# PENGGUNAAN METODE PENGELOMPOKAN *K-MEANS* PADA KLASIFIKASI KNN UNTUK PENENTUAN JENIS KANKER BERDASARKAN SUSUNAN PROTEIN

**SKRIPSI**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer


Disusun oleh:

**RIA KURNIANTI**
**NIM 0810963065**


**PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER**
**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**
**UNIVERSITAS BRAWIJAYA**
**MALANG**
**2013**

**LEMBAR PERSETUJUAN**

**PENGGUNAAN METODE PENGELOMPOKAN *K-MEANS* PADA KLASIFIKASI KNN UNTUK PENENTUAN JENIS KANKER BERDASARKAN SUSUNAN PROTEIN**

**SKRIPSI**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

**RIA KURNIANTI**
**NIM 0810963065**

Telah diperiksa dan disetujui oleh :

| Dosen Pembimbing 1 | Dosen Pembimbing 2 |
|---|---|
| **Drs. Marji, MT.** | **Widodo., S.Si, M.Si, Ph.D.Med, Sc,.** |
| **NIP. 19670801 199203 1 001** | **NIP. 19730811 200003 1 002** |

**LEMBARPENGESAHAN**

**PENGGUNAAN METODE PENGELOMPOKAN *K-MEANS* PADA KLASIFIKASI KNN UNTUK PENENTUAN JENIS KANKER BERDASARKAN SUSUNAN PROTEIN**

**SKRIPSI**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh :

**RIA KURNIANTI**
**NIM 0810963065**

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 10 Januari 2013

| Penguji I | Penguji II |
|---|---|
| **Lailil Muflikhah, S.Kom., M.Sc.** | **Rekyan Regasari MP., S.T., M.T.** |
| **NIP. 19741113 200501 2 001** | **NIK. 770414 061 2 0253** |

Penguji III

**Ahmad Afif Supianto, S.Si., M.Kom.**

Mengetahui
Ketua Prodi Teknik Informatika

**Drs. Marji, S.Kom.**
**NIP. 19670801 199203 1 001**

**PERNYATAAN**
**ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Januari 2013

Penulis,

**Ria Kurnianti**
**NIM 0810963065**

# ABSTRAK

**Ria Kurnianti. 2013,** Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, *Penggunaan Metode Pengelompokan K-Means Pada Klasifikasi KNN Untuk Penentuan Jenis Kanker Berdasarkan Susunan Protein,* **Dosen Pembimbing : Drs. Marji, M.T. dan Widodo, S,Si., M.Si.,Ph.D. Med, Sc,.**

Kanker merupakan penyebab kematian kedua di dunia. Penyakit ini sulit disembuhkan jika penyebarannya sudah meluas, tetapi dengan pendeteksian dini dapat mengurangi resikonya. Deteksi yang sudah ada selama ini menggunakan biopsy, pengecekan imun maupun CT Scan. Metode pendeteksian tersebut hanya bisa mendeteksi kanker yang sudah meluas. Deteksi dini dapat dilakukan dengan melakukan klasifikasi pada sekuen protein. Metode KNN merupakan salah satu metode klasifikasi yang merupakan bagian dari data mining dimana objek dikelaskan berdasarkan kemunculan kelas terbanyak pada data latih. Disamping biaya komputasi tinggi dan bergantung pada nilai k nya, metode KNN ini tergolong mudah dan efektif digunakan pada data yang besar. Hasil klasifikasi tersebut bergantung pada data latih yang digunakan. Metode KNN dapat dioptimalkan dengan cara mengelompokan data latih sebelum dilakukan proses klasifikasi. Untuk metode pengelompokan yang mampu menangani data dalam jumlah besar dengan waktu komputasi yang cukup cepat dan relative efisien, dapat digunakan metode *K-Means*. Implementasi penggunaan metode ini dapat memberikan hasil error minimum yang cukup baik, yaitu sebesar 20.00% dan mendapatkan kestabilan nilai error pada saat data sudah terkelompok secara optimal. Data yang digunakan sebanyak 847 data latih dan 135 data uji.

**Kata Kunci** : *K-Means*, KNN, *Data Mining*, Bioinformatika, Kanker, Protein.

# ABSTRACT

**Ria Kurnianti. 2013,** *Informatics Engineering Program Study, Information Technology and Computer Science Program Brawijaya University, Application of K-Means Clustering Method on KNN Classification Method to Determine Cancer Type Based on Their Protein Sequences,* **Advisor : Drs. Marji, M.T. and Widodo, S,Si., M.Si.,Ph.D. Med, Sc,.**

*Cancer is the world second dead cause. This disease is hard to cure once it is spread, but with an earlier detection can reduce the cause. The exiting detection so far using biopsy, immune checking or CT Scan. They could only detect cancer that has spread. Early detection could be done by clasificating protein sequence. KNN(K-Nearest Neighbour) method is a classification method which is a part of data mining where the object classified by the most result that occurred in training dataset. Regardless from the computation high cost and the dependence to the K value, this method is quite easy and effective to be use in a large data. Those classification results depend on the training dataset. The KNN method could be optimized by clustering the training dataset before the classification process. Clustering method that capable to handle a large quantity of data with a quite fast computation and relatively efficient, K-Means method can be used. Implementation of using this method could give the quite good minimum error result; it is about 20.00% and achieving stability error where data has been clustered in optimal condition. This research using 847 training dataset and 135 testing data set.*

***Keywords**: K-Means, KNN, Data Mining, Bioinformatics, Cancer, Protein Sequence.*

# PENGANTAR

Puji syukur kehadirat Tuhan yang Maha Esa, karena hanya dengan kasih dan karunia Nya, penulis dapat menyelesaikan skripsi yang berjudul "Penggunaan Metode Pengelompokan *K-Means* Pada Klasifikasi KNN Untuk Penentuan Jenis Kanker Berdasarkan Susunan Protein". Skripsi ini merupakan salah satu syarat untuk memenuhi persyaratan akademis untuk menyelesaikan studi di program Sarjana Ilmu Komputer Universitas Brawijaya.

Selama mengerjakan skripsi ini, penulis mendapat bantuan dan dukungan dari banyak pihak. Untuk itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar – besarnya kepada

1. Drs. Marji, MT selaku Dosen Pembimbing I yang telah dengan bijaksana dan sabar dalam membimbing penulis dalam menyelesaikan skripsi ini sekaligus sebagai Ketua Program Studi Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
2. Widodo.,S.Si, M.Si, Ph.D.Med,Sc,. selaku Dosen Pembimbing II yang telah dengan bijaksana serta sabar dalam membimbing penulis menyelesaikan skripsi ini dan sering memberikan nasehat moril kepada penulis.
3. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Dani Primanita Kartikasari,S.T selaku dosen pembimbing akademik yang telah memberikan nasehat serta bimbingan selama penulis menyelesaikan masa studi di bangku perkuliahan ini.
5. Segenap Bapak Ibu Dosen yang telah mengajarkan semua ilmu yang dimiliki serta memberikan nasehat dan masukan kepada penulis selama menempuh studinya.
6. Secara khusus penulis ingin mengucapkan terima kasih kepada Mami dan Alm. Papi serta Kakak tercinta yang telah mengajarkan kehidupan dan selalu memberikan dukungan materi maupun moril yang tak henti – hentinya kepada penulis.

7. Dian Cahyono dan Abang Suluh Husodo yang telah dengan rela meluangkan waktu serta tenaga dan pikirannya dalam membantu penulis menyelesaikan skripsi ini.
8. Eric Aditya Prabowo yang telah memberi dukungan dan pengalaman berharga selama ini.
9. Keluarga kecil penulis selama di Malang : Inass, Indah, Astrid, Trisna, Chichi, Mirza, Echi, Alfath, Hari, Noka, Mutya, Rilla, Dito, Toyek, Miko, Mas Pur yang telah menemani dan mengisi hari – hari penulis dan selalu ada ketika penulis membutuhkan.
10. Keluarga besar Gostreem Streetball Community Malang serta BCA Basketball Club dan MIPA Basketball Team, warga KKN Desa Slumbung 2012 yang telah memberikan banyak pengalaman berharga kepada penulis selama ini.
11. Teman – teman ilkom yang sudah memberikan dukungan kepada penulis dan teman – teman lain yang tidak bisa penulis sebutkan satu persatu.

Semoga segala pertolongan dan bantuan maupun dukungan semuanya mendapatkan berkah melimpah dari Tuhan YME. Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, maka saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Penulis dapat dihubungi melalui email mg.ria.kurnianti@gmail.com atau mg.ria.kurnianti@ilkomers.org .

Malang, 10 Januari 2013

Penulis

## DAFTAR ISI

## DAFTAR GAMBAR

# DAFTAR TABEL

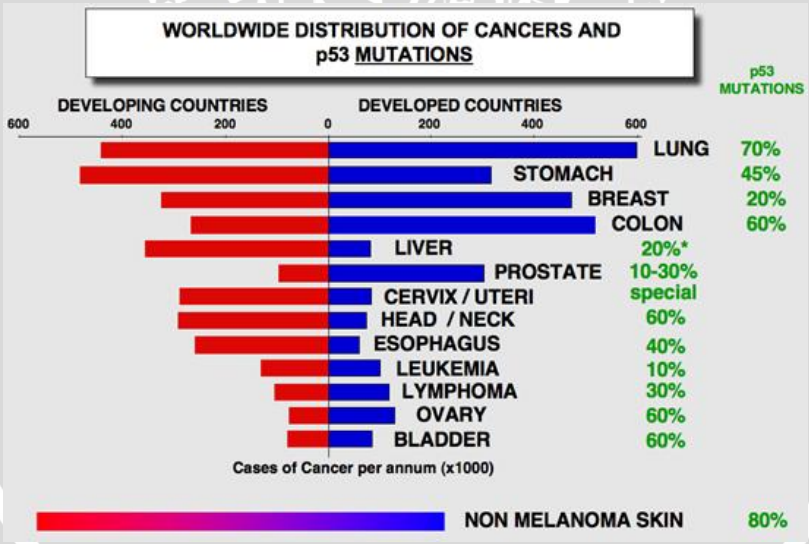
## DAFTAR SOURCE CODE

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Kanker merupakan penyebab kematian kedua di dunia setelah penyakit jantung dan pembuluh (Tjay & Raharja, 1999). Kanker terjadi karena adanya mutasi pada gen, dimana sel – sel yang ada akan membelah secara abnormal. Penyakit ini termasuk dalam penyakit ganas yang sulit disembuhkan, jika penyebarannya sudah meluas. Namun dengan pendeteksian dini, resiko tersebut dapat dikurangi. Deteksi yang digunakan untuk penyakit kanker ini antara lain melalui *biopsy*, pengecekan imun, hingga CT Scan. Metode deteksi – deteksi tersebut hanya bisa mendeteksi kanker yang sudah menyebar (stadium lanjut).

Berdasarkan data yang dimiliki situs http://p53.free.fr, Kanker Kolon *(Colorectal Cancer),* Kanker Payudara (*Breast Cancer*), dan Kanker Paru – Paru (*Lung Cancer*) merupakan jenis kanker dengan pengidap terbanyak saat ini.




Gambar 1.1 Distribusi Penyakit Kanker

Pemicu kanker yang berbeda –beda menyebabkan jenis kanker yang berbeda – beda pula. Kanker umumnya disebabkan karena adanya mutasi gen. Salah satunya adalah p53. Mutasi pada gen tersebut telah banyak teridentifikasi menyebabkan berbagai jenis kanker (Gambar 1.1). Mutasi tersebut mengakibatkan perbedaan urutan asam amino protein p53.

Protein terdiri dari kombinasi 20 asam amino yang disintesis oleh ribosom berdasarkan kode genetik yang terbentuk dari DNA. Jika DNA ini mengalami mutasi, maka susunan protein pun akan menjadi salah. Hal ini dapat menyebabkan berbagai penyakit maupun kelainan seperti penyakit kanker. Oleh karena itu, pendeteksian dini dapat dilakukan melalui tes darah untuk menganalisa sekuen protein yang dimilikinya.

Bioinformatika menjadi isu hangat belakangan ini dan keberadaannya terus dikembangkan. Disiplin ilmu ini tidak lepas dari penggunaan data yang berkaitan dengan kedokteran maupun biologi yang diolah dengan sistem komputasi. Data – data ini dapat digunakan, contohnya untuk mendeteksi suatu penyakit. Data yang digunakan pada bidang ini berkisar antara data DNA maupun data protein.

Data yang digunakan pada bioniformatika tersebut sering kali berjumlah besar dan jenisnya bisa bermacam – macam. Dengan data yang banyak tersebut, proses pencarian informasi pun menjadi lebih lama dan kadang kurang akurat. Oleh karena itu, diperlukan data mining untuk menanggulangi masalah tersebut.

Dengan bantuan data mining, jenis – jenis kanker dapat dikelaskan bahkan dapat digunakan sebagai pendeteksian dini terhadap suatu jenis kanker. Fungsi pada data mining tersebut dinamakan fungsi klasifikasi. Fungsi ini berfungsi untuk mengelompokan data – data ke dalam kategori – kategori yang sudah ditentukan sebelumnya. Ada beberapa metode yang digunakan pada klasifikasi seperti KNN(*K Nearest Neighbour*). Kelebihan dari metode ini sangat mudah digunakan dan efektif digunakan pada data yang besar, tetapi sangat boros dalam biaya komputasi dan sangat bergantung pada nilai k (Hosein Alizadeh et al, 2009). Hasil Klasifikasi metode KNN tersebut bergantung pada data latih yang digunakan sehingga nilai k tidak mengalami kerancuan.

Metode KNN dapat dioptimalkan dengan cara mengelompokan data latih sebelum dilakukan proses klasifikasi. Dalam data mining, fungsi ini disebut *clustering* (pengelompokan). Tugas *Clustering* (pengelompokan) yaitu mengelompokan sejumlah data atau objek ke dalam *cluster* sehingga setiap dalam *cluster* tersebut akan berisi data yang semirip mungkin dan berbeda dengan objek dalam *cluster* lainnya(Tahta Alfina et al, 2012). Metode *K-Means* merupakan metode pengelompokan yang paling umum(Budi Santosa, 2007). Hal ini

dikarenakan *K-Means* mempunyai kemampuan mengelompokan data dalam jumlah yang cukup besar dengan waktu komputasi yang relatif cepat dan efisien(Kohei Arai dan Ali Ridho Barakbah, 2007).

Penelitian sebelumnya yang telah dilalukan oleh Gayathri dan Marimuthu, (2011) terhadap penggunaan metode klasifikasi KNN berdasarkan pengelompokan *K-Means* diterapkan pada klasifikasi teks. Berdasarkan latar belakang yang sudah dipaparkan diatas, maka judul untuk skripsi ini adalah "**PENGGUNAAN METODE PENGELOMPOKAN *K-MEANS* PADA KLASIFIKASI KNN UNTUK PENENTUAN JENIS KANKER BERDASARKAN SUSUNAN PROTEIN**"

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang diangkat pada penelitian ini adalah:

1. Bagaimana membangun suatu model klasifikasi menggunakan KNN yang didasari metode pengelompokan *K-Means*?
2. Bagaimana menentukan banyaknya pengelompokan sehingga setiap data tepat berada dalam satu pengelompokan?
3. Bagaimana pengaruh variasi data terhadap keoptimalan kelompok?
4. Bagaimana penerapan metode tersebut pada penentuan jenis kanker (*Colorectal Cancer*, *Breast Cancer*, *Lung Cancer*)?

## 1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah :

1. Penulis hanya menganalisa sekuen protein penunjuk kanker.
2. Jenis kanker hanya dibatasi pada *Colorectal Cancer*, *Breast Cancer,* dan *Lung Cancer*.
3. Mutasi yang digunakan hanyalah mutasi titik, dimana menyangkut subtitusi asam amino saja.
4. Metode pengelompokan yang digunakan hanya metode *K-Means*.
5. Iterasi dibatasi hingga 100 iterasi saja.
6. Metode klasifikasi yang digunakan metode KNN.

7. Peneliti tidak meneliti pengaruh banyaknya data terhadap tingkat akurasi.
8. Data protein yang digunakan memiliki panjang yang sama.
9. Data latih yang digunakan pada proses klasifikasi merupakan data latih yang sudah dikelompokan pada proses pengelompokan.

## 1.4 Tujuan

Tujuan dari penelitian ini adalah:

1. Membangun suatu model klasifikasi menggunakan KNN berdasarkan metode pengelompokan *K-Means*.
2. Dapat menentukan banyak pengelompokan dimana setiap data berada tepat pada satu kelompok.
3. Dapat mengetahui pengaruh variasi data terhadap keoptimalan kelompok.
4. Dapat menentukan jenis kanker yang diteliti berdasarkan data urutan asam amino protein.

## 1.5 Manfaat

Manfaat yang didapat dari penulisan skripsi ini adalah:

1 Didapatkannya sebuah model klasifikasi yang dapat memberikan hasil klasifikasi yang lebih akurat.

2 Memudahkan dalam menganalisa jenis kanker yang didasarkan pada data protein.

## 1.6. Sistematika Penulisan

Untuk memberikan gambaran tentang skripsi ini, berikut disajikan secara garis besar pembahasan dari keseluruhan isi laporan skripsi untuk setiap bab, sebagai berikut:

**BAB I Pendahuluan**

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan.

**BAB II Dasar teori**

Menguraikan teori tentang protein dan mutasi, kanker, bioinformatika, data mining serta teori-teori yang berhubungan

dengan penggunaan metode klasifikasi yang berdasarkan pengelompokan anggota kelas yang optimal untuk menentukan jenis kanker.

**BAB III Metode dan Perancangan**

Berisi metode-metode yang digunakan dalam membuat sistem dimana data dapat terklasifikasi berdasarkan pengelompokan anggota kelas yang optimal untuk menentukan jenis kanker.

**BAB IV Hasil dan Pembahasan**

Berisi tentang penjelasan implementasi sistem dan hasil pengujian yang dilakukan dari penerapan metode klasifikasi yang berdasarkan pengelompokan anggota kelas yang optimal untuk menentukan jenis kanker.

**BAB V Penutup**

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran untuk pengembangan lebih lanjut.

# BAB II
# TINJAUAN PUSTAKA

## 2.1. Protein

### 2.1.1 Struktur Protein

Protein memiliki 4 tingkatan level yang berbeda berdasarkan strukturnya, yaitu struktur primer, struktur sekunder, struktur tersier, dan struktur kuaterner (Wang, Zaki, Toivonen, & Shasha, 2006). Protein mengurus regulasi fungsi sel dan juga membawa banyak tugas yang berhubungan dengan kehidupan. Protein merupakan molekul kompleks yang berbentuk blok rantai sederhana yang disebut asam amino (Chen & Lonardi, 2010). Keempat struktur protein dijelaskan sebagai berikut:

1. Struktur Primer

   20 jenis asam amino yang berbeda saling berikatan pada ikatan peptide yang merupakan struktur primer unik protein. Sebuah sekuen protein unik untuk protein tertentu berdasar struktur dan fungsinya. Struktur primer tidak dapat menentukan fungsi protein secara langsung, tetapi secara tidak langsung bertanggung jawab pada pertambahan level strukturnya.

2. Struktur Sekunder

   Struktur ini merupakan struktur yang tersusun rapi berdasarkan interaksi antara ikatan kimia dari asam amino. Karena itu ada beberapa struktur primer yang memungkinkan, beberapa pola karakteristik yang sering muncul yaitu *α helix*, *βsheet* (dikenal juga sebagai *β pleated sheet*) dan *β-turn*.

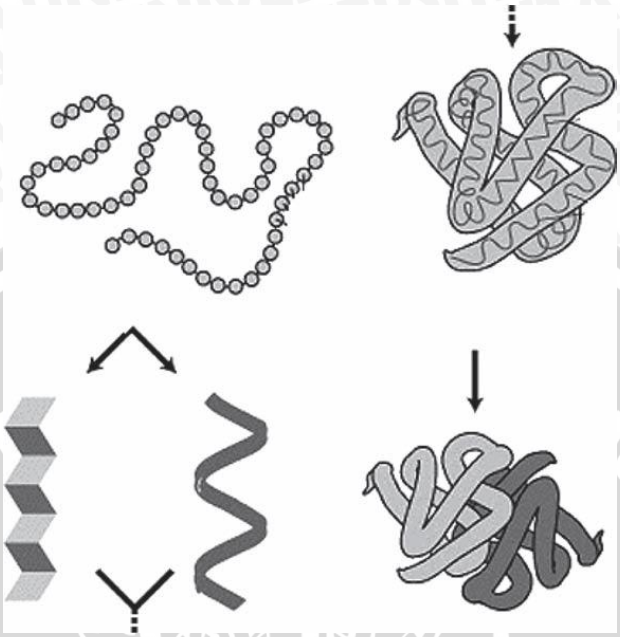
3. Struktur Tersier

   Dikenal juga sebagai “lipatan” protein, merupakan struktur tiga dimensi dan menyebabkan struktur sekunder melipat pada dirinya sendiri. Fungsi dari protein ditentukan oleh struktur tiga dimensi ini.

4. Struktur kuaterner

   Struktur final dari protein dihasilkan oleh interaksi antara beberapa molekul protein. Struktur kuaterner ini merupakan unit aktif yang

dihasilkan dari kumpulan rantai polipeptida. Tidak semua protein memiliki struktur ini.



Gambar 2.1 Struktur Protein
(Chen & Lonardi, 2010)

### 2.1.2. Hubungan Protein dan Kode Genetik

Huruf A, G, T, C menyatakan nukleotida – nukleotida yang terdapat dalam DNA. Huruf – huruf ini tersusun membentuk kode tiga huruf yang disebut dengan kodon. Kumpulan kodon ini akan membentuk kode genetik (Murray, Granner, & Rodwell, 2006).

Kode gen untuk berbagai produk yang digunakan oleh sel untuk membentuk jaringan organisme. Produk tersebut disebut dengan protein yang memiliki tiga fungsi, yaitu membawa sinyal, mengangkut molekul seperti oksigen dan dapat mengatur proses sel, seperti mekanisme pertahanan (Keedwell & Narayanan, 2005).

Pengetahuan tentang sekuen DNA memungkinkan struktur primer polipeptida dapat disimpulkan. Penentuan sekuen DNA hanya membutuhkan sejumlah kecil DNA dan mudah menghasilkan ratusan sekuen nukleotida. Penentuan sekuen DNA mengungkapkan urutan bagaimana asam – asam amino ditambahkan ke rantai polipeptida yang baru sewaktu disintesis di ribosom (Murray, Granner, & Rodwell, 2006).

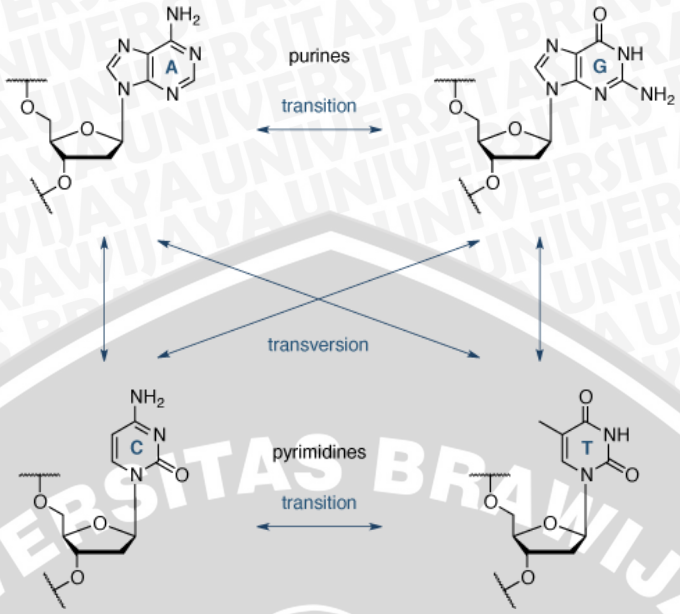
Proses dimana gen dibuat menjadi protein diawali dengan RNA polymerase bersentuhan dengan kromosom dan mengidentifikasi titik awal sebuah gen. Molekul – molekul ini membuka struktur heliks ganda untuk mengekspose untai DNA yang membentuk gen, dan saling melengkapi salinan gen untuk dibaca. Proses menyalin gen ke dalam mRNA disebut Transkripsi, dan proses mengkonversi mRNA menjadi protein disebut Translasi (Keedwell & Narayanan, 2005)

| | T | | | C | | | A | | | G | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | TTT | Phe | F | TCT | Ser | S | TAT | Tyr | Y | TGT | Cys | C |
| | TTC | Phe | F | TCC | Ser | S | TAC | Tyr | Y | TGC | Cys | C |
| | TTA | Leu | L | TCA | Ser | S | TAA | stop | * | TGA | stop | * |
| | TTG | Leu | L | TCG | Ser | S | TAG | stop | * | TGG | Trp | W |
| C | CTT | Leu | L | CCT | Pro | P | CAT | His | H | CGT | Arg | R |
| | CTC | Leu | L | CCC | Pro | P | CAC | His | H | CGC | Arg | R |
| | CTA | Leu | L | CCA | Pro | P | CAA | Gln | Q | CGA | Arg | R |
| | CTG | Leu | L | CCG | Pro | P | CAG | Gln | Q | CGG | Arg | R |
| A | ATT | Ile | I | ACT | Thr | T | AAT | Asn | N | AGT | Ser | S |
| | ATC | Ile | I | ACC | Thr | T | AAC | Asn | N | AGC | Ser | S |
| | ATA | Ile | I | ACA | Thr | T | AAA | Lys | K | AGA | Arg | R |
| | ATG | Met | M | ACG | Thr | T | AAG | Lys | K | AGG | Arg | R |
| G | GTT | Val | V | GCT | Ala | A | GAT | Asp | D | GGT | Gly | G |
| | GTC | Val | V | GCC | Ala | A | GAC | Asp | D | GGC | Gly | G |
| | GTA | Val | V | GCA | Ala | A | GAA | Glu | E | GGA | Gly | G |
| | GTG | Val | V | GCG | Ala | A | GAG | Glu | E | GGG | Gly | G |

Gambar 2.2 Kode Genetik
(Sumber: **http://www.cs.au.dk/**)

### 2.1.3 Mutasi

Mutasi adalah perubahan permanen dalam urutan DNA dari gen. Mutasi pada urutan DNA gen dapat mengubah urutan asam amino dari protein yang dikode oleh gen (http://learn.genetics.utah.edu/, diakses 15 Mei 2012). Perubahan satu basa (*point mutation*, mutasi titik) dapat berupa transisi atau transversi. Pada transisi, sebuah pirimidin diganti oleh pirimidin lain atau purin diganti oleh purin lain (T ↔ C, A ↔ G). Pada transversi, sebuah purin berubah menjadi salah satu dari dua jenis pirimidin atau sebaliknya.




Gambar 2.3 Diagram Transisi dan Transversi
(Sumber: Brown & Brown, 2005)

Jika sekuen nukleotida gen yang mengandung mutasi ditranskripsikan menjadi sebuah molekul RNA, molekul tersebut akan membentuk basa komplementer sekuens nukleotida gen mutasi tersebut. Jika ditranslasikan menjadi protein, perubahan satu basa molekul mRNA akan menimbulkan salah satu dari beberapa efek berikut (Murray, Granner, & Rodwell, 2006):

a. *Silent Mutation*

Mungkin tidak ada efek yang dapat dideteksi karena adanya *degeneracy* kode, kemungkinan terjadi mutasi ini akan meningkat jika basa yang berubah di molekul mRNA merupakan nukleotida ketiga kodon.

b. Efek *Missense*

Efek ini akan terjadi jika yang dipasang di tempat tersebut adalah asam amino lain. Asam amino yang salah ini –atau *missense*, bergantung pada lokasinya di protein, mungkin *acceptable* (dapat diterima), *partially* (sebagian diterima), atau *unacceptable* (tidak dapat diterima) bagi fungsi molekul protein yang bersangkutan. Berdasarkan penelitian mendalam mengenai kode genetik, dapat disimpulkan

bahwa sebagian besar perubahan satu basa menyebabkan digantikannya sebuah asam aminooleh asam amino lainnya yang gugus fungsionalnya serupa.

c. Kodon *Nonsense*

Hal ini dapat menyebabkan terminasi/penghentian dini (*premature termination*) pembentukan rantai peptide, sehingga hanya berbentuk sebagian/sepotong dari molekul protein yang ingin diproduksi. Besar kemungkinan hal ini menyebabkan molekul protein tidak berfungsi.

## *2.2.* Kanker

Banyak faktor penyebab terjadinya kanker, baik internal maupun eksternal. Faktor internal terutama keberadaan gen - gen yang berperan pada siklus sel telah menjadi pusat perhatian dalam hubungannya dengan proses terjadinya pertumbuhan tumor. Dalam hubungannya dengan pertumbuhan tumor, terdapat dua golongan gen: Pertama adalah kelompok pemicu terjadinya tumor yang lazim disebut tumor oncogenes, seperti: gen c-myc dan gen ras; Kedua adalah kelompok penekan terjadinya tumor yang lazim disebut tumor suppressor gene, seperti: gen p53 dan gen Rb. Hingga saat ini banyak peneliti sementara menyimpulkan bahwa penyebab terjadinya kanker (50%) adalah adanya mutasi pada gen-gen tersebut (Pusztai, Lewis, & Yap, 1996).

Model epigenetik kanker kehilangan daya tariknya sebagian besar karena array dari gen pengendali mutant ditemukan dalam sel tumor pada manusia. Jadi fokus bergeser semakin ke gen, lebih khusus genom kanker sel.

### 2.2.1. Gen TP53

Protein TP53 yang dikode gen p53 berfungsi sebagai faktor transkripsi tetramerik yang ditemukan pada tingkat yang sangat rendah pada sel yang tidak mengalami stress. Setelah terjadi stress, berbagai jalur dilakukan menuju ke arah modifikasi pasca-translasional protein dan stabilisasinya. Akumulasi ini mengaktifkan transkripsi sejumlah besar gen yang terlibat dalam berbagai aktivitas di dalam sel meliputi penghambatan siklus sel dan apoptosis yang bergantung pada konteks selular, besarnya luka, atau parameter lain yang belum

diketahui. Mutasi p53 adalah perubahan genetik yang paling umum ditemukan pada kanker manusia dan fungsi p53 hilang secara tidak langsung baik oleh eksklusi inti (Syaifudin, 2007).

### 2.2.2. Kanker Usus (*Colorectal Cancer*)

*Colon and Rectal Cancer (CRC)* merupakan penyebab terbesar kasus kematian di negara barat dan keberadaannya terus meningkat seiring dengan adopsi gaya hidup barat. Beberapa analisis menyebutkan CRC dibagi menjadi 5 kelas utama, yaitu: *Hereditary nonpolyposis colorectal cancer (HNPCC), suspected HNPCC, juvenile cases, familial tumours*, dan *apparently sporadic cases*. Semua kasus tersebut terkait dengan ekspresi molekul tertentu seperti (Cesario & Frederick, 2011):

1. Status *DNA microsatellite instability (MSI)* yang dikelompokan menjadi: MSI-tinggi (MSI-H), MSI-rendah (MSI-L) dan MS stabil (MSS)
2. *CpG Island methylated phenotype (CIMP)* yang terbagi menjadi: CIMP-tinggi, CIMP-rendah dan CIMP negatif (CIMP-neg).

Menurut korelasi morfologinya, dapat dibagi menjadi lima tipe molekul:

1. Tipe 1 (CIMP-high/MSI-H/BRAF mutasi)
2. Tipe 2 (CIMP-high/MSI-L atau MSS / BRAF mutasi)
3. Tipe 3 (CIMP-low/MSS atau mutasi MSI-L/KRAS)
4. Tipe 4 (CIMP-neg/MSS)
5. Tipe 5 atau *Lynch sindrom* (CIMP-neg/MSI-H)

Tahap molekul – molekul ini dapat dideteksi pada awal evolusinya dan terdapat pada polip dengan lesi prakanker.

### 2.2.3. Kanker Payudara (*Breast Cancer*)

Pada awal 1900-an, penemuan gen kanker payudara ditemukan oleh sebuah kelompok yang dipimpin oleh Mary-Claire King di University of California di Berkeley yang disebut dengan Breast *Cancer Susceptibility Gene 1*(BRCA1), gen pertama yang terkait dengan kanker payudara terdapat di suatu tempat pada kromosom 17. Selanjutnya para ilmuwan meneliti dan menemukan satu gen lagi yang berkaitan dengan kanker payudara yang terkait dengan kanker

ovarium dan payudara wanita maupun pria, gen ini diberi nama BRCA2 (Parthasarathy, 2007)

Menurut McCafferty et al, yang dikutip oleh Alfredo Cesario dan Frederick B. Marcus dalam bukunya yang berjudul *Cancer Systems Biology, Bioinformatics and Medicine: Research and Clinical Applications*, ada empat jenis utama dari kanker payudara. Jenis ini dikenal sebagai: *A Luminal, B Luminal, Basal-like dan HER2/neu*. Semua itu diidentifikasi berdasarkan hormon *Oestrogen Receptor (ER), Progesteron Receptor (PR), HER2/neu* dan *Ki-67 Proliferation Index.*

Meskipun terdapat bukti bahwa sekresi hormone dan metabolism dapat dipengaruhi oleh lingkungan, misalnya melalui diet atau aktivitas fisik, control pola hormon genetic sebagian besar sudah diatur. Telah dihipotesiskan bahwa model multigenik dari predisposisi kanker payudara dapat dikembangkan mencakup polimorfisme dalam gen yang terlibat biosintesis estrogen dan *intracellular binding* (Henderson, Ponder, & Ross, 2003).

### 2.2.4. Kanker Paru-Paru (*Lung Cancer*)

Kanker paru – paru merupakan penyakit mematikan manusia yang belum dapat didiagnosa secara dini. Staging masih didasarkan pada hispatologi dan kriteria klinis yang memiliki keterbatasan untuk memprediksi penyakit akan kambuh dan kelangsungan hidup pasien. Tahun – tahun terakhir ini, dilakukan upaya besar memperkenalkan profil molekul yang melambangkan berbagai macam karsinoma bronkus untuk mendapatkan prediksi yang lebih baik. Contohnya pada mutasi EGFR untuk mengidentifikasi pasien dengan non-sel kanker paru – paru yang dapat merespon baik terhadap inhibitor *EGFR Tyrosine Kinase* (Cesario & Frederick, 2011).

Empat jenis kanker paru – paru utama berdasarkan histologisnya dibagi menjadi: *squamous carcinoma, adenocarcinoma, large-cell carcinoma, small-cell carcinoma*. Selain itu ada pula jenis gabungan seperti *adeno-squamous, neuroendocrine (carcinoids), sarcomatoid,* dan beberapa karsinoma lainnya (WD, Muller, HK, & Harri, 2004). Asal kanker paru – paru tergantung pada sejumlah

interaksi antara lingkungan dan kerentanan genetic, termasuk perubahan dalam deregulasi jalur sinyal.

## 2.3. Bioinformatika

### 2.3.1. Pengertian Bioinformatika

Bioinformatika lahir setelah para ahli biologi menemukan bagaimana urutan DNA dan apa saja yang dihasilkan dari kombinasi keempat basa DNA tersebut. Para ahli bioinformatika menggunakan algoritma, statistik, dan teknik matematika lainnya untuk menguraikan bahasa DNA (Jones & Pevzner, 2004)

Bioinformatika merupakan bagian intergral di hampir semua aspek biologi, termasuk genetika. Hal ini mencakup :

a. Pengetahuan manajemen dan ekspansi
b. Manajemen data dan data mining
c. Studi desain dan dukungan
d. Analisis data
e. Penentuan fungsi

Baik genetik maupun bioinformatika sama - sama terfokus pada mengajukan pertanyaan yang tepat, menghasilkan dan menguji hipotesis dan mengatur dan menafsirkan sejumlah besar data untuk mendeteksi pola biologis (Barnes & Gray, 2003).

### *2.3.2. Protein Sequencing*

Sekuensing pada protein berupa gabungan dari 20 asam amino yang direpresentasikan sebagai string. Untuk gambaran contoh sekuen protein dapat dilihat pada gambar 2.4:


Gambar 2.4 Sekuen Protein

(Sumber: Lengauer, 2007)

### *2.3.3. Subtitution Matrix*

Untuk memberikan nilai antara dua residu protein yang sama atau tidak sama dalam *sequence alignment* digunakan suatu matrik yang disebut *subtitution matrix*. Nilai yang terdapat pada *subtitution matrix* menunjukkan kemungkinan masing-masing karakter residu untuk digantikan karakter residu yang lain berdasarkan hubungan revolusioner. Terdapat dua jenis *subtitution matrix* yaitu PAM dan BLOSUM (Schenkel & Hätinen, 2006).

#### *2.3.3.1 Point Accepted Mutation Matrix (PAM)*

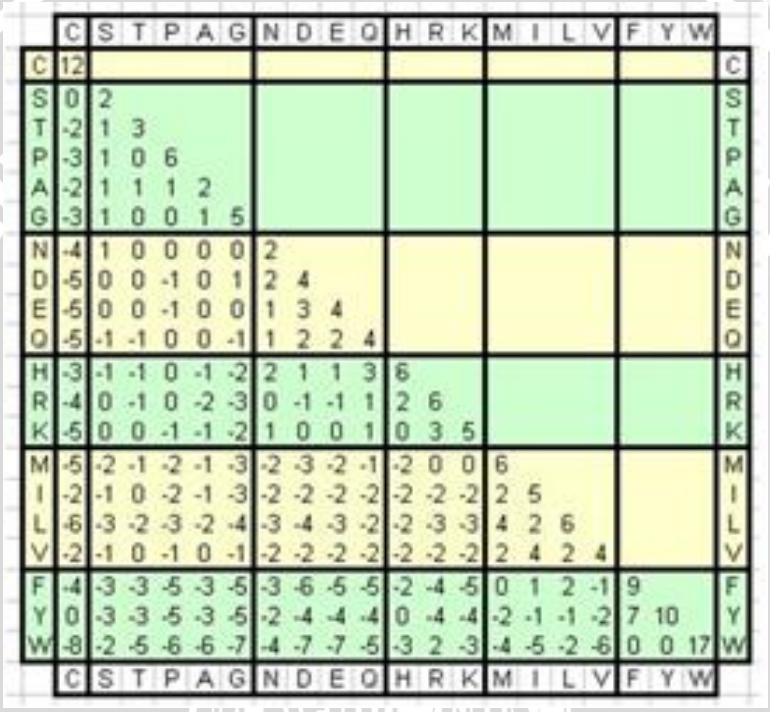
*Point Accepted Mutation Matrix* atau yang lebih dikenal dengan sebutan PAM, merupakan *subtitution matrix* yang pertama kali dipaparkan oleh Margaret Dayhoff pada tahun 1978. Sekumpulan PAM (PAM1 sampai PAM250) berasal dari penurunan *sequence* yang memiliki hubungan kekerabatan yang dekat. Maksudnya adalah PAM2, PAM3 dan seterusnya nilai kemungkinan yang terdapat dalam matrik berasal dari perkalian matrix. PAM2 diperoleh dari perkalian antara PAM1 dan PAM1 sedangkan PAM3 diperoleh dari perkalian antara PAM2 dan PAM1, begitu pula seterusnya (Nilges M, 2002).

*Subtitution matrix* PAM penulisannya diikuti dengan sebuah nomer yang menunjukkan jarak evolusi, misalnya PAM1, PAM250. Pada *subtitution matrix* PAM1, nilai yang mengikutinya berarti perkiraan besarnya nilai subtitusi yang akan diperoleh jika 1% asam amino berubah. Matrik PAM1 digunakan sebagai dasar untuk menghitung matrik lain dengan asumsi bahwa mutasi yang berulang akan mengikuti pola yang sama dengan matrik PAM1 dan banyak subtistusi dapat terjadi pada lokasi yang sama. Dengan logika tersebut dapat diperoleh matrik PAM250.

Kekurangan dari *subtitution matrix* PAM adalah mengasumsikan bahwa semua tipe mutasi pada protein terdistribusi seragam dan PAM menggunakan data dari protein yang memiliki hubungan kekerabatan yang dekat untuk menginferensi protein yang memiliki hubungan yang jauh (Holbert, 2002).

Pada gambar 2.5, alfabet yang terdapat pada tepi gambar merupakan 20 macam alfabet yang mewakili nama 20 macam residu asam amino. Macam-macam residu asam amino tersebut berurutan dari atas ke bawah yaitu Sistein

(diwakili dengan alfabet C), Serin (S), Treonin (T), Prolin (P), Alanin (A), Glisin (G), Asparagin (N), Asam Aspartat (D), Asam Glutamat (E), Glutamin (Q), Histidin (H), Arginin (R), Lisin (K), Metionin (M), Isoleusin (I), Leusin (L), Valin (V), Fenilalanin (F), Tirosin (Y), Tritopfan (W) (Yuwono, 2005). Nilai – nilai yang terdapat dalam gambar adalah nilai subtitusi yang merupakan kemungkinan alfabet residu asam amino yang satu untuk digantikan (disubtitusi) dengan residu asam amino yang lain.




Gambar 2.5 Matriks PAM250
(Sumber: Schenkel & Hätinen, 2006)

### *2.3.3.2 Block Subtitution Matrix (BLOSUM)*

Metodologi yang telah dipaparkan oleh Dayhoff, yang diimplementasikan pada matrik PAM yaitu dengan membandingkan kedekatan hubungan spesies yang dihasilkan tidak bekerja baik untuk *sequence alignment* yang secara evolusi menyebar atau memiliki hubungan kekerabatan yang jauh. Oleh karena itu, muncul *subtitution matrix* jenis lain yaitu *Block Subtitution Matrix* atau yang lebih dikenal dengan sebutan BLOSUM yang mampu mengatasi masalah tersebut. BLOSUM dibuat oleh Steven Henikoff dan Jorja Henikoff pada tahun 1992 dan

dapat digunakan untuk *multiple alignment* dari protein yang secara evolusi memiliki hubungan kekerabatan yang jauh. Sekumpulan BLOSUM (BLOSUM50, BLOSUM62,...) berasal dari basis data BLOCK yang diturunkan dari *alignment* yang memiliki hubungan kekerabatan jauh (jmomand, 2001).

| | C | S | T | P | A | G | N | D | E | Q | H | R | K | M | I | L | V | F | Y | W | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 9 | | | | | | | | | | | | | | | | | | | | C |
| S | -1 | 4 | | | | | | | | | | | | | | | | | | | S |
| T | -1 | 1 | 4 | | | | | | | | | | | | | | | | | | T |
| P | -3 | -1 | 1 | 7 | | | | | | | | | | | | | | | | | P |
| A | 0 | 1 | -1 | -1 | 4 | | | | | | | | | | | | | | | | A |
| G | -3 | 0 | 1 | -2 | 0 | 6 | | | | | | | | | | | | | | | G |
| N | -3 | 1 | 0 | -2 | -2 | 0 | 6 | | | | | | | | | | | | | | N |
| D | -3 | 0 | 1 | -1 | -2 | -1 | 1 | 6 | | | | | | | | | | | | | D |
| E | -4 | 0 | 0 | -1 | -1 | -2 | 0 | 2 | 5 | | | | | | | | | | | | E |
| Q | -3 | 0 | 0 | -1 | -1 | -2 | 0 | 0 | 2 | 5 | | | | | | | | | | | Q |
| H | -3 | -1 | 0 | -2 | -2 | -2 | 1 | 1 | 0 | 0 | 8 | | | | | | | | | | H |
| R | -3 | -1 | -1 | -2 | -1 | -2 | 0 | -2 | 0 | 1 | 0 | 5 | | | | | | | | | R |
| K | -3 | 0 | 0 | -1 | -1 | -2 | 0 | -1 | 1 | 1 | -1 | 2 | 5 | | | | | | | | K |
| M | -1 | -1 | -1 | -2 | -1 | -3 | -2 | -3 | -2 | 0 | -2 | -1 | -1 | 5 | | | | | | | M |
| I | -1 | -2 | -2 | -3 | -1 | -4 | -3 | -3 | -3 | -3 | -3 | -3 | -3 | 1 | 4 | | | | | | I |
| L | -1 | -2 | -2 | -3 | -1 | -4 | -3 | -4 | -3 | -2 | -3 | -2 | -2 | 2 | 2 | 4 | | | | | L |
| V | -1 | -2 | -2 | -2 | 0 | -3 | -3 | -3 | -2 | -2 | -3 | -3 | -2 | 1 | 3 | 1 | 4 | | | | V |
| F | -2 | -2 | -2 | -4 | -2 | -3 | -3 | -3 | -3 | -3 | -1 | -3 | -3 | 0 | 0 | 0 | -1 | 6 | | | F |
| Y | -2 | -2 | -2 | -3 | -2 | -3 | -2 | -3 | -2 | -1 | 2 | -2 | -2 | -1 | -1 | -1 | -1 | 3 | 7 | | Y |
| W | -2 | -3 | -3 | -4 | -3 | -2 | -4 | -4 | -3 | -2 | -2 | -3 | -3 | -1 | -3 | -2 | -3 | 1 | 2 | 11 | W |
| | C | S | T | P | A | G | N | D | E | Q | H | R | K | M | I | L | V | F | Y | W | |

Gambar 2.6 Matriks BLOSUM62

(Sumber: Schenkel & Hätinen, 2006)

## *2.4. Data Mining*

### 2.4.1. Konsep *Data Mining*

Data adalah kenyataan yang menggambarkan suatu kejadian – kejadian dan kesatuan nyata. Kejadian – kejadian adalah sesuatu yang terjadi pada saat tertentu. Kesatuan nyata adalah berupa suatu objek yang nyata seperti tempat, benda dan orang yang betul – betul ada dan terjadi. Dari definisi dan uraian data tersebut dapat disimpulkan bahwa data adalah bahan mentah yang diproses untuk menyajikan informasi (Sutabri, 2005).

Data merupakan asset penting dalam organisasi. Berbagai pendapat yang sering dikemukakan oleh berbagai pakar berkaitan dengan system computer yang kontemporer adalah: perangkat keras memiliki umur yang singkat, perangkat

lunak memilki umur yang lebih panjang, sedangkan data memiliki umur yang paling panjang (Wahyuni, 2004).

Jumlah informasi yang tersedia bagi kita sangat banyak dan nilai data sebagai asset organisasi telah diakui secara luas. Untuk memanfaatkan dataset yang besar dan kompleks, pengguna memerlukan alat yang memudahkan tugas mengatur data dan mengekstraksi informasi yang beguna dalam cara yang baik (Ramakrishna & Gehrke, 2003)

Data mentah jarang bermanfaat secara langsung. Nilai yang sebenarnya didasarkan pada: kemampuan untuk mengekstrak informasi yang berguna untuk membantu pengambilan keputusan atau eksplorasi dan memahami fenomena yang mengatur sumber data (Mitra & Acharya, 2003).

Istilah data mining memiliki beberapa padanan, seperti *knowledge discovery* atau *pattern recognition*. Kedua istilah tersebut sebenarnya memiliki ketepatannya masing – masing. Istilah *knowledge discovery* atau penemuan pengetahuan tepat digunakan karena tujuan utama dari data mining memang untuk mendapatkan pengetahuan yang masih tersembunyi di dalam bongkahan data. Istilah *pattern recognition* atau pengenalan pola pun tepat untuk digunakan karena pengetahuan yang hendak digali memang berbentuk pola – pola yang mungkin juga masih perlu digali dari dalam bongkahan data yang tengah dihadapi (Susanto & Suryadi, 2010).

Data mining adalah analisis dari data set observasional untuk menemukan hubungan tak terduga antar data dan untuk meringkas data dengan cara baru dimana dapat berguna dan dimengerti oleh pemilik data. Hubungan dan ringkasan tersebut didapatkan melalui pelatihan data mining yang disebut model atau pola. Contohnya termasuk persamaan linier, aturan (*rule*), pengelompokan (*cluster*), grafik (*graph*), pohon struktur (*tree structure*), dan pola berulang pada time series (*recurrent pattern in time series*). (Hand, Mannila, & Smyth, 2001).

Dalam prakteknya, ada dua tujuan dari data mining, yaitu prediksi dan deskripsi. Prediksi melibatkan beberapa variable atau field dalam dataset untuk memprediksi nilai yang tidak diketahui atau yang akan dating dari variable lain. Deskripsi, di sisi lain, focus pada menemukan pola tersembunyi pada data yang

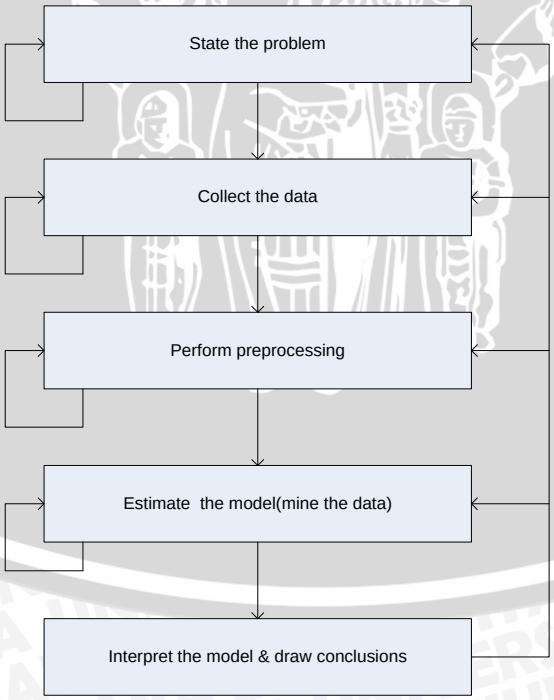
bisa ditafsirkan manusia. Oleh karena itu, data mining dapat dikategorikan menjadi 2, yaitu: (Kantardzic, 2003)

a. Prediksi Data Mining, dimana model prosedur dari system dijelaskan oleh kumpulan dataset.
b. Deskriptif Data Mining, dengan prosedur baru, informasi nontrivial berdasarkan dari dataset yang tersedia.

Menurut Mehmed Kantardzic dalam bukunya yang berjudul Data Mining; Concepts, Models, Methods, and Algorithms, proses pada data mining adalah proses menemukan berbagai macam model, ringkasan, dan nilai – nilai yang diturunkan dari kumpulan data. Proses tersebut terdiri dari:

a. Menyatakan masalah dan merumuskan hipotesa
b. Mengumpulkan data
c. Preprosesing data
d. Estimasi model
e. Interpretasi model dan menarik kesimpulan

Diagram proses data mining dapat digambarkan sebagai berikut:




Gambar 2.7 Proses Data Mining

(Sumber: Kantardzic, 2003)

Tugas yang dilakukan data mining didasarkan pada objektifitas dari orang yang menganalisa datanya, tetapi hal tersebut dapat dirangkum menjadi beberapa jenis menurut David Hand et all dalam bukunya yang berjudul Principles of Data Mining(2001) Tugas - tugas tersebut adalah sebagai berikut :

a. *Eksploratory Data Analysis* (EDA)

Tujuan dari tugas data mining yang pertama ini hanyalah mengeksplorasi data tanpa adanya ide yang jelas tentang apa yang dicari. Biasanya teknik - teknik pada EDA ini lebih bersifat interaktif dan visual. Hanya saja pada beberapa kasus yang lebih kompleks, akan lebih sulit untuk memvisualisasikan dengan teknik EDA.

b. *Descriptive Modeling*

Tujuan dari model descriptif adalah menggambarkan semua data atau proses menghasilkan data. Contoh dari model ini adalah model untuk distribusi probabilitas data, analisa cluster dan segmentasi, model ketergantungan.

c. *Predictive Modeling* (*Classification and Regression*)

Tujuan dari model ini adalah untuk membangun model yang hanya memperbolehkan satu nilai variabel yang akan diprediksi dari nilai - nilai variabel yang ada. Pada klasifikasi, variabel yang akan diprediksi bersifat kategorikal, sedangkan pada regresi bersifat kuantitatif.

d. Menemukan Pola dan aturan (*discovering pattern and rule*)

Selain untuk membangun model, tugas data mining juga untuk menemukan pola dan aturan yang terbentuk. Biasanya algoritma yang digunakan adalah algoritma berdasarkan aturan asosiasi.

e. Pengambilan berdasarkan konten (*retrieval by content*)

Pada tugas data mining yang satu ini, pengguna memiliki pola ketertarikan dan keinginan untuk menemukan pola yang sama pada suatu dataset. Hal ini sering digunakan pada data yang berupa teks atau gambar. Selain pola kesamaan, teknik pencarian pun sangat penting pada pengambilan yang didasarkan pada konten ini.

### *2.4.2. Preprocessing*

Sebagian besar data yang ada di dunia nyata seringkali masih dalam data mentah yang terkadang mengandung :

a. Redudansi data
b. Nilai yang hilang (*missing value*)
c. Adanya outlier
d. Data yang tidak cocok untuk data mining
e. Nilai yang tidak konsisten atau tidak masuk akal

Oleh karena alasan diatas, maka pada data mining diperlukan tahap *preprocessing*. Menurut (Larose, 2005) dalam bukunya Discovering knowledge in data : an introduction to data mining mengatakan bahwa tujuan dari *preprocessing* adalah untuk meminimalkan "sampah" yang ada pada sistem sehingga dapat meminimalisir jumlah "sampah" yang nantinya dihasilkan oleh sistem tersebut.

Dalam buku yang berjudul Data Mining Concepts and Techniques, (Han & Kamber, 2006) menyebutkan jika metode *preprocessing* terbagi menjadi 4, yaitu: pembersihan data, integerasi data, transformasi data, reduksi data.

#### 2.4.2.1 Pembersihan Data *(Data Cleaning)*

Data dalam bentuk standar belum tentu bebas dari kesalahan. Dalam dunia nyata dataset dapat memiliki kesalahan seperti kesalahan pengukuran, penilaian subjektif atau *error* yang disebabkan hal lainnya. Akan sangat sulit melihat kesalahan - kesalahan nilai pada data yang memiliki cariabel dalam jumlah besar sekaligus jumlah data yang besar pula. Karena hal itulah, pembersihan data sangatlah dibutuhkan pada proses tahap preprocessing. Hal ini seperti yang dikatakan oleh (Bramer, 2007) dalam bukunya yang berjudul Principles of Data Mining.

#### 2.4.2.2 Integerasi Data (*Data Integration*)

Data mining terkadang memerlukan integerasi data (penggabungan data dari beberapa data). Integerasi data ini disini dapat menyababkan terjadinya redudansi data. Untuk menghindari hal tersebut pada variabel numerik dapat digunakan koefisien korelasi dimana mengevaluasi korelasi antara dua atribut. Selain itu masalah lain dari integerasi data adalah mendeteksi dan memecahkan

nilai yang konflik. Hal ini dapat dipecahkan dengan heterogenitas semantik dan struktur data yang baik sehingga dapat meningkatkan tingkat akurasi dan membantu proses data mining selanjutnya. (Han & Kamber, 2006)

#### 2.4.2.3 Transformasi Data (*Data Transformation*)

Transformasi data adalah merubah data menjadi bentuk yang sesuai untuk proses data mining. Jiawei Han dan Micheline Kamber menyebutkan dalam bukunya Data Mining Concepts and Techniques (2006) bahwa proses ini dapat meliputi 5 jenis, yaitu:

a. Penghalusan (*smoothing*), tujuannya untuk menghilangkan noise dari data. Teknik - teknik tersebut meliputi binning, regresi dan pengelompokan.
b. Agregasi (*Aggregation*), data digabungkan untuk mendapatkan data lain. Misalkan, data penjualan harian dapat dikumpulkan untuk menghitung jumlah total penjualan bulanan dan tahunan. Langkah ini biasanya digunakan dalam membangun sebuah kubus data untuk analisis data pada granularities ganda.
c. Generalisasi (*Generalization*), data primitif atau data mentah diganti dengan konsep data yang lebih tinggi menggunakan konsep hirarki. Sebagai contoh, atribut kategorikal, seperti jalan, dapat digeneralisasi untuk konsep tingkat yang lebih tinggi, seperti kota atau negara. Demikian pula, nilai-nilai untuk atribut numerik, seperti usia, dapat dipetakan ke konsep tingkat yang lebih tinggi, seperti pemuda, setengah baya, dan senior.
d. Normalisasi (*Normalization*), skala atribute di perkecil untuk mendapatkan jarak nilai yang lebih kecil. Ada 3 jenis normalisasi yang biasa digunakan, seperti yang dipaparkan oleh Mehmed Kantardzic dalam bukunya Data Mining - Concepts, Models, Methods, and Algorithms, yaitu : *Decimal Scaling, Min-Max Normalization, Standart Deviation Normalization*.

e. Kontruksi atribut (*Attribute Construction*), dimana atribut baru yang dibangun dan ditambahkan oleh himpunan atribut untuk membantu proses data mining.

#### 2.4.2.4 Reduksi Data (*Data Reduction*)

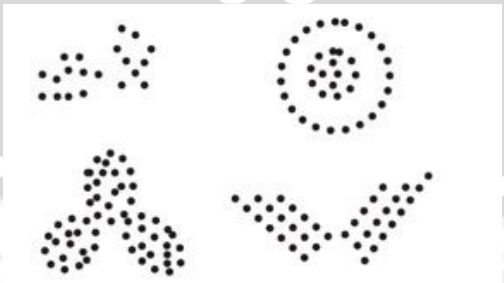
Tujuan utama dari reduksi data adalah untuk menyederhanakan volume data tanpa mengurangi integritas hasil data tersebut. Ada beberapa stategi yang bisa digunakan untuk mereduksi data, yaitu : data cube aggregation, attribute subset selection, dimensionality reduction, numerosity reduction, Discretization and concept hierarchy generation.

## *2.5. Clustering*

### 2.5.1. Pengertian *Cluster*

Menurut Rui Xu dan Don Wunsch dalam buku nya yang berjudul Clustering (2009) yang mengutip dari Everitt et al (2001), sulit untuk mengartikan kelompok, bahkan terkadang salah mengartikan tetapi kelompok dapat diilustrasikan sebagai berikut:

a. Sebuah kelompok adalah satu set entitas yang sama dan entitas dari kelompok yang lain berbeda. Kelompok adalah sebuah agregasi dari titik di ruang tes seperti bahwa jarak antara dua titik dalam kelompok lebih kecil dari jarak antara titik dalam kelompok dan titik di luar itu.
b. Kelompok dapat digambarkan sebagai daerah kontinu dari ruang (d-fitur dimensi ruang) yang berisi kepadatan titik yang tinggi, terpisah dari daerah lain seperti tiap daerah mengandung kepadatan titik yang rendah.



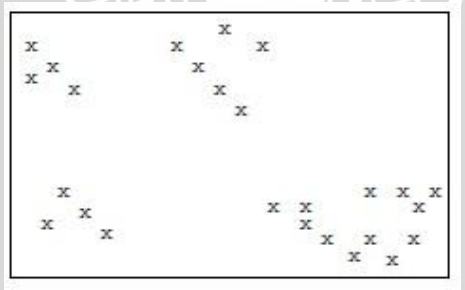
Gambar 2.8 Contoh Kelompok
(Sumber: Xu & Wunsch, 2009)

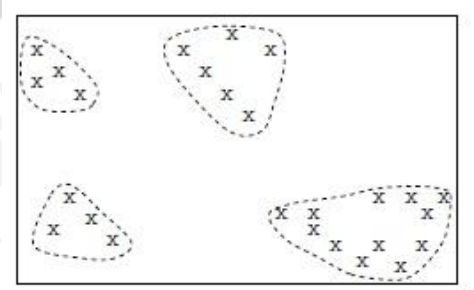
### 2.5.2. Konsep *Clustering*

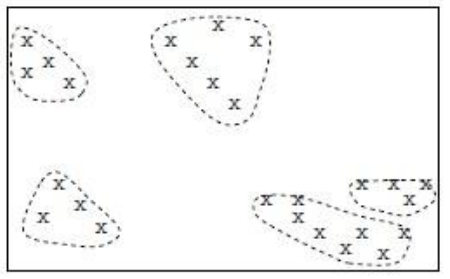
Data Pengelompokan (*clustering*), juga disebut analisis kelompok, analisis segmentasi, analisis taksonomi, atau *unsupervised classification* adalah sebuah metode yang mengelompokan objek atau kelompok, sedemikian sehingga objek didalam satu kelompok memiliki kemiripan dan objek yang berbeda memiliki jarak yang cukup dengan kelompok lain. Data klastering seringkali rancu dengan klasifikasi, dimana objek sudah di kelaskan terlebih dahulu. Pada pengelompokan, kelas juga sudah didefinisikan. (Gan, Ma, & Wu, 2007)

Pengelompokan juga dapat berfungsi sebagai langkah *preprosesing* untuk mengenali kelompok yang homogen untuk membangun model supervised. Model pengelompokan berbeda dari model supervised dimana hasil nya tidak diketahui, dengan kata lain tidak ada atribut target. Model supervised memprediksi nilai untuk atribut target dan tingkat kesalahan (*error rate*) antara target dan nilai prediksi dapat dihitung untuk membimbing pembangunan objek. Model pengelompokan, di sisi lain, dibangun menggunakan optimasi kriteria dimana mendukung kesamaan intraklaster tinggi dan interklaster rendah. Model tersebut dapat digunakan untuk menetapkan indentifikasi titik data. (Taft, et al., 2005)

Ilustrasi dari pengelompokan ini dipaparkan oleh Max Bramer dalam buku yang berjudul Principles of Data Mining (2007) dapat dilihat dibawah ini:







Gambar 2.9 Data yang Terkelompok

(Sumber: Max Bramer, 2007)

Dalam buku Data Mining: Concept and Technique (2006), Jiawei Han dan Micheline Kamber menyebutkan syarat pengelompokan adalah sebagai berikut:

a. Skalabilitas (Scalability): beberapa algoritma pengelompokan bekerja dengan baik pada dataset kecil yang mengandung sedikit objek daripada yang mengandung ribuan objek data. Namun, database yang besar mungkin memiliki miliaran objek. Pengelompokan pada sample seperti ini akan membuat hasil ang bias.
b. Kemampuan untuk menangani berbagai jenis atribut (Ability to deal with different types of attributes): Beberapa algoritma di desain untuk data numeric cluster interval based. Namun, pada kenyataannya pengelompokan membutuhkan tipe data lain, seperti binary atau kategori (nominal), dan data ordinat atau gabungan dari tipe – tipe data tersebut.
c. Penemuan kelompok dengan bentuk acak (Discovery of Clusters with arbitrary shape): Beberapa algoritma pengelompokan menentukan kelompok berdasarkan ukuran jarak Euclidean atau Jarak Manhattan. Algoritma berdasarkan jarak tersebut cenderung untuk menemukan kelompok dengan ukuran dan kepadatan yang serupa. Namun, sebuah kelompok dapat berbentuk apapun. Penting untuk mengembangkan algoritma yang dapat mendeteksi kelompok dengan bentuk acak.
d. Persyaratan minimal untuk domain pengetahuan untuk menentukan parameter input: Beberapa algoritma pengelompokan membutuhkan

pengguna untuk menginputkan parameter dalam analisa kelompok (seperti jumlah kelompok yang ingin dibentuk). Hasil pengelompokan bisa menjadi cukup peka untuk parameter input. Terkadang sulit untuk menentukan parameter, terutama untuk data set yang mengandung objek dimensi tiggi. Hal ini tidak hanya membebani pengguna, tetapi juga membuat kualitas pengelompokan sulit dikendalikan.

e. Kemampuan untuk menangani data yang memiliki noisy: Sebagian besar database mengandung data yang outlier, atau hilang, atau tidak diketahui atau error. Beberapa algoritma pengelompokan peka terhadap data seperti itu dan dapat membuat kualitas pengelompokan menjadi buruk.
f. Increment pada pengelompokan dan ketidakpekaan pada urutan catatan masuk: Beberapa algoritma pengelompokan tidak dapat menggabungkan data yang baru dimasukan (contohnya database update) pada pengelompokan yang sudah ada dan sebaliknya harus menentukan kelompok baru dari awal. Beberapa algoritma pengelompokan peka terhadap urutan penginputan data. Artinya, diberikan sebuah data objek, seperti algoritma mungkin mengembalikan kelompok yang berbeda tergantung pada urutan dari presentasi objek inputan. Membangun algoritma increment pengelompokan dan algoritma yang tidak peka terhadap urutan inputan sangatlah penting.
g. Dimensi tinggi: Database atau data warehouse bisa mengandung beberapa dimensi atau atribut. Beberapa algoritma pengelompokan baik dalam menangani data berdimensi rendah, melibatkan hanya dua atau tiga dimensi. Mata manusia baik dalam menilai kualitas pengelompokan hingga tiga dimensi data. Menemukan kelompok dari objek data pada dimensi tinggi merupakan tantangan.
h. Pengelompokan berdasarkan constraint: Aplikasi pada dunia nyata mungkin memerlukan pengelompokan dengan berbagai constraint. Misalkan pekerjaan anda adalah memilih sejumlah lokasi untuk sejumlah tertentu mesin ATM baru. Untuk memutuskannya, anda

mungkin mengelompokan berdasarkan constraint seperti sungai dan jalur jalan raya, dan tipe maupun jumlah customer. Tantangannya adalah menemukan kumpulan data dengan pengelompokan yang bai yang dapat memenuhi constraint.

i. Interpretability dan kegunaan: Pengguna mengharapkan hasil pengelompokan dapat di tafsirkan, dipahami dan digunakan. Artinya, pengelompokan mungkin perlu terikat pada interpretasi dan aplikasi semantic tertentu. Mempelajari bagaimana tujuan sebuah aplikasi dapat mempengaruhi metode dan fitur pengelompokan yang dipilih.

### *2.5.3. Distance and Similarity*

Pada metode pengelompokan, konsep yang digunakan adalah konsep matrik kesamaan data (*similarity*) maupun distance (*dissimilarity*). Kedua konsep tersebut dapat digunakan untuk mengukur kesamaan antara dua data. Untuk pemilihan konsep yang digunakan bergantung pada model yang akan dibangun (Sullivan, 2012).

Karena kesamaan adalah dasar definisi kelompok, ukuran kesamaan antara dua pola yang diambil dari ruang fitur yang sama adalah paling penting untuk algoritma pengelompokan. Langkah ini harus dipilih sangat hati-hati karena kualitas proses pengelompokan tergantung pada keputusan ini. Hal yang paling umum yang digunakan untuk menghitung, bukan ukuran kesamaan, melainkan ketidaksamaan antara dua sampel menggunakan ukuran jarak yang ditentukan pada ruang fitur. Ukuran yang digunakan mungkin ukuran jarak metrik atau kuasi-metrik pada ruang sampel, dan digunakan untuk mengukur perbedaan sampel (Kantardzic, 2003).

Tabel 2.1 Fungsi Jarak

(Sumber: Pedrycz, 2005)

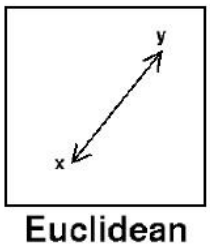
| Distance Function | Formula and Comments |
|---|---|
| Euclidean Distance | $d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2}$ |
| Hamming (city block) Distance | $d(x,y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$ |
| Tchebyschev Distance | $d(x,y) = \max_{i=1,2,\dots n} \lvert x_i - y_i \rvert$ |
| Minkowski Distance | $d(x,y) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}$ ,$p > 0$ |
| Canberra Distance | $d(x,y) = \sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{x_i + y_i}$ $x_i \text{ and } y_i \text{ are positive}$ |
| Angular Separation | $d(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{[\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2]^{1/2}}$ |

### 2.5.4. *Euclidean Distance*

Fungsi Euclidean Distance mengukur jarak antara titik X (X1, X2, dll) dan titik Y (Y1, Y2, dll) adalah:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \dots\dots\dots (2.1)$$

Menderivasi jarak Euclidean antara dua titik data melibatkan perhitungan perbedaan akar kuadrat dari jumlah kuadrat dari antara nilai-nilai yang sesuai, dapat digambarkan sebagai berikut:




Gambar 2.10 Ilustrasi Jarak Euclidean

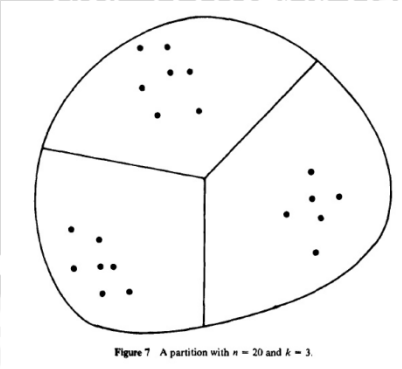
(Sumber: Euclidean and Euclidean Square, 2004)

### 2.5.5. Metode *Clustering*

Menurut Jiawei Han dan Micheline Kamber dalam bukunya yang berjudul Data Mining Concepts and Teqniques (2005) mengatakan bahwa ada 5 jenis metode yang ada dalam konsep pengelompokan, yaitu : metode partisi, metode hirarki, metode berdasarkan kepadatan, metode berdasarkan grid, dan metode berdasarkan pengelompokan.

a. Metode Partisi (*Partitioning Methods*)

Metode ini mengatur objek yang ada ke dalam partisi sejumlah k, dimana masing - masing partisi tersebut mewakili kelompok yang ada. Kelompok dibentuk untuk mengoptimalkan kriteria partisi objektif, seperti fungsi perbedaan berdasarkan jarak, sehingga obyek dalam kelompok adalah sama sedangkan objek dari kelompok yang lain berbeda. Contoh algoritma darimetode ini adalah K-Means dan K-Medoids.




Gambar 2.11 Data yang Terkelompok Menjadi 3

(Sumber: Kaufman & Rousseeuw, 2005)

b. Metode Hirarki (*Hierarchical Methods*)

Metode hirarki ini bekerja dengan mengelompokan objek data menjadi pohon kelompok. Metode ini lebih sering digunakan untuk pengelompokan data yang didasarkan pada matriks kesamaan yang terbentuk perhitungan jarak dari semua titik yang ada (Han & Kamber, 2006).

Ada 2 macam algoritma pada metode hirarki ini, yaitu metode partisi rekursif (metode difisif) atau metode penggabungkan (agglomeratif) dari kelompok yang ada. Algoritma agglomeratif membagi data menjadi kelompok - kelompok kecil sendiri lalu langkah selanjutnya adalah menggabungkan dua kelompok terdekat menjadi satu kelompok baru. Dengan cara ini, jumlah kelompok akan terus berkurang setiap langkahnya. Pada akhirnya semua data akan berkumpul di satu kelompok besar (Larose, 2005).

Untuk algoritma partisi rekursif, mulanya data berkumpul menjadi satu kelompok besar, dan data yang paling berbeda akan memisahkan diri secara rekursif ke dalam kelompok terpisah hingga semua data sudah terkelompok - kelompok sendiri (Larose, 2005).

c. Metode Berdasar Kepadatan (*Density-Based Methods*)

Untuk membentuk kelompok dengan bentuk acak digunakanlah metode berdasar kepadatan. Metode ini menganggap kelompok sebagai daerah yang padat objek yang dipisahkan oleh daerah yang memiliki noise rendah. Ada 3 algoritma yang biasa digunakan dalam metode ini, yaitu: (Han & Kamber, 2006)

1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
2. OPTICS (Ordering Points to Identify the Clustering Structure)
3. DENCLUE (Clustering Based on Density Distribution Functions)

d. Metode Berdasar Grid (*Grid-Based Methods*)

Pendekatan menggunakan metode berdasarkan grid ini menggunakan struktur data grid multiresolusi. Metode ini menghitung ruang objek ke dalam jumlah yang terbatas untuk membentuk struktur data berupa grid. keuntungan dari pendekatan ini adalah waktu pemrosesan yang cepat dimana tidak bergantung pada jumlah objek melainkan pada jumlah cell yang terdapat pada setiap dimensi dalam ruang yang terkuantitasi. Algoritma yang biasa digunakan dalam pendekata ini adalah:

1. Sting, menggunakan statistik informasi yang tersimpan dalam sel-sel grid
2. WaveCluster, mengelompokan objek menggunakan metode wavelet transform.
3. CLIQUE, pendekatan grid-and density-based approach untuk pengelompokan dalam dimensi tinggi.

e. Metode Berdasar Model Pengelompokan (*Model-Based Clustering Methods*)

Metode ini merupakan metode yang berusaha mengoptimalkan kecocokan antara data dengan model matematika. Metode ini didasari pada asumsi dari campuran distribusi probabilitas yang mendasarinya.

## 2.6. Klasifikasi

### 2.6.1. Konsep Klasifikasi

Klasifikasi adalah membagi objek ke dalam kategori atau kelas. Dalam konteks data mining, klasifikasi dilakukan dengan menggunakan model yang dibangun pada data historis. Tujuan dari klasifikasi adalah untuk memprediksi kelas target untuk setiap record data secara akurat dimana data tidak berada dalam data historis (Taft, et al., 2005).

Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan.Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan

tersebut digunakan pada data-data baru untuk diklasifikasi. Teknik ini menggunkan supervised induction, yang memanfaatkan kumpulan pengujian dari record yang terklasifikasi untuk menentukan kelas-kelas tambahan (Kusnawi, 2007).

Teknik klasifikasi berbeda dengan teknik pengelompokan, memang kedua teknik ini hampir sama karena mengelompokan objek kedalam kelas - kelas. Tidak seperti pengelompokan, klasifikasi mengharuskan pengguna mengetahui terlebih dahulu cara kelas didefinisikan. Hal ini sangat dibutuhkan bahwa setiap data sudah memiliki nilai untuk menentukan kelas. Oleh karena itu, klasifikasi lebih mengeksplorisasi daripada pengelompokan (Colet)

Proses klasifikasi dimulai dengan membangun data latih dimana nilai target diketahui. Pada algoritma klasifikasi yang berbeda akan menggunakan teknik yang berbeda pula untuk menemukan relasi antara nilai atribut prediktor dan nilai atribut target pada data latih. Hubungan ini dirangkum dalam model, model tersebut dapat diterapkan dalam kasus baru dimana nilai target tidak diketahui untuk memprediksi nilai target (Taft, et al., 2005).

### 2.6.2. Algoritma Klasifikasi

Menurut Margareth Taft et al dalam bukunya yang berjudul Oracle Data Mining Concepts menyebutkan ada empat algoritma klasifikasi yang biasanya digunakan, yaitu:

*a. Decision Tree Algorithm*

Pohon keputusan secara otomatis menghasilkan aturan, yaitu pernyataan bersyarat yang mengungkapkan logika yang digunakan untuk membangun pohon.

*b. Naive Bayes Algorithm*

Naive Bayes menggunakan Teorema Bayes ', formula yang menghitung probabilitas dengan menghitung frekuensi dari nilai-nilai dan kombinasi dari nilai-nilai dalam data historis.

*c. Adaptive Bayes Network Algorithm*

Algoritma Naive Bayes yang bersifat adaptif.

d. *Support Vector Machine Algorithm*

*Support Vector Machines* (SVM) adalah algoritma yang kuat berdasarkan regresi linier dan nonlinier untuk mengimplementasikan SVM untuk klasifikasi biner dan multiclass.

| Feature | Naive Bayes | Adaptive Bayes Network | Support Vector Machine | Decision Tree |
|---|---|---|---|---|
| Speed | Very fast | Fast | Fast with active learning | Fast |
| Accuracy | Good in many domains | Good in many domains | Significant | Good in many domains |
| Transparency | No rules (black box) | Rules for Single Feature Build only | No rules (black box) | Rules |
| Missing value interpretation | Missing value | Missing value | Sparse data | Missing value |

Gambar 2.12 Perbedaan Algoritma Klasifikasi

(Sumber: Larose, 2005)

Jiawei Han dan Micheline Kamber menambahkan tujuh algoritma yang ada pada klasifikasi dalam bukunya yang berjudul Data Mining: Concepts and Techniques Second Edition. Algoritma tersebut adalah:

a. *Rule Based Classification*

Algoritma ini menggunakan aturan IF-THEN. Dapat dihasilkan dari decision tree ataupun secara langsung.

b. *Backpropagation*

Merupakan algoritma jaringan syaraf tiruan dimana jaringan menyesuaikan bobot sehingga dapat memprediksi label kelas yang benar.

c. *Associative Classification*

Seringkali pola yang menarik atau sering berelasi dengan berelasi dengan kondisi atribut dan label kelas. Oleh karena itu, aturan asosiasi ini sering digunakan pula untuk klasifikasi.

d. *Lazy Learners (Learning from Your Neighbors)*

Algoritma ini kurang bekerja ketika proses pelatihan, sebaliknya bekerja pada saat proses klasifikasi berjalan. Hal ini

dikarenakan algoritma menyimpan "*instance*" tuple sebagai contoh, sehingga sering juga disebut algoritma Instance-based.

*e. Genetic Algorithms*

Algoritma ini digunakan pada masalah masalah optimasi klasifikasi.

*f. Rough Set Approach*

Digunakan pada klasifikasi untuk menemukan hubungan struktural dalam data yang tidak tepat atau noise. Algoritma ini berlaku untuk atribut dengan nilai diskrit. Untuk atribut dengan nilai kontinu, harus di ubah menjadi nilai diskrit terlebih dahulu.

*g. Fuzzy Set Approaches*

Algoritma ini sangat berguna pada sistem data mining yang memberlakukan aturan asosiasi.

## 2.7. *K-Means*

Algoritma *K-Means* merupakan sebuah algoritma *clustering* (pengelompokan) dimana membagi data berdasarkan jarak antar data ke jumlah kelompok yang telah ditetapkan (asalkan ada cukup banyak kasus yang berbeda). Algoritma berbasis jarak ini bergantung pada jarak metrik (fungsi) untuk mengukur kesamaan antara titik data. Untuk menghitung jarak metrik biasa digunakan Jarak Euclidean, cosine, atau Jarak *Fast Cosine* . Data dimasukkan ke kelompok terdekat sesuai dengan hasil jarak metrik yang digunakan (Taft, et al., 2005)

Pengelompokan menggunakan *K-Means* bermaksud untuk mempartisi n objek ke dalam k kelompok dimana setiap objek dimasukan ke dalam mean k terdekat. Metode ini menghasilkan kelompok k dengan perbedaan yang memungkinkan. Jumlah terbaik dari kelompok k didasari pada jarak yang disebut dengan apriori dan harus dihitung dari data yang ada. Tujuan dari metode ini adalah meminimalkan jumlah varian antar klaster. Dengan fungsi kesalahan kuadrat sebagai berikut :

$$J = \sum_{j=1}^{K} \sum_{n \epsilon S_j} (x_n - \mu_j)^2 \quad \ldots\ldots\ldots (2.2)$$

(Sumber: Sayad, 2012)

Dimana k adalah jumlah kelompok Si, (i = 1, 2,... k), μj adalah titik centroid atau rata-rata semua xn poin dalam Si.

Untuk menhitung *centroid* digunakan perhitungan dengan mencari nilai tengah dari kumpulan data dalam sebuah kelompok. Perhitungan ini didasarkan pada rumus *average*:

$$average = \sum_{i=1}^{n} \frac{x_i}{n} \ldots\ldots\ldots (2.3)$$

Menurut Max Bramer dalam buku *Principles of Data Mining* (2007) algoritma pengelompokan K-Means secara umum adalah sebagai berikut:

1. Pilih nilai k
2. Pilih objek k secara random. Gunakan nilai tersebut sebagai set awal nilai centroid.
3. Masukan masing - masing objek ke dalam kelompok yang terdekat dengan nilai centroid.
4. Hitung ulang nilai centroid k.
5. Ulangi langkan 3 dan 4 hingga centroid tidak bergerak.

Untuk Pseudocode dari algoritma K-Means ini dijabarkan sebagai berikut: (Kadous, 2002)

```
Inputs:
     I = {i1, …, ik } (Instance to be clustered)
     N (Number of clusters)
Outputs :
     C= {c1, …, Cn} (cluster centroids)
     M : I → C (cluster membership)

Procedure KMeans
     Set C to initial value (e.g. random selection of I)
     For each i, € I
          M(ij) = argmin distance(ij, ck)
                k€{1…n}
     End
     While m has changed
          For each j € {1..n}
               Recompute ij as the centroid of {i|m(i) = j}
```

```
            End
            For each ij € P
                 M(ij) = argmin distance(ij, ck)
                         k€{i..n}
            End
      End
Return C
End
```

## 2.8. KNN *(K-Nearest Neighbour)*

Algoritma KNN (k- Nearest Neighbor ) pertama kali dijelaskan pada tahun 1950-an dan kurang mendapat popularitas sampai akhir tahun 1960-an. Ketika terjadi peningkatan daya komputasi, barulah algoritma ini sering digunakan dalam pengenalan pola(pattern recognize). (Han & Kamber, 2006)

Algoritma KNN merupakan algoritma yang paling sering digunakan pada klasifikasi, meskipun dapat juga digunakan untuk estimasi dan prediksi. KNN adalah contoh dari instance-based learning dimana kumpulan data latih disimpan, sehingga data yang belum terklasifikasi dapat dikelaskan dengan cara membandingkan dengan catatan yang paling mirip pada data latih (Larose, 2005). KNN biasanya digunakan untuk semua atribut yang sifatnya kontinu, meskipun dapat dimodifikasi untuk menangani atribut yang bersifat kategorikal (Bramer, 2007).

Algoritma ini merupakan algoritma yang sederhana dibanding algoritma pembelajaran yang lainnya. Sebuah objek dikelaskan berdasarkan jarak kedekatan dengan tetangganya k. Nilai k merupakan nilai integer, biasanya kecil . Untuk menentukan ‘tetangga’, posisi objek diwakili dengan vektor dalam ruang multidimensi. Biasanya menggunakan Euclidean Distance ataupun Manhattan Distance dapat digunakan sebagai gantinya (Phyu, 2009).

Untuk menentukan nilai k yang terbaik dilakukan melalui eksperimen (percobaan). Dimulai dengan k = 1 digunakan satu set tes untuk memperkirakan tingkat kesalahan dari classifier. Proses ini dapat diulang dengan menambah nilai k untuk sebuah “tetangga” lagi. Nilai k yang memiliki nilai error minimum yang akan dipilih. Secara umum, semakin banyak jumlah data maka semakin banyak nilai k yang akan diperoleh. Karena jumlah data yang bisa saja tak terhingga dan k= 1, nilai error tidak lebih buruk dari dua kali bayesian error rate (teori

minimum). Jika k juga mendekati tak terhingga, tingkat kesalahan mendekati tingkat kesalahan Bayes (Han & Kamber, 2006).

Algoritma KNN umumnya dapat dijabarkan sebagai berikut menurut (Keller, Gray, & Givens, 1985):

$$W = \{x_1, x_2, \dots, x_n\} \dots\dots\dots (2.4)$$

```
BEGIN
 Input y, of unknown classification
 Set K, 1 ≤ K ≤ n
 Initialize i = 1
 DO UNTIL (K-nearest neighbors found)
  Compute distance from y to xi
  IF (i ≤ K) THEN
   Include xi in the set of K-nearest neighbor
   ELSE IF (xi closer to y than any previous nearest neighbor)
   THEN
    Delete the farthest of the K-nearest neighbors
    Include x, in the set of K-nearest neighbors
   END IF
   Increment i
  END DO UNTIL
   Determine the majority class represented in the set of K-
   Nearest neighbor
   IF (a tie exists) THEN
    Compute sum of distance of neighbor in each class which
    tied
    IF (no tie occurs) THEN
     Classify y in the class of minimum sum
    ELSE
     Classify y in the class of last minimum found
    END IF
   ELSE
    Classify y in the majority class
   END IF
END
```

## 2.9. Evaluasi

Evaluasi merupakan tahap akhir pada penelitian dimana proses ini akan memeriksa keakuratan system yang dibuat. Untuk menghitung tingkat keakuratan pengelompokan digunakan *Error Ratio*. *Error Ratio* merupakan sebuah nilai kesalahan yang diperoleh dari hasil perhitungan *cluster* oleh suatu metode dimana hasil yang diharapkan dari suatu proses pengklasteran tidak sesuai dengan yang diharapkan. *Error Ratio* dipakai jika *dataset* yang digunakan adalah *supervised*, namun bisa juga digunakan untuk mengukur tingkat presisi dari metode

*Cluster*ing. Menurut (Barakhbah, 2006), nilai *error* ini diperoleh dari perhitungan matematis sesuai rumus pada persamaan 2.5

$$Error\ Ratio = \frac{Missclassified}{Total\ Data} X\ 100\% \dots\dots\dots (2.5)$$

# BAB III
# METODOLOGI DAN PERANCANGAN

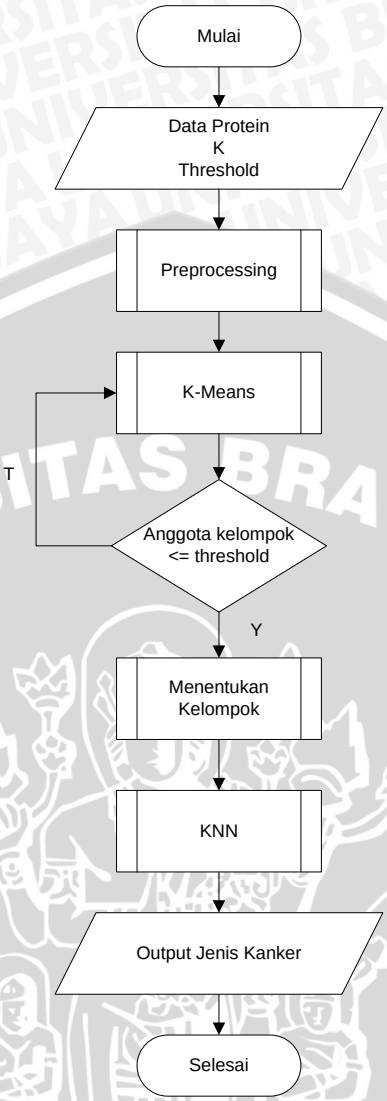
Pada bab metodologi dan perancangan ini akan dibahas metode, rancangan yang digunakan dan langkah langkah yang dilakukan dalam penelitian pengembangan metode klasifikasi berdasarkan pengelompokan anggota kelas yang optimal untuk menentukan jenis kanker berdasarkan susunan protein. Penelitian dilakukan dengan tahapan-tahapan berikut ini :

1. Mempelajari metode dan proses pengelompokan menggunakan K-Mean.
2. Mempelajari metode klasifikasi seperti KNN.
3. Mempelajari *Scoring System* untuk protein.
4. Membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
5. Melakukan proses pelatihan (pembelajaran) terhadap perangkat lunak dengan memasukkan sejumlah data protein yang didapat dari database TP53 sebagai data latih.
6. Melakukan uji coba perangkat lunak yang telah dibuat menggunakan data tes berupa data protein sebagai data uji.

## 3.1. Rancangan Sistem

Sistem yang akan dibuat merupakan implementasi dari penggunaan metode pengelompokan K-Means dan klasifikasi KNN. Sistem ini pada akhirnya akan mampu menkelaskan 3 jenis kanker yang sebelumnya sudah dikelompokan terlebih dahulu menjadi kelompok. Dimana kelompok – kelompok tersebut akan diambil yang paling optimal lalu dilakukan klasifikasi terhadap data yang ada.

Pada tahap awal diperlukan tahap *preprocessing* yaitu mentransformasi data string menjadi numerik. Tahap ini akan membandingkan sebuah sekuen protein dengan sebuah data pembanding sepanjang sekuen tersebut. Selanjutnya data akan dikelompokan menjadi kelompok - kelompok yang didasarkan pada jenis kanker yang akan dikelaskan (*Breast Cancer, Colorectal Cancer, Lung Cancer*) dan non kanker. Sistem yang dibuat diharapkan pada akhirnya dapat menampung satu kelas tepat pada satu kelompok.




Gambar 3.1 Alur Rancangan Sistem

Gambar 3.1 menunjukan alur system yang akan dibuat secara umum. Sistem memuat 4 proses yang utama, yaitu:

1. Preprocessing

   Merupakan proses transformasi data dari data berupa string ke dalam bentuk numerik sehingga jarak antar data dapat dihitung. Proses ini akan lebih lanjut dijabarkan pada subbab 3.3.2.

2. K-Means

   Merupakan proses pengelompokan data dimana data akan dikelompokan menjadi 4 kelompok. Data akan dihitung jaraknya menggunakan *Euclidean*

*Distance* dan akan di periksa apakah benar satu kelompok mewakili tepat satu kelas. Proses ini akan dijabarkan pada subbab 3.3.3.

3. Menentukan Kelompok

   Merupakan proses dalam menentukan kelompok yang akan dilakukan proses KNN. Proses ini dijabarkan pada subbab 3.3.4.

4. KNN

   Merupakan proses klasifikasi data dimana data yang sudah terkelompokan akan dikelaskan untuk menemukan jenis kanker yang dimiliki. Proses ini didasarkan pada jarak data dengan k tetangganya dan akan dijabarkan pada subbab 3.3.5.

## 3.2. Rancangan Penelitian

Sesuai dengan alur rancangan sistem yang terdapat pada gambar 3.1, maka setiap proses pada alur tersebut dapat dijelaskan sebagai berikut:

### 3.3.1. Analisis Data

Pada penelitian ini, data yang digunakan merupakan sekuen protein terdiri dari 20 residu asam amino yang berupa gabungan string sepanjang 393. Untuk protein yang bersifat *wild type* (normal) didapat secara online dari gen bank milik Swiss yang bernama Uniprot dengan alamat *http://www.uniprot.org/*. Data protein yang digunakan adalah *human* TP53 *isoform 1*. Untuk data kanker maupun non kanker yang akan diteliti didapat dari database TP53 dengan alamat http://p53.free.fr/. Sumber tersebut merupakan gen bank yang dapat dipercaya dan sering digunakan untuk penelitian. Oleh sebab itu, data tersebut digunakan dalam penelitian ini.

Data pada uniprot tersebut diambil dengan pembaharuan terakhir database pada 21 Maret 2012. Untuk data pada database TP53 pernaharuan terakhir pada tanggal 15 Juli 2010. Data yang ada di gen bank tersebut dibagi menjadi 2, sebagian untuk data latih atau training set dan sebagian lagi digunakan untuk data uji atau testing dalam bentuk teks (*.xml). Data latih digunakan pada saat proses pengelompokan dan data uji digunakan pada saat proses klasifikasi. Digunakan pula data pembanding merupakan *human* TP53 *isoform 1* yang bertipe wild type (normal).

Pada penelitian ini, jumlah data yang digunakan adalah sebanyak 982 record. Data tersebut terdiri dari data yang digunakan untuk pelatihan system sebanyak 847 record dan untuk data pengujian digunakan sebanyak 135 record data. Pengambilan data sampel tersebut didasarkan pada teknik sampling acak sederhana. Untuk rincian *dataset* yang akan digunakan oleh system dapat dilihat pada table 3.1.

Tabel 3.1 Rincian Data

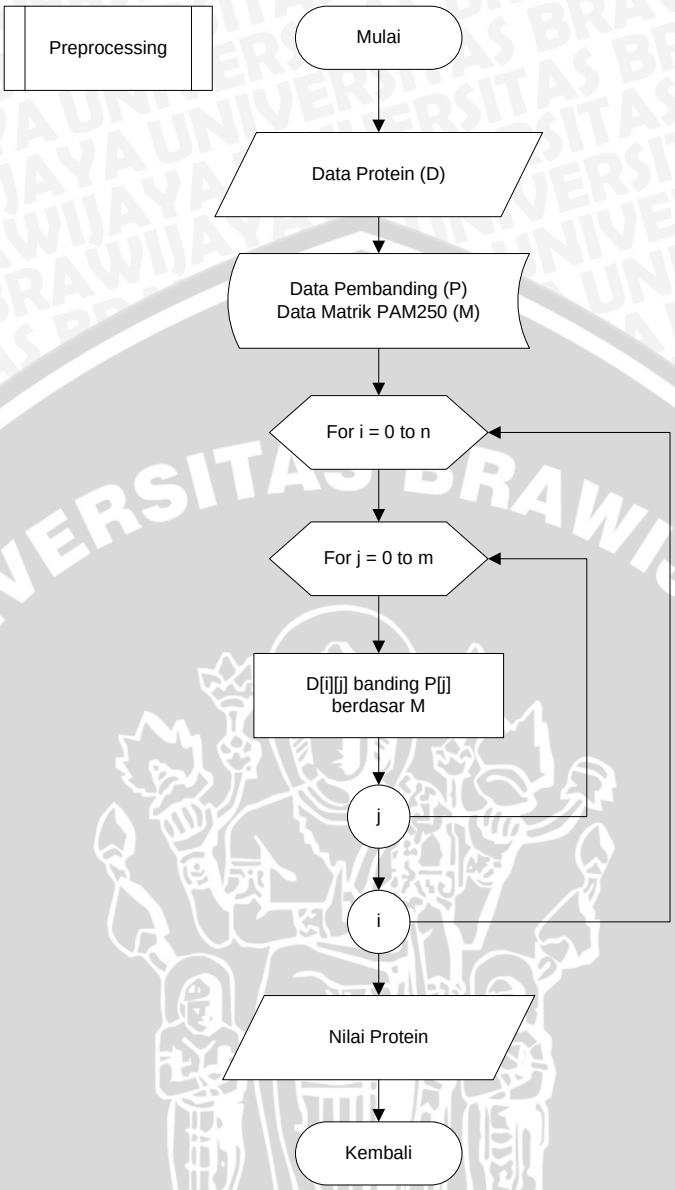
| Jenis | Data Latih | Data Uji |
|---|---|---|
| Non Cancer | 451 | 39 |
| Breast Cancer | 132 | 32 |
| Colorectal Cancer | 132 | 32 |
| Lung Cancer | 132 | 32 |

Banyak variable yang akan digunakan adalah sepanjang 393 string yang dimiliki oleh data protein TP53, sehingga jumlah variable yang dimiliki untuk setiap record data pada system adalah 393.

### 3.3.2. *Preprocessing*

Protein yang terdiri dari 20 residu asam amino direpresentasikan dengan string sepanjang 393. Dimana untuk proses pengelompokan dan klasifikasi dibutuhkan data yang berupa data riil (numerik), maka asam amino tersebut perlu di konversikan ke dalam nilai riil dengan menggunakan sebuah metode yang disebut *scoring system*.

Pada tahap ini, untuk setiap data protein dan setiap asam amino nya akan dibandingkan dengan data pembanding lalu akan diberi nilai sesuai dengan nilai pada matrik PAM250. Hasil dari transformasi ini akan mengisi tiap – tiap variable dan selanjutnya akan disebut dengan nilai protein. Sehingga semua data yang akan digunakan pada system ini akan berbentuk nilai numeric. Alur proses *prepocessing* dijelaskan pada gambar 3.2.




Gambar 3.2 Alur *Preprocessing*

### 3.3.3. Proses Pengelompokan Menggunakan *K-Means*

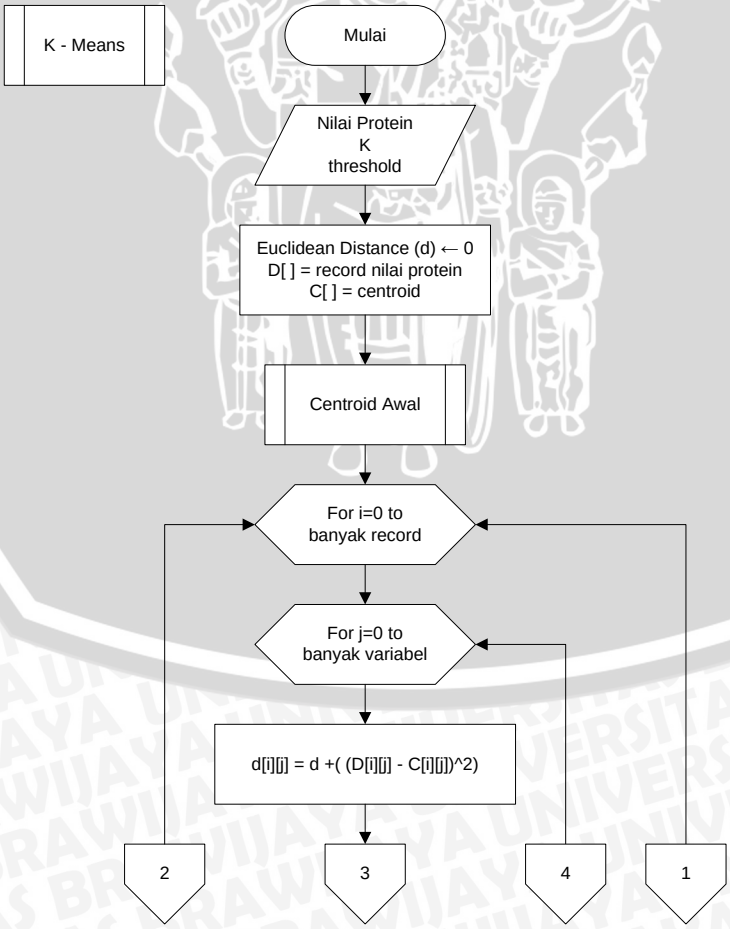
Prinsip dari proses pengelompokan dengan menggunakan *K-Means* pada penelitian ini adalah membentuk kelompok sebanyak inputan *user*. Jumlah kelompok ini yang akan diteliti pada penelitian ini dengan harapan kelompok yang terbentuk akan berisi kelas yang sama.

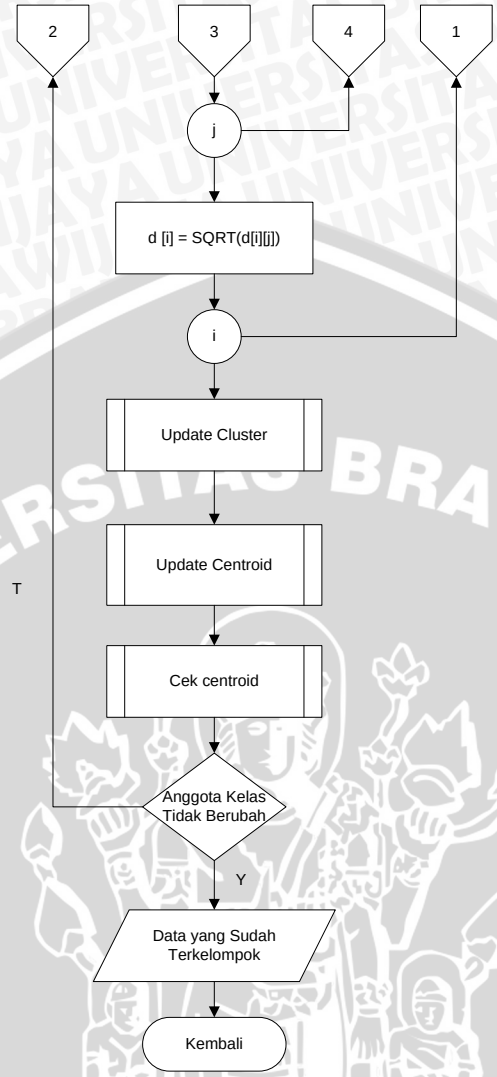
Proses ini dijabarkan sebagai berikut :

1. Masukan (input) berupa data protein yang telah ditransformasikan terlebih dahulu (nilai protein).

2. Masukan (input) nilai k.
3. Menentukan nilai *Euclidean Distance* awal (d) = 0.
4. Menentukan centroid awal (untuk data latih dapat dilihat pada subbab 3.3.3.1).
5. Untuk setiap variable pada setiap data lakukan perhitungan d (*Euclidean Distance*) menggunakan rumus 2.1.
6. Melakukan update *cluster* yaitu, memasukkan data ke dalam kelompok dengan jarak minimum ke centroid berdasarkan nilai d. Untuk penjelasan lebih lanjut dapat dilihat pada subbab 3.3.3.2.
7. Jika sudah satu perulangan, periksa centroid berubah atau tidak, jika iya maka berlanjut ke langkah 8. Jika tidak maka proses kembali ke langkah 5. Untuk penjelasan lebih lanjut dapat dilihat pada subbab 3.3.3.3.

Untuk gambaran alur system pengelompokan K-Means dapat dilihat pada gambar 3.3.






Gambar 3.3 Alur Pengelompokan K-Means

### 3.3.3.1. Penentuan Centroid Awal

Pada data latih, untuk menentukan centroid awal diambil data dengan perbedaan kelas. Centroid untuk klaster pertama diambil dari data dengan indeks 0 (data pertama). Centroid klaster kedua dan seterusnya diambil dari data selanjutnya yang berbeda kelas dengan data sebelumnya.

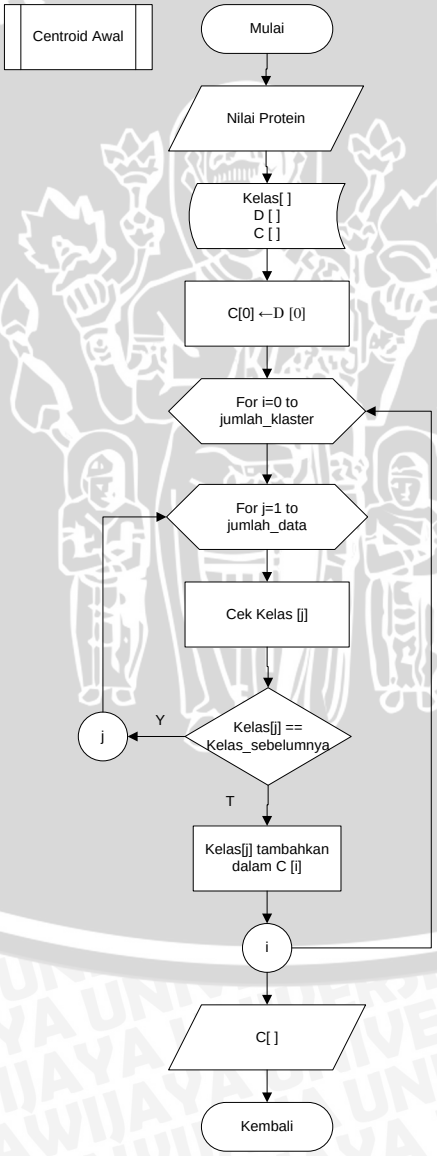
Langkah – langkah proses tersebut dapat dijabarkan sebagai berikut :

1. Masukan (input) berupa data protein yang sudah ditransformasikan(nilai protein).
2. Sebanyak kelompok yang ingin dibentuk, untuk nilai kelas dari 1 sampai jumlah kelas lakukan pemeriksaan apakah kelas data yang diperiksa sama

dengan kelas data sebelumnya, jika iya maka dilakukan perulangan, jika tidak maka indeks data dimasukan ke dalam C.

3. Untuk kelompok pertama hingga sejumlah kelompok, masukan nilai protein ke dalam C yang merupakan variable penyimpan data centroid.
4. Jika semua nilai sudah terpenuhi, maka proses berhenti dan output akan berupa nilai – nilai protein yang akan dijadikan centroid sejumlah kelompok yang akan dibentuk.

Untuk gambaran penentuan centroid awal pada data latih dapat dilihat pada gambar 3.4.



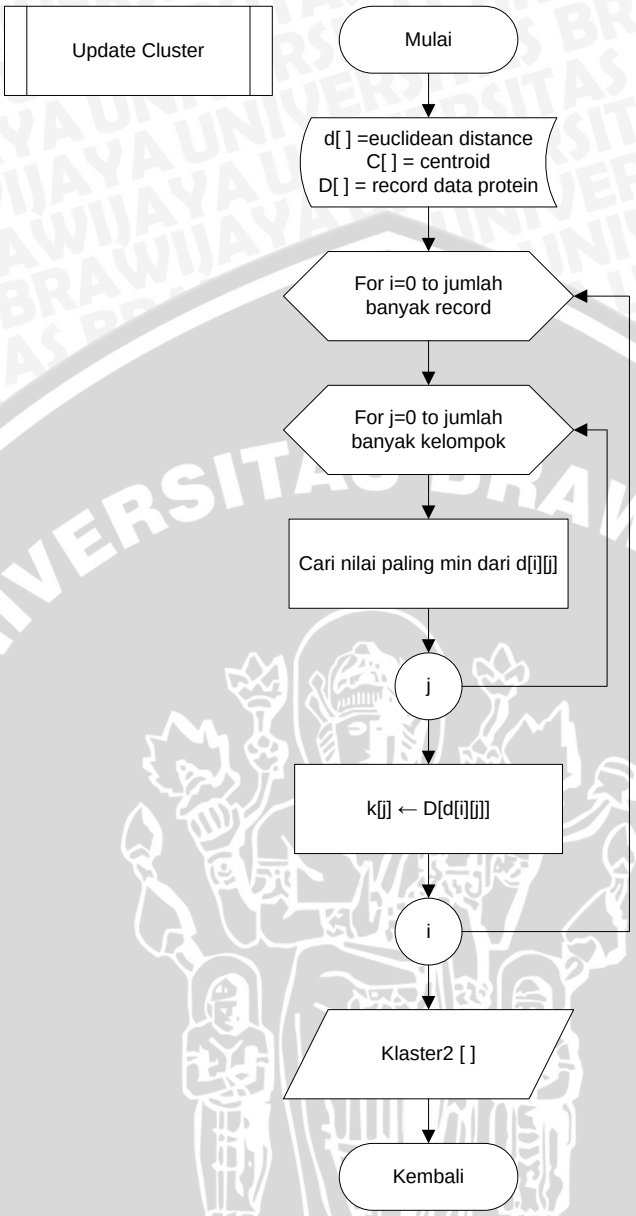

Gambar 3.4 Alur Penentuan Centroid Awal

#### 3.3.3.2. Proses Update Anggota Kelompok

Proses ini merupakan proses menentukan anggota kelompok di tiap kelompok yang sudah ditentukan. Proses ini terjadi pada langkah iterasi awal dan mengupdate anggota kelompok pada iterasi selanjutnya. Proses ini melibatkan hasil perhitungan *Euclidean Distance* dari setiap data terhadap setiap *centroid* kelompok yang terbentuk.

Untuk menentukan anggota yang akan dimasukkan ke dalam kelompok adalah dengan mencari nilai minimum (jarak minimum) tiap data dengan *centroid* yang didapat dari perhitungan *Euclidean Distance* tersebut. Nilai tersebut menunjukan data berada lebih dekat dengan centroid yang keberapa, yang selanjutnya data akan dikelompokan ke dalam nya. Pada proses ini, data sebelum dilakukan update akan disimpan jg sebagai pembanding pada proses pengecekan anggota kelompok.

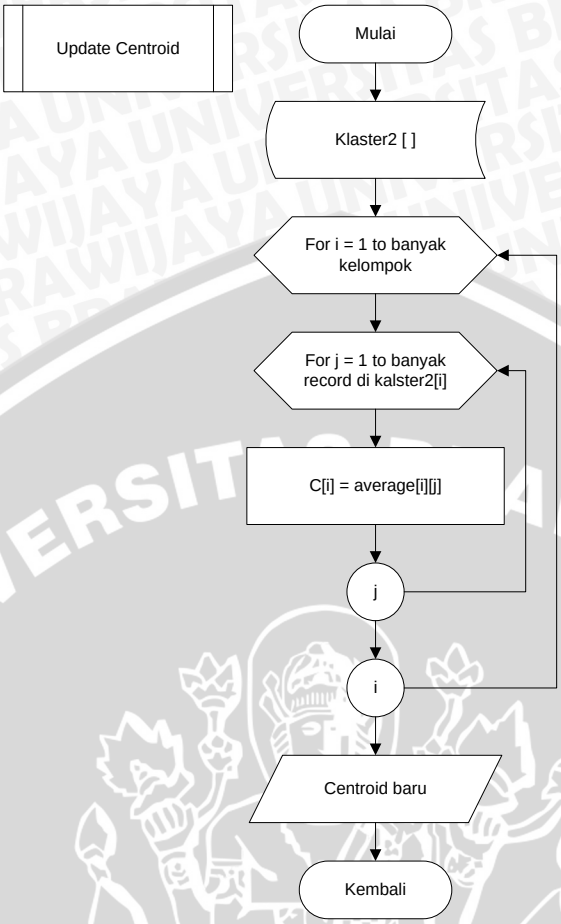
Prosesnya dapat dijelaskan lebih jelas dengan flowchart pada gambar 3.5.




Gambar 3.5 Alur *Update* Kelompok

### 3.3.3.3. Proses Update Centroid

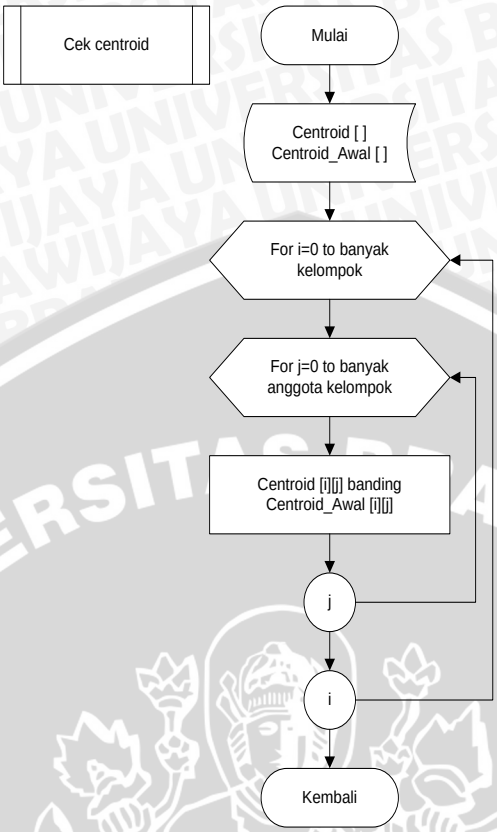
Proses ini akan mengupdate centroid dari tiap – tiap kelompok yang baru terbentuk. Untuk menentukan centroid baru, maka perhitungan dilakukan dengan menghitung rata – rata setiap variable pada setiap record yang terdapat di dalam kelompok. Perhitungannya digunakan persamaan 2.3. Alur proses ini dijabarkan pada gambar 3.6.




Gambar 3.6 Alur Proses *Update Centroid*

### 3.3.3.4. Proses Pengecekan Centroid

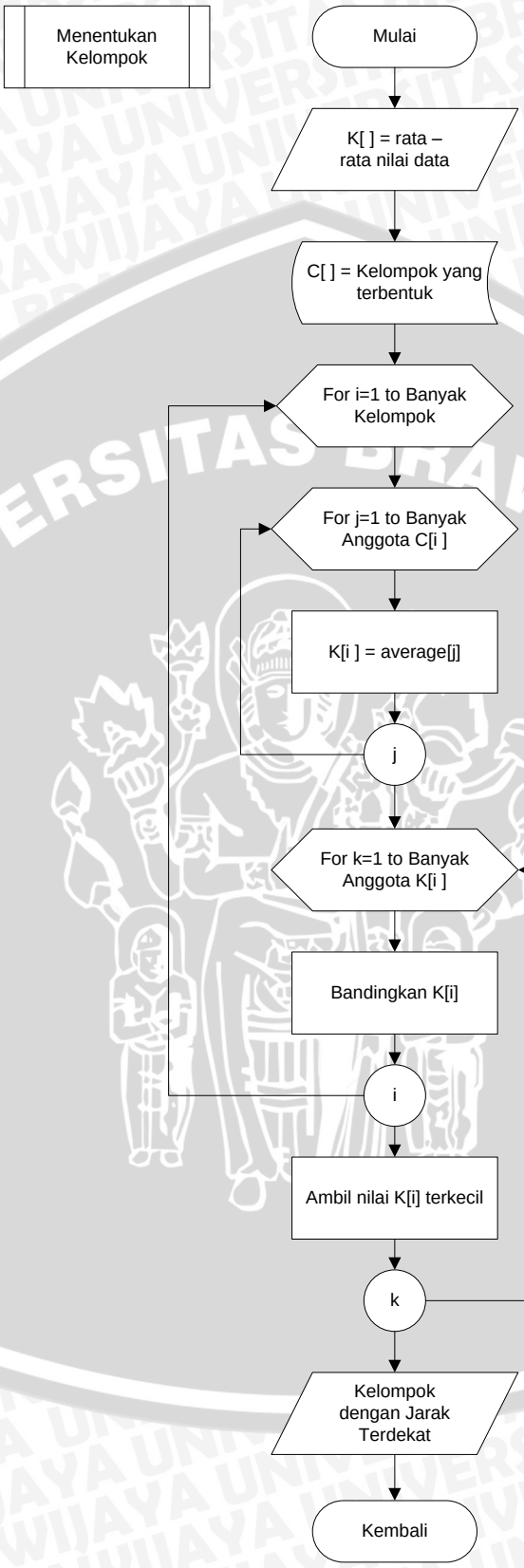
Setelah semua data sudah terkelompokkan, proses selanjutnya adalah melakukan pemeriksaan(pengecekan) centroid apakah pada iterasi selanjutnya nilai centroid masih tetap sama atau berubah setelah proses *update cluster*. Centroid yang sudah terupdate dibandingkan dengan centroid sebelumnya. Jika centroid berubah, maka proses akan kembali dengan menghitung jarak tiap data. Jika tidak, maka proses akan berlanjut ke proses selanjutnya. Untuk alur proses ini dapat dilihat pada gambar 3.7.




Gambar 3.7 Alur Pengecekan Centroid

### 3.3.4. Proses Menentukan Kelompok

Setelah semua kelompok terbentuk, dilakukan proses menentukan kelompok dimana pada proses ini akan ditentukan kelompok mana yang akan diterapkan proses KNN. Prosesnya dengan menghitung nilai rata – rata data pada setiap kelompok lalu dibandingkan jaraknya dengan data uji menggunakan *Euclidean Distance.* Setelah itu barulah diambil indeks kelompok yang memiliki nilai terkecil atau dengan kata lain kelompok dengan jarak terdekat ke data latih. Proses tersebut alurnya dapat dilihat pada gambar 3.10.

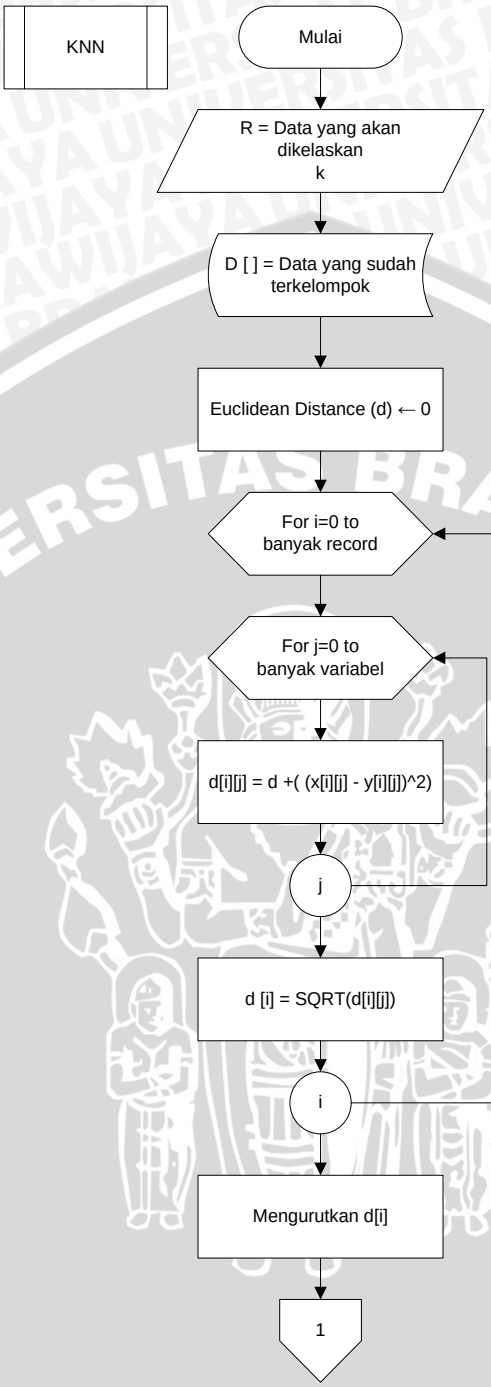


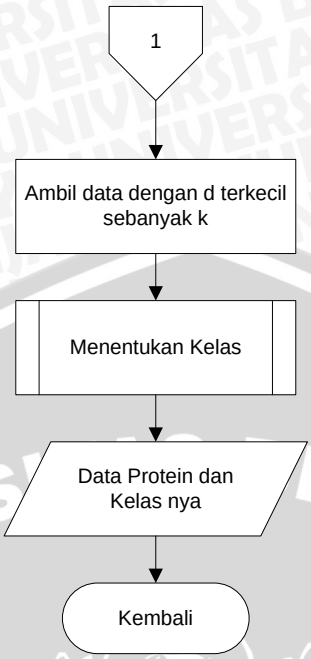

Gambar 3.8 Alur Proses Menentukan Kelompok

### 3.3.5. Proses Klasifikasi Menggunakan KNN (k-Nearest Neighbour)

Proses klasifikasi ini menggunakan data latih yang sudah dikelompokan sebagai data latihnya dengan tujuan untuk mengklasifikasikan sekuen protein yang akan diuji ke dalam empat kelas. Langkah – langkah proses ini dijelaskan sebagai berikut:

1. Masukan data yang akan dikelaskan sebagai inputan dan nilai k yang diinginkan oleh user.
2. Menginisialisasi nilai awal dr *Euclidean Distance* (d) = 0.
3. Untuk setiap iterasi mulai dari record pertama hingga record terakhir pada data latih yang sudah terkelompok dan untuk setiap variable pertama hingga terakhir pada setiap recordnya dilakukan perhitungan jarak *Euclidean Distance* (d) dengan data inputan user.
4. Mengurutkan nilai jarak yang didapat, dari yang paling kecil ke yang paling besar.
5. Ambil data sebanyak k sesuai yang diinputkan user.
6. Melakukan proses penentuan kelas yang akan dijelaskan lebih lanjut pada subbab 3.3.4.1.
7. Keluaran (output) berupa jenis kanker dari data uji.


KNN
Mulai
R = Data yang akan dikelaskan
k
D [ ] = Data yang sudah terkelompok
Euclidean Distance (d) ← 0
For i=0 to banyak record
For j=0 to banyak variabel
d[i][j] = d +( (x[i][j] - y[i][j])^2)
j
d [i] = SQRT(d[i][j])
i
Mengurutkan d[i]
1

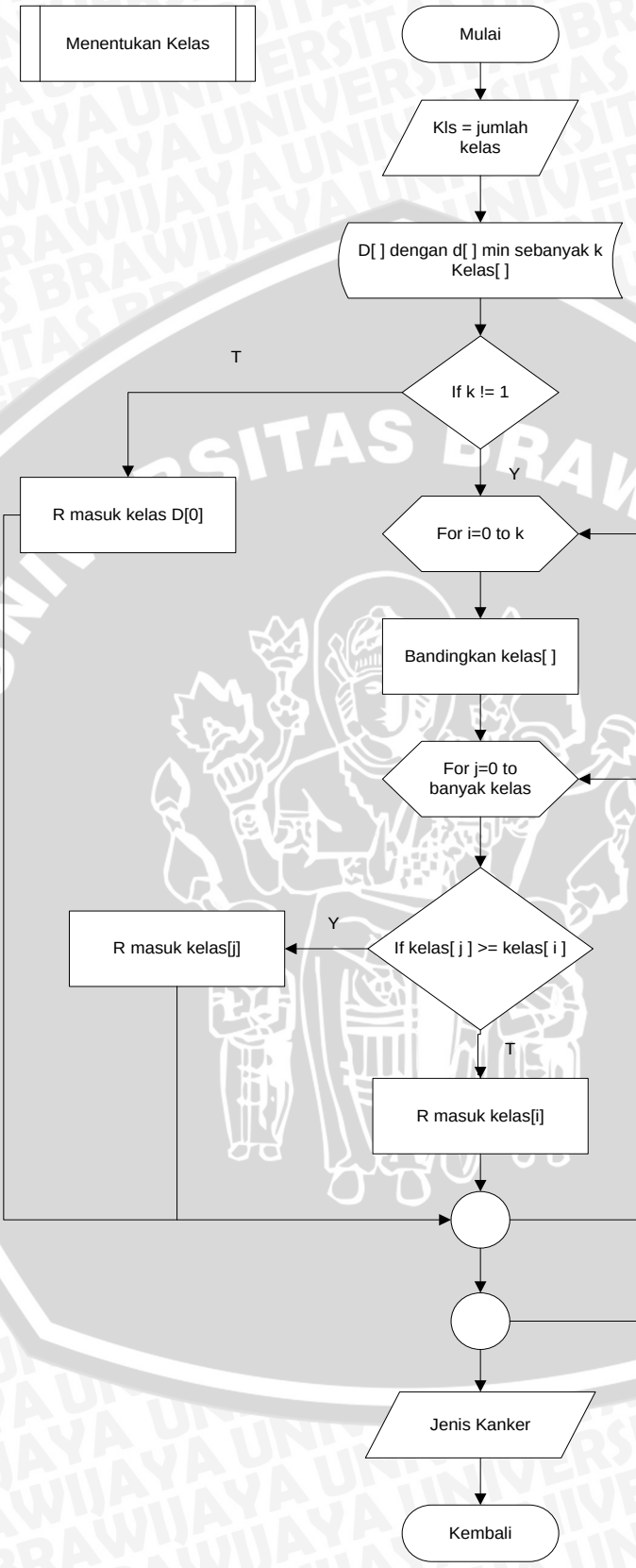



Gambar 3.9 Alur Klasifikasi KNN

### 3.3.5.1.Proses Penentuan Kelas

Proses ini menangani penentuan kelas pada proses klasifikasi di system. Sistem sebelumnya sudah menyimpan data sejumlah nilai k yang diinputkan oleh user dengan nilai jarak d terkecil. Untuk flowchart nya dapat dilihat pada gambar 3.9, proses lebih lanjut dijelaskan sebagai berikut:

1. Jika nilai k sama dengan 1, maka kelas yang dipilih merupakan kelas data pertama.
2. Jika nilai k tidak sama dengan 1 maka dilakukan iterasi sebanyak nilai k lalu bandingkan kelas yang dimiliki data latih.
3. Kelas dibandingkan dengan kelas yang ada.
4. Keluaran (output) berupa jenis kankernya.




Gambar 3.10 Proses Penentuan Kelas

### 3.3.6. Perhitungan Manual

Pada subbab ini akan ditampilkan contoh perhitungan manual untuk proses klasifikasi berdasarkan pengelompokan yang optimal pada data protein untuk menentukan jenis kanker dengan pengelompokan *K-Means* dan klasifikasi menggunakan metode KNN. Untuk perhitungan manual ini, data latih yang digunakan sebanyak 15 data dan data uji pada proses klasifikasi sebanyak 1 data. Data latih yang digunakan tercantum pada table 3.2 dan untuk data uji tercantum pada table 3.3.

Tabel 3.2 Data Latih (*Training Dataset*)

| Data | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | KELAS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | K | S | V | T | C | T | Y | S | P | A | BC |
| D2 | E | S | V | T | C | T | Y | S | P | A | LC |
| D3 | Q | S | V | T | C | T | Y | S | P | A | LC |
| D4 | R | S | V | T | C | T | Y | S | P | A | LC |
| D5 | K | S | V | N | C | T | Y | S | P | A | BC |
| D6 | K | S | V | T | S | T | Y | S | P | A | NC |
| D7 | K | S | V | T | G | T | Y | S | P | A | NC |
| D8 | K | S | V | T | W | T | Y | S | P | A | NC |
| D9 | K | S | V | T | C | T | D | S | P | A | CC |
| D10 | K | S | V | T | C | T | H | S | P | A | CC |
| D11 | K | S | V | T | C | T | Y | F | P | A | BC |
| D12 | K | S | V | T | C | T | Y | S | L | A | BC |
| D13 | M | S | V | T | C | T | Y | S | P | A | LC |
| D14 | K | S | V | T | C | M | Y | S | P | A | BC |
| D15 | K | S | V | T | C | T | Y | S | P | A | BC |

Dari 15 data latih tersebut memiliki 4 kelas yaitu *non cancer* (NC), *breast cancer* (BC), *colorectal cancer* (CC) dan *lung cancer* (LC).

Tabel 3.3 Data Uji (*Data Testing*)

| Data | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | TARGET KELAS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DU1 | K | S | V | T | R | T | Y | S | P | A | NC |

Langka pertama sebelum melakukan pengelompokan adalah melakukan pengecekan apakah jumlah data (record) melebihi threshold yang sudah ditentukan. Pada perhitungan manual ini, threshold yang digunakan sebanyak 5. Jika jumlah data melebihi threshold maka dilakukan proses pengelompokan terlebih dahulu sebelum diterapkan proses klasifikasi, jika data sudah memenuhi syarat threshold maka proses yang dikerjakan adalah proses klasifikasi.

Langkah kedua dilakukan proses *preprocessing data* yaitu dengan melakukan transformasi data dimana string protein yang mewakili setiap variable diubah nilainya menjadi numeric sehingga didapatkan hasil seperti pada table 3.4 untuk data latih dan table 3.5 untuk data uji. Untuk kelas akan dinotasikan dengan angka *non cancer* = 0, *breast cancer* = 1, *colorectal cancer* = 2, *lung cancer* = 3. Untuk memudahkan membaca, nilai yang berbeda (menunjukan mutasi) diberi warna yang berbeda yaitu warna *orange*.

Tabel 3.4 Data Latih yang Sudah Ditransformasi

| Data | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | KELAS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 5 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 | 1 |
| D2 | 0 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 | 3 |
| D3 | 1 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 | 3 |
| D4 | 3 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 | 3 |
| D5 | 5 | 2 | 4 | 0 | 12 | 3 | 10 | 2 | 6 | 2 | 1 |
| D6 | 5 | 2 | 4 | 3 | 0 | 3 | 10 | 2 | 6 | 2 | 0 |
| D7 | 5 | 2 | 4 | 3 | -3 | 3 | 10 | 2 | 6 | 2 | 0 |
| D8 | 5 | 2 | 4 | 3 | -8 | 3 | 10 | 2 | 6 | 2 | 0 |
| D9 | 5 | 2 | 4 | 3 | 12 | 3 | -4 | 2 | 6 | 2 | 2 |
| D10 | 5 | 2 | 4 | 3 | 12 | 3 | 0 | 2 | 6 | 2 | 2 |
| D11 | 5 | 2 | 4 | 3 | 12 | 3 | 10 | -3 | 6 | 2 | 1 |
| D12 | 5 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | -3 | 2 | 1 |
| D13 | 0 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 | 3 |
| D14 | 5 | 2 | 4 | 3 | 12 | -1 | 10 | 2 | 6 | 2 | 1 |
| D15 | 5 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 0 | 2 | 1 |

Tabel 3.5 Data Uji yang Sudah Di transformasi

| Data | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | TARGET KELAS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DU1 | 5 | 2 | 4 | 3 | -4 | 3 | 10 | 2 | 6 | 2 | 0 |

Langkah selanjutnya adalah proses pengelompokan menggunakan *K-Means*. Untuk lebih jelasnya langkah – langkah pengelompokan dijabarkan sebagai berikut:

1. Menentukan nilai k, pada contoh perhitungan ini digunakan k = 2.
2. Menentukan centroid awal dimana diambil dari atribut kelas yang berbeda. Centroid awal data latih dapat dilihat pada table 3.6 dan untuk data uji pada table 3.7.

Tabel 3.6 Centroid Awal Data Latih

| Centroid | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| c1 | 5 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |
| c2 | 0 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |

Untuk centroid 1 (c1) data yang digunakan adalah D1, untuk c2 data yang digunakan adalah D2.

3. Menghitung jarak antar data dengan menggunakan *Euclidean Distance* (d). Jarak ini dihitung menggunakan persamaan 2.1. Berikut contoh perhitungan jarak antara record pertama data latih dengan centroid pertama pada data latih.

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d = \sqrt{\begin{matrix}(5-5)^2 + (2-2)^2 + (4-4)^2 + \\ (3-3)^2 + (12-12)^2 + \\ (3-3)^2 + (10-10)^2 + (2-2)^2 + (6-6)^2 + (2-2)^2\end{matrix}}$$

$$d = \sqrt{0}$$

$$d = 0$$

Selanjutnya adalah menghitung jarak record pertama dengan centroid kedua. Demikian selanjutnya hingga semua didapatkan jarak semua record terhadap masing – masing centroid. Tabel 3.7 memperlihatkan hasil perhitungan jarak pada data latih.

Tabel 3.7 Hasil Perhitungan *Euclidean Distance* pada Data Latih

| Data | c1 | c2 | k |
|---|---|---|---|
| D1 | 0 | 5 | K1 |
| D2 | 5 | 0 | K2 |
| D3 | 4 | 1 | K2 |
| D4 | 2 | 3 | K1 |
| D5 | 3 | 5.830952 | K1 |
| D6 | 12 | 13 | K1 |
| D7 | 15 | 15.81139 | K1 |
| D8 | 20 | 20.61553 | K1 |
| D9 | 14 | 14.86607 | K1 |
| D10 | 10 | 11.18034 | K1 |
| D11 | 5 | 7.071068 | K1 |
| D12 | 9 | 10.29563 | K1 |
| D13 | 5 | 0 | K2 |
| D14 | 4 | 6.403124 | K1 |
| D15 | 6 | 7.81025 | K1 |

K merupakan kelompok yang terbentuk. k1 mewakili kelompok 1 dan k2 mewakili kelompok 2. Untuk menentukan kelompok, dipilih jarak paling minimum.

4. Langkah selanjutnya mengelompokan data berdasarkan jarak yang sudah dihitung dan mencari centroid baru dengan menghitung nilai rata – rata tiap variable nya. Perhitungan ini dilakukan menggunakan persamaan 2.3. Hasilnya dapat dilihat pada table 3.8 untuk data latih.

Tabel 3.8 Hasil Perhitungan Centroid Baru pada Data Latih

| Centroid | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| c1 | 4.75 | 2 | 4 | 3 | 12 | 2.5 | 7 | 1.38 | 4.13 | 2 |
| c2 | 0.333333 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |

5. Hitung kembali jarak antara data ke centroid seperti pada langkah kedua. Setelah iterasi kedua ini dilakukan proses pengecekan apakah data pada tiap kelompok tetap atau berpindah. Pada hasil perhitungan iterasi kedua, dapat dilihat jika data masih berpindah. Maka proses iterasi terus berlanjut. Hasil pada iterasi kedua dapat dilihat pada table 3.9 untuk data latih.

Tabel 3.9 Hasil Perhitungan *Euclidean Distance* Iterasi Kedua

| Data | c1 | c2 | k |
|---|---|---|---|
| D1 | 3.63576 | 4.666667 | K1 |
| D2 | 5.976517 | 0.333333 | K2 |
| D3 | 5.217159 | 0.666667 | K2 |
| D4 | 4.027251 | 2.666667 | K2 |
| D5 | 4.713677 | 5.547772 | K1 |
| D6 | 12.53869 | 12.87547 | K1 |
| D7 | 15.43434 | 15.70916 | K1 |
| D8 | 20.32778 | 20.53723 | K1 |
| D9 | 11.19012 | 14.7573 | K1 |
| D10 | 7.295118 | 11.0353 | K1 |
| D11 | 5.654091 | 6.839428 | K1 |
| D12 | 7.776166 | 10.13794 | K1 |
| D13 | 5.976517 | 0.333333 | K2 |
| D14 | 5.021827 | 6.146363 | K1 |
| D15 | 5.169018 | 7.60117 | K1 |

6. Setelah melakukan iterasi sebanyak lima kali, barulah didapatkan data tidak ada yang berubah lagi. Dilakukan proses pengecekan apakah jumlah data sudah memenihi syarat threshold atau belum. Jika tidak, maka proses pengelompokan diberlakukan terhadap kelompok yang jumlah anggotanya melebihi threshold. Jika sudah memenuhi syarat, maka kelompok tersebut akan diterapkan proses KNN. Pada iterasi

kelima didapatkan hasil yang sudah optimal. Tabel 3.10 menunjukan hasil perhitungan centroid untuk data latih.

Tabel 3.10 Hasil Perhitungan Centroid Iterasi Keempat

| Centroid | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| c1 | 5 | 2 | 4 | 3 | 12 | 3 | -2 | 2 | 6 | 2 |
| c2 | 1 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |

Untuk hasil perhitungan dari jarak antara data dengan centroid dapat dilihat pada table 3.11 untuk data latih.

Tabel 3.11 Hasil Perhitungan *Euclidean Distance* Iterasi Kelima

| Data | c1 | c2 | k |
|---|---|---|---|
| D1 | 12 | 4 | K2 |
| D2 | 13 | 1 | K2 |
| D3 | 12.64911 | 0 | K2 |
| D4 | 12.16553 | 2 | K2 |
| D5 | 12.36932 | 5 | K2 |
| D6 | 16.97056 | 12.64911 | K2 |
| D7 | 19.20937 | 15.52417 | K2 |
| D8 | 23.32381 | 20.39608 | K2 |
| D9 | 2 | 14.56022 | K1 |
| D10 | 2 | 10.77033 | K1 |
| D11 | 13 | 6.403124 | K2 |
| D12 | 15 | 9.848858 | K2 |
| D13 | 13 | 1 | K2 |
| D14 | 12.64911 | 5.656854 | K2 |
| D15 | 13.41641 | 7.211103 | K2 |

Didapatkan bahwa kelompok 1 memiliki 2 anggota, sedangkan untuk kelompok 2 memiliki 13 anggota. Hasil pengelompokan dapat dilihat pada table 3.12. Dikarenakan kelompok 2 tidak memenuhi syarat threshold, maka kelompok 2 akan dipecah lagi menggunakan proses pengelompokan K-Means.

Tabel 3.12 Anggota Kelompok Hasil Pengelompokan

| | |
|---|---|
| K1 | D9 D10 |
| K2 | D1 D2 D3 D4 D5 D6 D7 D8 D11 D12 D13 D14 D15 |

7. Untuk kelompok 2 dilakukan proses seperti pada langkah nomer 2, yaitu dengan menentukan centroid awal dari anggota kelompok yang ada. Tabel 3.13 menunjukan centroid awal pada kelompok 2.

Tabel 3.13 Centroid Awal pada Kelompok Kedua

| Centroid | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| c1 | 5 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |
| c2 | 0 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |

8. Melakukan perhitungan jarak antara data dengan centroid. Hasil perhitungan dapat dilihat pada tabel 3.14.

Tabel 3.14 Perhitungan Jarak antara Data dengan Centroid Kelompok Dua

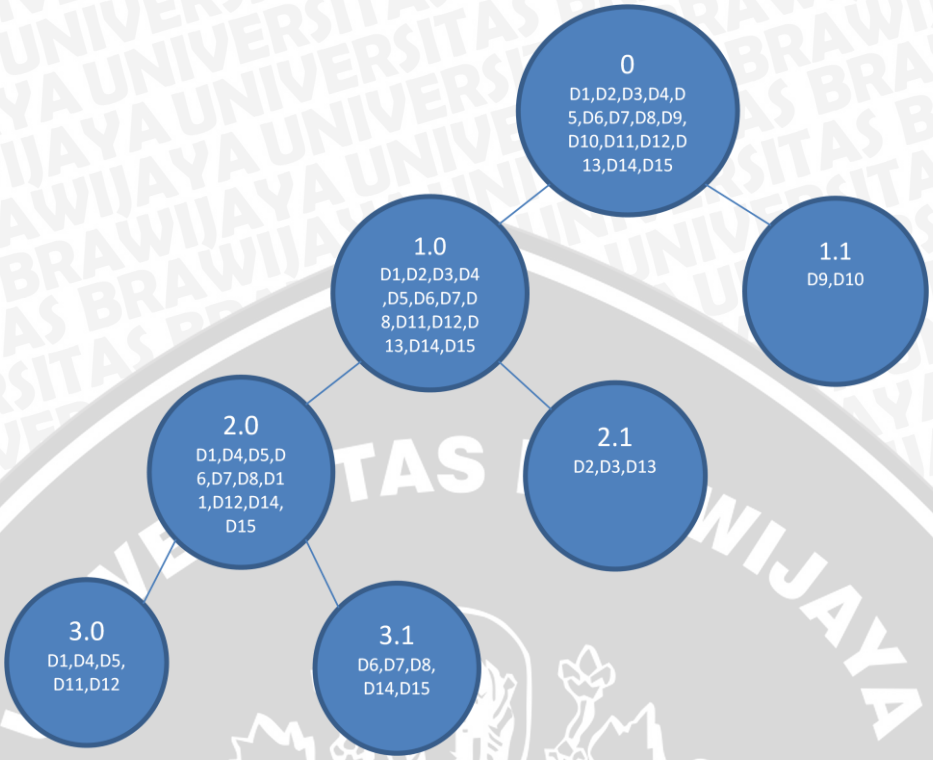
| Data | c1 | c2 | k |
|---|---|---|---|
| D1 | 0 | 5 | K1 |
| D2 | 5 | 0 | K2 |
| D3 | 4 | 1 | K2 |
| D4 | 2 | 3 | K1 |
| D5 | 3 | 5.831 | K1 |
| D6 | 12 | 13 | K1 |
| D7 | 15 | 15.811 | K1 |
| D8 | 20 | 20.616 | K1 |
| D11 | 5 | 7.0711 | K1 |
| D12 | 9 | 10.296 | K1 |
| D13 | 5 | 0 | K2 |
| D14 | 4 | 6.4031 | K1 |
| D15 | 6 | 7.8103 | K1 |

9. Setelah itu, dilakukan perhitungan centroid baru dari masing – masing kelompok. Hasil perhitungan tersebut dapat dilihat pada tabel 3.15.

Tabel 3.15 Hasil Perhitungan Centroid Baru

| Centroid | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| c1 | 4.8 | 2 | 4 | 2.7273 | 7.3 | 2.6 | 10 | 1.5 | 4.6363 64 | 2 |
| c2 | 0.3333 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |

10. Setelah iterasi berlanjut hingga centroid tidak berubah. Maka proses kembali mengecek apakah syarat threshold terpenuhi. Jika tidak, maka akan dilakukan proses pemecahan kelompok lagi hingga syarat threshold terpenuhi. Hasil pengelompokan dalam bentuk tree dapat dilihat pada gambar 3.12.




Gambar 3.11 Pohon Kelompok yang Terbentuk

11. Dari hasil pengelompokan yang sudah optimal tersebut, akan dilakukan proses klasifikasi yang diterapkan pada node anak yang sudah terbentuk. Proses dimulai dengan menghitung jarak *Euclidean Distance* data input dengan centroid dari masing – masing node anak. Jarak ini dihitung sesuai dengan persamaan 2.1. Dibawah ini contoh perhitungan centroid masing – masing node anak.

Tabel 3.16 Perhitungan Centroid Masing - Masing Node Anak

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1.1 | 5 | 2 | 4 | 3 | 12 | 3 | -2 | 2 | 6 | 2 |
| 2.1 | 0.333333 | 2 | 4 | 3 | 12 | 3 | 10 | 2 | 6 | 2 |
| 3.0 | 4.6 | 2 | 4 | 2.4 | 12 | 3 | 10 | 1 | 4.2 | 2 |
| 3.1 | 5 | 2 | 4 | 3 | 2.6 | 2.2 | 10 | 2 | 4.8 | 2 |

Setelah itu dihitung jarak data input ke masing – masing centroid, seperti contoh perhitungan dibawah ini:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d = \sqrt{\begin{matrix}(5-5)^2 + (2-2)^2 + (4-4)^2 + \\ (3-3)^2 + (12-12)^2 + \\ (3-3)^2 + (10-(-2))^2 + (2-2)^2 + (6-6)^2 + (2-2)^2\end{matrix}}$$

$$d = 20$$

Tabel 3.17 menunjukan jarak yang didapat pada masing – masing centroid ke data uji.

Tabel 3.17 Nilai *Euclidean Distance* Data Uji ke Data Latih

| | 1.1 | 2.1 | 3.0 | 3.1 |
|---|---|---|---|---|
| DU1 | 20 | 16.66667 | 16.14806 | 6.755738 |

12. Berdasarkan perhitungan diatas, didapatkan kelompok yang mendekati kesamaan dengan data uji adalah kelompok dengan label 3.18. Selanjutnya dilakukan perhitungan jarak lagi antara data uji dengan anggota kelompok yang terpilih menggunakan persamaan 2.1. Hasil perhitungan dapat dilihat pada tabel

Tabel 3.18 Jarak Data Uji terhadap Data Pada Kelompok Terpilih

| | d |
|---|---|
| D6 | 4 |
| D7 | 1 |
| D8 | 4 |
| D14 | 16.49242 |
| D15 | 17.08801 |

13. Mengurutkan data berdasarkan nilai *Euclidean Distance* mulai dari yang paling kecil ke yang besar. Hasil pengurutan data dapat dilihat pada table 3.19.

Tabel 3.19 Hasil Pengurutan Jarak *Euclidean Distance*

| | d |
|---|---|
| D7 | 1 |
| D6 | 4 |
| D8 | 4 |
| D14 | 16.49242 |
| D15 | 17.08801 |

14. Ambil data sebanyak nilai k yang diinputkan oleh user. Misalkan k = 3. Maka akan didapatkan hasil seperti table 3.20.

Tabel 3.20 Data dengan k=3

| | d |
|---|---|
| D7 | 1 |
| D6 | 4 |
| D8 | 4 |

15. Bandingkan kelas yang dimiliki oleh data pada no 8. Pada table 3.21 menunjukan kelas yang dimiliki masing – masing data.

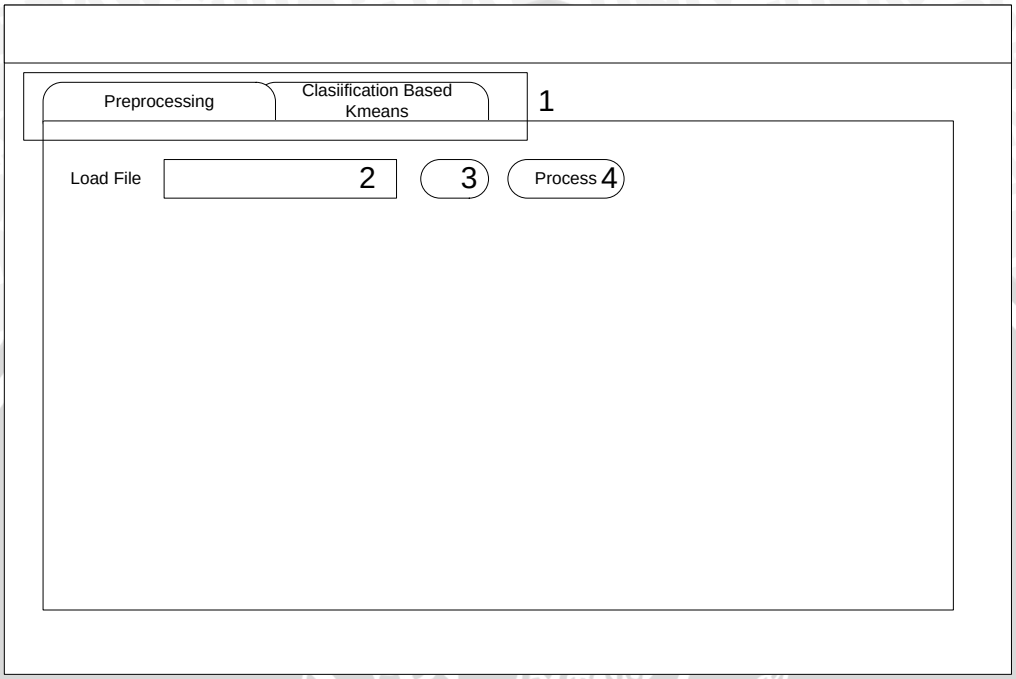
Tabel 3.21 Data dan Variabel Kelas yang Dimiliki

| Data | Kelas |
|---|---|
| D7 | 0 |
| D6 | 0 |
| D8 | 0 |

Didapatkan hasil sekuen protein sebagai inputan user tersebut masuk ke dalam kelas 0 atau termasuk *non cancer*.

### 3.3.7. Perancangan Antarmuka

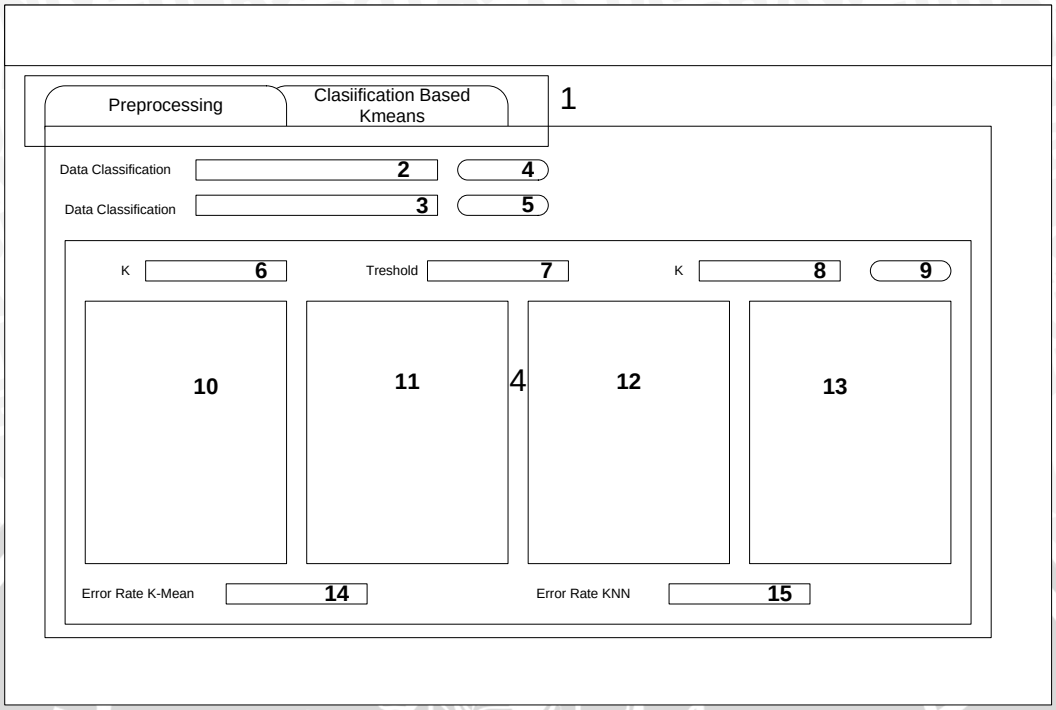
Antarmuka system (*interface*) untuk pengelompokan dan klasifikasi masing – masing ditunjukkan pada gambar 3.12 dan 3.13.




Gambar 3.12 Antarmuka Proses *Preprocessing*

Pada gambar 3.13, antarmuka system terdiri dari:

1. *Tabbed Pane* untuk navigasi memilih proses *preprocessing* atau *classification based Kmeans*.
2. *Text Field* untuk menampilkan nama *file* yang dipilih.
3. Tombol *Load* untuk menampilkan jendela *Load File* yang digunakan untuk memilih *file*.
4. Tombol *Process* untuk memproses proses *preprocessing*.




Gambar 3.13 Antarmuka Proses Utama

Pada gambar 3.14, antarmuka system terdiri dari:

1. *Tabbed Pane* untuk navigasi memilih proses *preprocessing* atau *classification based Kmeans*.
2. *Text Field* untuk menampilkan nama *file clustering* yang dipilih.
3. *Text Field* untuk menampilkan nama *file classification* yang dipilih.
4. Tombol *Load* untuk menampilkan jendela *Load File* yang digunakan untuk memilih *file clustering.*
5. Tombol *Load* untuk menampilkan jendela *Load File* yang digunakan untuk memilih *file classification.*
6. *Text Field* untuk memasukan nilai k yang diinginkan pada proses pengelompokan.
7. *Text Field* untuk memasukan nilai threshold yang diinginkan.
8. *Text Field* untuk memasukan nilai k yang diinginkan pada proses klasifikasi.
9. Tombol *Process* untuk memproses proses utama.

10. *Text Area* untuk menampilkan pohon yang terbentuk pada proses *clustering*.
11. *Text Area* untuk menampilkan hasil pengelompokan terakhir.
12. *Text Area* untuk menampilkan protein sejumlah k pada proses klasifikasi.
13. *Text Area* untuk menampilkan hasil klasifikasi data.
14. *Text Field* untuk menampilkan hasil *error ratio* proses pengelompokan.
15. *Text Field* untuk menampilkan hasil *error ratio* proses klasifikasi.

### 3.3.8. Perancangan Uji Coba

Setelah sistem selesai dibuat, selanjutnya adalah dilakukan uji coba terhadap system. Pengujian dilakukan untuk mengetahui tingkat keakurasian system dalam melakukan klasifikasi yang didasarkan pada kelompok yang sudah optimal.

Ada 3 jenis pengujian yang akan dilakukan pada penelitian ini, yaitu:

1. Pengujian untuk mengetahui pengaruh nilai k pada proses pengelompokan terhadap keoptimalan kelompok.
2. Pengujian untuk mengetahui pengaruh nilai k pada proses klasifikasi terhadap penentuan kelas.
3. Pengujian untuk mengetahui pengaruh variasi data terhadap keoptimalan kelompok.

#### 3.3.8.1. Uji Pengaruh Nilai k pada Pengelompokan Terhadap Keoptimalan Kelompok

Pengujian dan analisis hasil pengelompokan didasarkan pada kebenaran anggota yang dimiliki setiap kelompok memiliki tepat satu kelas. Pengujian dimulai dari k=4 hingga k=15 dengan threshold T=10, 15, 20, 25, 30, 35, 40, 45, 50, dan 55. Untuk perhitungannya dilakukan menggunakan persamaan 2.5

Tabel 3.22 Uji Pengaruh Nilai k Pada Proses Pengelompokan

| Threshold | k | Error Rate(%) |
|---|---|---|
| 10 | 4 | |
| | 5 | |
| | … | |
| | 15 | |
| 15 | 4 | |
| | 5 | |
| | … | |
| | 15 | |
| … | … | |
| 55 | 4 | |
| | 5 | |
| | … | |
| | 15 | |

Keterangan:

1. Treshold merupakan ambang batas jumlah anggota kelompok maksimal yang dimiliki setiap kelompok
2. K = nilai k yang akan diuji
3. Error Rate merupakan hasil perhitungan tingkat akurasi sesuai persamaan 2.5

### 3.3.8.2. Uji Pengaruh Nilai k pada Proses Klasifikasi Terhadap Penentuan Kelas

Pengujian dan analisis ini dilakukan pada proses klasifikasi. Pada pengujian ini, data latih yang digunakan merupakan data latih yang sudah dikelompokan terlebih dahulu secara optimal. Nilai k dibatasi hanya sampai 20.

Tabel 3.23 Uji Pengaruh Nilai k Pada Proses Klasifikasi

| Threshold | k K-Mean | k | Error Rate(%) |
|---|---|---|---|
| 10 | 4…15 | 1 | |
| | | … | |
| | | 20 | |
| 15 | 4…15 | 1 | |
| | | … | |
| | | 20 | |
| … | … | … | |
| 55 | 4…15 | 1 | |
| | | … | |
| | | 20 | |

Keterangan:

1. Treshold merupakan ambang batas jumlah anggota kelompok maksimal yang dimiliki setiap kelompok
2. K K-Mean merupakan nilai k paling optimal pada proses pengelompokan pada setiap threshold
3. K = nilai k yang akan diuji
4. Error Rate merupakan hasil perhitungan tingkat akurasi sesuai persamaan 2.5.

### 3.3.8.3. Uji Pengaruh Variasi Data Terhadap Keoptimalan Kelompok

Pengujian dan analisis ini dilakukan pada proses pengelompokan dengan menggunakan data yang diacak(variasi). Perlakuan pada pengujian ini sama dengan pengujian pertama, dengan T=10, 15, 20, 25, 30, 35, 40, 45, 50, 55 dan K=4 hingga K=15.

Table 3.24 Uji Pengaruh Variasi Data Terhadap Keoptimalan Kelompok

| Threshold | k | Error Rate(%) |
|---|---|---|
| 10 | 4 | |
| | 5 | |
| | … | |
| | 15 | |
| 15 | 4 | |
| | 5 | |
| | … | |
| | 15 | |
| … | … | |
| 55 | 4 | |
| | 5 | |
| | … | |
| | 15 | |

Keterangan:

1. Treshold merupakan ambang batas jumlah anggota kelompok maksimal yang dimiliki setiap kelompok
2. K = nilai k yang akan diuji
3. Error Rate merupakan hasil perhitungan tingkat akurasi sesuai persamaan 2.5

# BAB IV
# IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai implementasi sistem, pembahasan serta analisis dari hasil pengujian sistem. Implementasi sistem merupakan penerapan dari rancangan sistem yang telah dibahas pada bab sebelumnya.

## 4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalam sub bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

### 4.1.1. Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah:

1. Prosesor Intel(R) Core(TM)2 Duo CPU T5750 @2.00GHz
2. Memori 1.50 GB
3. VGA 1 GB
4. *Harddisk* dengan kapasitas 250 GB
5. Monitor 14”
6. *Keyboard*
7. *Mouse*

### 4.1.2. Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah:

1. Sistem Operasi *Microsoft Windows 7 Ultimate 32-bit*
2. *NetBeans IDE 7*
3. *Java(TM) SE versi 1.6.0_31*

## 4.2 Implementasi Program

Pada subbab ini akan dibahas implementasi perangkat lunak sesuai dengan analisis dan perancangan proses yang telah dipaparkan pada subbab 3.3.

### 4.2.1. Gambaran Singkat Program

Input pada system berupa file xml yang dipisahkan berdasarkan *tag* isi (berisi sekuen protein) dan *tag* kelas (berisi kelas kanker). File input terdiri dari dua macam, yaitu file data latih dan file transform. Selanjutnya dilakukan

*preprocessing* dengan cara membandingkan isi kedua file tersebut. Pada proses, ini file direpresentasikan dalam bentuk list. Output pada proses ini akan berbentuk xml dengan *tag* yang sama dengan file sebelumnya.

Proses selanjutnya file tersebut dikelompokan dengan menggunakan *K-Means*. Jika kelompok sudah memenuhi syarat maka akan dilakukan proses penentuan kelompok sebelum diterapkan proses KNN. Selanjutnya dilakukan proses KNN itu sendiri.

Adapun struktur data yang digunakan dalam program ini ditunjukan pada table 4.1.

Tabel 4.1 Struktur Data dalam Program

| No | Kelas | Struktur Data | Penjelasan |
|---|---|---|---|
| 1 | protein | String<br>Int | Representasi isi protein, kelas serta indeks yang dimilikinya. |
| 2 | HashMap2D | HashMap< > | Representasi PAM250. |
| 3 | XMLReader | ArrayList< > | Mengambil sekuen dalam protein |
| 4 | XMLWriter | ArrayList< ><br>HashMap< > | Representasi nilai protein dan penyimpanan dalam bentuk xml |
| 5 | Cluster | ArrayList< > | Representasi cluster pada proses pengelompokan |
| 6 | comparator | HashMap< > | Mengurutkan data pada proses KNN |
| 7 | Tree | Tree | Membangun pohon |
| 8 | KlasterKMean | Cluster[ ] | Representasi proses Kmean |
| 9 | KNN | ArrayList< ><br>Cluster[ ] | Representasi proses KNN |
| 10 | FResearch | JForm | Jendela Antarmuka Program |

### 4.2.2. Impelementasi Kelas Protein

Kelas protein merupakan struktur data yang merepresentasikan isi protein, kelas serta indeks yang dimiliki sekuen tersebut. Di dalam kelas ini terdapat method – method yang berfungsi untuk mengambil dan menyimpan isi protein dan kelas maupun method untuk membandingkan dua protein. Implementasi kelas Protein dapat dilihat pada *Source Code* 4.1.

```
package Data_Structure;

public class protein {
    protected String isi, kelas;
    private int index;

    public protein(int index) {
        this.index = index;
    }

    public int index(){
        return index;
    }

    public void setIsi(String isi){
        this.isi=isi;
    }

    public String getIsi(){
        return isi;
    }

    public void setKelas(String kelas){
        this.kelas = kelas;
    }

    public String getKelas(){
        return kelas;
    }
    public boolean equals(Object obj){
        protein pro = (protein) obj;
        return index == pro.index() && isi.equals(pro.isi) &&
kelas.equals(pro.kelas);
    }
}
```

Source Code 4.1 Implementasi Kelas Protein

### 4.2.3. Implementasi Kelas HashMap2D

Kelas HashMap2D merupakan kelas yang digunakan untuk membuat *array list* untuk menyimpan maupun mengambil nilai yang merepresentasikan matriks PAM250. Indeks pada *array list* ini merupakan gabungan pasangan string protein

dan isi dari *array list* merupakan nilai dari pasangan string protein tersebut. Sehingga pada akhirnya kelas ini dapat membentuk data menjadi seperti berbentuk tabel. Struktur data yang digunakan pada kelas ini adalah HashMap< >. Implementasi kelas HashMap2D dapat dilihat pada *Source Code* 4.2.

```
    public HashMap2D(){
        hashMap = new HashMap <Row_Key, HashMap<Column_Key,
Value>>();
    }
    public Value put(Row_Key key1, Column_Key key2, Value value){
        HashMap <Column_Key, Value> map;
        if(hashMap.containsKey(key1)){
            map=hashMap.get(key1);
        }
        else{
            map=new HashMap<Column_Key, Value>();
            hashMap.put(key1, map);
        }
        return map.put(key2, value);
    }
    public Value get(Row_Key key1, Column_Key key2){
        if (hashMap.containsKey(key1)){
            return hashMap.get(key1).get(key2);
        }
        else{
            return null;
        }
    }
    public Set<Column_Key> column_key(Row_Key key){
        return hashMap.get(key).keySet();
    }
```

Source Code 4.2 Implementasi Kelas HashMap2D

### 4.2.4. Implementasi Pembacaan File XML

XMLReader merupakan kelas untuk membaca file xml. Di dalam kelas ini terdapat *method* ***getXMList*** yang fungsinya untuk melakukan *parsing* data xml ke dalam string dengan mengidentifikasi element berdasarkan tag nya yaitu tag

<protein>. *Method* ***getTagValue*** digunakan untuk mengambil nilai yang dimiliki oleh element di dalam tag yang sudah ditentukan sebelumnya. Implementasi pembacaan file XML dapat dilihat pada *Source Code* 4.3.

```
public class XMLReader extends ArrayList<protein>{
    //membaca isi xml
    public XMLReader(String filename){
        NodeList nList = getXMList(filename);
        for (int temp = 0; temp < nList.getLength(); temp++) {
                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    protein p = new protein(temp);
                    p.setIsi(getTagValue("isi", eElement));
                    p.setKelas(getTagValue("kelas", eElement));
                    add(p);
                }
            }
    }
    //parsing data xml ke string
    public static NodeList getXMList(String filename){
        try {
            File fXmlFile = new File(filename);
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder =
dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
            doc.getDocumentElement().normalize();

            return doc.getElementsByTagName("protein");
        }
        catch(IOException ex){
            return null;
        }
        catch(SAXException xe){
            return null;
        }
```

```
        catch(ParserConfigurationException ex){
            return null;
        }
    }
    //mengambil value element xml
    public static String getTagValue(String sTag, Element
eElement) {
        NodeList nlList =
eElement.getElementsByTagName(sTag).item(0).getChildNodes();
        Node nValue = (Node) nlList.item(0);
        return nValue.getNodeValue();
    }
}
```

Source Code 4.3 Implementasi Pembacaan File XML

### 4.2.5. Implementasi Penyimpanan File XML

XMLWriter merupakan kelas yang digunakan untuk menulis hasil preproccesing program ke dalam bentuk file xml. Didalamnya terdapat method mengisi table hashMap, membandingkan protein serta menulis ke dalam file xml itu sendiri.

#### 4.2.5.1. Implementasi Method untuk Mengisi HashMap

Method ini digunakan untuk mengisi nilai pada HashMap dengan indeks berupa string dan nilai berupa numeric. Sedangkan mirrorInsert digunakan untuk menyamakan nilai posisi indeks string. Implementasi method untuk mengisi HashMap dapat dilihat pada *Source Code* 4.4.

```
private static void isiTabel(){
        mirrorInsert('C', 'C', 12);
        mirrorInsert('S', 'C', 0);
        mirrorInsert('S', 'S', 2);
        mirrorInsert('T', 'C', -2);
        mirrorInsert('T', 'S', 1);
        mirrorInsert('T', 'T', 3);
        mirrorInsert('P', 'C', -3);
        mirrorInsert('P', 'S', 1);
        mirrorInsert('P', 'T', 0);
```

```
        mirrorInsert('P', 'P', 6);
        mirrorInsert('A', 'C', -2);
        mirrorInsert('A', 'S', 1);
        mirrorInsert('A', 'T', 1);
        mirrorInsert('A', 'P', 1);
        }
//Memberi nilai yang sama pada pasangan string protein
    private static void mirrorInsert(char A, char B, int value){
        table.put(A, B, value);
        table.put(B, A, value);
    }
```

Source Code 4.4 Impelementasi Method untuk Mengisi HashMap

#### 4.2.5.2. Implementasi Method untuk Membandingkan Nilai Protein

Pada method ini terdapat fungsi yang akan mengambil string untuk setiap karakter dari string tersebut dibandingkan dan dipisahkan menggunakan “,” untuk hasilnya nanti. Implementasi method ini dapat dilihat pada *Source Code* 4.5.

```
//membandingkan protein
    private static String bandingProtein(String uji, String
pembanding){
        StringBuilder sb = new StringBuilder();
        sb.append(table.get(uji.charAt(0), pembanding.charAt(0)));
        for(int i = 1; i < uji.length(); i++)
            sb.append(",").append(table.get(uji.charAt(i),
pembanding.charAt(i)));
        return sb.toString();
    }
```

Source Code 4.5 Implementasi Method untuk Membandingkan Protein

#### 4.2.5.3. Implementasi Penulisan ke dalam Format XML

Method ini berfungsi untuk menulis hasil preprocessing ke dalam bentuk file xml. Di dalamnya akan membaca file sebagai file pembanding dan tag yang dibaca adalah tag isi. Outputnya akan berada di dalam element “hasil” dengan tag yang sama seperti file sebelumnya. Implementasi method ini dapat dilihat pada *Source Code* 4.6.

```
//menulis file xml hasil
    public static void write (ArrayList<protein> list, String
namaFile){
        isiTabel();
        try {
            NodeList nodeList = XMLReader.getXMList("data
transform.xml");
            Element element = (Element) nodeList.item(0);
            String transform = XMLReader.getTagValue("isi",
element);

            DocumentBuilderFactory docFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder =
docFactory.newDocumentBuilder();
            File fileOut = new File(namaFile);

            FileWriter write = new FileWriter(fileOut);
            // root elements
            Document doc = docBuilder.newDocument();
            Element rootElement = doc.createElement("hasil");
            doc.appendChild(rootElement);
            for(int i = 0; i < list.size(); i++){
                Element pro = doc.createElement("protein");
                rootElement.appendChild(pro);
                protein p = list.get(i);
                String hasil = bandingProtein(p.getIsi(),
transform);

                Element firstname = doc.createElement("isi");
                firstname.appendChild(doc.createTextNode(hasil));
                pro.appendChild(firstname);

                Element lastname = doc.createElement("kelas");

lastname.appendChild(doc.createTextNode(p.getKelas()));
                pro.appendChild(lastname);
            }
```

```
            TransformerFactory transformerFactory =
TransformerFactory.newInstance();
            Transformer transformer =
transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);
            StreamResult result = new StreamResult(write);

            transformer.transform(source, result);

            // write the content into xml file
        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (TransformerException tfe) {
            tfe.printStackTrace();
        } catch(IOException ex){}
    }
```

Source Code 4.6 Implementasi Penulisan ke dalam File XML

### 4.2.6. Implementasi Kelas Cluster

Pada kelas Cluster ini terdapat beberapa method yang digunakan untuk mengatur isi klaster sekaligus menghitung jarak *Euclidean*, menghitung centroid baru dan menghitung *error rate* pada proses pengelompokan.

#### 4.2.6.1. Method untuk Mengatur Isi Cluster

Di dalam method ini memuat perintah untuk membaca dan menulis isi, kelas maupun indeks yang dimiliki oleh ArrayList<protein>. Implementasi method ini dapat dilihat pada *Source Code* 4.7.

```
public Cluster(String isi, String kelas, int index){
        super();
        this.isi = isi;
        this.kelas = kelas;
        this.index = index;
    }
    public Cluster(protein pro){
        this(pro.getIsi(), pro.getKelas(), pro.index());
    }
    public int index(){
        return index;
    }
    public void setIsi(String isi){
        this.isi=isi;
    }
    public String getIsi(){
        return isi;
    }
    public void setKelas(String kelas){
        this.kelas = kelas;
    }
    public String getKelas(){
        return kelas;
    }
```

Source Code 4.7 Implementasi Method untuk Mengatur Isi Cluster

#### 4.2.6.2. Implementasi Perhitungan Jarak *Euclidean*

Method ini mengambil nilai protein dengan membaca setiap karakter yang dipisahkan oleh ",". Inisialisasi awal dari nilai jarak adalah 0, setelah itu dilakukan perulangan sebanyak protein yang ada. Untuk menghitungnya menggunakan rumus *Euclidean Distance*. Implementasi method ini dapat dilihat pada *Source Code* 4.8.

```
//hitung euclidean distance
    public double compare(protein pro){
        String[] isiProtein = pro.getIsi().split(","),
                 isiCluster = isi.split(",");
        double sum = 0, n = isiProtein.length;
        for(int i = 0; i < n; i++)
            sum += Math.pow(Double.parseDouble(isiProtein[i])-
Double.parseDouble(isiCluster[i]), 2);
        return Math.sqrt(sum);
    }
```

Source Code 4.8 Implementasi Perhitungan Jarak *Euclidean*

#### 4.2.6.3. Implementasi Perhitungan Centroid Baru

Method ini berfungsi untuk menghitung centroid baru dengan cara menghitung rata – rata jarak antar data di dalam setiap kelompok yang sudah terbentuk dengan kondisi kelompok tidak boleh kosong. Implementasi method ini dapat dilihat pada *Source Code* 4.9.

```
//hitung centroid baru
    public void average(){
        if(!isEmpty()){
            String[][] isiProtein = new String[size()][];
            for (int i = 0; i < size(); i++)
                isiProtein[i] = get(i).getIsi().split(",");
            StringBuilder result = new StringBuilder();
            for (int i = 0; i < isiProtein[0].length; i++){
                double sum = 0;
                for(int j = 0; j < size(); j++)
                    sum += Double.parseDouble(isiProtein[j][i]);
                result.append((sum/(double)size())).append(",");
            }
            isi = result.toString().substring(0, result.length()-
1);
        }
    }
```

Source Code 4.9 Implementasi Perhitungan Centroid Baru

#### 4.2.6.4. Implementasi Perhitungan *Error Rate* Proses KNN

Implementasi method ini dapat dilihat pada *Source Code* 4.10.

```
public double errorRate(){
        int[] kls = new int[4];
        for(int i = 0; i < size(); i++)
            kls[Integer.parseInt(get(i).getKelas())]++;
        double max = Double.MIN_VALUE;
        int idx = 0, hitung = 0;
        for(int i = 0; i < kls.length; i++){
            if(max < kls[i]){
                max = kls[i];
                idx = i;
            }
        }
        for(int i = 0; i < size(); i++)
            if (!get(i).getKelas().equals(String.valueOf(idx)))
                hitung++;
        return hitung / (double) size();
    }
```

Source Code 4.10 Implementasi *Error Rate* Proses Pengelompokan

### 4.2.7. Implementasi Method Pengurutan Data Pada Proses KNN

Method ini berada pada kelas comparator yang digunakan untuk mengurutkan data dengan menggunakan HashMap. Data diperlakukan sebagai object pada kelas ini lalu dibandingkan. Implementasi method ini dapat dilihat pada *Source Code* 4.11.

```
public class comparator implements Comparator {
    HashMap urut;

    public comparator(HashMap urut){
        this.urut = urut;
    }
    @Override
    public int compare(Object a,Object b){
        if((Double)urut.get(a) < (Double)urut.get(b)) {
        return 1;
        } else if((Double)urut.get(a) == (Double)urut.get(b)) {
            return 0;
            } else {
                return -1;
                }
        }
}
```

Source Code 4.11 Implementasi Pengurutan Data

### 4.2.8. Implementasi Pembentukan *Tree*

Pembentukan *tree* menggunakan sebuah library yang disediakan oleh java dan disimpan dalam kelas *tree* dan diimplementasikan pada kelas utama. Method ini digunakan pada proses pengelompokan dengan node *parent* sebagai node induk dan node *child* sebagai *leaf* nya. Untuk pembentukan node child mengikuti syarat yang ada pada proses pengelompokan yaitu jika jumlah data pada sebuah node parent melebihi batas threshold atau kurang dari jumlah k, maka node *parent* tersebut dianggap sebagai node *child*.

```
//pembentukan pohon
    private String pembentukan_tree(DefaultMutableTreeNode node,
KlasterKMean kmean){
        StringBuilder sb = new StringBuilder();
        Cluster[] clusters = kmean.getClusters();
        for(int i = 0; i < clusters.length; i++){
            String parent = node.getUserObject().toString();
            if(!parent.isEmpty()) parent += "." + i;
            else parent = String.valueOf(i);
            DefaultMutableTreeNode child = new
DefaultMutableTreeNode(parent);
            node.add(child);
            Cluster cluster = clusters[i];
            if(cluster.isEmpty()){
                String str = parent+" - Empty Cluster";
                child.setUserObject(str);
                sb.append("\n---").append(str).append("---\n");
            }
            else if(cluster.size() < threshold || cluster.size() <
jumlahKelas){
                sb.append(baca_child(child, cluster));
            }
            else{
                KlasterKMean _kmean = new KlasterKMean(cluster,
jumlahKelas, iterasi);
                if(!kmean.cekKlaster(_kmean))
                    sb.append(pembentukan_tree(child, _kmean));
                else{
                    sb.append(baca_child(child, cluster));
                }
            }
        }
        return sb.toString();
    }
```

Source Code 4.12 Implementasi Pembentukan *Tree*

### 4.2.9. Implementasi Pembacaan *Node Child*

Di dalam metode ini berisi cara membaca node anak yaitu dengan mengambil indeks dari kelompok yang terakhir terbentuk. Selain itu terdapat perhitungan mencari rata – rata nilai tengah data pada setiap kelompok terakhir(node anak) yang terbentuk. Implementasi method ini dapat dilihat pada *Source Code* 4.13

```
private String baca_child(DefaultMutableTreeNode node, Cluster
cluster){
        error_rate += cluster.errorRate();
        jmlKnn++;
        StringBuilder sb = new StringBuilder("\n---
"+node.getUserObject()+"---");
        for(protein i : cluster){
          DefaultMutableTreeNode child = new
DefaultMutableTreeNode("protein "+i.index());
           node.add(child);
           sb.append(i.toString()).append("\n");
        }
        cluster.average();
        forKNN.add(cluster);
        return sb.toString();
    }
```

Source Code 4.13 Implementasi Pembacaan *Node Child*

### 4.2.10. Implementasi Pencarian Kelompok Terdekat

Untuk mencari kelompok terdekat dengan data uji adalah dengan membandingkan data uji ke masing – masing nilai tengah (centroid) yang dimiliki setiap kelompok. Kelompok dengan nilai terkecil akan disimpan untuk nantinya digunakan pada proses KNN. Implementasi method ini dapat dilihat pada *Source Code* 4.14.

```
private Cluster getClosestCluster(protein uji){
        double max = Double.MAX_VALUE;
        Cluster result = null;
        for (Cluster object : forKNN) {
            double avg = object.compare(uji);
            if(avg < max){
                max = avg;
                result = object;
            }
        }
        return result;
    }
```

Source Code 4.14 Implementasi Pencarian Kelompok Terdekat

### 4.2.11. Implementasi Proses Pengelompokan Menggunakan *K-Mean*

Kelas ini merupakan kelas utama yang menangani proses pengelompokan menggunakan k-means. Prosesnya terdiri dari menentukan centroid awal, mengisi anggota kelompok, update centroid, cek centroid, cek anggota kelas, dan proses K-Mean itu sendiri.

#### 4.2.11.1. Implementasi Penentuan Centroid Awal

Pada method ini centroid awal diambil dari indeks 0 (data pertama). Setelah itu dilakukan pengecekan kelas pada data selanjutnya sama atau tidak dengan kelas pada data yang sudah dijadikan centroid. Jika tidak sama, maka indeks data akan disimpan sebagai centroid. Perulangan dilakukan sebanyak input k(kelompok yang ingin dibentuk). Dengan catatan, data dengan centroid yang sudah terambil tidak akan terbaca lagi pada proses ini. Jika pada akhir *array*, jumlah k masih belum terpenuhi, maka proses berulang dari awal terhadap sisa data. Implementasi method ini dapat dilihat pada *Source Code* 4.15.

```
//menentukan centroid awal
    private void initCentroid(){
        ArrayList<protein> N = new ArrayList<protein>();
        clusters[kelas_terisi] = new Cluster(copy.get(0));
        kelas_terisi++;
        N.add(copy.get(0));
        for (int i = 1; i < copy.size(); i++) {
            protein A = copy.get(i), B = copy.get(i-1);
            if(kelas_terisi >= jumlahKelas)
                    break;
            else if(!A.getKelas().equals(B.getKelas())){
                clusters[kelas_terisi] = new Cluster(A);
                N.add(A);
                kelas_terisi++;
            }
        }
        copy.removeAll(N);
        if(kelas_terisi < jumlahKelas)
            initCentroid();
    }
```

Source Code 4.15 Implementasi Penentuan Centroid Awal

#### 4.2.11.2. Implementasi Penentuan Anggota Kelompok

Untuk menentukan setiap anggota kelompok diambil dari jarak terdekat data terhadap centroid yang ada pada tiap kelompoknya. Pada setiap data dilakukan perbandingan dengan setiap centroid yang ada. Proses ini menggunakan method *compare* yang merupakan method untuk menghitung jarak *Euclidean* yang terdapat pada kelas *Cluster*. Implementasi method ini dapat dilihat pada *Source Code* 4.16.

```
//assign anggota cluster
    private void isiCluster(){
        for(int i=0;i<clusters.length;i++)
            clusters[i].clear();
        for(int i = 0; i < parent.size(); i++){
            double temp = Double.MAX_VALUE;
            int index = 0;
            protein pro = parent.get(i);
            for(int j = 0; j < clusters.length; j++){
                double comp = clusters[j].compare(pro);
                if(comp < temp){
                    temp = comp;
                    index = j;
                }
            }
            clusters[index].add(pro);
        }
    }
```

Source Code 4.16 Implementasi Penentuan Anggota Kelompok

#### 4.2.11.3. Implementasi *Update Centroid*

Pada method ini digunakan System.arraycopy yang berfungsi untuk menyalin isi kelompok ke dalam arraylist baru. Untuk menghitung centroid baru dilakukan perhitungan *average* yang merupakan method dari kelas *cluster*. Impelementasi method ini dapat dilihat pada *Source Code* 4.17.

```
//update centroid
    public void updateCentroid(){
        cluster_awal=new Cluster[jumlahKelas];
        System.arraycopy(clusters, 0, cluster_awal, 0,
jumlahKelas);
        for(int j = 0; j < clusters.length; j++)
            clusters[j].average();
    }
```

Source Code 4.17 Implementasi *Update Centroid*

#### *4.2.11.4.* Implementasi Pengecekan *Centroid*

Pada method ini dilakukan pengecekan centroid dengan membandingkan centroid awal dengan centroid yang sudah terupdate pada proses sebelumnya. Centroid sebelumnya disimpan pada sebuah *arraylist* lalu dibandingkan dengan centroid batu yang terbentuk. Implementasi method ini dapat dilihat pada *Source Code* 4.18.

```
//cek centroid
    private boolean isCentroidSama(){
        boolean same = true;
        for(int j = 0; j < clusters.length; j++)
if(!clusters[j].getIsi().equals(cluster_awal[j].getIsi())){
                same = false;
                break;
            }
        return same;
    }
```

Source Code 4.18 Implementasi Pengecekan Centroid

#### 4.2.11.5. Implementasi Proses *K-Mean*

Method ini berisi proses yang dikerjakan oleh k-means dengan memanggil method – method lainnya. Awal proses adalah memasukan anggota kelompok awal yaitu semua data, lalu dilakukan *update centroid*. Setelah itu selama iterasi masih seusai iterasi yang ditetapkan maka proses berlanjut dengan mengisi anggota kelompok yang baru. Proses Selanjutnya adalah pengecekan centroid. Jika tidak memenuhi syarat, maka centroid akan di *update* kembali. Implementasi method ini dapat dilihat pada *Source Code* 4.19.

```
//proses k-mean
    private void process(){
        detail.append("----------ITERASI 0----------\n");
        isiCluster();
        updateCentroid();
        detail.append(lastResult());
        if(iterasi > 1){
            for (int j = 1; j < iterasi; j++) {
                detail.append("----------ITERASI
").append(j).append("----------\n");
                isiCluster();
                if(isCentroidSama())
                    break;
                updateCentroid();
                detail.append(lastResult());
            }
        }
    }
```

Source Code 4.19 Implementasi Proses *K-Means*

### 4.2.12. Implementasi Proses Klasifikasi Menggunakan KNN

Kelas ini merupakan kelas yang menangani proses klasifikasi menggunakan KNN. Di dalam proses ini terdapat method untuk menghitung jarak data uji dengan data latih, *sorting*, dan membandingkan kelas.

#### 4.2.12.1. Implementasi Perhitungan Jarak

Perhitungan jarak data uji ke data latih yang sudah dikelompokan menggunakan *Euclidean Distance*. Setelah mendapatkan jarak tersebut, dilakukan pencarian jarak data terkecil dengan menggunakan HashMap. Implementasi Method ini dapat dilihat pada *Source Code* 4.20.

```
private double euclid(protein pro){
        String[] ujiStrings = uji.getIsi().split(","), isiProtein
= pro.getIsi().split(",");
        double jarak = 0, n = isiProtein.length;
        for(int i = 0; i < n; i++){
            jarak += Math.pow(Double.parseDouble(ujiStrings[i]) -
Double.parseDouble(isiProtein[i]), 2);
        }return Math.sqrt(jarak);
    }
   private HashMap<Integer, Double> cari_jarak(){
        HashMap<Integer, Double> map = new HashMap<Integer,
Double>();
        for(protein pro : cluster){
            double hasil = euclid(pro);
            map.put(pro.index(), hasil);
        }
        return map;
    }
```

Source Code 4.20 Implementasi Perhitungan Jarak

#### 4.2.12.2. Implementasi *Sorting*

Method sorting ini menggunakan struktur data TreeMap dengan membuat object dari kelas comparator. Implementasi method ini dapat dilihat pada *Source Code* 4.21.

```
private TreeMap<Integer,Double> sorting(HashMap<Integer, Double>
map){
        comparator bvc =  new comparator(map);
        TreeMap <Integer,Double> sorted_map = new TreeMap(bvc);
        sorted_map.putAll(map);
        return sorted_map;
    }
```

Source Code 4.21 Implementasi *Sorting*

#### 4.2.12.3. Implementasi Membandingkan Kelas

Method ini menjalankan perintah untuk membandingkan atribut kelas yang dimiliki oleh data. Data dibandingkan dengan menggunakan HashMap. Data yang

akan dibandingkan diambil mulai dari indeks 0 hingga input k. Implementasi method ini dapat dilihat pada *Source Code* 4.22.

```
private String bandingKelas(Object[] indexes){

        HashMap<Integer, Double> map = new HashMap<Integer,
Double>();
        if(cluster.size() < k){
            int kls =
Integer.parseInt(list.get((Integer)indexes[indexes.length-
1]).getKelas());
            if(map.get(kls) != null){
                Double value = map.get(kls);
                value += 1;
                map.put(kls, value);
            }
            else
                map.put(kls, 1.0);
        }else
            for(int m=0;m<k;m++){
                int kls =
Integer.parseInt(list.get((Integer)indexes[indexes.length-m-
1]).getKelas());
                if(map.get(kls) != null){
                    Double value = map.get(kls);
                    value += 1;
                    map.put(kls, value);
                }
                else
                    map.put(kls, 1.0);
            }
        TreeMap<Integer,Double> treemap = sorting(map);
        return treemap.keySet().toArray()[0].toString();
    }
```

Source Code 4.22 Implementasi Membandingkan Kelas

### 4.2.13. Implementasi Proses Utama

Pada proses utama diawali dengan pembacaan file XML yang kemudian dilakukan proses pengelompokan menggunakan *K-Mean*. Jika syarat sudah terpenuhi(kelompok termasuk dalam *Node Child*) maka akan dilakukan proses klasifikasi menggunakan KNN. Setelah semua *Node Child* terbentuk maka dilakukan proses perhitungan nilai *error rate* pengelompokan.

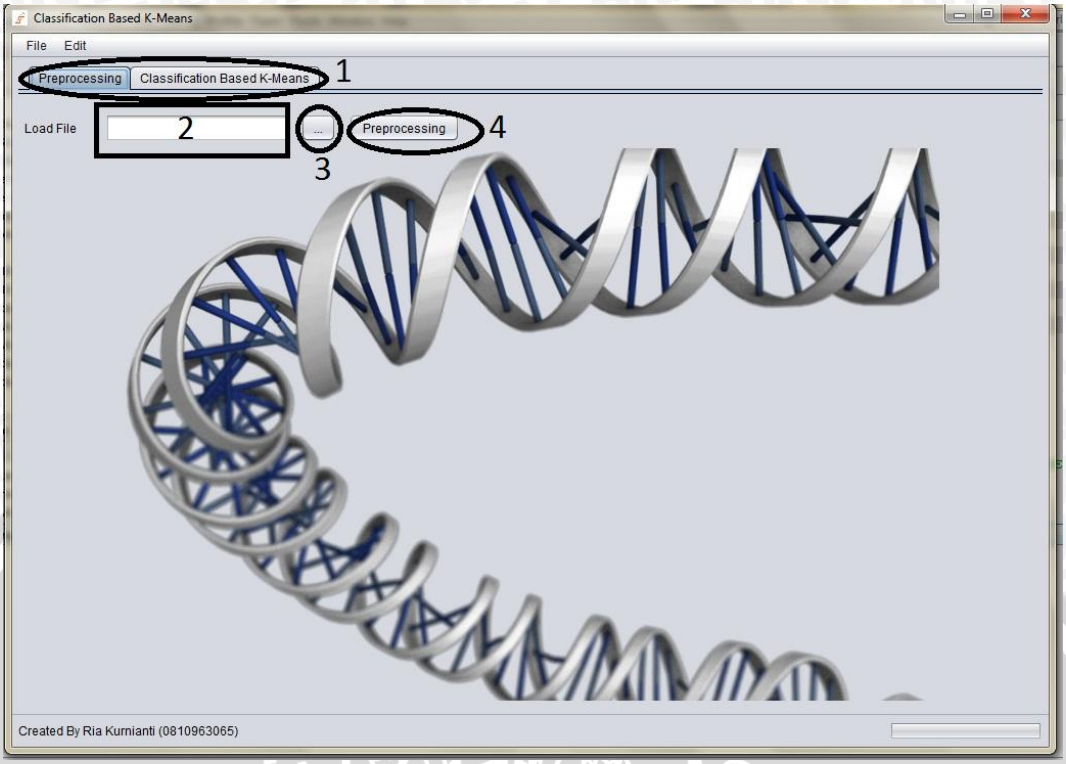
```
private void BProsesActionPerformed(java.awt.event.ActionEvent
evt) {
        try{
            jumlahKelas =
Integer.parseInt(txtK_mean.getText().toString());
            threshold =
Integer.parseInt(txtTreshold.getText().toString());
            int k = Integer.parseInt(txtKnn.getText().toString());
            ArrayList<protein> list = new
XMLReader(txtLoad1.getText());
            ArrayList<protein> list2 = new
XMLReader(txtLoad2.getText());
            if(list.size() < jumlahKelas){
                JOptionPane.showMessageDialog(null,"Input is too
large","Error", JOptionPane.ERROR_MESSAGE);
            }
            else{
                Cluster kelompok = new Cluster(list.get(0));
                for (protein pro : list) {
                    kelompok.add(pro);
                }
root = new DefaultMutableTreeNode("");
                tree = new Tree(root);
                KlasterKMean kmean = new
KlasterKMean(kelompok ,jumlahKelas, iterasi);
                forKNN = new ArrayList<Cluster>();
                error_rate = 0;
                jmlKnn = 0;
```

```
                txtIterasi.setText(pembentukan_tree(root, kmean));
                root.setUserObject("root");
                pane.setViewportView(tree);
                int error_knn = 0;
                String result = "", kelas = "";
for(int l=0;l<list2.size();l++){
                    protein uji = list2.get(l);
                    Cluster akhir = getClosestCluster(uji);
                    KNN knn = new KNN(list, akhir, uji, k);

                    result += "Data " + (l+1) + "\n" +
knn.sortingResult + "\n";
                    kelas += "Protein " + (l+1) + " = " +
getType(knn.kelas) +"\n";

                    if(!uji.getKelas().equals(knn.kelas))
                        error_knn++;
                }
                txtKdata1.setText(result);
                txtType1.setText(kelas);
                txtError.setText(error_rate *
100/(double)jmlKnn+"");
                txtErrorKnn.setText(error_knn * 100/
(double)list2.size()+"");
            }
        }
        catch(NumberFormatException ex){
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Format input
salah atau jumlah kelas yang dipilih melebihi
batas","Error",JOptionPane.ERROR_MESSAGE);
        }
    }
```

Source Code 4.23 Implementasi Proses Utama

### 4.2.14. Tampilan Antarmuka Program

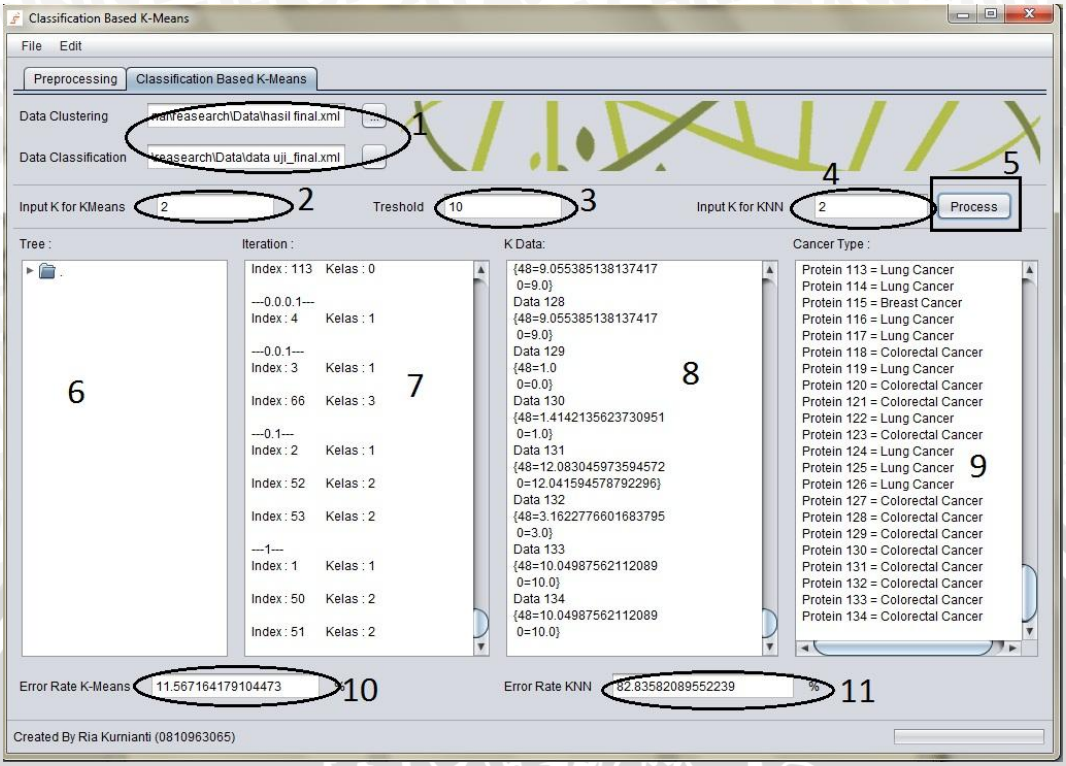
Berdasarkan rancangan antarmuka yang telah dijabarkan pada bab 3, terdapat dua tab yang digunakan pada antarmuka program, yaitu tab untuk preprocessing dan tab untuk proses klasifikasi berdasarkan pengelompokan.




Gambar 4.1 Tampilan Antarmuka *Tab Preprocessing*

Keterangan:

1. Pada tab *Preprocessing* dan *Classification Based K-Mean* untuk memilih proses yang akan dijalankan
2. *Text Field* untuk menampilkan nama file yang terpilih
3. *Button load* yang digunakan untuk memilih file yang akan digunakan.
4. *Button Preprocessing* digunakan untuk memulai proses *preprocessing*.




Gambar 4.2 Tampilan Antarmuka Proses Utama

Keterangan:

1. *Textfield* dan *Button load* yang digunakan untuk memilih file yang akan digunakan sebagai data pengelompokan (data latih) dan data yang akan di klasifikasikan (data uji).
2. Textfield untuk memasukan inputan nilai K yang akan digunakan pada pengelompokan K-Means.
3. *Textfield* untuk memasukan inputan nilai *threshold* yaitu batasan jumlah anggota pada setiap kelompok.
4. *Textfield* untuk memasukan inputan nilai K yang akan digunakan pada klasifikasi KNN.
5. *Button Process* untuk memulai proses klasifikasi berdasarkan pengelompokan K-Means.
6. *Textarea* untuk menampilkan hasil *Tree* yang terbentuk ketika proses pengelompokan.
7. *Textarea* untuk menampilkan hasil pengelompokan pada setiap iterasinya.
8. *Textarea* untuk menampilkan hasil seleksi data pada proses KNN.

9. *Textarea* untuk menampilkan hasil klasifikasi
10. *Textfield* untuk menampilkan hasil *error rate* pada proses pengelompokan.
11. *Textfield* untuk menampilkan hasil *error rate* pada proses klasifikasi

### 4.3. Implementasi Uji Coba

Pada Subbab ini akan dibahas tentang implementasi dari metode pengujian yang telah dilakukan oleh system dan hasil dari pengujian itu sendiri. Untuk pengujian pengaruh nilai k pada proses pengelompokan digunakan 848 data, sedangkan untuk pengujian pengaruh nilai k pada proses klasifikasi digunakan 135 data.

#### 4.3.1. Contoh Sekuen Protein yang Digunakan

```
<protein>
<isi>MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPD
EAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLAPPQH
LIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILTIITLED
SSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSPQPKKKPLDG
EYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPD
SD</isi>
<kelas>0</kelas>
</protein>

<protein>
<isi>MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPD
EAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMSEVVRRCPHHERCSDSDGLAPPQH
LIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILTIITLED
SSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSPQPKKKPLDG
EYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPD
SD</isi>
<kelas>1</kelas>
</protein>
```

### 4.3.2. Contoh Sekuen Protein yang Sudah Ditransformasi

```
<protein>
<isi>6,4,4,6,4,2,4,6,2,4,4,6,6,6,2,4,4,3,9,2,4,6,17,5,6,6,6,4,2,2,
4,6,2,6,6,6,2,4,2,6,4,4,6,6,6,2,6,4,4,5,4,4,17,9,3,4,4,6,5,6,4,4,2
,6,6,6,6,4,2,2,6,6,4,2,6,2,6,2,2,6,3,6,2,2,6,2,6,2,6,2,17,6,6,2,2,
2,4,6,2,4,5,3,10,4,5,2,10,5,9,6,6,5,9,6,6,2,5,3,2,5,2,4,3,12,3,10,
2,6,2,6,2,5,6,9,12,4,6,2,5,3,12,6,4,4,6,17,4,4,2,3,6,6,6,5,3,6,4,6
,2,6,2,5,10,5,4,2,4,6,6,3,4,4,4,6,6,12,6,6,6,4,6,12,2,4,2,4,5,6,2,
6,6,4,6,6,5,6,4,4,5,2,6,6,4,4,10,6,4,4,6,2,3,9,6,6,2,4,4,4,6,10,4,
6,6,4,4,5,2,4,12,3,3,5,6,10,2,10,6,12,2,2,2,12,6,5,5,6,2,6,6,6,5,6
,3,5,5,3,6,4,4,2,2,5,2,6,6,5,6,2,2,9,4,4,6,4,12,2,12,6,5,6,4,6,6,3
,4,4,4,2,6,2,5,5,5,4,6,6,6,4,6,6,6,5,2,3,5,6,2,6,6,2,2,3,2,2,2,6,4
,6,5,5,5,6,6,4,5,4,10,9,3,6,4,5,6,5,6,4,6,9,4,6,9,6,4,6,2,4,2,6,4,
6,5,4,2,4,2,5,5,4,6,5,5,2,6,2,6,2,2,6,6,5,2,5,5,5,4,2,3,2,6,6,5,5,
6,6,9,5,3,4,5,6,4,2,4</isi>
<kelas>1</kelas>
</protein>

<protein>
<isi>6,4,4,6,4,2,4,6,2,4,4,6,6,6,2,4,4,3,9,2,4,6,17,5,6,6,6,4,2,2,
4,6,2,6,6,6,2,4,2,6,4,4,6,6,6,2,6,4,4,5,4,4,17,9,3,4,4,6,5,6,4,4,2
,6,6,6,6,4,2,2,6,6,4,2,6,2,6,2,2,6,3,6,2,2,6,2,6,2,6,2,17,6,6,2,2,
2,4,6,2,4,5,3,10,4,5,2,10,5,9,6,6,5,9,6,6,2,5,3,2,5,2,4,3,12,3,10,
2,6,2,6,2,5,6,9,12,4,6,2,5,3,12,6,4,4,6,17,4,4,2,3,6,6,6,5,3,6,4,6
,2,6,2,5,10,5,4,2,4,6,6,3,4,4,4,6,6,12,6,6,6,4,6,12,2,4,2,4,5,6,2,
6,6,4,6,6,5,6,4,4,5,2,6,6,4,4,10,6,4,4,6,2,3,9,6,6,2,4,4,4,6,10,4,
6,6,4,4,5,2,4,12,3,3,5,6,10,2,10,6,12,2,2,2,12,6,5,5,6,2,6,6,6,5,6
,3,5,5,3,6,4,4,2,2,5,2,6,6,5,6,2,2,9,4,4,6,4,12,2,12,6,5,6,4,6,6,3
,4,4,4,2,6,-
3,5,5,5,4,6,6,6,4,6,6,6,5,2,3,5,6,2,6,6,2,2,3,2,2,2,6,4,6,5,5,5,6,
6,4,5,4,10,9,3,6,4,5,6,5,6,4,6,9,4,6,9,6,4,6,2,4,2,6,4,6,5,4,2,4,2
,5,5,4,6,5,5,2,6,2,6,2,2,6,6,5,2,5,5,5,4,2,3,2,6,6,5,5,6,6,9,5,3,4
,5,6,4,2,4</isi>
<kelas>1</kelas>
</protein>
```

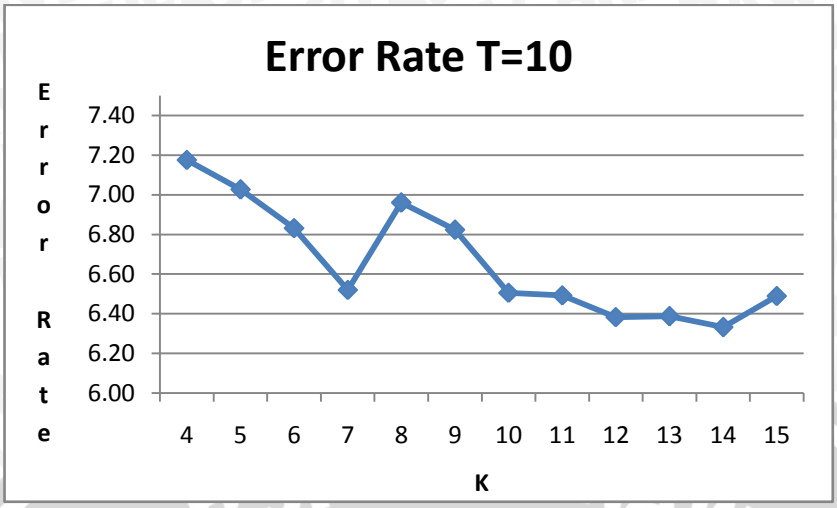
### 4.3.3. Uji Pengaruh Nilai k Pada Proses Pengelompokan Terhadap Keoptimalan Kelompok

Pengujian yang pertama adalah mengetahui pengaruh nilai k pada proses pengelompokan terhadap keoptimalan kelompok. Pengujian ini melibatkan data latih sebanyak 848 data. Untuk pengujian, digunakan threshold 10, 15, 20, 25, 30, 35, 40, 45, 50 dan 55 yang diuji dengan nilai k mulai dari 4 hingga 15. Hasil pengujian untuk threshold 10 ditampilkan pada table 4.2.

Tabel 4.2 Tabel Hasil Uji Pengelompokan dengan T=10

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 10 | 4 | 7.18 |
| | 5 | 7.03 |
| | 6 | 6.83 |
| | 7 | 6.52 |
| | 8 | 6.96 |
| | 9 | 6.82 |
| | 10 | 6.51 |
| | 11 | 6.49 |
| | 12 | 6.38 |
| | 13 | 6.39 |
| | 14 | 6.33 |
| | 15 | 6.49 |

Berdasarkan Tabel 4.2 nilai error rate yang didapatkan berkisar antara 6% hingga 7%. Untuk nilai error ratio K=4 adalah 7.18%. Untuk K= 5 didapatkan error ratio sebesar 7.03%. Untuk K=6 didapatkan error ratio sebesar 6.83%. Untuk K=7 didapatkan error ratio sebesar 6.52%. Untuk K=8 didapatkan error ratio sebesar 6.96%. Untuk K=9 didapatkan error ratio sebesar 6.82%. Untuk K=10 didapatkan error ratio sebesar 6.51%. Untuk K=11 didapatkan error ratio sebesar 6.49%. Untuk K=12 didapatkan error ratio sebesar 6.38%. Untuk K=13 didapatkan error ratio sebesar 6.39%. Untuk K=14 didapatkan error ratio sebesar 6.33%. Untuk K=15 didapatkan error ratio sebesar 6.49%.




Gambar 4.3 Grafik Hubungan k dengan *Error Rate untuk T=10*

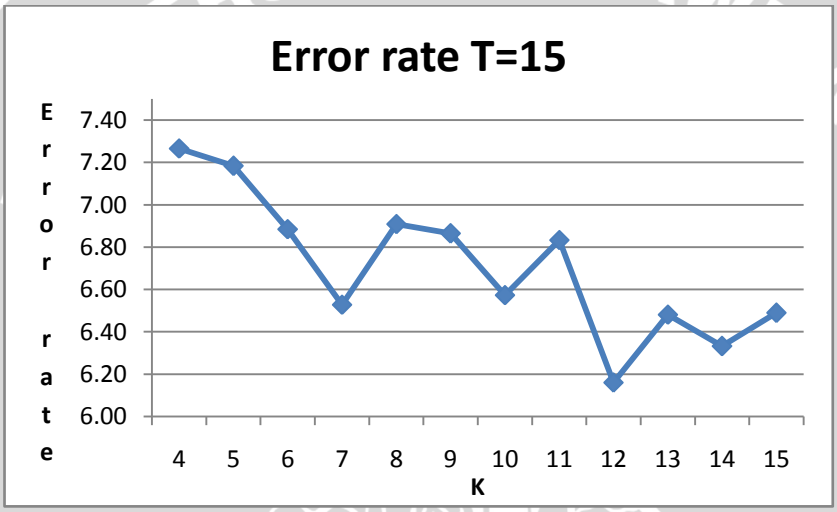
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.18%, sedangkan nilai error terkecil didapatkan saat K=14, yaitu sebesar 6.33%.

Selanjutnya untuk hasil pengujian dengan threshold = 15 ditampilkan pada tabel 4.3.

Tabel 4.3 Hasil Uji Pengelompokan dengan T=15

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 15 | 4 | 7.27 |
| | 5 | 7.18 |
| | 6 | 6.89 |
| | 7 | 6.53 |
| | 8 | 6.91 |
| | 9 | 6.87 |
| | 10 | 6.57 |
| | 11 | 6.83 |
| | 12 | 6.16 |
| | 13 | 6.48 |
| | 14 | 6.33 |
| | 15 | 6.49 |

Berdasarkan tabel 4.3 nilai error rate yang didapatkan berkisar antara 6% hingga 7%. Untuk nilai error ratio K=4 adalah 7.27%. Untuk K= 5 didapatkan error ratio sebesar 7.18%. Untuk K=6 didapatkan error ratio sebesar 6.89%. Untuk K=7 didapatkan error ratio sebesar 6.53%. Untuk K=8 didapatkan error ratio sebesar 6.91%. Untuk K=9 didapatkan error ratio sebesar 6.87%. Untuk K=10 didapatkan error ratio sebesar 6.57%. Untuk K=11 didapatkan error ratio sebesar 6.83%. Untuk K=12 didapatkan error ratio sebesar 6.16%. Untuk K=13 didapatkan error ratio sebesar 6.48%. Untuk K=14 didapatkan error ratio sebesar 6.33%. Untuk K=15 didapatkan error ratio sebesar 6.49%.




Gambar 4.4 Grafik Hubungan k dengan *Error Rate* untuk T=15

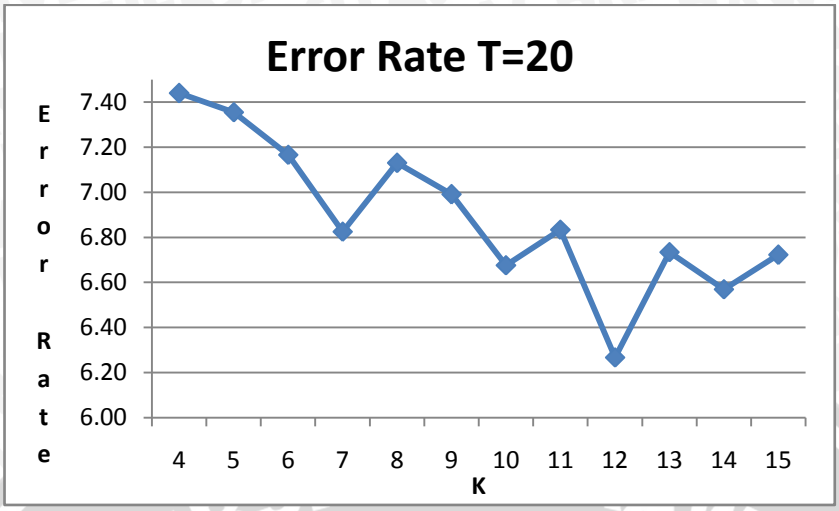
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.27%, sedangkan nilai error terkecil didapatkan saat K=12 dan K=13, yaitu sebesar 6.16%.

Selanjutnya untuk hasil pengujian dengan threshold = 20 ditampilkan pada tabel 4.4.

Tabel 4.4 Hasil Uji Pengelompokan dengan T=20

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 20 | 4 | 7.44 |
| | 5 | 7.36 |
| | 6 | 7.17 |
| | 7 | 6.83 |
| | 8 | 7.13 |
| | 9 | 6.99 |
| | 10 | 6.68 |
| | 11 | 6.83 |
| | 12 | 6.27 |
| | 13 | 6.73 |
| | 14 | 6.57 |
| | 15 | 6.72 |

Berdasarkan tabel 4.4 nilai error rate yang didapatkan berkisar antara 6% hingga 7%. Untuk nilai error ratio K=4 adalah 7.44%. Untuk K= 5 didapatkan error ratio sebesar 7.36%. Untuk K=6 didapatkan error ratio sebesar 7.17%. Untuk K=7 dan K=11 didapatkan error ratio sebesar 6.83%. Untuk K=8 didapatkan error ratio sebesar 7.13%. Untuk K=9 didapatkan error ratio sebesar 6.99%. Untuk K=10 didapatkan error ratio sebesar 6.68%. Untuk K=12 didapatkan error ratio sebesar 6.27%. Untuk K=13 didapatkan error ratio sebesar 6.73%. Untuk K=14 didapatkan error ratio sebesar 6.57%. Untuk K=15 didapatkan error ratio sebesar 6.72%.




Gambar 4.5 Grafik Hubungan k dengan *Error Rate* untuk T=20

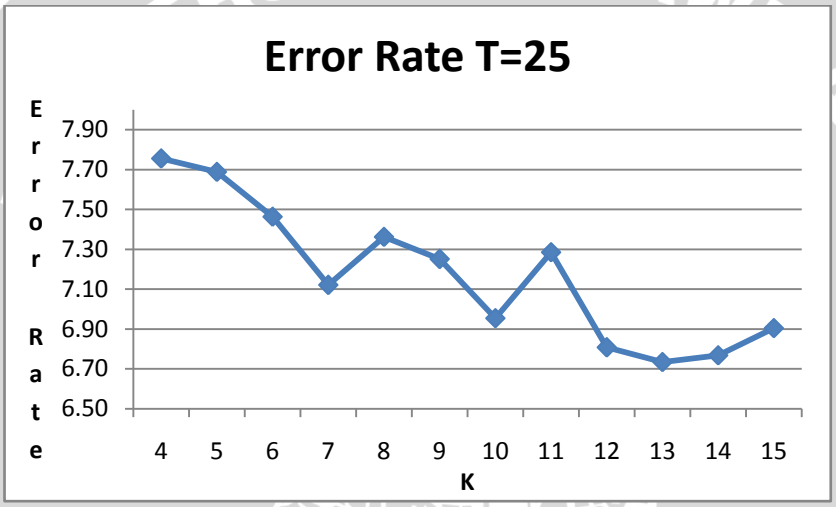
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.44% sedangkan nilai error terkecil didapatkan saat K=12, yaitu sebesar 6.27%.

Selanjutnya untuk hasil pengujian dengan threshold = 25 ditampilkan pada tabel 4.5.

Tabel 4.5 Hasil Uji Pengelompokan dengan T=25

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 25 | 4 | 7.76 |
| | 5 | 7.69 |
| | 6 | 7.46 |
| | 7 | 7.12 |
| | 8 | 7.36 |
| | 9 | 7.25 |
| | 10 | 6.95 |
| | 11 | 7.29 |
| | 12 | 6.81 |
| | 13 | 6.73 |
| | 14 | 6.77 |
| | 15 | 6.90 |

Berdasarkan tabel 4.5 nilai error rate yang didapatkan berkisar antara 6% hingga 7%. Untuk nilai error ratio K=4 adalah 7.76%. Untuk K= 5 didapatkan error ratio sebesar 7.69%. Untuk K=6 didapatkan error ratio sebesar 7.46%. Untuk K=7 didapatkan error ratio sebesar 7.12%. Untuk K=8 didapatkan error ratio sebesar 7.36%. Untuk K=9 didapatkan error ratio sebesar 7.25%. Untuk K=10 didapatkan error ratio sebesar 6.95%. Untuk K=11 didapatkan error ratio sebesar 7.29%. Untuk K=12 didapatkan error ratio sebesar 6.81%. Untuk K=13 didapatkan error ratio sebesar 6.73%. Untuk K=14 didapatkan error ratio sebesar 6.77%. Untuk K=15 didapatkan error ratio sebesar 6.90%.




Gambar 4.6 Grafik Hubungan k dengan *Error Rate* untuk T=25

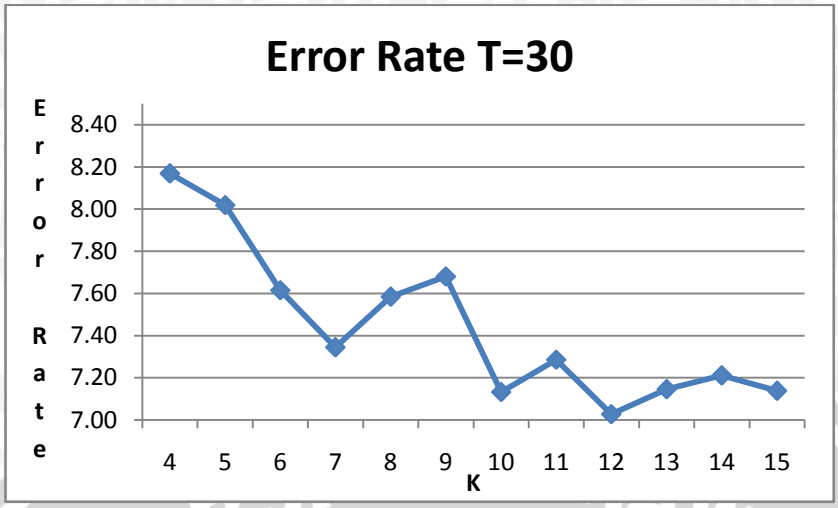
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.76% sedangkan nilai error terkecil didapatkan saat K=9, yaitu sebesar 6.73%.

Selanjutnya untuk hasil pengujian dengan threshold = 30 ditampilkan pada tabel 4.6.

Tabel 4.6 Hasil Uji Pengelompokan dengan T=30

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 30 | 4 | 8.17 |
| | 5 | 8.02 |
| | 6 | 7.62 |
| | 7 | 7.34 |
| | 8 | 7.58 |
| | 9 | 7.68 |
| | 10 | 7.13 |
| | 11 | 7.29 |
| | 12 | 7.03 |
| | 13 | 7.14 |
| | 14 | 7.21 |
| | 15 | 7.14 |

Berdasarkan tabel 4.6 nilai error rate yang didapatkan berkisar antara 7% hingga 8%. Untuk nilai error ratio K=4 adalah 8.17%. Untuk K= 5 didapatkan error ratio sebesar 8.02%. Untuk K=6 didapatkan error ratio sebesar 7.62%. Untuk K=7 didapatkan error ratio sebesar 7.34%. Untuk K=8 didapatkan error ratio sebesar 7.58%. Untuk K=9 didapatkan error ratio sebesar 7.68%. Untuk K=10 didapatkan error ratio sebesar 7.13%. Untuk K=11 didapatkan error ratio sebesar 7.29%. Untuk K=12 didapatkan error ratio sebesar 7.03%. Untuk K=13 didapatkan error ratio sebesar 7.14%. Untuk K=14 didapatkan error ratio sebesar 7.21%. Untuk K=15 didapatkan error ratio sebesar 7.14%.




Gambar 4.7 Grafik Hubungan k dengan *Error Rate* untuk T=30

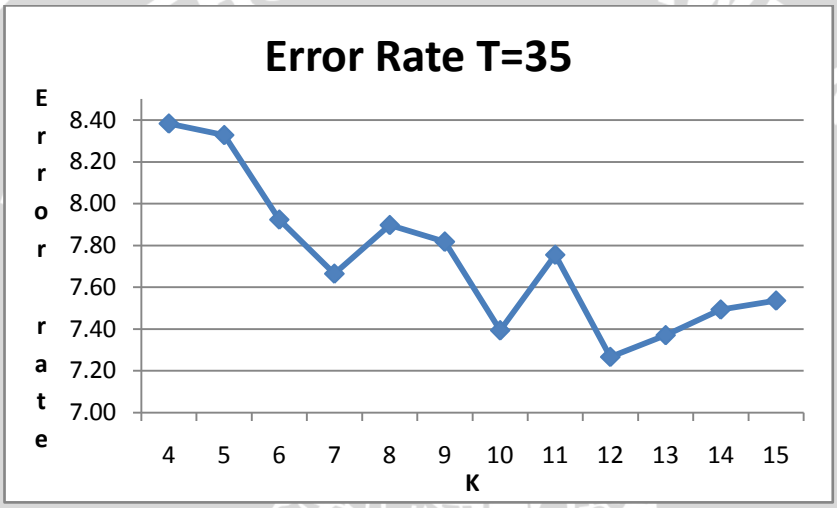
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 8.17% sedangkan nilai error terkecil didapatkan saat K=12, yaitu sebesar 7.03%.

Selanjutnya untuk hasil pengujian dengan threshold = 35 ditampilkan pada tabel 4.7.

Tabel 4.7 Hasil Uji Pengelompokan dengan T=35

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 35 | 4 | 8.38 |
| | 5 | 8.33 |
| | 6 | 7.92 |
| | 7 | 7.66 |
| | 8 | 7.90 |
| | 9 | 7.82 |
| | 10 | 7.39 |
| | 11 | 7.76 |
| | 12 | 7.27 |
| | 13 | 7.37 |
| | 14 | 7.49 |
| | 15 | 7.54 |

Berdasarkan tabel 4.7 nilai error rate yang didapatkan berkisar antara 7% hingga 8%. Untuk nilai error ratio K=4 adalah 8.38%. Untuk K= 5 didapatkan error ratio sebesar 8.33%. Untuk K=6 didapatkan error ratio sebesar 7.92%. Untuk K=7 didapatkan error ratio sebesar 7.66%. Untuk K=8 didapatkan error ratio sebesar 7.90%. Untuk K=9 didapatkan error ratio sebesar 7.82%. Untuk K=10 didapatkan error ratio sebesar 7.39%. Untuk K=11 didapatkan error ratio sebesar 7.76%. Untuk K=12 didapatkan error ratio sebesar 7.27%. Untuk K=13 didapatkan error ratio sebesar 7.37%. Untuk K=14 didapatkan error ratio sebesar 7.49%. Untuk K=15 didapatkan error ratio sebesar 7.54%.




Gambar 4.8 Grafik Hubungan k dengan *Error rate* untuk T=35

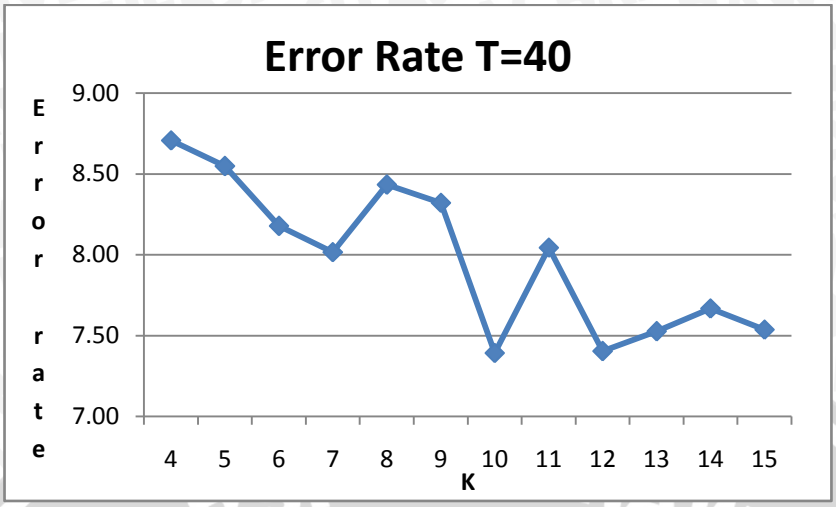
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 8.38% sedangkan nilai error terkecil didapatkan saat K=12, yaitu sebesar 7.27%.

Selanjutnya untuk hasil pengujian dengan threshold = 40 ditampilkan pada tabel 4.8.

Tabel 4.8 Hasil Uji Pengelompokan dengan T=40

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 40 | 4 | 8.71 |
| | 5 | 8.55 |
| | 6 | 8.18 |
| | 7 | 8.02 |
| | 8 | 8.43 |
| | 9 | 8.32 |
| | 10 | 7.39 |
| | 11 | 8.04 |
| | 12 | 7.40 |
| | 13 | 7.53 |
| | 14 | 7.67 |
| | 15 | 7.54 |

Berdasarkan tabel 4.8 nilai error rate yang didapatkan berkisar antara 7% hingga 8%. Untuk nilai error ratio K=4 adalah 8.71%. Untuk K= 5 didapatkan error ratio sebesar 8.55%. Untuk K=6 didapatkan error ratio sebesar 8.18%. Untuk K=7 didapatkan error ratio sebesar 8.02%. Untuk K=8 didapatkan error ratio sebesar 8.43%. Untuk K=9 didapatkan error ratio sebesar 8.32%. Untuk K=10 didapatkan error ratio sebesar 7.39%. Untuk K=11 didapatkan error ratio sebesar 8.04%. Untuk K=12 didapatkan error ratio sebesar 7.40%. Untuk K=13 didapatkan error ratio sebesar 7.53%. Untuk K=14 didapatkan error ratio sebesar 7.67%. Untuk K=15 didapatkan error ratio sebesar 7.54%.




Gambar 4.0.9 Grafik Hubungan k dengan *Error Rate* untuk T=40

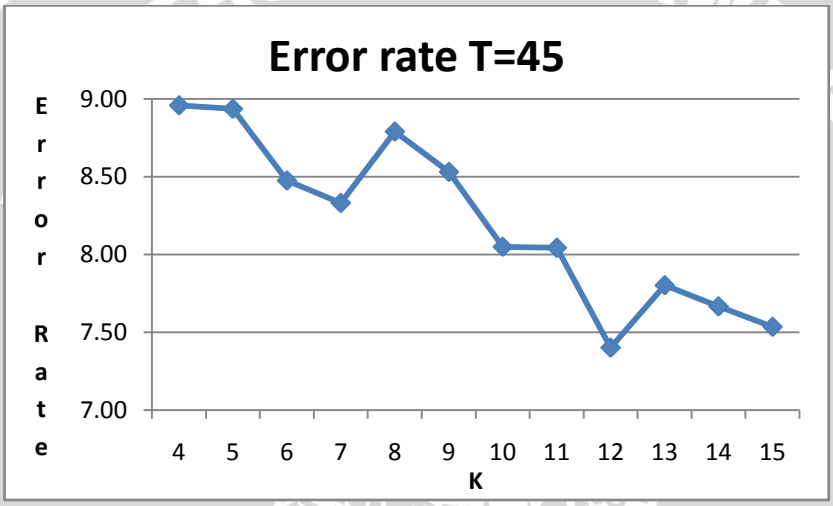
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 8.71% sedangkan nilai error terkecil didapatkan saat K=10, yaitu sebesar 7.39%.

Selanjutnya untuk hasil pengujian dengan threshold = 45 ditampilkan pada tabel 4.9.

Tabel 4.9 Hasil Uji Pengelompokan dengan T=45

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 45 | 4 | 8.96 |
| | 5 | 8.94 |
| | 6 | 8.48 |
| | 7 | 8.33 |
| | 8 | 8.79 |
| | 9 | 8.53 |
| | 10 | 8.05 |
| | 11 | 8.04 |
| | 12 | 7.40 |
| | 13 | 7.80 |
| | 14 | 7.67 |
| | 15 | 7.54 |

Berdasarkan tabel 4.9 nilai error rate yang didapatkan berkisar antara 7% hingga 8%. Untuk nilai error ratio K=4 adalah 8.96%. Untuk K= 5 didapatkan error ratio sebesar 8.94%. Untuk K=6 didapatkan error ratio sebesar 8.48%. Untuk K=7 didapatkan error ratio sebesar 8.33%. Untuk K=8 didapatkan error ratio sebesar 8.79%. Untuk K=9 didapatkan error ratio sebesar 8.53%. Untuk K=10 didapatkan error ratio sebesar 8.05%. Untuk K=11 didapatkan error ratio sebesar 8.04%. Untuk K=12 didapatkan error ratio sebesar 7.40%. Untuk K=13 didapatkan error ratio sebesar 7.80%. Untuk K=14 didapatkan error ratio sebesar 7.67%. Untuk K=15 didapatkan error ratio sebesar 7.54%.




Gambar 4.10 Grafik Hubungan k dengan *Error rate* untuk T=45

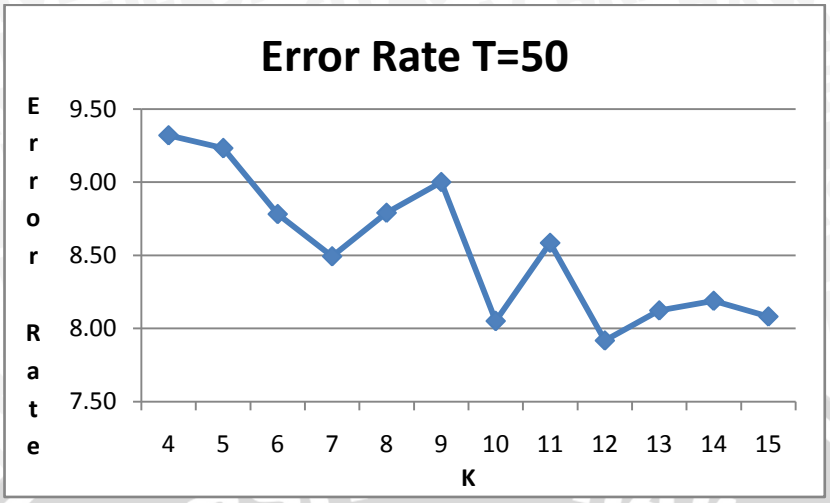
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 8.96% sedangkan nilai error terkecil didapatkan saat K=12, yaitu sebesar 7.40%.

Selanjutnya untuk hasil pengujian dengan threshold = 50 ditampilkan pada tabel 4.10.

Tabel 4.10 Hasil Uji Pengelompokan dengan T=50

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 50 | 4 | 9.32 |
| | 5 | 9.23 |
| | 6 | 8.78 |
| | 7 | 8.49 |
| | 8 | 8.79 |
| | 9 | 9.00 |
| | 10 | 8.05 |
| | 11 | 8.59 |
| | 12 | 7.92 |
| | 13 | 8.12 |
| | 14 | 8.19 |
| | 15 | 8.08 |

Berdasarkan tabel 4.10 nilai error rate yang didapatkan berkisar antara 7% hingga 9%. Untuk nilai error ratio K=4 adalah 9.32%. Untuk K= 5 didapatkan error ratio sebesar 9.23%. Untuk K=6 didapatkan error ratio sebesar 8.78%. Untuk K=7 didapatkan error ratio sebesar 8.49%. Untuk K=8 didapatkan error ratio sebesar 8.79%. Untuk K=9 didapatkan error ratio sebesar 9.00%. Untuk K=10 didapatkan error ratio sebesar 8.05%. Untuk K=11 didapatkan error ratio sebesar 8.59%. Untuk K=12 didapatkan error ratio sebesar 7.92%. Untuk K=13 didapatkan error ratio sebesar 8.12%. Untuk K=14 didapatkan error ratio sebesar 8.19%. Untuk K=15 didapatkan error ratio sebesar 8.08%.




Gambar 4.11 Grafik Hubungan k dengan *Error Rate* untuk T=50

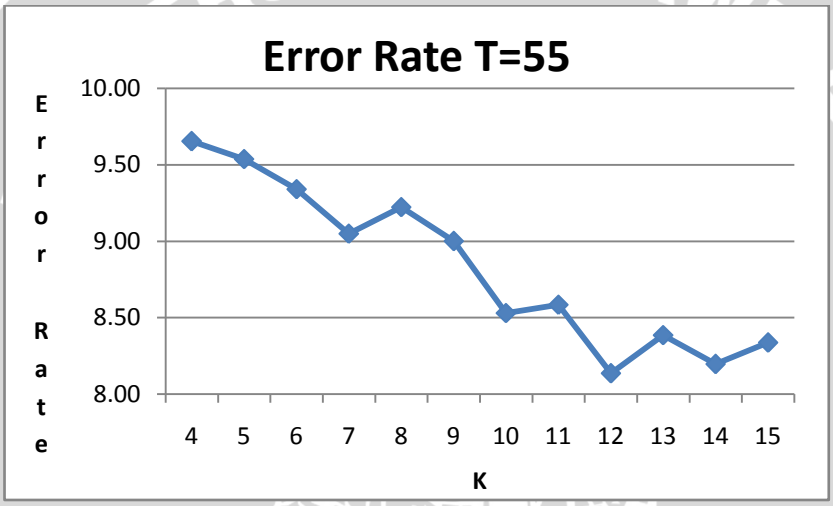
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 9.32% sedangkan nilai error terkecil didapatkan saat K=12, yaitu sebesar 7.92%.

Selanjutnya untuk hasil pengujian dengan threshold = 55 ditampilkan pada tabel 4.11.

Tabel 4.11 Hasil Uji Pengelompokan dengan T=55

| Threshold | k | *Error rate*(%) |
|---|---|---|
| 55 | 4 | 9.65 |
| | 5 | 9.54 |
| | 6 | 9.34 |
| | 7 | 9.05 |
| | 8 | 9.22 |
| | 9 | 9.00 |
| | 10 | 8.53 |
| | 11 | 8.59 |
| | 12 | 8.14 |
| | 13 | 8.39 |
| | 14 | 8.20 |
| | 15 | 8.34 |

Berdasarkan tabel 4.11 nilai error rate yang didapatkan berkisar antara 15% hingga 18%. Untuk nilai error ratio K=4 adalah 9.65%. Untuk K= 5 didapatkan error ratio sebesar 9.54%. Untuk K=6 didapatkan error ratio sebesar 9.34%. Untuk K=7 didapatkan error ratio sebesar 9.05%. Untuk K=8 didapatkan error ratio sebesar 9.22%. Untuk K=9 didapatkan error ratio sebesar 9.00%. Untuk K=10 didapatkan error ratio sebesar 8.53%. Untuk K=11 didapatkan error ratio sebesar 8.59%. Untuk K=12 didapatkan error ratio sebesar 8.14%. Untuk K=13 didapatkan error ratio sebesar 8.39%. Untuk K=14 didapatkan error ratio sebesar 8.20%. Untuk K=15 didapatkan error ratio sebesar 8.34%.




Gambar 4.12 Grafik Hubungan k dengan *Error Rate* untuk T=55

Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 9.65% sedangkan nilai error terkecil didapatkan saat K=12, yaitu sebesar 8.14%.

### 4.3.4. Uji Pengaruh Nilai k pada Proses Klasifikasi Terhadap Penentuan Kelas

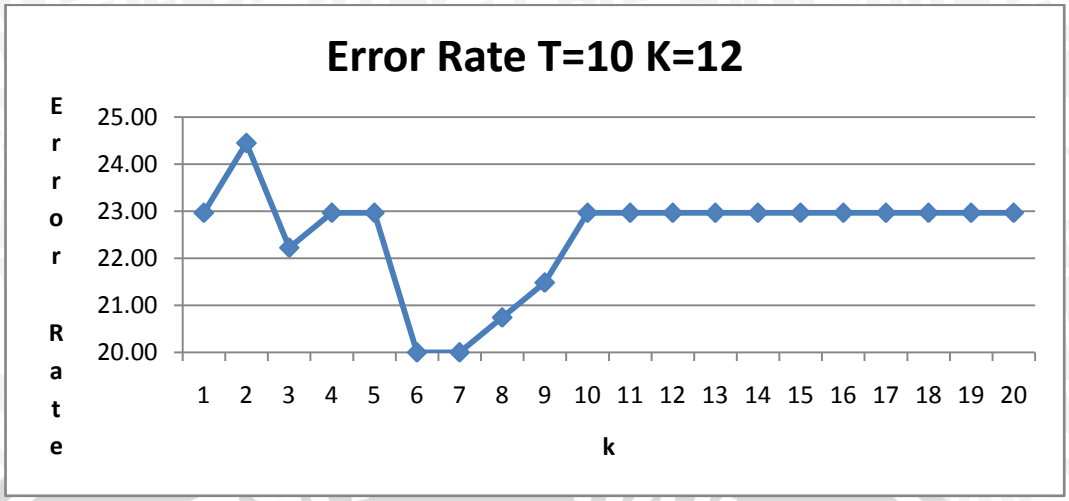
Pengujian kedua adalah pengujian untuk mengetahui pengaruh nilai k pada proses klasifikasi terhadap penentuan kelas. Pengujian ini melibatkan 135 data uji. Uji nilai K dimulai dari K=1 hingga K=20 dengan mengambil nilai k paling optimal pada proses pengelompokan untuk T=10 hingga T=55.

Hasil pengujian untuk T=10 dengan k paling optimal yaitu 14 dapat dilihat pada tabel 4.12.

Tabel 4.12 Hasil Uji Klasifikasi dengan T=10 k=14

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 10 | 14 | 1 | 22.96 |
| | | 2 | 24.44 |
| | | 3 | 22.22 |
| | | 4 | 22.96 |
| | | 5 | 22.96 |
| | | 6 | 20.00 |
| | | 7 | 20.00 |
| | | 8 | 20.74 |
| | | 9 | 21.48 |
| | | 10 | 22.96 |
| | | 11 | 22.96 |
| | | 12 | 22.96 |
| | | 13 | 22.96 |
| | | 14 | 22.96 |
| | | 15 | 22.96 |
| | | 16 | 22.96 |
| | | 17 | 22.96 |
| | | 18 | 22.96 |
| | | 19 | 22.96 |
| | | 20 | 22.96 |

Berdasarkan tabel 4.12 nilai *error rate* berkisar antara 22% hingga 24%. Pada K=1, K=4, K=5, K=10 hingga K=20 didapatkan nilai *error rate* sebesar 22.96%. Pada K=2 didapatkan nilai *error rate* sebesar 24.44%. Pada K=3 didapatkan nilai *error rate* sebesar 22.22%. Pada K=6 dan K=7 didapatkan nilai *error rate* sebesar 20.00%. Pada K=8 didapatkan nilai *error rate* sebesar 20.74%. Pada K=9 didapatkan *error rate* sebesar 21.48%.




Gambar 4.13 Grafik Hubungan k dengan *Error Rate* untuk T=10 K=12

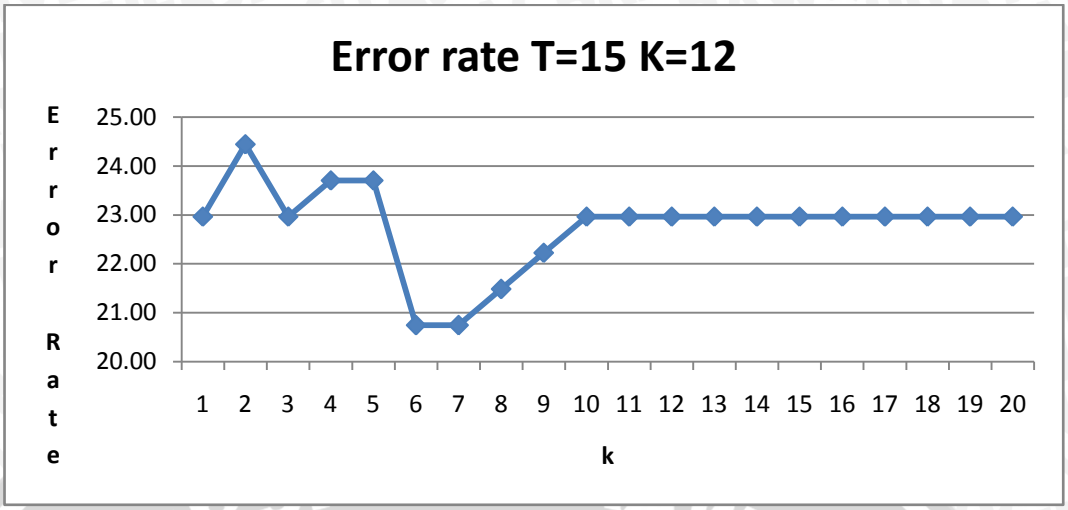
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=2 yaitu sebesar 24.44% sedangkan nilai error terkecil didapatkan saat K=6 dan K=7, yaitu sebesar 20.00%. Untuk K=10 hingga K=20 didapatkan nilai yang konstan yaitu sebesar 22.96%.

Hasil pengujian untuk T=15 dengan k paling optimal yaitu 12 dapat dilihat pada tabel 4.13

Tabel 4.13 Hasil Uji Klasifikasi dengan T=15 K=12

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 15 | 12 | 1 | 22.96 |
| | | 2 | 24.44 |
| | | 3 | 22.96 |
| | | 4 | 23.70 |
| | | 5 | 23.70 |
| | | 6 | 20.74 |
| | | 7 | 20.74 |
| | | 8 | 21.48 |
| | | 9 | 22.22 |
| | | 10 | 22.96 |
| | | 11 | 22.96 |
| | | 12 | 22.96 |
| | | 13 | 22.96 |
| | | 14 | 22.96 |
| | | 15 | 22.96 |
| | | 16 | 22.96 |
| | | 17 | 22.96 |
| | | 18 | 22.96 |
| | | 19 | 22.96 |
| | | 20 | 22.96 |

Berdasarkan tabel 4.13 nilai *error rate* berkisar antara 22% hingga 24%. Pada K=1, K=3, K=10 hingga K=20 didapatkan nilai *error rate* sebesar 22.96%. Pada K=2 didapatkan nilai *error rate* sebesar 24.44%. Pada K=4 dan K=5 didapatkan nilai *error rate* sebesar 23.70%. Pada K=6 dan K=7 didapatkan nilai *error rate* sebesar 20.74%. Pada K=8 didapatkan nilai *error rate* sebesar 21.48%. Pada K=9 didapatkan nilai *error rate* sebesar 22.22%.




Gambar 4.14 Grafik Hubungan k dengan *Error Rate* untuk T=15 K=12

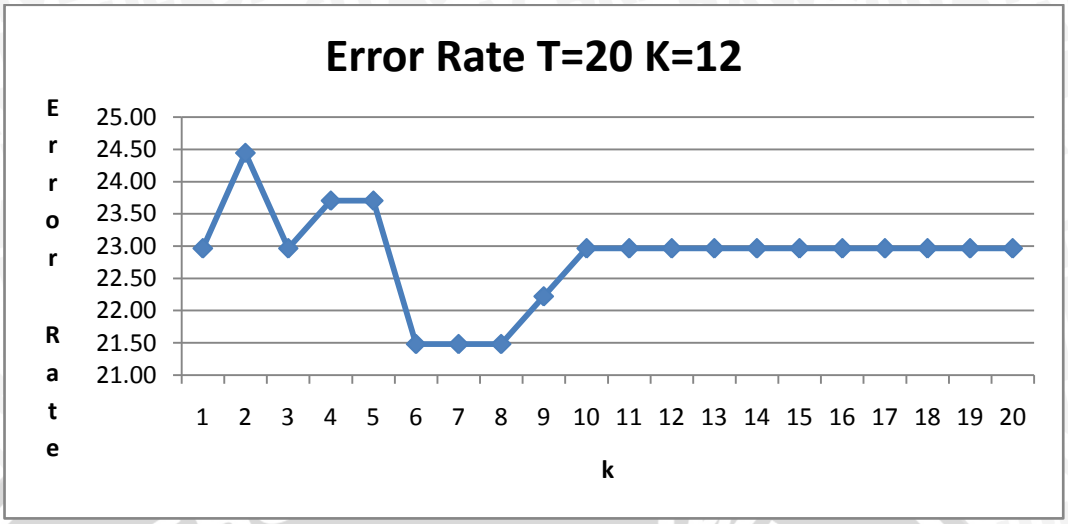
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=2 yaitu sebesar 24.44% sedangkan nilai error terkecil didapatkan saat K=6 dan K=7 yaitu sebesar 20.74%. Pada K=10 hingga K=20 didapatkan nilai yang konstan yaitu sebesar 22.96%.

Hasil pengujian untuk T=20 dengan k paling optimal yaitu 12 dapat dilihat pada tabel 4.14.

Tabel 4.14 Hasil Uji Klasifikasi dengan T=20 K=12

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 20 | 12 | 1 | 22.96 |
| | | 2 | 24.44 |
| | | 3 | 22.96 |
| | | 4 | 23.70 |
| | | 5 | 23.70 |
| | | 6 | 21.48 |
| | | 7 | 21.48 |
| | | 8 | 21.48 |
| | | 9 | 22.22 |
| | | 10 | 22.96 |
| | | 11 | 22.96 |
| | | 12 | 22.96 |
| | | 13 | 22.96 |
| | | 14 | 22.96 |
| | | 15 | 22.96 |
| | | 16 | 22.96 |
| | | 17 | 22.96 |
| | | 18 | 22.96 |
| | | 19 | 22.96 |
| | | 20 | 22.96 |

Berdasarkan tabel 4.14 nilai *error rate* berkisar antara 22% hingga 24%. Pada K=1, K=3, K=10 hingga K=20 didapatkan nilai *error rate* sebesar 22.96%. Pada K=2 didapatkan nilai *error rate* sebesar 24.44%. Pada K=4 dan K=5 didapatkan nilai *error rate* sebesar 23.70%. Pada K=6, K=7, dan K=8 didapatkan nilai *error rate* sebesar 21.48%. Pada K=9 didapatkan nilai *error rate* sebesar 22.22%.




Gambar 4.15Grafik Hubungan k dengan *Error Rate* untuk T=20 K=12

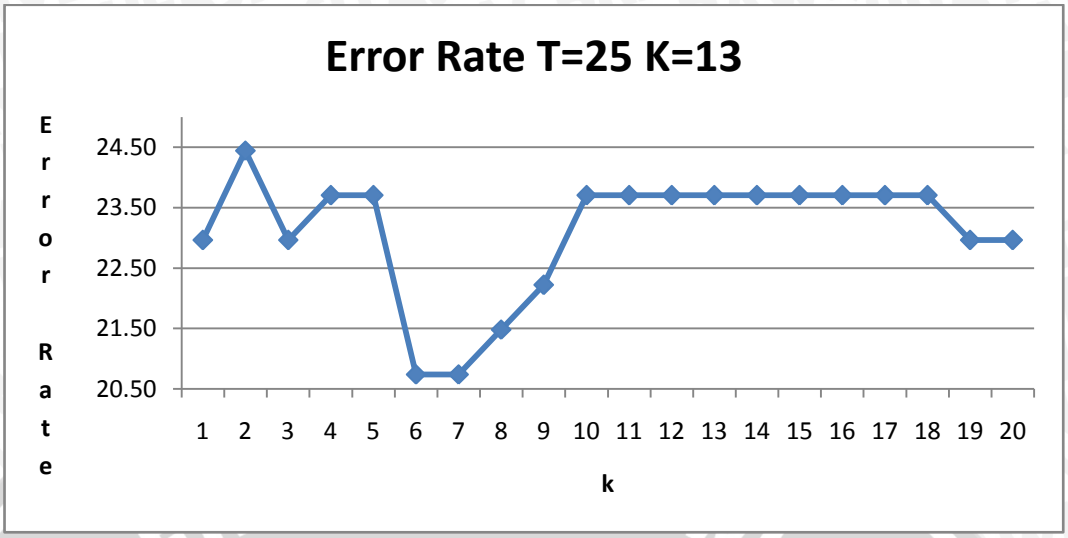
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=2 yaitu sebesar 24.44% sedangkan nilai error terkecil didapatkan saat K=6, K=7, K=8 yaitu sebesar 21.48%. Pada K=10 hingga K=20 didapatkan nilai konstan yaitu sebesar 22.96%.

Hasil pengujian untuk T=25 dengan k paling optimal yaitu 13 dapat dilihat pada tabel 4.15.

Tabel 4.15 Hasil Uji Klasifikasi dengan T=25 K=13

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 25 | 13 | 1 | 22.96 |
| | | 2 | 24.44 |
| | | 3 | 22.96 |
| | | 4 | 23.70 |
| | | 5 | 23.70 |
| | | 6 | 20.74 |
| | | 7 | 20.74 |
| | | 8 | 21.48 |
| | | 9 | 22.22 |
| | | 10 | 23.70 |
| | | 11 | 23.70 |
| | | 12 | 23.70 |
| | | 13 | 23.70 |
| | | 14 | 23.70 |
| | | 15 | 23.70 |
| | | 16 | 23.70 |
| | | 17 | 23.70 |
| | | 18 | 23.70 |
| | | 19 | 22.96 |
| | | 20 | 22.96 |

Berdasarkan tabel 4.15 nilai *error rate* berkisar antara 22% hingga 24%. Pada K=1, K=3, K=19 dan K=20 didapatkan nilai *error rate* sebesar 22.96%. Pada K=2 didapatkan nilai *error rate* sebesar 24.44%. Pada K=4, K=5, K=10 hingga K=18 didapatkan nilai *error rate* sebesar 23.70%. Pada K=7 didapatkan nilai *error rate* sebesar 20.74%. Pada K=8 didapatkan nilai *error rate* sebesar 21.48%. Pada K=9 didapatkan nilai *error rate* sebesar 22.22%.




Gambar 4.16 Grafik Hubungan k dengan *error rate* untuk T=25 K=13

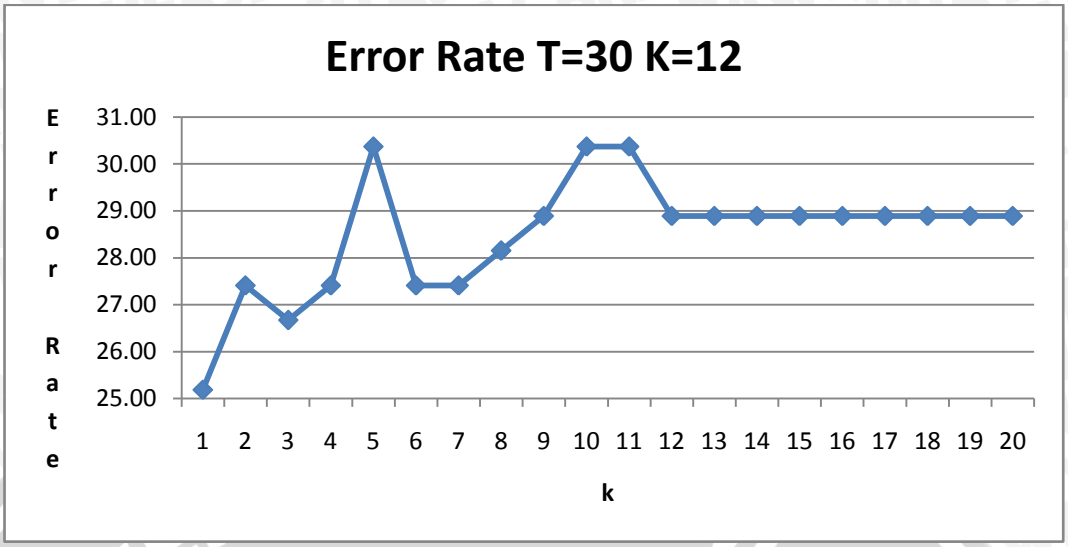
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=2 sebesar 24.44% sedangkan nilai error terkecil didapatkan saat K=6 dan K=7 sebesar 20.74%. Nilai konstan didapatkan pada saat K=10 hingga K=18 yaitu sebesar 23.70%.

Hasil pengujian untuk T=30 dengan k paling optimal yaitu 12 dapat dilihat pada tabel 4.15.

Tabel 4.16 Hasil Uji Klasifikasi dengan T=30 K=12

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 30 | 12 | 1 | 25.19 |
| | | 2 | 27.41 |
| | | 3 | 26.67 |
| | | 4 | 27.41 |
| | | 5 | 30.37 |
| | | 6 | 27.41 |
| | | 7 | 27.41 |
| | | 8 | 28.15 |
| | | 9 | 28.89 |
| | | 10 | 30.37 |
| | | 11 | 30.37 |
| | | 12 | 28.89 |
| | | 13 | 28.89 |
| | | 14 | 28.89 |
| | | 15 | 28.89 |
| | | 16 | 28.89 |
| | | 17 | 28.89 |
| | | 18 | 28.89 |
| | | 19 | 28.89 |
| | | 20 | 28.89 |

Berdasarkan tabel 4.16 nilai *error rate* berkisar antara 25% hingga 30%. Pada K=1 didapatkan nilai *error rate* sebesar 25.19%. Pada K=2, K=4, K=6 dan K=7 didapatkan nilai *error rate* sebesar 27.41%. Pada K=3 didapatkan nilai *error rate* sebesar 26.67%. Pada K=5, K=10 dan K=11 didapatkan nilai *error rate* sebesar 30.37%. Pada K=8 didapatkan nilai *error rate* sebesar 28.15%. Pada K=9, K=12 hinga K=20 didapatkan nilai *error rate* sebesar 28.89%.




Gambar 4.17 Grafik Hubungan k dengan *Error Rate* untuk T=30 K=12

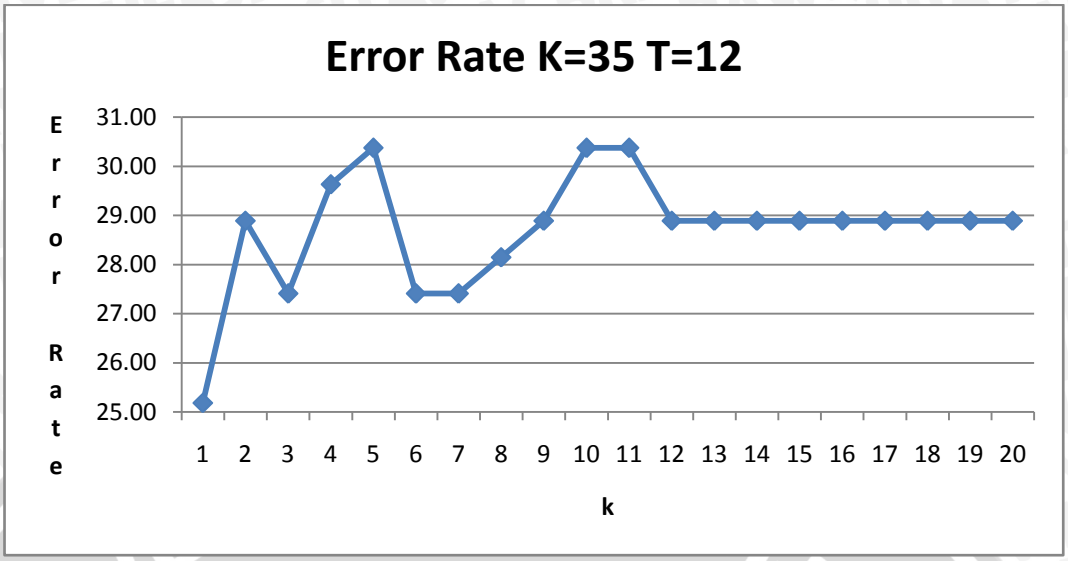
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=5 yaitu sebesar 30.37% sedangkan nilai error terkecil didapatkan saat K=1 sebesar 25.19%. Nilai konstan didapatkan pada saat K=12 hingga K=20 yaitu sebesar 28.89%.

Hasil pengujian untuk T=35 dengan k paling optimal yaitu 10 dapat dilihat pada tabel 4.17.

Tabel 4.17 Hasil Uji Klasifikasi dengan T=35 K=10

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 35 | 12 | 1 | 25.19 |
| | | 2 | 28.89 |
| | | 3 | 27.41 |
| | | 4 | 29.63 |
| | | 5 | 30.37 |
| | | 6 | 27.41 |
| | | 7 | 27.41 |
| | | 8 | 28.15 |
| | | 9 | 28.89 |
| | | 10 | 30.37 |
| | | 11 | 30.37 |
| | | 12 | 28.89 |
| | | 13 | 28.89 |
| | | 14 | 28.89 |
| | | 15 | 28.89 |
| | | 16 | 28.89 |
| | | 17 | 28.89 |
| | | 18 | 28.89 |
| | | 19 | 28.89 |
| | | 20 | 28.89 |

Berdasarkan tabel 4.17 nilai *error rate* berkisar antara 25% hingga 30%. Pada K=1 didapatkan nilai *error rate* sebesar 25.19%. Pada K=2, K=9 dan K=12 hingga K=20 didapatkan nilai *error rate* sebesar 28.89%. Pada K=3, K=6, dan K7 didapatkan nilai *error rate* sebesar 27.41%. Pada K=4 didapatkan nilai *error rate* sebesar 29.63%. Pada K=5, K=10 dan K=11 didapatkan nilai *error rate* sebesar 30.37%.




Gambar 4.18 Grafik Hubungan k dengan *Error Rate* untuk T=35 K=10

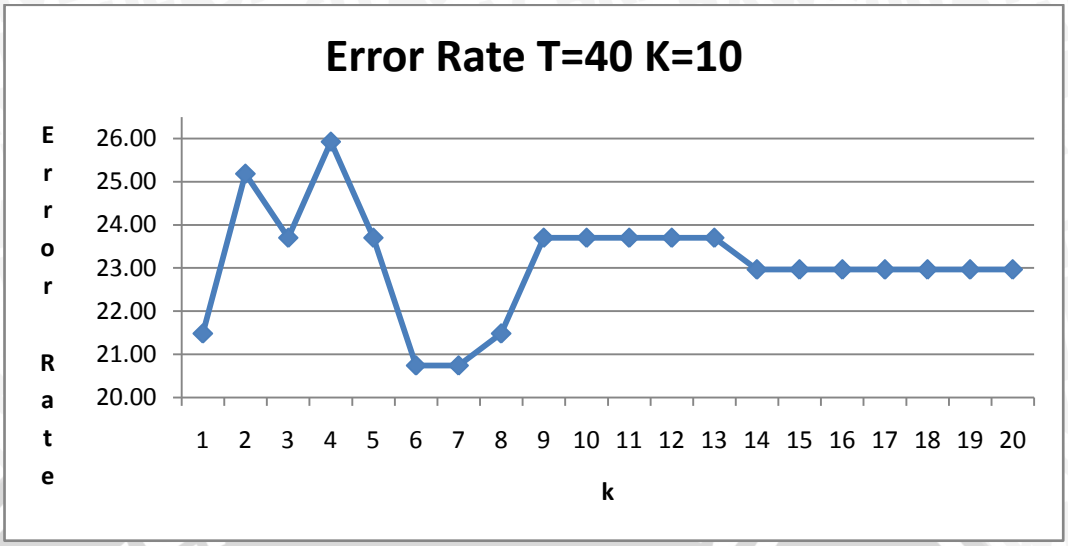
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=5, K=10 dan K=11 yaitu sebesar 30.37%. Sedangkan nilai error terkecil didapatkan saat K=1 sebesar 25.19%. Nilai konstan didapatkan pada saat K=12 hingga K=20 yaitu sebesar 28.89%.

Hasil pengujian untuk T=40 dengan k paling optimal yaitu 10 dapat dilihat pada tabel 4.18.

Tabel 4.18 Hasil Uji Klasifikasi dengan T=40 K=10

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 40 | 10 | 1 | 21.48 |
| | | 2 | 25.19 |
| | | 3 | 23.70 |
| | | 4 | 25.93 |
| | | 5 | 23.70 |
| | | 6 | 20.74 |
| | | 7 | 20.74 |
| | | 8 | 21.48 |
| | | 9 | 23.70 |
| | | 10 | 23.70 |
| | | 11 | 23.70 |
| | | 12 | 23.70 |
| | | 13 | 23.70 |
| | | 14 | 22.96 |
| | | 15 | 22.96 |
| | | 16 | 22.96 |
| | | 17 | 22.96 |
| | | 18 | 22.96 |
| | | 19 | 22.96 |
| | | 20 | 22.96 |

Berdasarkan tabel 4.18 nilai *error rate* berkisar antara 20% hingga 25%. Pada K=1 didapatkan nilai *error rate* sebesar 21.48%. Pada K=2 didapatkan nilai *error rate* sebesar 25.19%. Pada K=3, K=5, K=9 hingga K=13 didapatkan nilai *error rate* sebesar 23.70%. Pada K=4 didapatkan nilai *error rate* sebesar 25.93%. Pada K=6 dan K=7 didapatkan nilai *error rate* sebesar 20.74%. Pada K=14 hingga K=20 didapatkan nilai *error rate* yang konstan yaitu sebesar 22.96%.




Gambar 4.19 Grafik Hubungan k dengan *Error Rate* untuk T=40 K=10

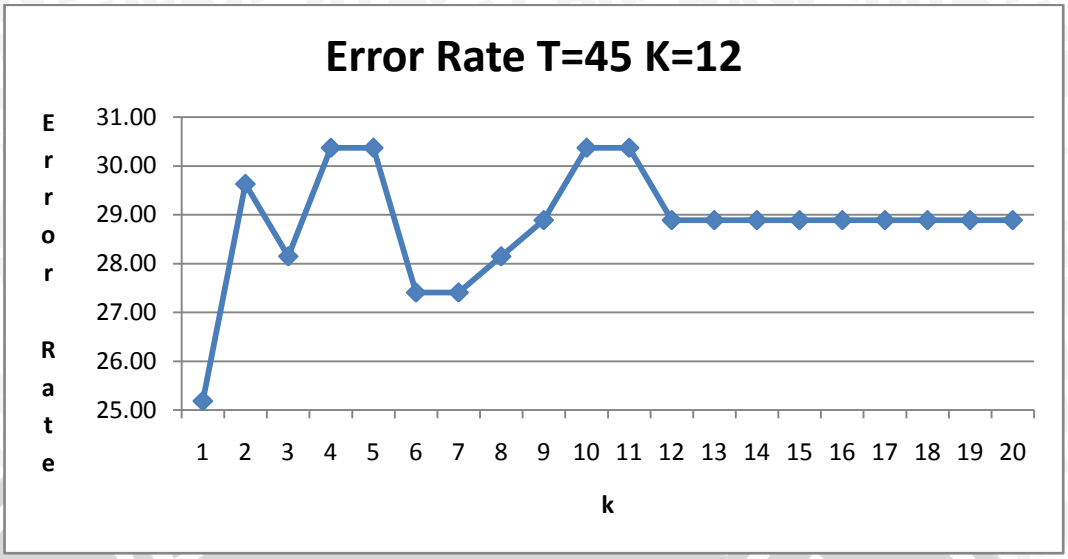
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 25.93% sedangkan nilai error terkecil didapatkan saat K=6 dan K=7 sebesar 20.74%. Nilai konstan didapatkan pada saat K=9 hingga K=13 yaitu sebesar 23.70% dan K=14 hingga K=20 yaitu sebesar 22.96%.

Hasil pengujian untuk T=45 dengan k paling optimal yaitu 12 dapat dilihat pada tabel 4.19.

Tabel 4.19 Hasil Uji Klasifikasi dengan T=45 K=12

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 45 | 12 | 1 | 25.19 |
| | | 2 | 29.63 |
| | | 3 | 28.15 |
| | | 4 | 30.37 |
| | | 5 | 30.37 |
| | | 6 | 27.41 |
| | | 7 | 27.41 |
| | | 8 | 28.15 |
| | | 9 | 28.89 |
| | | 10 | 30.37 |
| | | 11 | 30.37 |
| | | 12 | 28.89 |
| | | 13 | 28.89 |
| | | 14 | 28.89 |
| | | 15 | 28.89 |
| | | 16 | 28.89 |
| | | 17 | 28.89 |
| | | 18 | 28.89 |
| | | 19 | 28.89 |
| | | 20 | 28.89 |

Berdasarkan tabel 4.14 nilai *error rate* berkisar antara 25% hingga 30%. Pada K=1 didapatkan nilai *error rate* sebesar 25.19%. Pada K=2 didapatkan nilai *error rate* sebesar 29.63%. Pada K=3 dan K=8 didapatkan nilai *error rate* sebesar 28.15%. Pada K=4, K=5, K=10 dan K=11 didapatkan nilai *error rate* sebesar 30.37%. Pada K=6 dan K=7 didapatkan nilai *error rate* sebesar 27.41%. Pada K=9, K=12 hingga K=20 didapatkan nilai *error rate* sebesar 28.89%.




Gambar 4.20 Grafik Hubungan k dengan *Error Rate* untuk T=45 K=12

Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4, K=4, K=10 dan K=11 yaitu sebesar 30.37% sedangkan nilai error terkecil didapatkan saat K=1 sebesar 25.19%. Nilai konstan didapatkan pada saat K=12 hingga K=20 yaitu sebesar 28.89%.

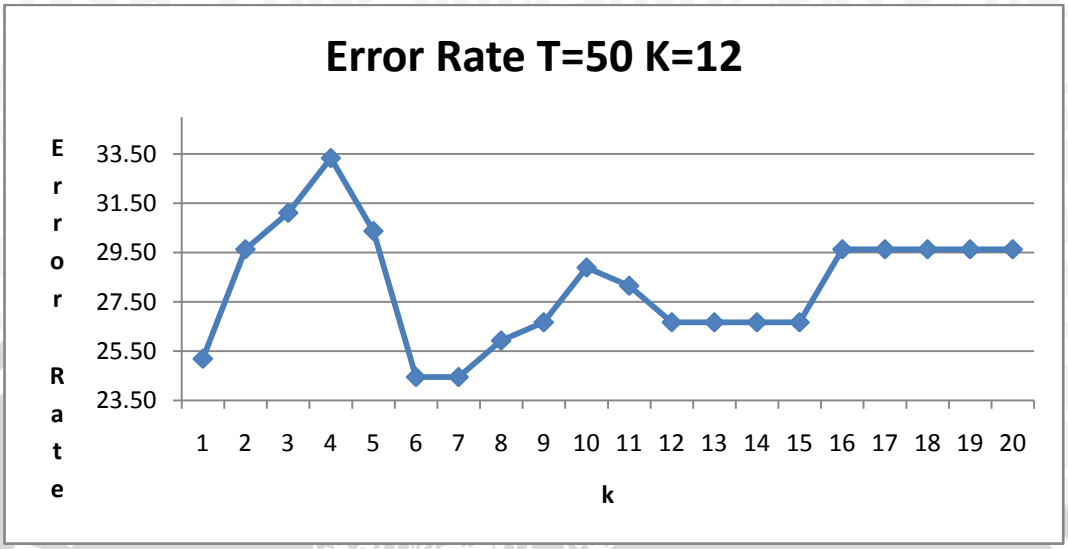
Hasil pengujian untuk T=50 dengan k paling optimal yaitu 12 dapat dilihat pada tabel 4.20.

Tabel 4.20 Hasil Uji Klasifikasi dengan T=50 K=12

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 50 | 12 | 1 | 25.19 |
| | | 2 | 29.63 |
| | | 3 | 31.11 |
| | | 4 | 33.33 |
| | | 5 | 30.37 |
| | | 6 | 24.44 |
| | | 7 | 24.44 |
| | | 8 | 25.93 |
| | | 9 | 26.67 |
| | | 10 | 28.89 |
| | | 11 | 28.15 |
| | | 12 | 26.67 |
| | | 13 | 26.67 |
| | | 14 | 26.67 |
| | | 15 | 26.67 |
| | | 16 | 29.63 |
| | | 17 | 29.63 |
| | | 18 | 29.63 |
| | | 19 | 29.63 |
| | | 20 | 29.63 |

Berdasarkan tabel 4.20 nilai *error rate* berkisar antara 24% hingga 33%. Pada K=1 didapatkan nilai *error rate* sebesar 25.19%. Pada K=2 didapatkan nilai *error rate* sebesar 29.63%. Pada K=3 didapatkan nilai *error rate* sebesar 31.11%. Pada K=4 didapatkan nilai *error rate* sebesar 33.33%. Pada K=5 didapatkan nilai *error rate* sebesar 30.37%. Pada K=6 dan K=7 didapatkan nilai *error rate* sebesar 24.44%. Pada K=8 didapatkan nilai *error rate* sebesar 25.93%. Pada K=9 didapatkan nilai *error rate* sebesar 26.67%. Pada K=10 didapatkan nilai *error rate* sebesar 28.89% Pada K=11 didapatkan nilai *error rate* sebesar 28.15%. Pada

K=12 hingga K=15 didapatkan nilai *error rate* sebesar 26.67%. Pada K=16 hingga K=20 didapatkan nilai *error rate* sebesar 29.63%.




Gambar 4.21 Grafik Hubungan k dengan *Error Rate* untuk T=50 K=12

Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 33.33% sedangkan nilai error terkecil didapatkan saat K=6 dan K=7 sebesar 24.44%. Nilai konstan didapatkan pada saat K=12 hingga K=15 yaitu sebesar 26.67%.

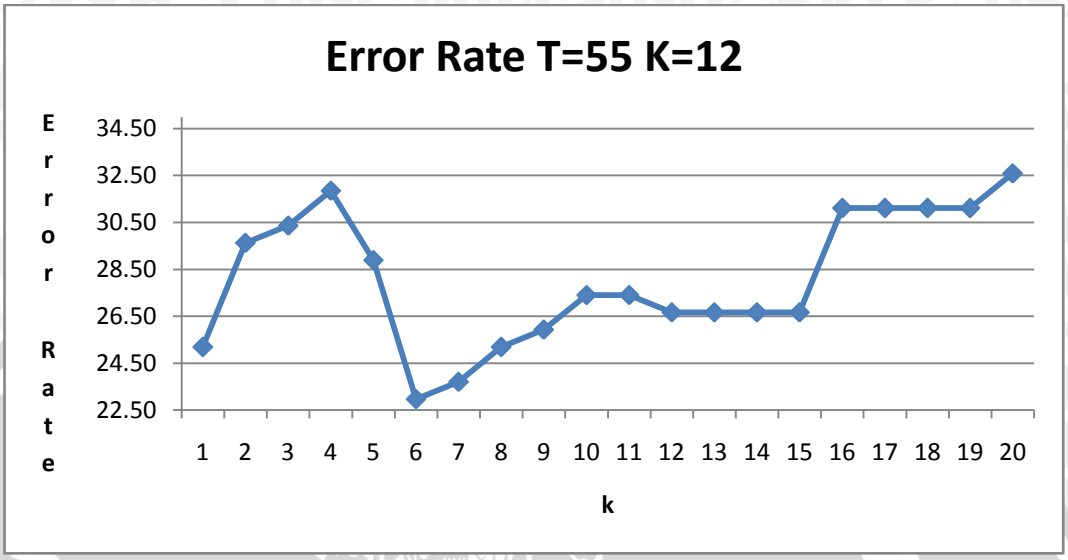
Hasil pengujian untuk T=55 dengan k paling optimal yaitu 12 dapat dilihat pada tabel 4.21.

Tabel 4.21 Hasil Uji Klasifikasi dengan T=55 K=12

| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 55 | 12 | 1 | 25.19 |
| | | 2 | 29.63 |
| | | 3 | 30.37 |
| | | 4 | 31.85 |
| | | 5 | 28.89 |
| | | 6 | 22.96 |
| | | 7 | 23.70 |
| | | 8 | 25.19 |
| | | 9 | 25.93 |
| | | 10 | 27.41 |
| | | 11 | 27.41 |
| | | 12 | 26.67 |
| | | 13 | 26.67 |
| | | 14 | 26.67 |
| | | 15 | 26.67 |
| | | 16 | 31.11 |
| | | 17 | 31.11 |
| | | 18 | 31.11 |
| | | 19 | 31.11 |
| | | 20 | 32.59 |

Berdasarkan tabel 4.21 nilai *error rate* berkisar antara 25% hingga 31%. Pada K=1 dan K=8 didapatkan nilai *error rate* sebesar 25.19%. Pada K=2 didapatkan nilai *error rate* sebesar 29.63%. Pada K=3 didapatkan nilai *error rate* sebesar 30.37%. Pada K=4 didapatkan nilai *error rate* sebesar 31.85%. Pada K=5 didapatkan nilai *error rate* sebesar 28.89%. Pada K=6 didapatkan nilai *error rate* sebesar 22.96%. Pada K=7 didapatkan nilai *error rate* sebesar 23.70%. Pada K=9 didapatkan nilai *error rate* sebesar 25.93%. Pada K=10 dan K=11 didapatkan nilai *error rate* sebesar 27.41%. Pada K=12 hingga K=15 didapatkan nilai *error*

*rate* sebesar 26.67%. Pada K=16 hingga K=19 didapatkan nilai *error rate* sebesar 31.11%. Pada K=20 didapatkan nilai *error rate* sebesar 32.59%.




Gambar 4.22 Grafik Hubungan k dengan *Error Rate* untuk T=55 K=12

Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=20 yaitu sebesar 32.59% sedangkan nilai error terkecil didapatkan saat K=6 sebesar 22.96%. Nilai konstan didapatkan pada saat K=12 hingga K=15 yaitu sebesar 26.67% dan pada K=16 hingga K=19 yaitu sebesar 31.33%.

### 4.3.5. Uji Pengaruh Variasi Data Terhadap Keoptimalan Kelompok

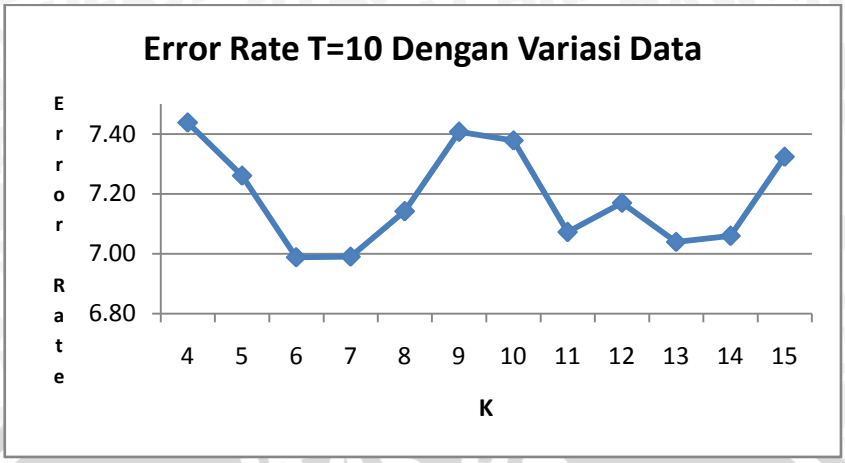
Pengujian ketiga ini untuk mengetahui pengaruh variasi data terhadap *error rate* kelompok yang terbentuk. Uji ini melibatkan 848 data latih yang berbeda dengan pengujian pertama. Nilai threshold dan nilai K yang digunakan tetap sama yaitu untuk threshold mulai dari 10, 15 hingga 55 dan untuk nilai K mulai dari 4 hingga 15. Hasil pengujian untuk threshold 10 ditampilkan pada table 4.22.

Tabel 4.22 Hasil Uji Variasi Data dengan T=10

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 10 | 4 | 7.44 |
| | 5 | 7.26 |
| | 6 | 6.99 |
| | 7 | 6.99 |
| | 8 | 7.14 |
| | 9 | 7.41 |
| | 10 | 7.38 |
| | 11 | 7.07 |
| | 12 | 7.17 |
| | 13 | 7.04 |
| | 14 | 7.06 |
| | 15 | 7.32 |

.

Berdasarkan Tabel 4.22 nilai error rate yang didapatkan berkisar antara 6% hingga 7%. Untuk nilai error ratio K=4 adalah 7.44%. Untuk K= 5 didapatkan error ratio sebesar 7.26%. Untuk K=6 dan K=7 didapatkan error ratio sebesar 6.99%. Untuk K=8 didapatkan error ratio sebesar 7.14%. Untuk K=9 didapatkan error ratio sebesar 7.41%. Untuk K=10 didapatkan error ratio sebesar 7.38%. Untuk K=11 didapatkan error ratio sebesar 7.07%. Untuk K=12 didapatkan error ratio sebesar 7.17%. Untuk K=13 didapatkan error ratio sebesar 7.04%. Untuk K=14 didapatkan error ratio sebesar 7.06%. Untuk K=15 didapatkan error ratio sebesar 7.32%.




Gambar 4.23 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=10

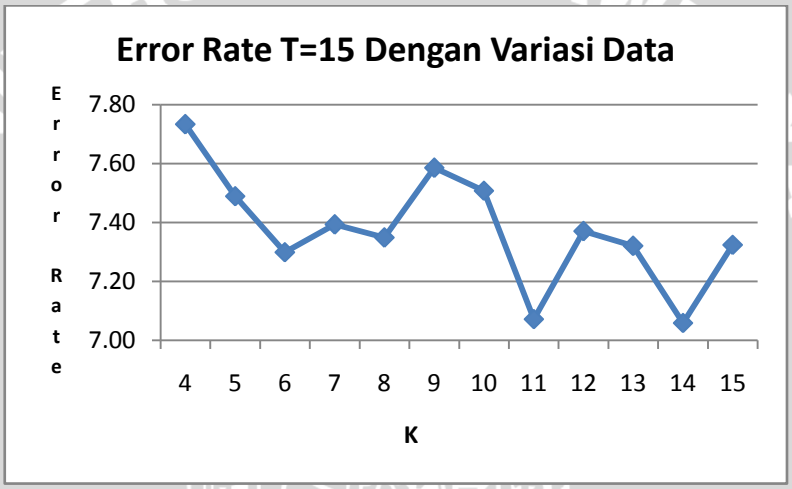
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.44%, sedangkan nilai error terkecil didapatkan saat K=6 dan K=7, yaitu sebesar 6.99 %.

Selanjutnya untuk hasil pengujian dengan threshold = 15 ditampilkan pada tabel 4.23.

Tabel 4.23 Hasil Uji Variasi Data dengan T=15

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 15 | 4 | 7.73 |
| | 5 | 7.49 |
| | 6 | 7.30 |
| | 7 | 7.39 |
| | 8 | 7.35 |
| | 9 | 7.59 |
| | 10 | 7.51 |
| | 11 | 7.07 |
| | 12 | 7.37 |
| | 13 | 7.32 |
| | 14 | 7.06 |
| | 15 | 7.32 |

Berdasarkan Tabel 4.23 nilai error rate yang didapatkan berkisar antara 7%. Untuk nilai error ratio K=4 adalah 7.73%. Untuk K= 5 didapatkan error ratio sebesar 7.49%. Untuk K=6 didapatkan error ratio sebesar 7.30%. Untuk K=7 didapatkan error ratio sebesar 7.39%. Untuk K=8 didapatkan error ratio sebesar 7.35%. Untuk K=9 didapatkan error ratio sebesar 7.59%. Untuk K=10 didapatkan error ratio sebesar 7.51%. Untuk K=11 didapatkan error ratio sebesar 7.07%. Untuk K=12 didapatkan error ratio sebesar 7.37%. Untuk K=13 didapatkan error ratio sebesar 7.32%. Untuk K=14 didapatkan error ratio sebesar 7.06%. Untuk K=15 didapatkan error ratio sebesar 7.32%.




Gambar 4.24 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=15

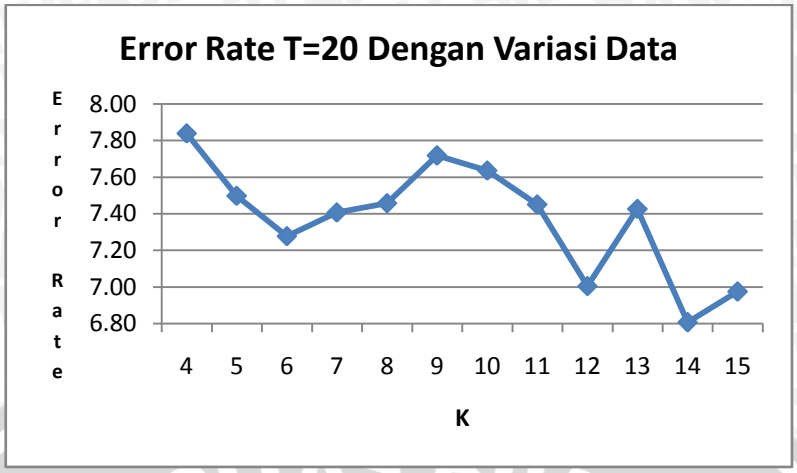
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.73%, sedangkan nilai error terkecil didapatkan saat K=14 dan K=7, yaitu sebesar 7.06 %.

Selanjutnya untuk hasil pengujian dengan threshold = 20 ditampilkan pada tabel 4.24.

Tabel 4.24 Hasil Uji Variasi Data Dengan T=20

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 20 | 4 | 7.84 |
| | 5 | 7.50 |
| | 6 | 7.28 |
| | 7 | 7.41 |
| | 8 | 7.46 |
| | 9 | 7.72 |
| | 10 | 7.64 |
| | 11 | 7.45 |
| | 12 | 7.01 |
| | 13 | 7.43 |
| | 14 | 6.81 |
| | 15 | 6.97 |

Berdasarkan Tabel 4.24 nilai error rate yang didapatkan berkisar antara 6% hingga 7%. Untuk nilai error ratio K=4 adalah 7.84%. Untuk K= 5 didapatkan error ratio sebesar 7.50%. Untuk K=6 didapatkan error ratio sebesar 7.28%. Untuk K=7 didapatkan error ratio sebesar 7.41%. Untuk K=8 didapatkan error ratio sebesar 7.46%. Untuk K=9 didapatkan error ratio sebesar 7.72%. Untuk K=10 didapatkan error ratio sebesar 7.64%. Untuk K=11 didapatkan error ratio sebesar 7.45%. Untuk K=12 didapatkan error ratio sebesar 7.01%. Untuk K=13 didapatkan error ratio sebesar 7.43%. Untuk K=14 didapatkan error ratio sebesar 6.81%. Untuk K=15 didapatkan error ratio sebesar 6.97%.




Gambar 4.25 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=20

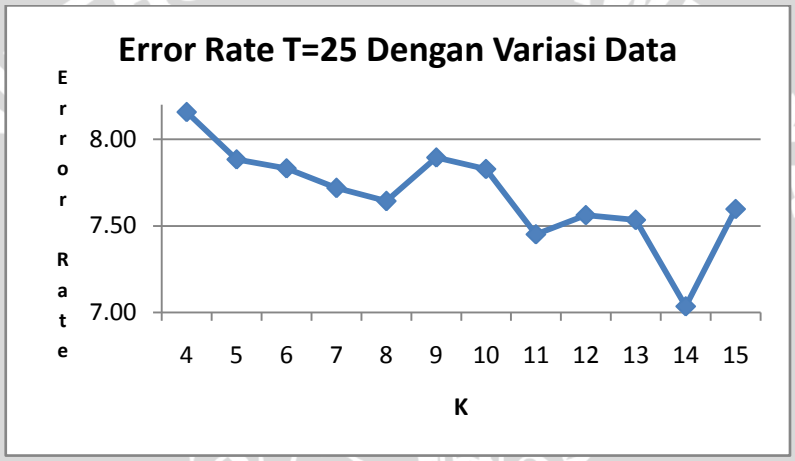
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 7.84%, sedangkan nilai error terkecil didapatkan saat K=14, yaitu sebesar 6.81 %.

Selanjutnya untuk hasil pengujian dengan threshold = 25 ditampilkan pada tabel 4.25.

Tabel 4.25 Hasil Uji Variasi Data dengan T=25

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 25 | 4 | 8.16 |
| | 5 | 7.88 |
| | 6 | 7.83 |
| | 7 | 7.72 |
| | 8 | 7.64 |
| | 9 | 7.89 |
| | 10 | 7.83 |
| | 11 | 7.45 |
| | 12 | 7.56 |
| | 13 | 7.53 |
| | 14 | 7.04 |
| | 15 | 7.60 |

Berdasarkan Tabel 4.25 nilai error rate yang didapatkan berkisar antara 7% hingga 8%. Untuk nilai error ratio K=4 adalah 8.16%. Untuk K= 5 didapatkan error ratio sebesar 7.88%. Untuk K=6 didapatkan error ratio sebesar 7.83%. Untuk K=7 didapatkan error ratio sebesar 7.72%. Untuk K=8 didapatkan error ratio sebesar 7.64%. Untuk K=9 didapatkan error ratio sebesar 7.89%. Untuk K=10 didapatkan error ratio sebesar 7.83%. Untuk K=11 didapatkan error ratio sebesar 7.45%. Untuk K=12 didapatkan error ratio sebesar 7.56%. Untuk K=13 didapatkan error ratio sebesar 7.53%. Untuk K=14 didapatkan error ratio sebesar 7.04%. Untuk K=15 didapatkan error ratio sebesar 7.60%.




Gambar 4.26 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=25

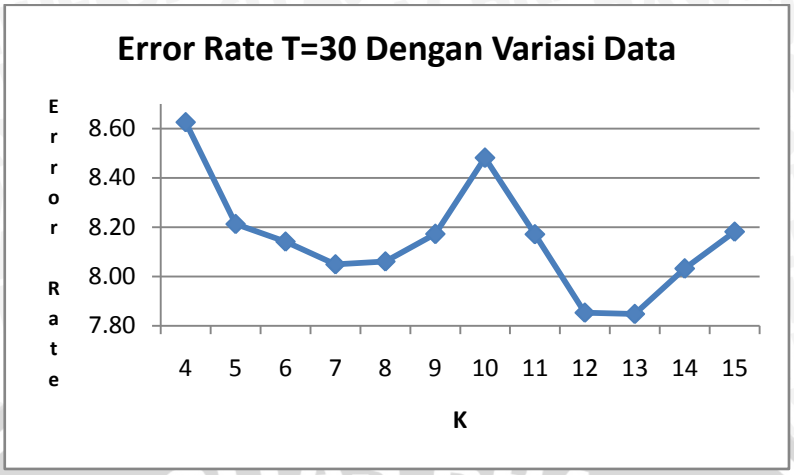
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 8.16%, sedangkan nilai error terkecil didapatkan saat K=14, yaitu sebesar 7.04 %.

Selanjutnya untuk hasil pengujian dengan threshold = 30 ditampilkan pada tabel 4.26.

Tabel 4.26 Hasil Uji Variasi Data dengan T=30

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 30 | 4 | 8.63 |
| | 5 | 8.21 |
| | 6 | 8.14 |
| | 7 | 8.05 |
| | 8 | 8.06 |
| | 9 | 8.17 |
| | 10 | 8.48 |
| | 11 | 8.17 |
| | 12 | 7.85 |
| | 13 | 7.85 |
| | 14 | 8.03 |
| | 15 | 8.18 |

Berdasarkan Tabel 4.26 nilai error rate yang didapatkan berkisar antara 7% hingga 8%. Untuk nilai error ratio K=4 adalah 8.63%. Untuk K= 5 didapatkan error ratio sebesar 8.21%. Untuk K=6 didapatkan error ratio sebesar 8.14%. Untuk K=7 didapatkan error ratio sebesar 8.05%. Untuk K=8 didapatkan error ratio sebesar 8.06%. Untuk K=9 didapatkan error ratio sebesar 8.17%. Untuk K=10 didapatkan error ratio sebesar 8.48%. Untuk K=11 didapatkan error ratio sebesar 8.17%. Untuk K=12 dan K=13didapatkan error ratio sebesar 7.85%. Untuk K=14 didapatkan error ratio sebesar 8.03%. Untuk K=15 didapatkan error ratio sebesar 8.18%.




Gambar 4.27 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=30

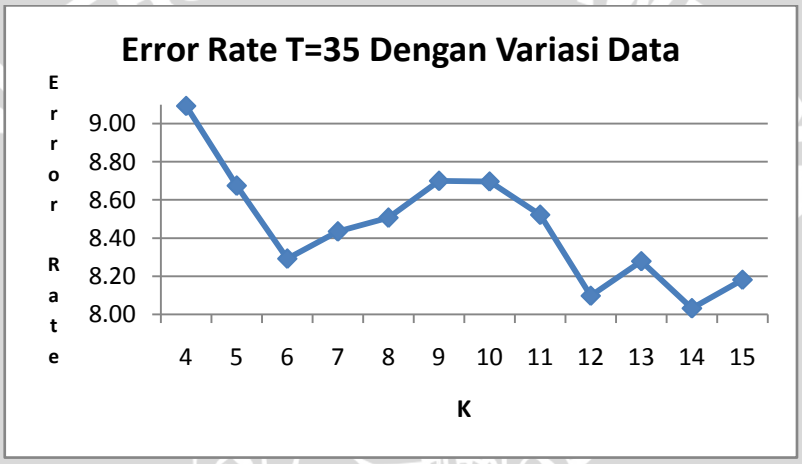
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 8.63%, sedangkan nilai error terkecil didapatkan saat K=12 dan K=13, yaitu sebesar 7.85 %.

Selanjutnya untuk hasil pengujian dengan threshold = 35 ditampilkan pada tabel 4.27.

Tabel 4.27 Hasil Uji Variasi Data dengan T=35

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 35 | 4 | 9.09 |
| | 5 | 8.67 |
| | 6 | 8.29 |
| | 7 | 8.43 |
| | 8 | 8.51 |
| | 9 | 8.70 |
| | 10 | 8.70 |
| | 11 | 8.52 |
| | 12 | 8.10 |
| | 13 | 8.28 |
| | 14 | 8.03 |
| | 15 | 8.18 |

Berdasarkan Tabel 4.27 nilai error rate yang didapatkan berkisar antara 8% hingga 9%. Untuk nilai error ratio K=4 adalah 9.09%. Untuk K= 5 didapatkan error ratio sebesar 8.67%. Untuk K=6 didapatkan error ratio sebesar 8.29%. Untuk K=7 didapatkan error ratio sebesar 8.43%. Untuk K=8 didapatkan error ratio sebesar 8.51%. Untuk K=9 dan K=10 didapatkan error ratio sebesar 8.70%. Untuk K=11 didapatkan error ratio sebesar 8.52%. Untuk K=12 didapatkan error ratio sebesar 8.10%. Untuk K=13 didapatkan error ratio sebesar 8.28%. Untuk K=14 didapatkan error ratio sebesar 8.03%. Untuk K=15 didapatkan error ratio sebesar 8.18%.




Gambar 4.28 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=35

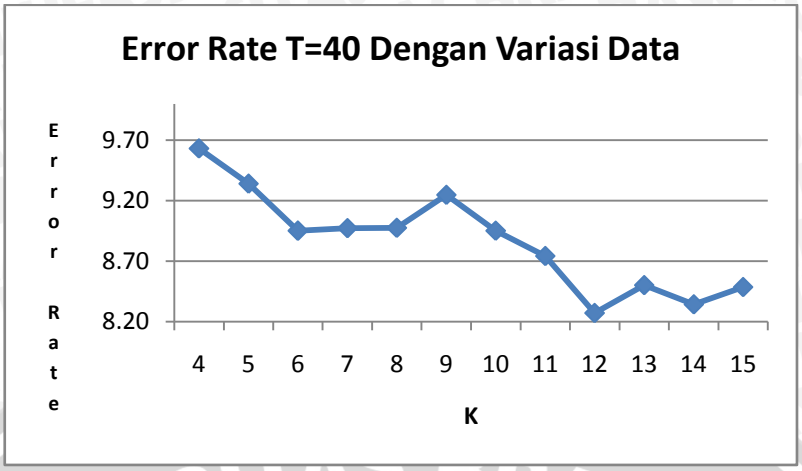
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 9.09%, sedangkan nilai error terkecil didapatkan saat K=14 dan K=7, yaitu sebesar 8.03 %.

Selanjutnya untuk hasil pengujian dengan threshold = 40 ditampilkan pada tabel 4.28.

Tabel 4.28 Hasil Uji Variasi Data dengan T=40

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 40 | 4 | 9.63 |
| | 5 | 9.34 |
| | 6 | 8.95 |
| | 7 | 8.97 |
| | 8 | 8.98 |
| | 9 | 9.25 |
| | 10 | 8.95 |
| | 11 | 8.74 |
| | 12 | 8.27 |
| | 13 | 8.50 |
| | 14 | 8.34 |
| | 15 | 8.49 |

Berdasarkan Tabel 4.28 nilai error rate yang didapatkan berkisar antara 8% hingga 9%. Untuk nilai error ratio K=4 adalah 9.63%. Untuk K= 5 didapatkan error ratio sebesar 9.34%. Untuk K=6 didapatkan error ratio sebesar 8.95%. Untuk K=7 didapatkan error ratio sebesar 8.97%. Untuk K=8 didapatkan error ratio sebesar 8.98%. Untuk K=9 didapatkan error ratio sebesar 9.25%. Untuk K=10 didapatkan error ratio sebesar 8.95%. Untuk K=11 didapatkan error ratio sebesar 8.74%. Untuk K=12 didapatkan error ratio sebesar 8.27%. Untuk K=13 didapatkan error ratio sebesar 8.50%. Untuk K=14 didapatkan error ratio sebesar 8.34%. Untuk K=15 didapatkan error ratio sebesar 8.49%.




Gambar 4.29 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=40

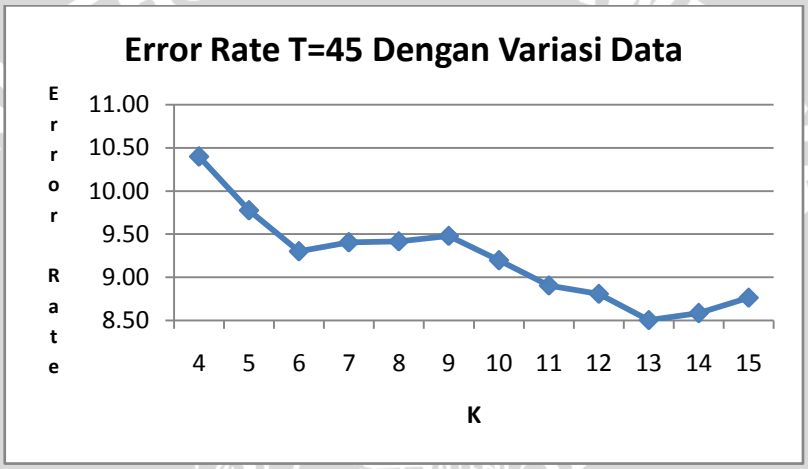
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 9.63%, sedangkan nilai error terkecil didapatkan saat K=12 yaitu sebesar 8.27 %.

Selanjutnya untuk hasil pengujian dengan threshold = 45 ditampilkan pada tabel 4.29.

Tabel 4.29 Hasil Uji Variasi Data dengan T=45

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 45 | 4 | 10.40 |
| | 5 | 9.77 |
| | 6 | 9.30 |
| | 7 | 9.40 |
| | 8 | 9.42 |
| | 9 | 9.48 |
| | 10 | 9.20 |
| | 11 | 8.90 |
| | 12 | 8.81 |
| | 13 | 8.50 |
| | 14 | 8.58 |
| | 15 | 8.76 |

Berdasarkan Tabel 4.29 nilai error rate yang didapatkan berkisar antara 8% hingga 10%. Untuk nilai error ratio K=4 adalah 10.40%. Untuk K= 5 didapatkan error ratio sebesar 9.77%. Untuk K=6 didapatkan error ratio sebesar 9.30%. Untuk K=7 didapatkan error ratio sebesar 9.40%. Untuk K=8 didapatkan error ratio sebesar 9.42%. Untuk K=9 didapatkan error ratio sebesar 9.48%. Untuk K=10 didapatkan error ratio sebesar 9.20%. Untuk K=11 didapatkan error ratio sebesar 8.90%. Untuk K=12 didapatkan error ratio sebesar 8.81%. Untuk K=13 didapatkan error ratio sebesar 8.50%. Untuk K=14 didapatkan error ratio sebesar 8.58%. Untuk K=15 didapatkan error ratio sebesar 8.76%.




Gambar 4.30 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=45

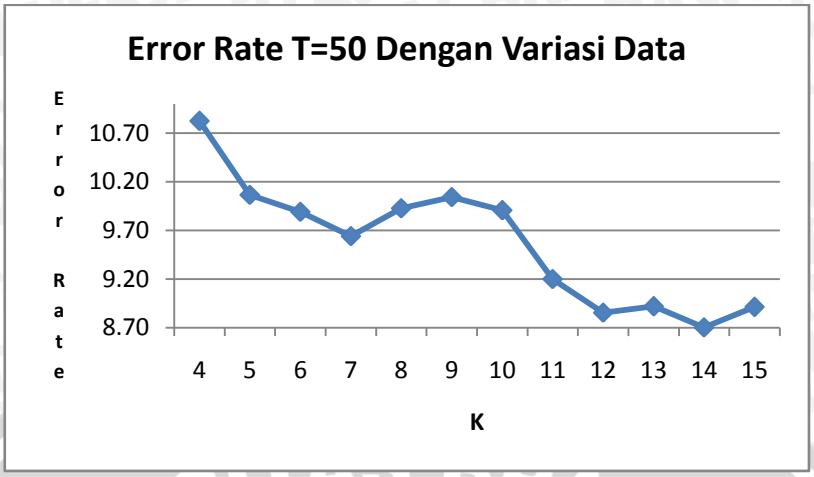
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 10.40%, sedangkan nilai error terkecil didapatkan saat K=13 yaitu sebesar 8.50 %.

Selanjutnya untuk hasil pengujian dengan threshold = 50 ditampilkan pada tabel 4.30.

Tabel 4.30 Hasil Uji Variasi Data dengan T=50

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 50 | 4 | 10.82 |
| | 5 | 10.07 |
| | 6 | 9.89 |
| | 7 | 9.64 |
| | 8 | 9.93 |
| | 9 | 10.04 |
| | 10 | 9.91 |
| | 11 | 9.20 |
| | 12 | 8.86 |
| | 13 | 8.92 |
| | 14 | 8.70 |
| | 15 | 8.91 |

Berdasarkan Tabel 4.30 nilai error rate yang didapatkan berkisar antara 8% hingga 10%. Untuk nilai error ratio K=4 adalah 10.82%. Untuk K= 5 didapatkan error ratio sebesar 10.07%. Untuk K=6 didapatkan error ratio sebesar 9.89%. Untuk K=7 didapatkan error ratio sebesar 9.64%. Untuk K=8 didapatkan error ratio sebesar 9.93%. Untuk K=9 didapatkan error ratio sebesar 10.04%. Untuk K=10 didapatkan error ratio sebesar 9.91%. Untuk K=11 didapatkan error ratio sebesar 9.20%. Untuk K=12 didapatkan error ratio sebesar 8.86%. Untuk K=13 didapatkan error ratio sebesar 8.92%. Untuk K=14 didapatkan error ratio sebesar 8.70%. Untuk K=15 didapatkan error ratio sebesar 8.91%.




Gambar 4.31 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=50

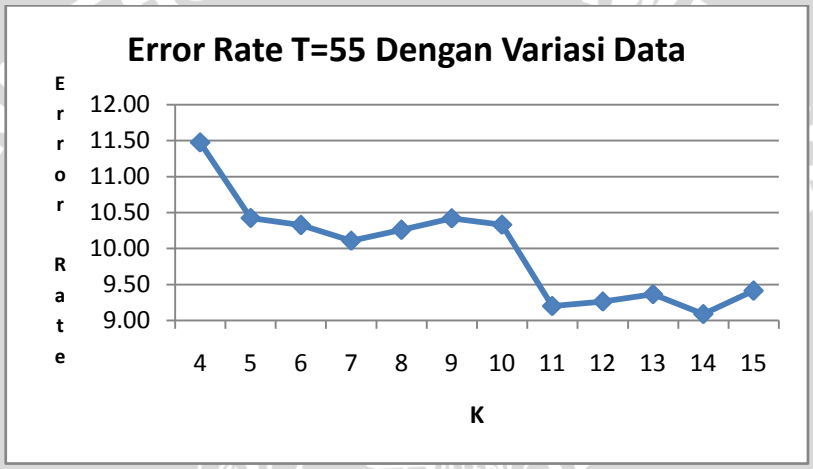
Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 10.82%, sedangkan nilai error terkecil didapatkan saat K=14 yaitu sebesar 8.70 %.

Selanjutnya untuk hasil pengujian dengan threshold = 55 ditampilkan pada tabel 4.31.

Tabel 4.31 Hasil Uji Variasi Data dengan T=55

| Threshold | K | *Error rate*(%) |
|---|---|---|
| 55 | 4 | 11.48 |
| | 5 | 10.43 |
| | 6 | 10.33 |
| | 7 | 10.11 |
| | 8 | 10.26 |
| | 9 | 10.42 |
| | 10 | 10.33 |
| | 11 | 9.20 |
| | 12 | 9.26 |
| | 13 | 9.37 |
| | 14 | 9.09 |
| | 15 | 9.41 |

Berdasarkan Tabel 4.31 nilai error rate yang didapatkan berkisar antara 8% hingga 10%. Untuk nilai error ratio K=4 adalah 11.48%. Untuk K= 5 didapatkan error ratio sebesar 10.43%. Untuk K=6 didapatkan error ratio sebesar 10.33%. Untuk K=7 didapatkan error ratio sebesar 10.11%. Untuk K=8 didapatkan error ratio sebesar 10.26%. Untuk K=9 didapatkan error ratio sebesar 10.42%. Untuk K=10 didapatkan error ratio sebesar 10.33%. Untuk K=11 didapatkan error ratio sebesar 9.20%. Untuk K=12 didapatkan error ratio sebesar 9.26%. Untuk K=13 didapatkan error ratio sebesar 8.37%. Untuk K=14 didapatkan error ratio sebesar 9.09%. Untuk K=15 didapatkan error ratio sebesar 9.41%.




Gambar 4.32 Grafik Hubungan Nilai K Dengan Variasi Data Terhadap Error Ratio Untuk T=55

Dari grafik diatas dapat dilihat bahwa nilai error terbesar didapatkan pada saat K=4 yaitu sebesar 11.48%, sedangkan nilai error terkecil didapatkan saat K=14 yaitu sebesar 9.09 %.

## 4.4. Analisa Hasil

### 4.4.1. Analisa Hasil Pengaruh Nilai k pada Proses Pengelompokan Terhadap keoptimalan Kelompok

Berdasarkan pengujian nilai K pada proses pengelompokan terhadap keoptimalan kelompok dapat diketahui bahwa nilai k berpengaruh pada nilai *error ratio* yang dihasilkan. Semakin tinggi nilai k, maka akan didapatkan semakin kecil nilai *error ratio* yang dihasilkan. Jadi dapat dikatakan bahwa nilai k berbanding terbalik dengan nilai *error ratio.* Dapat dikatakan pula, semakin tinggi

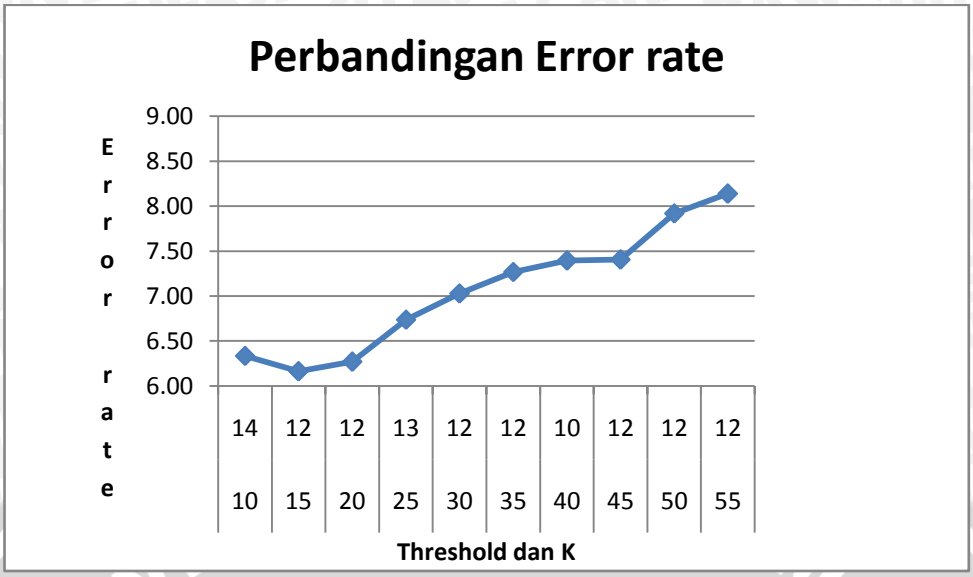
threshold yang digunakan maka semakin besar pula nilai *error rate* yang dihasilkan. Berarti nilai threshold berbanding lurus dengan nilai *error rate* yang dihasilkan. Untuk grafik nilai *error rate* masing – masing threshold dapat dilihat pada gambar 4.34.

Untuk nilai *error rate* terkecil yang didapatkan dari masing – masing threshold dapat dilihat pada tabel 4.32.

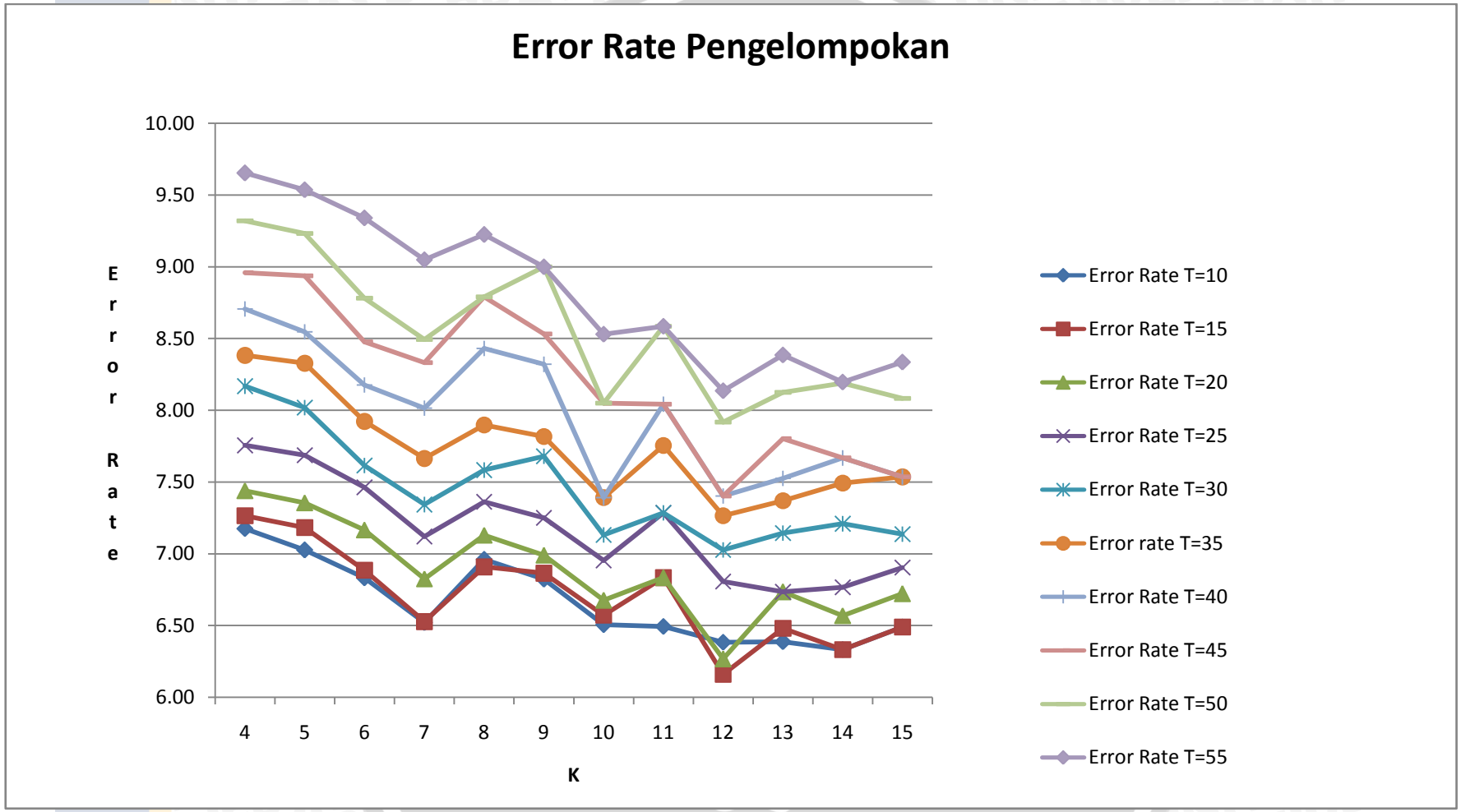
Tabel 4.32 Nilai *Error Rate* Terkecil untuk Masing - Masing Threshold

| Threshold | K | *Error rate* (%) |
|---|---|---|
| 10 | 14 | 6.33 |
| 15 | 12 | 6.16 |
| 20 | 12 | 6.27 |
| 25 | 13 | 6.73 |
| 30 | 12 | 7.03 |
| 35 | 12 | 7.27 |
| 40 | 10 | 7.39 |
| 45 | 12 | 7.40 |
| 50 | 12 | 7.92 |
| 55 | 12 | 8.14 |

Nilai *error rate* terkecil didapatkan pada K=12 dengan threshold 15, yaitu sebesar 6.16%.Pada gambar 4.33 dapat dilihat jika rata – rata keoptimalan kelompok didapatkan pada saat K=12 di tiap thresholdnya.




Gambar 4.33 Grafik Perbandingan *Error Rate* Terkecil pada Masing - Masing Threshold




Gambar 4.34 Grafik Perbandingan *Error Rate*

### 4.4.2. Analisa Hasil Pengaruh Nilai k Pada Proses Klasifikasi Terhadap Penentuan Kelas

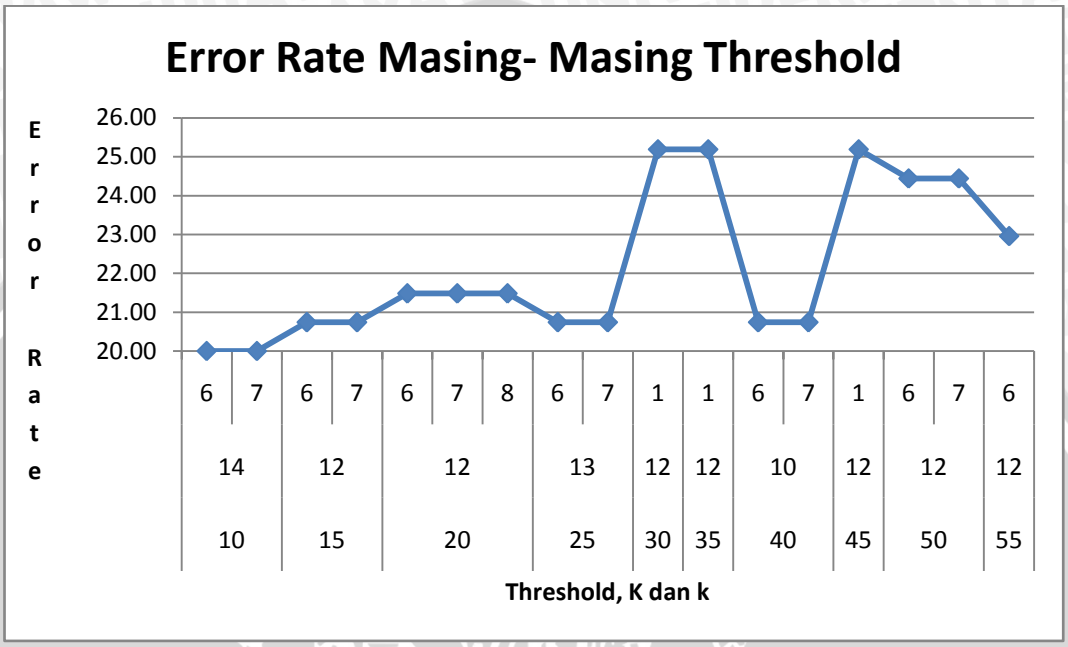
Berdasarkan pengujian nilai k pada proses klasifikasi yang diuji pada setiap nilai K pengelompokan yang optimal(nilai *error rate* terkecil) dapat diketahui bahwa grafik memiliki kecenderungan kenaikan pada saat k= 2 dan mengalami kecenderungan penurunan drastis pada saat k=6. Nilai konstan yang didapat pada tiap threshold nya berbeda – beda, hal ini disebabkan oleh jumlah anggota kelompok yang dimiliki setiap kelompoknya. Sebagai contoh pada threshold 10, nilai konstan didapatkan pada saat k=10 hingga k=20 hal ini dikarenakan jumlah maksimum anggota kelompok yang dimiliki untuk setiap kelompoknya adalah 10.

Pada tabel 4.33 menunjukan nilai *error rate* untuk masing – masing nilai threshold.

Tabel 4.33 Tabel Perbandingan Nilai *Error Rate* Masing - Masing Threshold

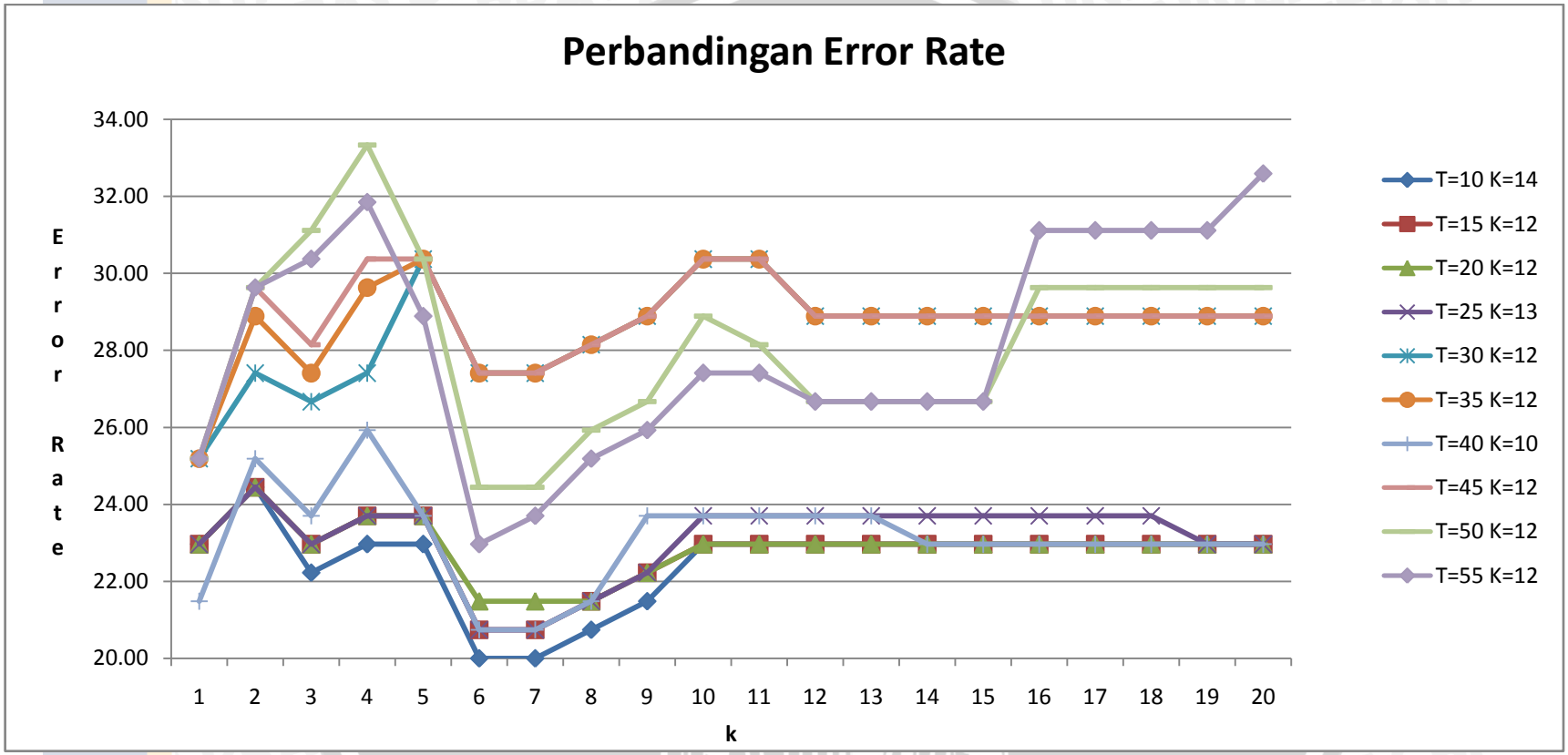
| Threshold | K | k | *Error Rate*(%) |
|---|---|---|---|
| 10 | 14 | 6 | 20.00 |
| | | 7 | |
| 15 | 12 | 6 | 20.74 |
| | | 7 | |
| 20 | 12 | 6 | 21.48 |
| | | 7 | |
| | | 8 | |
| 25 | 13 | 6 | 20.74 |
| | | 7 | |
| 30 | 12 | 1 | 25.19 |
| 35 | 12 | 1 | |
| 40 | 10 | 6 | 20.74 |
| | | 7 | |
| 45 | 12 | 1 | 25.19 |
| 50 | 12 | 6 | 24.44 |
| | | 7 | |
| 55 | 12 | 6 | 22.96 |

Pada Gambar 4.35 dapat dilihat bahwa nilai *error rate* terkecil rata – rata didapatkan pada saat k=6 atau k=7. Memiliki kenaikan yang signifikan pada saat k=1. Dengan nilai *error ratio* terkecil didapatkan pada saat T=10 dan K=14.




Gambar 4.35 Grafik Perbandingan Nilai *Error Rate* Terkecil Masing - Masing Threshold

Gambar 4.36 menunjukan bahwa grafik perbandingan untuk setiap threshold dengan K yang optimal memiliki nilai yang naik seiring dengan kenaikan threshold. Dengan kata lain, nilai threshold berbanding lurus dengan nilai *error rate* klasifikasi. Pergerakan grafik yang paling stabil didapatkan pada saat T=15 dan K=12, dimana T=15 dan K=12 merupakan kelompok yang paling optimal yang didapatkan pada pengujian sebelumnya.




Gambar 4.36 Grafik Perbandingan *Error Rate* Klasifikasi

### 4.4.3. Analisa Hasil Pengaruh Variasi Data Terhadap Keoptimalan Pengelompokan

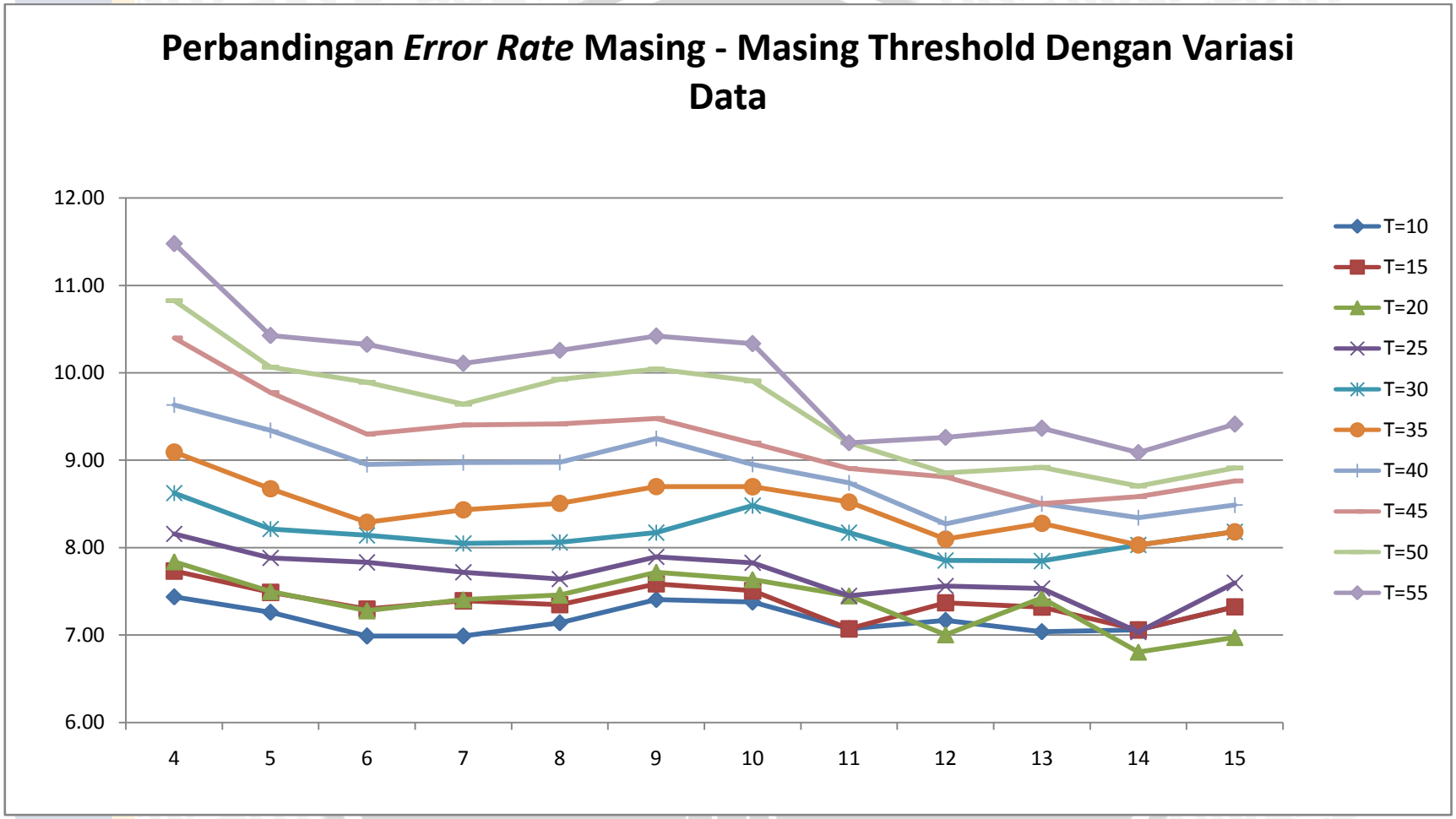
Berdasarkan pengujian variasi data pada proses pengelompokan terhadap nilai *error rate* yang dihasilkan dapat diketahui bahwa variasi data berpengaruh pada nilai *error rate* yang dihasilkan. Data yang semakin menyebar akan mempengaruhi nilai *error rate* yang dihasilkan. Sama hal nya dengan hasil uji terhadap data pada pengujian I, hasil yang didapatkan *error rate* berbanding lurus dengan kenaikan threshold. Semakin tinggi threshold, maka akan semakin tinggi nilai *error rate* nya.

Untuk nilai *error rate* terkecil yang didapatkan dari masing – masing threshold dapat dilihat pada tabel 4.34.

Tabel 4.34 Nilai Error Rate Terkecil untuk Masing - Masing Threshold Pada Variasi Data

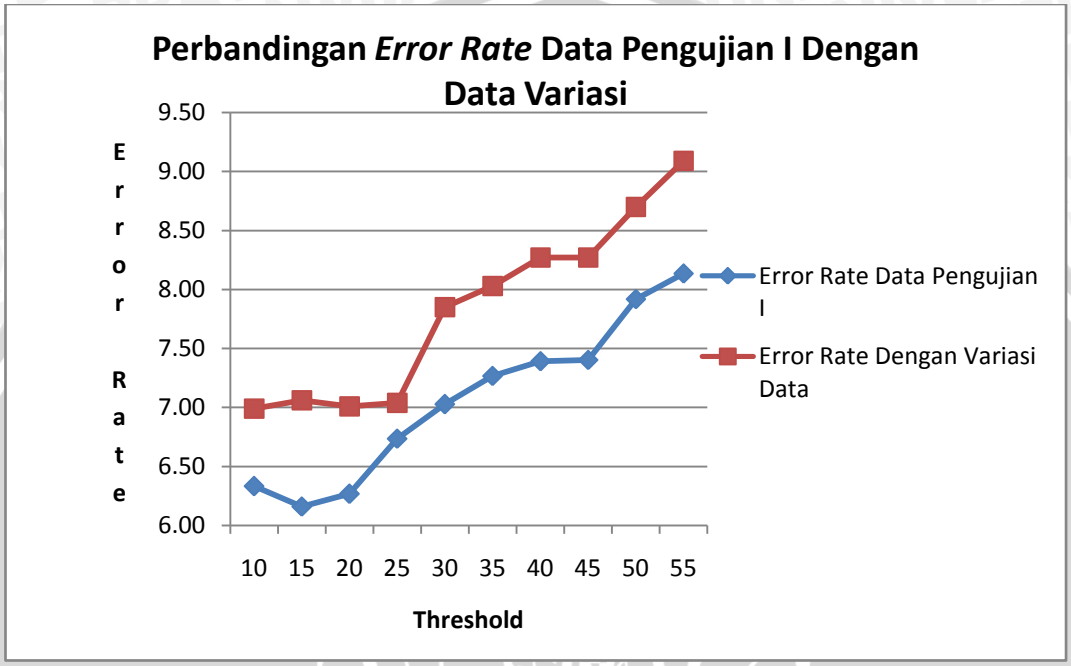
| Threshold | K | *Error rate* (%) |
|---|---|---|
| 10 | 6 | 6.99 |
| 15 | 14 | 7.06 |
| 20 | 12 | 7.01 |
| 25 | 14 | 7.04 |
| 30 | 12 | 7.85 |
| 35 | 14 | 8.03 |
| 40 | 12 | 8.27 |
| 45 | 13 | 8.27 |
| 50 | 14 | 8.70 |
| 55 | 14 | 9.09 |

Nilai *error rate* terkecil didapatkan pada K=6 dengan threshold 10, yaitu sebesar 6.99%. Untuk grafik nilai *error rate* masing – masing threshold dapat dilihat pada gambar 4.37.




Gambar 4.37 Grafik Perbandingan *Error Rate* Masing -Masing Threshold Dengan Variasi Data

Nilai *error rate* terkecil yang dihasilkan oleh kedua data pengujian menunjukan kenaikan yang tidak terlalu jauh dengan rentang *error rate* berkisar antara 6% - 9%. Grafiknya dapat dilihat pada gambar 4.38.




Gambar 4.38 Grafik Perbandingan *Error Rate* Data Pengujian I Dengan Data Variasi

# BAB V
# KESIMPULAN DAN SARAN

Pada bab ini akan dipaparkan kesimpulan dan saran dari hasil implementasi serta pengujian terhadap sistem yang telah dibangun. Kesimpulan yang diberikan merupakan jawaban atas masalah penelitian yang diambil. Saran merupakan himbauan untuk penelitian yang akan datang berdasarkan hasil penelitian yang telah dilakukan saat ini.

## 5.1. Kesimpulan

Dari hasil uji dan analisis yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Berdasarkan hasil implementasi didapatkan sebuah model yang dapat menentukan jenis kanker berdasarkan pengelompokan yang optimal menggunakan *K-Means.*
2. Nilai K pada saat proses pengelompokan yang paling optimal didapatkan pada saat T=15 dengan K=12 dengan nilai *error rate* sebesar 6.16%. Dengan rata – rata K optimal untuk setiap threshold nya adalah 12.
3. Variasi data berpengaruh terhadap nilai *error rate* kelompok. Data yang semakin menyebar akan menyebabkan nilai *error rate* yang semakin tinggi pula. K optimal didapatkan pada saat T=10 dengan K=6.
4. Nilai k pada saat proses klasifikasi berdasarkan pengelompokan yang optimal menggunakan *K-Means* didapatkan pada saat T=10 dengan k=6 dan k=7. Tetapi memiliki kestabilan pada saat pengelompokan optimal, yaitu T=15 dengan k=6 dan k=7.

## 5.2. Saran

Beberapa saran yang dapat disampaikan untuk penelitian selanjutnya yaitu:

1. Selanjutnya dapat diteliti pengaruh jumlah data terhadap keakuratan proses pengelompokan maupun akurasi.
2. Untuk lebih mengoptimalkan hasil, pohon yang terbentuk dapat lebih diseimbangkan.

3. Penentuan centroid awal akan berpengaruh terhadap hasil, oleh sebab itu penelitian lebih lanjut tentang hal tersebut dapat dilakukan.
4. Menggunakan panjang sekuen yang tidak sama sehingga akan terdapat GAP pada sekuen tersebut dan dapat ditangani dengan metode *sequence alignment* yang terdapat dalam disiplin ilmu bioinformatika.
5. Karena metode *K-Means* bergantung pada penentuan centroid awal, maka untuk metode pengelompokan dapat menggunakan metode lain yang digabungkan dengan metode *KNN*.
6. Dapat dilakukan pengujian terhadap metode KNN tanpa melakukan pengelompokan terlebih dahulu pada data latih yang digunakan sebagai perbandingan keakuratan klasifikasi.

# DAFTAR PUSTAKA

Barakhbah. (2006). *Cluster Analysis.* Surabaya: Jurusan Teknologi Informasi Politeknik Elektronika Negeri Surabaya.

Barnes, M. R., & Gray, I. C. (2003). *Bioinformatics For Geneticts.* Chichester, England: John Wiley & Sons Ltd.

Bramer, M. (2007). *Principles of Data Mining.* London: Springer-Verlag.

Brown, T., & Brown, D. (2005). *Mutagenesis and DNA repai*. Dipetik Mei 17, 2012, dari http://www.atdbio.com

Cesario, A., & Frederick, M. B. (2011). *Cancer Systems Biology, Bioinformatics and Medicine : Research and Clinical Applications.* New York: Springer Dordrecht Heidelberg London.

Chen, J. Y., & Lonardi, S. (2010). *Biological Data Mining.* Chapman & Hall/CRC Broken Sound Parkway NW.

*Codon exercise 1*. (t.thn.). Dipetik May 16, 2012, dari http://www.cs.au.dk

Colet, E. (t.thn.). Dipetik Mei 15, 2012, dari http://www.tgc.com/dsstar/00/0704/101861.html

*Euclidean and Euclidean Square*. (2004). Dipetik Mei 14, 2012, dari IOS Improved Outcomes Software: http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm

Gan, G., Ma, C., & Wu, J. (2007). *Data Clustering, Theory, Algorithms, and Applications.* The American Statistical Association and the Society for Industrial and Applied Mathematics.

Gayathri, K., & Marimuthu, A. (2011). An Improved KNN Text Classification algorithm by using K-Mean Clustering. *nternational Journal of Computing*

*Technology and Information Security* , 73-76.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann Publishers.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining.* Massachusetts London: The MIT Press Cambridge.

Henderson, B. E., Ponder, B., & Ross, R. K. (2003). *Genes and Cancer.* Oxford University Press Inc.Hormones.

Holbert, D. (2002). *Scoring Matrices : The Arrays Used to Find and Evaluate Protein Homologies.* Dipetik Mei 15, 2012, dari http://biochem118.stanford.edu/Projects/2002/holbert.pdf

Jones, N. C., & Pevzner, P. A. (2004). *An Introduction to Bioinformatics Algorithms.* Massachusetts London: The MIT Press Cambridge.

Kadous, M. W. (2002). *Doing the search*. Dipetik Mei 17, 2012, dari http://www.cse.unsw.edu.au/~waleed/phd/html/node58.html#fig:kmeans.alg

Kantardzic, M. (2003). *Data Mining - Concepts, Models, Methods, and Algorithms.* New Jersey: John Wiley & Sons, Inc.

Kaufman, L., & Rousseeuw, P. J. (2005). *Finding Groups in Data : An Introduction to Cluster Analysis.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Keedwell, E., & Narayanan, A. (2005). *Intelligent Bioinformatics : The Application of Artificial Intelligence Techniques to Bioinformatics Problems.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A Fuzzy K-Nearest Neighbor. *IEEE Transactions On System, Man And Cybernetics , SMC-15 NO 4.*

Kusnawi. (2007). Dipetik Mei 15, 2012, dari http://p3m.amikom.ac.id/p3m/56%20-%20PENGANTAR%20SOLUSI%20DATA%20MINING.pdf

Larose, D. T. (2005). *Discovering Knowledge in Data : an Introduction to Data Mining.* New Jersey: John Wiley & Sons, Inc.

Lengauer, T. (2007). *Bioinformatics From Genomes to Therapies* (Vol. III).

WILEY-VCH Verlag GmbH & Co KGaA Weinheim.

Mitra, S., & Acharya, T. (2003). *Data Mining Multimedia, Soft Computing, and Bioinformatics.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Mitra, S., & Acharya, T. (2003). *Data Mining Multimedia, Soft Computing, and Bioinformatics.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Murray, R. K., Granner, D. K., & Rodwell, V. W. (2006). *Harper's Illustrated Biochemistry* (27 ed.). The McGraw-Hill Companies inc.

Nugraha, D. (2006). *Diagnosis Gangguan Sistem Urinari pada Anjing dan Kucing menggunakan VFI 5*. Bandung: IPB.

Parthasarathy, S. (2007). *Building Genetic Medicine : technology, breast cancer, and the comparative politics of health care.* Massachusetts London: The MIT Press Cambridge.

Pedrycz, W. (2005). *Knowledge-based Clustering : from Data to Information Granules.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Phyu, T. N. (2009). Survey of Classification Techniques in Data Mining. *Proceedings of the International MultiConference of Engineers and Computer Scientists, I.* Hong Kong.

Pusztai, L., Lewis, C., & Yap, E. (1996). *Cell Proliferation in Cancer-Regulation Mechanisms of Neoplastic Cell Growth.* Oxford: Oxford University Press.

Ramakrishna, R., & Gehrke, J. (2003). *Database Management System* (3rd ed.). The McGraw-Hill Companies Inc.

Sayad, S. (2012). *An Introduction to Data Mining.* University of Toronto.

Schenkel, A., & Hätinen, T. (2006). *Pairwise and Multiple Sequence Alignments.* Bioinformatics group. Institute of Biotechnology. University of Helsinki.

Soussi, T. (t.thn.). *p53 Mutation*. Dipetik Mei 14, 2012, dari The TP53 Web Site: http://p53.free.fr

Sullivan, R. (2012). *Introduction to Data Mining for The Life Science.* London: Springer New York Dordrecht Heidelberg.

Susanto, S., & Suryadi, D. (2010). *Pengantar Data Mining : Menggali*

*Pengetahuan dari Bongkahan Data.* Yogyakarta: Penerbit Andi.

Sutabri, T. (2005). *Sistem Informasi Manajemen.* Yogyakarta: Penerbit Andi.

Sutabri, T. (2005). *Sistem Informasi Manajemen.* Yogyakarta: Penerbit Andi.

Syaifudin, M. (2007). Gen Penekan Tumor TP53, Kanker dan Radiasi Pengion. *IPTEK Ilmiah Populer* , 119-128.

Taft, M., Krishnan, R., Hornick, M., Muhkin, D., Tang, G., Shiby, et al. (2005). *Oracle Data Mining Concepts.* Oracle.

Tjay, & Raharja. (1999). *Obat - Obat Penting: Khasiat, Penggunaan dan Efek - Efek Samping.* Jakarta: PT. Elex Media Kompotindo Kelompok Gramedia.

Wahyuni, M. (2004). *Sistem Berkas.* Yogyakarta: Penerbit Andi.

Wang, J. T., Zaki, M. J., Toivonen, H. T., & Shasha, D. (2006). *Data mining in bioinformatic (Advanced information and knowledge processing).* Berlin Heidelberg: Springer London.

WD, T., Muller, B. E., HK, H., & Harri. (2004). *Pathology & Genetics of Tumours of The Lung, Pleura, Thymus and Heart.* Lyon: IARC press—WHO.

*What is a mutation*. (2012). Dipetik May 15, 2012, dari Learn.Genetics.

Xu, R., & Wunsch, D. (2009). *Clustering.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Yuwono, T. (2005). *Biologi Molekular.* Jakarta: Penerbit Erlangga.