mengetahui pengaruh penggunaan jumlah data *training*, dan nilai *k* (jumlah tetangga terdekat) yang digunakan dalam penelitian terhadap nilai tingkat akurasi yang dihasilkan. Selanjutnya akan digunakan beberapa parameter yang berkaitan dengan pengujian ini. Yang pertama adalah paramater jumlah data *training*. Jumlah data *training* yang digunakan pada penelitian ini berjumlah 75, 100, 125, 150, 175, dan 200 buah data. Yang kedua adalah parameter nilai *k*. Nilai parameter *k* yang digunakan pada penelitian ini adalah dari rentang *k* = 1 hingga jumlah maksimal data *training*. Diharapkan dari pengujian ini dapat diketahui pengaruh penggunaan parameter jumlah data *training*, dan nilai *k* optimal yang menghasilkan tingkat akurasi paling baik yang dihitung berdasarkan nilai *recall*, *precision*, dan *F-measure*.

Pengujian kedua dilakukan untuk mengetahui pengaruh penggunaan metode pencarian jarak yang digunakan pada penelitian yaitu, *cosine similarity* dan *euclidean distance* terhadap nilai tingkat akurasi yang dihasilkan. Perbedaan penggunaan kedua metode tersebut sangat berpengaruh pada proses penghitungan nilai keanggotaan semua kelas untuk data *testing* x. Parameter pengujian kedua ini sama seperti halnya pada pengujian pertama. Diharapkan dari pengujian ini dapat diketahui metode manakah yang menghasilkan tingkat akurasi lebih baik yang dihitung berdasarkan nilai *recall*, *precision*, dan *F-measure*.

Karena penghitungan *recall*, *precision*, dan *F-measure* berkenaan dengan suatu kategori tertentu, maka dalam pengujian akan langsung dilihat sekaligus optimalitas penggunaan jumlah data *training*, nilai *k*, dan pada penggunaan

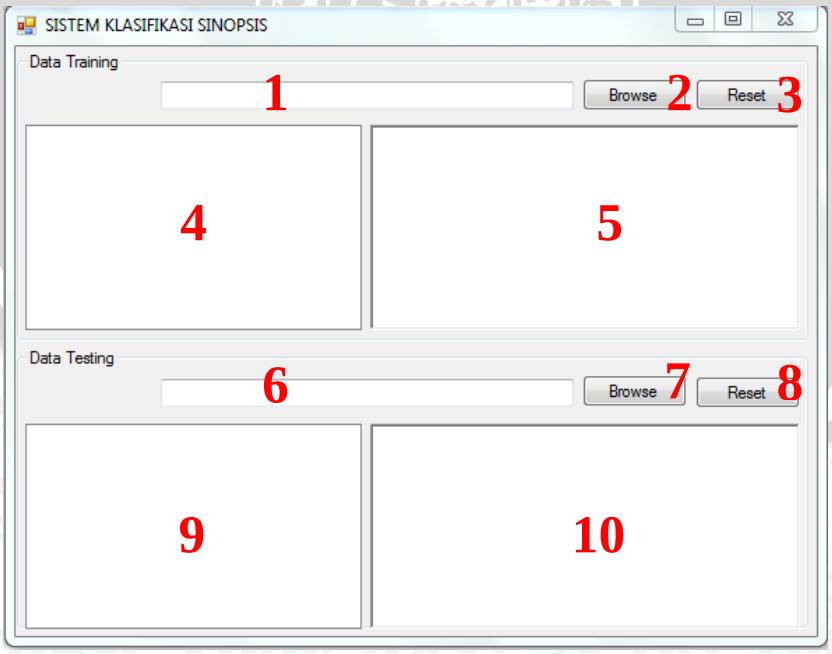
metode pencarian jarak. Rancangan hasil pengujian pengaruh variasi jumlah data *training*, nilai *k*, dan metode pencarian jarak ditunjukkan pada tabel 3.8

Tabel 3.8 Rancangan Hasil Pengujian

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	5						
	10						
	25						
Drama	5						
	10						
	25						

3.6 Rancangan Antar Muka Sistem

Rancangan antar muka sistem klasifikasi ini terdiri dari empat bagian utama, yaitu bagian *load data*, *training*, *classify*, dan *evaluation*. Rancangan antar muka bagian *load data* ditunjukkan pada gambar 3.11.

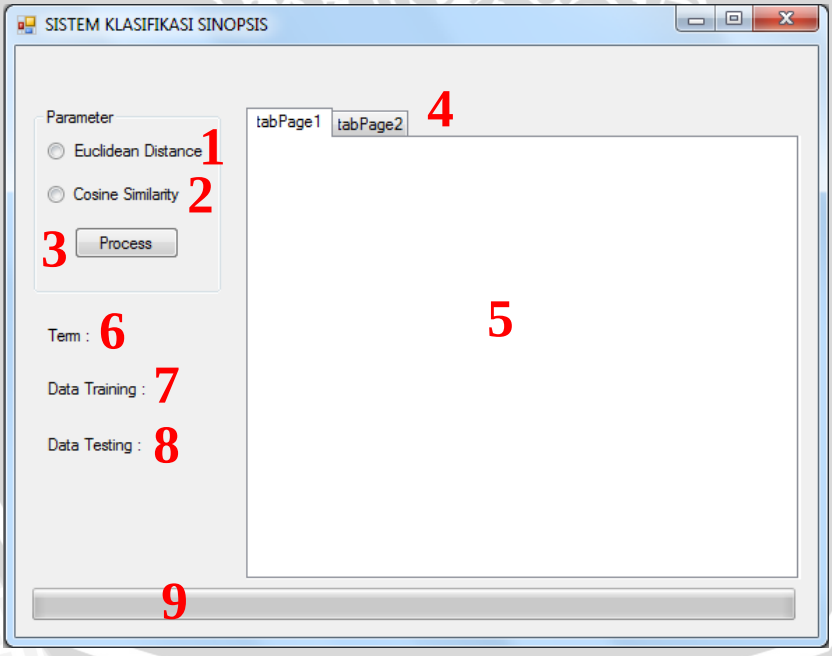


Gambar 3.11 Rancangan Antar Muka Bagian *Load Data*

Berikut keterangan gambar 3.11 :

1. *TextBox*, digunakan untuk menampilkan direktori folder data *training*.
2. *Button Browse*, digunakan untuk menginputkan lokasi folder data *training*.
3. *Button Reset*, digunakan untuk menghapus semua data *training*.
4. *ListBox*, digunakan untuk menampilkan semua data *training* satu per satu.
5. *RichTextField*, digunakan untuk menampilkan isi teks data *training*.
6. *TextBox*, digunakan untuk menampilkan direktori folder data *testing*.
7. *Button Browse*, digunakan untuk menginputkan lokasi folder data *testing*.
8. *Button Reset*, digunakan untuk menghapus semua data *testing*.
9. *ListBox*, digunakan untuk menampilkan semua data *testing* satu per satu.
10. *RichTextField*, digunakan untuk menampilkan isi teks data *testing*.

Rancangan antar muka bagian *training* ditunjukkan pada gambar 3.12.



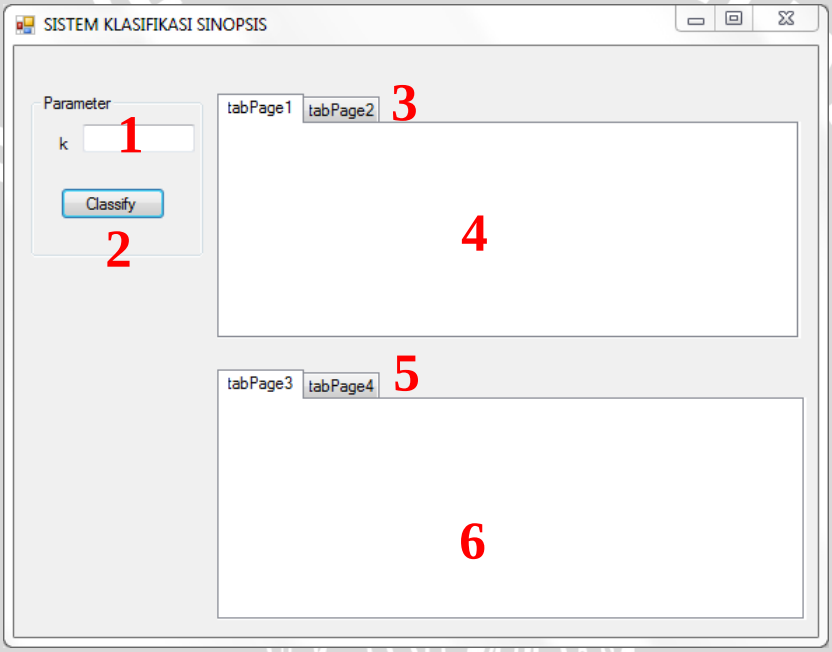
Gambar 3.12 Rancangan Antar Muka Bagian *Training*

Berikut keterangan gambar 3.12 :

1. *RadioButton*, digunakan untuk memilih parameter *cosine similarity*.
2. *RadioButton*, digunakan untuk memilih parameter *euclidean distance.*
3. *Button Process*, digunakan untuk memulai proses.

4. *TabControl*, digunakan untuk menampilkan *datagridview*.
5. *DataGridView*, digunakan untuk menampilkan hasil dari proses komputasi.
6. *Label*, digunakan untuk menampilkan jumlah *term* proses preprocessing.
7. *Label*, digunakan menampilkan jumlah data *training* yang diinputkan *user*.
8. *Label*, digunakan menampilkan jumlah data *testing* yang diinputkan *user*.
9. *Progress Bar*, digunakan untuk melihat progress sistem ketika melakukan komputasi.

Rancangan antar muka bagian *classify* ditunjukkan pada gambar 3.13.

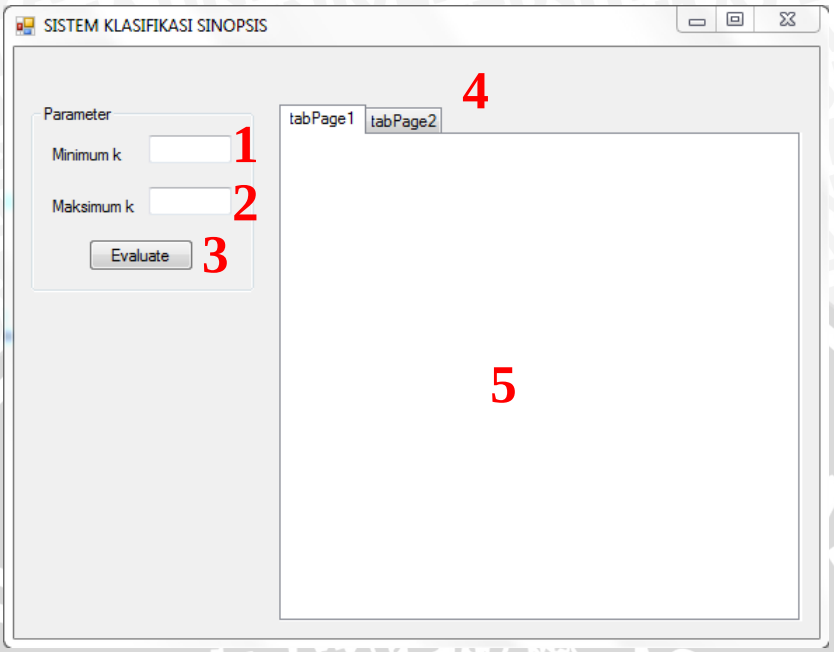


Gambar 3.13 Rancangan Antar Muka Bagian *Classify*

Berikut keterangan gambar 3.13 :

1. *Textbox*, digunakan untuk menginputkan nilai *k* sebagai paramater klasifikasi.
2. *Button Classify*, digunakan untuk memulai proses pengklasifikasian.
3. *TabControl*, digunakan untuk menampilkan *datagridview*.
4. *Datagridview*, digunakan untuk menampilkan hasil akhir klasifikasi.
5. *TabControl*, digunakan untuk menampilkan *datagridview*.
6. *Datagridview*, digunakan untuk menampilkan hasil evaluasi tingkat akurasi sistem.

Rancangan antar muka bagian *evaluation* ditunjukkan pada gambar 3.14.



Gambar 3.14 Rancangan Antar Muka Bagian *Evaluation*

Berikut keterangan gambar 3.14 :

1. *Textbox*, digunakan untuk menginputkan nilai *k* minimum sebagai paramater evaluasi.
2. *Textbox*, digunakan untuk menginputkan nilai *k* maksimum sebagai paramater evaluasi.
3. *Button Evaluate*, digunakan untuk memulai proses evaluasi tingkat akurasi sistem secara keseluruhan.
4. *TabControl*, digunakan untuk menampilkan *datagridview*.
5. *Datagridview*, digunakan untuk menampilkan hasil dari proses komputasi dan hasil akhir evaluasi tingkat akurasi sistem secara keseluruhan.

BAB IV
IMPLEMENTASI

Pada bab implementasi ini akan diberikan penjelasan mengenai implementasi algoritma yang digunakan pada proses pembuatan sistem klasifikasi *genre* film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN*.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang berkaitan dalam proses pembuatan sistem klasifikasi *genre* film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN* meliputi, lingkungan perangkat keras (*hardware*), dan lingkungan perangkat lunak (*software*).

4.1.1 Lingkungan Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) yang digunakan dalam penelitian ini untuk membuat sistem klasifikasi sesuai dengan algoritmanya antara lain :

1. *Intel Pentium Dual Core Processor* T2390 1,86 GHz.
2. *RAM Memory* DDR2 2 GB.
3. *Hardisk Memory* 160 GB.
4. Monitor 14”.
5. *Keyboard*.
6. *Mouse*.

4.1.2 Lingkungan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan dalam penelitian ini untuk membuat sistem klasifikasi sesuai dengan algoritmanya antara lain :

1. *Windows 7 Ultimate*, sebagai sistem operasi komputer.
2. *C#.net*, sebagai bahasa pemrograman yang digunakan.
3. *Microsoft Visual Studio 2010 Professional*, sebagai *software development* pembuatan sistem.
4. *Notepad*, sebagai *software* editor data penelitian yang disimpan pada *file* .txt
5. *Adobe Photosop CS5*, sebagai *software* untuk pembuatan *interface* sistem.

4.2 Implementasi Program

Secara garis besar terdapat 3 proses utama dalam implementasi algoritma sistem, yaitu *Preprocessing*, *Weighting*, dan *Fuzzy k-NN Classifier*. Masing-masing proses utama tersebut dipecah menjadi beberapa sub-proses. Pada sub-bab berikut, akan dijelaskan mengenai implementasi proses utama beserta sub-prosesnya.

4.2.1 Implementasi Proses *Preprocessing*

Semua data yang digunakan dalam penelitian ini, data *training* maupun data *testing* terlebih dulu melewati proses *preprocessing*. Proses *preprocessing* terdiri dari 3 sub-proses, yaitu *tokenizing*, *filtering*, dan *stemming*. Berikut adalah penjelasan mengenai implementasi sub-proses dari *preprocessing*.

4.2.1.1 Implementasi Sub-Proses *Tokenizing*

Tujuan utama sub-proses *tokenizing* adalah dokumen utuh diurai menjadi daftar kata (*token*). Pada sub-proses ini dilakukan proses pengubahan semua huruf menjadi huruf kecil. Spesial karakter seperti tanda baca dan angka akan digantikan dengan spasi. Kemudian dokumen akan dipisah (*splitting*) menjadi kata tunggal jika terdapat spasi, dan hasil dari daftar kata (*token*) disimpan dalam *List <string> token*. Implementasi sub-proses *tokenizing* ditunjukkan pada source code 4.1.

```
private List<string> Tokenizing(String teks_sinopsis)
{
    teks_sinopsis = teks_sinopsis.ToLower();
    char[] sp_karakter = new char[] { '~', '}', '{', '|', '*', ':',
    '\\', '_', '[', ']', '/', ';', '<', '>', ' ', '-', '=', '+', '.',
    ',', ',', '"', '!', '?', '@', '#', '$', '%', '^', '&', '(', ')',
    '\r', '\n', '\t', '\v', '1', '2', '3', '4', '5', '6', '7', '8',
    '9', '0', '\'' };
    List<string> token = teks_sinopsis.Split(sp_karakter,
    StringSplitOptions.RemoveEmptyEntries).ToList();
    return token;
}
```

Sourcecode 4.1 Fungsi *tokenizing()*

4.2.1.2 Implementasi Sub-Proses *Filtering*

Sub-proses berikutnya adalah *filtering*. Dalam sub-proses ini dilakukan proses penghapusan kata yang tidak merepresentasikan isi (*stopword*) pada data *traning* maupun data *testing*. Daftar *stopword* sudah disimpan pada file stopword.txt. Kemudian sistem akan melakukan pengecekan setiap *token* hasil sub-proses *tokenizing*, jika *token* yang dicek termasuk dalam kelompok *stopword* maka *token* itu akan dihapus. Sub-proses *filtering* terdapat pada fungsi stopword(). Implementasi sub-proses *filtering* ditunjukkan pada sourcecode 4.2.

```
private List<string> stopword(List<string> token) {
    string[] stopword = new string[1000];
    int m = stopword.GetLength(0);
    int n = token.Count();
    string[] _ceksplit = new string[n];
    bool sama = true;
    int jmlh = 0;
    FileStream sr = new FileStream("stopwords.txt", FileMode.Open);
    StreamReader str = new StreamReader(sr);
    int a = 0;
    while (!str.EndOfStream) {
        stopword[a] = str.ReadLine();
        a++;
    }
    sr.Close();
    str.Close();
    for (int j = 0; j < n; j++) {
        sama = true;
        for (int k = 0; k < m; k++)
        {
            if (token[j] == stopword[k])
            sama = false;
        }
        if (sama != false)
        {
             _ceksplit[j] = token[j];
             jmlh++;
        }
    }
    List<string> words = new List<string>();
    for (int l = 0; l < n; l++) {
        if (_ceksplit[l] != null) {
            words.Add(_ceksplit[l]);
        }
    }
    return words;
}
```

Sourcecode 4.2 Fungsi *stopword()*

4.2.1.3 Implementasi Sub-Proses *Stemming*

Sub-proses berikutnya adalah *stemming*. *Stemming* dilakukan dengan tujuan membentuk kata menjadi bentuk dasarnya. Semua imbuhan kata (*term*) akan dihapus pada sub-proses ini. Sub-proses *stemming* dilakukan oleh class `Stemmer.cs` yang didalamnya berisi *library method* proses *stemming* sesuai yang dikembangkan Martin F. Potter. *Library method stemming* diperoleh dari www.tartarus/~martin/PorterStemmer. Class `Stemmer.cs` dipanggil pada implementasi sourcecode 4.3.

```
private List<string> stemming(List<string> term)
{
    List<string> stem_term = new List<string>();
    Stemmer stem = new Stemmer();
    for (int i = 0; i < term.Count; i++)
    {
        stem_term.Add(stem.Porter.stemTerm(term[i]));
    }
    return stem_term;
}
```

Sourcecode 4.3 Method Pemanggilan *Class stemmer*

4.2.2 Implementasi Proses *Weighting*

Proses *weighting* adalah proses utama yang ke-2 setelah proses *preprocessing*. Proses ini bertujuan untuk menghitung bobot *term* hasil proses *preprocessing* dengan menggunakan *TF-IDF*. Terlebih dahulu dihitung nilai *DF*, kemudian nilai *DF* digunakan untuk menghitung nilai *IDF* yang nantinya digunakan untuk menghitung bobot masing-masing *term*. Bobot masing-masing *term* dihitung dengan mengalikan nilai *TF* dari masing-masing *term* dengan *IDF*-nya, dan disimpan pada array `dataBobotTR` untuk bobot *term* data *training*. Implementasi sub-proses hitung *IDF* dan bobot *term* ditunjukkan pada sourcecode 4.4.

```
int df = 0;
double[,] dataBobotTR = new double[dataTR.Length, term.Count];
for (int i = 0; i < term.Count; i++) {
    for (int j = 0; j < dataTR.Length; j++) {
        if (dataTraining[j, i] != 0) {
            df++;
        }
```

```
    }
    dataTraining[dataTR.Length, i] = df;
    df = 0;
    dataTraining[(dataTR.Length + 1), i] = Math.Round
        (Math.Log10(dataTR.Length / dataTraining[dataTR.Length, i]),
        4, MidpointRounding.AwayFromZero);
    idf.Add(dataTraining[(dataTR.Length + 1), i]);
    DataGrid_Vector.Rows.Add();
    DataGrid_Vector.Rows[i].Cells[0].Value = term[i];
    for (int k = 0; k < dataTR.Length; k++)
    {
        dataBobotTR[k, i] =
            dataTraining[k, i] * dataTraining[(dataTR.Length + 1), i];
        if (double.IsNaN(dataBobotTR[k, i]) ||
            double.IsInfinity(dataBobotTR[k, i]))
        {
            dataBobotTR[k, i] = 0;
        }
        DataGrid_Vector.Rows[i].Cells[k + 1].Value = dataBobotTR[k, i];
    }
}
```

Sourcecode 4.4 Proses Penghitungan *IDF* Dan Bobot *Term*

4.2.3 Implementasi Proses *Fuzzy k-NN Classifier*

Proses utama yang terakhir adalah proses *fuzzy k-NN classifier*. Proses ini digunakan untuk melakukan penghitungan *membership* pada keseluruhan kelas yang nantinya digunakan untuk pengklasifikasian *data testing*. Proses *fuzzy k-NN classifier* dibagi menjadi 2 sub-proses lagi yaitu, *cosine similarity* atau *euclidean distance*, dan *fuzzy k-NN*. Berikut adalah penjelasan mengenai implementasi sub-proses dari *fuzzy k-NN classifier*.

4.2.3.1 Implementasi Sub-Proses *Cosine Similarity*

Sub-proses *cosine similarity* digunakan untuk menghitung kemiripan antara data *testing* dengan keseluruhan data *training* berdasarkan bobot *term* yang sudah dihitung pada proses sebelumnya. Nilai bobot *term* data *testing* dan data *training* yang ada pada indeks yang sama akan dikalikan, kemudian dijumlahkan dan hasilnya ditampung pada variabel `double XxD`. Nilai bobot *term* data *testing* dan data *training* akan dikuadratkan dan dijumlah kesemuanya, kemudian dilakukan proses pengakaran kuadrat, hasilnya ditampung pada `double X` untuk proses penghitungan bobot *term* untuk data *testing*, dan `double D` untuk proses penghitungan bobot *term* untuk data *training*. Untuk mendapatkan nilai *cosine*

similarity, variabel `double XxD` akan dibagi dengan variabel `double X` yang telah dikalikan variabel `double D`, hasil *cosine similarity* akan disimpan pada `array similarity`. Implementasi sub-proses *cosine similarity* ditunjukkan pada sourcecode 4.5.

```
similarity = new double[dataTS.Length, dataTR.Length];
double XxD = 0;
double X = 0;
double D = 0;

for (int i = 0; i < dataTS.Length; i++)
{
    string biasa = "";
    String judul = dataTS[i];
    int id_slash = judul.LastIndexOf("\\");
    judul = judul.Remove(0, id_slash + 1);
    biasa = biasa + judul;
    DataGrid_Distance.Rows.Add();
    DataGrid_Distance.Rows[i].Cells[0].Value = biasa;

    for (int j = 0; j < dataTR.Length; j++)
    {
        for (int n = 0; n < nTerm; n++)
        {
            XxD+= (dataBobotTS[i, n] * dataBobotTR[j, n]);
            X+= (dataBobotTS[i, n] * dataBobotTS[i, n]);
            D+= (dataBobotTR[j, n] * dataBobotTR[j, n]);
        }
        X = Math.Sqrt(X);
        D = Math.Sqrt(D);
        similarity[i,j] = Math.Round(XxD /
(X*D),4,MidpointRounding.AwayFromZero);
        DataGrid_Distance.Rows[i].Cells[j + 1].Value = similarity[i, j];
        X = 0;
        D = 0;
        XxD = 0;
    }
}
```

Sourcecode 4.5 Proses Penghitungan Jarak Menggunakan *Cosine Similarity*

4.2.3.2 Implementasi Sub-Proses *Euclidean Distance*

Sub-proses *euclidean distance* digunakan untuk menghitung jarak antara data *testing* dengan keseluruhan data *training* berdasarkan bobot *term* yang sudah dihitung pada proses sebelumnya. Nilai bobot *term* data *testing* akan dikurangi dengan nilai bobot *term* data *training* pada indeks yang sama, hasilnya akan disimpan pada `double selisih_bobot`. Kemudian hasil pengurangan tadi akan dikuadratkan, hasilnya akan disimpan pada `double kuadrat`. Selanjutnya hasil

pengurangan yang sudah dikuadratkan akan dijumlah, hasil dari penjumlahan akan diakar kuadrat untuk mendapatkan nilai jarak. Nilai jarak disimpan pada array `distance`. Implementasi sub-proses *euclidean distance* ditunjukkan pada sourcecode 4.6.

```
distance = new double[dataTS.Length, dataTR.Length];
double total_kuadrat = 0;
for (int i = 0; i < dataTS.Length; i++)
{
    string biasa = "";
    String judul = dataTS[i];
    int id_slash = judul.LastIndexOf("\\");
    judul = judul.Remove(0, id_slash + 1);
    biasa = biasa + judul;
    DataGrid_Distance.Rows.Add();
    DataGrid_Distance.Rows[i].Cells[0].Value = biasa;

    for (int j = 0; j < dataTR.Length; j++)
    {
        for (int n = 0; n < nTerm; n++)
        {
            double selisih_bobot = (dataBobotTS[i, n] - dataBobotTR[j, n]);
            double kuadrat = Math.Round
                (Math.Pow(selisih_bobot, 2.0),
                4, MidpointRounding.AwayFromZero);
            total_kuadrat = total_kuadrat + kuadrat;
        }
        distance[i, j] = Math.Round
            (Math.Sqrt(total_kuadrat), 4, MidpointRounding.AwayFromZero);
        DataGrid_Distance.Rows[i].Cells[j + 1].Value = distance[i,j];
        total_kuadrat = 0;
    }
}
```

Sourcecode 4.6 Proses Penghitungan Jarak Menggunakan *Euclidean Distance*

4.2.3.3 Implementasi Sub-Proses *Fuzzy k-NN*

Sub-proses *fuzzy k-NN* digunakan sebagai perhitungan akhir untuk menentukan hasil klasifikasi untuk data *testing*. Namun sebelumnya akan diambil terlebih dahulu sejumlah *k-Nearest Neighbour* yang memiliki nilai *cosine similarity* paling tinggi (jika metode *cosine similarity* yang dipilih oleh *user*) atau nilai *euclidean distance* yang terdekat (jika metode *euclidean distance* yang dipilih oleh *user*). Penentuan sejumlah *k-Nearest Neighbour* dilakukan dengan mengurutkan nilainya terlebih dahulu menggunakan *method* `OrderByDescending`, kemudian untuk mengambil sejumlah *k-Nearest Neighbour* menggunakan *method*

Take. Implementasi proses penentuan *k-Nearest Neighbour* ditunjukkan pada sourcecode 4.7.

```
List<Dictionary<string, double>> list_dictionary =
new List<Dictionary<string, double>>();

if (distance != null)
{
    for (int i = 0; i < dataTS.Length; i++)
    {
        Dictionary<string, double> dictionary =
            new Dictionary<string, double>();
        for (int j = 0; j < dataTR.Length; j++)
        {
             //Proses penghitungan nilai weight voting
             weight_voting[i, j] = Math.Round
                (1 / Math.Pow(Math.Abs(distance[i, j]), (2 / (m_value - 1))),
                 4, MidpointRounding.AwayFromZero);
             dictionary.Add(dataTR[j], weight_voting[i, j]);
        }
        //Proses Pengambilan Sejumlah k-Nearest Neighbour
        k_value = int.Parse(TB_inputK.Text);
        Dictionary<string, double> filter = dictionary.
            OrderByDescending(x => x.Value)
            .Take(k_value).ToDictionary(x => x.Key, x => x.Value);
        list_dictionary.Add(filter);
    }
}
else
{
    for (int i = 0; i < dataTS.Length; i++)
    {
        Dictionary<string, double> dictionary =
            new Dictionary<string, double>();
       for (int j = 0; j < dataTR.Length; j++)
       {
            dictionary.Add(dataTR[j], similarity[i, j]);
       }
       //Proses Pengambilan Sejumlah k-Nearest Neighbour
       k_value = int.Parse(TB_inputK.Text);
       Dictionary<string, double> filter = dictionary.
           OrderByDescending(t => t.Value)
           .Take(k_value).ToDictionary(t => t.Key, t => t.Value);
       list_dictionary.Add(filter);
    }
}
```

Sourcecode 4.7 Proses Penentuan *k-Nearest Neighbour*

Setelah didapatkan data sejumlah *k-Nearest Neighbour*, proses selanjutnya adalah menentukan klasifikasi dengan memberikan nilai *membership fuzzy* pada data *testing* yaitu *membership* semua *genre* yang digunakan pada penelitian ini berdasarkan data sejumlah *k-Nearest Neighbour*. Proses klasifikasi yaitu dengan

memilih nilai *membership* yang paling tinggi. Implementasi proses penentuan klasifikasi data *testing* ditunjukkan pada sourcecode 4.8.

```
//Proses penghitungan nilai keanggotaan kelas ke-i pada tetangga ke-j
Dictionary<string, double> miu_IJsama = new Dictionary<string, double>();
Dictionary<string, double> miu_IJbeda = new Dictionary<string, double>();
K = double.Parse(TB_inputK.Text);
foreach (var g in GenreFrek) {
    is_member = 0.51 + (g.Value / K) * 0.49;
    non_member = (g.Value / K) * 0.49;
    miu_IJsama.Add(g.Key, is_member);
    miu_IJbeda.Add(g.Key, non_member);
}
IS_member.Add(miu_IJsama);
NON_member.Add(miu_IJbeda);

Dictionary<string, double> membership_ix = new Dictionary<string, double>();
int index_g = 0;
var miu = miu_IJsama.Keys.ToList();
var listMiu_sama = miu_IJsama.Values.ToList();
var listMiu_beda = miu_IJbeda.Values.ToList();
for (int j = 0; j < list_genre.Length; j++) {
    double total_count_atas = 0;
    double total_count_bawah = 0;

    for (int k = 0; k < Value_grupkNN.Count; k++) {
        if (miu.Contains(list_genre[j]))
        {
            index_g = miu.IndexOf(list_genre[j]);
            List<string> l_genre = himp_genre[k];
            if (l_genre.Contains(miu[index_g])) {
                total_count_atas += Value_grupkNN[k] * listMiu_sama[index_g];
                total_count_bawah += Value_grupkNN[k];
            }
            else {
                total_count_atas += Value_grupkNN[k] * listMiu_beda[index_g];
                total_count_bawah += Value_grupkNN[k];
            }
        }
        else {
            total_count_atas += Value_grupkNN[k] * 0.0;
            total_count_bawah += Value_grupkNN[k];
        }
    }
    double miu_ix = Math.Round (total_count_atas / total_count_bawah,
        4, MidpointRounding.AwayFromZero);
    membership_ix.Add(list_genre[j], miu_ix);
}
list_miu_datagrid.Add(membership_ix);

Dictionary<string, double> filter_membership_ix =
        membership_ix.OrderByDescending(p => p.Value)
        .Take(1).ToDictionary(p => p.Key, p => p.Value);
list_membership.Add(filter_membership_ix);
```

Sourcecode 4.8 Proses Penentuan Klasifikasi Data *Testing*

4.3 Implementasi Antar Muka Sistem

Antar muka sistem klasifikasi sinopsis ini diimplementasi menggunakan *form* standar beserta *toolbox* yang disediakan oleh *Microsoft Visual Studio* 2010 dengan menggunakan bahasa pemrograman *C#*. Antar muka sistem ini terdiri atas 1 *main form* yang memiliki 4 *user control* utama, yaitu :

1. *User Control Load Data*
2. *User Control Training*
3. *User Control Classify*
4. *User Control Evaluation*

Dimana masing-masing dari *user control* utama memiliki fungsi yang berbeda. Pada sub-bab berikut, akan dijelaskan mengenai implementasi setiap *user control* utama sistem.

4.3.1 *User Control Load Data*

User control load data digunakan untuk melakukan penginputan data yang berkaitan dengan penelitian ini, yaitu data *training* maupun data *testing* yang berisi teks judul dan sinopsis dari suatu film. *User* menginputkan data *traning* maupun *testing* melalui tombol *browse* yang sudah disediakan. *User* juga dapat menghapus semua data *training* maupun *testing* melalui tombol *reset* yang sudah disediakan. Tampilan *user control load data* ditunjukkan pada gambar 4.1.

Gambar 4.1 *User Control Load Data*

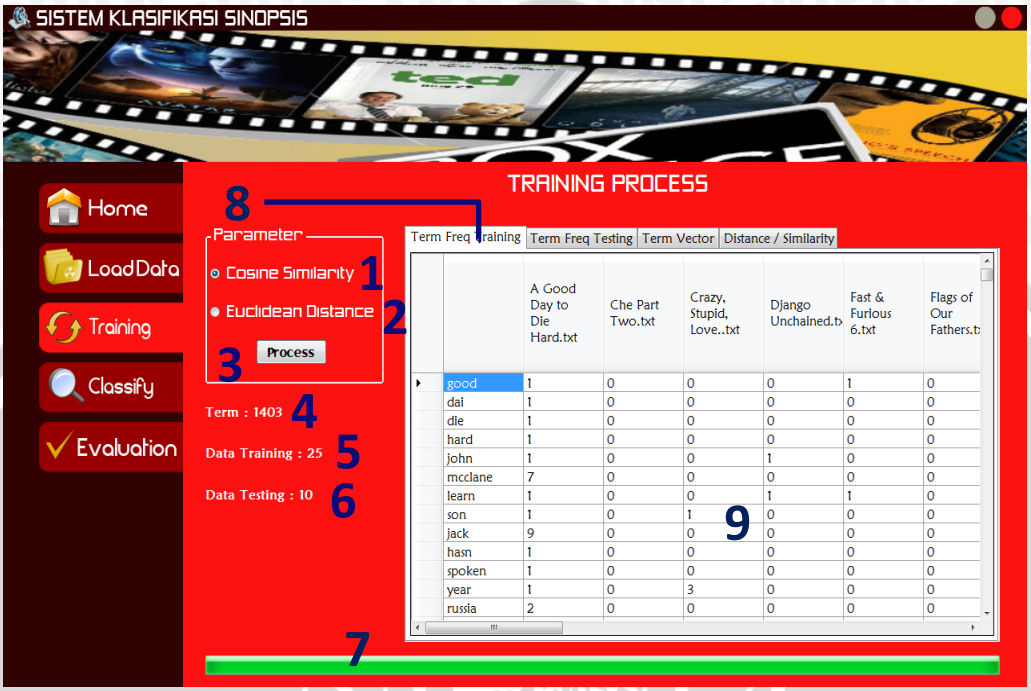
Berikut keterangan gambar 4.1 :

1. *TextBox*, digunakan untuk menampilkan direktori folder data *training*.
2. *Button Browse*, digunakan untuk menginputkan lokasi folder data *training*.
3. *Button Reset*, digunakan untuk menghapus semua data *training*.
4. *ListBox*, digunakan untuk menampilkan semua data *training* satu per satu.
5. *RichTextField*, digunakan untuk menampilkan isi teks data *training*.
6. *TextBox*, digunakan untuk menampilkan direktori folder data *testing*.
7. *Button Browse*, digunakan untuk menginputkan lokasi folder data *testing*.
8. *Button Reset*, digunakan untuk menghapus semua data *testing*.
9. *ListBox*, digunakan untuk menampilkan semua data *testing* satu per satu.
10. *RichTextField*, digunakan untuk menampilkan isi teks data *testing*.

4.3.2 *User Control Training*

Form training digunakan untuk memulai proses *preprocessing*, proses *weighting*, dan sub-proses *cosine similarity* atau *euclidean distance*. Untuk memulai proses, *user* terlebih dahulu memilih paramater metode pencarian jarak yang telah disediakan pada *RadioButton*, dan setelah itu cukup menekan tombol

process. Hasil dari proses akan ditampilkan pada *datagridview* yang terdapat pada 4 *tabpage*. Tampilan *user control training* beserta masing-masing *tabpage* ditunjukkan pada gambar 4.2, 4.3, 4.4, dan 4.5.

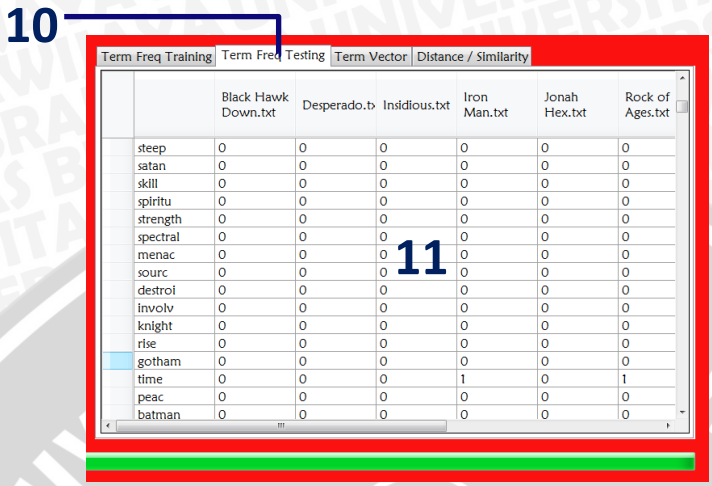


Gambar 4.2 *User Control Training*, dan *Tabpage Term Freq Training*

Berikut keterangan gambar 4.2 :

1. *RadioButton*, digunakan untuk memilih parameter *cosine similarity*.
2. *RadioButton*, digunakan untuk memilih parameter *euclidean distance*.
3. *Button Process*, digunakan untuk memulai proses.
4. *Label*, digunakan untuk menampilkan jumlah *term* proses preprocessing.
5. *Label*, digunakan untuk menampilkan jumlah data *training* yang diinputkan *user*.
6. *Label*, digunakan untuk menampilkan jumlah data *testing* yang diinputkan *user*.
7. *Progress Bar*, digunakan untuk melihat progress sistem ketika melakukan komputasi.
8. *TabPage TermFreq Training*, digunakan untuk menampilkan *datagridview*.

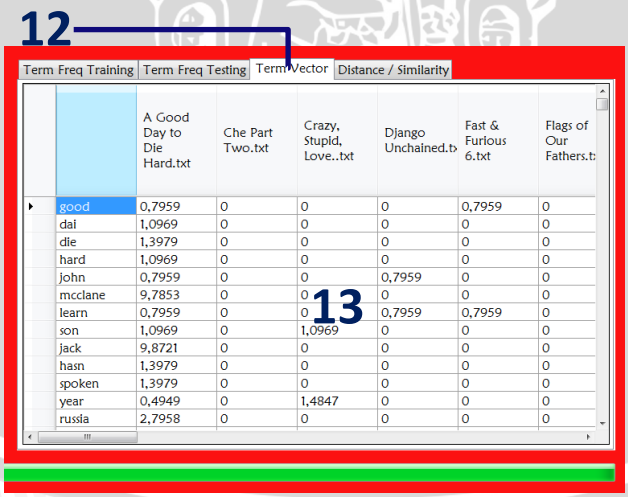
9. *DataGridView*, digunakan untuk menampilkan frekuensi *term* pada keseluruhan data *training*.



Gambar 4.3 *User Control Training*, dan *Tabpage Term Freq Testing*

Berikut keterangan gambar 4.3 :

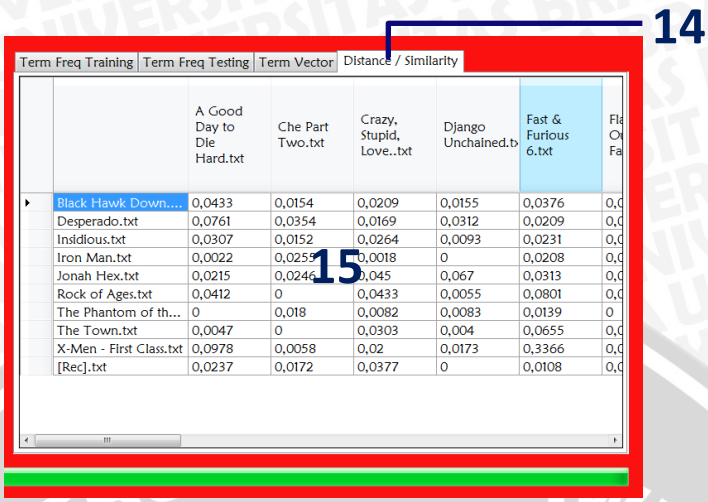
10. *TabPage TermFreq Testing*, digunakan untuk menampilkan *datagridview*.
11. *DataGridView*, digunakan untuk menampilkan frekuensi *term* pada keseluruhan data *testing*.



Gambar 4.4 *User Control Training*, dan *Tabpage Term Vector*

Berikut keterangan gambar 4.4 :

12. *TabPage Term Vector*, digunakan untuk menampilkan *datagridview*.
13. *DataGridView*, digunakan untuk menampilkan hasil perhitungan bobot *term*.



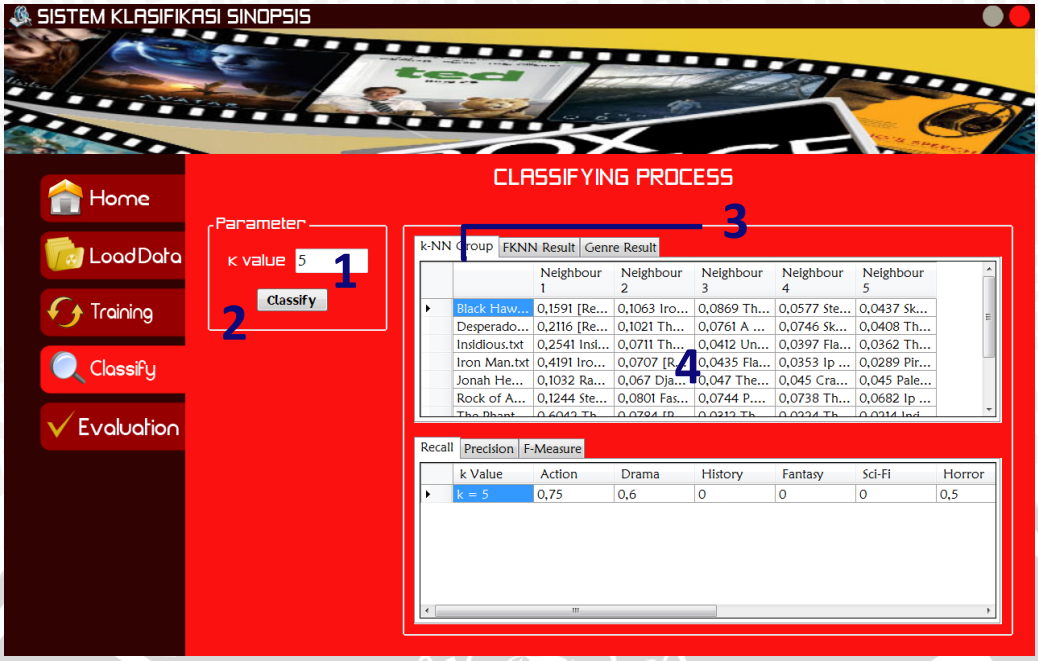
Gambar 4.5 *User Control Training*, dan *Tabpage Distance / Similarity*

Berikut keterangan gambar 4.5 :

14. *TabPage Distance / Similarity*, digunakan untuk menampilkan *datagridview*.
15. *DataGridView*, digunakan untuk menampilkan hasil perhitungan jarak atau kemiripan antara vektor data *testing* dan data *training* sesuai parameter metode pencarian

4.3.3 *User Control Classify*

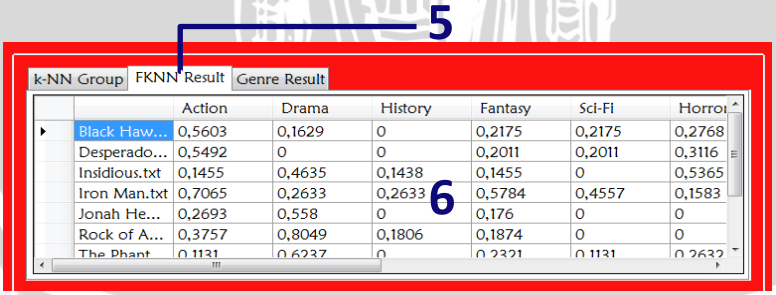
User control classify digunakan untuk memulai proses klasifikasi data *testing*, melihat hasil klasifikasi, serta evaluasi tingkat akurasi sistem satu per satu sesuai parameter nilai *k* yang diinginkan *user*. *User* diharuskan terlebih dahulu menentukan parameter pengujian klasifikasi yaitu memasukkan nilai *k* pada *textbox*. Untuk memulai proses, *user* cukup menekan tombol *classify*. Hasil dari proses akan ditampilkan pada *datagridview* yang terdapat pada 6 *tabpage*. Tampilan *user control classify* beserta masing-masing *tabpage* ditunjukkan pada gambar 4.6, 4.7, 4.8, 4.9, 4.10, dan 4.11.



Gambar 4.6 *User Control Classify*, dan *Tabpage k-NN Group*

Berikut keterangan gambar 4.6 :

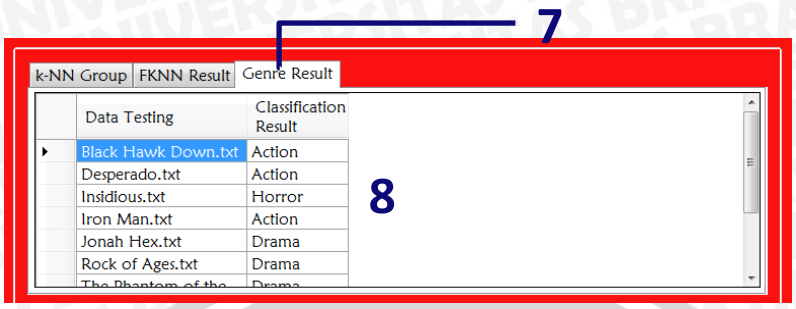
1. *Textbox*, digunakan untuk menginputkan nilai *k* sebagai paramater klasifikasi.
2. *Button Classify*, digunakan untuk memulai proses pengklasifikasian.
3. *TabPage k-NN Group*, digunakan untuk menampilkan *datagridview*.
4. *Datagridview*, digunakan untuk menampilkan hasil penentuan himpunan *k-Nearest Neighbour*.



Gambar 4.7 *User Control Classify*, dan *Tabpage FKNN Result*

Berikut keterangan gambar 4.7 :

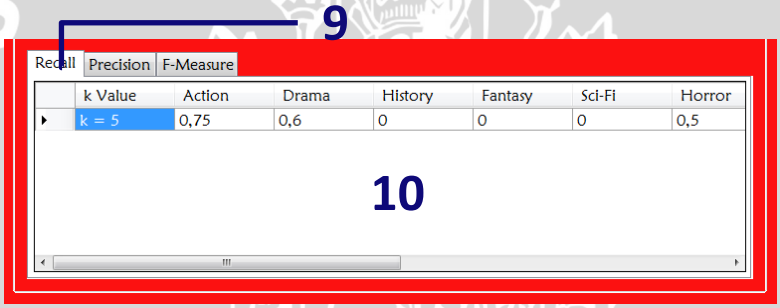
5. *TabPage FKNN Result*, digunakan untuk menampilkan *datagridview*.
6. *DataGridView*, digunakan untuk menampilkan hasil perhitungan akhir *membership fuzzy* pada data *testing*.



Gambar 4.8 *User Control Classify*, dan *Tabpage Genre Result*

Berikut keterangan gambar 4.8 :

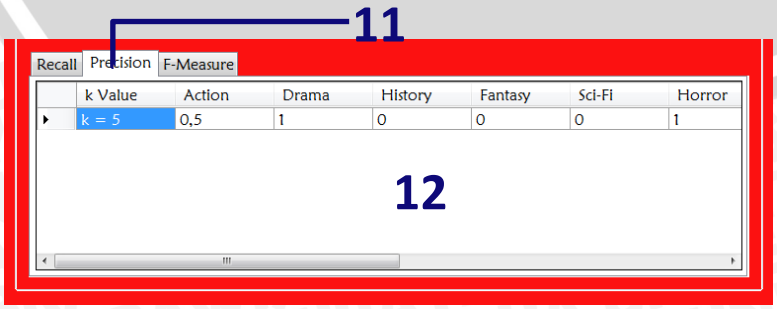
7. *TabPage Genre Result*, digunakan untuk menampilkan *datagridview*.
8. *DataGridView*, digunakan untuk menampilkan hasil klasifikasi data testing berdasarkan nilai *membership* tertinggi.



Gambar 4.9 *User Control Classify*, dan *Tabpage Genre Recall*

Berikut keterangan gambar 4.9 :

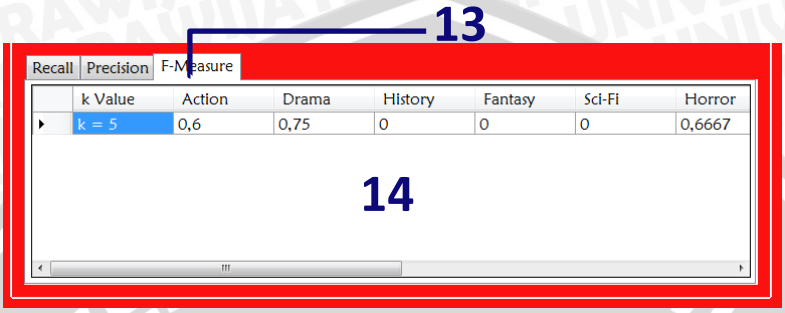
9. *TabPage Recall*, digunakan untuk menampilkan *datagridview*.
10. *DataGridView*, digunakan untuk menampilkan hasil perhitungan *recall* yang menjadi salah satu ukuran evaluasi sistem.



Gambar 4.10 *User Control Classify*, dan *Tabpage Precision*

Berikut keterangan gambar 4.10 :

11. *TabPage Precision*, digunakan untuk menampilkan *datagridview*.
12. *DataGridView*, digunakan untuk menampilkan hasil perhitungan *precision* yang menjadi salah satu ukuran evaluasi sistem.



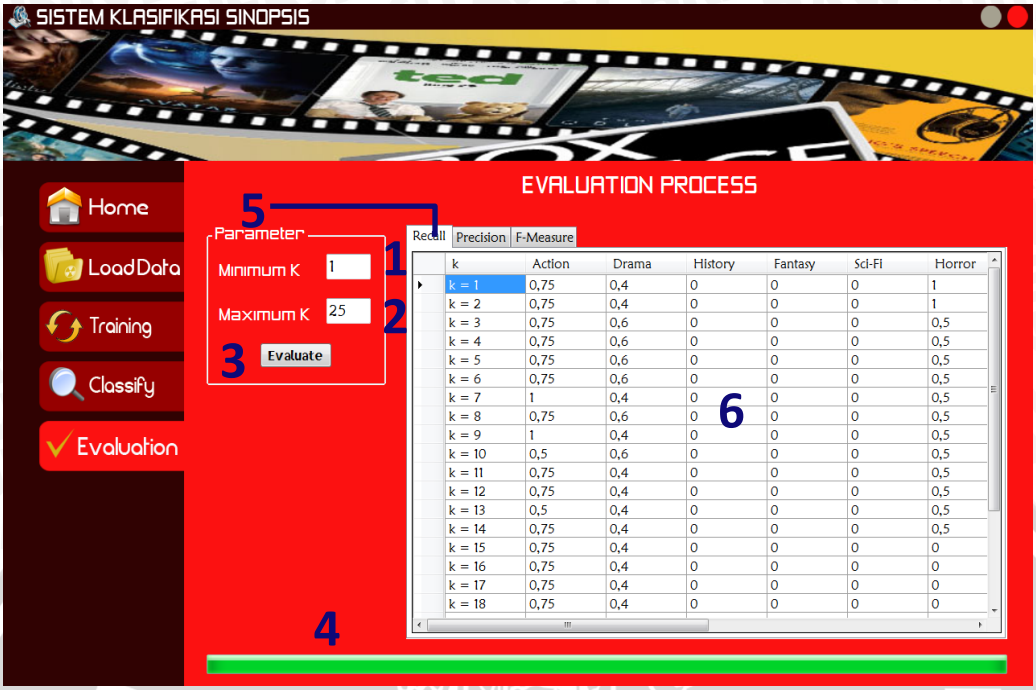
Gambar 4.11 *User Control Classify*, dan *Tabpage F-Measure*

Berikut keterangan gambar 4.11 :

13. *TabPage F-Measure*, digunakan untuk menampilkan *datagridview*.
14. *DataGridView*, digunakan untuk menampilkan hasil perhitungan *F-measure* yang menjadi salah satu ukuran evaluasi sistem.

4.3.4 *User Control Evaluation*

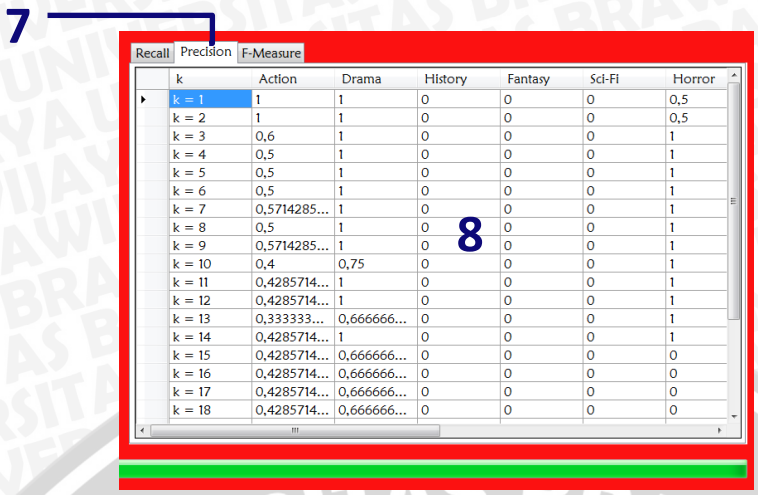
User control evaluation digunakan untuk melihat dan mengevaluasi tingkat akurasi sistem secara keseluruhan pada rentang titik nilai *k* yang ditentukan *user*. *User* diharuskan terlebih dahulu menentukan parameter evaluasi yaitu memasukkan nilai minimum dan maksimum *k* pada *textbox*. Untuk memulai proses, *user* cukup menekan tombol *evaluate*. Hasil dari proses akan ditampilkan pada *datagridview* yang terdapat pada 3 *tabpage*. Tampilan *user control evaluation* beserta masing-masing *tabpage* ditunjukkan pada gambar 4.12, 4.13, dan 4.14.



Gambar 4.12 *User Control Evaluation*, dan *Tabpage Recall*

Berikut keterangan gambar 4.12 :

1. *Textbox*, digunakan untuk menginputkan nilai minimum *k* sebagai paramater evaluasi sistem.
2. *Textbox*, digunakan untuk menginputkan nilai maksimum *k* sebagai paramater evaluasi sistem.
3. *Button Evaluate*, digunakan untuk memulai proses evaluasi sistem.
4. *Progress Bar*, digunakan untuk melihat progress sistem ketika melakukan komputasi.
5. *TabPage Recall*, digunakan untuk menampilkan *datagridview*.
6. *Datagridview*, digunakan untuk menampilkan hasil perhitungan *recall* secara keseluruhan pada rentang titik nilai *k* yang ditentukan *user*.

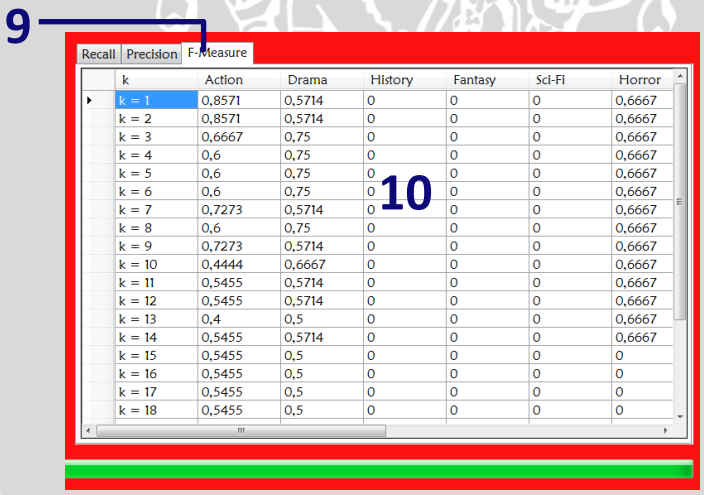


k	Action	Drama	History	Fantasy	Sci-Fi	Horror
k = 1	1	1	0	0	0	0,5
k = 2	1	1	0	0	0	0,5
k = 3	0,6	1	0	0	0	1
k = 4	0,5	1	0	0	0	1
k = 5	0,5	1	0	0	0	1
k = 6	0,5	1	0	0	0	1
k = 7	0,5714285...	1	0	0	0	1
k = 8	0,5	1	0	0	0	1
k = 9	0,5714285...	1	0	0	0	1
k = 10	0,4	0,75	0	0	0	1
k = 11	0,4285714...	1	0	0	0	1
k = 12	0,4285714...	1	0	0	0	1
k = 13	0,333333...	0,666666...	0	0	0	1
k = 14	0,4285714...	1	0	0	0	1
k = 15	0,4285714...	0,666666...	0	0	0	0
k = 16	0,4285714...	0,666666...	0	0	0	0
k = 17	0,4285714...	0,666666...	0	0	0	0
k = 18	0,4285714...	0,666666...	0	0	0	0

Gambar 4.13 *User Control Evaluation,* dan *Tabpage Precision*

Berikut keterangan gambar 4.13 :

7. *TabPage Precision*, digunakan untuk menampilkan *datagridview*.
8. *Datagridview*, digunakan untuk menampilkan hasil perhitungan *precision* secara keseluruhan pada rentang titik nilai *k* yang ditentukan *user*.



k	Action	Drama	History	Fantasy	Sci-Fi	Horror
k = 1	0,8571	0,5714	0	0	0	0,6667
k = 2	0,8571	0,5714	0	0	0	0,6667
k = 3	0,6667	0,75	0	0	0	0,6667
k = 4	0,6	0,75	0	0	0	0,6667
k = 5	0,6	0,75	0	0	0	0,6667
k = 6	0,6	0,75	0	0	0	0,6667
k = 7	0,7273	0,5714	0	0	0	0,6667
k = 8	0,6	0,75	0	0	0	0,6667
k = 9	0,7273	0,5714	0	0	0	0,6667
k = 10	0,4444	0,6667	0	0	0	0,6667
k = 11	0,5455	0,5714	0	0	0	0,6667
k = 12	0,5455	0,5714	0	0	0	0,6667
k = 13	0,4	0,5	0	0	0	0,6667
k = 14	0,5455	0,5714	0	0	0	0,6667
k = 15	0,5455	0,5	0	0	0	0
k = 16	0,5455	0,5	0	0	0	0
k = 17	0,5455	0,5	0	0	0	0
k = 18	0,5455	0,5	0	0	0	0

Gambar 4.14 *User Control Evaluation,* dan *Tabpage F-Measure*

Berikut keterangan gambar 4.14 :

9. *TabPage F-Measure*, digunakan untuk menampilkan *datagridview*.
10. *Datagridview*, digunakan untuk menampilkan hasil perhitungan *F-measure* secara keseluruhan pada rentang titik nilai *k* yang ditentukan *user*.

BAB V
HASIL DAN PEMBAHASAN

Pada bab hasil dan pembahasan ini akan dijelaskan hasil dari uji coba penelitian tentang klasifikasi *genre* film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN*. Selain itu juga akan dijelaskan tentang pembahasan yang berkaitan dengan hasil uji coba penelitian.

5.1 Implementasi Uji Coba

Implementasi uji coba pada sistem klasifikasi *genre* film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN*, mengacu pada perancangan uji coba pada subbab 3.5. Pengujian dilakukan untuk mengetahui keakuratan sistem dalam melakukan klasifikasi terhadap data *testing* pada sejumlah kelas kategori yang digunakan pada penelitian ini, yaitu *genre* film.

Masing-masing data *testing* telah disimpan pada sebuah file berekstensi .txt. File data *testing* berisi teks judul beserta sinopsis dari suatu film. File data *testing* yang digunakan sebagai bahan penelitian berjumlah 40 data yang memiliki sebaran kelas *genre* acak yang bervariasi jumlahnya.

Sedangkan kategori *genre* film yang digunakan pada penelitian ini terdiri dari 12 jenis *genre* yang berbeda. Daftar *genre* yang digunakan pada penelitian ini antara lain : *action, drama, history, fantasy, sci-fi, horror, comedy, crime, musical, adventure, war, western*.

Keakuratan sistem dalam melakukan klasifikasi kemudian dinilai melalui parameter nilai *recall, precision,* dan *F-measure*. Penjelasan mengenai parameter tersebut dijelaskan pada subbab 2.14.

5.2 Hasil Implementasi Pengujian

Berikut adalah hasil pengujian yang telah dilakukan sesuai rancangan yang telah dijelaskan pada subbab 3.5. Karena penghitungan *recall, precision*, dan *F-measure* berkenaan dengan suatu kategori tertentu, maka dalam pengujian akan langsung dilihat sekaligus optimalitas penggunaan jumlah data *training*, nilai *k* (jumlah tetangga terdekat) dan penggunaan metode pencarian jarak untuk setiap

kategori. Hasil pengujian pengaruh variasi jumlah data *training*, nilai *k*, dan metode pencarian jarak ditunjukkan pada tabel 5.1 hingga tabel 5.6

Tabel 5.1 Hasil Pengujian Pengaruh Jumlah Data *Training* Sebanyak 75 Data, Nilai *k*, dan Metode Pencarian Jarak

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	2	0,8000	0,9231	0,8571	0,3333	1,0000	0,5000
	10	0,8000	0,8000	0,8000	0,0000	0,0000	0,0000
	25	0,7333	0,6111	0,6667	0,0000	0,0000	0,0000
	50	0,8667	0,6191	0,7222	0,2667	0,6667	0,381
	75	0,8667	0,5909	0,7027	0,1333	1,0000	0,2353
Drama	2	0,6667	0,8333	0,7047	0,1333	1,0000	0,2353
	10	0,8000	0,9231	0,8571	0,4667	0,875	0,6087
	25	0,6667	0,8333	0,7047	1,0000	0,375	0,5455
	50	0,8000	0,7500	0,7742	0,9333	0,4117	0,5714
	75	0,8000	0,7500	0,7742	1,0000	0,3947	0,566
History	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Fantasy	2	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Sci-Fi	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Horror	2	0,7500	0,5000	0,6000	0,7500	0,0938	0,1667
	10	0,7500	1,0000	0,8571	1,0000	0,1333	0,2353
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Comedy	2	0,5000	0,6667	0,5714	0,0000	0,0000	0,0000
	10	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Crime	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Musical	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Adventure	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,4167	1,0000	0,5882	0,0000	0,0000	0,0000
	25	0,4167	0,8333	0,5556	0,0000	0,0000	0,0000
	50	0,1667	1,0000	0,2857	0,0000	0,0000	0,0000
	75	0,1667	1,0000	0,2857	0,0000	0,0000	0,0000
War	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Western	2	0,5000	0,6667	0,5714	0,2500	0,0000	0,4000
	10	0,5000	1,0000	0,6667	0,2500	0,0000	0,4000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Pada tabel 5.1 menunjukkan nilai *recall*, *precision*, dan *F-measure* yang dihasilkan pada pengujian pengaruh penggunaan jumlah data *training* 75 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak. Nilai-nilai tersebut dihitung pada setiap kategori yang digunakan pada penelitian.

Untuk hasil pengujian pengaruh penggunaan jumlah data *training* 100 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak ditunjukkan pada tabel 5.2.

Tabel 5.2 Hasil Pengujian Pengaruh Jumlah Data *Training* Sebanyak 100 Data, Nilai *k*, dan Metode Pencarian Jarak

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	2	0,8667	0,8667	0,8667	0,4000	1,0000	0,5714
	10	0,8000	0,8000	0,8000	0,0000	0,0000	0,0000
	25	0,8667	0,6842	0,7647	0,0000	0,0000	0,0000
	50	1,0000	0,6522	0,7895	0,0000	0,0000	0,0000
	75	0,8667	0,5652	0,6842	0,2000	0,7500	0,3158
Drama	2	0,6000	0,8182	0,6923	0,1333	1,0000	0,2353
	10	0,8000	1,0000	0,8889	0,4667	0,5000	0,4828
	25	0,7333	0,9167	0,8148	1,0000	0,3750	0,5455
	50	0,8000	0,9231	0,8571	1,0000	0,3750	0,5455
	75	0,8000	0,9231	0,8571	1,0000	0,4167	0,5882
History	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Fantasy	2	0,3750	1,0000	0,5455	0,1250	1,0000	0,2222
	10	0,1250	1,0000	0,2222	0,0000	0,0000	0,0000

	25	0,1250	1,0000	0,2222	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Sci-Fi	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Horror	2	0,7500	0,6000	0,6667	0,7500	0,1000	0,1765
	10	0,7500	1,0000	0,8571	0,7500	0,1200	0,2069
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Comedy	2	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	10	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Crime	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Musical	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Adventure	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	10	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
War	2	0,2000	1,0000	0,3333	0,0000	0,0000	0,0000
	10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Western	2	0,7500	1,0000	0,8571	0,2500	1,0000	0,4000
	10	0,7500	1,0000	0,8571	0,2500	1,0000	0,4000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Pada tabel 5.2 menunjukkan nilai *recall*, *precision*, dan *F-measure* yang dihasilkan pada pengujian pengaruh penggunaan jumlah data *training* 100 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak. Nilai-nilai tersebut dihitung pada setiap kategori yang digunakan pada penelitian.

Untuk hasil pengujian pengaruh penggunaan jumlah data *training* 125 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil Pengujian Pengaruh Jumlah Data *Training* Sebanyak 125 Data, Nilai *k*, dan Metode Pencarian Jarak

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	5	0,7333	1,0000	0,8462	0,1333	1,0000	0,2353
	25	0,8000	0,8000	0,8000	0,0000	0,0000	0,0000
	50	0,8000	0,6316	0,7059	0,0000	0,0000	0,0000
	75	0,9333	0,6087	0,7368	0,0000	0,0000	0,0000
	100	0,9333	0,6087	0,7368	0,1333	0,6667	0,2222
Drama	5	0,5333	0,7273	0,6154	0,0667	1,0000	0,1250
	25	0,8000	1,0000	0,8889	1,0000	0,3750	0,5455
	50	0,8000	0,8571	0,8276	1,0000	0,3750	0,5455
	75	0,8000	0,8571	0,8276	1,0000	0,3750	0,5455
	100	0,8000	0,8571	0,8276	0,9333	0,3784	0,5385
History	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Fantasy	5	0,3750	1,0000	0,5455	0,0000	0,0000	0,0000
	25	0,1250	1,0000	0,2222	0,0000	0,0000	0,0000
	50	0,1250	1,0000	0,2222	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Sci-Fi	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Horror	5	0,5000	1,0000	0,6667	1,0000	0,1111	0,2000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Comedy	5	0,7500	1,0000	0,8571	0,0000	0,0000	0,0000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Crime	5	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Musical	5	0,7500	0,7500	0,7500	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Adventure	5	0,1667	1,0000	0,2857	0,0000	0,0000	0,0000
	25	0,4167	1,0000	0,5882	0,0000	0,0000	0,0000
	50	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
	75	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	100	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
War	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Western	5	0,7500	1,0000	0,8571	0,2500	1,0000	0,4000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Pada tabel 5.3 menunjukkan nilai *recall*, *precision*, dan *F-measure* yang dihasilkan pada pengujian pengaruh penggunaan jumlah data *training* 125 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak. Nilai-nilai tersebut dihitung pada setiap kategori yang digunakan pada penelitian.

Untuk hasil pengujian pengaruh penggunaan jumlah data *training* 150 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak ditunjukkan pada tabel 5.4.

Tabel 5.4 Hasil Pengujian Pengaruh Jumlah Data *Training* Sebanyak 150 Data, Nilai *k*, dan Metode Pencarian Jarak

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	5	0,7333	1,0000	0,8462	0,2000	1,0000	0,3333
	25	0,7333	0,6875	0,7097	0,0000	0,0000	0,0000
	50	0,8667	0,6190	0,7222	0,0000	0,0000	0,0000
	75	0,8667	0,5652	0,6842	0,0000	0,0000	0,0000
	100	0,9333	0,6087	0,7368	0,0000	0,0000	0,0000
Drama	5	0,6000	0,7500	0,6667	0,1333	1,0000	0,2353
	25	0,8000	0,9231	0,8571	0,9333	0,3590	0,5185
	50	0,7333	0,8462	0,7857	1,0000	0,3750	0,5455
	75	0,7333	0,8462	0,7857	1,0000	0,3750	0,5455
	100	0,8000	0,8571	0,8276	1,0000	0,3750	0,5455
History	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Fantasy	5	0,3750	1,0000	0,5455	0,0000	0,0000	0,0000
	25	0,1250	1,0000	0,2222	0,0000	0,0000	0,0000
	50	0,1250	1,0000	0,2222	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Sci-Fi	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Horror	5	0,5000	1,0000	0,6667	1,0000	0,1250	0,2222
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Comedy	5	0,7500	1,0000	0,8571	0,0000	0,0000	0,0000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Crime	5	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Musical	5	0,5000	0,6667	0,5714	0,2500	1,0000	0,4000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Adventure	5	0,1667	1,0000	0,2857	0,0000	0,0000	0,0000
	25	0,4167	1,0000	0,5882	0,0000	0,0000	0,0000
	50	0,2500	0,7500	0,3750	0,0000	0,0000	0,0000
	75	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
	100	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
War	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Western	5	0,7500	1,0000	0,8571	0,5000	1,0000	0,6667
	25	0,2500	1,0000	0,4000	0,2500	1,0000	0,4000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Pada tabel 5.4 menunjukkan nilai *recall*, *precision*, dan *F-measure* yang dihasilkan pada pengujian pengaruh penggunaan jumlah data *training* 150 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak. Nilai-nilai tersebut dihitung pada setiap kategori yang digunakan pada penelitian.

Untuk hasil pengujian pengaruh penggunaan jumlah data *training* 175 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak ditunjukkan pada tabel 5.5.

Tabel 5.5 Hasil Pengujian Pengaruh Jumlah Data *Training* Sebanyak 175 Data, Nilai *k*, dan Metode Pencarian Jarak

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	5	0,7333	1,0000	0,8462	0,2000	1,0000	0,3333
	25	0,8667	0,7222	0,7879	0,0000	0,0000	0,0000
	50	0,8000	0,5714	0,6667	0,0000	0,0000	0,0000
	75	0,9333	0,6087	0,7368	0,0000	0,0000	0,0000
	100	0,9333	0,6087	0,7368	0,0000	0,0000	0,0000
Drama	5	0,6667	0,7692	0,7143	0,2000	1,0000	0,3333
	25	0,8000	0,9231	0,8571	1,0000	0,3750	0,5455
	50	0,6667	0,8333	0,7407	1,0000	0,3750	0,5455
	75	0,8000	0,8571	0,8276	1,0000	0,3750	0,5455
	100	0,8000	0,8571	0,8276	1,0000	0,3750	0,5455
History	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Fantasy	5	0,3750	1,0000	0,5455	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Sci-Fi	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Horror	5	0,7500	1,0000	0,8571	1,0000	0,1429	0,2500
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Comedy	5	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Crime	5	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Musical	5	0,2500	0,5000	0,3333	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Adventure	5	0,1667	1,0000	0,2857	0,1667	1,0000	0,2857
	25	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
	50	0,4167	0,8333	0,5556	0,0000	0,0000	0,0000
	75	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	100	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
War	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Western	5	0,7500	1,0000	0,8571	0,7500	0,7500	0,7500
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Pada tabel 5.5 menunjukkan nilai *recall*, *precision*, dan *F-measure* yang dihasilkan pada pengujian pengaruh penggunaan jumlah data *training* 175 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak. Nilai-nilai tersebut dihitung pada setiap kategori yang digunakan pada penelitian.

Untuk hasil pengujian pengaruh penggunaan jumlah data *training* 200 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak ditunjukkan pada tabel 5.6.

Tabel 5.6 Hasil Pengujian Pengaruh Jumlah Data *Training* Sebanyak 200 Data, Nilai *k*, dan Metode Pencarian Jarak

Genre	*k*	*Cosine Similarity*			*Euclidean Distance*		
		R	*P*	*F*	*R*	*P*	*F*
Action	5	0,7333	1,0000	0,8462	0,1333	1,0000	0,2353
	25	0,8667	0,7647	0,8125	0,0000	0,0000	0,0000
	50	0,8000	0,6000	0,6857	0,0000	0,0000	0,0000
	75	0,8667	0,5652	0,6842	0,0000	0,0000	0,0000
	100	0,9333	0,5833	0,7179	0,0000	0,0000	0,0000
Drama	5	0,6000	0,8182	0,6923	0,2667	0,5000	0,3478
	25	0,7333	0,9167	0,8148	1,0000	0,3750	0,5455
	50	0,7333	0,8462	0,7857	1,0000	0,3750	0,5455
	75	0,7333	0,8462	0,7857	1,0000	0,3750	0,5455
	100	0,7333	0,8462	0,7857	1,0000	0,3750	0,5455
History	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Fantasy	5	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Sci-Fi	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Horror	5	0,7500	1,0000	0,8571	0,7500	0,1667	0,2727
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Comedy	5	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Crime	5	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Musical	5	0,5000	0,6667	0,5714	0,0000	0,0000	0,0000
	25	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Adventure	5	0,0833	0,5000	0,1429	0,3333	0,0000	0,5000
	25	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
	50	0,4167	1,0000	0,5882	0,0000	0,0000	0,0000
	75	0,3333	1,0000	0,5000	0,0000	0,0000	0,0000
	100	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
War	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	25	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Western	5	0,7500	1,0000	0,8571	0,7500	0,4286	0,5455
	25	0,5000	1,0000	0,6667	0,0000	0,0000	0,0000
	50	0,2500	1,0000	0,4000	0,0000	0,0000	0,0000
	75	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

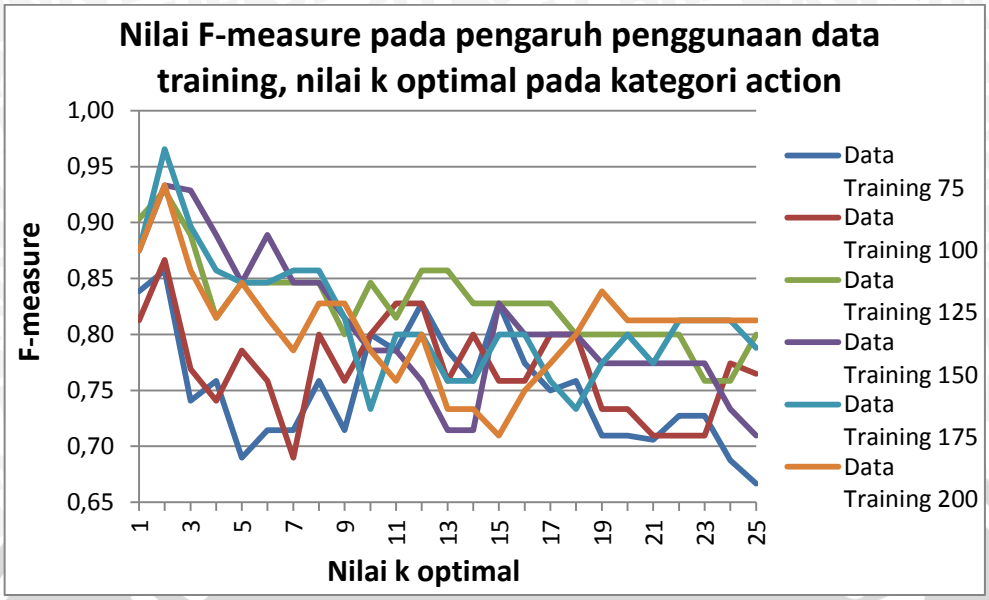
Pada tabel 5.6 menunjukkan nilai *recall*, *precision*, dan *F-measure* yang dihasilkan pada pengujian pengaruh penggunaan jumlah data *training* 200 data, nilai *k* (jumlah tetangga terdekat), dan metode pencarian jarak. Nilai-nilai tersebut dihitung pada setiap kategori yang digunakan pada penelitian.

5.3 Analisis Hasil Implementasi Pengujian

Pada subbab ini akan dijelaskan mengenai pembahasan dan analisis dari hasil implementasi pengujian. Analisis yang dibahas meliputi analisis hasil implementasi pengujian pengaruh jumlah data *training* dan nilai *k* (jumlah tetangga terdekat), dan hasil implementasi pengujian pengaruh penggunaan metode pencarian jarak.

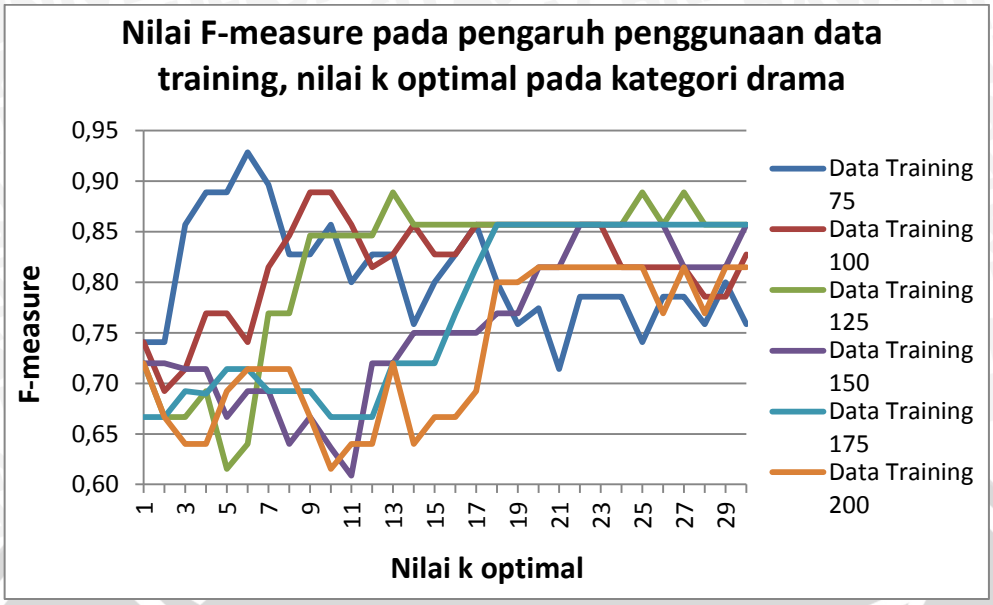
5.3.1 Analisis Hasil Implementasi Pengujian Pengaruh Jumlah Data *Training* dan Nilai *k* (Jumlah Tetangga Terdekat)

Dari hasil implementasi pengujian pengaruh jumlah data *training* dan nilai *k*, diketahui bahwa kedua parameter tersebut mempengaruhi tingkat akurasi berdasarkan nilai *F-measure* yang dihasilkan oleh sistem pada setiap kategori. *F-measure* merupakan ukuran nilai relatif antara *recall*, dan *precision*. *Recall* merupakan nilai akurasi sistem dalam mengenali kategori tertentu tanpa melihat ketepatannya dalam mengenali kategori tertentu. Sementara *precision* merupakan nilai akurasi ketepatan sistem dalam melakukan klasifikasi pada kategori tertentu tanpa melihat banyak data yang dikenali pada kategori tertentu. Kemudian berdasarkan hasil tersebut maka dapat dibuat suatu grafik hubungan antara jumlah data *training*, nilai *k* dan nilai tingkat akurasi pada setiap kategori. Dari grafik tersebut dapat digunakan sebagai media dalam melakukan analisis terhadap hasil implementasi pengujian yang terkait agar diketahui penggunaan jumlah data training berapakah dan nilai *k* optimal berapakah yang menghasilkan nilai akurasi paling baik untuk setiap kategori. Grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *action* ditunjukkan pada gambar 5.1.



Gambar 5.1 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Action*

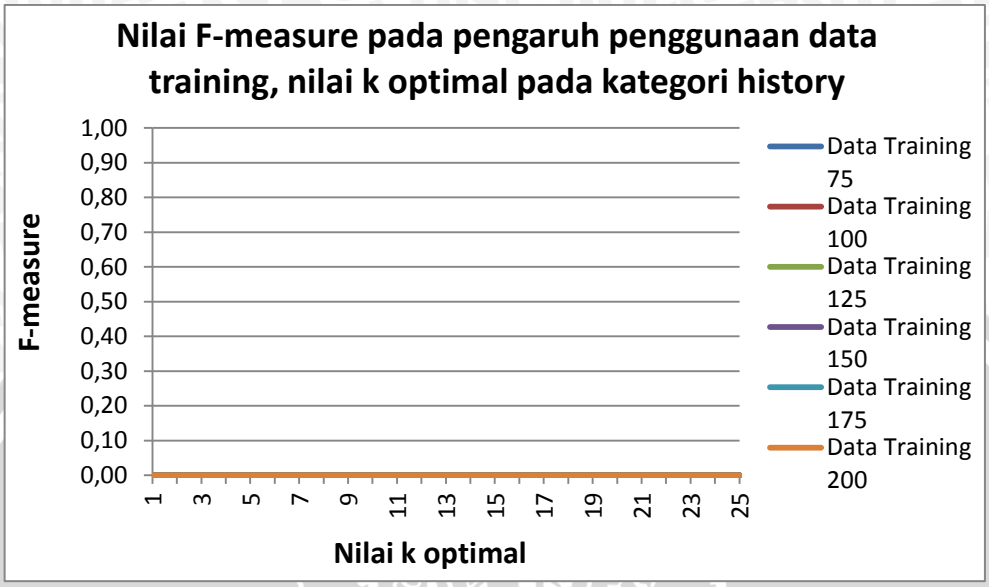
Pada gambar 5.1, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *action*. Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan data *training* sebanyak 175 data. Untuk titik *k* yang menghasilkan nilai akurasi paling baik adalah pada titik *k* = 2 dengan nilai akurasinya sebesar 0,9286. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 2 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 2 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *drama* ditunjukkan pada gambar 5.2.



Gambar 5.2 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Drama*

Pada gambar 5.2, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *drama*. Dari grafik terlihat bahwa nilai akurasi cenderung naik turun seperti pada grafik kategori *action*, disemua titik *k* optimal. Namun untuk penggunaan jumlah data training yang paling optimal adalah pada data training 100 data, karena nilai *F-measure* yang dihasilkan lebih stabil dan tidak mengalami kenaikan atau penurunan yang drastis, selain itu rata-rata *F-measure* yang dihasilkan pada penggunaan data *training* 100 data lebih tinggi dibandingkan pada penggunaan data *training* yang lain. Untuk titik *k* yang menghasilkan nilai akurasi paling baik adalah pada titik $k = 9$, dan $k = 10$ dengan nilai akurasinya sebesar 0,8889. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 9 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 10 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

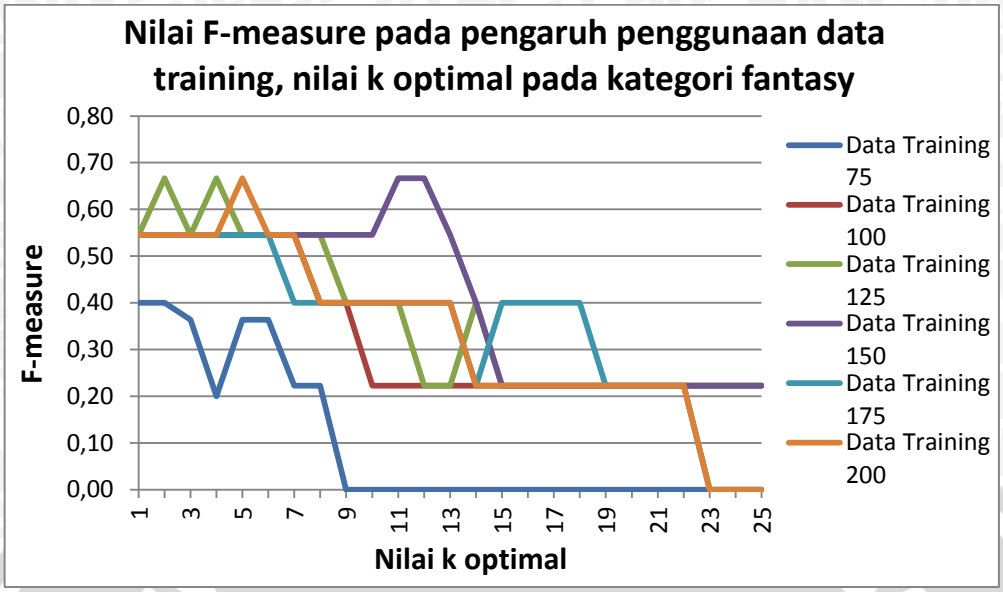
Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *history* ditunjukkan pada gambar 5.3.



Gambar 5.3 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *History*

Pada gambar 5.3, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai k optimal pada kategori *history*. Dari grafik terlihat bahwa nilai akurasi yang dihasilkan selalu bernilai 0 pada seluruh variasi data *training*, dan juga pada keseluruhan nilai *k*. Hal ini disebabkan karena sistem tidak berhasil mengenali data dengan kategori *history*, dibuktikan dengan nilai *recall* bernilai 0 pada keseluruhan variasi data *training*, dan juga pada keseluruhan nilai *k*. Sistem justru mengenali data kategori *history* kedalam kategori *drama*. Pada kenyataannya data sinopsis kategori *history* hampir selalu memiliki kategori *drama* dan *history* didalamnya. Karena jumlah kategori *drama* yang lebih mendominasi diantara kategori yang lain, maka dimungkinkan data kategori *history* justru dikenali sebagai data kategori *drama*, bukan data kategori *history*.

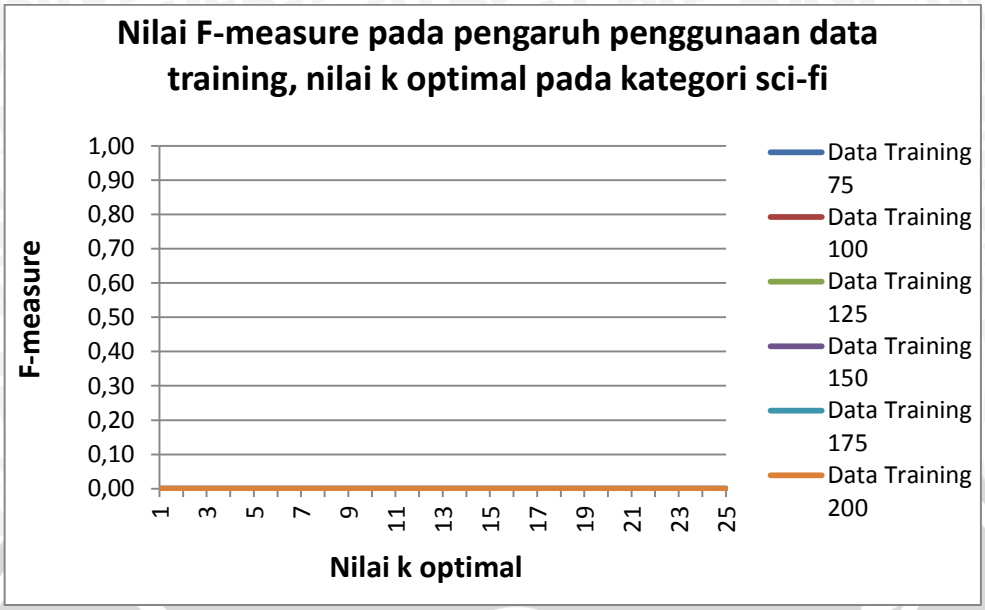
Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *fantasy* ditunjukkan pada gambar 5.4.



Gambar 5.4 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Fantasy*

Pada gambar 5.4, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *fantasy*. Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan jumlah data *training* 150 data. Penggunaan jumlah data *training* 150 juga yang paling optimal untuk kategori *fantasy*, karena nilai rata-rata *F-measure* yang paling baik diantara variasi penggunaan jumlah data *training* yang lain. Untuk nilai *k* yang optimal berada pada titik $k = 11$, dan $k = 12$ dengan nilai akurasinya sebesar 0,6667. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 11 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 12 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

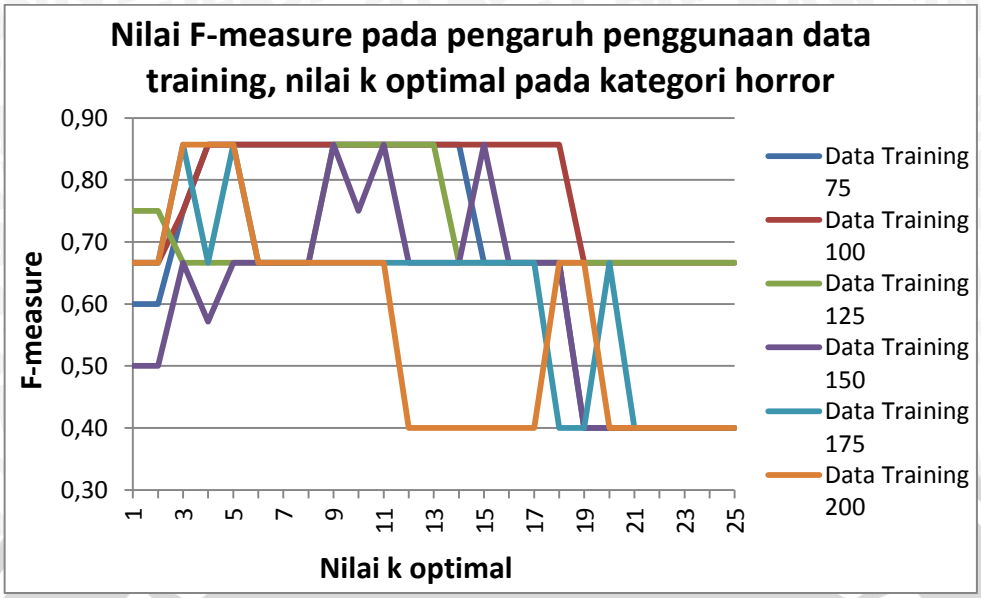
Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *sci-fi* ditunjukkan pada gambar 5.5.



Gambar 5.5 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Sci-Fi*

Pada gambar 5.5, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *sci-fi*. Dari grafik terlihat bahwa nilai akurasi yang dihasilkan selalu bernilai 0 pada seluruh variasi data *training*, dan juga pada keseluruhan nilai *k*. Sama seperti pada kategori *history*, hal ini disebabkan karena sistem tidak berhasil mengenali data dengan kategori *sci-fi*, dibuktikan dengan nilai *recall* bernilai 0 pada keseluruhan variasi data *training*, dan juga pada keseluruhan nilai *k*. Sistem justru mengenali data kategori *sci-fi* kedalam kategori *action*. Pada kenyataannya data sinopsis kategori *sci-fi* selalu memiliki kategori *action* dan *sci-fi* didalamnya. Karena jumlah kategori *action* yang lebih mendominasi diantara kategori yang lain, maka dimungkinkan data kategori *sci-fi* justru dikenali sebagai data kategori *action*, bukan data kategori *sci-fi*.

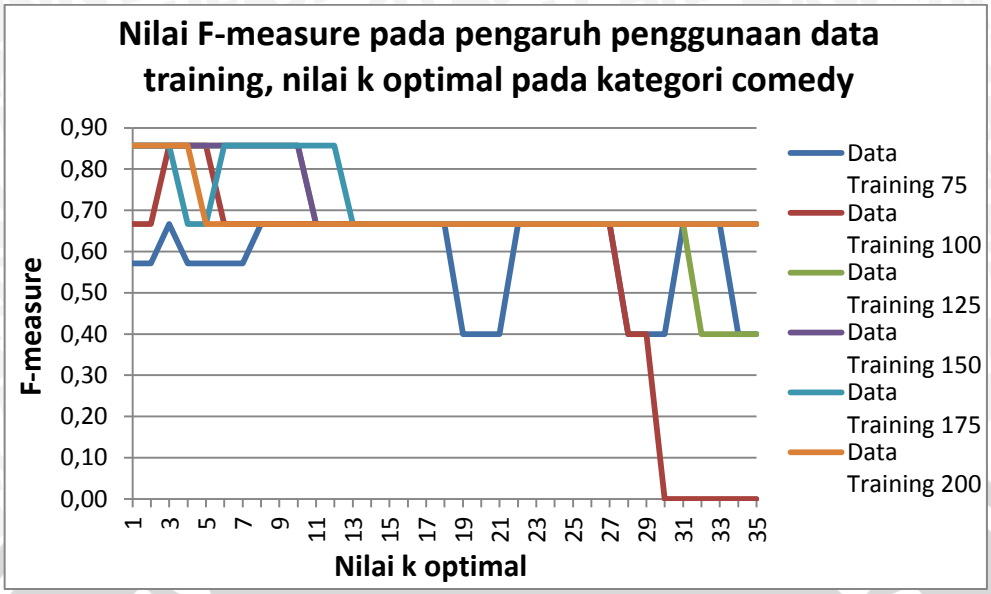
Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *horror* ditunjukkan pada gambar 5.6.



Gambar 5.6 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Horror*

Pada gambar 5.6, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *horror*. Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan jumlah data *training* 100 data. Penggunaan jumlah data *training* 100 juga yang paling optimal untuk kategori *horror*, karena nilai rata-rata *F-measure* yang paling baik diantara variasi penggunaan jumlah data *training* yang lain. Untuk nilai *k* yang optimal berada pada titik *k* = 4 hingga *k* = 18 dengan nilai akurasinya sebesar 0,8571. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 4 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 18 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

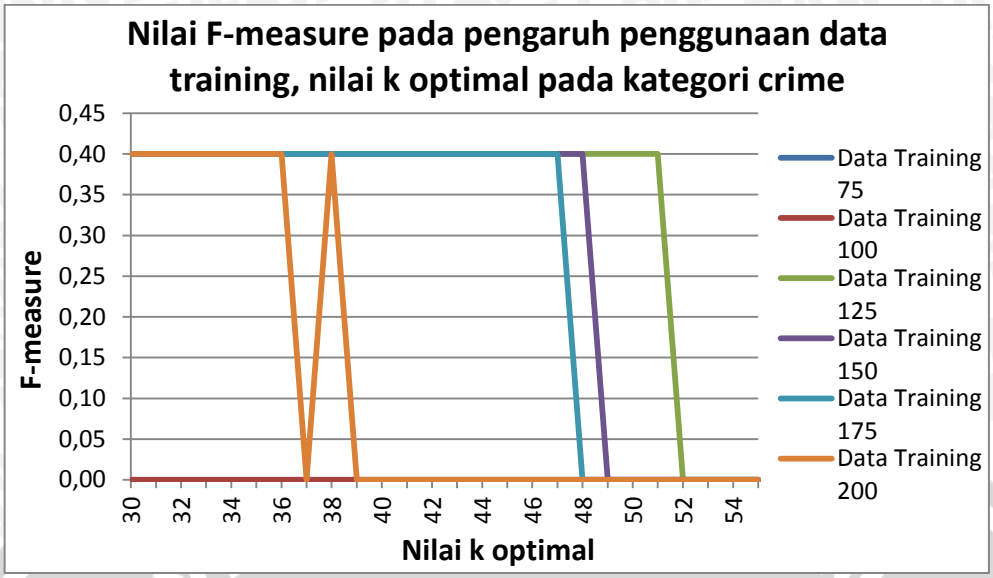
Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *comedy* ditunjukkan pada gambar 5.7.



Gambar 5.7 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Comedy*

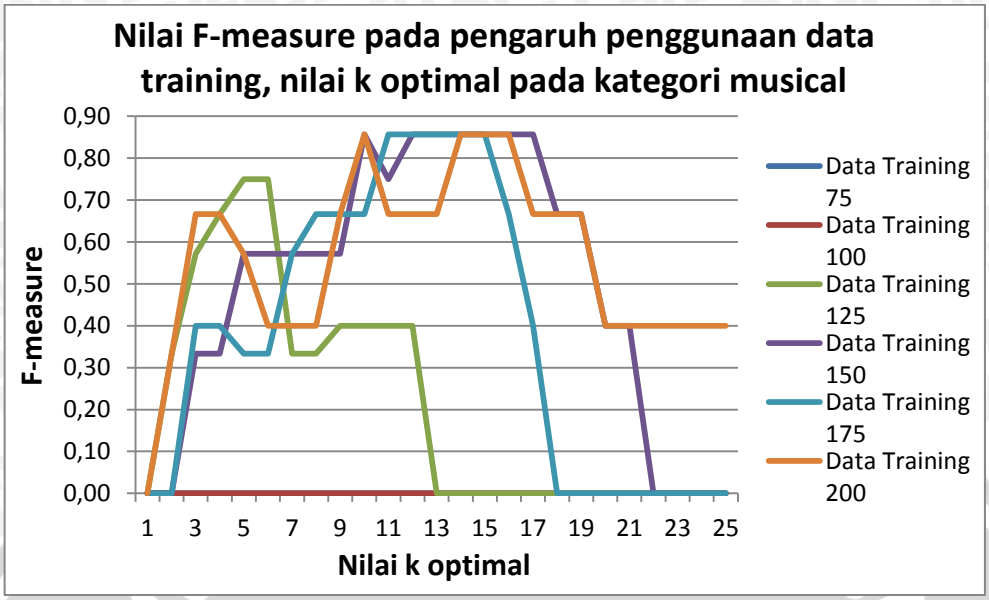
Pada gambar 5.7, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *comedy*. Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan jumlah data *training* 175 data. Penggunaan jumlah data *training* 175 juga yang paling optimal untuk kategori *comedy*, karena nilai rata-rata *F-measure* yang paling baik diantara variasi penggunaan jumlah data *training* yang lain. Untuk nilai *k* yang optimal berada pada titik *k* = 6 hingga *k* = 12 dengan nilai akurasinya sebesar 0,8571. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 6 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 12 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *crime* ditunjukkan pada gambar 5.8.



Gambar 5.8 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Crime*

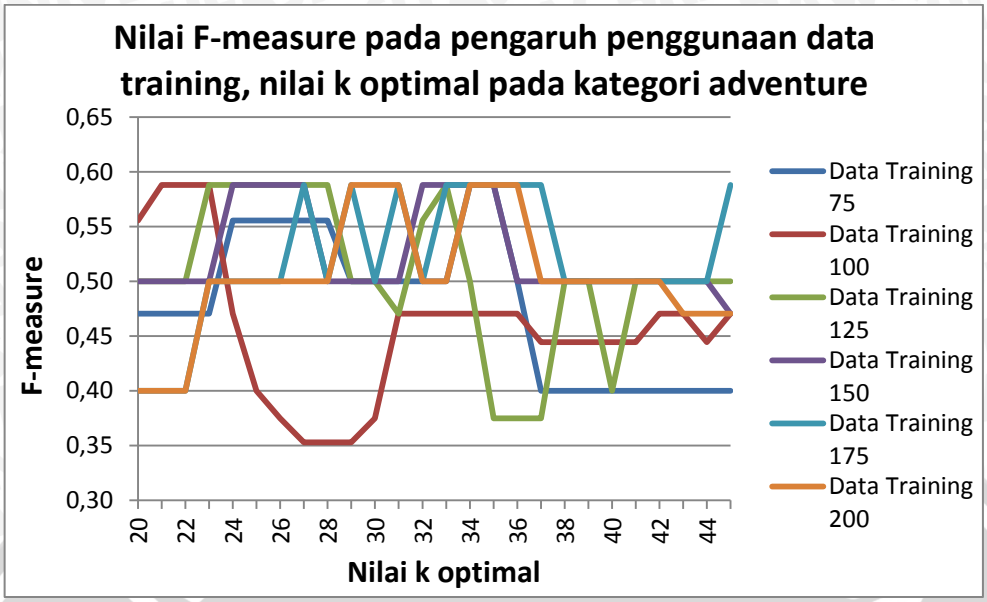
Pada gambar 5.8, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *crime.* Dari grafik terlihat bahwa nilai akurasi paling optimal berada pada penggunaan data *training* 125. Walaupun pada variasi penggunaan data *training* yang lain dihasilkan nilai akurasi yang sama tingginya, penggunaan data *training* 125 dinilai lebih optimal karena nilai rata-rata *F-measure* yang lebih baik jika dibandingkan variasi penggunaan data training yang lain. Untuk penggunaan nilai *k*, cenderung stabil dari titik *k* = 1 hingga *k* = 50 dengan nilai akurasinya sebesar 0,4000. Namun nilai akurasi akan menurun ke titik 0 ketika nilai *k* yang digunakan lebih dari 52. Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *musical* ditunjukkan pada gambar 5.9.



Gambar 5.9 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Musical*

Pada gambar 5.9, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *musical.* Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan jumlah data *training* 175 data. Penggunaan jumlah data *training* 175 juga yang paling optimal untuk kategori *musical*, karena nilai rata-rata *F-measure* yang paling baik diantara variasi penggunaan jumlah data *training* yang lain. Untuk nilai *k* yang optimal berada pada titik *k* = 14 hingga *k* = 16 dengan nilai akurasinya sebesar 0,8571. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 14 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 16 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

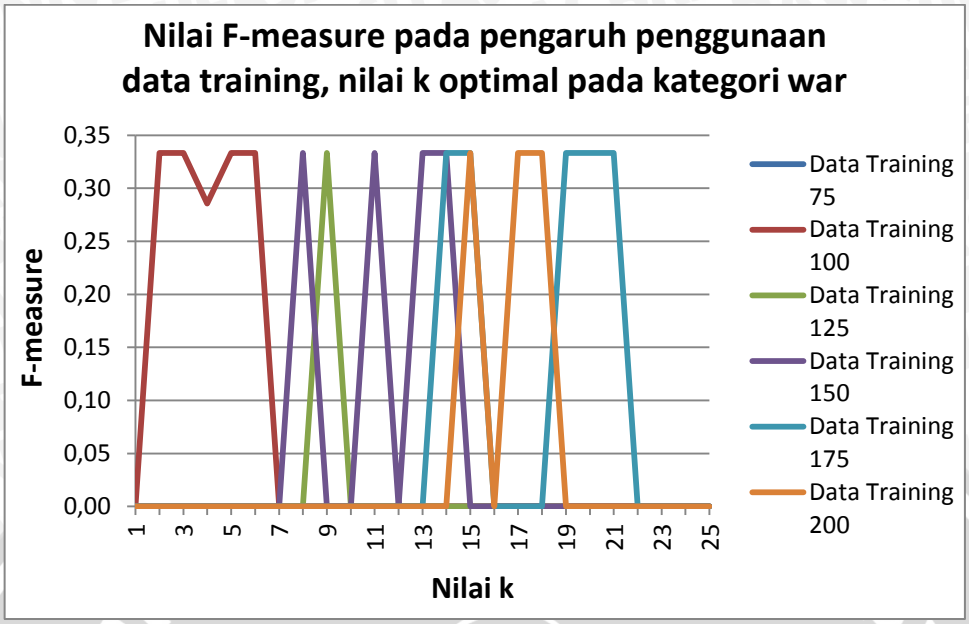
Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *adventure* ditunjukkan pada gambar 5.10.



Gambar 5.10 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Adventure*

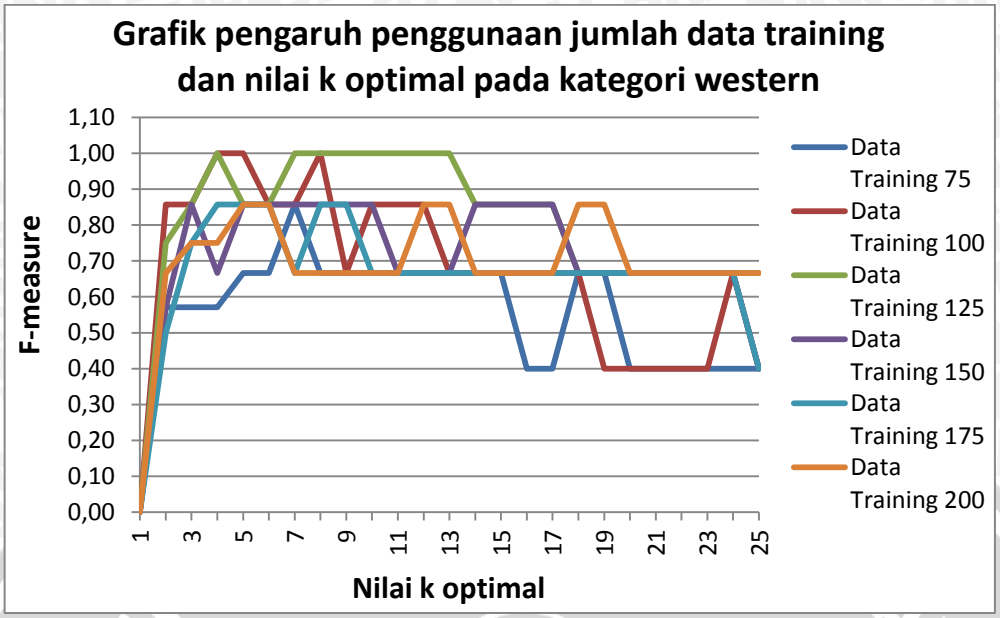
Pada gambar 5.10, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *adventure*. Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan jumlah data *training* 175 data. Penggunaan jumlah data *training* 175 juga yang paling optimal untuk kategori *adventure*, karena nilai rata-rata *F-measure* yang paling baik diantara variasi penggunaan jumlah data *training* yang lain. Untuk nilai *k* yang optimal berada pada titik *k* = 33 hingga *k* = 37 dengan nilai akurasinya sebesar 0,5882. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 33 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 37 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *war* ditunjukkan pada gambar 5.11.



Gambar 5.11 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *War*

Pada gambar 5.11, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *war*. Dari grafik terlihat bahwa nilai akurasi yang dihasilkan cenderung tidak stabil karena terjadi kenaikan dan penurunan secara drastis. Penggunaan jumlah data *training* 100 data dinilai sebagai yang mendekati optimal, karena memiliki rata-rata *F-measure* yang paling baik, dan juga menghasilkan nilai *F-measure* yang relatif stabil jika dibandingkan dengan penggunaan variasi data *training* yang lain. Untuk titik *k* yang optimal berada pada titik $k = 5$, dan $k = 6$ dengan nilai akurasinya sebesar 0,3333. Untuk grafik tingkat akurasi pada pengaruh penggunaan jumlah data *training*, nilai *k* optimal pada kategori *western* ditunjukkan pada gambar 5.12.



Gambar 5.12 Grafik Nilai *F-measure* Pada Pengaruh Penggunaan Data *Training*, dan Nilai *k* Optimal Pada Kategori *Western*

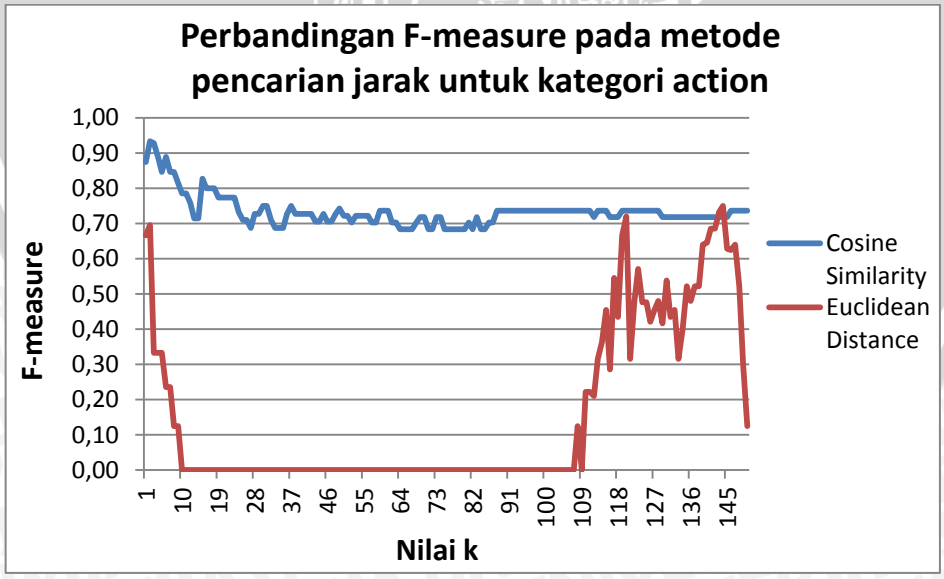
Pada gambar 5.12, menunjukkan grafik nilai *F-measure* pada variasi pengaruh penggunaan data *training*, nilai *k* optimal pada kategori *western.* Dari grafik terlihat bahwa nilai akurasi tertinggi berada pada penggunaan jumlah data *training* 125 data. Untuk nilai *k* yang optimal berada pada titik *k* = 7 hingga *k* = 13 dengan nilai akurasinya sebesar 1,0000. Penggunaan nilai *k* yang terlalu terlalu kecil, atau pada rentang nilai *k* dibawah 7 menyebabkan data terdekat hanya terfokus pada kategori tertentu menghasilkan nilai akurasi yang kurang baik, sehingga nilai akurasi yang dihasilkan kurang baik. Sementara penggunaan nilai *k* yang terlalu besar, atau pada rentang nilai *k* diatas 13 menyebabkan data yang memiliki tingkat relevansi rendah kemungkinan bisa ikut terambil, sehingga nilai akurasi yang dihasilkan juga menjadi kurang baik.

Sebagai tambahan analisis, data pada penelitian mayoritas terdiri dari lebih dari 2 kategori *genre* didalamnya. Oleh sebab itu juga, terdapat kategori *genre* yang selalu muncul pada masing-masing data sehingga jumlahnya mendominasi dibandingkan jumlah kategori *genre* yang lain, yaitu *genre action*, dan *drama*. Hal ini mempengaruhi sistem dalam mengenali dan menentukan klasifikasi untuk kategori *genre* lain yang jumlahnya sedikit. Sebagai contoh data kategori *sci-fi*

yang selain terdiri dari kategori *sci-fi* itu sendiri, juga selalu terdiri kategori *action* pula. Sistem cenderung mengenali dan mengklasifikasi sebagai data kategori *action*. Hal ini dimungkinkan karena jumlah kategori *action* yang mendominasi diantara kategori yang lain.

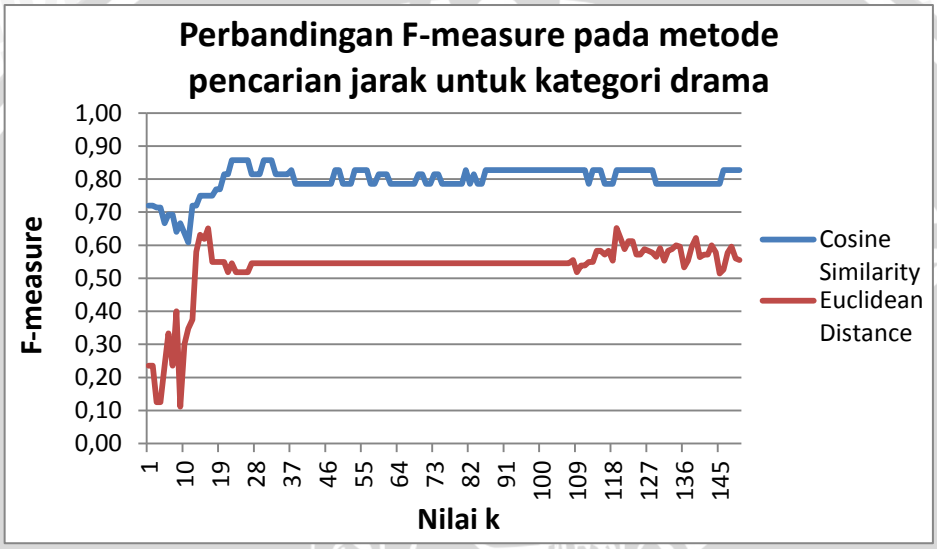
5.3.2 Analisis Hasil Implementasi Pengujian Pengaruh Penggunaan Metode Pencarian Jarak

Pada hasil implementasi pengujian pengaruh penggunaan metode pencarian jarak, yaitu metode *cosine similarity* dan *euclidean distance* didapatkan hasil tingkat akurasi berdasarkan nilai *F-measure* yang berbeda setiap pengujian penggunaan metode tersebut pada sejumlah data *training*, dan pada setiap kategori. Berdasarkan hasil tersebut maka dapat dibuat suatu grafik hubungan antara penggunaan metode pencarian jarak, jumlah data *training*, nilai *k* dan nilai tingkat akurasi untuk setiap kategori. Untuk analisis ini digunakan hasil pengujian pada variasi data *training* 150 data, karena jumlahnya yang tidak terlalu banyak dan tidak terlalu sedikit pula. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *action* ditunjukkan pada gambar 5.13.



Gambar 5.13 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Action*

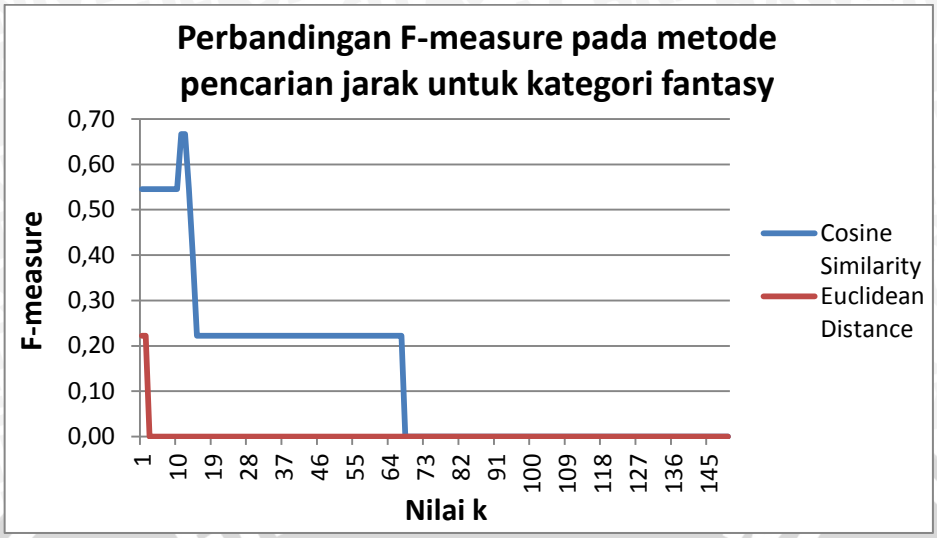
Pada gambar 5.13, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *action*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *drama* ditunjukkan pada gambar 5.14



Gambar 5.14 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Drama*

Pada gambar 5.14, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *drama*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*.

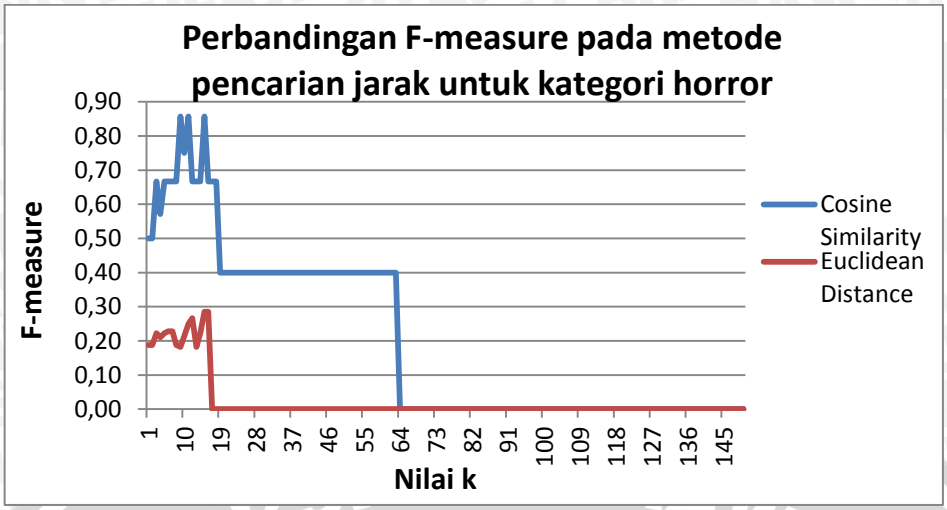
Untuk kategori *history*, karena sistem tidak berhasil mengenali dan menentukan hasil klasifikasi data kategori *history*, baik pada penggunaan metode *cosine similarity* dan *euclidean distance*, maka analisis pada kategori ini tidak dapat dilakukan. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *fantasy* ditunjukkan pada gambar 5.15.



Gambar 5.15 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Fantasy*

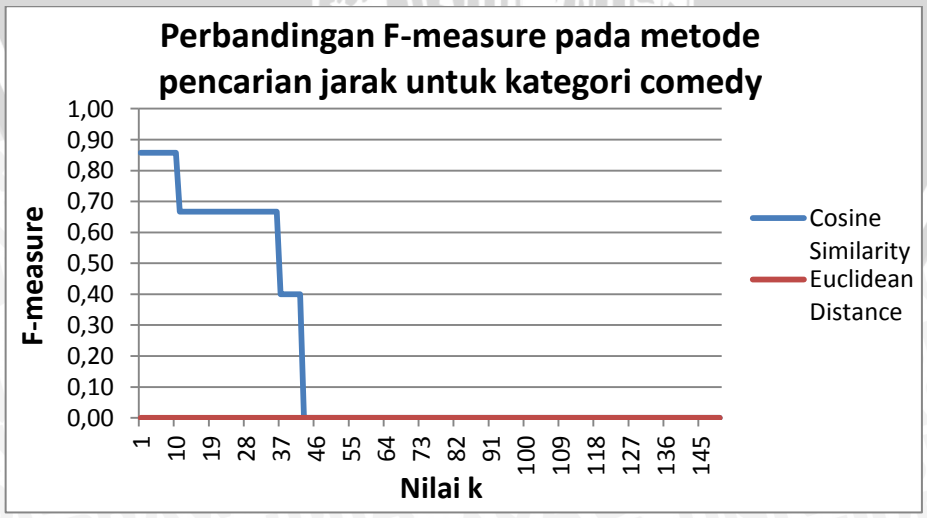
Pada gambar 5.15, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *fantasy*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*.

Untuk kategori *sci-fi*, karena sistem tidak berhasil mengenali dan menentukan hasil klasifikasi data kategori *sci-fi*, baik pada penggunaan metode *cosine similarity* dan *euclidean distance*, maka analisis pada kategori ini tidak dapat dilakukan. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *horror* ditunjukkan pada gambar 5.16



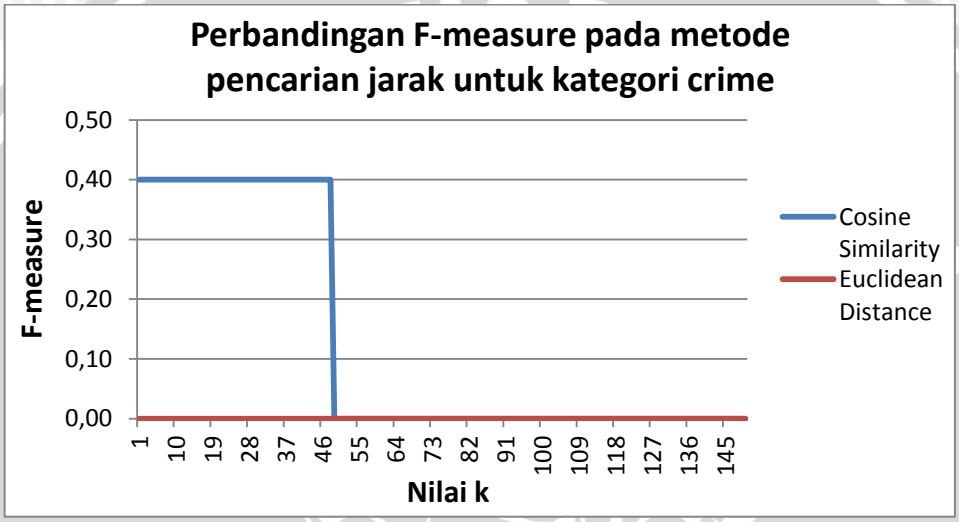
Gambar 5.16 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Horror*

Pada gambar 5.16, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *horror*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *comedy* ditunjukkan pada gambar 5.17



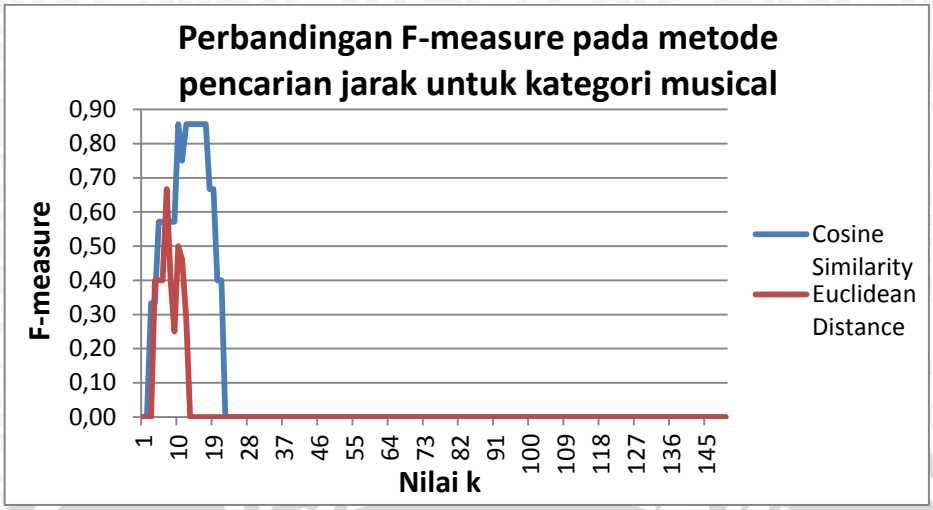
Gambar 5.17 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Comedy*

Pada gambar 5.17, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *comedy*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *crime* ditunjukkan pada gambar 5.18



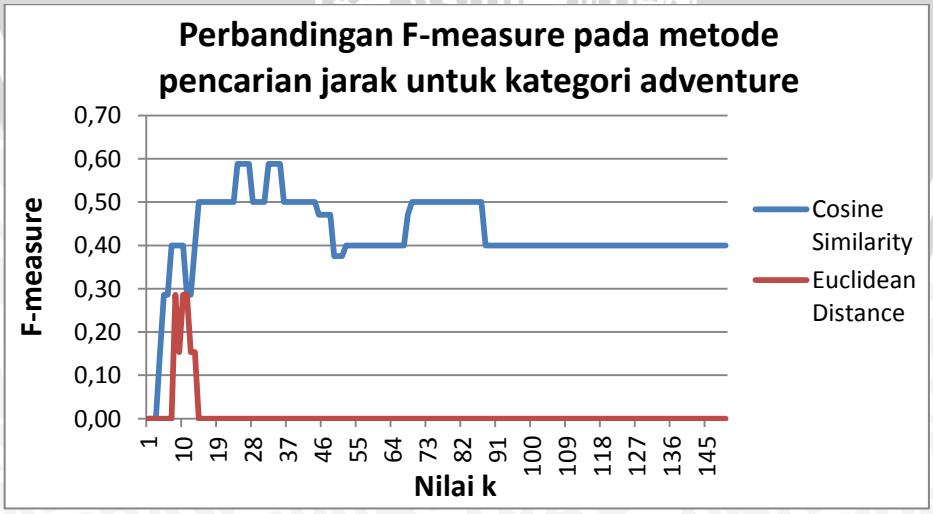
Gambar 5.18 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Crime*

Pada gambar 5.18, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *crime*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *musical* ditunjukkan pada gambar 5.19



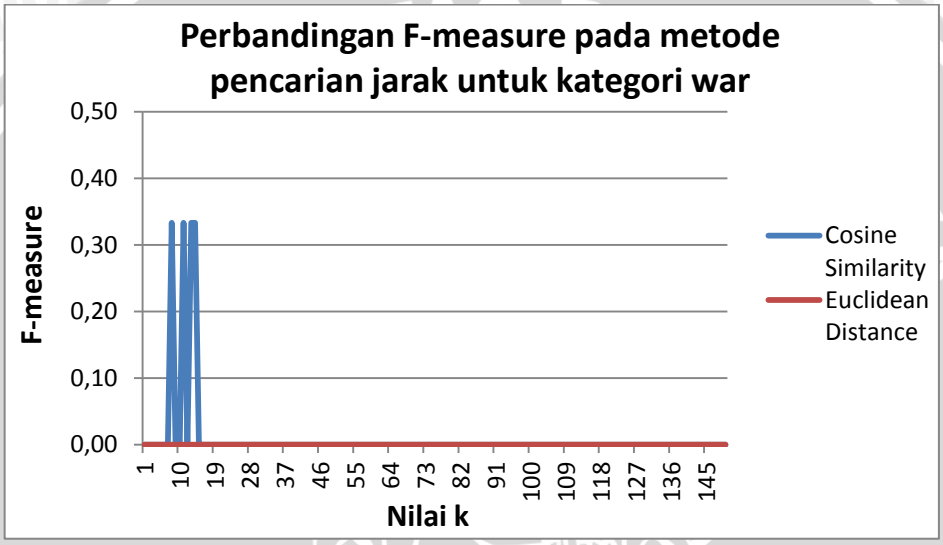
Gambar 5.19 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Musical*

Pada gambar 5.19, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *musical*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *adventure* ditunjukkan pada gambar 5.20.



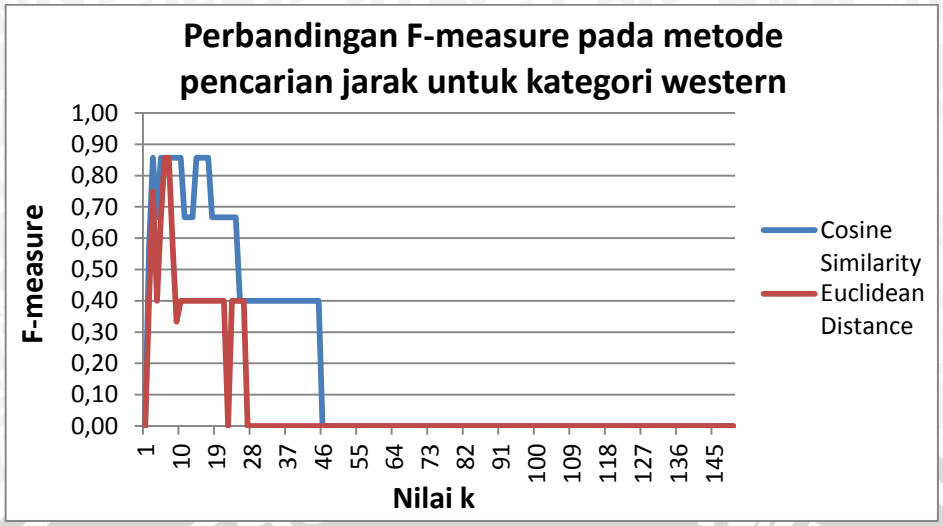
Gambar 5.20 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Adventure*

Pada gambar 5.20, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *adventure*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *war* ditunjukkan pada gambar 5.21.



Gambar 5.21 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *War*

Pada gambar 5.21, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *war*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*. Grafik perbandingan tingkat akurasi pada metode pencarian jarak untuk kategori *western* ditunjukkan pada gambar 5.22



Gambar 5.22 Perbandingan *F-measure* Pada Metode Pencarian Jarak Untuk Kategori *Western*

Pada gambar 5.22, menunjukkan grafik nilai tingkat akurasi pada perbedaan penggunaan metode pencarian jarak, pada data *training* sebanyak 150 yang diuji disemua titik *k* untuk kategori *western*. Dari grafik terlihat bahwa hasil tingkat akurasi yang dihasilkan penggunaan metode *cosine similarity* masih lebih baik jika dibandingkan metode *euclidean distance*.

Untuk hasil pengujian beserta analisis pengaruh penggunaan metode pencarian jarak pada keseluruhan kategori yang ditunjukkan dari gambar 5.13 hingga gambar 5.22 dapat diketahui bahwa nilai tingkat akurasi yang dihasilkan dari penggunaan metode *cosine similarity* selalu lebih baik jika dibandingkan metode *euclidean distance*. Hal ini berlaku pada seluruh pengujian pada penggunaan jumlah data *training* yang berbeda. Sehingga bisa disimpulkan bahwa penggunaan metode *cosine similarity* menghasilkan nilai tingkat akurasi yang lebih baik jika dibandingkan metode *euclidean distance*.

Ada hal yang menyebabkan mengapa metode *euclidean distance* menghasilkan nilai akurasi yang kurang baik, yaitu karena pada metode *euclidean distance* tidak mempunyai konsep normalisasi panjang vektor data. Sehingga konsep implementasi metode *euclidean distance* dipengaruhi oleh panjang vektor data. Jika suatu data *testing* dan data *training* yang akan dicari jaraknya memiliki panjang vektor yang tidak sama, atau bahkan berbeda jauh, maka hasil pencarian

jaraknya menjadi kurang akurat jika dilihat secara aktual yang juga berimbas pada hasil keakuratan klasifikasi data *testing*.

Berbeda dengan konsep implementasi metode *cosine similarity* yang mempunyai konsep normalisasi panjang vektor data. Metode *cosine similarity* dalam implementasinya tidak dipengaruhi oleh panjang vektor data. Hasil penghitungan kemiripan (*similarity*) menghasilkan akurasi yang baik jika dilihat secara aktual, sehingga hasil keakuratan klasifikasi data *testing* juga menjadi lebih baik.

BAB VI
PENUTUP

Pada bab penutup ini akan disampaikan kesimpulan dari hasil dan analisis penelitian tentang klasifikasi *genre* film berdasarkan judul dan sinopsisnya menggunakan metode *Fuzzy k-NN* yang telah dilakukan. Serta saran agar penelitian selanjutnya pada bidang yang sama dapat menghasilkan hasil penelitian yang baru.

6.1 Kesimpulan

Berdasarkan hasil dan analisis penelitian yang telah dibahas pada bab 5, didapatkan kesimpulan antara lain :

1. Implementasi metode *Fuzzy k-NN* dapat digunakan untuk melakukan klasifikasi *genre* film berdasarkan judul dan sinopsisnya. Langkah awal implementasi metode ini adalah dengan menghitung nilai kemiripan atau jarak vektor data *testing* terhadap vektor data *training*. Selanjutnya diambil sejumlah *k* tetangga terdekat sesuai hasil penghitungan kemiripan atau jarak. Kemudian dilakukan penghitungan nilai keanggotaan kelas untuk data *testing*. Data *testing* diklasifikasi kedalam kelas kategori yang memiliki nilai keanggotaan kelas untuk data *testing* yang paling tinggi.
2. Untuk pengaruh jumlah data *training* dan besarnya nilai *k* (jumlah tetangga terdekat) terhadap tingkat akurasi pada tiap kategori, didapatkan hasil bahwa kemampuan sistem dalam mengklasifikasi data tiap kategori sangat bervariatif. Tingkat akurasi tertinggi yang dihasilkan dari tiap kategori rata-rata berada pada penggunaan nilai *k* pada rentang $k = 10$ hingga $k = 20$. Penggunaan nilai *k* yang terlalu kecil atau terlalu besar menghasilkan tingkat akurasi yang kurang baik pada pengklasifikasian. Penggunaan nilai *k* yang terlalu kecil akan menyebabkan data terdekat hanya terfokus pada kategori tertentu. Sedangkan nilai *k* yang terlalu besar akan menyebabkan data yang memiliki tingkat relevansi rendah akan ikut terambil.
3. Pengaruh penggunaan metode *cosine similarity* dan *euclidean distance* pada penelitian ini telah didapatkan kesimpulan bahwa penggunaan metode *cosine*

similarity ternyata menghasilkan tingkat akurasi yang lebih baik jika dibandingkan dengan metode *euclidean distance*. Karena pada metode *cosine similarity* dalam implementasinya mempunyai konsep normalisasi panjang vektor data. Sedangkan pada metode *euclidean distance* dalam implementasinya tidak mempunyai konsep normalisasi panjang vektor data, sehingga nilai akurasi metode ini dipengaruhi oleh panjang data *training* dan data *testing* yang digunakan pada penelitian ini.

6.2 Saran

Berikut adalah saran agar penelitian selanjutnya pada bidang yang sama dapat menghasilkan hasil penelitian yang baru.

1. Data yang digunakan pada penelitian ini kurang berimbang. Cenderung lebih didominasi kategori *action*, dan *drama* saja. Misalnya, *genre action* selalu dimiliki oleh data sinopsis *genre sci-fi*, dan *crime*, sehingga jumlah *genre action* menjadi lebih dominan. Hal ini berimbas pada sistem yang justru mengenali data sinopsis *genre sci-fi*, dan *crime* sebagai data *genre action*. Untuk kedepannya perlu lebih diselaraskan antara jumlah *genre* yang digunakan pada penelitian yang sejenis, sehingga sistem bisa mengenali semua data untuk semua kategori.
2. Perlunya dikembangkan pencarian nilai *k* (jumlah tetangga terdekat) optimal secara otomatis. Karena dalam penelitian ini pencarian nilai *k* optimal masih dilakukan secara manual.

DAFTAR PUSTAKA

[ANO-92] Anonymous. 1992. Penjelasan Atas Undang-Undang Republik Indonesia Nomor 8 Tahun 1992 Tentang Perfilman, http://djpp.depkumham.go.id. Diakses pada tanggal 31 Oktober 2012.

[ANO-12] Anonymous. 2012. Ikhtisar, Resensi, Sinopsis dan Abstrak Bersaudara, Tapi Bukan Kembar Identik, http:// http://bahasa.kompasiana.com. Diakses pada tanggal 18 Desember 2013.

[GAR-05] Garcia, E.,Dr. 2005. *Document Indexing Tutorial*, http://www.miislita.com/information-retrieval-tutorial/indexing.html, diakses pada tanggal 31 Oktober 2012.

[HAN-12] Han, Jiawei, Micheline Kamber, dan Jian Pei. 2012. *Data Mining Concepts and Technique Third Edition*. Morgan Kaufmann : Waltham, Massachusetts.

[HEA-03] Hearst, Marti. 2003. *What Is Text Mining ?*, http://people.ischool.berkeley.edu/~hearst/text-mining.html. Diakses pada tanggal 30 Oktober 2012.

[JAC-02] Jackson, Peter dan Isabelle Moulinier. 2002. *Natural Language Processing for Online Applications : Text Retrieval, Extraction and Categorization*. John Benjamins Publishing Company : Philadelphia.

[JOA-98] Joachims, Thorsten. 1998. *Text Categorization with Support Vector Machines : Learning with Many Relevant Features*. Universität Dortmund : Dortmund.

[KUS-09] Kusrini, dan Luthfi, Emha Taufiq. 2009. Algoritma Data Mining. Andi Offset : Yogyakarta.

[KUS-10] Kusumadewi, Sri, dan Hari Purnomo. 2010. Aplikasi Logika Fuzzy untuk Pendukung Keputusan Edisi 2. Graha Ilmu : Yogyakarta.

[LIP-02] Li Ping Jing, Hou Kuan Huang, dan Hong Bo Shi. 2002. *Improved Feature Selection Approach TFIDF in Text Mining*. School of Computer & Information Technology Northern JiaoTong : University Beijing.

[POR-80] Porter, M.F. 1980. *An Algorithm for Suffix Striping*, Computer Laboratory. Corn Exchange Street : Cambridge.

[PRA-08] Pratista, Himawan. 2008. Memahami Film. Homerian Pustaka : Yogyakarta.

[PRA-12] Prasetyo, Eko. 2012. *Fuzzy k-Nearest Neighbour In Every Class* Untuk Klasifikasi Data. Seminar Nasional Teknik Informatika (SANTIKA 2012), 10 Maret 2012. Universitas Pembangunan Nasional Veteran Jawa Timur.

[SOU-05] Soucy, Pascal dan Guy Mineau. 2005. *Beyond TF IDF Weighting for Text Categorization in the Vector Space Model*.

[TUR-05] Turban, E., dkk. 2005. *Decision Support Systems and Intelligent Systems*. Andi Offset : Yogyakarta.

[WIS-13] Wisdarianto, Ardhy. 2013. Penerapan Metode *Fuzzy* k-*Nearest Neighbor* (FK-NN) Untuk Pengklasifikasian Spam Email. Tugas Akhir Program Studi Informatika / Ilmu Komputer Program Teknologi Informasi Dan Ilmu Komputer Universitas Brawijaya : Malang.

[YAN-08] Yang, Yiming dan Thorsten Joachims. 2008. *Text Categorization*, http://www.scholarpedia.org/article/Text_Categorization. Diakses pada tanggal 31 Oktober 2012.

[ZAF-08] Zafikri, Atika. 2008. Implementasi Metode Term Frequency Inverse Document Frequency (TF-IDF) Pada Sistem Temu Kembali Informasi. Tugas Akhir Program Studi Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara : Medan.

[ZHA-09] Zhang, Juan, Yi Niu, dan Huabei Nie. 2009. *Web Document Based on Fuzzy k-NN Algorithm*. Dongguan University of Technology : Guangdong.

LAMPIRAN

Lampiran 1 : Daftar *Stopword*

Daftar *term* yang termasuk dalam *stopword* untuk penelitian ini tercantum pada tabel 1.

Tabel 1 *Daftar Stopword*

No	Term	No	Term	No	Term
1	a	226	immediate	451	several
2	able	227	immediately	452	shall
3	about	228	importance	453	she
4	above	229	important	454	shed
5	abst	230	in	455	shell
6	accordance	231	inc	456	shes
7	according	232	indeed	457	should
8	accordingly	233	index	458	shouldnt
9	across	234	information	459	show
10	act	235	instead	460	showed
11	actually	236	into	461	shown
12	added	237	invention	462	showns
13	adj	238	inward	463	shows
14	adopted	239	is	464	significant
15	affected	240	isnt	465	significantly
16	affecting	241	it	466	similar
17	affects	242	itd	467	similarly
18	after	243	itll	468	since
19	afterwards	244	its	469	six
20	again	245	itself	470	slightly
21	against	246	ive	471	so
22	ah	247	j	472	some
23	all	248	just	473	somebody
24	almost	249	k	474	somehow
25	alone	250	keep	475	someone
26	along	251	keeps	476	somethan
27	already	252	kept	477	something
28	also	253	keys	478	sometime
29	although	254	kg	479	sometimes
30	always	255	km	480	somewhat
31	am	256	know	481	somewhere
32	among	257	known	482	soon
33	amongst	258	knows	483	sorry

34	an	259	l	484	specifically
35	and	260	largely	485	specified
36	announce	261	last	486	specify
37	another	262	lately	487	specifying
38	any	263	later	488	state
39	anybody	264	latter	489	states
40	anyhow	265	latterly	490	still
41	anymore	266	least	491	stop
42	anyone	267	less	492	strongly
43	anything	268	lest	493	sub
44	anyway	269	let	494	substantially
45	anyways	270	lets	495	successfully
46	anywhere	271	like	496	such
47	apparently	272	liked	497	sufficiently
48	approximately	273	likely	498	suggest
49	are	274	line	499	sup
50	aren	275	little	500	sure
51	arent	276	ll	501	t
52	arise	277	look	502	take
53	around	278	looking	503	taken
54	as	279	looks	504	taking
55	aside	280	ltd	505	tell
56	ask	281	m	506	tends
57	asking	282	made	507	th
58	at	283	mainly	508	than
59	auth	284	make	509	thank
60	available	285	makes	510	thanks
61	away	286	many	511	thanx
62	awfully	287	may	512	that
63	b	288	maybe	513	thatll
64	back	289	me	514	thats
65	be	290	mean	515	thatve
66	became	291	means	516	the
67	because	292	meantime	517	their
68	become	293	meanwhile	518	theirs
69	becomes	294	merely	519	them
70	becoming	295	mg	520	themselves
71	been	296	might	521	then
72	before	297	million	522	thence
73	beforehand	298	miss	523	there
74	begin	299	ml	524	thereafter
75	beginning	300	more	525	thereby

76	beginnings	301	moreover	526	thered
77	begins	302	most	527	therefore
78	behind	303	mostly	528	therein
79	being	304	mr	529	therell
80	believe	305	mrs	530	thereof
81	below	306	much	531	therere
82	beside	307	mug	532	theres
83	besides	308	must	533	thereto
84	between	309	my	534	thereupon
85	beyond	310	myself	535	thereve
86	biol	311	n	536	these
87	both	312	na	537	they
88	brief	313	name	538	theyd
89	briefly	314	namely	539	theyll
90	but	315	nay	540	theyre
91	by	316	nd	541	theyve
92	c	317	near	542	think
93	ca	318	nearly	543	this
94	came	319	necessarily	544	those
95	can	320	necessary	545	thou
96	cannot	321	need	546	though
97	cant	322	needs	547	thoughh
98	cause	323	neither	548	thousand
99	causes	324	never	549	throug
100	certain	325	nevertheless	550	through
101	certainly	326	new	551	throughout
102	co	327	next	552	thru
103	com	328	nine	553	thus
104	come	329	ninety	554	til
105	comes	330	no	555	tip
106	contain	331	nobody	556	to
107	containing	332	non	557	together
108	contains	333	none	558	too
109	could	334	nonetheless	559	took
110	couldnt	335	noone	560	toward
111	d	336	nor	561	towards
112	date	337	normally	562	tried
113	did	338	nos	563	tries
114	didnt	339	not	564	truly
115	different	340	noted	565	try
116	do	341	nothing	566	trying
117	does	342	now	567	ts

118	doesnt	343	nowhere	568	twice
119	doing	344	o	569	two
120	done	345	obtain	570	u
121	dont	346	obtained	571	un
122	down	347	obviously	572	under
123	downwards	348	of	573	unfortunately
124	due	349	off	574	unless
125	during	350	often	575	unlike
126	e	351	oh	576	unlikely
127	each	352	ok	577	until
128	ed	353	okay	578	unto
129	edu	354	old	579	up
130	effect	355	omitted	580	upon
131	eg	356	on	581	ups
132	eight	357	once	582	us
133	eighty	358	one	583	use
134	either	359	ones	584	used
135	else	360	only	585	useful
136	elsewhere	361	onto	586	usefully
137	end	362	or	587	usefulness
138	ending	363	ord	588	uses
139	enough	364	other	589	using
140	especially	365	others	590	usually
141	et	366	otherwise	591	v
142	et-al	367	ought	592	value
143	etc	368	our	593	various
144	even	369	ours	594	ve
145	ever	370	ourselves	595	very
146	every	371	out	596	via
147	everybody	372	outside	597	viz
148	everyone	373	over	598	vol
149	everything	374	overall	599	vols
150	everywhere	375	owing	600	vs
151	ex	376	own	601	w
152	except	377	p	602	want
153	f	378	page	603	wants
154	far	379	pages	604	was
155	few	380	part	605	wasnt
156	ff	381	particular	606	way
157	fifth	382	particularly	607	we
158	first	383	past	608	wed
159	five	384	per	609	welcome

160	fix	385	perhaps	610	well
161	followed	386	placed	611	went
162	following	387	please	612	were
163	follows	388	plus	613	werent
164	for	389	poorly	614	weve
165	former	390	possible	615	what
166	formerly	391	possibly	616	whatever
167	forth	392	potentially	617	whatll
168	found	393	pp	618	whats
169	four	394	predominantly	619	when
170	from	395	present	620	whence
171	further	396	previously	621	whenever
172	furthermore	397	primarily	622	where
173	g	398	probably	623	whereafter
174	gave	399	promptly	624	whereas
175	get	400	proud	625	whereby
176	gets	401	provides	626	wherein
177	getting	402	put	627	wheres
178	give	403	q	628	whereupon
179	given	404	que	629	wherever
180	gives	405	quickly	630	whether
181	giving	406	quite	631	which
182	go	407	qv	632	while
183	goes	408	r	633	whim
184	gone	409	ran	634	whither
185	got	410	rather	635	who
186	gotten	411	rd	636	whod
187	h	412	re	637	whoever
188	had	413	readily	638	whole
189	happens	414	really	639	wholl
190	hardly	415	recent	640	whom
191	has	416	recently	641	whomever
192	hasnt	417	ref	642	whos
193	have	418	refs	643	whose
194	havent	419	regarding	644	why
195	having	420	regardless	645	widely
196	he	421	regards	646	will
197	hed	422	related	647	willing
198	hence	423	relatively	648	wish
199	her	424	research	649	with
200	here	425	respectively	650	within

201	hereafter	426	resulted	651	without
202	hereby	427	resulting	652	wont
203	herein	428	results	653	words
204	heres	429	right	654	world
205	hereupon	430	run	655	would
206	hers	431	s	656	wouldnt
207	herself	432	said	657	www
208	hes	433	same	658	x
209	hi	434	saw	659	y
210	hid	435	say	660	yes
211	him	436	saying	661	yet
212	himself	437	says	662	you
213	his	438	sec	663	youd
214	hither	439	section	664	youll
215	home	440	see	665	your
216	how	441	seeing	666	youre
217	howbeit	442	seem	667	yours
218	however	443	seemed	668	yourself
219	hundred	444	seeming	669	yourselves
220	i	445	seems	670	youve
221	id	446	seen	671	z
222	ie	447	self	672	zero
223	if	448	selves		
224	ill	449	sent		
225	im	450	seven		

Lampiran 2 : Data *training* dan data *testing* untuk contoh perhitungan manual

Data *Training*

Dokumen ke-1

Isi dokumen :

Agora

Adventure | Drama | History

Alexandria, 391 AD: Hypatia teaches astronomy, mathematics, and philosophy. Her student Orestes is in love with her, as is Davus, her personal slave. As the city's Christians, led by Ammonius and Cyril, gain political power, the institutions of learning may crumble along with the governance of slavery. Jump ahead 20 years: Orestes, the city's prefect, has an uneasy peace with Christians, led by Cyril. A group from the newly empowered Christians has now taken to enforce their cultural hegemony zealously; first they see the Jews as their obstacle, then nonbelievers. Hypatia has no interest in faith; she's concerned about the movement of celestial bodies and "the brotherhood of all". Although her former slave doesn't see it that way.

Dokumen ke-2

Isi dokumen :

Crank

Action | Crime

Chev Chelios is a professional assassin working for the West Coast crime syndicate. Chev's girlfriend Eve doesn't know what Chev does and Chev is planning to quit the crime syndicate so he can spend more time with her. But for Chev, things about to get very bad, when he learns he has been injected with a poison called "The Beijing Cocktail" by his rival Verona, which will kill him if his heart rate drops. Trying to stay alive and seeking help from friend, Kaylo and Doc Miles, to keep his heart pumping. Chev sets out to find answers as well as protecting Eve, and get his revenge on those who have betrayed him before the poison kills him.

Dokumen ke-3

Isi dokumen :

Fort Apache

Western

In John Ford's sombre exploration mythologising of American heroes, he slowly reveals the character of Owen Thursday, who sees his new posting to the desolate Fort Apache as a chance to claim the military honour which he believes is rightfully his. Arrogant, obsessed with military form and ultimately self-destructive, Thursday attempts to destroy the Apache chief Cochise after luring him across the border from Mexico, against the advice of his subordinates.

Dokumen ke-4

Isi dokumen :

Harry Potter and The Goblet of Fire

Fantasy | Adventure

Harry Potter and the Goblet of Fire takes us deeper into the characters' minds and the darkness of the Wizarding World. At the Quidditch World Cup, Voldemort's followers gather and wreak havoc. Then, at Hogwarts, a legendary event takes place. The Triwizard Tournament! The Goblet of Fire judges who gets in and who doesn't. On the fateful night, three champions are selected. But then the Goblet spits out one other. Harry's. These two major events point to the return of Lord Voldemort. Dumbledore and the other teachers sense it, but it is inevitable. And Harry is no longer safe at Hogwarts. This fourth installment is the most dramatic, and also the scariest. Let me just say that all does not necessarily end well...

Dokumen ke-5

Isi dokumen :

Life of Pi

Adventure | Drama | Fantasy

An aspiring Canadian author interviews the Indian storyteller Pi Patel to hear the firsthand account of his adventures. Pi recounts his upbringing in French-occupied India, where his father owned a zoo. When Pi's family business fails, they embark

on a sea voyage to Canada to begin a new life. One night aboard their Japanese cargo ship in the middle of the ocean, a deadly storm hits and sinks nearly all that Pi holds dear. He survives in a lifeboat with several of their zoo animals, including a fearsome Bengal tiger. In a struggle to survive, Pi and the tiger forge an unexpected connection that gives him daily motivation to live.

Dokumen ke-6

Isi dokumen :

Paths of Glory

Drama | War

In "Paths of Glory" war is viewed in terms of power. This film about a true episode in World War I combines the idea that class differences are more important than national differences with the cannon-fodder theory of war, the theory that soldiers are merely pawns in the hands of generals who play at war is if it were a game of chess.

Dokumen ke-7

Isi dokumen :

Silent Hill

Horror

After the continuous sleep walking episodes of Sharon, the young daughter of Rose Da Silva, the decision is made to take Sharon to the place only mentioned in her restless dreams- Silent Hill. However, the road to Silent Hill is anything but easy to access, and Rose creates a high speed chase between herself and a police officer only to end in a crash for them both. When she wakes up, Sharon has disappeared and Rose is at the entrance to the deserted, dream-like town of Silent Hill. As Rose begins the search for her daughter, she does not realize the terror and mystery surrounding her. Rose is led on a blind search for her beloved daughter, finding herself getting more and more entwined into disturbing past of Silent Hill.

Dokumen ke-8

Isi dokumen :

The Hangover

Comedy

Just two days before his marriage with Tracy Garner, Doug Billings, in the company of two friends: Phil Wenneck and Stu Price; and Tracy's eccentric brother, Alan, head out to party in Vegas. Driving his father's Mercedez, they rent a pricey villa at Caesar's and head for the rooftop to have a good time. Three of them later wake up with a hangover, unable to re-collect what exactly happened. With the villa in a wreck, they find that they have a baby in the closet; a grown tiger in the bathroom; Stu has a missing tooth and a hooker for a bride; and Doug is missing. Hilarious chaos results as the trio head out to re-trace their steps as well as try to locate Doug and bring him home in one piece before the wedding.

Dokumen ke-9

Isi dokumen :

The Phantom of Opera

Drama | Musical

The story of a young chorus girl, Christine - a young talented singer who, with the right training, could become world famous. While rehearsing at the Opera Populaire, where weird and unexplainable things happen, she captures the attention and the heart of The Phantom, or as the Opera Populaire call him...The Opera Ghost. But he is no ghost - he is a disfigured musical genius who has hidden away for years to avoid the cruel stares of strangers. With the Phantoms help, Christine becomes the venue's leading lady, but tragedy awaits as the young soprano has fallen for the charms of handsome noble Viscount Raoul De Chagny, not realizing her Angel of Music is deeply in love with her. Insane with jealousy and unable to see the object of his affection, and ultimately is obsession, in the arms of another man, The Phantom kidnaps Christine - unaware of the lengths Raoul is prepared to go to get her back.

Dokumen ke-10

Isi dokumen :

Transformers

Action | Sci-Fi | Adventure

High-school student Sam Witwicky buys his first car, who is actually the Autobot Bumblebee. Bumblebee defends Sam and his girlfriend Mikaela Banes from the Decepticon Barricade, before the other Autobots arrive on Earth. They are searching for the Allspark, and the war on Earth heats up as the Decepticons attack a United States military base in Qatar. Sam and Mikaela are taken by the top-secret agency Sector 7 to help stop the Decepticons, but when they learn the agency also intends to destroy the Autobots, they formulate their own plan to save the world.

Data *Testing*

Dokumen ke-1

Isi Dokumen :

Transformers: Dark of The Moon

Action | Adventure | Sci-Fi

Autobots Bumblebee, Ratchet, Ironhide, Mirage (aka Dino), Wheeljack (aka Que) and Sideswipe led by Optimus Prime, are back in action taking on the evil Decepticons, who are eager to avenge their recent defeat. The Autobots and Decepticons become involved in a perilous space race between the United States and Russia, to reach a hidden Cybertronian spacecraft on the moon and learn its secrets, and once again Sam Witwicky has to come to the aid of his robot friends. The new villain Shockwave is on the scene while the Autobots and Decepticons continue to battle it out on Earth.