Penentuan Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Modified K-Nearest Neighbor (MKNN)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Arintha Retwitasari
NIM : 0910963107

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

Penentuan Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Modified K-Nearest Neighbor (MKNN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Arintha Retwitasari
NIM : 0910963107

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Drs. Marji, M.T	Widodo, S.Si, M.Si, Ph. D Med Sc.
NIK : 19670801 199203 1 001	NIK : 19730811 200003 1 002

Mengetahui
Ketua Program Studi Teknik Informatika dan Ilmu Komputer

Drs. Marji, M.T
NIK : 19670801 199203 1 001

PERNYATAAN ORISINALITAS

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 disitasi dalam naskah ini dan disebutkan dalam 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, 26 Januari 2016

Arintha Retwitasari
NIM: 0910963107

KATA PENGANTAR

Puji syukur atas kehadirat Allah SWT atas rahmat dan karunia yang telah diberikan-Nya, sehingga skripsi yang berjudul "**Penentuan Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Modified K-Nearest Neighbor (MKNN)"** telah disusun dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Tenik Informatika dan Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Pada kesempatan ini penulis mengucapkan terima kasih atas bantuan dan dukungan dari banyak pihak untuk menyelesaikan skripsi ini. Untuk itu penulis ingin menyampaikan rasa terima kasih kepada:

1. Drs. Marji, M.T selaku dosen Pembimbing I dan Widodo, S.Si, M.Si, Ph.D Med Sc. Selaku dosen Pembimbing II, yang telah bijaksana dan sabar membimbing dan menyalurkan ilmu kepada penulis dalam penyusunan skripsi ini.
2. Ir. Sutrisno, M.T selaku Ketua Program Studi Tenik Informatika dan Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
3. Hendarmawan, S.Kom selaku dosen penasehat akademik akademik
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Fakultas Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
5. Segenap staff dan karyawan di Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
6. Secara khusus penulis ingin mengucapkan terima kasih kepada Papa Dwi dan Almh. Mama Etik, Adekku Angga, Mbak Nia, Bude Wiwik, Kakek dan Nenek, serta seluruh keluarga tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.
7. Sahabat penulis khususnya Cahyo, Ira, Ayu, Yanita, Hardika, Aldi, Diantika, Melati, Nanik, Hadi ,Afief, Adi, Hadi, Bonita, Defina, Nining yang telah memberi dukungan, doa serta nasihat.
8. Teman-teman ilmu computer 2009, serta kakak dan adik tingkat di Ilmu Komputer.
9. Semua pihak yang terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu.

Semoga segala kebaikan dan pertolongan semuanya mendapatkan berkah dari Allah SWT. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari kesempurnaan, maka saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Penulis berharap skripsi ini

dapat bermanfaat dan berguna bagi semua pihak, baik penulis maupun pembaca, dan semoga Allah SWT meridhoi dan dicatat sebagai ibadah. Amin.

Malang, 7 Januari 2016

Penulis

HALAMAN PERSEMBAHAN

Alhamdulillah, terima kasih kepada Allah SWT yang telah memudahkan penulis dalam proses pengerjaan skripsi ini. Puji syukur selalu penulis panjatkan kehadirat-Nya. Skripsi ini khusus penulis persembahkan untuk para pembaca yang bersedia membaca skripsi telah penulis susun, semoga setelah membaca skripsi ini pembaca mendapatkan ide dalam penyusunan skripsinya.

ABSTRAK

Arintha Retwitasari. 2016. Penentuan Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Modified K-Nearest Neighbor (MKNN). Skripsi Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing:

Kanker merupakan salah satu jenis penyakit yang sangat ditakuti oleh banyak orang di dunia. Kanker merupakan kumpulan sel yang abnormal tidak terkendali dan secara terus menerus tumbuh. Sel kanker mampu menyerang jaringan lain yang sehat, merusak jaringan tersebut dan tumbuh subur di atas jaringan lain. Penyebab kanker sampai saat ini masih sulit untuk diketahui secara pasti karena merupakan gabungan dari sekumpulan faktor, salah satunya adalah faktor internal. Faktor internal disebabkan karena adanya mutasi gen. Mutasi gen yang paling banyak menjadi penelitian kanker pada manusia adalah gen p53. Gen p53 terjadi karena adanya mutasi DNA. Jika DNA ini mengalami mutasi, maka susunan protein yang ada akan berubah. Mutasi DNA menghasilkan protein p53 mutan dan hilangnya fungsi p53 *wild type (normal)* untuk mengendalikan siklus sel. Bila gen p53 ini tidak berfungsi dengan baik maka perkembangbiakan sel tidak dapat terkendali dan menimbulkan kanker.

Pada skripsi ini membahas tentang penerapan algoritma MKNN untuk penentuan jenis kanker berdasarkan struktur protein. Algoritma MKNN merupakan salah satu metode klasifikasi yang merupakan bagian dari data mining dimana objek dikelaskan berdasarkan kemunculan kelas terbanyak pada data latih. Modifikasi dari metode ini bertujuan untuk mengatasi kelemahan mengenai jarak data dengan *weight* yang memiliki banyak masalah dalam outlier pada metode KNN. Penelitian ini dilakukan dengan menggunakan 3 data latih yang berbeda yaitu 100, 150 dan 200. Sedangkan untuk data uji digunakan 3 data uji yang berbeda yaitu 60, 70 dan 80. Pengujian dilakukan 3 tahapan yaitu pengujian pengaruh data latih terhadap tingkat akurasi, pengujian pengaruh data uji terhadap tingkat akurasi dan pengujian pengaruh nilai k terhadap tingkat akurasi. Akurasi tertinggi yang diperoleh dari pengujian ini adalah sebesar 52,57% pada saat data latih 100 dan data uji 70.

Kata kunci : Kanker, Protein p53, Klasifikasi, *Modified K-Nearest Neighbor*

ABSTRACT

Arintha Retwitasari. 2016. Cancer type decision according its Protein Structure using Modified K=Nearest Neighbor (MKNN) algorithm. Skripsi Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing:

Cancer is a disease which consider a dreadful one for most people in this world. Cancer is a group of abnormal cells which grows uncontrollable. Cancer cells can attack another normal tissue, destroy them, and even living upon them. What causing cancer is still difficult to detect correctly because it is occurred due to some factors, one of them is internal factor. Internal factor caused by genes mutation. Genes which become the most used subject for human cancer research is p53 genes. P53 genes occurred due to DNA mutation. If the DNA are mutated, then its protein structure will be changed. This DNA mutation produce mutant p53 protein causing malfunction for p53 normal type to control cell cycle. If this p53 genes do not functioned well, therefore cells breeding become uncontrollable causing cancer.

This research discuss about applying MKNN algorithm to determine cancer type based on its protein structure. MKNN algorithm is one of classification method which is part of data mining where object is being classified based on the number of class occurrence on data training. Modification of this method is used to overcome weakness between data length and weight which causing outlier problem in normal KNN method. This research use 3 different data training divided by its number which are 100. 150, and 200. There are 3 different data testing also divided by its number which are 60, 70, and 80. Experimental testing for this research will take 3 phases, the first one is to test the influence of data training against accuracy ratio, second is influence of data testing against accuracy ratio, and the third is influence of k value against accuracy ratio. Among those experimental testing phase, the highest accuracy ratio is 52,57% where in condition with 100 data training and 70 data testing.

Key word : Cancer, p53 Protein, Classification, Modified K-Nearest Neighbor

Daftar Isi

Daftar Tabel

Daftar Gambar

Daftar Source Code

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kanker merupakan salah satu jenis penyakit yang sangat ditakuti oleh banyak orang di dunia. Kanker termasuk suatu golongan penyakit yang ditimbulkan oleh sel tunggal yang tumbuh abnormal dan tidak terkendali. Kanker bisa terjadi dimana saja, dari berbagai jaringan, dalam berbagai organ. Sel-sel kanker membentuk suatu massa dari jaringan ganas yang menyusup ke jaringan di dekatnya dan bisa menyebar ke seluruh tubuh. Setelah dilakukan diagnosis, penyakit kanker biasanya dirawat dengan kemoterapi [MUL-13].

Penyebab kanker sampai saat ini masih sulit untuk diketahui secara pasti karena merupakan gabungan dari sekumpulan faktor, salah satunya disebabkan karena adanya perubahan genetik yang dapat menyebabkan mutasi gen. Gen p53 adalah perubahan genetik yang paling umum ditemukan pada kanker manusia. Pada sel normal didapatkan protein p53 *wild type (normal)* yang berfungsi mengendalikan replikasi DNA. Gen p53 terjadi karena adanya mutasi DNA. Jika DNA ini mengalami mutasi, maka susunan protein yang ada akan berubah. Mutasi DNA menghasilkan protein p53 mutan dan hilangnya fungsi p53 *wild type (normal)* untuk mengendalikan siklus sel.

Bioinformatika, sesuai dengan asal katanya yaitu “bio” dan “informatika”, adalah gabungan antara ilmu biologi dan imu teknik informasi (TI). Pada umumnya, bioinformatika didefinisikan sebagai aplikasi dari alat komputasi dan analisa untuk menangkap dan mengiterprestasikan data-data biologi. Kemajuan ilmu bioinformatika ini lebih didesak lagi oleh genome project dan menghasilkan tumpukan informasi gen dari berbagai makhluk hidup, mulai dari makhluk hidup tingkat rendah sampai makhluk hidup tingkat tinggi. [UTA-03].

Data mining didefinisikan sebagai proses otomatis mengekstrak suatu informasi dari sekumpulan data yang berjumlah besar. Salah satu aplikasi dari penerapan data mining di bioinformatika ini adalah pengembangan industri farmasi dan kedokteran [NUG-03]. Data mining berfungsi untuk mengklasifikasi sebuah data ke dalam kategori yang sudah ditentukan sebelumnya.

Metode klasifikasi pada data mining yang digunakan antara lain adalah K-Nearest Neighbor, ID3, C45, Bayesian dan beberapa metode yang lainnya. Metode K-Nearest Neighbor (KNN) merupakan salah satu metode data mining yang digunakan untuk klasifikasi data [HUD-13]. K-Nearest Neighbor (KNN) adalah suatu pendekatan klasifikasi yang mencari semua data latih yang relatif mirip dengan data uji [ARA-12]. Metode KNN memiliki kelebihan yaitu tangguh terhadap *training data* yang memiliki *noise* dan efektif *apabila training data* yang digunakan berjumlah besar. Kelemahan

dari KNN sendiri adalah KNN perlu menentukan nilai k (jumlah tetangga terdekat), berdasarkan jarak tidak jelas mengenai jenis jarak yang digunakan.

Dari kelemahan K-Nearest Neighbor (KNN) maka dikembangkan metode *Modified K-Nearest Neighbor* (MKNN). *Modified K Nearest Neighbor* (*MKNN*) merupakan modifikasi dari metode KNN untuk klasifikasi penentuan jenis kanker. *Modified K-Nearest Neighbor* (MKNN) dilakukan dengan menempatkan kelas data sesuai dengan nilai k pada *traning data* yang telah divalidasi dalam perhitungan *weighted* KNN [ADE-13].

Penelitian sebelumnya pernah diterapkan oleh Farisa Adelia pada data penentuan potensi tsunami akibat gempa bumi bawah laut, dimana didapatkan hasil bahwa MKNN dengan akurasi 73,74% lebih unggul dibandingkan dengan KNN yang hanya mencapai akurasi 72,07%. Berdasarkan latar belakang tersebut maka penelitian ini diberi judul **"Penentuan Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Modified K-Nearest Neighbor (MKNN)".**

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam skripsi ini adalah:

1. Bagaimana menerapkan *Modified K-Nearest Neighbor* (*MKNN*) *Classification* untuk menentukan jenis kanker berdasarkan struktur protein?
2. Bagaimana tingkat akurasi algoritma *Modified K-Nearest Neighbor* (*MKNN*) dengan sejumlah nilai data k pada data kanker?

1.3 Tujuan

Berdasarkan rumusan masalah tersebut, makan tujuan dalam penelitian ini adalah :

1. Untuk menerapkan algoritma *Modified K-Nearest Neighbor (MKNN) Classification* pada sequence p53.
2. Mendapatkan tingkat akurasi yang tepat untuk memprediksi jenis kanker.

1.4 Manfaat

Manfaat yang diambil dari penelitian ini adalah penulis dapat memperoleh pengetahuan dan pengalaman dalam mengembangkan algoritma *Modified K-Nearest Neihbor (MKNN)* dan memudahkan dalam menganalisa jenis kanker.

1.5 Batasan Masalah

Pada skripsi ini, batasan masalah penelitian ini dibatasi pada hal-hal berikut:

1. Pada skripsi ini penulis hanya menentukan jenis kanker berdasarkan susunan protein.

2. Jenis kanker yang diklasifikasikan hanya 3 jenis kanker yaitu *breast cancer, colorectal cancer,* dan *lung cancer.* Data yang digunakan untuk pengujian pada skripsi ini, didapatkan dari http://www.uniprot.org/ untuk protein yang bersifat normal dan http://p53.free.fr/ untuk data yang bersifat kanker maupun non kanker.
3. Algoritma klasifikasi hanya menggunakan *Modified K-Nearest Neighbor (MKNN).*
4. Data dari susunan protein yang digunakan mempunyai panjang yang sama.

1.6 Sistematika Pembahasan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB 1 PENDAHULUAN

   Bab ini berisi latar belakang penulisan, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika pembahasan skripsi.

2. BAB 2 KEPUSTAKAAN

   Bab ini berisi tentang teori penyakit kanker, jenis kanker, *data mining*, dan MKNN.

3. BAB 3 METODOLOGI

   Bab ini berisi tentang metode-metode yang digunakan untuk penentuan jenis kanker menggunakan algoritma *Modified K-Nearest Neighbor* (MKNN).

4. BAB 4 PERANCANGAN DAN IMPLEMENTASI

   Bab ini berisi penjelasan tentang implemetasi (OS, perangkat keras dan bahasa pemrograman yang digunakan), batasan-batasan implementasi kalau ada, file-file implementasi dari masing-masing modul atau klas (relasinya), serta algoritma operasi-operasi yang akan diimplementasikan untuk sistem diagnosis penyakit jantung koroner.

5. BAB 5 PEMBAHASAN

   Bab ini berisi penjelasan dari implementasi penentuan jenis kanker menggunakan algoritma *Modified K-Nearest Neighbor* (MKNN) pada sistem dan hasil pengujian yang dilakukan.

6. BAB 6 PENUTUP

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

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini membahas tentang penentuan jenis kanker berdasarkan susunan protein dengan menggunakan klasifikasi *Modified K-nearest Neighbor* (*MKNN*). Penelitian sebelumnya telah dilakukan oleh Farisa Adelia (2013) untuk klasifikasi penentuan potensi tsunami akibat gempa bumi dengan menggunakan klasifikasi MKNN. Dalam bab ini akan diuraikan beberapa kajian pustaka yang berhubungan dengan penelitian ini, diantaranya kanker, gen TP53, protein, mutasi, bioinformatika, data mining, *K-Nearest Neighbor* (KNN), *Modified K-Nearest Neighbor*.

2.2 Dasar Teori

2.2.1 Kanker

Kanker merupakan kumpulan sel yang abnormal tidak terkendali dan secara terus menerus tumbuh, sehingga dapat menjadi kanker yang mengakibatkan rusaknya sel atau jaringan sehat. Pertumbuhan sel yang tidak terkendali akan menyebabkan jaringan menjadi besar dan disebut tumor. Tumor merupakan istilah yang dipakai untuk semua bentuk pembengkakan atau benjolan dalam tubuh [NUR-08]. Kanker bisa terjadi dimana saja, dari berbagai jaringan, dalam berbagai organ. Sel-sel kanker membentuk suatu massa dari jaringan ganas yang menyusup ke jaringan di dekatnya dan bisa menyebar ke seluruh tubuh. Sel kanker mampu menyerang jaringan lain yang sehat, merusak jaringan tersebut dan tumbuh subur di atas jaringan lain.

Penyebab kanker sampai saat ini masih sulit untuk diketahui secara pasti karena merupakan gabungan dari sekumpulan faktor, baik internal maupun eksternal. Faktor internal disebabkan karena adanya perubahan genetik yang dapat menyebabkan mutasi gen. Dalam hubungannya dengan pertumbuhan tumor, terdapat dua golongan gen, yaitu onkogen dan gen penekan tumor. Onkogen adalah gen pemicu kanker, seperti gen *c-myc* dan gen ras. Gen penekan tumor adalah kelompok penekan terjadinya tumor yang lazim disebut *tumor suppressor gene,* seperti gen p53 dan gen Rb [PRA-05]. Gen p53 adalah gen yang paling banyak menjadi tema penelitian kanker pada manusia.

2.2.1.1 Gen TP53

Salah satu protein pengaktif transkripsi adalah protein penekan tumor yang dikode oleh TP53 yang dikenal sebagai p53 dan nama ini diambil dari berat molekulnya sebesar 53 kilodalton. TP53 mempunyai peranan yang sangat vital dalam melindungi sel atau jaringan dari proses transformasi agar pembelahan sel

tetap terkontrol. Bila tidak berfungsi dengan baik maka perkembangbiakan sel tidak dapat terkendali dan menimbulkan kanker. Mutasi p53 adalah perubahan genetik yang paling umum ditemukan pada kanker manusia [SYA-07].

2.2.1.2 Breast Cancer (Kanker Payudara)

Kanker payudara merupakan penyebab utama kematian oleh kanker pada wanita. Kanker payudara adalah tumor ganas yang menyerang jaringan payudara yang mencakup kelenjar susu dan saluran air susu. Kanker payudara terjadi karena sel telah kehilangan pengendalian dan mekanisme normalnya, sehingga menyebabkan kerusakan gen pengatur pertumbuhan yang mengakibatkan perkembangbiakan sel itu tidak terkendali.

Kanker payudara dapat terjadi dibagian mana saja dalam payudara, tetapi mayoritas terjadi pada kuadran atas terluar dimana sebagian besar jaringan payudara terdapat [MUL-13].

2.2.1.3 Colorectal Cancer (Kanker Usus)

Colorectal cancer atau disebut dengan kanker usus merupakan salah satu penyebab kematian di dunia menempati urutan ake 4 penyebab kematian karena kanker [ZEN-09]. Kanker usus didefinisikan sebagai keganasan yang terjadi pada usus besar, yang merupakan bagian dari system pencernaan. Kanker usus sering tanpa gejala hingga tahap lanjut, karena pola pertumbuhan yang lambat [HAR-04].

2.2.1.4 Lung Cancer (Kanker Paru-paru)

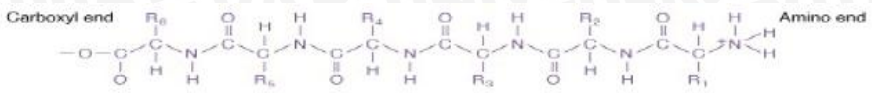
Kanker paru-paru adalah tumor ganas yang berasal dari saluran napas. Terjadi kanker ditandai dengan pertumbuhan sel yang tidak normal, tidak terbats dan merusak sel-sel jaringan yang normal. Penyebab yang pasti dari kanker paru-paru belum diketahui, karena terdapat beberapa faktor penyebab seperti kekebalan tubuh, genetik atau terkena paparan suatu zat yang bersifat karsinogenik [CHR-10].

2.2.2 Protein

Protein berasal dari bahasa Yunani yaitu *proteus* yang berarti “yang pertama” atau “yang terpenting”. Seorang ahli kimia dari Belanda bernama Mulder, mengisolasi susunan tubuh yang mengandung nitrogen dan menamakannya protein, terdiri dari satuan dasarnya asam amino. Molekul protein tersusun dari satuan-satuan dasar kimia yaitu asam amino. Satu molekul protein dapat terdiri dari 12 sampai 20 macam asam amino dan dapat mencapai jumlah ratusan asam amino [WID-11].

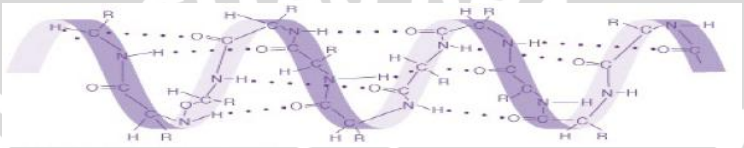
Struktur asam amino dapat dibagi menjadi beberapa bentuk yaitu struktur primer, sekunder, tersier dan kuartener.

1. **Struktur primer** suatu protein merupakan struktur yang sederhana dengan urutan-urutan asam amino yang tersusun secara linier yang mirip seperti tatanan huruf dalam sebuah kata dan tidak terjadi percabangan rantai [SAR-07].



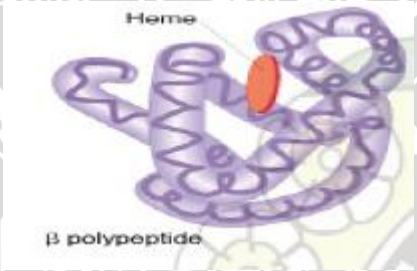
Gambar 2.1 Struktur Primer [SAR-07]

2. **Struktur sekunder** merupakan polipeptida terlipat-lipat, berbentuk tiga dimensi dengan cabang-cabang rantai polipeptidanya tersusun saling berdekatan [WID-11].



Gambar 2.2 Struktur Sekunder [SAR-07]

3. **Struktur tersier** menjelaskan bagaimana seluruh rantai polipeptida melipat sendiri sehingga membentuk stuktur 3 dimensi [RUS-10].



Gambar 2.3 Struktur Tersier [SAR-07]

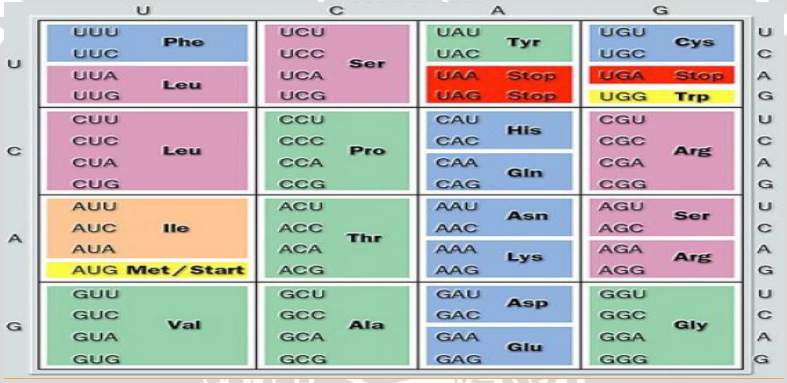
4. **Struktur kuartener** melibatkan beberapa polipeptida dalam membentuk suatu protein. Ikatan-ikatan yang terjadi sampai terbentuknya protein dengan ikatan-ikatan yang terjadi pada struktur tersier [WID-11].

Gambar 2.4 Struktur Kuartener [SAR-07]

2.2.2.1 Kode Genetik Terhadap Protein

Semenjak tahun 1960 semakin nyata bahwa ada paling sedikit tiga residu nukleotida DNA diperlukan untuk mengkode untuk masing-masing asam amino. Empat huruf kode DNA yaitu A, T, G dan C tersusun membentuk tiga huruf yang disebut dengan kodon [ANO-11]. Hubungan dari kode genetik terhadap protein sebagai pembawa sinyal, mengangkut molekul seperti oksigen, mengatur proses sel dan sebagai mekanisme pertahanan [RIA-13].

Pada prosesnya di dalam sel, terjadi proses transkripsi yaitu sintesis RNA dengan DNA sebagai cetakannya. RNA yang membawa sandi alias pesan yang sama dengan resep pada DNA ini kemudian bertindak sebagai cetakan untuk sintesis protein. Setiap kodon mengkodekan 1 asam amino. Sementara itu jumlah asam amino penyusun protein diketahui hanya 20 saja (dengan beberapa tambahan asam amino yang jarang). Dengan demikian berarti ada asam amino yang dikodekan oleh lebih dari satu kodon [RAH-12].



Gambar 2.5 Kode Genetik [ANO-11]

Dengan dipecahkannya misteri kode genetik, kini kita bisa mengetahui protein apa yang dihasilkan suatu gen tanpa harus menganalisa proteinnya secara langsung, urutan basa DNA bisa diterjemahkan menjadi urutan asam amino protein [RAH-12].

2.2.3 Mutasi

Mutasi adalah perubahan pada materi genetik suatu makhluk yang terjadi secara tiba-tiba. Mutasi gen yaitu perubahan terjadi pada material genetik, dalam prosesnya mutasi terjadi karena adanya perubahan urutan (*sequence*) nukleotida DNA kromosom, yang mengakibatkan terjadinya perubahan pada bentuk protein [SUD-09]

Mutasi dapat terjadi secara spontan dan terjadi melalui induksi. Mutasi spontan adalah mutasi yang akibatnya tidak diketahui secara jelas, baik dari lingkungan luar ataupun lingkungan internal dari organisme itu sendiri. Mutasi induksi adalah mutasi yang terjadi akibat paparan yang jelas, contoh : paparan sinar UV [WAR-11]. Individu yang memperlihatkan perubahan sifat akibat mutasi disebut dengan *mutan*. Dalam

kajian genetik mutan bisa dibandingkan dengan individu yang tidak mengalami perubahan sifat [ANO-09].

Mutasi pada urutan DNA gen dapat mengubah urutan asam amino dari protein yang dikode oleh gen. Perubahan satu basa (*point mutation*) dapat berupa transisi atau transversi [KUR-13].

Transisi merupakan perubahan basa purin/pirimidin menjadi basa purin/pirimidin lainnya, sedangkan transversi berupa perubahan basa purin menjadi basa pirimidin dan sebaliknya [PRA-08].

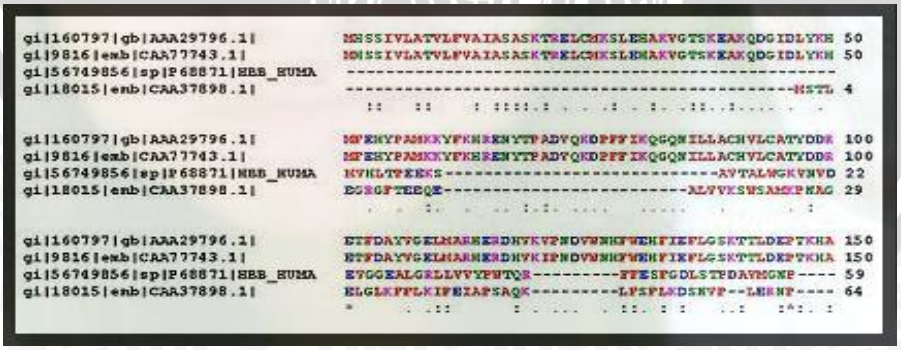
2.2.4 Bioinformatika

Bioinformatika merupakan kajian yang memadukan disiplin biologi molekul, matematika dan teknik informatika (TI). Bioinformatika bertujuan untuk menyelesaikan masalah-masalah biologi dengan menggunakan sekuen DNA dan asam amino dan informasi-informasi yang terkait dengannya [APR-04].

Kemajuan ilmu bioinformatika ini lebih didesak lagi oleh *genome project* yang menghasilkan tumpukan informasi gen dari berbagai makhluk hidup [UTA-03]. Semua data yang dihasilkan dari *genome project* disusun dan disimpan rapi sehingga bisa digunakan untuk berbagai keperluan, baik keperluan penelitian maupun keperluan di bidang medis. Dengan data yang memerlukan analisa bioinformatika dapat diketahui kuantitas dan kualitas transkripsi satu gen sehingga bisa menunjukkan gen-gen apa saja yang aktif terhadap perlakuan tertentu, misalnya timbulnya kanker [APR-04].

2.2.4.1 Protein Sequencing

Sekuensing protein adalah penentuan urutan asam amino pada suatu protein [TOM-10]. Hal ini berguna untuk menemukan gen sejenis pada beberapa organisme atau untuk memeriksa keabsahan hasil sekuensing atau untuk memeriksa fungsi gen hasil sekuensing [HAD-12].



Gambar 2.6 Deretan Sekuensing Protein [ENY-13]

2.2.4.2 Subtitution Matrix

Matriks subtitusi (*subtiturion matrix*) adalah suatu matriks kesamaan (*similarity matrix*) yang digunakan untuk menyatakan *residue substitution score*. Contoh dari matriks subtitusi ini salah satunya adalah PAM (*Point Accepted Mutation Matrix*).

2.2.4.3 PAM (*Point Accepted Mutation*)

PAM (*Point Accepted Mutation*) merupakan sekumpulan PAM1 – PAM250 yang berasal dari penurunan *sequence* yang memiliki hubungan kekerabatan yang dekat [KUR-13]. Pada tabel PAM titik yang termutasi pada protein adalah perubahan pada salah satu asam amino, yang terpilih secara alami. Terdapat dua proses yang berbeda ketika terjadi perubahan pada asam amino, yaitu:

1. Yang pertama terjadinya mutasi pada bagian gen yang memproduksi asam amino dari protein.
2. Yang kedua terjadinya mutasi oleh jenis baru yang lebih dominan. Agar dapat diterima oleh asam amino yang lain biasanya asam amino yang baru ini membuat dirinya mirip dan mempunyai fungsi dengan asam amino yang lama.

Jumlah dari matriks PAM (PAM1, PAM250) menunjukkan sebuah evolusi jarak. Semakin besar jumlahnya maka semakin besar pula jaraknya [DOR-07]. Untuk memperoleh nilai pada PAM maka dilakukan perkalian, contoh PAM2 diperoleh dari perkalian antara PAM1 dan PAM1, begitu pula dengan PAM3 diperoleh dari perkalian antara PAM1dan PAM2, begitu seterusnnya. Matrik PAM1 merupakan dasar untuk menghitung matrik yang lain dengan anggapan mutasi yang berulang akan mengikuti aturan yang sama dengan matrik PAM1, dengan logika tersebut dapat diperoleh matrik PAM250 [KUR-13]. Berikut ini adalah gambar tabel PAM :

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2	-2	0	0	-2	0	0	1	-1	0	-2	-1	-1	-3	1	1	1	-6	-4	0
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-5	-2
N	0	0	2	2	-3	1	2	1	2	-2	-3	1	-2	-3	0	1	1	-4	-2	-2
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-5	-1	0	0	-7	-4	-2
C	-2	-3	-4	-5	12	-5	-5	-4	-3	-3	-6	-5	-5	-4	-2	0	-2	-8	0	-2
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-4	0	-1	-1	-5	-4	-2
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	0	0	0	-7	-4	-2
G	1	-3	0	1	-3	-1	0	5	-2	-2	-4	-2	-3	-5	0	1	0	-7	-5	-1
H	-1	1	1	1	-3	3	0	-3	6	-3	-3	0	-3	-2	0	-1	-1	-3	0	-3
I	-1	-2	-2	-2	-2	-2	-2	-3	-3	4	2	-2	2	1	-2	-1	0	-5	-1	4
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-2	4	2	-2	-3	-2	-2	-1	2
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-4	-5	-2
M	-1	-1	-2	-3	-5	-1	-2	-3	-2	2	4	1	6	0	-2	-2	0	-4	-3	2
F	-3	-4	-3	-5	-4	-4	-5	-4	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1
P	1	0	0	-1	-3	0	0	0	0	-2	-2	-1	-2	-4	6	1	0	-6	-5	-1
S	2	1	2	1	1	0	1	2	0	-1	-2	1	-1	-2	2	2	2	-2	-2	0
T	0	-2	0	-1	-3	-2	-1	-1	-2	-1	-2	-1	-1	-4	0	1	2	-6	-4	0
W	-6	2	-5	-7	-7	-6	-7	-7	-5	-6	-7	-4	-6	1	-6	-2	-5	17	1	-8
Y	-3	-5	-2	-4	1	-4	-4	-5	0	-1	-1	-5	-2	7	-5	-3	-3	0	10	-2
V	0	-2	-2	-2	-2	-2	-2	-2	-2	4	2	-2	2	-1	-1	-1	0	-6	-3	4

Gambar 2.7 Tabel Matrik PAM250 [EDD-04]

Dari tabel di atas diketahui terdapat 20 macam asam amino yang digunakan untuk mensintesis protein di ribosom. Sebab kedua puluh asam amino ini saja yang memiliki sandi genetik [UYH-10]. Macam-macam residu asam amino yang diurutkan dari atas ke bawah yaitu Sistein Alanin (Ala) = A, Arginin (Arg) = R, Asparagin (Asn) = N, Asam Aspartat (Asp) = D, (Cys) = C, Glutamin (Gin) = Q, Asam Glutamat (Glu) = E, Glisin (Gly) = G, Histidin (His) = H, Isoleusin (Ile) = I, Leusin (Leu) = L, Lisin (Lys) = K, Metionin (Met) = M, Fenilalanin (Phe) = F, Prolin (Pro) = P, Serin (Ser) = S, Treonin (Tgr) = T, Triptofan (Trp) = W, Tirosin (Tyr) = Y, Valin (Val) = V [KUR-13].

2.2.5 Data Mining

Data mining adalah sebuah proses untuk menemukan pola atau pengetahuan yang bermanfaat secara otomatis dari sekumpulan data yang berjumlah banyak [SUN-10]. Secara sederhana, data mining digunakan untuk menggali informasi yang tersembunyi pada sebuah database yang memiliki data yang berjumlah besar. sehingga menemukan pola yang belum diketahui [MER-13].

Data mining sering dianggap sebagai bagian dari *Knowledge Discovery in Database* (KDD) yaitu sebuah proses mencari pengetahuan yang bermanfaat dari data proses KDD. Tahapan-tahapan dari proses *Knowledge Discovery in Database* (KDD) adalah sebagai berikut [SUN-10]:

1. Data Selection
   Seleksi data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi. Data yang telah diseleksi kemudian disimpan dalam bentuk berkas, terpisah dari basis data operasional.
2. Pre-processing / Cleaning
   Proses *cleaning* mencakup membuang duplikasi data, memeriksa data yang *inkonsisten*, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Pada proses ini dapat dilakukan proses *enrichment* data untuk memperkaya dan menambah data dengan informasi lain yang relevan dan diperlukan.
3. Transformation
   Proses transformasi data yang telah dipilih, sehingga telah sesuai dengan proses data mining. Proses ini merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.
4. Interpretation / Evaluation
   Interpretation mencakup apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya. Agar informasi yang dihasilkan dari proses data mining perlu di tampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.

Data mining sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistic dan database. Beberapa metode yang sering disebut-sebut dalam literatur data mining antara lain

clustering, classification, association rule mining, neural network, genetic algorithm dan lain-lain [WIR-11].

2.2.5.1 Klasifikasi

Klasifikasi merupakan pengelompokan obyek ke dalam satu atau beberapa kelompok berdasarkan kategori yang telah ditetapkan [OKT-13]. Klasifikasi merupakan suatu teknik yang dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan tersebut digunakan pada data baru untuk diklasifikasi [ADE-13].

Pendekatan umum yang digunakan dalam klasifikasi yang harus tersedia adalah *data training* berisi *record* yang mempunyai label *class*. Data training ini digunakan untuk membuat model klasifikasi dan menerapkannya pada *test* yang berisi *record-record* yang belum diketahui *class*-nya [OKT-13].

Terdapat beberapa proses klasifikasi yang sering digunakan dalam data mining, diantaranya *decision tree, Bayesian, fuzzy, neural network, support vector machine (SVM)* dan *k-nearest neighbor (KNN)* [ADE-13].

2.2.5.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) adalah metode klasifikasi yang memilih obyek latih yang memiliki sifat ketetanggaan (*neighborhood*) yang paling dekat [RID-13]. Algoritma KNN mengklasifikasikan data baru yang belum diketahui kelasnya dengan memilih data sejumlah k yang letak terdekat dari data baru tersebut. *Class* terbanyak dari data terdekat sejumlah k tersebut dipilih sebagai *class* yang diprediksikan untuk data yang baru [MER-13].

Algoritma KNN merupakan metode yang sederhana, mudah diimplementasikan, dapat menangani *data training* yang mengandung noise dan efektif jika data training besar. Selain itu, metode KNN juga memiliki beberapa kelemahan seperti berikut [SUL-12]:

a. Biaya komputasi cukup tinggi karena perlu untuk menghitung jarak setiap *data training.*
b. Membutuhkan memori yang besar.
c. Perlu untuk menentukan nilai K parameter,jumlah tetangga terdekat;
d. Menggunakan perhitungan jarak, yang belum diketahui pasti jenis jarak yang digunakan.
e. Belum diketahui atribut yang lebih baik untuk menghasilkan hasil terbaik.

2.2.5.3 Proses K-Nearest Neighbor (KNN)

Untuk mendefinisikan jarak antara dua titik yaitu titik pada data training (x) dan titik pada data testing (y) maka digunakan rumus Euclidean, seperti yang ditunjukkan pada persamaan [ZAI-14]:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\ , \qquad (2\text{-}1)$$

Dengan d adalah jarak antara titik pada data trainng x dan titik data testing y yang akan diklasifikasi, dimana x = x_1, x_2,…, x_i dan y = y_1, y_2,…, y_i dan 1 merepresentasikan nilai atribut, serta n merupakan dimensi atribut.

2.2.5.4 Modified K-Nearest Neighbor (MKNN)

Tujuan dari *Modified K-Nearest Neighbor* (MKNN) ini adalah menentukan kelas label dari *query instance* ke dalam k *data training* yang telah divalidasi. Setelah itu, *weighted* KNN akan dilakukan pada setiap data uji [ADE-13]. *Modified K-Nearest Neighbor (MKNN)* adalah menempatkan label kelas data sesuai dengan *k* divalidasi poin data yang sudah ditetapkan dengan perhitungan *K-Nearest Neighbor (KNN)* tertimbang. Dalam proses algoritma MKNN, terdapat beberapa tambahan proses dibanding dengan KNN, yaitu: validitas data *training* dan *weight voting.* [SUL-12]:

a. *Validitas* digunakan untuk menghitung jumlah titik dengan label yang sama untuk data tersebut. Persamaan yang digunakan untuk menghitung validitas dari setiap data adalah sebagai berikut [ADE-13]:

$$Validitas(x) = \frac{1}{\mathrm{k}}\sum_{i=1}^{k} \mathrm{S}\left(label(x), (label(N_i(x)))\right) \qquad (2\text{-}2)$$

Dimana:

k : jumlah titik terdekat

label (x) : kelas x

label $N_i(x)$: label kelas titik terdekat x

Fungsi S sendiri menyamakan kelas data *x* dengan kelas data terdekat ke-*i*. Fungsi S adalah sebagai berikut [HUD-13]:

$$S(a,b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases} \qquad (2\text{-}3)$$

Melalui persamaan (2-4) ditunjukkan bahwa a dan b adalah label kelas kategori pada data latih. S akan bernilai 1, jika label kategori a sama dengan label kategori b. S bernilai 0, jika label kategori a tidak sama dengan label kategori b [ADE-13].

b. *Weight Voting* KNN adalah salah satu variasi metode KNN yang menggunakan k tetangga terdekat.

$$W_{(i)} = \frac{1}{d + \alpha} \quad (2\text{-}4)$$

Dimana *d* adalah jarak dan *α* merupakan nilai *regulator smoothing*, dalam penelitian ini menggunakan *α* = 0,5 [ADE-13]. Kemudian validitas dari tiap data pada *data latih* dikalikan dengan *weight voting* berdasarkan pada jarak *Euclidean*. Dalam metode MKNN, perhitungan *weight voting* tiap tetangga seperti pada persamaan (2-6) [HUD-13]:

$$W_{(i)} = Validitas(i) x \frac{1}{d+0,5} \quad (2\text{-}5)$$

Dimana:

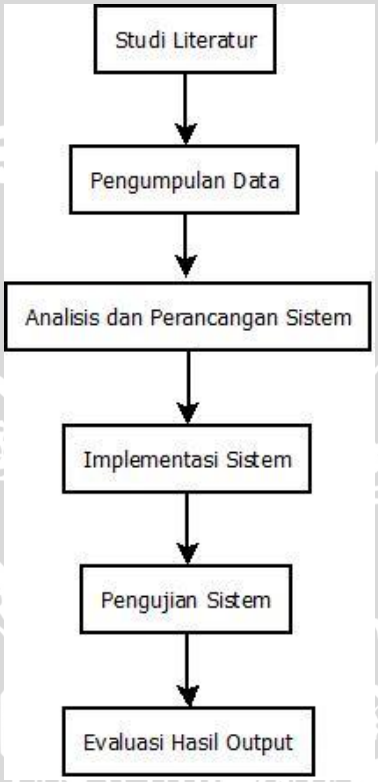
$W_{(i)}$: perhitungan weight voting
$Validitas_{(i)}$: nilai validitas
D : jarak Euclidean

c. *Akurasi Sistem* merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya. Akurasi dihitung dengan membagi jumlah data uji yang diklasifikasikan benar dengan jumlah total data uji dikalikan 100%, seperti persamaan berikut [HUD-13]:

$$Akurasi = \frac{Jumla\ h\ data\ uji\ benar}{Jumla\ h\ data} x\ 100\% \quad (2\text{-}6)$$

BAB 3 METODOLOGI

Pada bab ini akan dibahas metode yang digunakan, perancangan system, dan langkah-langkah yang dilakukan dalam penelitian. Adapun tahapan penelitian dapat digambarkan dalam bentuk diagram alir yang ditunjukkan gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Penjelasan pada diagram alur 3.1 adalah sebagai berikut:

1. Mempelajari literatur mengenai *modified k-nearest neighbor, breast cancer, colorectal cancer* dan *lung cancer*.
2. Pengumpulan data mengenai *breast cancer, colorectal cancer* dan *lung cancer* yang didasarkan pada perubahan genetika pada susunan protein.
3. Melakukan analisis data dan merancang sistem berdasarkan algoritma *modified k-nearest neighbor* untuk menentukan jenis kanker.
4. Pada tahap ini dibuat sebuah sistem atau bisa dikatakan pembangunan perangkat lunak yang berdasarkan analisis dan perancangan sistem.

5. Pengujian dilakukan untuk memastikan perangkat lunak yang dibangun sesuai dengan yang diharapkan dengan memasukkan data uji ke sistem pengujian.
6. Pada tahap evaluasi ini dilakukan pengukuran akurasi yang dihasilkan oleh sistem yang dibuat.

3.1 Studi Literatur

Studi literatur dibutuhkan untuk mendukung penelitian, memahami permasalahan yang diteliti, dan penyelesaian masalah. Mencari dasar teori mengenai *breast cancer, colorectal cancer, lung cancer*, struktur protein, algoritma *K-Nearest Neighbor* (KNN), dan algoritma *Modified K-Nearest Neighbor* (MKNN) yang dapat diperoleh dari berbagai sumber, seperti buku, jurnal, browsing internet, dan sumber lain yang dinilai dapat mendukung pada penelitian ini.

3.2 Pengumpulan Data

Pada tahapan pengumpulan data ini, menggunakan dataset yang didapat dari http://www.uniprot.org/, berupa sekuensing protein yang merupakan gabungan dari 20 residu asam amino berupa gabungan string sepanjang 393. Data protein yang digunakan adalah *human* TP53 *isoform 1*, dan dari database TP53 dengan alamat http://p53.free.fr/ untuk menentukan data yang bersifat kanker maupun non kanker.

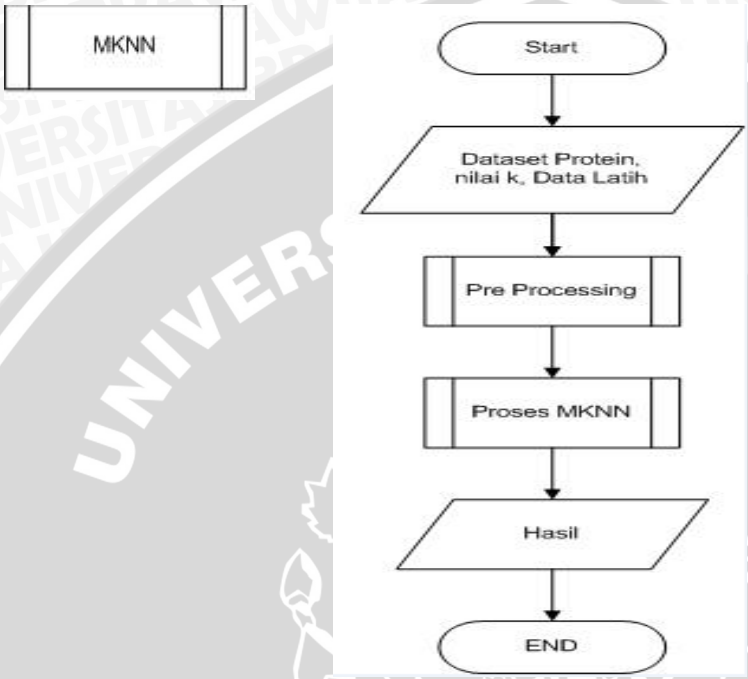
3.3 Deskripsi Sistem

Sistem yang dibuat merupakan perangkat lunak untuk mengolah data inputan berupa data latih dan data uji. Sistem akan melakukan klasifikasi menggunakan algoritma *modified k-nearest neighbor* terhadap perubahan genetikan pada susunan protein yang menghasilkan keluaran berupa *non cancer, breast cancer, colorectal cancer* dan *lung cancer*. Pada sistem ini juga menguji keakuratan hasil dari klasifikasi dataset *lung cancer* terhadap data yang sebenarnya. Selain itu parameter yang diujikan berkaitan dengan nilai k tetangga terdekat yang berpengaruh pada hasil akurasi.

3.4 Perancangan Sistem

Pada subbab ini akan dijelaskan tahapan-tahapan dalam membangun sebuah sistem. Pada tahap awal inputkan *dataset protein, nilai k* dan *data latih,* setelah itu dilakukan proses *pre processing* untuk merubah data yang berupa string menjadi numerik. Dalam *pre processing* akan membandingkan sebuah sekuen protein

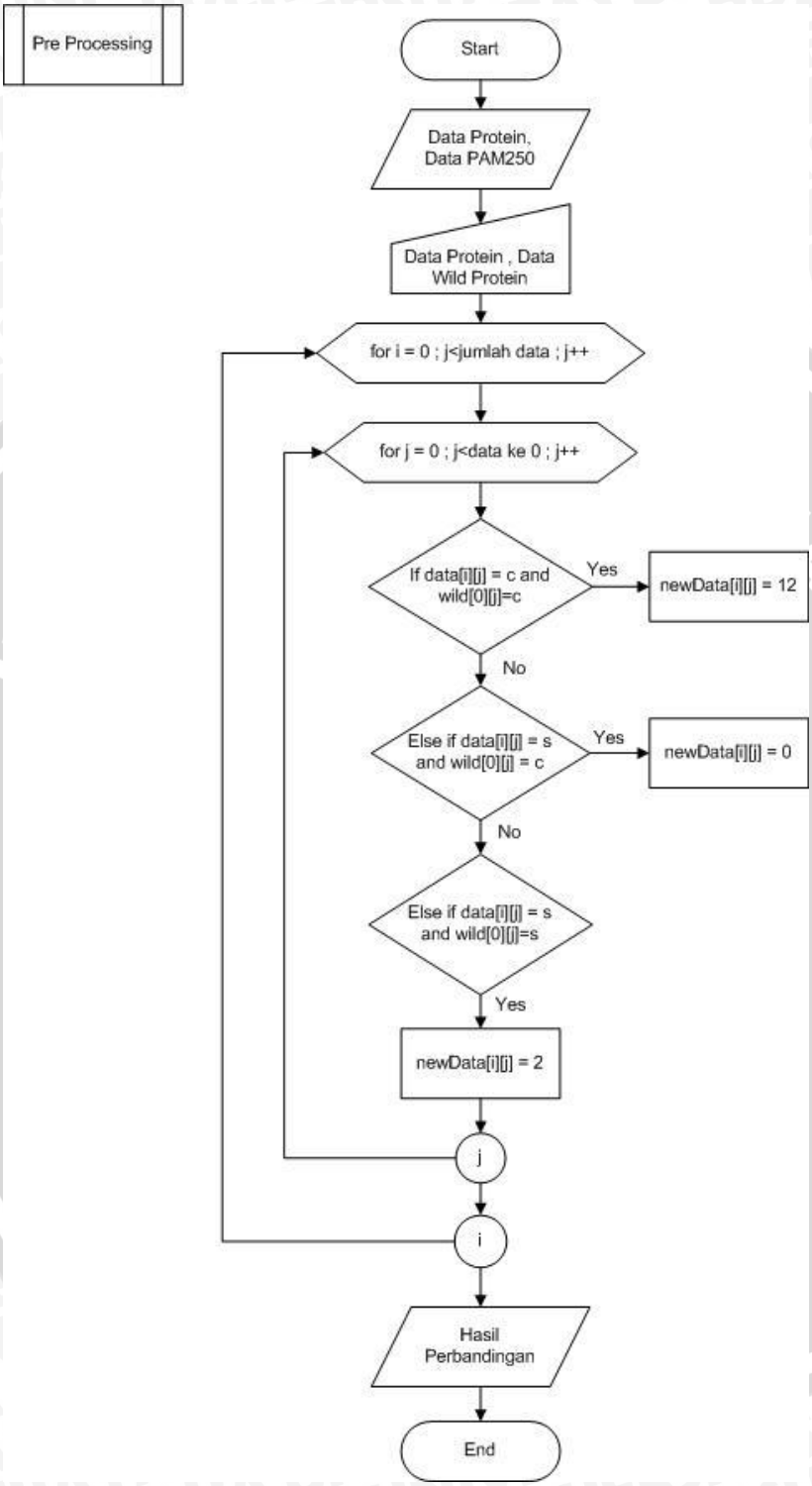
dengan sebuah data pembanding sepanjang sekuen tersebut. Kemudian sistem akan melakukan proses klasifikasi data kanker (*breast cancer, colorectal cancer* dan *lung cancer*) sesuai dengan data uji yang dimasukkan dengan menggunakan algoritma MKNN dan menghasilkan output hasil dari perhitungan akhir.



Gambar 3.2 Flowchart MKNN

3.4.1 Pre Processing

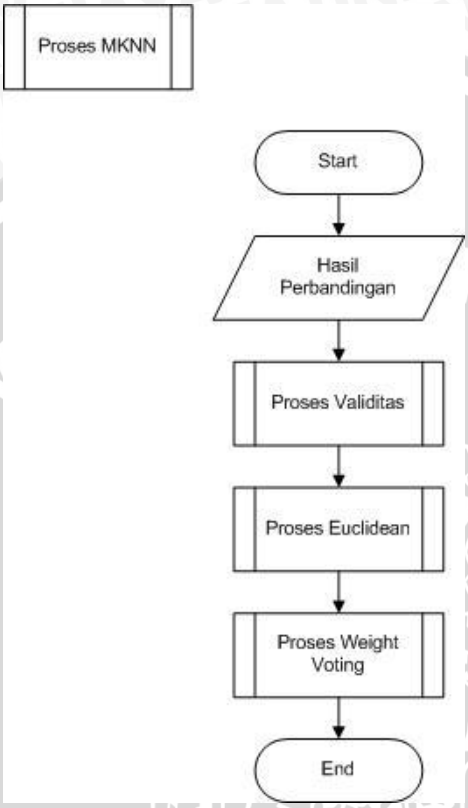
Pada tahap ini setiap data protein yang terdiri dari 20 residu asam amino berupa gabungan string sepanjang 393 akan dibandingkan dengan data pembanding kemudian diberi nilai sesuai dengan nilai pada tabel PAM250. Metode yang mengkonversikan asam amino kedalam nilai numerik disebut *scoring system.* Hasil dari konversi data ini akan mengisi tiap-tiap variabel dan disebut dengan nilai protein. Alur proses *pre processing* ditunjukkan pada gambar 3.3.



Gambar 3.3 Flowchart Pre Processing

3.4.2 Proses Modified K-Nearest Neighbor

Dalam proses MKNN terdapat 3 proses, yaitu proses validitas, proses euclidean, dan proses weight voting. Alur proses MKNN dijelaskan pada gambar 3.4.



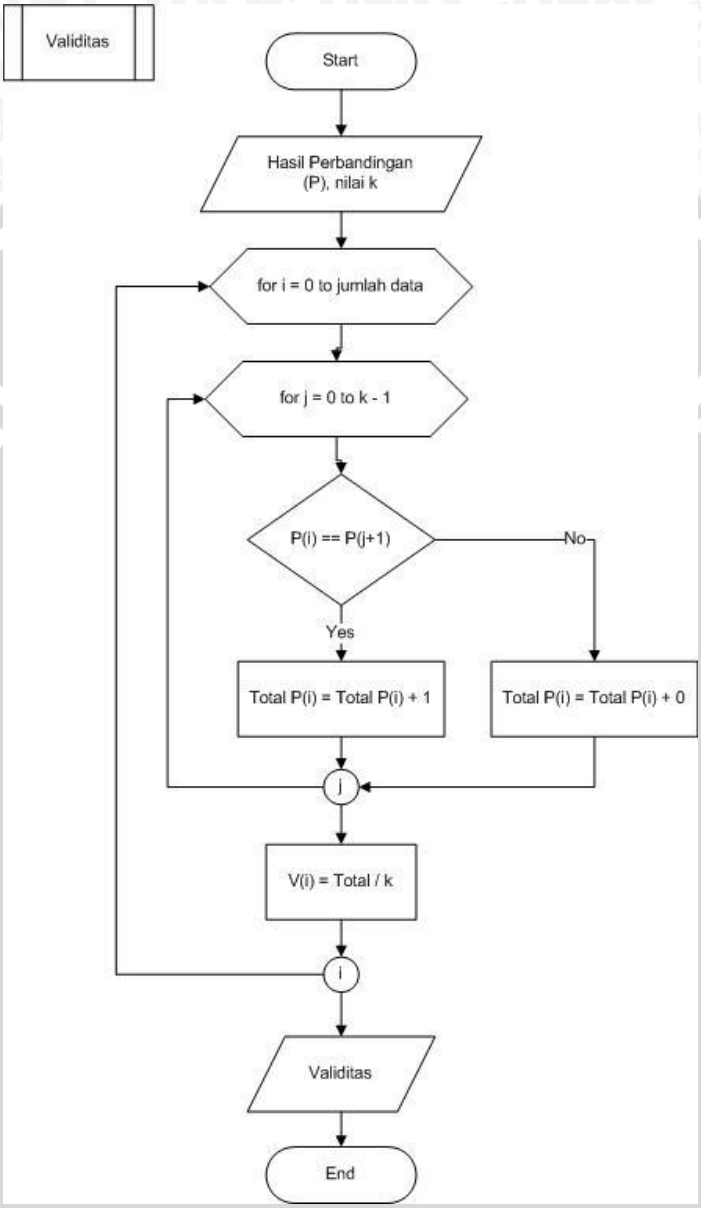
Gambar 3.4 Flowchart Proses MKNN

Pertama dilakukan inputan nilai protein, kemudian dilakukan proses validitas pada semua data training. Setelah itu dilakukan proses Euclidean untuk menghitung nilai jarak kedekatan tetangga data uji terhadap data latih. Selanjutnya dilakukan proses weight voting sesuai nilai k terbesar dan dihasilkan output hasil kanker yang di uji.

3.4.2.1 Proses Validitas

Perhitungan validitas dijelaskan alur tahapan yang terdiri dari beberapa tahapan yaitu menginputkan data latih dan input nilai k nya, kemudian dilakukan perhitungan pada persamaan (2-2). Setelah itu dilakukan perhitungan validitas dengan membandingkan kelas-kelas pada latihnya, dengan ketentuan jika kelasnya sama maka nilainya V[i] 1 dan jika kelasnya tidak sama maka nilainya V[i] 0. Data dibandingkan sebanyak k. Hasil dari pembandingan V[i] kemudian dijumlah dan

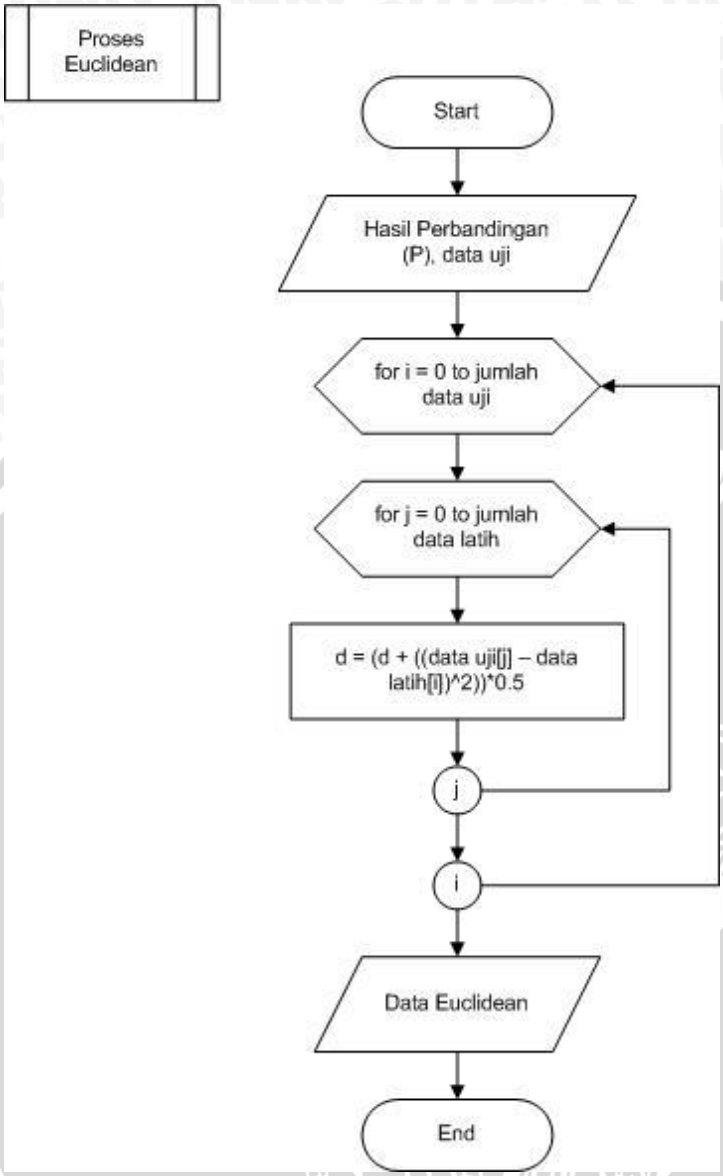
dibagi sebanyak k yang diinputkan. Pada proses terakhir akan didapatkan output berupa nilai validitas dari data latih.



Gambar 3.5 Flowchart Proses Validitas

3.4.2.2 Proses Euclidean

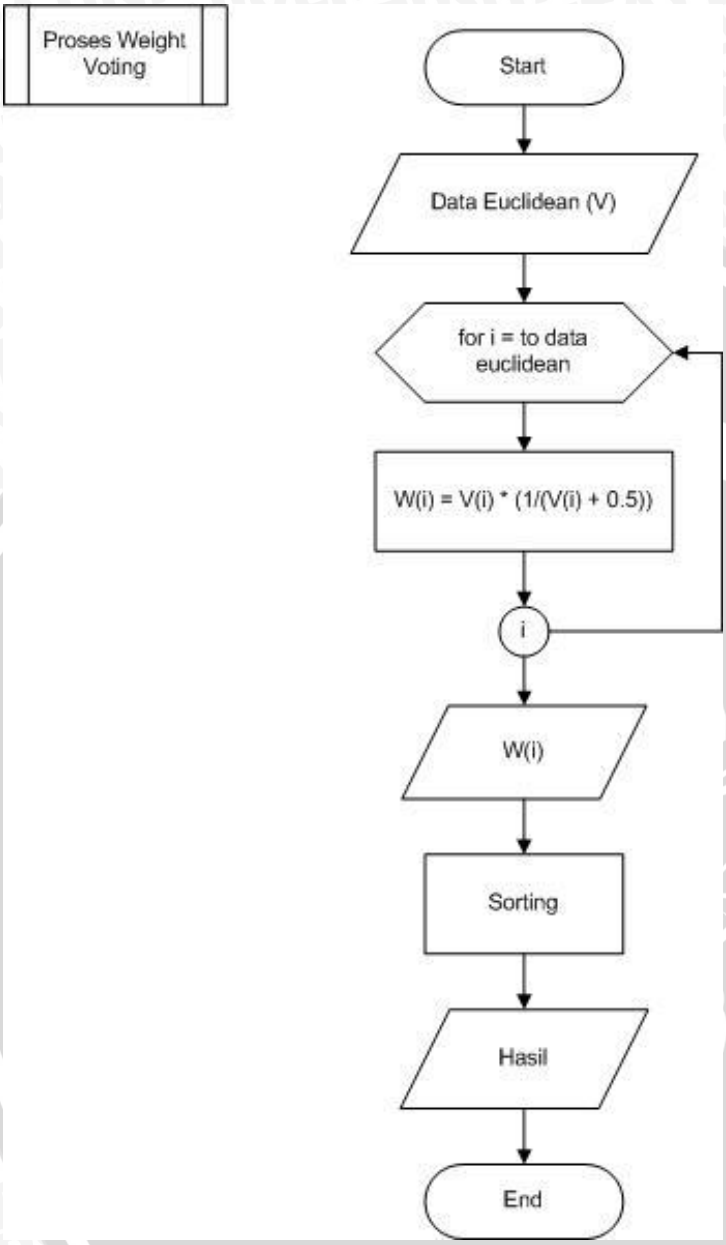
Pada tahap ini dilakukan proses perhitungan nilai jarak kedekatan tetangga (*Euclidean*) pada data uji terhadap data latih menggunakan persamaan (2-1). Alur proses euclidean dijelaskan pada gambar 3.6.



Gambar 3.6 Flowchart Proses Euclidean

3.4.2.3 Proses Weight Voting

Dalam proses weight voting yang pertama dilakukan adalah memasukkan nilai hasil validitas data latih dan nilai hasil euclidean. Kemudian dilakukan proses weight voting sesuai persamaan (2-5) dan menghasilkan output nilai weight voting.



Gambar 3.7 Flowchart Proses Weight Voting

3.5 Perhitungan Manual

Pada subbab ini digunakan sampel data yang diambil dari *dataset* protein. Data yang diambil berjumlah 15 data untuk data latih dan 1 data untuk data uji. Dari 15 data latih terdapat 2 kelas yaitu *non cancer* (NC) dan *lung cancer* (LC). Data yang telah diambil kemudian diklasifikasikan dengan menggunakan metode *Modified K-Nearest Neighbor*. Berikut ini pada tabel 3.1 menunjukkan data wild(normal), tabel 3.2 menunjukkan data latih dan tabel 3.3 menunjukkan data uji.

Tabel 3.1 Data Wild

Variabel	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Data Wild	L	R	V	E	Y	L	D	D	R	N

Tabel 3.2 Data Latih

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

Tabel 3.3 Data Uji

Variabel	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	Kelas
Data Uji	L	R	V	E	Y	L	D	D	T	N	?

Perhitungan pertama dimulai dengan melakukan *pre processing* data yaitu melakukan transformasi data pada setiap variabelnya yang berupa string protein menjadi numerik dengan membandingkan data yang termutasi dengan data string protein *wild* (normal) **LRVEYLDDRN** sehingga didapatkan hasil pada tabel 3.3 untuk data latih dan 3.4 untuk data uji. Untuk kelas akan dinotasikan dengan angka

non cancer = 0, *breast cancer* = 1, *colorectal cancer* = 2, *lung cancer* = 3, dan angka yang menunjukkan mutasi diberi warna hijau.

Tabel 3.4 Data Latih Yang Sudah Ditransform

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

Tabel 3.5 Data Uji Yang Sudah Ditransformasi

Data Uji	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	Kelas
Data Uji 1	6	6	4	4	10	6	4	4	-1	2	?

Perhitungan kedua dilakukan proses menentukan nilai k tetangga terdekat dengan nilai k = 3. Kemudian dilanjutkan dengan menghitung nilai *validitas* data latih dengan persamaan (2-2).

$$Validitas(x) = \frac{1}{\mathrm{k}} \sum_{i=1}^{k} \mathrm{S}\left(label(x_i), (label(N_i(x)))\right)$$

$$= \frac{1}{3} \sum_{i=1}^{k} \mathrm{S}\left(label(x = 1), (label(N_i(x = 3)))\right)$$

$$= \frac{1}{3}(0 + 0 + 0)$$

$$= 0$$

Perhitungan yang sama dilakukan untuk semua data latih. Hasil perhitungan nilai *validitas* ditunjukkan dalam tabel 3.5.

Tabel 3.6 Validitas

Data Latih	k = 1	k = 2	k = 3	SUM S(a,b)	Validitas
1	0	0	0	0	0
2	1	1	0	2	0.666667
3	1	0	0	1	0.333333
4	0	0	0	0	0
5	0	0	0	0	0
6	1	1	0	2	0.666667
7	1	0	0	1	0.333333
8	0	0	0	0	0
9	1	0	0	1	0.333333
10	0	0	0	0	0
11	1	0	1	2	0.666667
12	0	1	1	2	0.666667
13	0	0	0	0	0
14	1	0	1	2	0.666667
15	1	0	1	2	0.666667

Setelah mendapatkan nilai validitas, kemudian dilanjutkan melakukan perhitungan jarak terdekat dari data uji dengan data latih, dengan rumus Euclidean seperti pada persamaan (2-1).

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

$$= \sqrt{\begin{matrix}(6-6)^2 + (6-6)^2 + (4-4)^2 + (4-4)^2 + (10-10)^2 + \\ (6-6)^2 + (4-4)^2 + (4-4)^2 + (-1-6)^2 + (2-2)^2\end{matrix}}$$

$$= \sqrt{49}$$

$$= 7$$

Kemudian dilakukan perhitungan antara record data uji dengan data latih lainnya. Hasil perhitungan dari jarak *Euclidean* ditunjukkan pada tabel 3.6.

Tabel 3.7 Hasil Perhitungan Euclidean

Data Latih	SUM Euclidean	Euclidean
D1	49	7
D2	65	8.062258
D3	53	7.28011
D4	130	11.40175
D5	149	12.20656
D6	65	8.062258
D7	149	12.20656
D8	65	8.062258
D9	53	7.28011
D10	193	13.89244
D11	58	7.615773
D12	58	7.615773
D13	85	9.219544
D14	16	4
D15	65	8.062258

Selanjutnya melakukan perhitungan *weight voting* dengan memasukkan nilai validitas dan nilai Euclidean sesuai dengan persamaan (2-5). Sebagai contoh perhitungan dapat dilihat sebagai berikut.

$$W_{(i)} = Validitas(i) x \frac{1}{d + 0{,}5}$$

$$= 0 \; x \frac{1}{7 + 0{,}5}$$

$$= 0$$

Perhitungan *weight voting* yang sama dilakukan pada semua data latih, kemudian dilakukan pengurutan dari yang terbesar. Hasil dari *weight voting* ditunjukkan pada tabel 3.7.

Tabel 3.8 Hasil Perhitungan Weight Voting

Data Latih	Weight Voting	Kelas
D14	0.148148148	1
D11	0.082144567	1
D12	0.082144567	1
D2	0.077861084	3
D6	0.077861084	0
D15	0.077861084	1
D3	0.042844296	3
D9	0.042844296	2
D7	0.026233178	0
D1	0	1
D4	0	3
D5	0	1
D8	0	0
D10	0	2
D13	0	3

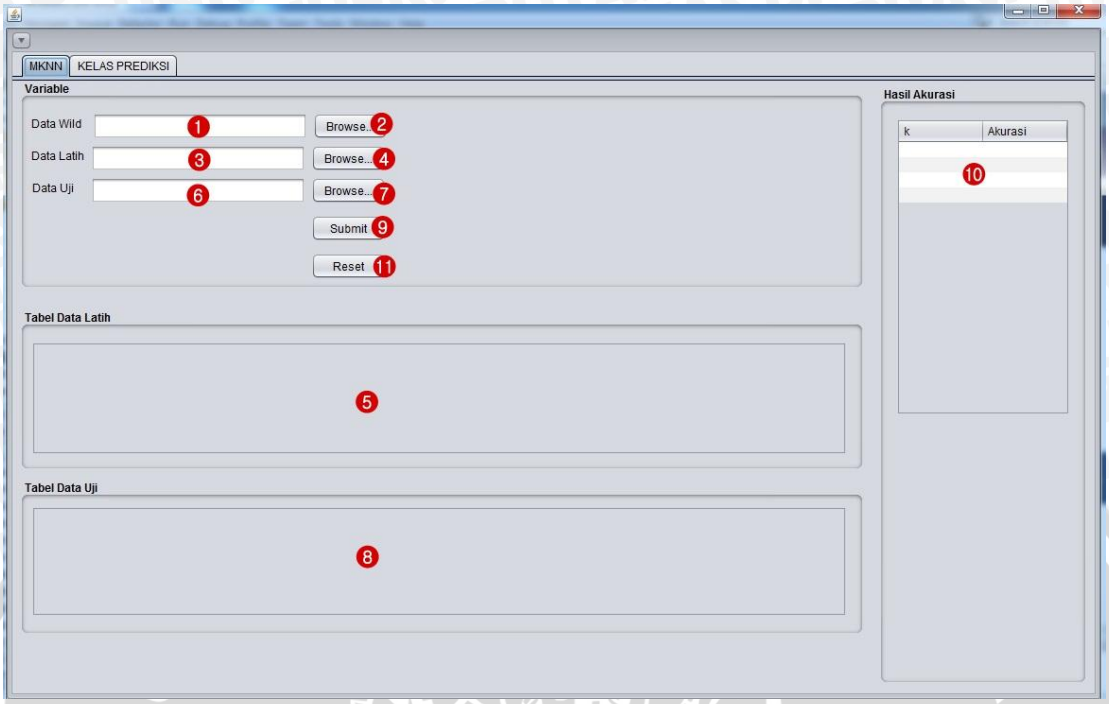
Setelah dihitung menggunakan *weight voting* maka dicari nilai terbesar sesuai dengan nilai k yang telah ditentukan, diawal telah ditentukan nilak k = 3. Diketahui nilai terbesar terdapat pada data ke 14 dari data latih yaitu sebesar 0.148148148 dengan kelas *breast cancer*, data ke 11 dari data latih yaitu sebesar 0.082144567 dengan kelas *breast cancer*, dan data ke 12 dari data latih yaitu sebesar 0.082144567 dengan kelas *breast cancer*. Dari data uji pertama maka kelas *cancer* yang ditentukan adalah *breast cancer.*

3.6 Perancangan Antarmuka

Pada subbab ini akan dijelaskan mengenai perancangan antarmuka *(interface)* sistem yaitu terdiri dari perhitungan akurasi nilai k dan kelas prediksi kanker.

3.6.1 Interface Perhitungan Akurasi Nilai k

Pada tahap ini bertujuan untuk menghitung data mencari akurasi pada setiap nilai k menggunakan metode Modified K-Nearest Neighbor (MKNN). Rancangan antarmuka untuk proses perhitungan akurasi nilai k ditunjukkan pada gambar 3.8.



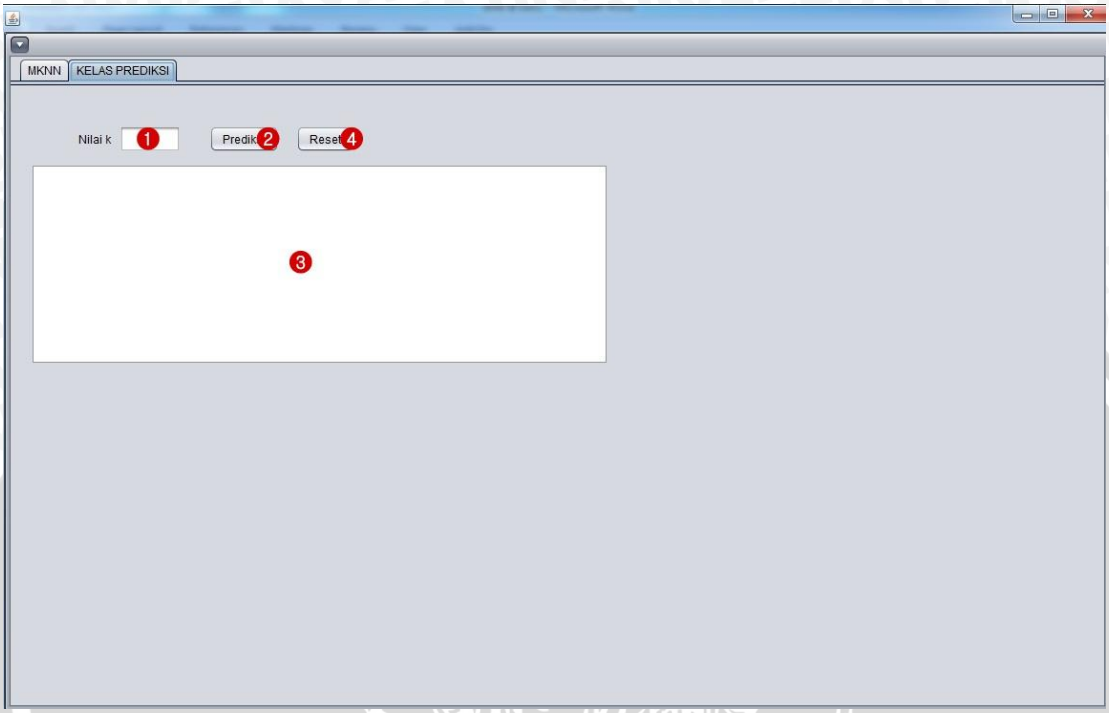
Gambar 3.8 Interface Perhitungan Nilai k

Pada gambar 3.8 interface sistem terdiri dari:

1. *TextField* digunakan untuk menampilkan *file* data *wild* yang dipilih.
2. Tombol *browse* digunakan untuk mencari data *wild* yang akan diinputkan ke dalam sistem.
3. *TextField* digunakan untuk menampilkan *file* data latih yang dipilih.
4. Tombol *browse* digunakan untuk mencari data latih yang akan diinputkan ke dalam sistem.
5. *Table* untuk menampilkan hasil transform dari sekuensing protein data latih ke numerik.
6. *TextField* digunakan untuk menampilkan *file* data uji yang dipilih.
7. Tombol *browse* digunakan untuk mencari data uji yang akan diinputkan ke dalam sistem.
8. *Table* untuk menampilkan hasil transform dari sekuensing protein data uji ke numerik.
9. Tombol *submit* digunakan untuk melakukan proses perhitungan MKNN.
10. Kolom *table* digunakan untuk menampilkan hasil akurasi setiap nilai k
11. Tombol *reset* untuk mengapus proses yang telah dilakukan sebelumnya.

3.6.2 Interface Kelas Prediksi Kanker

Bagian ini bertujuan untuk menampilkan kelas prediksi kanker dari setiap nilai k. Rancangan antarmuka untuk kelas prediksi kanker ditunjukkan pada gambar 3.9



Gambar 3.9 Interface MKNN Classification

Pada gambar 3.8 interface sistem terdiri dari:

1. *Textbox* digunakan untuk menginputkan nilai k yang akan ditampilkan kelas prediksi kankernya.
2. Tombol *prediksi* untuk memulai proses penampilan kelas prediksi kanker sesuai nilai k yang diinputkan.
3. *Table* untuk menampilkan kelas prediksi kanker.
4. Tombol *reset* untuk menghapus proses yang telah dilakukan sebelumnya.

3.6.3 Perancangan Pengujian Jumlah Dataset

Pengujian jumlah dataset terhadap untuk menentukan nilai akurasi dengan menggunakan persentase *data uji* yang berbeda, yaitu 20%, 30% dan 40%. Dalam proses pengujian ini menggunakan dataset 100, 150 dan 200. Pengujian jumlah dataset ditunjukkan pada tabel 3.8.

Tabel 3.9 Pengujian Jumlah Dataset

Jumlah Dataset	Nilai k	Persentase Data Uji			Akurasi Rata-rata(%)
		20%	30%	40%	
	1				
	2				
	-				
	-				
	20				
Akurasi Rata-rata(%)					

BAB IV PERANCANGAN DAN IMPLEMENTASI

4.1 Lingkungan Implementasi

Beberapa aspek yang perlu diperhatikan dalam lingkungan implementasi yaitu lingkungan implementasi perangkat keras dan perangkat lunak. Implementasi ini bertujuan untuk memenuhi kebutuhan sistem yang akan dikembangkan dan metode yang diimplementasikan.

4.1.1 Lingkungan Implementasi Perangkat Keras

Komponen perangkat keras yang digunakan untuk penelitian penentuan jenis kanker menggunakan algoritma *Modified K-nearest neighbor* ini adalah sebagai berikut:

1. Prosesor Intel® Core™ i5-4200M CPU @ 2.50GHz
2. Memori 2 GB
3. Harddisk 500 GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Untuk melakukan penelitian ini dibutuhkan beberapa perangkat lunak. Perangkat lunak yang digunakan adalah sebagai berikut :

1. Sistem operasi yang digunakan Windows 7
2. Aplikasi dibangun menggunakan GUI dan code menggunakan NetBeans IDE 8.0.2
3. Bahasa pemrograman yang digunakan Java dengan komponen java yang digunakan Java Development Kit (JDK) 1.8.0
4. Notepad (jenis data .xml)

4.2 Implementasi Program

Pada subbab ini akan dijelaskan mengenai implementasi perangkat lunak, dimulai dari proses pembacaan file XML, preprocessing data, perhitungan validitas, perhitungan jarak *Euclidean*, dan perhitungan *weight voting.*

4.2.1 Class ReadXML

Kelas ReadXML merupakan kelas untuk membaca file xml yang disimpan menggunakan notepad. Di dalam kelas ini terdapat variabel path yang bertipe string dan variabel `data`, `data2`, `dataKelas`, dan `dataKelas2` dengan tipe string dalam bentuk array. Fungsi dari variabel `data2` adalah untuk mengambil data tag "`isi`" pada file XML dan hasilnya ditampung pada variabel `data`. Untuk variabel dataKelas2 fungsinya untuk mengambil data tag "`protein`" pada file XML dan

hasilnya ditampung pada variabel `dataKelas`. Implementasi ReadXML dapat dilihat pada source code 4.1.

```
public class ReadXml
{
  private String path;
  private String[][] data,data2;
  private int[] dataKelas,dataKelas2;

 //constructor untuk menampung dimana tempat file xml disimpan
  public ReadXml(String path)
  {
      this.path = path;
  }
   public void read() throws ParserConfigurationException, SAXException,
   IOException
  {
   DocumentBuilderFactory builderFactory =
   DocumentBuilderFactory.newInstance();
   DocumentBuilder builder = builderFactory.newDocumentBuilder();
   File file = new File(path);
  //conversi file xml agar terbaca di java
   Document document = builder.parse(file);
  //Mengambil tag hasil
   Element hasil = (Element)
   document.getElementsByTagName("hasil").item(0);
  //Mengambil tag protein
   NodeList list = hasil.getElementsByTagName("protein");
   Element protein1 = (Element) list.item(0);
   Node a = protein1.getElementsByTagName("isi").item(0);
   data2=new String [list.getLength()][a.getTextContent().length()];
   dataKelas2 = new int[list.getLength()];
   for(int i=0; i<list.getLength(); i++)
   {
      Element protein = (Element) list.item(i);
      Node isi = protein.getElementsByTagName("isi").item(0);
```

```
      data2[i] = isi.getTextContent().split("");
      Node kelas = protein.getElementsByTagName("kelas").item(0);
      dataKelas2[i]=Integer.valueOf(kelas.getTextContent());
    }
   data= new String[data2.length][data2[0].length-1];
   dataKelas= new int[dataKelas2.length];
   for(int i=0; i<data.length; i++)
   {
      for(int j=0; j<data[0].length; j++)
      {
         data[i][j] = data2[i][j+1];
      }
      dataKelas[i]=dataKelas2[i];
   }
}
public String[][] getData()
{
  return data;
}
public void setData(String[][] data)
{
  this.data = data;
}
public int[] getDataKelas()
{
  return dataKelas;
}
public void setDataKelas(int[] dataKelas)
{
  this.dataKelas = dataKelas;
}
```

Source Code 4.1 Pembacaan File XML

4.2.2 Class Preprocessing Data

Kelas preprocessing data merupakan transformasi data string menjadi numerik, sehingga jarak antar data dapat dihitung. Di dalam kelas ini terdapat variabel `data,wild,` dan `newData`. Keyword `this.data` berfungsi untuk menampung data file XML dari tag "`isi`". Keyword `this.wild` berfungsi untuk menampung data `wild type` (protein yang bersifat normal). Instansiasi variabel `newData` digunakan untuk menampung hasil transformasi data string menjadi numerik. Implementasi preprocessing data dapat dilihat pada source code 4.2.

```
public class PreProsesingData
{
   private String[][] data,wild,newData;
   public PreProsesingData(String [][] data, String [][] wild)
  {
     this.data=data;
     this.wild=wild;
     newData = new String[data.length][data[0].length];
   }
//konversi tabel PAM250
public void proses()
{
   for (int i=0; i<data.length; i++)
   {
     for(int j=0; j<data[0].length; j++)
     {
         if (data[i][j].equals("C") && wild[0][j].equals("C"))
         {
            newData[i][j]="12";
         }
         else if (data[i][j].equals("S") && wild[0][j].equals("C"))
         {
           newData[i][j]="0";
         }
           else if (data[i][j].equals("S") && wild[0][j].equals("S"))
         {
           newData[i][j]="2";
```

```
}
  else if (data[i][j].equals("T") && wild[0][j].equals("C"))
{
  newData[i][j]="-2";
}
  else if (data[i][j].equals("T") && wild[0][j].equals("S"))
{
  newData[i][j]="1";
}
  else if (data[i][j].equals("T") && wild[0][j].equals("T"))
{
  newData[i][j]="3";
}
  else if (data[i][j].equals("P") && wild[0][j].equals("C"))
{
  newData[i][j]="-3";
}
  else if (data[i][j].equals("P") && wild[0][j].equals("S"))
{
  newData[i][j]="-1";
}
  else if (data[i][j].equals("P") && wild[0][j].equals("T"))
{
  newData[i][j]="0";
}
  else if (data[i][j].equals("P") && wild[0][j].equals("P"))
{
  newData[i][j]="6";
}
  else if (data[i][j].equals("A") && wild[0][j].equals("C"))
{
  newData[i][j]="-2";
}
  else if (data[i][j].equals("A") && wild[0][j].equals("S"))
{
```

```
        newData[i][j]="1";
      }
        else if (data[i][j].equals("A") && wild[0][j].equals("T"))
      {
        newData[i][j]="1";
      }
        else if (data[i][j].equals("A") && wild[0][j].equals("P"))
      {
        newData[i][j]="1";
      }
        else if (data[i][j].equals("A") && wild[0][j].equals("A"))
      {
        newData[i][j]="2";
      }
        else if (data[i][j].equals("G") && wild[0][j].equals("C"))
      {
        newData[i][j]="3";
      }
        else if (data[i][j].equals("G") && wild[0][j].equals("S"))
      {
        newData[i][j]="1";
      }
        else if (data[i][j].equals("G") && wild[0][j].equals("T"))
      {
        newData[i][j]="0";
      }
        else if (data[i][j].equals("G") && wild[0][j].equals("P"))
      {
        newData[i][j]="-1";
      }
        else if (data[i][j].equals("G") && wild[0][j].equals("A"))
      {
        newData[i][j]="1";
      }
        else if (data[i][j].equals("G") && wild[0][j].equals("G"))
```

```
            {
              newData[i][j]="5";
            }
            else
            {
              newData[i][j]="0";
            }
        }
    }
 }
 public String[][] getNewData()
 {
    return newData;
 }
}
```

Source Code 4.2 Proses Preprocessing Data

4.2.3 Proses Modified K-Nearest Neighbor

Tahapan proses pada klasifikasi MKNN diantaranya perhitungan *validitas, Euclidean,* dan *weight voting* yang akan digunakan untuk menentukan kelas pada data testing.

4.2.3.1 Proses Validitas

Fungsi dari validitas adalah untuk menghitung jumlah titik dengan label yang sama pada data tersebut. Pada proses validitas dilakukan perbandingan kelas pada data latih, jika kelas dengan kategori sama maka akan bernilai 1, dan jika kelas dengan kategori berbeda maka akan bernilai 0. Kemudian nilai validitas didapatkan dengan menjumlahkan hasil perbandingan dan dibagi sebanyak nilai k yang ditentukan sebelumnya. Tahapan proses validitas ditunjukkan pada source code 4.3.

```
public class Validitas
{
    private int[] kelas;
    private double[] valid;
    private int[][] ke;
```

```
    private int k;

    public Validitas(int[] kelas, int k)
    {
       this.kelas = kelas;
       this.k = k;
       ke = new int[k][kelas.length];
    }
    public double[] valid()
    {
       int g=1;
       valid = new double[kelas.length];
       for(int j=0; j<kelas.length; j++)
       {
          valid[j]=0;
          for(int i=0; i<k ; i++)
          {
             if(i+g<kelas.length)
             {
                 ke[i][j]=i+g;
             }
             else
             {
                 ke[i][j]=(i+g)-kelas.length;
             }
          }
          g++;
       }
       for(int w=0; w<valid.length; w++)
       {
           for(int q=0; q<k; q++)
           {
               if(kelas[ke[q][w]]!=kelas[w])
               {
                  valid[w]=valid[w]+0;
```

```
                }
                else
                {
                   valid[w]=valid[w]+1;
                }
            }
            valid[w]=valid[w]/3;
            System.out.println(valid[w]);
        }
       return valid;
    }
}
```

Source Code 4.3 Proses Perhitungan Validitas

4.2.3.2 Proses Euclidean

Pada proses perhitungan *Euclidean* dilakukan inisialisasi awal pada array 2 dimensi dari nilai jarak 0. Kemudian dilakukan perulangan sebanyak data protein yang ada. Fungsi *Euclidean* untuk memperpendek jarak dari data uji dengan data latih. Implementasi untuk menghitung *Euclidean* ditunjukkan pada source code 4.4.

```
public void ecludian()
{
    dataEcludian = new double[dataUji.length][dataLatih.length];
    for (int i = 0; i < dataUji.length; i++)
    {
        for (int j = 0; j < dataLatih.length; j++)
        {
            dataEcludian[i][j] = 0;
        }
    }
    for (int k = 0; k < dataUji.length; k++)
    {
        for (int i = 0; i < dataLatih.length; i++)
        {
            for (int j = 0; j < dataLatih[0].length; j++)
```

```
            {
                dataEcludian[k][i]      =      (dataEcludian[k][i]      +
                (Math.pow((dataUji[k][j] - dataLatih[i][j]), 2)));
            }
            dataEcludian[k][i] = Math.sqrt(dataEcludian[k][i]);
            System.out.println(dataEcludian[k][i]);
        }
    }
}
```

Source Code 4.4 Proses Perhitungan Euclidean

4.2.3.3 Proses Weight Voting

Pada proses *weight voting* dilakukan perhitungan dengan mengalikan nilai validitas dan dibandingkan dengan nilai *Euclidean,* kemudian dilakukan sorting untuk menentukan kelas pada data uji dengan mengambil nilai terbesar. Teknik *weight voting* ini berpengaruh terhadap data yang mempunyai nilai validitas lebih tinggi an paling dekat dengan data. Proses *weight voting* ditujukkan pada source code 4.5.

```
public void weightVoting()
{
    dataWeightVoting = new
    double[dataEcludian.length][dataEcludian[0].length];
    dataWeightVotingTemp = new
    double[dataEcludian.length][dataEcludian[0].length];
    double[dataEcludian.length][dataEcludian[0].length];
    kelasSorting = new
    int[dataWeightVoting.length][dataWeightVoting[0].length];
    kelasSortingTemp = new
    int[dataWeightVoting.length][dataWeightVoting[0].length];

    for (int i = 0; i < dataWeightVoting.length; i++)
    {
        for (int j = 0; j < dataWeightVoting[0].length; j++)
        {
            dataWeightVoting[i][j] = (validitas[j] * (1 /
            (dataEcludian[i][j] + 0.5)));
```

```
            kelasSorting[i][j] = kelas[j];
            System.out.println(dataWeightVoting[i][j]);

        }
    }
}
```

Source Code 4.5 Proses Weight Voting

4.2.3.4 Proses Sorting Nilai

Pada proses ini dilakukan untuk mengurutkan/mensorting nilai dari *weight voting* mulai dari yang terbesar ke yang terkecil. Implementasi mengurutkan nilai dari yang terbesar ke yang terkecil dapat dilihat pada source code 4.6.

```
public void weightSorting2()
{
   for (int i = 0; i < dataWeightVoting.length; i++)
   {
      for (int c = 0; c < (dataWeightVoting[0].length - 1); c++)
      {
         for (int d = 0; d < dataWeightVoting[0].length - c - 1; d++)
         {
            if (dataWeightVoting[i][d] > dataWeightVoting[i][d + 1]) /* For
            descending order use < */
            {
               swap = dataWeightVoting[i][d];
               dataWeightVoting[i][d] = dataWeightVoting[i][d + 1];
               dataWeightVoting[i][d + 1] = swap;
               swapk = kelasSorting[i][d];
               kelasSorting[i][d] = kelasSorting[i][d + 1];
               kelasSorting[i][d + 1] = swapk;
            }
         }
      }
   }
   for (int i = 0; i < dataWeightVoting.length; i++)
```

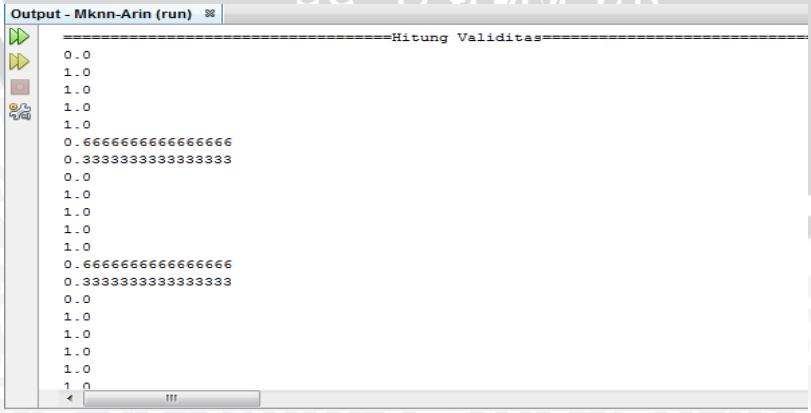
```
    {
        for (int j = 0; j < dataWeightVoting[0].length; j++)
        {

            dataWeightVotingTemp[i][(dataWeightVotingTemp[0].length - 1) - j] =
            dataWeightVoting[i][j];
            kelasSortingTemp[i][(kelasSortingTemp[0].length - 1) - j] =
            kelasSorting[i][j];
        }
    }
    System.out.println("hasil sorting");
    for (int i = 0; i < dataWeightVoting.length; i++)
    {
        for (int j = 0; j < dataWeightVoting[0].length; j++)
        {
            System.out.println(kelasSortingTemp[i][j]);
        }
    }
}
```

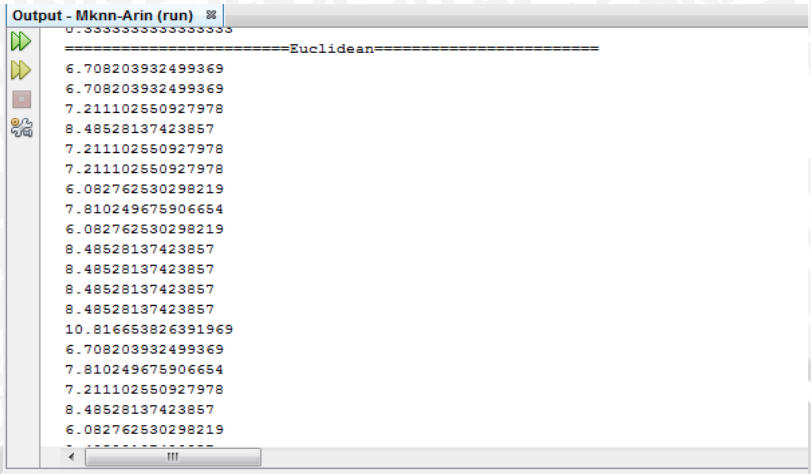
Source Code 4.6 Proses Sorting Nilai

4.3 Implementasi Program Perhitungan Manual

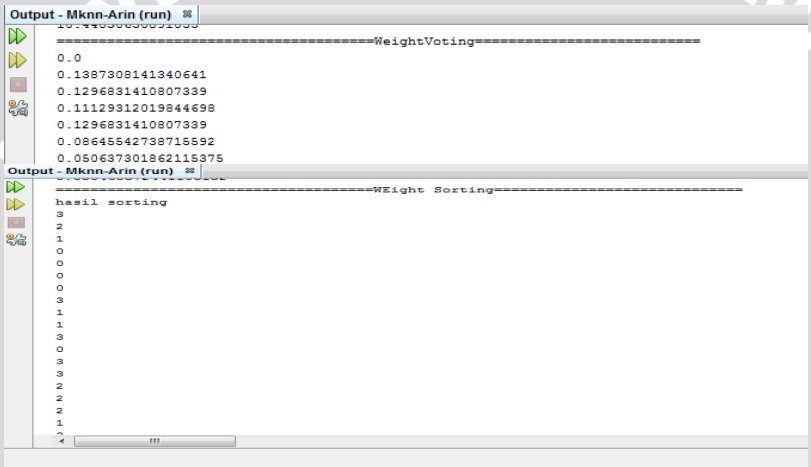
Perhitungan manual pada bagian 3.5 diimplementasikan untuk menguji hasil dari implementasi program. Hasil implementasi perhitungan dapat dilihat pada gambar 4.1, 4.2, 4.3, dan 4.4.



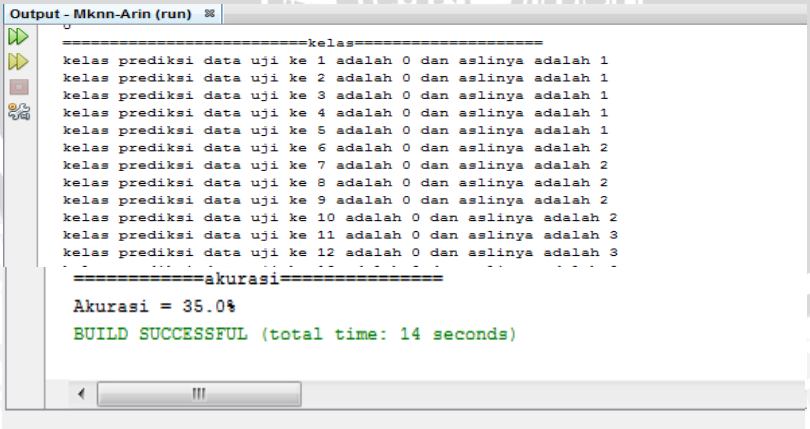
Gambar 4.1 Hasil Perhitungan Manual Validitas



Gambar 4.2 Hasil Perhitungan Manual Euclidean



Gambar 4.3 Hasil Perhitungan Manual Weight Voting



Gambar 4.4 Hasil Prediksi Kelas dan Akurasi

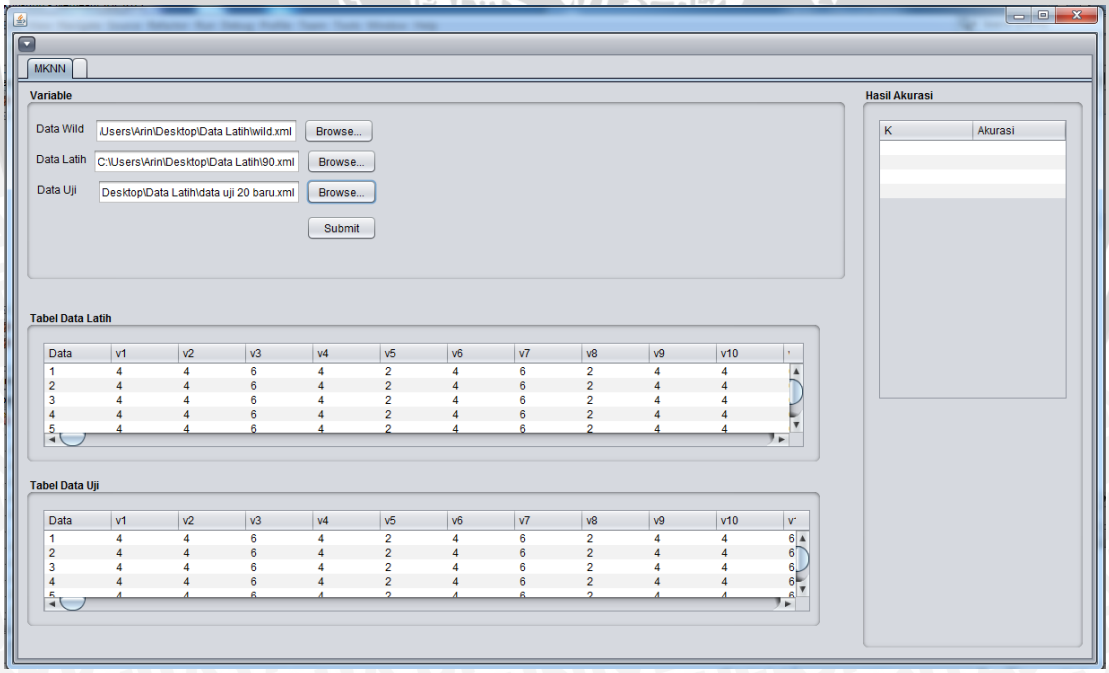
4.4 Implementasi Antarmuka

Implementasi antarmuka pada aplikasi penentuan jenis kanker pada struktur protein adalah sebagai berikut:

1. Form Hasil Preprocessing Data
   Form ini bertujuan untuk menampilkan hasil transform data string ke dalam bentuk numeric dengan menginputkan data wild (normal), data latih dan data uji.
2. Form Akurasi (berdasarkan nilai k)
   Form ini bertujuan untuk mengetahui nilai keseluruhan akurasi berdasarkan nilai k yang disesuaikan sebanyak data latih yang digunakan.

4.4.1 Form Preprocessing Data

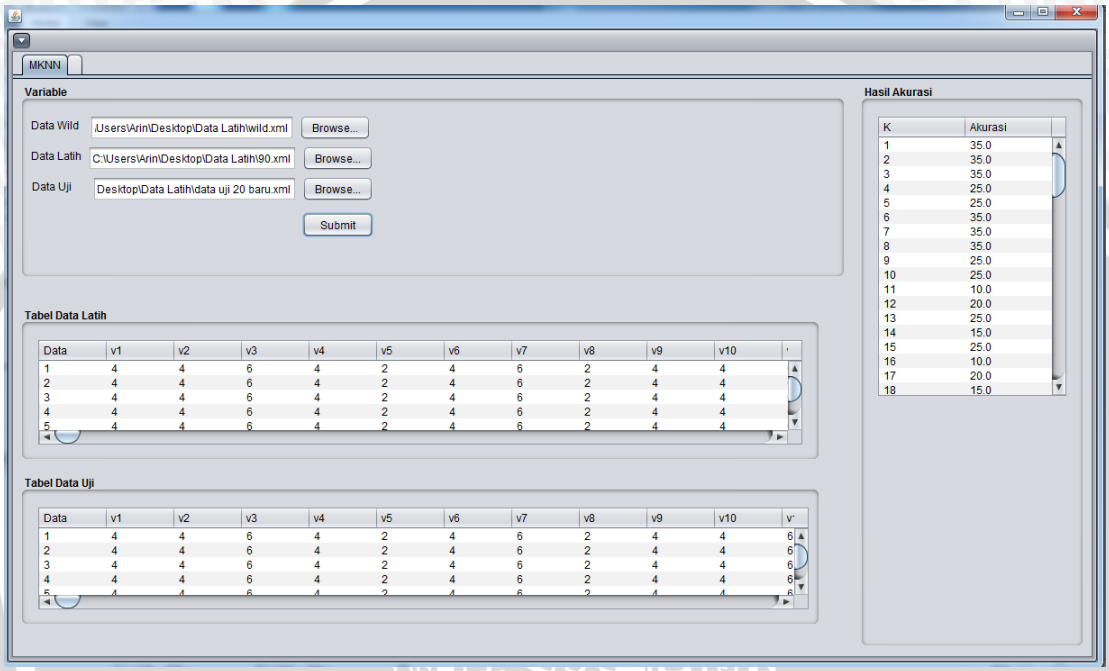
Form preprocessing data ini digunakan untuk menampilkan hasil dari transform data protein dari bentuk string ke dalam bentuk numerik. Dengan memasukkan data wild (normal), kemudian memasukkan data latih yang berupa string dan dibandingkan dengan data wild, hasil dari transform data latih bernilai numerik ditampilkan pada tabel data latih. Selanjutnya masukkan data uji yang akan dibandingkan dengan data wild dan hasil transform data uji akan terlihat pada tabel data uji. Antarmuka dari form preprocessing data ditunjukkan pada gambar 4.5.



Gambar 4.5 Form Preprocessing Data

4.4.2 Form Akurasi (Berdasarkan Nilai k)

Pada form ini bertujuan untuk menampilkan akurasi dari sistem, dengan menampilkan keseluruhan akurasi yang didapatkan sebanyak nilai k. Nilai k maksimum adalah sebanyak data latih yang digunakan. Untuk memulai proses akurasi hanya dengan mengklik tombol submit dan akan ditampilkan hasil akurasi berdasar nilai k sesuai dengan banyaknya data latih. Antarmuka dari form akurasi (berdasarkan nilai k) ditampilkan pada gambar 4.6.



Gambar 4.6 Form Akurasi (Berdasarkan Nilai k)

Contoh dalam gambar 4.6 yang digunakan adalah sebanyak 90 data protein untuk data latih dan 20 data protein untuk data uji, kemudian dilakukan proses preprocessing yang akan berlanjut untuk menghitung akurasi berdasarkan nilai k dengan mengklik tombol submit. Hasil dari akurasi dapat dilihat dalam gambar 4.6 pada tabel akurasi. Hal ini dapat memudahkan dalam melihat perbandingan hasil akurasi antara nilai k yang berbeda.

BAB 5 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis dari pengujian penentuan jenis kanker berdasarkan struktur protein dengan menggunakan metode *Modified K-Nearest Neighbor*.

5.1 Pengujian Sistem

Pada pengujian sistem ini digunakan 3 jumlah dataset yang berbeda, yaitu 100, 150, dan 200. Pada masing-masing dataset yang ditentukan akan dibagi secara acak menjadi data latih dan data uji, dengan persentase data latih : data uji sebesar 80%:20%, 70%:30% dan 60%:40%. Data yang digunakan dalam sistem dibagi menjadi beberapa kelas, yaitu *breast cancer, colorectal cancer,* dan *lung cancer*. Pengujian sistem ini dilakukan untuk mengetahui pengaruh jumlah data latih terhadap tingkat akurasi, uji pengaruh jumlah data uji terhadap tingkat akurasi, dan uji pengaruh nilai k terhadap tingkat akurasi.

5.1.1 Pengujian Pada Jumlah Dataset 100

Pada pengujian persentase data uji sebesar 20% didapatkan hasil pengujian dengan nilai akurasi maksimum pada jumlah dataset 100 adalah 65% dan akurasi minimum terdapat pada akurasi 40%. Setiap kenaikan nilai k maka hasil akurasi dari dataset 100 ini semakin menurun. Pada pengujian persentase 30% didapatkan akurasi maksimum pada 46,67% dan akurasi minimum pada 40%. Pada persentase 40% akurasi maksimumnya adalah 47,5% dan minimumnya adalah 40%. Dari pengujian dataset 100 ini nilai k=1 sampai dengan k=20 mengalami penurunan yang merata dan stabil. Hasil uji coba pada dataset 100 ditunjukkan Tabel 5.1.

Tabel 5.1 Tabel Hasil Uji Coba pada Jumlah Dataset 100

Jumlah Dataset	Nilai k	Persentase Data Uji			Akurasi Rata-rata(%)
		20%	30%	40%	
100	1	65	46.67	47.5	53.05
	2	50	43.33	40	44.44
	3	40	40	40	40
	4	40	40	40	40
	5	40	40	40	40
	6	40	40	40	40
	7	40	40	40	40
	8	40	40	40	40
	9	40	40	40	40
	10	40	40	40	40
	11	40	40	40	40

	12	40	40	40	40
	13	40	40	40	40
	14	40	40	40	40
	15	40	40	40	40
	16	40	40	40	40
	17	40	40	40	40
	18	40	40	40	40
	19	40	40	40	40
	20	40	40	40	40
Akurasi Rata-rata(%)		41.75	40.5	40.37	40.87

5.1.2 Pengujian Pada Jumlah Dataset 150

Pada pengujian persentase data uji sebesar 20% didapatkan hasil pengujian dengan nilai akurasi maksimum pada jumlah dataset 150 adalah 76,67% dan akurasi minimum terdapat pada akurasi 43,33%. Setiap kenaikan nilai k maka hasil akurasi dari dataset 150 ini semakin menurun. Pada pengujian persentase 30% didapatkan akurasi maksimum pada 62,22% dan akurasi minimum pada 40%. Pada persentase 40% akurasi maksimumnya adalah 55% dan minimumnya adalah 36,67%. Dari pengujian dataset 150 ini nilai k=1 sampai dengan k=6 mengalami peningkatan dan penurunan yang tidak stabil stabil, kemudian pada k=7 sampai dengan k=20 terjadi peningkatan akurasi yang stabil. Hasil uji coba pada dataset 150 ditunjukkan Tabel 5.2.

Tabel 5.2 Tabel Hasil Uji Coba pada Jumlah Dataset 150

Jumlah Dataset	**Nilai k**	**Persentase Data Uji**			**Akurasi Rata-rata(%)**
		20%	**30%**	**40%**	
150	1	76.67	62.22	55	64.63
	2	46.67	42.22	36.67	41.85
	3	50.00	46.67	41.67	46.11
	4	46.67	42.22	40	42.96
	5	50.00	40	40	43.33
	6	43.33	40	38.33	40.56
	7	46.67	40	40	42.22
	8	46.67	40	40	42.22
	9	46.67	40	40	42.22
	10	46.67	40	40	42.22
	11	46.67	40	40	42.22
	12	46.67	40	40	42.22
	13	46.67	40	40	42.22
	14	46.67	40	40	42.22
	15	46.67	40	40	42.22

	16	46.67	40	40	42.22
	17	46.67	40	40	42.22
	18	46.67	40	40	42.22
	19	46.67	40	40	42.22
	20	46.67	40	40	42.22
Akurasi Rata-rata(%)		48.33	41.67	40.58	43.53

5.1.3 Pengujian Pada Jumlah Dataset 200

Pada pengujian persentase data uji sebesar 20% didapatkan hasil pengujian dengan nilai akurasi maksimum pada jumlah dataset 200 adalah 52,5% dan akurasi minimum terdapat pada akurasi 35%. Dari persentase sebesar 20% ini nilai k=1 sampai dengan k=13 mengalami peningkatan dan penurunan yang tidak stabil, kemudian pada k=14 sampai dengan k=20 memiliki nilai akurasi yang stabil. Setiap kenaikan nilai k maka hasil akurasi dari dataset 200 ini semakin menurun. Pada pengujian persentase 30% didapatkan akurasi maksimum pada 53,33% dan akurasi minimum pada 31,67%. Dari persentase sebesar 30% ini nilai k=1 sampai dengan k=12 mengalami peningkatan dan penurunan yang tidak stabil, kemudian pada k=13 sampai dengan k=20 memiliki nilai akurasi yang stabil. Pada persentase 40% akurasi maksimumnya adalah 61,25% dan minimumnya adalah 32,5%. Dari persentase sebesar 40% ini nilai k=1 sampai dengan k=9 mengalami peningkatan dan penurunan yang tidak stabil, kemudian pada k=10 sampai dengan k=20 memiliki nilai akurasi yang stabil. Hasil uji coba pada dataset 200 ditunjukkan Tabel 5.2.

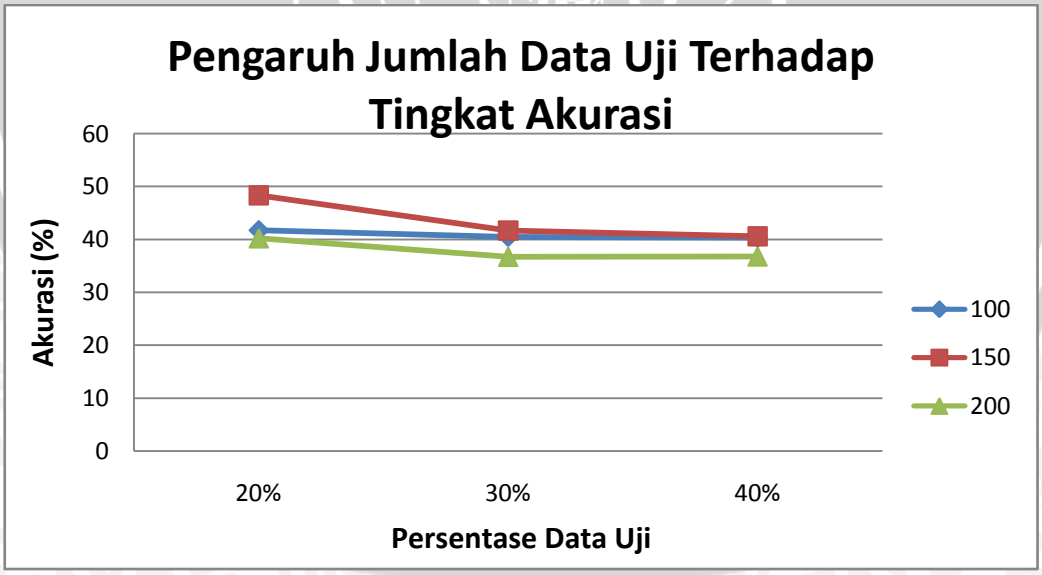
Jumlah Dataset	Nilai k	Persentase Data Uji			Akurasi Rata-rata(%)
		20%	30%	40%	
200	1	52.5	53.33	61.25	55.69
	2	35	38.33	37.5	36.94
	3	40	41.67	47.5	43.06
	4	47.5	40	35	40.83
	5	52.5	45	42.5	46.67
	6	47.5	40	42.5	43.33
	7	42.5	41.67	40	41.39
	8	37.5	33.33	33.75	34.86
	9	45	40	38.75	41.25
	10	42.5	38.33	32.5	37.78
	11	42.5	35	32.5	36.67
	12	35	35	32.5	34.17
	13	40	31.67	32.5	34.72
	14	35	31.67	32.5	33.06
	15	35	31.67	32.5	33.06
	16	35	31.67	32.5	33.06
	17	35	31.67	32.5	33.06

	18	35	31.67	32.5	33.06
	19	35	31.67	32.5	33.06
	20	35	31.67	32.5	33.06
Akurasi Rata-rata(%)		40.25	36.75	36.81	37.93

5.2 Analisa Hasil

5.2.1 Pengujian Pengaruh Jumlah Data Uji Terhadap Tingkat Akurasi

Pada pengujian pengaruh jumlah data uji terhadap tingkat akurasi, terlihat bahwa jumlah data training sangat berpengaruh pada besar nilai akurasi yang dihasilkan. Dalam pengujian jumlah dataset 100 untuk persentase data uji 20% menghasilkan rata-rata akurasi sebesar 41,75%, persentase data uji 30% menghasilkan rata-rata akurasi sebesar 40,5% dan persentase data uji 40% menghasilkan rata-rata akurasi sebesar 40,38%. Dalam pengujian jumlah dataset 150 untuk persentase data uji 20% menghasilkan rata-rata akurasi sebesar 48,33%, persentase data uji 30% menghasilkan rata-rata akurasi sebesar 41,67% dan persentase data uji 40% menghasilkan rata-rata akurasi sebesar 40,58%. Untuk pengujian jumlah dataset 200 untuk persentase data uji 20% menghasilkan rata-rata akurasi sebesar 40,25%, persentase data uji 30% menghasilkan rata-rata akurasi sebesar 36,75% dan persentase data uji 40% menghasilkan rata-rata akurasi sebesar 36,81%. Grafik pengaruh jumlah data uji terhadap tingkat akurasi dapat dilihat pada grafik 5.1.



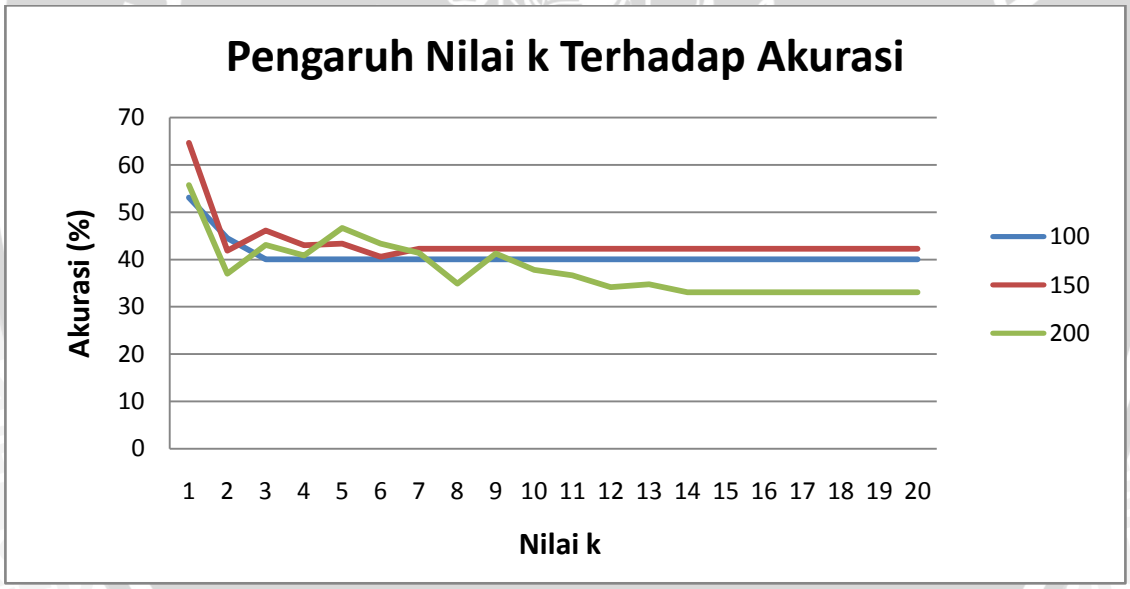
Gambar 5.1 Grafik Pengujian Pengaruh Jumlah Data Uji Terhadap Tingkat Akurasi

Dari grafik diatas dapat disimpulkan bahwa semakin tinggi persentase data uji belum tentu tingkat akurasi yang didapat semakin tinggi dan semakin tinggi persentase data uji sebenarnya berpeluang semakin banyak kebenaran dalam

pengklasifikasian data uji, namun berpeluang juga semakin banyak data uji yang salah dalam pengklasifikasiannya. Hal ini dapat terjadi karena banyaknya data latih yang digunakan lebih banyak jumlahnya daripada jumlah data ujinya dan kemungkinan pada pengujian data uji tersebut sebaran data latih yang digunakan representatif. Representatif dapat diartikan sebagai variasi data yang diujikan sudah banyak diwakili pada data latihnya.

5.2.2 Pengujian Pengaruh Nilai k Terhadap Tingkat Akurasi

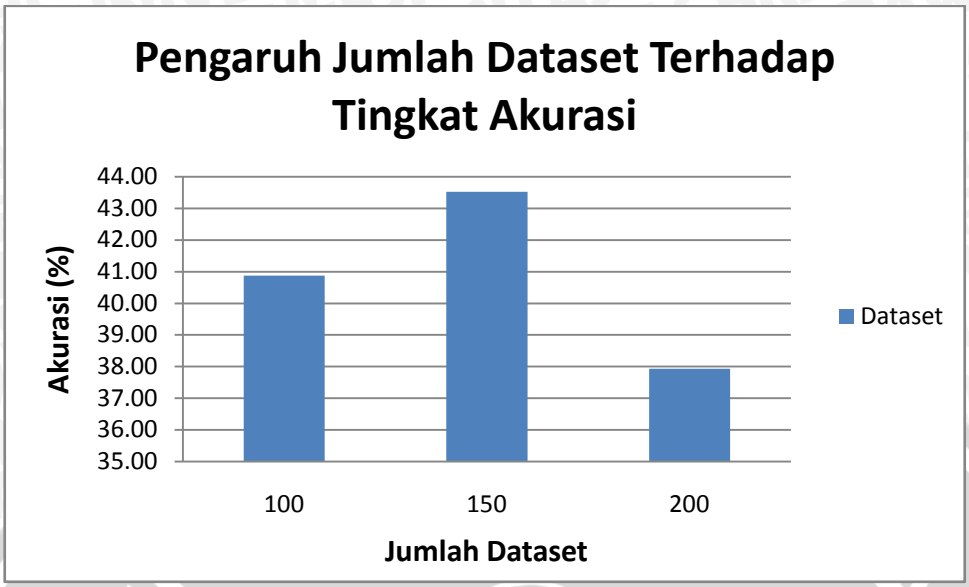
Berdasarkan hasil uji coba dengan metode MKNN pada masing-masing dataset untuk pengujian pengaruh nilai k terhadap tingkat akurasi dihasilkan pada setiap dataset 100, 150 dan 200 rata-rata nilai akurasi pada analisa hasil pengaruh nilai k terhadap tingkat akurasi cenderung semakin menurun sejalan dengan adanya penambahan nilai k. Akurasi maksimum cenderung terdapat pada k=1, karena semakin kecil nilai k akan mengurangi noise dan perbandingan dengan tetangga terdekat hanya sedikit. Grafik pengaruh nilai k terhadap tingkat akurasi ditunjukkan Gambar 5.2.



Gambar 5.2 Grafik Pengujian Pengaruh Nilai k Terhadap Tingkat Akurasi

5.2.3 Pengujian Pengaruh Dataset Terhadap Tingkat Akurasi

Berdasarkan pengujian yang dilakukan, terlihat bahwa jumlah dataset berpengaruh pada nilai akurasi yang dihasilkan. Grafik rata-rata pengaruh dataset terhadap tingkat akurasi ditunjukkan Gambar 5.3.



Gambar 5.3 Grafik Pengujian Pengaruh Dataset Terhadap Tingkat Akurasi

Dalam pengujian jumlah dataset dihasilkan nilai akurasi rata-rata dari dataset 100 sebesar 40,88%, akurasi rata-rata dari dataset 150 sebesar 43,53% dan akurasi rata-rata dari dataset 200 sebesar 37,94%. Pada pengujian pengaruh jumlah dataset terhadap tingkat akurasi disimpulkan bahwa semakin sedikit dataset maka nilai akurasi yang didapat akan semakin tinggi. Tetapi pada dataset 150, nilai akurasi yang didapat lebih tinggi dibandingkan dengan data latih 100 dan 200. Semakin banyak dataset belum tentu semakin tinggi tingkat akurasinya, karena yang mempengaruhi tinggi atau rendahnya tingkat akurasi adalah sebaran data dan nilai k.

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Tahapan prediksi kanker berdasarkan struktur protein yaitu preprocessing data, menghitung nilai validitas data latih, menghitung jarak Euclidean data latih, dan menghitung nilai weight voting.
2. Rata-rata nilai akurasi maksimum yang dihasilkan sistem sebesar 43,53% pada jumlah dataset 150 dan akurasi minimumm didapat sebesar 37,93% yang dihasilkan pada saat jumlah dataset 200.
3. Pada pengujian pengaruh jumlah data uji terhadap tingkat akurasi disimpulkan semakin tinggi persentase data uji belum tentu tingkat akurasi yang didapat semakin tinggi dan semakin tinggi persentase data uji sebenarnya berpeluang semakin banyak kebenaran dalam pengklasifikasian data uji, namun berpeluang juga semakin banyak data uji yang salah dalam pengklasifikasiannya.
4. Pada uji pengaruh nilai k terhadap tingkat akurasi, diperoleh nilai k yang paling tinggi tingkat akurasinya adalah pada k=1. Rata-rata nilai akurasi cenderung semakin menurun dengan sejalannya penambahan nilai k, ini dikarenakan adanya kelas yang mendominasi pada dataset, sehingga untuk data yang diambil dalam perhitungan memiliki kelas yang sama.
5. Pada uji pengaruh dataset terhadap tingkat akurasi diperoleh kesimpulan semakin banyak jumlah dataset akan mempengaruhi tingkat akurasi yang dihasilkan.

6.2 Saran

Berkaitan degnan penelitian ini, penulis menemukan beberapa hal yang mungkin peru dikembangkan untuk kedepannya, yaitu:

1. Penentuan prediksi kanker berdasarkan struktur protein hendaknya dilakukan oleh para ahli sehingga data yang digunakan untuk penelitian menjadi lebih valid.
2. Dapat menggunakan metode yang lain untuk pengklasifikasian yang dapat digabungkan dengan metode KNN
3. Dapat dilakukan pengujian terhadap metode KNN dengan melakukan pengelompokkan terlebih dahulu pada dataset yang digunakan sebagai perbandingan keakuratan klasifikasi.

Daftar Pustaka

Adelia, Farisa, 2013, *Penentuan Potensi Tsunami Akibat Gempa Bumi Bawah Laut Dengan Metode Modified K-Nearest Neighbor (MKNN)*, Universitas Brawijaya, Malang

Destuardi, I., dan Sumpeno, S. 2009, *Klasifikasi Emosi Untuk Teks Bahasa Indonesia Menggunakan Metode Naive Bayes*, Institut Teknologi Sepuluh Nopember, Surabaya

Dor, Shifra Ben, 2007, Scoring Matrices, Weizmann Institute Of Science, Rehovot

Eddy, 2004, PAM and BLOSUM Substitution Matices, Nature Biotech 22: 1035-1036

Harahap, Ikhsanuddin Ahmad, 2004, Perawatan Pasien Dengan Kolostomi Pada Penderita Cancer Colorectal, Universitas Sumatera Utara, Medan

Huda A, M. Abdullah, 2013, *Implementasi Modified K-Nearest Neighbor Untuk Tanaman Hortikultura*, Universitas Brawijaya, Malang

Kurnianti, Ria, 2013, *Penggunaan Metode Pengelompokkan K-Means Pada Klasifikasi KNN Untuk Penentuan Jenis Kanker Berdasarkan Susunan Protein,* Universitas Brawijaya, Malang

Meristika, Yanita Selly, 2013, *Perbandingan K-Nearest Neighbor dan Fuzzy K-Nearest Neighbor pada Diagnosis Penyakit Diabetes Melitus,* Universitas Brawijaya, Malang

Mulyana, Sri, 2013, Penerapan *Hidden Markov Model Dalam Clustering Sequence Protein Globin*, Universitas Brawijaya, Malang

Nugroho, Anto Satriyo, 2003, *Bioinformatika dan Pattern Recognition,* IlmuKomputer.Com

Nurmaya, 2008, *Karakteristik Wanita Penderita Kanker Payudara Rawat Inap Di Rumah Sakit St. Elisabeth Medan Tahun 2003-2007,* Universitas Sumatera Utara, Medan

Prayitno, Adi, Ruben Darmawan, 2005, *Ekspresi Protein p53, Rb, dan c-myc pada Kanker Serviks Uteri dengan Pengecatan Immunohistokimia,* Universitas Sebelas Maret, Surakarta

Prayuni, Kinasih, 2008, *Isolasi DNA Genom Padi (Oryza sativa L.) Kultivar, Rojolele, Nipponbare, dan Batutegi,* Universitas Indonesia, Depok

Rahman, Dila, 2012, http://katakatatanpabahasa.blogspot.com/2012/07/kode-genetik-sandi-3-huruf-yang.html

Ridwan, Ardilla Ayu Dewanti, 2013, *Pengenalan Gender Memanfaatkan Wajah Manusia Dengan Menggunakan Metode Klasifikasi Nearest Neighbor*, Universitas Kristen Satya Wacana, Salatiga

Rustam, Yepy Hardi, 2010, http://sciencebiotech.net/struktur-molekul-protein/

S, Christine N.S, 2010, Hubungan Merokok Dengan Kanker Paru di RSUP Haji Adam Malik Tahun 2009, Universitas Sumatera Utara, Medan

Sari, Dr. Mutiara Indah, 2007, *Struktur Protein*, Universitas Sumatera Utara, Medan

Sudarka, Ir. Wayan, 2009, Pemuliaan Kelainan Genetik dan Sitogenesik Pada Tanaman, Universitas Udayana, Bali

Sulistyandari, 2012, *Penerapan Algoritma Modified K-Nearest Neighbor (MKNN) Untuk Mengklasifikasikan Letak Protein Pda Bakteri E-Coli*, Universitas Brawijaya, Malang

Sunjiana, 2010, Aplikasi Mining Data Mahasiswa Dengan Metode Klasifikasi Decision Tree, Universitas Widyatama, Jawa Barat

Syaifudin, Mukh, 2007, *Gen Penekan Tumor p53, Kanker dan Radiasi Pengion,* Pusat Teknologi Keselamatan dan Metrologi Radiasi, Batam

Utama, Andi, 2003, *Peranan Bioinformatika Dalam Dunia Kedokteran,* IlmuKomputer.Com

Uyha, 2010, http://uyha06.wordpress.com/ilmu-keperawatan/laporan-praktikum-biokimia/protein/

Warianto, Chaidar, 2011, Mutasi, Universitas Airlangga, Surabaya

Widiawati, Laili Ratna, 2011, *Penetapan Kadar Protein Pada Daun Katuk (Sauropus androgynus) Segar Yang Disimpan Pada Suhu Ruang (25^{o}C) Dan Suhu Dingin (13^{o}C),* Universitas Muhammadiyah Semarang, Semarang

Wirdasari, Dian, 2011, Penerapan Data Mining Untuk Mengolah Data Penempatan Buku Di Perpustakaan SMK TI PAB 7 Lubuk Pakam Dengan Metode Association Rule, Universitas Sumatera Utara, Medan

Zainuddin, Sofa, 2014, *Penerapan Algoritma Modified K-Nearest Neighbour (M-KNN) Pada Pengklasifikasian Penyakit Tanaman Kedelai,* Universitas Brawijaya, Malang