PENCARIAN MOTIF SEKUENS DNA DENGAN ALGORITMA *PREFIXSPAN* UNTUK DIAGNOSA PENYAKIT KANKER PAYUDARA

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

ILHAM YULIANTORO

105090600111029

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2014

LEMBAR PERSETUJUAN

PENCARIAN MOTIF SEKUENS DNA DENGAN ALGORITMA *PREFIXSPAN* UNTUK DIAGNOSA PENYAKIT KANKER PAYUDARA

SKRIPSI
KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan untuk
Mencapai gelar Sarjana Komputer

Disusun Oleh:
ILHAM YULIANTORO
105090600111029

Telah diperiksa dan disetujui oleh
Dosen Pembimbing

Pembimbing I,	**Pembimbing II,**
Lailil Muflikhah, S.kom M.Sc	Widodo, S.Si, M.Si, Ph.D Med Sc.
NIP. 19741113 200501 2 001	**NIP. 19730811 200003 1 002**

LEMBAR PENGESAHAN

PENCARIAN MOTIF SEKUENS DNA DENGAN ALGORITMA *PREFIXSPAN* UNTUK DIAGNOSA PENYAKIT KANKER PAYUDARA

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI
Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh :

Ilham Yuliantoro
105090600111029

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 11 April 2014
dan dinyatakan memenuhi syarat untuk memperoleh
gelar sarjana dalam bidang Ilmu Komputer

Penguji I,	**Penguji II,**
Dian Eka Ratnawati, S.Si., M.Kom	Imam Cholissodin S.Si., M.Kom
NIP. 19730619 200212 2 001	**NIK. 850719 16 1 1 0422**

Penguji III,

Novanto Yudistira, S.Kom., M.Sc.
NIK. 83110 16 1 1 0425

Mengetahui,
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, MT.
NIP. 196708011992031001

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 di dalam 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, Mei 2014

Mahasiswa,

Ilham Yuliantoro

NIM. 105090600111029

KATA PENGANTAR

Syukur dan alhamdulillah penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul: " **PENCARIAN MOTIF SEKUENS DNA DENGAN ALGORITMA *PREFIXSPAN* UNTUK DIAGNOSA PENYAKIT KANKER PAYUDARA** ".

Skripsi ini diajukan sebagai syarat ujian skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi Dan Ilmu Komputer (PTIIK), Program Studi Informatika/Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Ibu Lailil Muflikhah, S.kom., M.Sc., selaku Dosen Pembimbing Skripsi pertama yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
2. Bapak Widodo, S.Si, M.Si, Ph.D, Med Sc, selaku Dosen Pembimbing Skripsi kedua yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
3. Drs. Marji, MT. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya
4. Ir. Sutrisno, MT., selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
5. Orang tua dan saudara Penulis yang telah selalu mendukung dan mendoakan sehingga terselesaikannya skripsi ini.
6. Sahabat Penulis serta teman – teman ilmu komputer angkatan 2010 yang memotivasi dan saling menyemangati.
7. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
8. Segenap staff dan karyawan di Program Studi Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu dalam hal administrasi Penulis dalam pelaksanaan penyusunan skripsi ini.

9. Penulis menyadari bahwa skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat Penulis harapkan dari berbagai pihak demi penyempurnaan penulisan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, Mei 2014

Penulis,

Ilham Yuliantoro

NIM. 105090600111029

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SOURCECODE

ABSTRAK

Ilham Yuliantoro. 2014 : Pencarian Motif Sekuens DNA dengan Algoritma PrefixSpan untuk Diagnosa Penyakit Kanker Payudara.
Dosen Pembimbing : Lailil Muflikhah, S.Kom, M.Sc. dan Widodo, S.Si, M.Si, Ph.D Med Sc.

Di negara maju kanker menjadi penyebab kematian kedua setelah penyakit jantung. Salah satu jenis kanker yang sering mengakibatkan kematian adalah kanker payudara. Menurut WHO, pada tahun 2000 diperkirakan 1,2 juta wanita terdiagnosis kanker payudara dan lebih dari 700,000 meninggal karenanya. Salah satu algoritma yang mampu digunakana dalam diagnosa penyakit kanker payudara dalam proses pecarian motif sekuens DNA adalah prefixspan yang merupakan algoritma yang tergolong dalam Sequential Pattern Mining. Prefix-Projected Sequential Pattern Growth (PrefixSpan) adalah algoritma yang memakai pendekatan pengembangan sequence untuk mencari sequential pattern. PrefixSpan akan mencari frequent sequence satu elemen dan kemudian mengembangkan sequence-sequence tersebut dengan cara menambahkan elemen satu persatu. Dalam penelitian ini digunakan Exon ke-7 dari 20 gen DNA TP 53 penderita kanker payudara. Uji coba pengaruh minimum support dilakukan dengan melakukan pengujian minimum support dari 2, 4 hingga 12 untuk dilakukan pengamatan terhadap nilai lift rasio dari motif yang dihasilkan. Hasil Ujicoba yang dilakukan menunjukkan pengaruh bahwa semakin besar minimum support maka berdampak kurang baik terhadap motif yang dihasilkan untuk uji coba pada 20 data. Namun akurasi motif yang dihasilkan dalam uji coba ini mencapai rata-rata 100%. Hal ini menunjukkan algoritma prefixspan mampu dan cukup baik dalam melakukan pencarian motif sekuens DNA untuk deteksi penyakit kanker payudara.

Kata kunci : Sequential Patern Mining, Prefixspan, Kanker Payudara, Sekuens DNA

ABSTRACT

Ilham Yuliantoro. 2014 : Search of DNA Sequences Motif with PrefixSpan Algorithm for Diagnosis of Breast Cancer

Advisor : Lailil Muflikhah, S.Kom, M.Sc. and Widodo, S.Si, M.Si, Ph.D Med Sc.

In developed countries, cancer is the second number that cause of death after heart disease. One type of cancer that effect to death is breast cancer. Acording to WHO, in 2000, 1.2 million women are diagnosed with breast cancer and more than 700,000 women die from it. One of the algorithms that can be used to find the pattern is prefixspan which is belonging to Sequential Pattern Mining. PrefixSpan (Prefix - Projected Sequential Pattern Growth) is an algorithm that use the concept of sequence to searh of sequential pattern. PrefixSpan will finde frequent sequences of the elements and then develop it by adding elements one by one . This study is use 20 DNA TP 53 of breast cancer patients in Exon 7 . Trial of this study are use the conducted of minimum support which is use minimum support 2 , 4 until 12 for observe the value of liftratio from the generated motifs . The Tests that has been done is show that a high minimum support give the bad effect in the lift ratio of the resulting motifs for trial on 20 data. However, the accuracy of the resulting motif are reaches 100 % . This is shows that prefixspan algorithm is capable in searching DNA sequence motifs for the detection of breast cancer .

Keywords : Mining Sequential Patern , Prefixspan , Breast Cance, DNA Sequences.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di negara yang telah maju penyakit kanker merupakan penyebab kematian kedua setelah penyakit jantung. Penyakit kanker tergolong penyakit yang ganas dan berbahaya karena sulit untuk disembuhkan. Menurut badan penelitian kanker *World Cancer Research Fund*, jumlah kasus kanker meningkat 20% kurang dari sepuluh tahun terakhir dan saat ini berjumlah 12 juta setiap tahun.

Terdapat berbagai macam jenis kanker yang telah ditemukan di dunia. Salah satu diantaranya adalah kanker payudara. Penyakit kanker payudara terbilang penyakit kanker yang paling umum menyerang kaum wanita. Menurut WHO 8-9% wanita akan mengalami kanker payudara. Namun meski demikian pria pun memiliki kemungkinan mengalami penyakit ini dengan perbandingan 1 : 1000. Kanker payudara pada umumnya menyerang wanita usia 50 tahun ke atas (lansia). Data WHO menunjukkan bahwa 78% kanker payudara terjadi pada wanita usia 50 tahun ke atas dan hanya 6%-nya terjadi pada mereka yang berusia kurang dari 40 tahun.

Kanker Payudara merupakan penyebab utama kematian pada wanita akibat kanker. Menurut WHO, pada tahun 2000 diperkirakan 1,2 juta wanita terdiagnosis kanker payudara dan lebih dari 700,000 meninggal karenanya. Di Amerika Serikat, setiap tahunnya 44,000 pasien meninggal sedangkan di Eropa lebih dari 165,000 pasien yang meninggal karena penyakit ini. Setelah menjalani perawatan, sekitar 50% pasien mengalami kanker payudara stadium akhir dan hanya bertahan hidup 18 – 30 bulan.

Kanker umumnya disebabkan karena adanya mutasi gen, Salah satunya adalah *p53*. Mutasi pada gen tersebut telah banyak teridentifikasi menyebabkan berbagai jenis kanker. Mutasi tersebut mengakibatkan perbedaan urutan asam amino protein *p53* [PUS-96]. Mutasi *p53* atau *TP53* adalah perubahan genetik yang paling sering ditemukan pada kanker manusia. Terdapat 30.000 mutasi somatik dari berbagai jenis kanker pada database *TP53* yang terkumpul lebih dari 20 tahun. Analisis mutasi menyebabkan banyak penelitian dan informasi tentang protein *TP53* dan fungsinya. Kemajuan terbaru dalam metodologi *sequencing*

genom kanker berdampak pada tatalaksana terapi penyembuhan dan manajemen data [SAN-01]. Menurut Soussi, analisa terhadap pola mutasi *p53* telah menjadi hal yang esensial terhadap hal-hal yang mengibatkan kanker. Dari hasil penelitian yang telah dilakukan didapatkan kejarangan mutasi dikaitkan dengan aktifitas yang normal dari protein *p53*.

Tingginya angka kematian yang disebabkan oleh penyakit kanker, membuat deteksi dini penyakit ini menjadi langkah penting untuk dikembangkan hingga saat ini. Salah satu bidang yang diharapkan mampu memberikan perannya adalah bidang bioinformatika. Bioinformatika merupakan kajian yang memadukan disiplin biologi molekul, matematika dan teknik informasi (TI). Ilmu ini didefinisikan sebagai aplikasi dari alat komputasi dan analisa untuk menangkap dan menginterpretasikan data-data biologi molekul. Biologi molekul sendiri juga merupakan bidang interdisipliner, mempelajari kehidupan dalam level molekul [SAN-01]. Bioinformatika mempunyai peranan yang sangat penting, diantaranya adalah untuk manajemen data-data biologi molekul, terutama sekuen DNA dan informasi genetika yang memiliki volume yang cukup besar.

Salah satu proses dibidang TI (informasi) yang dapat dimanfaatkan untuk mencari pola DNA gen *p53* penyebab kanker adalah *Sequential Pattern Mining*. Sequential pattern mining merupakan proses data mining yang menghasilkan pengetahuan mengenai serangkaian kejadian-kejadian yang memiliki frekuensi kemunculan yang melebihi nilai threshold yang ditentukan. Pencarian pola DNA penyebab kanker ini diharapkan nantinya membantu dalam proses diagnosa dini penyakit kanker. Dalam skripsi ini akan dijelaskan proses pencarian pola sekuens DNA penyebab penyakit kanker dengan algoritma prefixspan [SAP-06].

Prefixspan adalah salah satu algoritma yang merupakan bagian dari *Sequential Pattern Mining*. Alasan dipilihnya algoritma ini dibandingkan algoritma *sequential patern* yang lain seperti *Aprioriall* adalah karena kinerja algoritma ini dianggap yang terbaik. Seperti hasil penelitian yang dilakukan oleh Danny & Rully (2006), dalam penelitiannya yang berjudul Analisis Kinerja Algoritma *Prefixspan* dan *AprioriAll* pada penggalian pola sekuensial menyimpulkan bahwa secara umum durasi eksekusi dan utilasi memori

Prefixspan lebih kecil daripada *AprioriAll* dan secara grafis dapat dikatakan bahwa *Prefixspan* lebih skalabel dan terpercaya daripada *AprioriAll* [SAP-06].

1.2 Rumusan Masalah

Rumusan masalah dalam skripsi ini adalah :

1. Bagaimana mencari motif sekuens DNA dengan algoritma *prefixspan* dari penderita penyakit kanker payudara.
2. Bagaimana performansi motif sekuens DNA yang dihasilkan dengan algoritma *prefixspan* untuk dignosa penyakit kanker payudara.

1.3 Batasan Masalah

Dari permasalahan yang dirumuskan diatas, maka batasan permasalahan yang digunakan untuk merancang dan membuat sistem ini adalah :

1. Jenis kanker yang dipilih untuk diteliti dalam skripsi ini adalah jenis kanker payudara.
2. Jenis DNA yang digunakan untuk diteliti dalam skripsi ini adalah gen TP53.
3. Jenis Exon dari DNA yang digunakan dalam pengujian adalah Exon ke-7
4. Spesies penderita kanker payudara yang diteliti dalam skripsi ini adalah jenis homo sapiens (manusia)
5. Data DNA kanker payudara yang digunakan sebagai data dalam skripsi ini di unduh dari situs www.ncbi.nlm.nih.gov

1.4 Tujuan

Tujuan dari skripsi ini adalah :

1. Melakukan pencarian motif sekuens DNA penderita kanker payudara dengan algoritma *prefixspan* untuk diagnosa penyakit kanker payudara
2. Menguji performansi motif sekuens DNA yang dihasilkan dengan algoritma *prefixspan* untuk diagnosa penyakit kanker payudara.

1.5 Manfaat

Manfaat yang dapat dihasilkan dari skripsi ini adalah :

1. Mempermudah dan membantu dalam proses diagnosa dini terhadap pasien pengidap penyakit kanker payudara.

2. Mendapatkan pola-pola sekuens DNA spesifik terhadap seseorang yang terkena penyakit kanker payudara sehingga dapat digunakan sebagai data diagnosa seseorang menderita kanker payudara.

1.6 Sistematika Penulisan

Pembuatan tugas akhir ini dilakukan dengan sistematika penulisan sebagai berikut :

1. BAB I PENDAHULUAN

   Berisi latar belakang, permasalahan, tujuan, batasan masalah, dan manfaat serta sistematika penulisan skripsi.

2. BAB II KAJIAN PUSTAKA DAN DASAR TEORI

   Berisi teori tentang kajian teori yang berhubungan dalam penelitian skrispsi ini serta terori dasar tentang kanker, kanker payudara, gen TP53, *Sequential Patern Mining*, *Prefixspan* dan teori-teori yang berhubungan dalam skripsi ini.

3. BAB III METODOLOGI DAN PERANCANGAN

   Berisi algoritma-algoritma yang digunakan dalam pembuatan sistem pencarian motif sekuens DNA dengan menggunakan algoritma *prefixspan*.

4. BAB IV IMPLEMENTASI

   Berisi tentang penjelasan implementasi dari sistem, bagaimana user interface sistem dan *source code* untuk mengembankan sistem

5. BAB V PENGUJIAN DAN ANALISIS

   Berisi tentang penjelasn proses pengujian dan hasil pengujian serta analisis dari pengujian tersebut.

6. BAB VI PENUTUP

   Berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran untuk pengembangan.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Berdasarkan judul skripsi yang dibahas, penulis menemukan beberapa hasil penelitian yang relevan untuk mendukung penelitian dalam skripsi ini, antara lain: Riza Ramadan (ITB : 2007), dalam tugas akhirnya yang berjudul Strategi Implementasi Peningkatan waktu Proses Algoritma *PrefixSpan* untuk *Sequential Pattern Mining* menjelaskan algoritma *PrefixSpan* sebagai algoritma yang baik dalam melakukan pencarian pola sekuensial. Namun dikarenakan waktu proses yang dibutuhkan untuk menyelesaikan algoritma ini tidak terlalu baik maka dilakukan perubahan dengan mengganti algoritma rekursif yang sebelumnya digunakan pada pengaplikasian algoritma *PrefixSpan* menjadi algoritma iterasi.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl (2001), dalam jurnal internasional yang berjudul *prefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth* menjelaskan pengembangan proses algoritma prefixspan untuk mempercepat waktu proses dengan menerapkan model *bi-level-projection* dan *pseudo-projection*. Dalam penelitian tersebut juga dibanding waktu proses dari algoritma sequential pattern yang lain yaitu GSP, Freespan serta modifikasi algoritma prefixspan. Dan didapatkan kesimpulan bahwa dengan menerapakan *bi-level-projection* dan *pseudo-projection* dapat meningkatkan efisiensi proses penggalian data.

Berdasarkan jurnal tersebut maka kemudian penelitian dalam skripsi ini dilakuan. Perbedaannya adalah jika kedua jurnal tersebut menjelaskan penelitian tentang bagaimana meningkatkan algoritma *PrefixSpan* maka dalam skripsi ini adalah bagaimana mengaplikasikan algoritma prefixspan dengan performance terbaik untuk menyelesaikan kasus pencarian sekuens DNA penyebab kanker payudara.

2.2 Kanker

Kanker merupakan suatu kondisi yang dihasilkan dari sel yang mengalami pertumbuhan tidak terkendali secara abnormal. Perkembangannya kompleks melalui beberapa tahap yaitu: aktivasi, inisiasi, promotor, progresi (perkembangan

dan penyebaran), dan kemungkinan remisi (sukses pengobatan atau pembalikan). Menurut Krinke (2005), tahap transformasi sel normal menjadi sel kanker adalah sebagai berikut:

1. Aktivasi. Beberapa bahan kimia dan/atau radiasi dapat memicu perubahan sel. Dalam proses yang normal, tubuh seseorang dapat menghilangkan zat-zat berbahaya, dalam beberapa kasus substansi menetap dan menempel pada DNA dalam sel.
2. Inisiasi. DNA berubah atau bermutasi dalam sel yang disalin. Jika itu terjadi dalam DNA tertentu, ini akan membuat sel lebih sensitif terhadap zat berbahaya dan/atau radiasi.
3. Promosi. Ketika sel menjadi sensitif, promotor mendorong sel-sel membelah dengan cepat. Jika urutan normal dari DNA rusak, gumpalan sel abnormal mengikat bersama untuk membentuk suatu masa atau tumor.
4. Progresi. Sel-sel terus berkembang biak dan menyebar ke jaringan terdekat. Jika mereka memasuki sistem getah bening, sel-sel abnormal akan diangkut ke organ tubuh lain.
5. Pembalikan. Tujuan dari pembalikan adalah untuk mencegah perkembangan kanker atau untuk memblokir salah satu dari keempat tahap sebelumnya.

2.3 Kanker Payudara

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 [PAR-07].

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* [PON-03].

2.4 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, 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.

Sejak ditemukannya gen supresor tumor *p53* telah ditemukan untuk bermutasi pada lebih dari 50% kanker pada manusia, telah menarik minat banyak peneliti. Kapasitas *p53* untuk beberapa fungsi biologis dapat dikaitkan dengan kemampuannya untuk bertindak sebagai urutan faktor transkripsi yang spesifik untuk mengatur ekspresi dari lebih dari seratus yang berbeda target, dan dengan demikian untuk memodulasi berbagai proses seluler termasuk apoptosis, sel penangkapan siklus dan perbaikan DNA. Protein *p53* dengan struktur unik C-dan N-terminal adalah kaku dimodulasi oleh beberapa proses biologis penting seperti fosforilasi,asetilasi dan ubiquitination, melalui yang efektif mengatur sel pertumbuhan dan kematian sel. mutasi *p53* dapat menyebabkan baik untuk kehilangan atau perubahan mengikat *p53* kegiatan untuk target hilir dan dengan

demikian dapat menyebabkan proliferasi sel menyimpang, dengan konsekuen transformasi seluler ganas. Berdasarkan peran *p53* penting dalam karsinogenesis, para ilmuwan telah mengembangkan beberapa strategi yang efektif untuk mengobati kanker dengan meningkatkan fungsi stabilitas *p5*3 [LIN-06].

2.5 *Genome Sequencing*

Pengurutan DNA merupakan suatu proses penentuan urutan pasti tiga juta basa nukleotida, yang terdiri dari adenin, guanin, sitosin dan timin (A, T, G, C) dalam suatu molekul DNA. Sedangkan *Sequencing* genom adalah penentuan urutan nukleotida DNA atau basa dalam genom dalam tubuh suatu organisme. Hasil sekuen berupa urutan huruf yang menyatakan basa nukleotdia dalam sustu DNA tertentu, contoh AGTCCGCAGGCTCGGT.

Sequencing genome selalu dibandingkan dengan proses pengkodean, padahal proses *sequencing* lebih dari sekedar hanya pengartian suatu kode. Secara sederhana, dapat dianalogikan sekuen genom berupa susunan huruf dari suatu bahasa misterius yang sebenarnya memiliki suatu makna yang penting dan spesifik. Jadi setelah sekuen genom diperoleh, tidak langsung dapat memberikan informasi genetik dalam suatu spesies. Para ilmuwan masih harus menterjemahkan hasil *Sequencing* untuk memahami bagaimana genom bekerja, apa gen yang menyusun genom tersebut, bagaimana gen yang berbeda dapat berhubungan dan berkoordinasi. *Sequencing* genom saat ini merupakan proses yang sangat penting dan di butuhkan dalam penelitian sains dna teknologi terutama di bidang kesehatan dan bioteknologi.

Adanya sedikit mutasi dari sekuen DNA bisa menyebabkan kelainan gen yang menjadikan penyakit pada seseorang. Sehingga dalam bidang kesehatan, salah satunya *sequencing* genom diterapkan dalam diagnosa penyakit kanker. Suatu kanker disebabkan oleh ketidaknormalan susunan basa DNA yang terdapat dalam sel tubuh tersebut. *Cancer Genome Project* menggunakan sekuen genom manusia dan teknik deteksi mutasi untuk mengidentifikasi sekuen yang mengalami mutasi dan mengidentifikasi gen tertentu yang dapat mempercepat perkembangan sel kanker. Perolehan data sekuen DNA ini dapat menginduksi perkembangan penemuan algoritma terapi untuk penyakit kanker.

2.5 Data Mining

Data mining merupakan suatu metoda pengalian data untuk mendapatkan informasi yang tersembunyi. Berbagai ragam tentang pendefinisian data mining [TAN-06] :

- Penguraian (yang tidak sederhana) dari sekumpulan data menjadi informasi yang memiliki potensi secara implisit (tidak nyata/jelas) yang sebelumnya tidak diketahui.
- Penggalian dan analisis, dengan menggunakan peranti otomatis atau semi otomatis, dari sejumlah besar data yang bertujuan untuk menemukan pola yang memiliki arti.
- Data mining juga merupakan bagian dari knowledge discovery dalam database (KDD)

Secara garis besar terdapat dua algoritma dalam data mining untuk melaksanakan perannya, meliputi [TAN-06] :

- Algoritma Prediksi

  Menggunakan beberapa variabel untuk memprediksi nilai yang tidak diketahui atau nilai di masa mendatang dari variabel lain.
- Algoritma Deskripsi

  Menemukan bentuk yang mampu diartikan manusia (*human-interpretable patterns)* yang dapat menjelaskan data tertentu.

2.6 Sequential Patern Mining

Sequential pattern mining merupakan proses data mining yang menghasilkan pengetahuan mengenai serangkaian kejadian-kejadian yang memiliki frekuensi kemunculan yang melebihi nilai threshold [PEI-01]. Sequential pattern adalah pola turunan dari assocation rules, karena sama-sama menunjukkan keterhubungan antara kejadian. Bedanya adalah bahwasanya sequential pattern menitikberatkan pada pencarian pola kejadian yang muncul setelah suatu kejadianm sedangkan association rules adalah pola kejadian yang muncul bersamaan dengan kejadian lain.

Sudah banyak algoritma-algoritma yang diciptakan khusus untuk melakukan sequential pattern mining ini, antara lain GSP (generalized Sequential Pattern), SPADE (Sequential Pattern Discovery using Equivalent classes) dan PrefixSpan. GSP adalah algoritma mining yang memakai pendekatan candidate-and-test, yaitu dengan membangkitkan pola-pola yang kemudian dihitung jumlah kemunculannya. Apabila melebihi nilai threshold, pola tersebut menjadi sequential pattern, dan bila tidak, pembangkitan pola-pola yang merupakan supersequence dari pola tersebut tidak akan pernah dilakukan. Hal ini dimaksudkan untuk semakin membatasi kandidat [RAM-07].

Mirip dengan pendekatan GSP, SPADE (Sequential Pattern Discovery using Equivalent classes) memakai pendekatan berbasis candidate-and-test, sehingga algoritma ini perlu melakukan sekali scan terhadap basis data untuk menemukan pila dengan satu elemen yang sering muncul. Untuk membangkitkan kandidat pola dengan dua elemen, dilakukan dengan cara menggabungkan seluruh pola satu elemen apabila pola tersebut melebihi nilai threshold dan memiliki identitas sequence yang sama. Hal ini dilakukan terus menerus hingga tidak ada lagi pembangkitan kandidat pola, yang berarti seluruh sequential pattern telah ditemukan [RAM-07].

2.7 PrefixSpan

Prefix-Projected Sequential Pattern Growth (PrefixSpan) adalah algoritma yang memakai pendekatan pengembangan sequence untuk mencari sequential pattern. PrefixSpan akan mencari frequent sequence satu elemen dan kemudian mengembangkan sequence-sequence tersebut dengan cara menambahkan elemen satu persatu. PrefixSpan dirancang sedemikian rupa sehingga sequnce hasil penambahan elemen tersebut tetap merupakan frequent sequence. Dengan cara ini tidak diperlukan pembangkitan dan pengujian kandidat [RAM-07].

2.7.1 Prefix

Jika terdapat sequence A= {a1, a2, ..., an}, sequence B = {a'1, a'2, ..., a'm} (m<=n) disebut prefix dari A jika dan hanya jika [PEI-01]:

1. a1 = a'1
2. a'm adalah himpunan bagian dari an
3. semua item di dalam (am-a'm) secara alphabetik muncul setelah a'm

2.7.2 Projected Database

Diberikan sequence A dan B sedemikian rupa sehingga B adaalah subsequence dari A. Suatu sequence A' yang merupakan subsequence A dapat dikatakan hasil proyeksi dengan prefix B jika dan hanya jika [PEI-01] :

1. A' memiliki prefix B.
2. Tidak ada supersequence A'' dari A' sedemikian rupa sehingga A'' merupakan sub-sequence dari A yang juga memiliki prefix B

2.7.3 Konsep Algoritma PrefixSpan

Berdasarkan deskripsi yang telah dijelaskan sebelumnya , konsep algoritma PrefixSpan adalah sebagai berikut [PEI-01].

Input :

Sequential pattern a, Sequence database S, panjang Sequence I, nilai minimum support min_sup

Output : himpunan sequential pattern

Pemanggilan Method : PrefixSpan (< >, 0, S)

Algoritma :

1. Melakukan scanning terhadap S sekali, untuk mendapatkan 1-length frequent item.
2. Untuk setiap frequent item b, tambahkan ke dalam akhir a atau gabungkan ke dalam a untuk mendapatkan Sequential Pattern a'. Kemuadian output-kan a'.
3. Untuk setiap a', bangun projected database S' yang diproyeksikan berdasarkan b. Kemudian panggil PrefixSpan (a',1+1,S')

2.8 Confidence

Confidence adalah ukuran yang menunjukkan hubungan antar 2 item secara kondisional [HAN-06]. Nilai *confidence* keandalan dari rule yang dibuat. Rumus yang digunakan untuk menentukan *confidence* adalah :

$$Confidence, C(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(A)} \quad (2.1)$$

Dimana :

- $\sigma(A \cup B)$ = Jumlah itemset di semua transaksi
- $\sigma(A)$ = Jumlah *antecedent* pada transaksi

2.8 Lift Rasio

Lift rasio digunakan untuk mengukur seberapa kuat rule yang dibentuk dari algoritma sequential patern mining. Nilai *lift* rasio berkisar antara 0 sampai dengan tak terhingga. Nilai minimum dari *lift* rasio tidak ditentukan seperti halnya *support* atau *confidence*. Jika nilai *lift* rasio kurang dari 1 dalam hal ini adalah nilai minimum maka *rule antecedent* berpengaruh negatif pada *rule consequent*. Jika nilai *lift* rasio sama dengan 1 maka rule tersebut sering muncul bersamaan tetapi independen. Rule yang independen merupakan rule dimana untuk mendapatkan *consequent* tidak tergantung pada *antecedent*. Pada *lift* rasio, rule yang direkomendasikan adalah jika *lift* rasio lebih dari 1 karena *antecedent* memiliki pengaruh positif pada *consequent*. Berikut rumus untuk menentukan *lift* rasio [FOM-11]:

$$Expected\ Confidence, EC\ (A \rightarrow B) = \frac{\sigma(B)}{m} \quad (2.2)$$

$$Lift = \frac{Confidence}{Expected\ Confidence} \quad (2.3)$$

Dimana:

- $\sigma(B)$= Jumlah *consequent* dalam transaksi
- m = Jumlah transaksi

2.9 Support

Support adalah ukuran dari seberapa sering koleksi item dalam asosiasi terjadi bersama sebagai persentase dari seluruh transaki. Nilai support sebuah item diperoleh dengan menggunkan rumus berikut [GOL-12]:

$$Support(A) = \frac{Jumlah\ Transaksi\ mengandung\ A}{Total\ Transaksi} \quad (2.4)$$

Nilai support dari 2 item diperoleh dengan menggunakan rumus :

$$Support(A, B) = P(A \cap B)$$

$$Support\ (A, B) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi} \quad (2.5)$$

Support digunakan dalam algoritma *AprioriAll* sebagai ukuran evaluasi dari algoritma tersebut.

BAB III
METODOLOGI DAN PERANCANGAN

3.1 Metodologi

Pada bab ini akan dibahas algoritma dan langkah-langkah yang digunakan untuk melakukan pencarian pola sekuens DNA gen *p53* untuk diagnosa dini penyakit kanker payudara dengan algoritma *prefixspan*.

Penelitian dilakukan dengan langkah-langkah sebagai berikut:

1. Melakukan studi literatur yang berkaitan dengan data mining, *sequential pattern mining*, algoritma prefixspan, dan DNA gen *p53* penyebab kanker.
2. Merancang sistem untuk sequential pattern mining pada data gen DNA.
3. Implementasi sistem berdasarkan analisis dan perancangan yang dilakukan.
4. Melakukan pengujian terhadap sistem.
5. Melakukan evaluasi tingkat keberhasilan sistem dan analisis hasil pengujian.

3.2 Analisis Sistem

3.2.1 Analisis Data

Data yang digunakan dalam skripsi ini didapatkan dari bank database DNA penderita kanker payudara yang disediakan di internet. Data diunduh dari situs www.ncbi.nlm.nih.gov. Data yang diambil adalah data gen *p53* pada manusia penderita kanker payudara. Berikut ini adalah salah satu contoh data yang diambil dari situs tersebut.

```
1 tccccttgc cgtcccaagc aatggatgat ttgatgctgt ccccggacga tattgaacaa
61 tggttcactg aagacccagg tccagatgaa gctcccagaa tgccagaggc tgctccccgc
121 gtggcccctg caccagcagc tcctacaccg gcggcccctg caccagcccc ctcctggccc
181 ctgtcatctt ctgtcccttc ccagaaaacc taccagggca gctagggttt ccgtctgggc
241 ttcttgcatt ctgggacagc caagtctgtg acttgcacg
```

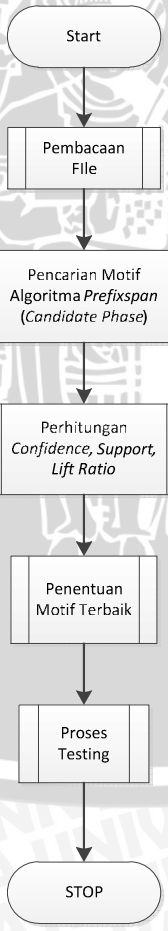
Gambar 3.1 Sekuens DNA gen *p53*

Data tersebut adalah data dari jenis spesies manusia (*homo sapiens*) yang diketahui menderita kanker payudara, bukan dari jenis hewan atau spesies lain. Sekuens gen yang diambil adalah sekuens dalam satu exon. Exon adalah urutan

sekuens yang merupakan bagian dari keseluruhan sekuens DNA. Dan data yang digunakan dalam penelitian ini adalah data gen pada exon ke-7.

3.2.2 Analisis Proses

Pada bagian ini akan dijelaskan rancangan proses atau gambaran umum sistem bagaimana nantinya sistem akan bekerja menggunakan algoritma *prefixspan*. Sistem yang akan dibangun tidak menggunkan database untuk menyimpan datanya, tetapi data disimpan dalam file berformat .txt. Sehingga proses awal dari sistem ini adalah pembacaan file berformat .txt tersebut. Dan data itulah nantinya yang akan dioalah oleh sistem menggunakan algoritma *prefixspan* untuk menghasilkan motif sekuens DNA penderita kanker payudara. Gambar 3.1 akan menjelaskan lebih jelas gambaran umum alur proses yang akan dijalankan oleh sistem yang dimulai dengan proses pembacaan *file*.

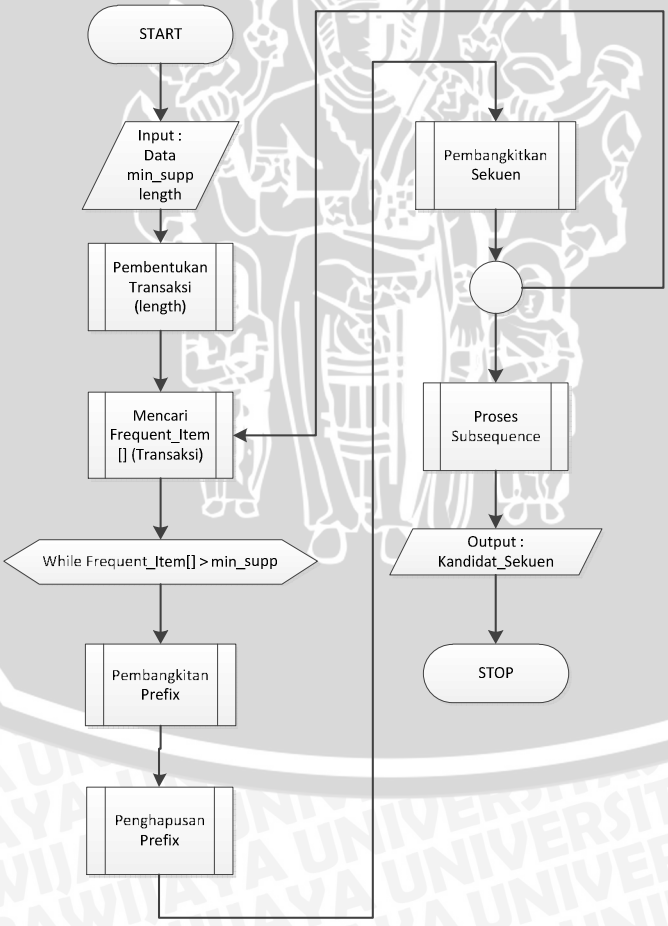


Gambar 3.1 Flowchart Sistem

Gambaran umum proses-proses utama yang akan dilakukan oleh sistem dalam penelitian ini dijelaskan pada Gambar 3.1. Pada dasarnya proses pencarian berakhir hingga ditemukannya suatu motif tetapi untuk menguji suatu motif tersebut maka diperlukan suatu testing. Detail dari masing-masing method dari Gambar 3.1 akan dijelaskan lebih jelas pada sub bab ini mulai dari proses algoritma prefixspan hingga proses testing.

3.2.2.1 Pencarian Motif dengan Algoritma Prefixspan

Proses ini adalah proses utama yang dijalankan oleh sistem untuk membentuk sekuen motif dari DNA penderita kanker payudara. Algoritma prefixspan yang digunakan dalam penelitian ini berjalan dengan cara *iterasi* bukan *rekursif*. Gambar 3.2 akan mengambarkan jalannya proses pencarian ini seperti berikut.



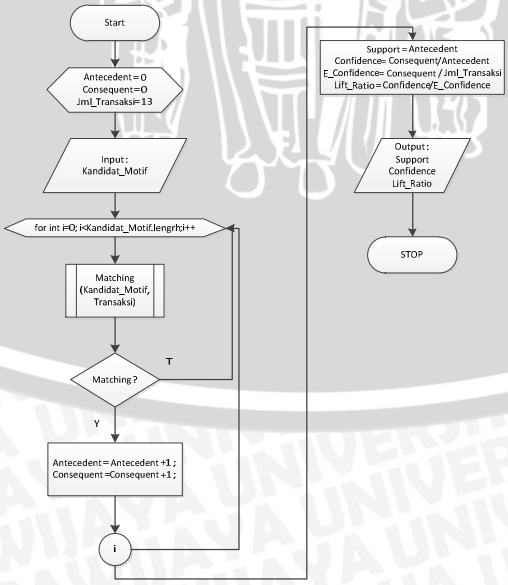
Gambar 3.2 Flowchart Algoritma *Prefixspan*

Penjelasan dari Gambar 3.2 tersebut adalah sebagai berikut,

1. Input yang diberikan berupa Data sekuens DNA penderita kanker payudara, *minimum support* dan length yang ditentukan.
2. Proses selanjutnya adalah membentuk transaksi dan mencari frekuensi item dari setiap item (A, T, G, C). Ketika jumlah frekuensi dari setiap item melebihi *minimum support* maka proses dilanjutkan dengan melakukan pembangkitan dan penghapusan *prefix* untuk dibangkitkannya suatu pola. Pembentukan pola berhenti ketika jumlah frekuensi item kurang dari *minimum support*. Dan pembentukan pola diakhiri dengan proses *subsequence* yaitu sekuens yang merupakan bagian dari sekuens lain akan dihapuskan.

3.2.2.2 Perhitungan Support, Confidence, dan Lift Ratio

Dalam menguji kekuatan dari motif yang dihasilkan maka terdapat beberapa indikator yang digunakan sebagai acuan penilaian. Indikator tersebut yaitu support, *confidence* dan lift rasio. Nilai dari indikator ini juga menentukan motif akhir sebagai hasil dari sistem. Adapun algoritma atau proses dalam melakukan penentuan nilai indikator-indikator ini dapat dilihat pada Gambar 3.3. Pada diagram ini akan dijelaskan rumusan untuk membentuk nilai ketiga indikator tersebut berdasarkan kandidat motif yang dihasilkan dari proses sebelumnya.



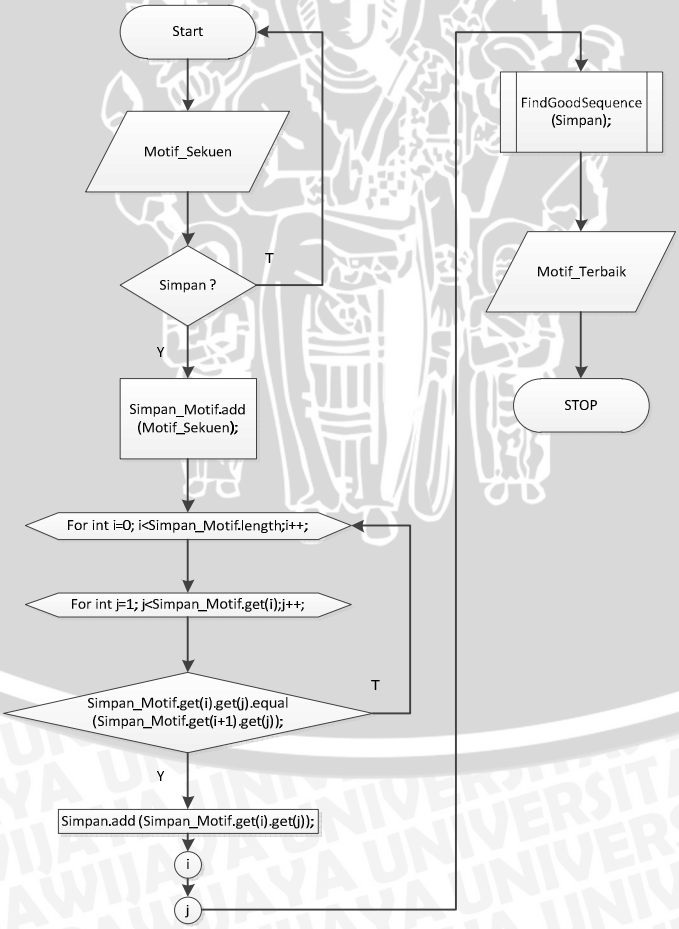
Gambar 3.3 Flowchart Perhitungan Support, Confidence, Lift Rasio

Penjelasan dari Gambar 3.3. diatas adalah sebagai berikut :

1. Input berupa Kandidat motif yang dihasilkan dari proses sebelumnya
2. Dilakukan proses string matching dari kandidat motid yang dihasilkan terhadap transaksi yang terjadi untuk didapatkan nilai *antecedent* dan *consequent*. Dari kedua nilai ini maka ditentukan nilai dari support, *confidence* dan lift ratio

3.2.2.3 Penentuan Motif Terbaik

Proses ini dilakukan sebelum proses *testing* dilakukan setelah didapatkan motif yang dianggap terbaik. Motif-motif tersebut disatukan untuk didapatkan yang paling baik. Dan dari motif yang terbaik inilah baru motif ini diujikan. Dengan anggapan bahwa jika motif terbaik terbukti maka motif lain dari hasil algoritma *prefixspan* pun bisa terbukti. Jalannya proses atau bagaimana proses ini terjadi akan dijelaskan pada Gambar 3.4 seperti berikut.



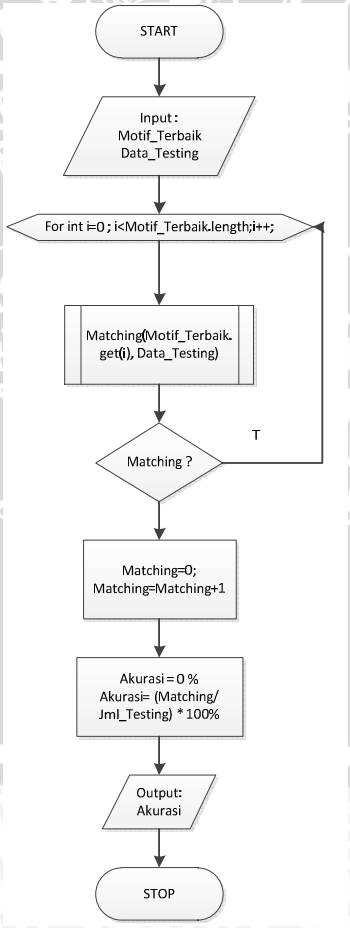
Gambar 3.4 Flowchart Penentuan Motif Terbaik

Penjelasan dari Gambar 3.4 adalah sebagai berikut :

1. Input berupa motif sekuen yang dihasilkan dari proses sebelumnya
2. Motif-motif sekuens dari beberapa pengujian dikumpulkan untuk diambil motif yang terbanyak muncul dengan melakukan “equal” dari tiap-tiap motif yang diakhiri dengan penentuan motif yang tersering muncul atau terbanyak muncul sebagai motif terbaik.

3.2.2.4 Proses Testing

Proses *testing* adalah proses pengujian dari motif terbaik yang dihasilkan terhadap sekuens DNA penderita kanker payudara lain. Untuk membuktikan apakah motif tersebut terbukti terdapat pada penderita kanker payudara ataukah tidak. Proses ini diakhiri dengan menghasilkan prosentase akurasi dari setiap motif yang dihasilkan. Bagaiman proses ini berlangsung dapat dilihat pada Gambar 3.5 berikut.



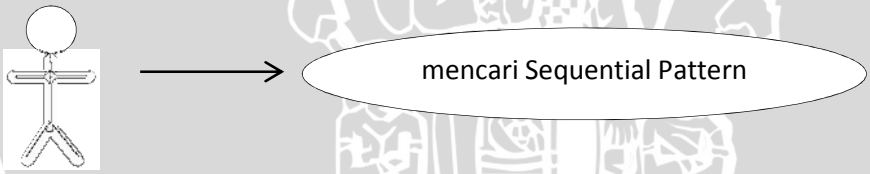
Gambar 3.5 Flowchart Proses Testing

Penjelasan dari Gambar 3.5 adalah sebagai berikut :

1. Input berupa motif terbaik yang dihasilkan dari proses sebelumnya serta data testing yang digunakan dalam proses pengujian.
2. Dilakukan proses *string matching* dari setiap motif yang dihasilkan, dimana ketika ditemukannya motif yang sesuai maka diberikan nilai 1 yang bersifat akmulasi. Dari nilai inilah dibanding dengan banyaknya *testing* yang dilakukan maka dihasilkan akurasi dari setiap motif yang dihasilkan.

3.3 Use Case Sistem

Implementasi algoritma *prefixspan* dalam sistem ini untuk dapat berjalan sebagaimana seharusnya, terdapat dua hal yang harus dilakukan terlebih dahulu yaitu mengatur nilai *minimum support* yang diinginkan dan menentukan *length* dari sekuens awal. Kemudian sistem memproses data untuk menemukan pola sekuens dengan algoritma *prefixspan*. Dengan cara kerja seperti ini, maka dapat dijelaskan diagram use case sistem seperti pada Gambar 3.6.



Gambar 3.6 Use Case Sistem

Aktor yang terlibat dalam sistem ini adalah user atau orang yang akan menggunakan sistem ini. Sebagaimana yang dapat diketahui dari uses case bahwa yang bisa dilakukan oleh aktor adalah mencari *sequential pattern*. Dan skenario dari use case ini adalah sebagai berikut.

1. Aktor menginputkan data file yang merupakan *sequence* database. Kemudian menentukan nilai *minimum support* yang akan digunakan sistem sebagai dasar untuk pemrosesan serta menentukan *length* untuk pembacaan dan pengelompokan data.

2. Sistem memproses dengan algoritma *prefixspan* terhadap *minimum support* dan *length* yang telah ditentukan untuk mendapatkan pola sekuens dari data yang diinputkan.

3.4 Perhitungan Manual

Dalam menjelaskan proses pencarian pola sekuens DNA dengan menggunakan algoritma *prefixspan*, maka diberikan contoh perhitungan manual untuk membarikan gambaran awal bagaimana sistem nantinya bekerja melakukan pencarian pola sekuens serta dapat digunakan sebagai verifikasi hasil akhir dari sistem untuk memastikan apakah sistem bekerja dengan benar. Berikut ini contoh perhitungan manual dengan menggunakan satu sekuens database DNA penderita kanker payudara yang didapatkan dari website www.ncbi.nlm.nih.gov.

Berikut ini sekuens database DNA yang dipilih untuk dijadikan contoh dalam perhitungan manual ini.

gttggctctgactgtaccaccatccactacaactacatgtgtaacagttcctgcatgggc
ggcatgaaccggaggcccatcctcaccatcatcacactggaagactccag

Gambar 3.7 Sekuens DNA sebagai Data Awal

Dari sekuens data tersebut kemudian dikelompokan sepanjang *length* yang ditentukan, dalam contoh ini ditentukan *length* sebesar 10 dan diinisialisasikan nilai *threshold* atau *minimun support* sebesar 2. Berikut ini hasil pengelompokan yang didapatkan.

Tabel 3.1 Pengelompokan Sekuens Database dengan length 10

ID	Sekuens
1	gttggctctg
2	actgtaccac
3	catccactac
4	aactacatgt
5	gtaacagttc
6	ctgcatgggc
7	ggcatgaacc
8	ggaggcccat

9	cctcaccatc
10	atcacactgg
11	aagactccag

Kemudian dihitung *frequent* dari elemen-elemen yang sering muncul dalam kasus ini terdapat 4 elemen dan berikut hasil perhitungan *frequent* yang didapatkan dengan analogi bobot.

Tabel 3.2 Frekuensi Elemen yang sering muncul

Elemen	Bobot
a	10
t	11
g	9
c	11

Langkah selanjutnya adalah pembentukan *prefix*, prefix awal yang digunakan adalah a, t, g, dan c. Perlakuan yang diberikan terhadap data adalah menghilangkan elemen sebelum *prefix* pertama yang ditemukan dan tidak mengunakan kelompok data yang tidak menggunakan *prefix* yang ditentukan. Dan berikut hasil dari proses tersebut.

Prefix :

Tabel 3.3 Pefix a

ID	Sekuens
1	
2	ctgtaccac
3	tccactac
4	actacatgt
5	acagttc
6	tgggc
7	tgaacc
8	ggcccat
9	ccatc
10	tcacactgg
11	agactccag

Tabel 3.4 Pefix t

ID	Sekuens
1	tggctctg
2	gtaccac
3	ccactac
4	acatgt
5	aacagttc
6	gcatgggc
7	gaacc
8	
9	caccatc
10	cacactgg
11	ccag

Tabel 3.5 Pefix g

ID	Sekuens
1	ttggctctg
2	taccac
3	catccactac
4	t
5	taacagttc
6	catgggc
7	gcatgaacc
8	gaggcccat
9	cctcaccatc
10	g
11	gactccag

Tabel 3.6 Pefix c

ID	Sekuens
1	Tctg
2	ctgtaccac
3	atccactac
4	tacatgt
5	agttc
6	tgcatgggc
7	atgaacc
8	ccat
9	ctcaccatc
10	acactgg
11	tccag

Dalam contoh kasus ini hanya dilakukan pencarian pola untuk *prefix* a dan sekuens kelanjutan sekuens <a> saja yang mewakili *prefix* lain. Proses seperti ini terus dilakukan hingga frekuen yang didapatkan tidak ada lagi yang melebihi *minimum support*.

Tabel 3.7 Frekuensi Elemen prefix a

<a>	Bobot
a	7
t	7
g	4
c	8

Dari proses awal ini pola sekuens yang didapatkan adalah <aa>, <at>, <ag>, <ac>. Kemudian dilakukan kembali proses tersebut dan masih hanya untuk *prefix* a.

Tabel 3.8 Hasil proses Pefix a untuk sekuens <aa>

ID	Sekuens
1	
2	ccac
3	ctac
4	ctacatgt
5	cagttc
6	
7	tcc
8	t
9	tc

10	cactgg
11	gactccag

Tabel 3.9 Frekuensi Elemen prefix a untuk sekuens <aa>

<aaa>	Bobot
a	2
t	6
g	4
c	8

Sekuens hasil: <aaa> <aat> <aag> <aac>

Tabel 3.10 Hasil proses Pefix a untuk sekuens <aaa>

<aa>	Bobot
a	9
t	11
g	9
c	11

Tabel 3.11 Frekuensi Elemen prefix a untuk sekuens <aaa>

ID	**Sekuens**
1	
2	c
3	c
4	catgt
5	gttc
6	
7	cc
8	t
9	tc
10	ctgg
11	ctccag

Sekuens hasil: <aaaa> <aaat> <aaag> <aaac>

Tabel 3.12 Hasil proses Pefix a untuk sekuens <aaaa>

ID	Sekuens
1	
2	
3	
4	tgt
5	
6	
7	
8	
9	
10	
11	ag

Tabel 3.13 Frekuensi Elemen prefix a untuk sekuens <aaaa>

<aaaa>	Bobot
a	1
t	1
g	2
c	0

Sekuens hasil: <aaaag>

Proses berhenti disini untuk *prefix* a. Hal ini dikarenakan jika proses ini dilanjutkan tidak akan ada lagi frekuensi elemen yang nilainya lebih dari *minimmum support*. Namun sistem tidak hanya akan mencari sekuens dari *prefix* a saja namun juga *prefix* t, g dan c. Setelah semua sekuens dari setiap *prefix* didapatkan maka langkah selanjutnya adalah melakukan proses pengecekan *subsequence*. Dimana suatu sekuens yang merupakan *subsequence* dari sekuens yang lain maka sekuens tersebut tidak dipilih sebagai hasil pencarian pola oleh sistem. Dan berikut ilustrasinya namun hanya untuk sekuens pada *prefix* a.

Hasil Sekuens Prefix a : <aaaag> <aaaa> <aaat> <aaag> <aaac> <aaa> <aat> <aag> <aac> <aa> <at> <ag> <ac>

Hasil proses subsequnce : <aaaag> <aaat> <aaag> <aaac>

Hasil ini diperoleh karena sekuens <aaaa> adalah subsequence (bagian dari) <aaaag>, sedangkan sekuens <aaa> <aat> <aag> <aac> <aa> <at> <ag> <ac> adalah subsequence (bagian dari) sekuens <aaaag> <aaat> <aaag> <aaac>.

Proses selanjutnya adalah perhitungan lift rasio dari suatu setiap sekuens jika lift rasio dari suatu sekuens bernilai lebih dari 1 maka sekuens tersebut diambil dan digunakan sebagai hasil output proses pencarian pola. Hal ini karena jika nilai lift rasio lebih dari 1 maka rule tersebut sering muncul bersamaan dan independen.

3.5 Perancangan Uji Coba

Perancangan uji coba digunakan untuk mengambarkan skenario proses pengujian yang akan dilakukan pada sistem. Proses uji coba dilakukan dengan 20 data sekuens DNA dari obyek manusia yang berbeda namun pada exon yang sama yaitu Exon ke-7. *Minimum support* yang digunakan dalam pengujian ini nantinya tidak boleh melebihi batas maksimal dari transaksi yang terjadi karena jika melebihi jumlah transaki maka tentunya tidak akan ada pola yang dihasilkan. Sedangkan *length* yang digunakan untuk membentuk transaksi ditetapkan sama yaitu 10. Berikut ini tabel rancangan proses pengujian yang akan dilakukan.

Tabel 3.14 Rancangan Uji Coba Sistem

Individu ke-	Minimum Support	Length	Motif Sekuen	Lift Ratio
1	2	10		
	4	10		
	6	10		
	8	10		
	10	10		

3.6 Proses penentuan Tingkat Akurasi

Tingkat akurasi pola sekuens hasil algoritma prefixspan didapatkan melalui algoritma pengembangan *String Matching*. Algoritma ini adalah pengembangan dari algoritma *String Matching* dimana pengembangannya berupa, pola *string* yang dianggap sama bukan ketika pola string benar-benar sama persis

tapi juga ketika pola yanga tidak berdampingan diketahui sama maka pola dianggap sama. Dalam penelitian ini proses *string matching* dilakukan dengan membandingkan hasil output dengan setiap transaksi sekuens DNA. Ilustrasi dari algoritma ini dapat dijelaskan seperti contoh berikut.

Sekuens ID 1 : gttggctctg

Pattern yang dihasilkan : attca

Hasil : Not Matching

Sekuens ID 2 : actgtaccac

Pattern yang dihasilkan : attca

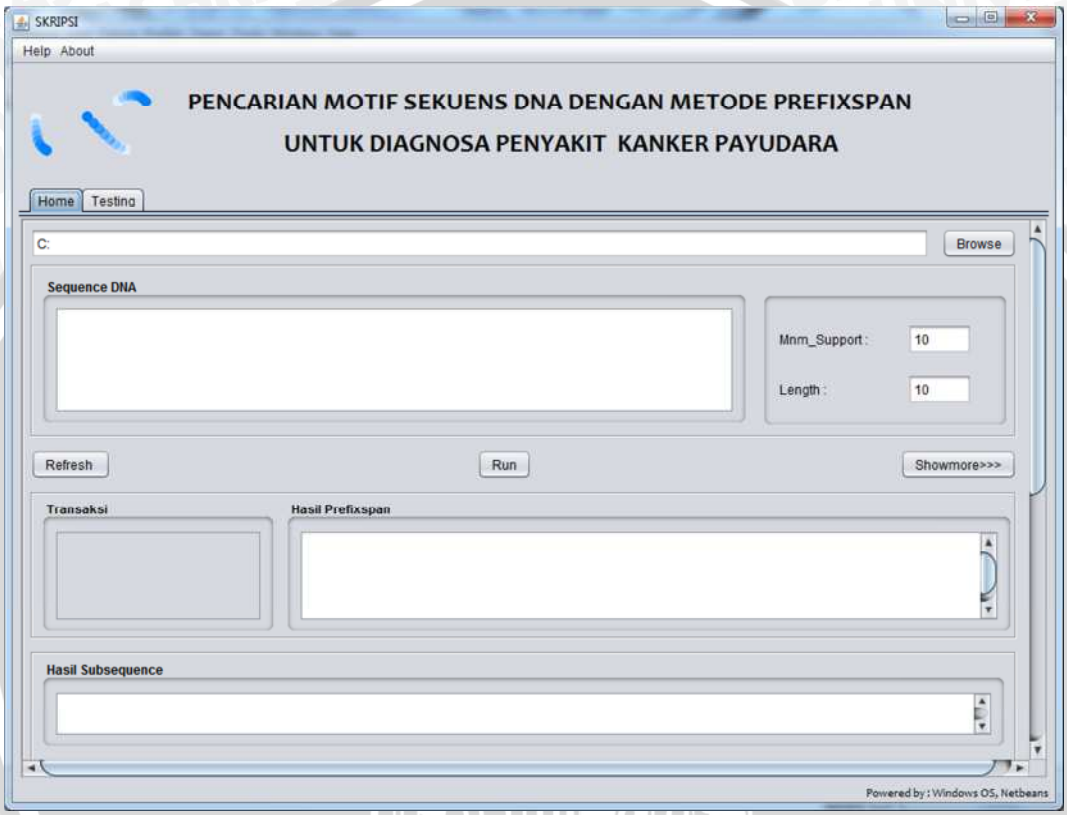
ac**tgtacca**c

Hasil : Matching

Ketika ditemukan kecocokan maka motif sekuens yang didapatkan dianggap sesuai. Jumlah kebenaran motif dibandingkan dengan jumlah salah dari seluruh motif dari beberapa uji coba dapat digunakan sebagai indikator penentuan nilai akurasi pencarian dengan algoritma *prefixspan*.

3.7 Rancangan Antarmuka

Antarmuka yang natinya akan dibuat diharapkan bersifat *user friendly* dimana pengguna akan mudah dalam menjalanka aplikasi. Dengan *one klick* diharapkan sistem telah bekerja untuk menjalankan proses yang dibutuhkan. Gambar 3. 3 akan menggambarkan rancangan antarmuka yang akan dibuat.



Gambar 3.8 Rancangan Antarmuka Sistem

BAB IV
IMPLEMENTASI

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalama sub bab ini adalah lingkungan implementasi perangkat lunak dan perangkat keras yang digunakan dalam mengimlementasikan sistem yang telah dibuat dalam penelitian ini.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam mengembangkan perangkat lunak dalam penelitian ini memiliki spesifikasi sebagai berikut :

1. Intel(R) Core(TM) i3 CPU M 350 @2.27GHz(4 CPUs)
2. Memori 3 GB
3. Harddisk 320 GB
4. Monitor 14”

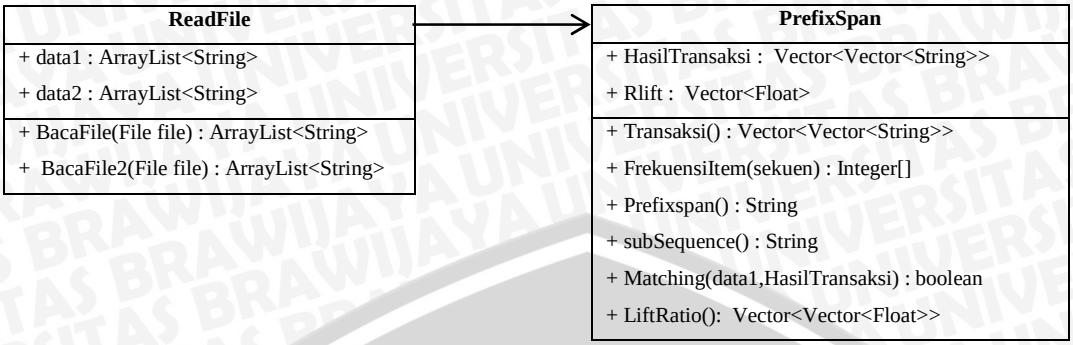
4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam mengembangkan sistem dan penelitian ini terdiri dari :

1. Sistem Operasi Windows 7 Ultimate 32 bit
2. NetBeans IDE 7.1
3. BioEdit
4. Notepad++

4.2 Implementasi Program

Berdasarkan analisa dan perancangan sistem yang terdapat pada bab 3, maka pada subbab ini akan dijelaskan implementasi proses-proses tersebut kedalam sistem dengan menggunakan bahasa pemrograman JAVA. Secara garis besar sistem dibangun dengan 3 class utama dan didalam masing-masing kelas terdapat *method* yang menjalankan setiap fungsi. Menjelaskan hal tersebut maka akan dijelaskan terlebih dahulu gambaran umum implementasi sistem dalam class diagram yang akan ditampilkan pada Gambar 4.1 berikut.



Gambar 4.1 Gambaran Class Diagram Implementasi Sistem

Gambaran umum class diagram yang dibangun pada sistem ditunjukkan pada Gambar 4.1. Namun class diagram ini masih belum menjelaskan keseluruhan method ataupun tipe data yang ada pada sistem. Class diagram tersebut hanya menggambarkan tipe data dan *method-method* terpenting yang ada pada sistem. Detail dan keseluruhan *method* yang ada dapat dilihat dalam implementasi *sourcecode* pada subbab ini. Sedangkan penjelasan *method* utama pada kedua class tersebut dapat dilihat pada Tabel 4.1 berikut.

Tabel 4.1 Penjelasan Fungsi Method Gambar 4.1

Method	Fungsi
BacaFile(File file)	Method ini adalah suatu method yang bertipe public yang menggunakan struktur data ArrayList, berfungsi untuk membaca data yang digunakan untuk proses pembentukan motif
BacaFile2(File file)	Method ini adalah suatu method yang bertipe public yang menggunakan struktur data ArrayList, berfungsi untuk membaca data yang digunakan untuk proses testing.
Transaksi()	Method ini adalah suatu method yang bertipe public yang menggunakan struktur data Vector, berfungsi untuk menjalankan proses pembentukan transaksi.
FrekuensiItem(sekuen)	Method ini adalah suatu method yang bertipe

	public yang menggunakan struktur data Array yang menyimpan data integer, berfungsi untuk menghitung frekuensi item pada transaksi.
Prefixspan()	Method ini adalah suatu method yang bertipe public yang menggunakan tipe data String yang berfungsi untuk menjalankan proses algoritma *prefixspan*.
subSequence()	Method ini adalah suatu method yang bertipe public yang menggunakan tipe data String yang berfungsi untuk menjalankan proses penentuan subsequence.
Matching()	Method ini adalah suatu method yang bertipe public yang menggunakan tipe data boolean yang berfungsi untuk menjalankan proses string Matching.
LiftRatio()	Method ini adalah suatu method yang bertipe public yang menggunakan struktur data Vector yang berfungsi untuk menjalankan proses penentuan LiftRatio

4.2.1 Pembacaan Data (Read File)

Dalam penelitian ini sistem atau aplikasi menerima inputan berupa data text yang disimpan dalam file berformat .txt. File ini berisi sekuens DNA dari seorang penderita kanker payudara. Implementasi fungsi ini kedalam bahasa pemrograman JAVA akan dijelaskan seperti pada Sourcecode 4.1 berikut.

```
public class ReadFile {
    public     static     ArrayList<String>     data     =     new
ArrayList<String>();
    public    ArrayList<String>    BacaFile(File    file)    throws
IOException {
        try {
            BufferedReader    br    =    new    BufferedReader(new
FileReader(file));
            String line;
            while ((line = br.readLine()) != null) {
                data.add(line);
            }
```

```
        } catch (FileNotFoundException ex) {

Logger.getLogger(ReadFile.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return data;
    }
}
```

Sourcecode 4.1 Fungsi Pembacaan Data (Read File)

4.2.2 Pembangkitan Transaksi

Proses awal atau preposesing dari data yang diinputkan sebelum di proses menggunakan algoritma prefixspan adalah proses pembentukan transaksi. Transaksi ini membagi data berdasarkan length yang ditentukan user. Transaksi ini berpengaruh penting pada proses algoritma *prefixspan* karena transaksi ini membagi jumlah data yang digunakan sebagai acuan dalam menjalankan algoritma *prefixspan*. Sourcode 4.2 berikut ini akan menjelaskan bagaimana proses pembangkitan transaksi ini diimplementasikan.

```
public Vector<Vector<String>> Transaksi() {
        int number = 1;
        for (int i = 0; i < input.data.size(); i++) {
            int a = 0;
            int b = length;
            while (b <= input.data.get(i).length()) {
                Vector<String> Simpan_data = new Vector<String>();
                Simpan_data.add(" " + number);
                Simpan_data.add(input.data.get(i).substring(a,
b));
                HasilTransaksi.add(Simpan_data);
                a += (length - 1);
                b += (length - 1);
                number++;
            }
            if (b != input.data.get(i).length()) {
                Vector<String> Simpan_data = new Vector<String>();
                Simpan_data.add(" " + number);
                Simpan_data.add(input.data.get(i).substring(a + 1,
input.data.get(i).length()));
                HasilTransaksi.add(Simpan_data);
            } else {
                break;
            }
        }
        return HasilTransaksi;
    }
```

Sourcecode 4.2 Proses Pembentukan Transaksi

4.2.3 Pencarian Sekuens menggunakan algoritma Prefixspan

Setalah transaksi dari data yang diinputkan terbentuk, maka kemudian data tersebut diproses dengan menggunakan algoritma *prefixspan*. Disinilah proses utama dalam sistem ini dibangun. Proses utama ini diimplementasikan kedalam suatu method yang bertipe string dimana tipe data yang digunakan didalamnya bertipe Vector. Sourcode 4.3 akan menjelaskan detail method tersebut seperti berikut.

```
public String prosesPrefix(String proses, char key) {
        int j = -1;
        boolean sisa = false;
        String hasil = "";
        for (int i = 0; i < proses.length(); i++) {
            if (proses.charAt(i) == key) {
                sisa = true;
                j = i;
                break;
            }
        }
        if (sisa) {
            hasil = proses.substring(j + 1, proses.length());
        } else {
            hasil = "";
        }
        return hasil;
    }
    public                                  Vector<Vector<String>>
removeCharacter(Vector<Vector<String>> Hasillama, Character key) {
        Vector<Vector<String>>        Hasil          =          new
Vector<Vector<String>>();
        for (int i = 0; i < Hasillama.size(); i++) {
            Vector<String> hsl_remove = new Vector<String>();
            hsl_remove.add("" + i);
            hsl_remove.add(prosesPrefix(Hasillama.get(i).get(1),
key));
            Hasil.add(hsl_remove);
        }
        return Hasil;
    }
    public String PrefixSpan() {
        Vector<Vector<String>> HasilA = HasilTransaksi;
        Vector<Vector<String>> HasilT = HasilTransaksi;
        Vector<Vector<String>> HasilG = HasilTransaksi;
        Vector<Vector<String>> HasilC = HasilTransaksi;
        Vector<Node> queueA = new Vector<Node>();
        Vector<Node> queueT = new Vector<Node>();
        Vector<Node> queueG = new Vector<Node>();
        Vector<Node> queueC = new Vector<Node>();
        queueA.add(new Node(HasilA, "A"));
        queueT.add(new Node(HasilT, "T"));
        queueG.add(new Node(HasilG, "G"));
        queueC.add(new Node(HasilC, "C"));
        while   (!queueA.isEmpty()   ||   !queueT.isEmpty()   ||
```

```
!queueG.isEmpty() || !queueC.isEmpty()) {
            if (!queueA.isEmpty()) {
                Node node = queueA.remove(0);
                Vector<Vector<String>>        HasilbaruA        =
removeCharacter(node.getDataBaru(), 'A');
                Integer[] frek = FrekuensiItem(HasilbaruA);
                if (frek[0] >= minSupport) {
                    queueA.add(new                    Node(HasilbaruA,
node.getDataOutput() + "A"));
                    out.add(new                       Node(HasilbaruA,
node.getDataOutput() + "A"));
                }
                if (frek[1] >= minSupport) {
                    queueT.add(new                    Node(HasilbaruA,
node.getDataOutput() + "T"));
                    out.add(new                       Node(HasilbaruA,
node.getDataOutput() + "T"));
                }
                if (frek[2] >= minSupport) {
                    queueG.add(new                    Node(HasilbaruA,
node.getDataOutput() + "G"));
                    out.add(new                       Node(HasilbaruA,
node.getDataOutput() + "G"));
                }
                if (frek[3] >= minSupport) {
                    queueC.add(new                    Node(HasilbaruA,
node.getDataOutput() + "C"));
                    out.add(new                       Node(HasilbaruA,
node.getDataOutput() + "C"));
                }
            }
            if (!queueT.isEmpty()) {
                Node node = queueT.remove(0);
                Vector<Vector<String>>        HasilbaruT        =
removeCharacter(node.getDataBaru(), 'T');
                Integer[] frek = FrekuensiItem(HasilbaruT);
                if (frek[0] >= minSupport) {
                    queueA.add(new                    Node(HasilbaruT,
node.getDataOutput() + "A"));
                    out.add(new                       Node(HasilbaruT,
node.getDataOutput() + "A"));
                }
                if (frek[1] >= minSupport) {
                    queueT.add(new                    Node(HasilbaruT,
node.getDataOutput() + "T"));
                    out.add(new                       Node(HasilbaruT,
node.getDataOutput() + "T"));
                }
                if (frek[2] >= minSupport) {
                    queueG.add(new                    Node(HasilbaruT,
node.getDataOutput() + "G"));
                    out.add(new                       Node(HasilbaruT,
node.getDataOutput() + "G"));
                }
                if (frek[3] >= minSupport) {
                    queueC.add(new                    Node(HasilbaruT,
node.getDataOutput() + "C"));
```

```
                    out.add(new                    Node(HasilbaruT,
node.getDataOutput() + "C"));
                }
            }
            if (!queueG.isEmpty()) {
                Node node = queueG.remove(0);
                Vector<Vector<String>>        HasilbaruG        =
removeCharacter(node.getDataBaru(), 'C');
                Integer[] frek = FrekuensiItem(HasilbaruG);
                if (frek[0] >= minSupport) {
                    queueA.add(new                 Node(HasilbaruG,
node.getDataOutput() + "A"));
                    out.add(new                    Node(HasilbaruG,
node.getDataOutput() + "A"));
                }
                if (frek[1] >= minSupport) {
                    queueT.add(new                 Node(HasilbaruG,
node.getDataOutput() + "T"));
                    out.add(new                    Node(HasilbaruG,
node.getDataOutput() + "T"));
                }
                if (frek[2] >= minSupport) {
                    queueG.add(new                 Node(HasilbaruG,
node.getDataOutput() + "G"));
                    out.add(new                    Node(HasilbaruG,
node.getDataOutput() + "G"));
                }
                if (frek[3] >= minSupport) {
                    queueC.add(new                 Node(HasilbaruG,
node.getDataOutput() + "C"));
                    out.add(new                    Node(HasilbaruG,
node.getDataOutput() + "C"));
                }
            }
            if (!queueC.isEmpty()) {
                Node node = queueC.remove(0);
                Vector<Vector<String>>        HasilbaruC        =
removeCharacter(node.getDataBaru(), 'C');
                Integer[] frek = FrekuensiItem(HasilbaruC);
                if (frek[0] >= minSupport) {
                    queueA.add(new                 Node(HasilbaruC,
node.getDataOutput() + "A"));
                    out.add(new                    Node(HasilbaruC,
node.getDataOutput() + "A"));}
                if (frek[1] >= minSupport) {
                    queueT.add(new                 Node(HasilbaruC,
node.getDataOutput() + "T"));
                    out.add(new                    Node(HasilbaruC,
node.getDataOutput() + "T"));}
                if (frek[2] >= minSupport) {
                    queueG.add(new                 Node(HasilbaruC,
node.getDataOutput() + "G"));
                    out.add(new                    Node(HasilbaruC,
node.getDataOutput() + "G"));}
                if (frek[3] >= minSupport) {
                    queueC.add(new                 Node(HasilbaruC,
node.getDataOutput() + "C"));
```

```
                    out.add(new                         Node(HasilbaruC,
node.getDataOutput() + "C")); }
            }
        }
        String var = "";
        for (int i = 0; i < out.size(); i++) {
            var += "<" + out.get(i).getDataOutput() + ">";
        }
        return var;
    }
```

Sourcecode 4.3 Proses Implementasi Algoritma PrefixSpan

4.2.4 Proses *Subsequence*

Setelah didapatkan pola-pola sekuen DNA dari proses algoritma *prefixspan* dari data yang diberikan, dilakukan proses *subsequence*. Proses ini adalah proses pemilihan sekuen dimana sekuen yang merupakan bagian dari sekuen lain maka akan dihilangkan. Hal ini bertujuan untuk mendapatkan suatu pola sekuen yang terbaik yang mengindikasikan mutasi yang terjadi penyebab dari penyakit kanker payudara. Adapaun implementasi proses ini dalam sistem akan dijelaskan pada Sourcode 4.4 berikut.

```
public boolean prosesSubsequence(String a, String b) {
        int i = 0;
        int j = a.length();
        while (j <= b.length()) {
            String c = b.substring(i, j);
            if (c.equals(a)) {
                return true;
            }
            i++;
            j++;
        }
        return false;
    }
    public String subSequence() {
        for (int i = 0; i < out.size(); i++) {
            boolean a = false;
            for (int j = i + 1; j < out.size(); j++) {
                if  (prosesSubsequence(out.get(i).getDataOutput(),
out.get(j).getDataOutput()) == true) {
                    a = true;
                }
            }
            if (a == false) {
                Subsequence.add(out.get(i).getDataOutput());
            }
        }
        String var2 = "";
        for (int i = 0; i < Subsequence.size(); i++) {
            var2 += "<" + Subsequence.get(i) + ">";
        }
```

```
        return var2;
    }
```

Sourcecode 4.4 Proses Subsequence

4.2.5 Penentuan LiftRatio, Confidence, Support

Setelah didapatkan pola dari hasil proses *subsequence*, pola-pola tersebut kemudian dihitung nilai *Lift*, *Confidence* dan *Support*nya hal ini bertujuan untuk menentukan hubungan pola tersebut terhadap data. Dan nilai lift rasio dari pola tersebut menentukan apakah pola tersebut berhubungan erat dengan data sehingga dapat digunakan sebagai pola akhir yang utama penyebab kanker payudara. Sourcode 4.5 akan menjelaskan proses pencarian nilai-nilai tersebut.

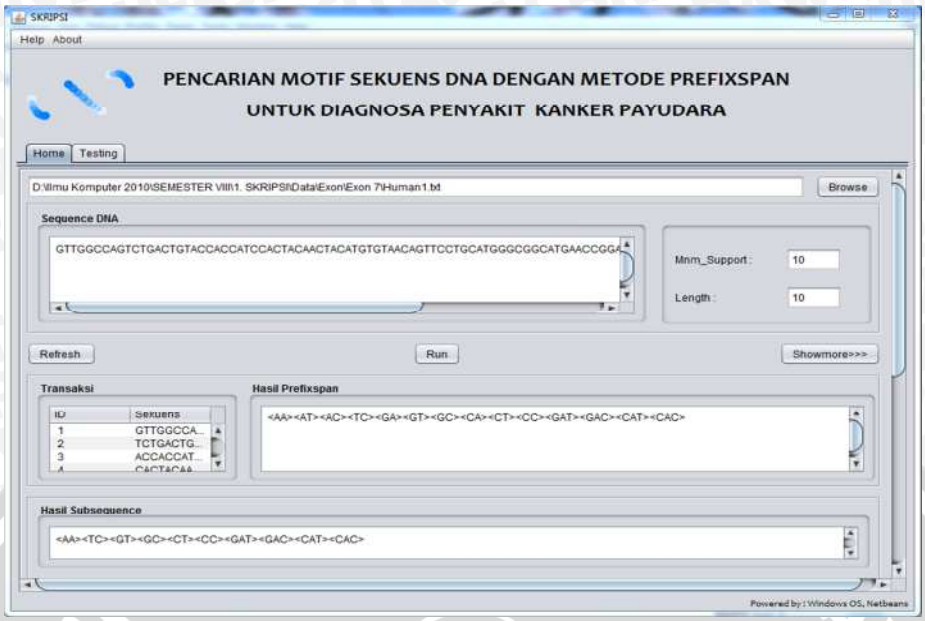
```
public Vector<Float> proseslift dll() {
        for (int i = 0; i < Subsequence.size(); i++) {
            float ac = 0;
            for (int j = 0; j < HasilTransaksi.size(); j++) {
                if (Matching(Subsequence.get(i),
HasilTransaksi.get(j).toString())) {
                    ac += 1;
                }
            }
            Smpn_Lift_Rat_Supp.add(ac);
        }
        return Smpn_Lift_Rat_Supp;
    }
    public Vector<Vector<Float>> LiftRatio() {
        float Lift = 0;
        float number = 1;
        for (int i = 0; i < Smpn_Lift_Rat_Supp.size(); i++) {
            Vector<Float> Ratio = new Vector<Float>(4);
            Ratio.add(number);
            float support = Smpn_Lift_Rat_Supp.get(i);
            float Confidence = Smpn_Lift_Rat_Supp.get(i) /
Smpn_Lift_Rat_Supp.get(i);
            float EConfidence = Smpn_Lift_Rat_Supp.get(i) /
HasilTransaksi.size();
            Lift = Confidence / EConfidence;
            RSupport.add(support);
            RConfidence.add(Confidence);
            RLift.add(Lift);
            Ratio.add(support);
            Ratio.add(Confidence);
            Ratio.add(Lift);
            Ratio2.add(Lift);
            HasilLift.add(Ratio);
            number++;
        }
        return HasilLift;
    }
    //#Pencarian Rata-Rata Lift, Confidence, Support
    public String getRata2Lift() {
        float R2Lift = 0;
```

```
        for (int i = 0; i < RLift.size(); i++) {
            R2Lift += RLift.get(i);
        }
        float Rata2Lift = R2Lift / RLift.size();
        return "" + Rata2Lift;
    }

    public String getRata2Support() {
        float R2Supp = 0;
        for (int i = 0; i < RSupport.size(); i++) {
            R2Supp += RSupport.get(i);
        }
        float Rata2Support = R2Supp / RSupport.size();
        return "" + Rata2Support;
    }

    public String getRata2Confidence() {
        float R2Con = 0;
        for (int i = 0; i < RConfidence.size(); i++) {
            R2Con += RConfidence.get(i);
        }
        float Rata2Confidence = R2Con / RConfidence.size();
        return "" + Rata2Confidence;
    }


    //#Penyimpanan Output
    public String Output() {
        for (int i = 0; i < Ratio2.size(); i++) {
            if (Ratio2.get(i) > 1) {
                Output.add(Subsequence.get(i));
            }
        }
        String var3 = " ";
        for (int i = 0; i < Output.size(); i++) {
            var3 += "<" + Output.get(i) + ">";
        }
        return var3;
    }
```

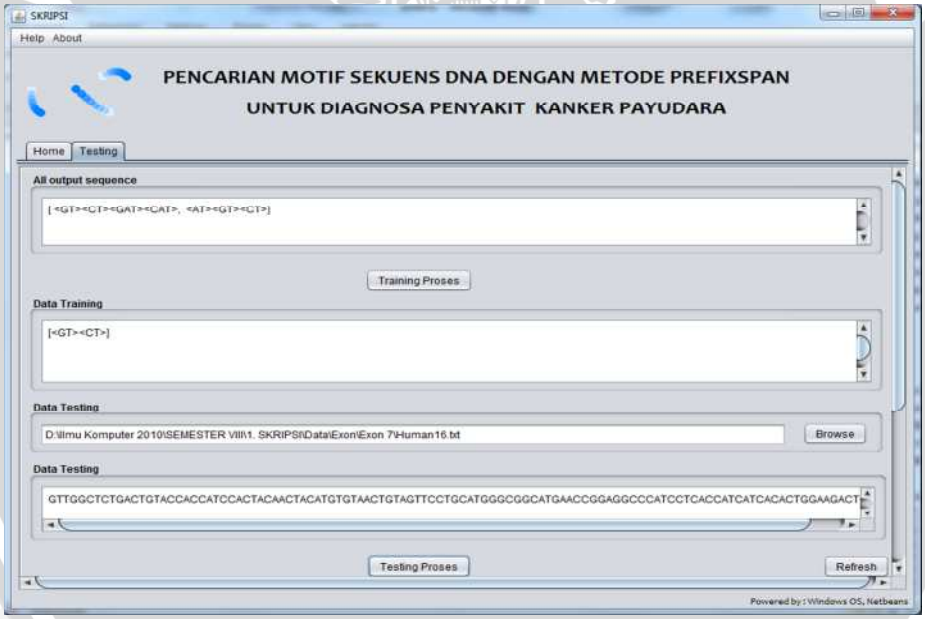
Sourcecode 4.5 Pencarian Nilai *Lift*, *Confidence* dan *Support*

4.3 Implementasi Antarmuka

Antar muka sistem terdiri atas satu form yang memiliki 2 tab. Tab pertama berfungsi untuk menjalankan proses pencarian motif sekuens DNA dan tab kedua merupakan form untuk melakukan proses testing akurasi sistem. Proses dijalankan dengan sistem *one klick* dan untuk melakukan proses pengulangan percobaan disediakan menu *refresh*. Gambar 4.1 dan 4.2 akan mengambarkan implementasi antarmuka dari sistem.



Gambar 4.1 Form Utama Sistem



Gambar 4.2 Form Testing Sistem

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas dan dianalisis dari hasil uji coba yang telah dilakukan dalam membentuk motif sekuens DNA.

5.1 Hasil Pengujian Sistem

Hasil pengujian terhadap sistem dilakukan untuk mengetahui motif-motif yang dihasilkan dari proses algoritma *prefixspan*. Dalam uji coba ini akan dilakukan pengujian terhadap 20 data gen DNA penderita kanker payudara pada exon ke-7 dengan *minimum support* 2, 4, 6, 8, 10 dan 12 sedangkan *length* yang digunakan adalah 10 untuk semua *minimum support*. Dalam sub bab ini tidak akan ditampilkan keseluruhan hasil yang didapatkan dari proses algoritma *prefixspan* dikarenakan akan terlalu banyak, namun keseluruhan hasil tersebut akan ditampilkan pada lampiran 1 tugas akhir skripsi ini. Tabel 5.1 akan dijelaskan sebagian hasil pengujian yang telah dilakukan.

Tabel 5.1 Hasil Uji Coba Sistem

Individu ke-	Minimum Support	Length	Motif Sekuens	Jumlah Motif
1	4	10	<TAT><TTT><AAAT><ATGT><ACT><AGGT><ATCT><AGAT><AGTT><ACGT><AGCT><ACAT><ACTT><TAGT><ACCT><TACT><TGGT><TGAT><TGAG><TCGT><TGCT><TCAT><TCAG><GAGT><TCCT><GATT><GTGT><GACT><GTCT><CAGT><CATT><CTGT><CACT><CTCT>	34
	8	10	<TT><AGT><ACT><TCT><GAT><GAG><CAT><CAG>	8

Hasil uji coba sistem ditunjukkan pada Tabel 5.1 dalam pencarian motif sekuens DNA penderita kanker payudara dengan algoritma *prefixspan*. Dari tabel 5.1 dapat diketahui motif-motif yang dihasilkan, dalam sekali proses motif yang didapatkan bisa lebih dari sama dengan 1. Setiap motif dipisahkan oleh tanda kurung siku (< / >). Motif yang diapit dalam tanda kurung siku (< / >) dihitung sebagai satu motif dan berbeda dengan motif yang lain.

5.2 Hasil Pengujian Motif

Proses uji coba ini dilakukan untuk mengetahui performansi dan keakurasian motif-motif sekuens DNA dari proses algoritma *prefixspan*. Performansi motif ditentukan dari nilai lift rasio, motif sekuen dikatakan baik apabila nilai lift rasio lebih dari 1, dan semakin tinggi semakin baik. Pada proses pengujian ini akan diuji pengaruh *minimum support* terhadap lift rasio dan support, pengaruh length serta akurasi dari motif yang dihasilkan. Pada *minimum support* berapa motif sekuen yang memiliki lift rasio motif yang terbaik dan berapa akurasi dari motif yang dihasilkan tersebut jika dilakukan pengujian terhadap data lain. Sedangkan pengujian terhadap support dilakukan sebagai pembanding untuk menentukan kekuatan rule. Data yang diujikan sebanyak 20 data manusia berbeda yang menderita kanker payudara, dimana gen DNA yang digunakan adalah Exon ke-7 dari manusia penderita kanker payudara. Hasilnya kemudian dianalisa untuk diambil suatu kesimpulan dari penelitian ini.

5.2.1 Hasil Uji Coba pengaruh Minimum Support

Dalam uji coba ini, nilai minimal support yang akan diujikan mengalami perubahan mulai *minimum support* 2, 4, 6, 8, 10 dan 12. Sedangkan panjang sekuens untuk setiap transaksi dibuat sama yaitu 10. Dalam proses ini akan digunakan 20 data dari manusia penderita kanker payudara dimana gen DNA yang diambil merupakan DNA Exon ke-7 dari penderita kanker payudara. Uji coba ini dilakukan dengan mencari dan menganalisis hasil nilai rata-rata support, *confidence* dan lift rasio dari motif yang didapatkan. Tabel 5.1 akan menampilkan hasil uji coba yang telah dilakukan.

Tabel 5.2 Hasil Uji Coba Pengaruh Minimum Support

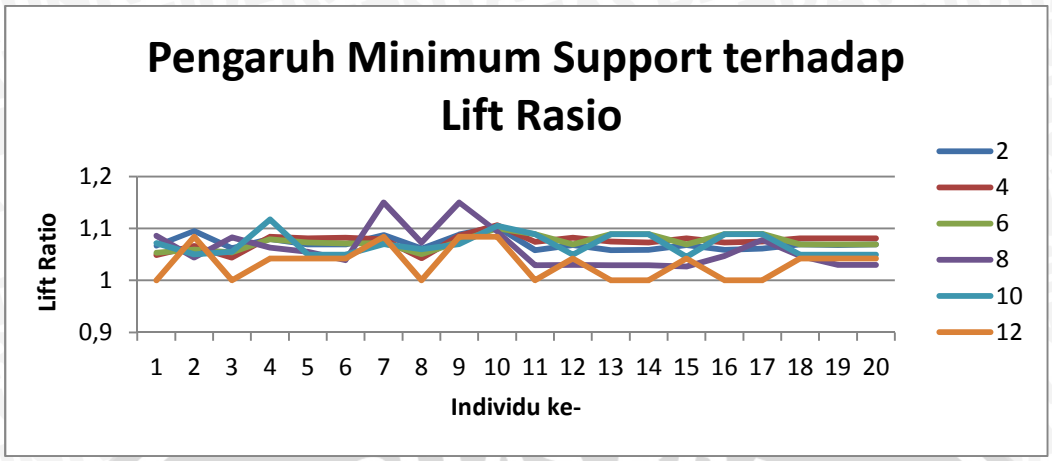
Individu ke-	Minimum Support	Length	Rata-Rata Confidence	Rata-Rata Support	Rata-Rata LiftRatio
1	2	10	1	12,28	1,067
	4	10	1	12,471	1,049
	6	10	1	12,42	1,053
	8	10	1	12,13	1,086
	10	10	1	12,2	1,072
	12	10	1	13	1
2	2	10	1	11,94	1,095
	4	10	1	12,21	1,066

	6	10	1	12,29	1,0595
	8	10	1	12,47	1,044
	10	10	1	12,4	1,05
	12	10	1	12	1,084
3	2	10	1	12,33	1,062
	4	10	1	12,51	1,044
	6	10	1	12,41	1,054
	8	10	1	12,17	1,083
	10	10	1	12,4	1,055
	12	10	1	13	1
4	2	10	1	12,11	1,0799
	4	10	1	12,04	1,084
	6	10	1	12,11	1,079
	8	10	1	12,29	1,063
	10	10	1	11,75	1,1176
	12	10	1	12,5	1,042
5	2	10	1	12,21	1,069
	4	10	1	12,07	1,081
	6	10	1	12,16	1,073
	8	10	1	12,38	1,055
	10	10	1	12,45	1,0497
	12	10	1	12,5	1,042
6	2	10	1	12,22	1,069
	4	10	1	12,06	1,082
	6	10	1	12,18	1,071
	8	10	1	12,54	1,039
	10	10	1	12,45	1,0497
	12	10	1	12,5	1,042
7	2	10	1	12,04	1,087
	4	10	1	12,12	1,079
	6	10	1	12,18	1,072
	8	10	1	11,5	1,15
	10	10	1	12,22	1,0699
	12	10	1	12	1,084
8	2	10	1	12,33	1,0614
	4	10	1	12,53	1,043
	6	10	1	12,44	1,051
	8	10	1	12,26	1,073
	10	10	1	12,33	1,06
	12	10	1	13	1
9	2	10	1	12,03	1,0884
	4	10	1	12,05	1,086
	6	10	1	12,15	1,075
	8	10	1	11,5	1,15
	10	10	1	12,22	1,0699

	12	10	1	12	1,084
10	2	10	1	11,795	1,103
	4	10	1	11,77	1,106
	6	10	1	11,82	1,1
	8	10	1	11,89	1,094
	10	10	1	11,78	1,105
	12	10	1	12	1,084
11	2	10	1	12,33	1,058
	4	10	1	12,14	1,074
	6	10	1	11,98	1,089
	8	10	1	12,67	1,029
	10	10	1	12	1,089
	12	10	1	13	1
12	2	10	1	12,22	1,068
	4	10	1	12,06	1,082
	6	10	1	12,2	1,069
	8	10	1	12,65	1,0299
	10	10	1	12,45	1,0497
	12	10	1	12,5	1,042
13	2	10	1	12,32	1,058
	4	10	1	12,13	1,075
	6	10	1	11,97	1,089
	8	10	1	12,67	1,029
	10	10	1	12	1,089
	12	10	1	13	1
14	2	10	1	12,31	1,059
	4	10	1	12,15	1,073
	6	10	1	11,98	1,089
	8	10	1	12,67	1,029
	10	10	1	12	1,089
	12	10	1	13	1
15	2	10	1	12,21	1,068
	4	10	1	12,07	1,081
	6	10	1	12,21	1,069
	8	10	1	12,68	1,027
	10	10	1	12,5	1,045
	12	10	1	12,5	1,042
16	2	10	1	12,31	1,059
	4	10	1	12,15	1,073
	6	10	1	12	1,089
	8	10	1	12,47	1,047
	10	10	1	12	1,089
	12	10	1	13	1
17	2	10	1	12,31	1,061
	4	10	1	12,13	1,075

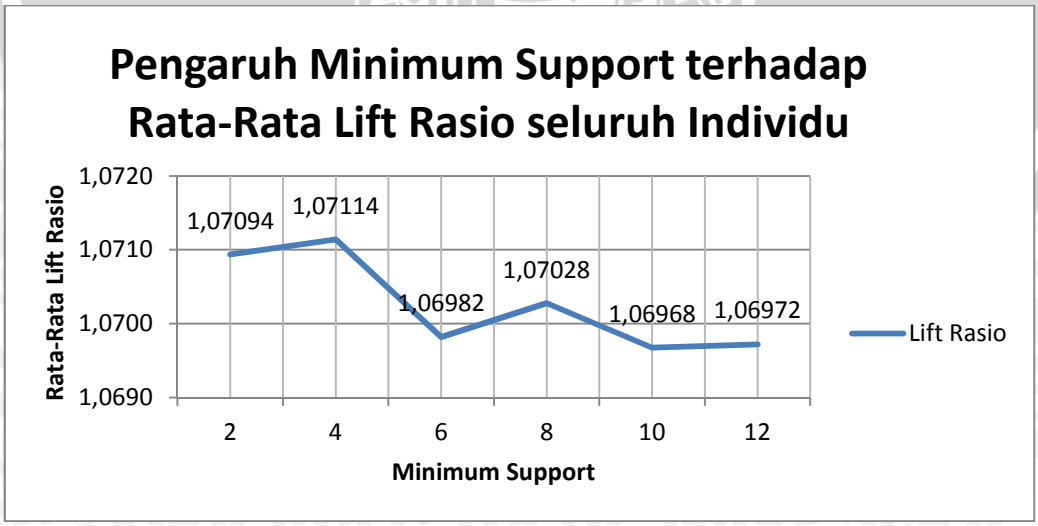
	6	10	1	11,98	1,0897
	8	10	1	12,24	1,078
	10	10	1	12	1,089
	12	10	1	13	1
18	2	10	1	12,22	1,069
	4	10	1	12,07	1,081
	6	10	1	12,196	1,069
	8	10	1	12,46	1,047
	10	10	1	12,45	1,0497
	12	10	1	12,5	1,042
19	2	10	1	12,22	1,068
	4	10	1	12,07	1,081
	6	10	1	12,21	1,069
	8	10	1	12,65	1,0299
	10	10	1	12,45	1,0497
	12	10	1	12,5	1,042
20	2	10	1	12,22	1,069
	4	10	1	12,07	1,081
	6	10	1	12,21	1,069
	8	10	1	12,65	1,0299
	10	10	1	12,45	1,0497
	12	10	1	12,5	1,042

Hasil rata-rata *confidence*, support dan lift rasio dari motif sekuens yang dihasilkan dari proses algoritma prefixspan ditampilkan pada Tabel 5.2. Hasil-hasil ini digunakan sebagai acuan untuk mengetahui kekuatan dari rule yang dihasilkan. Dalam menjelaskan hasil ini untuk lebih mudah dapat dibentuk suatu grafik pengaruh *minimum support* terhadap support dan lift rasio dari motif yang terbentuk. Gambar 5.1 dan 5.3 akan menjelaskan pengaruh *minimum support* terhadap lift rasio dan support yang dihasilkan dan pada akhirnya dapat menjadi bahan untuk analisis performansi motif. Pengujian terhadap support digunakan sebagai pembanding dalam menentukan performansi motif



Gambar 5.1 Grafik Pengaruh Minimum Support terhadap Lift Rasio

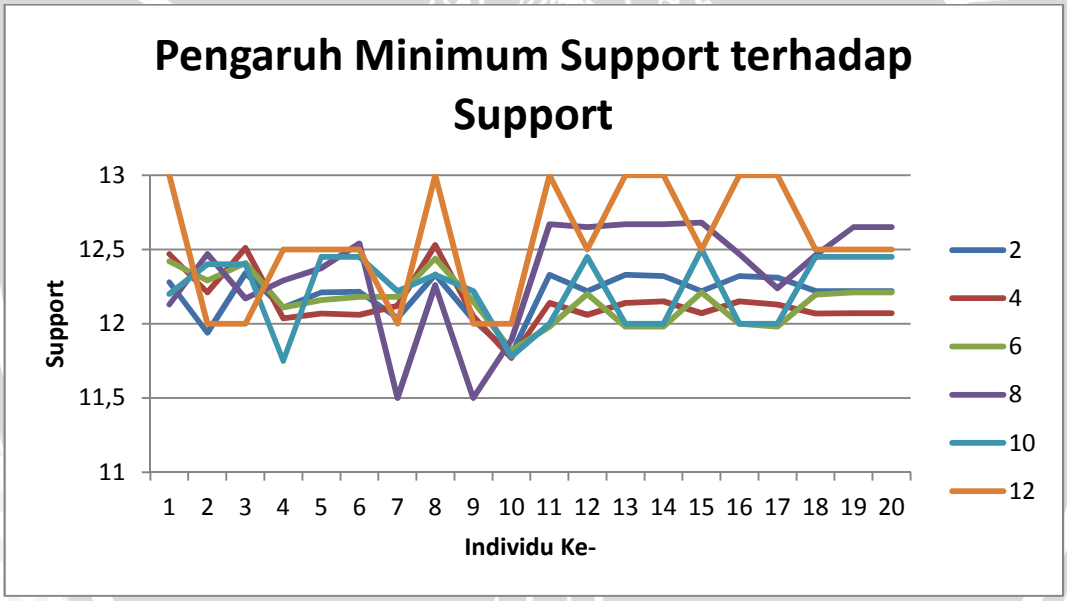
Hubungan atau pengaruh perubahan *minimum support* terhadap nilai lift rasio dari motif DNA penyebab kanker yang dihasilkan dari algoritma *prefixspan* dijelaskan pada Gambar 5.1. Dari uji coba ini didapatkan nilai lift rasio terbesar terdapat di *minimum support* 8 pada pasien manusia ke-7 dan 9 dengan nilai lift rasio 1,15. Dan pada *minimum support* 12 didapatkan nilai lift rasio terkecil yaitu 1 pada pasien manusia ke-1, 3, 8, 11, 13, 14, 16, dan 17. Hasil diatas adalah hasil uji coba untuk setiap individu, sehingga akan dibuat juga grafik rata-rata lift rasio dari seluruh individu sebagai pembanding yang dijelakan pada Gambar 5.2 berikut.



Gambar 5.2 Grafik Pengaruh Minimum Support terhadap Rata-Rata Lift Rasio seluruh Individu

Rata-rata lift rasio dari ke-20 individu penderita kanker payudara pada minimum support yang berbeda-beda ditunjukkan pada Gambar 5.2. Dan jika dilihat dari hasil rata-rata ini maka lift rasio yang memiliki nilai tertinggi terdapat pada *minimum support* 4 dengan nilai 1,07114 dan pada *minimum support* 12 didapatkan nilai rata-rata lift rasio terkecil yaitu 1,06972.

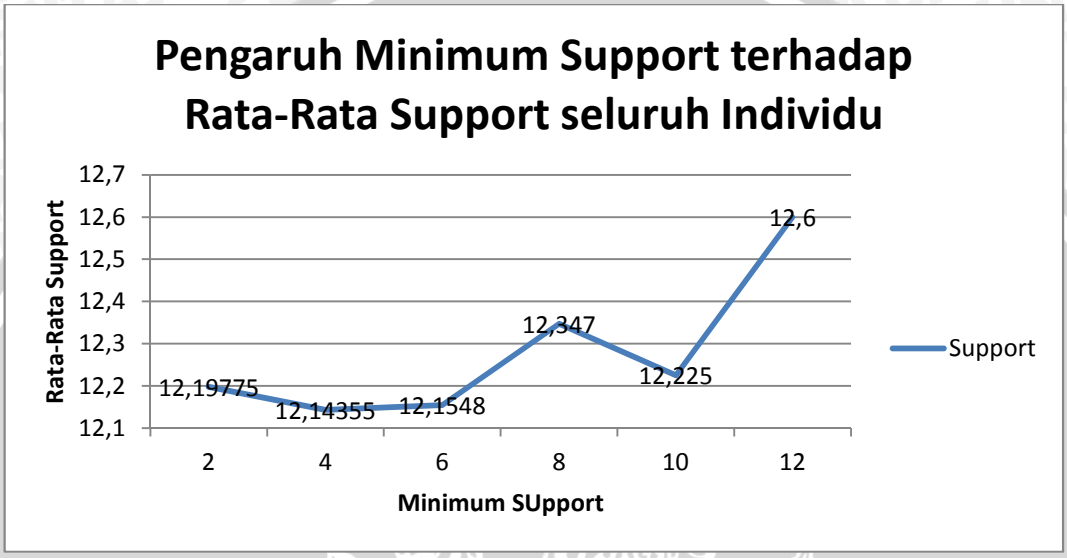
Uji coba selanjutnya adalah uji coba untuk mengetahui pengaruh minimum support terhadap support yang dihasilkan dari motif yang ditemukan menggunakan algoritma *prefixspan*. Support menunjukkan nilai kemunculan dari suatu motif dalam suatu transaksi. Dalam penelitian ini dilakukan pengujian ini untuk digunakan sebagai pembanding antara pengaruh *minimum support* terhadap lift rasio dan support. Gambar 5.3 akan menunjukkan hubungan antara *minimum support* dan support yang dihasilkan.



Gambar 5.3 Grafik Pengaruh Minimum Support terhadap Support

Dilihat dari Gambar 5.3 dapat diambil suatu kesimpulan bahwa support yang memiliki nilai tertinggi terdapat pada *minimum support* 12 dengan nilai 13 untuk individu ke-2, 8, 11, 13, 14, 16, dan 17. Sedangkan nilai terendah terdapat pada *minimum support* 8 dengan nilai 11,5 untuk individu ke-7 dan ke -9. Melihat grafik pada Gambar 5.3 jika dibandingkan dengan Gambar 5.1 maka dapa diambil suatu kesimpulan bahwa support berbanding terbalik dengan nilai lift rasio.

Minimum support yang menghasilkan nilai lift rasio tertinggi maka akan menghasilkan support terendah sedangkan yang menghasilkan nilai lift rasio terendah menghasilkan support tertinggi. Untuk semakin memperjelaskan perbandingan kedua hal ini maka akan digambarkan pula grafik rata-rata hubungan minimum support terhadap support yang akan digambarkan pada Gambar 5.4 berikut.



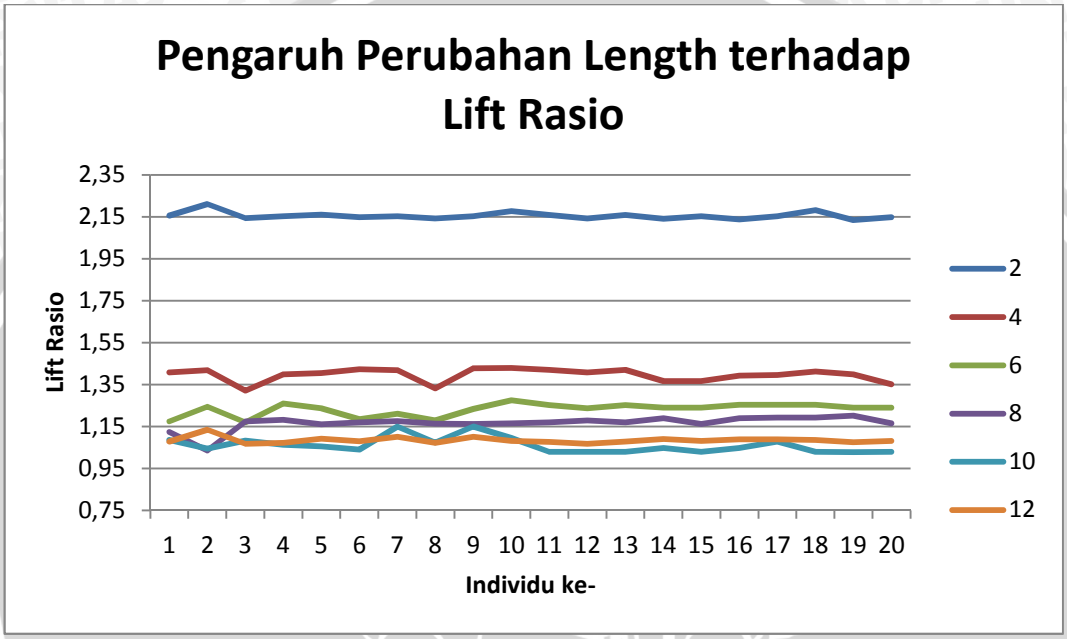
Gambar 5.4 Grafik Pengaruh Minimum Support terhadap Rata-Rata Support seluruh Individu

Pengaruh *minimum support* terhadap rata-rata support digambarkan pada Gambar 5.4. Suatu kesimpulan awal dapat kita ambil yaitu bahwa pada *minimum support* 12 didapatkan support tertinggi yaitu 12,6 dan pada *minimum support* 4 didapatkan nilai rata-rata support terendah yaitu 12,14355. Hal ini memperjelas hasil sebelumnya yang menyatakan bahwa support berbanding terbalik dengan lift rasio.

5.2.2 Hasil Uji Coba pengaruh Length

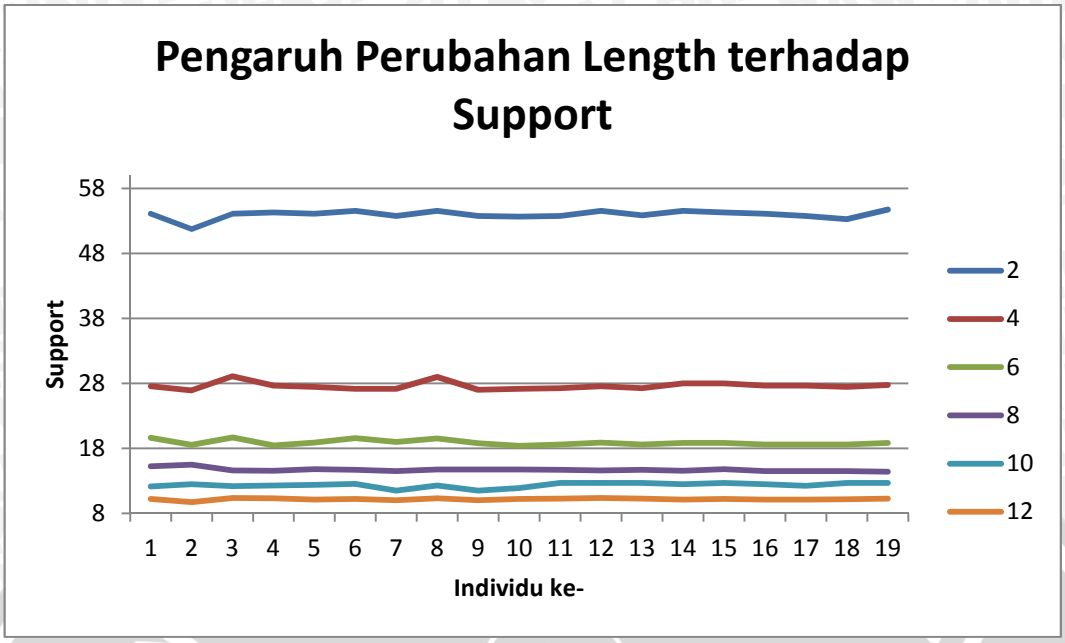
Dalam uji coba ini, nilai *length* yang akan diujikan mengalami perubahan mulai dari *length* 2, 4, 6, 8, 10 dan 12. Sedangkan minimum support yang diujikan dibuat sama yaitu 8. Dalam proses ini ,sama seperti sebelumnya, akan digunakan 20 data dari manusia penderita kanker payudara dimana gen DNA yang diambil merupakan DNA Exon ke-7 dari penderita kanker payudara. Uji coba ini

dilakukan untuk mengetahui pengaruh perubahan *length* terhadap motif yang dihasilkan dengan melihat nilai lift rasio dan support yang dihasilkan. Dan hasil uji coba ini ditampilkan pada lampiran 2. Sedangkan pada sub bab ini akan dijelaskan analsisi hasil yang diperoleh dalam bentuk grafik. Gambar 5.5 dan 5.6 akan menampilkan grafik hasil uji coba pengaruh length terhadap lift rasio dan support yang dihasilkan.



Gambar 5.5 Grafik Pengaruh Perubahan Length terhadap Lift Rasio

Pengaruh perubahan *Length* terhadap Lift Rasio digambarkan pada Gambar 5.5. Berdasarkan grafik ini nilai lift rasio terbesar dari semua individu yang diujikan didapatkan ketika nilai *length* 2 yang merupakan nilai *length* terkecil. Namun pada *length* terbesar yaitu 12 tidak didapatkan nilai lift rasio terkecil karena pada *length* 12 tedapat nilai lift rasio yang diatas nilai length 10. Sehingga tidak dapat diambil kesimpulan secara jelas dari grafik ini, karena tidak berarti jika pada *length* terkecil didapatkan nilai lift rasio terbesar maka pada length terbesar didapatkan nilai lift rasio terkecil. Namun dari grafik ini dapat dinyatakan bahwa pada length 2 lift rasio terbesar dihasilkan. Sehingga nilai *length* 2 dapat digunakan untuk proses uji coba lanjutan agar dihasilkan hasil yang lebih baik.



Gambar 5.6 Grafik Pengaruh Perubahan Length terhadap Support

Pengaruh perubahan Length terhadap Support digambarkan pada Gambar 5.6. Berdasarkan grafik ini nilai support terbesar dari semua individu yang diujikan didapatkan ketika panjang length 2. Dan nilai support terkecil didapatkan ketika panjang length 12. Hal ini menjelaskan bahwa semakin kecil length maka kemunculan dari motif yang dihasilkan atau support dari motif semakin besar dan sebaliknya.

5.2.3 Hasil Uji Coba Akurasi Sistem

Uji coba akurasi dilakukan dengan mengujikan pola-pola yang dihasilkan dengan sel DNA manusia lain penderita kanker payudara pada exon yang sama yaitu exon ke-7. Dalam proses ujicoba ini diasumsikan bahwa proses perhitungan akurasi yang dilakukan oleh sistem dianggap sama seperti yang dilakukan oleh pakar. *Minimum support* yang dipilih dalam uji coba ini adalah minimum support yang telah menghasilkan nilai lift rasio tertinggi yaitu minimun support 8 dan *minimum supoort* dengan nilai rata-rata kesulurahan tertinggi yaitu *minimum support* 4 yang diuji cobakan pada *length* 2 dan 10. Masing-masing *minimum support* akan dilakukan tiga kali perubahan jumlah data training. Tabel 5.3 akan

menampilkan hasil uji coba dengan *length* 2 dan Tabel 5.4 akan menampilkan hasil uji coba dengan *length* 10 seperti berikut.

Tabel 5.3 Akurasi Hasil Testing dengan nilai length 2

Minimum Support	Length	Jumlah Training Data	Jumlah Testing Data	Rata-rata Akurasi
4	2	5	5	100%
4	2	10	5	100%
4	2	15	5	100%
8	2	5	5	100%
8	2	10	5	100%
8	2	15	5	100%

Tabel 5.4 Akurasi Hasil Testing dengan nilai length 10

Minimum Support	Length	Jumlah Training Data	Jumlah Testing Data	Rata-rata Akurasi
4	10	5	5	100%
4	10	10	5	100%
4	10	15	5	100%
8	10	5	5	100%
8	10	10	5	100%
8	10	15	5	100%

Hasil akurasi dari uji coba motif sekuen yang dihasilkan ditampilkan pada Tabel 5.3 dan 5.4. Uji coba ini dilakukan dengan memproses data training terlebih dahulu, motif dari beberapa individu dikumpulkan yang kemudian diambil motif terbanyak yang sering terjadi. Kemudian motif-motif tersebut diujikan terhadap 5 data testing. Akurasi yang dihasilkan adalah akurasi rata-rata dari 5 kali testing yang terjadi. Dalam uji coba ini diuji pengaruh jumlah data training terhadap akurasi serta pengaruh perubahan *length*. Dan dari hasil uji coba dapat dilihat bahwa perubahan data training dan nilai *length* tidak menyebabkan perubahan akurasi, dimana akurasi keseluruhan dari motif yang didapatkan rata-rata mencapai 100%.

5.3 Analisis Hasil Pengujian

Subbab ini akan membahas analisa dari hasil uji coba yang telah dilakukan, apa yang terjadi dari proses uji coba yang dilakukan dan mengapa hal itu terjadi. Berdasarkan hasil uji coba yang dilakukan terhadap sistem untuk mengetahui motif yang dihasilkan dari algoritma *prefixspan*, seperti pada Tabel 5.1 menunjukkan bahwa sistem mampu bekerja dengan baik untuk melakukan pencarian motif sekuen DNA penyebab kanker payudara dengan metode *prefixspan*. Uji coba ini juga menjelaskan bahwa dalam sekali proses pengolahan suatu data oleh algoritma prefixspan dapat menghasilkan lebih dari satu motif dimana setiap motif terdiri minimal dari dua karakter.

Uji coba yang kedua adalah ujicoba untuk mengetahui performansi motif yang dihasilkan. Uji coba ini dilakukan dengan melihat pengaruh nilai *minimum support* terhadap rata-rata support dan lift rasio dari motif yang dihasilkan. Dengan melihat Gambar 5.1 dan 5.2 dapat dilihat hubungan antara *minimum support* dan lift rasio, dari grafik ini *minimum support* yang menghasilkan motif yang memiliki rata-rata lift rasio tertinggi terdapat pada uji coba dengan *minimum suppo*rt 8 dengan nilai rata-rata lift rasio tertinggi yaitu 1,15 pada data manusia ke-7 dan 9. Tetapi dari Gambar 5.2, secara rata-rata *minimum support* yang menghasilkan rata-rata lift rasio tertinggi adalah *minimum support* 4 dengan nilai lift rasio rata-rata 1,07114. Melihat hal ini tentunya kesulitan untuk mengambil suatu kesimpulan, namun jika dilihat secara keseluruhan dengan melihat Gambar 5.2 dapat dikatakan bahwa semakin besar nilai minimum support maka semakin kecil nilai lift rasio yang dihasilkan dibuktikan dengan grafik yang secara bertahap menurun. Namun penurunan tersebut masih tidak stabil dan mungkin grafik bisa kembali naik dengan data lebih dari 20, sehingga kesimpulan ini dibentuk untuk jumlah uji coba yang hanya mencapai 20 data saja.

Uji coba ketiga dilakukan juga untuk melihat pengaruh *minimum support* terhadap support yang nantinya dapat digunakan sebagai bahan perbandingan. Dengan melihat Gambar 5.3 dapat dilihat hubungan keduanya dimana *minimum support* yang menghasilkan support tertinggi terdapat pada *minimum support* 12 dengan support 13 pada individu ke-2, 8, 11, 13, 14, 16, dan 17. Sedangkan nilai

terendah terdapat pada *minimum support* 8 yaitu 11,5 untuk individu ke-7 dan ke -9. Sedangkan dari Gambar 5.4 dapat dilihat rata-rata support tertinggi terdapat pada *minimum support* 12 dengan nilai 12,6 dan pada *minimum support* 4 didapatkan nilai rata-rata support terendah yaitu 12,14355. Jika hal ini dibandingkan dengan pengaruh lift rasio seperti terlihat pada Gambar 5.2, maka terdapat hubungan berbanding terbalik karena pada *minimum support* tertinggi nilai lift rasio rendah sedangkan nilai support menjadi tinggi. Hal ini dapat menjelaskan bahwa support tidak bisa digunakan sebagai indikator penentu kekuatan dari suatu motif. Karena support hanya melihat dari faktor kemunculannya saja.

Uji Coba yang keempat dilakukan untuk melihat pengaruh perubahan nilai *length* terhadap nilai lift rasio dan support yang diberikan. Dari hasil uji coba yang telah dilakukan seperti pada Gambar 5.5 dihasilkan bahwa pada length 2 didapatkan nilai lift rasio terbesar. Namun pada *length* terbesar, dari proses uji coba, yaitu *length* 12 tidak didapatkan nilai lift rasio terkecil. Sehingga tidak berarti jika pada nilai *length* terkecil didapatkan nilai lift rasio terbesar maka pada *length* terbesar didapatkan nilai lift rasio terkecil. Melihat hal ini maka kesimpulan yang dapat diambil dari uji coba ini adalah length yang terbaik digunakan dalam proses uji coba adalah 2 yang dibuktikan dengan nilai lift rasio yang terbesar. Sedangkan untuk pengaruh perubahan nilai *length* terhadap support seperti yang dijelaskan pada Gambar 5.6 menunjukkan bahwa support terbesar dihasilkan pada *length* 2 dan support terkecil dihasilkan pada *length* 12. Jika melihat hal ini menunjukkan bahwa semakin kecil nilai *length* maka kemunculan atau support dari motif yang dihasilkan semakin besar dan sebaliknya. Melihat hasil uji coba keempat ini maka sebenarnya lebih tepatnya bahwa uji coba kedua dan ketiga dilakukan dengan panjang length 2.

Proses uji coba yang terakhir adalah proses uji coba untuk menentukan akurasi dari motif yang dihasilkan. Hal ini dilakukan dengan mengujikan motif yang telah didapatkan dengan sekuens DNA penderita kanker payudara lain. Uji coba dilakukan dengan mengujikan *data training* terhadap 5 data testing yang berbeda-beda. *Data training* disini adalah data motif-motif yang didapatkan dari beberapa individu dengan algoritma prefixspan pada minimum support dan

panjang length yang sama kemudian diambil motif yang terbaik. Motif yang terbaik adalah motif yang memiliki jumlah kemunculan terbanyak dari motif sekuen yang dihasilkan dari beberapa individu. Misalnya, akan dicari *data training* dari motif yang dihasilkan dari sekuens individu 1, 2, dan 3 pada *minimum support* 8. Berikut ini akan dijelaskan melalui contoh sederhana proses pencarian akurasi yang dilakukan,

Motif Sekuens Individu ke-1 :

<TT><AGT><ACT><TCT><GAT><GAG><CAT><CAG>

Motif Sekuens Individu ke-2 :

<TA><TT><AAT><AGA><AGT><ACA><ACT><GAT><CAT>

Motif Sekuen Individu ke-3 :

<TT><GG><CG><AGT><ACT><TCT><GAT><CAT>

Dari motif sekuens yang didapat ini dihitung kemunculan semua motif yang terjadi seperti berikut,

<TT>=3 ,<AGT>=3 ,<ACT>=3 ,<TCT>=2 ,<GAT>=3 ,<GAG>=1 ,<CAT>=3 ,<CAG>=1 , <TA>=1 ,<AAT>=1 ,<AGA>=1 ,<ACA>=1 ,<GG>=1 , <CG>=1 ,<AGT>=1

Berdasarkan hasil tersebut dapat ditemukan nilai kemunculan terbanyak yaitu 3 dan motif yang memiliki jumlah kemunculan 3 inilah yang menjadi *data training* untuk dilakukan proses *testing*.

Data Training : [<TT><AGT><ACT><GAT><CAT>]

Data training tersebut yang kemudian digunakan sebagai testing proses. *Testing* proses dilakukan dengan melakukan *string matching* terhadap masing-masing motif pada data training terhadap *data testing* yang merupakan data awal gen DNA penderita kanker payudara yang bukan merupakan data pembentuk *data training*. Berikut ilustrasinya,

Data testing : Data dari individu ke -4

GTTGGCTCTGACTGTACCACCATCCACTACGCAAACTACATGTGTAAC
AGTTCCTGCATGGGCGGCATGAACCGGAGGCCCATCCTCACCATCATC
ACACTGGAAGACTCCAG

Proses *String Matching* :

1. Membagi dan mengelompokkan data testing sepanjang *leng*th = 10

2. *String Matching* terhadap masing-masing kelompok
3. Penghitungan jumlah string matching dalam masing-masing motif yang sesuai dengan *minimum support*
4. Penghitungan akurasi

Contoh :

1. Pengelompokan *Data Testing* dengan *length*=10 (Pembentukan Transaksi)

1	GTTGGCTCTG
2	GACTGTACCA
3	ACCATCCACT
4	TACGCAAACT
5	TACATGTGTA
6	AACAGTTCCT
7	TGCATGGGCG
8	GGCATGAACC
9	CGGAGGCCCA
10	ATCCTCACCA
11	ATCATCACAC
12	CTGGAAGACT
13	CCAG

2. *String Matching* untuk motif <AGT>

1	GTTGGCTCTG
2	G**A**CT**GT**ACCA
3	ACCATCCACT
4	T**A**C**G**CAAAC**T**
5	T**A**CAT**GT**GTA
6	**A**ACA**GT**TCCT
7	TGCATGGGCG
8	GGCATGAACC
9	CGGAGGCCCA
10	ATCCTCACCA
11	ATCATCACAC
12	CTGG**A**A**G**AC**T**
13	CCAG

3. Jumlah *String Matching*

   Jumlah ini dihitung dengan menjumlahkan kelompok yang matching dari proses *String Matching*.
4. Perhitungan akurasi dilakukan dengan membandingkan jumlah *string matching* dengan *minimum support*. Jika jumlah *string matching* lebih dari

minimum support maka motif tersebut dianggap sesuai dan diberikan nilai 1 dan jika sebaliknya diberi nilai 0. Jumlah nilai 1 dari suatu motif yang diujikan pada data berbeda dibandingkan dengan jumlah pengujian data inilah yang digunakan untuk menentukan akurasi motif yang dihasilkan.

Contoh :

Minimum_support = 8

Data *Training* : <TT><AGT><ACT>

Hasil *Testing* :

Uji Coba\ID	**1**	**2**	**3**
1	11	11	11
2	11	10	11
3	10	11	11
4	11	11	11
5	9	9	11

Akurasi :

Karena support dari setiap ujicoba untuk semua ID melebihi minimum support maka.

ID 1 = 5/5 *100% = 100%

ID 2 = 5/5 *100% = 100%

ID 3 = 5/5 *100% = 100%

ID 4 = 5/5 *100% = 100%

ID 5 = 5/5 *100% = 100%

Catatan : Nilai 5 didapatkan karena dalam 5 kali uji coba semua nilai support dari setiap ID diatas minimum support

Proses uji coba akurasi dilakukan pada *minimum support* 4 yang mewakili *minimum support* yang menghasilkan rata-rata terbaik dan minimum support 8 yang mewakili minimum support yang mampu menghasilkan lift rasio tertinggi. Kombinasi jumlah *data training* yang diujikan berbeda-beda mulai dari 5, 10, dan 15. Sedangkan jumlah *data testing* dibuat sama yaitu 5 dengan data dari individu

ke 16 hingga ke-20. Dan setelah dilakukan uji coba seperti terdapat pada Tabel 5.2 didapatkan secara rata-rata akurasi motif yang dihasilkan mencapai 100%. Dalam hal ini perubahan *minimum support* dan jumlah *data training* tidak berpengaruh pada akurasi yang terjadi. Didapatkannya akurasi yang mencapai 100 % karena pengujian dilakukan pada data dengan Exon yang sama dan sama-sama penderita kanker payudara serta jumlah data yang diujikan kurang bervariasi dan kurang banyak. Data pada exon yang sama cenderung memiliki pola yang hampir sama. Hal ini menyebabkan dalam melakukan proses ujicoba akurasi, akurasi mampu mencapai 100%. Dan alasan tidak dilakukannya pengujian pada exon yang berbeda karena pengujian model tersebut tidak sesuai. Setiap exon mempunyai urutan sekuen yang berbeda.

BAB VI
PENUTUP

6.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil uji coba yang telah dilkukan dalam penelitian skripsi ini adalah sebagai berikut.

1. Algoritma *prefixspan* dapat diterapkan dengan cukup baik pada 20 data DNA penderita kanker payudara untuk ditemukannya motif sekuens DNA penyebab kanker payudara yang mana melalui beberapa tahapan yaitu pembacaan data, pemrosesan algoritma *prefixspan*, proses penentuan *subsequence* motif, proses perhitungan lift rasio, dan proses penentuan motif yang diakhiri dengan *testing* terhadap motif yang dihasilkan.
2. Kesimpulan yang dapat diambil dari hasil uji coba yang dilakukan terhadap *minimum support* untuk mengetahui performansi motif yang dihasilkan adalah bahwa semakin besar nilai minimum support maka semakin kecil nilai lift rasio yang dihasilkan dalam uji coba pada 20 data. Pembatasan pada 20 data dikarenakan hasil dari lift rasio yang masih tidak stabil dan dimungkinkan kesimpulan akan berubah jika uji coba dilakukan lebih dari 20 data. Hasil yang tidak stabil ini dimungkinkan karena penggunaan nilai *length* yang tidak mengambil nilai length terbaik yaitu 2 sepeti hasil uji coba keempat.
3. Penggunaan support untuk menentukan performansi motif dianggap kurang sesuai karena dihasilkan nilai support yang berbanding terbalik dengan lift rasio. Sedangkan akurasi yang dihasilkan dalam uji coba ini mencapai rata-rata 100% karena pengujian dilakukan pada data dengan exon yang sama dan sama-sama penderita kanker payudara serta jumlah data yang diujikan kurang bervariasi dan kurang banyak .

6.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian skripsi ini adalah:

1. Data yang digunakan dalam pencarian motif sekuens DNA agar diperbanyak untuk didapatkan hasil kesimpulan yang lebih baik.
2. Proses uji coba yang dilakukan agar lebih bervariasi agar didapatkan hasil yang lebih baik

DAFTAR PUSTAKA

[ARD-13] Ardiansyah, Rikanda. Marji. Muflikhah, Lailil. 2013, *Sequential Pattern Mining pada Data Transaksi Penjualan menggunakan Algoritma Sequential Pattern Discovery Using Equivalent Classes (SPADE)*, Jurnal Tugas Akhir, Universitas Brawijaya, Malang.

[FOM-11] Fomby, Tom. 2011, *Association Rules (Aka Affinity Analysis or Market Basket Analysis*, Departemen of Economics Southern Methodist University Dallas, Texas.

[GOL-12] Gunadi, Goldi. & Sensuse, Dana Indra. 2012, *Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Prodeuk Buku dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-Growth) : Studi Kasus Percetakan PT. Gramedia*, Jurnal Telematika MKOM Vol. 4, No.1.

[HAN-06] Han, J. & Kamber, M. 2006, *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers.

[LIN-06] Ling, Bai. & Wei, Guo Zhu. 2006, *Journal of Cancer Molecules 2*, No.4, hal. 141-153, Guilin, China.

[PAR-07] Parthasarathy, S. 2007, *Building Genetic Medicine : technology, breast cancer, and the comparative politics of health care*, The MIT Press Cambridge, Massachusetts London.

[PEI-01] Pei, J. Han, J. Mortazavi-Asl, J. Pinto, H. Chen, Q. Dayal, U. Hsu, M. 2001, *PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth*, 17th International Conference on Data Engineering (ICDE).

[PON-03] Ponder, Bruce A. J. Henderson, Brian E. Ross, Ronald K. 2003, *Hormones, Genes, and Cancer*, Oxford University Press, Oxford.

[PUS-96] Pustai, L. Lewis, C. & Yap, E. 1996, *Cell Proliferation in Cancer-Regulation Mechanisms of Neoplastic Cell Growth*, Oxford University Press, Oxford.

[RAM-07] Ramadan, Riza. 2007, *Strategi Implementasi Peningkatan wakt Proses Algoritma PrefixSpan untuk Sequential Pattern Mining*, Undergraduated Theses, Insitut Teknologi Bandung, Bandung.

[SAN-01] Sander, C. 2001, *Bioinformatcs challenges in 2001*, Bioinformatics, No.17, hal. 1-2.

[SAP-06] Saputra, Danny. & Soelaiman, Rully. 2006, *Analisis Kinerja Algoritma Prefixspan dan AprioriAll pada Penggalian Pola Sekuensial. Undergraduated Theses*, Insitut Teknologi Sepuluh Nopember, Surabaya.

[TAN-06] Tan, Pan-Ning. Steinbach, Michael. & Kumar,Vipin. 2006. *Introduction to Data Mining*, Pearson Addison-Wesley, New York.

[THI-11] Thierry, Soussi. 2011, *TP53 Mutations in Human Cancer: Database Reassessment and Prospects for the Next Decade*, Adv Cancer Res, No.110, hal. 107-139.