# IMPLEMENTASI METODE WU-MANBER BERDASARKAN *MULTI PATTERN MATCHING* DALAM PENCARIAN KESAMAAN DNA (Studi kasus : DNA Kanker Hati)

Bella Befica Putri[1], Ernawati[2], Diyah Puspitaningrum[3]

[1,2,3]Program Studi Teknik Infomatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

[1]befica.bella@yahoo.com,
[2]w_ier_na@yahoo.com,
[3]diyahpuspitaningrum@gmail.com

*Abstrak:*Pencarian kesamaan DNA berdasarkan *single pattern matching* dinilai kurang efektif apabila diterapkan untuk mencari banyak pola. Dengan demikian diperlukan aplikasi pencarian kesamaan DNA yang dapat memecahkan pencarian banyak pola dengan algoritma *multi pattern matching* sehingga dapat menghasilkan kecocokan DNA dengan waktu yang lebih efisien dan efektif. Pada riset ini akan dibangun aplikasi pencarian kesamaan DNA kanker hati menggunakan metode Wu-Manber. Aplikasi pencarian kesamaan DNA kanker hati ini diharapkan dapat memberikan kemudahan dalam menampilkan tingkat kesamaan dari dua buah sekuen DNA kanker hati. Masukan yang digunakan pada penelitian ini adalah barisan DNA yang di peroleh dari *National Center for Biotechnology Information*(*www.ncbi.nlm.nih.gov*)dan memiliki keluaran berupa persentase kesamaan serta eksekusi proses. Dari pengujian yang telah dilakukan, persentase kesamaan yang diperoleh dapat mencapai 85% pada pencarian kesamaan *sequence* AB073612 dengan *sequence* AY080393. Semakin besar jumlah pola yang ditemukan dan semakin besar *minimum length* serta semakin pendek karakter sekuen pola, maka persentase kesamaan akan semakin besar.

*Kata Kunci:* DNA kanker hati, Java, *multi pattern matching*, *sequence alignment*, *Wu-Manber.*

***Abstract:*Searching of DNA similarity based on "single pattern matching" is assessed less effective if it used to find many patterns. So, it needed application to searching of DNA similarity can be used in problem of "multi pattern matching" to compatible result of DNA more efficient and affective. At this research would be built similarity search applications liver cancer DNA using a method of Wu-Manber. Liver cancer is one of cancer that**

**occurred from liver and usually attack man more. This application of searching DNA similarity liver cancer is expected that give amenity in present of similarity level from two DNA sequence of liver cancer. The suggestion of this research is DNA sequence that obtainedfrom *National Center for Biotechnology Information* (*www.ncbi.nlm.nih.gov*) and has output similarity percentage and the time duration process. From the testing that has been done, the percentage of similarity can reach 85% in the similarity searchof AB073612 sequence with AY080393 sequence.The greater number of patterns was found and the greater minimum length, and the shorter character sequence pattern, then the similiraty percentage will be greater.**

**Keywords:DNA liver cancer, Java, multi pattern matching, sequence alignment, Wu-Manber.**

## I. PENDAHULUAN

Bioinformatika adalah gabungan antara ilmu Biologi dan ilmu Teknik Informatika (TI).Pada umumnya, bioinformatika didefinisikan sebagai aplikasi dari alat komputasi dan analisa untuk menangkap dan menginterpretasikan data-data biologi [9].

Salah satu data biologi yang sering dibahas adalah DNA (*Deoxyribonucleid Acid*).DNA tersusun dari molekul-molekul *nukleotida*.*Nukleotida*penyusun DNA terdiri dari 4 macam, yaitu G (*Guanine*), A (*Alanine*), T (*Thymine*), dan C (*Cytosine*) [1].

DNA yang membawa informasi mengenai kondisi tubuh makhluk hidup dapat digunakan untuk mendeteksi penyakit, salah satunya yaitu dengan tes genetik. Tes genetik adalah proses membaca kode DNA untuk mengidentifikasi adanya bagian yang tidak normal dengan didukung oleh perkembangan teknologi sekuen DNA yang membawa perkembangan dalam ilmu komputasi DNA [4]. Salah satu penyakit yang dapat dideteksi melalui tes genetik adalah penyakit kanker hati.

Kanker hati adalah suatu kanker yang timbul dari hati dan lebih banyak menyerang kaum lelaki dibandingkan kaum wanita.Penyakit kanker hati adalah kanker keenam yang paling umum di dunia, dan jenis kanker ketiga penyebab kematian akibat penyakit kanker di Asia Pasifik dan jenis kanker ketiga penyebab kematian secara global.Prevalensi kanker ini di Indonesia tergolong tinggi mencapai 10 sampai 15 per 100 ribu penduduk/tahun [8].

Untuk mengidentifikasi tingkat kemiripan DNA dan atau pencocokan *string*, terdapat beberapa metode algoritma yang dapat digunakan, diantaranya yaitu Aho-Corasick, Boyer-Moore, Wu-Manber, dan lain-lain.

Salah satu algoritma yang dapat menyelesaikan problema *multi pattern matching* adalah algoritma Aho Corasick.Algoritma Aho Corasick merupakan algoritma berwaktu linier yang didasari pada pendekakan automata dan hanya dapat bekerja optimal pada kasus terburuk (*worst case*).Dalam segi efisiensi waktu, algoritma Aho Corasick berbanding terbalik dengan algoritma Boyer Moore. Algoritma Boyer Moore merupakan salah satu algoritma *single pattern matching*. Algoritma ini beranggapan bahwa terdapat kemungkinan untuk mengabaikan sekumpulan teks dengan porsi besar pada saat pencarian, sehingga mampu memberikan waktu eksekusi yang lebih cepat daripada algoritma linier pada kasus rata-rata (*average case*).

Sekitar bulan Mei tahun 1994, Sun Wu dan Udi Manber melakukan pengembangan ide dari Boyer Moore tersebut dengan menerapkan problema *multi pattern matching*, sehingga dapat mendukung banyak pola.Problema *multi pattern matching* muncul karena algoritma *single pattern matching* atau *string matching* biasa tidak efisien apabila diterapkan untuk mencari beberapa pola sekaligus.Dengan digunakannya algoritma *multi pattern matching* pada pencarian kesamaan pola, maka hasil yang di peroleh akan jauh lebih mendekati tingkat kesamaan yang akurat jika dibandingkan dengan algoritma *single pattern matching* atau *string matching* biasa.Desain dari algoritma Wu-Manber berkonsentrasi kepada pencarian biasa bukan kepada keadaan kasus terburuk (*worst case*), sehingga diyakini bahwa algoritma ini menjadi jauh lebih cepat daripada algoritma lainnya pada prakteknya [10].

Berdasarkan uraian diatas, maka penulis tertarik untuk membuat aplikasi yang mengimplementasikan metode Wu-Manber berdasarkan *multi pattern matching* dalam pencarian kesamaan DNA dengan menggunakan studi kasus DNA kanker hati.

## II. LANDASAN TEORI

### *A. Bioinformatika*

Bioinformatika sesuai dengan asal katanya yaitu “bio” dan “informatika” adalah gabungan antara ilmu biologi dan ilmu teknik informatika.Pada umumnya, bioinformatika didefinisikan sebagai aplikasi dari alat komputasi dan analisa untuk menangkap dan menginterpretasikan data-data biologi.

Ilmu bioinformatika lahir atas insiatif para ahli ilmu komputer berdasarkan Kecerdasan Buatan atau *Artificial Intelligence*.Mereka berpikir bahwa semua gejala yang ada di alam ini bisa dibuat secara *artificial* melalui simulasi dari gejala-gejala tersebut.Untuk mewujudkan hal ini diperlukan data-data yang menjadi kunci penentu tindak-tanduk gejala alam tersebut, yaitu gen yang meliputi DNA atau RNA [9].

### *B. DNA*

DNA (*Deoxyribose Nucleic Acid*) adalah asam nukleotida, biasanya dalam bentuk heliks ganda yang mengandung instruksi genetik yang menentukan perkembangan biologis dari seluruh bentuk kehidupan sel. DNA seringkali dirujuk sebagai molekul hereditas karena ia bertanggung jawab untuk penurunan sifat genetika dari kebanyakan ciri yang diwariskan. Pada manusia, ciri-ciri ini misalnya dari warna rambut hingga kerentanan terhadap penyakit.

DNA bukanlah suatu molekul tunggal, melainkan sepasang molekul yang digandeng oleh ikatan hidrogen. Masing-masing untai DNA adalah rantai kimia molekul penyusun, yakni nukleotida, yang terdiri dari empat tipe yaitu *Adenine* (A), *Cytosine* (C), *Guanine* (G) dan *Thymine* (T) [1].

### *C. Kanker Hati*

Kanker hati adalah suatu kanker yang timbul dari hati dan lebih banyak menyerang kaum lelaki dibandingkan kaum wanita.Penyakit kanker hati adalah kanker keenam yang paling umum di dunia, dan jenis kanker ketiga penyebab kematian akibat penyakit kanker di Asia Pasifik dan jenis kanker ketiga penyebab kematian secara global.Prevalensi

kanker ini di Indonesia tergolong tinggi mencapai 10 sampai 15 per 100 ribu penduduk/tahun [8].

Kanker hati atau *livercancer* adalah kanker yang berasal dari sel hati (kanker hati primer). Akan tetapi, ada juga kanker yang terjadi pada hati dimulai dari bagian tubuh lain (seperti usus besar, paru-paru atau kanker payudara) kemudian menyebar ke hati. Dokter menyebut ini kanker metastatik (kanker hati sekunder).

Kanker hati primer yang berasal dari sel hati terbagi dalam beberapa tipe, antara lain [2]:

1. *Hepatocellular carcinoma* (HCC). Kanker hati yang paling umum terjadi pada anak-anak dan orang dewasa. Kanker ini dimulai dari *hepatocytes* yang merupakan tipe utama sel hati.
2. *Cholangiocarcinoma*. Kanker ini berasal dari saluran kantung empedu.
3. *Hepatoblastoma*. Ini adalah tipe kanker langka yang menyerang anak-anak berusia 4 tahun ke bawah. Tipe kanker ini banyak yang berhasil disembuhkan.
4. *Angiosarcoma* dan *hemangiosarcoma*. Tipe kanker langka ini dimulai di pembuluh darah di hati dan tumbuh dengan sangat cepat.

*D. Algoritma Wu-Manber*

Algoritma Sun Wu dan Udi Manber dipublikasikan pada bulan Mei 1994. Metode Wu-Manber tercipta ketika Sun Wu dan Udi Manber melakukan pengembangan ide dari Boyer-Moore yang digunakan untuk problema *multi-pattern matching*.

Proses operasional algoritma Wu-Manber mencakup dua tahap, yaitu[10]:

1. *Preprocessing*

Tahapan pertama adalah sebuah tahapan proses persiapan (*preprocessing*) kumpulan pola. Tiga buah tabel dibangun pada tahapan *preprocessing* ini, sebuah tabel *Shift*, *Hash*, dan *Prefix*. Asumsikan bahwa, *B* = *Block Size*, *B* = 2 atau *B* = 3 ; Teks "*T = t1, t2, ..., tn*", "*n*" adalah panjang dari "*T*" ; *p* = pola asli, *k* = panjang pola, *m* = *minimum length* dari pola, Pola set *P = k[p] – m*, maka "*P = {P1, P2, ..., Pj}*", "*j*" adalah banyak pola dari "*P*".

a. Pembentukan tabel *Shift*

Diasumsikan bahwa, kunci *shiftX* = $x_1$ ... $x_s$adalah *s* buah karakter sepanjang B pada teks DNA yang akan dibaca ; "*X*" dipetakan ke Posisi "i" dalam tabel *Shift*, yaitu *hash* (*X)* = i. Maka terdapat dua kasus, yaitu:

1) *X* tidak ditemukan sebagai sebuah *substring* pada sembarang pola *P*.

   Pada kasus ini, dapat dilakukan pergeseran *m – B + 1* buah karakter pada teks. Sembarang pergeseran yang lebih kecil akan membandingkan *B* buah karakter terakhir dari teks dengan sebuah *substring* dari salah satu pola yang merupakan sebuah ketidakcocokan. Nilai *m – B + 1* ini disimpan sebagai *shift*[i].

2) *X* ditemukan pada beberapa pola.

   Pada kasus ini, ditemukan kemunculan *X* pada beberapa pola paling kanan, misalkan *q* adalah nilai indeks paling besar dari *X* maka diasumsikan bahwa $X_{[s]}$ berada pada posisi *q* dari $P_{[j]}$ dan tidak lebih besar dari *q* pada pola lainnya *Pj*, maka nilai $X_{[s]} = m - q$ disimpan sebagai *shift*[i].

Atau dapat dirumuskan,

$$Shift[i] = \begin{cases} m - B + 1, \text{ jika "}X\text{" tidak ditemukan pada pola string } P \\ \min\{m - q \mid X_{[s]} [P_{[j]}], P_j \in P\}, \text{ jika "}X\text{" ditemukan pada pola string } P \end{cases}$$

Dalam persamaan di atas, "*q*" adalah indeks paling kanan atau paling besardari "*X*" pada pola.

b. Pembentukan tabel *Hash*

Selama nilai pergeseran lebih besar daripada 0, maka pergeseran dapat dilakukan dan proses pembacaan dapat dilanjutkan. Ketika nilai pergeseran bernilai sama dengan 0 maka terdapat kemungkinan bahwa *substring* yang bersangkutan pada teks cocok dengan beberapa pola pada daftar pola. Untuk mencegah perbandingan *substring* dengan setiap pola pada daftar pola, digunakan sebuah teknik *hash* untuk meminimumkan jumlah pola yang akan dibandingkan. Sebelumnya telah dihitung sebuah pemetaan dari *B* buah karakter ke sebuah integer yang akan digunakan sebagai indeks ke tabel *Shift*. Nilai integer yang sama juga digunakan sebagai indeks ke tabel lainnya, yang disebut *hash*. Entri ke-i dari tabel *Hash*, *hash* [i], terdiri dari sebuah penunjuk (*pointer*) ke sebuah daftar pola dimana B buah karakter di-*hash* ke i. Tabel *Hash* secara khusus akan sangat jarang, karena tabel ini hanya akan menyimpan pola sementara dari tabel *Shift* yang menyimpan semua kemungkinan *string* dengan ukuran B. Hal ini merupakan penggunaan memori yang kurang efisien, tetapi terdapat kemungkinan untuk menggunakan kembali fungsi *hash* (pemetaan), sehingga dapat menghemat banyak waktu.

Anggap h adalah nilai *hash* dari *suffix* pada teks dan asumsikan *shift*[h] = 0. Nilai dari *hash*[h] adalah sebuah penunjuk p yang akan menunjuk ke dua tabel terpisah pada waktu yang sama, sebuah daftar dari penunjuk ke pola, PAT_POINT, diurutkan berdasarkan nilai *hash* dari *B* buah karakter terakhir dari setiap pola. Penunjuk p menunjuk ke awal dari daftar pola dimana nilai *hash*-nya adalah sebesar h. Untuk menemukan akhir dari daftar ini, akan dilakukan penambahan terhadap penunjuk ini hingga nilainya sama dengan nilai pada *hash*[h+1], karena keseluruhan daftar diurutkan berdasarkan nilai *hash*. Jika *shift*[h] $\neq$ 0, maka *hash*[h] = *hash*[h+1], karena tidak ada pola yang memiliki *suffix* yang di-*hash* ke h. Sebagai tambahan, juga disimpan sebuah tabel yang disebut *prefix*.

c. Pembentukan tabel *Prefix*

*Suffix* tidak hanya akan sering muncul pada teks, tetapi juga mungkin muncul pada beberapa pola. Hal ini akan menyebabkan tabrakan pada tabel *Hash*, yaitu semua pola dengan *suffix* yang sama akan dipetakan ke entri yang sama pada tabel *Hash*. Ketika menemukan *suffix* yang bersangkutan pada teks, akan ditemukan bahwa nilai *shift*-nya sebesar 0 (asumsi *suffix* dari pola tertentu) dan dicoba secara terpisah, semua pola dengan *suffix* ini untuk melihat apakah mereka cocok dengan teks. Untuk mempercepat proses ini, diperkenalkan tabel lainya yang disebut *prefix*.

Sebagai tambahan untuk memetakan *B* buah karakter terakhir dari semua pola, juga dipetakan *B'* buah karakter pertama dari semua pola ke tabel *Prefix*. Para penemu algoritma menyarankan bahwa nilai *B'* = 2 merupakan nilai yang bagus. Ketika ditemukan sebuah *shift* bernilai 0, dan akan menuju ke tabel *Hash* untuk menentukan apakah terdapat sebuah kecocokan, maka akan dicek nilai pada tabel *Prefix* terlebih dahulu. Untuk setiap *suffix*, tabel *Hash* tidak hanya mencakup daftar dari semua pola dengan *suffix* ini, tetapi juga mencakup nilai *hash* dari prefiksnya. Prefiks yang bersangkutan pada teks akan dihitung dengan menggeser m -*B'* buah karakter ke kiri dan menggunakannya untuk memfilter pola yang memiliki *suffix* yang sama tetapi prefiksnya berbeda. Ini merupakan metode pemfilteran yang

efektif karena terdapat sedikit kemungkinan untuk memiliki pola berbeda yang menggunakan *prefix* dan *suffix* yang sama. Hal ini juga merupakan tindakan penyeimbangan yang bagus dimana kerja tambahan pada perhitungan fungsi *hash* dari prefiks hanya dilakukan jika nilai *shift* sebesar 0, dimana hal ini hanya akan muncul ketika terdapat banyak pola dan terdapat banyak kesamaan.

2. Tahapan *Scanning*

Tahapan *scanning* pada algoritma Wu-Manber adalah sebagai berikut :

1. Ambil posisi tempat pointer berhenti sebanyak *m* buah karakter teks (disebut dengan istilah *Match Window*).
2. Ambil B buah karakter *suffix* dari *Match Window*.
3. Ambil nilai dari tabel *Shift* sesuai dengan *suffix*.
4. Jika nilai *shift* tidak ditemukan, maka set nilai *shift* = *m – B + 1* ke kanan.
5. Jika nilai *shift* = 0, maka cari posisi pada daftar tabel *Hash* yang merupakan kemungkinan terdapat teks *Match Window*.
6. Ambil *B* buah karakter *prefix* (*B'*) dari teks *Match Window* dengan (*m - B'*) karakter ke kiri.
7. Jika *B' = prefix*[pola], maka periksa pola aktual pada daftar tabel *Hash*.
8. Jika pola aktual pada daftar tabel *Hash* ditemukan cocok, maka simpan teks yang cocok.
9. Pointer + 1 karakter ke kanan.
10. Ulangi langkah 1 hingga proses selesai menelusuri seluruh teks.

### *E. Kelebihan Multi Pattern Matching dengan Single Pattern Matching dan String Matching Lainnya*

Pada pencocokan *string single pattern*, paket atau informasi yang dicari berdasarkan pola satu *string* saja. Sedangkan pada pencocokan *string multiple pattern*, paket atau informasi yang dicari berdasarkan beberapa susunan pola *string* [6].

Penerapan *multi pattern matching* pada pencarian kesamaan *string* dimaksudkan agar dapat mendukung banyak pola. Penerapan *multi pattern matching* muncul karena algoritma *single pattern matching* atau *string matching* biasa tidak efisien apabila diterapkan untuk mencari beberapa pola sekaligus.Penerapan *multi pattern matching* juga sering digunakan pada pencarian DNA dengan mentranslasikan sebuah pencarian rata-rata ke sebuah pencarian untuk *pattern* tertentu yang berjumlah besar [10].

Dengan digunakannya algoritma *multi pattern matching* pada pencarian kesamaan pola, maka hasil yang diperoleh akan jauh lebih mendekati tingkat kesamaan yang akurat jika dibandingkan dengan algoritma *single pattern matching* atau *string matching* biasa.

### *F. Metode Pengembangan Sistem Waterfall*

Model sekuensial linier (*waterfall*) merupakan salah satu dari metode yang digunakan untuk pengembangan sistem.Metode pengembang sistem menggunakan model *waterfall*terdiri dari beberapa tahapan, yaitu analisis kebutuhan *software*, desain, pengkodean, pengujian, dan pemeliharaan [7].

### *G. Basis Data*

Basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang di desain untuk menemukan data yang dibutuhkan oleh sebuah organisasi.Di dalam basis data, semua data diintegrasikan dengan menghindari duplikasi data [3].

*H. Pemrograman Terstruktur*

Pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang berbagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program computer [7].

*I. Data Flow Diagram (DFD)*

MenurutYourdandan DeMarco, *DataFlowDiagram*(DFD) adalahalatpembuatan modelyangmemungkinkan profesional sistemuntuk menggambarkansistem sebagai suatujaringan proses fungsional yang dihubungkansatusamalaindenganalurdata,baik secaramanualmaupun komputerisasi [7].

*Data Flow Diagram* (DFD) terdiri dari 3 level, yaitu [7]:

a. Diagram Konteks **:**menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem.
b. Diagram Nol (*Diagram Level-1*) : merupakan pemecahan dari diagram Konteks ke diagram Nol. Di dalam diagram ini memuat penyimpanan data.
c. Diagram Rinci : merupakan diagram yang menguraikan proses apa yang ada dalam diagram Nol.

*J. Flowchart (Diagram Alir)*

Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem.*Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian [3].

## III. METODE PENELITIAN

*A. Jenis Penelitian*

Berdasarkan tujuan yang ingin dicapai, maka penelitian tentang implementasi algoritma Wu-Manber berdasarkan *multi-pattern matching* dalam pencarian kesamaan DNA kanker hati ini termasuk dalam jenis penelitian terapan (*appliedresearch*).Penelitian terapan adalah penelitian yang hasilnya digunakan untuk membuat suatu keputusan dalam rangka memecahkan persoalan atau menguji hipotesis.

Dalam penelitian ini, peneliti berusaha untuk membuat suatu aplikasi yang dapat mendukung dalam pengujian algoritma *multi-pattern matching* pada pencarian kesamaan DNA kanker hati dengan menggunakan metode Wu-Manber yang dapat menampilkan persentase tingkat kesamaan DNA yang dicari.

*B. Metode Pengumpulan Data*

Sumber data dalam sebuah penelitian dibedakan menjadi dua yaitu sumber data primer dan sumber data sekunder.Sumber primer adalah suatu objek atau dokumen *original*. Sedangkan sumber sekunder merupakan data yang dikumpulkan dari tangan kedua atau dari sumber lain yang telah tersedia sebelum penelitian dilakukan.

Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode studi pustaka dengan jenis data sekunder.Studi kepustakaan dilakukan dengan mengumpulkan data dan informasi yang digunakan sebagai acuan dalam pembuatan aplikasi ini, yaitu berupa buku-buku ilmiah, laporan penelitian, skripsi, jurnal, dan sumber-sumber tertulis lainnya yang berhubungan dengan pemahaman metode yang digunakan.

Jenis data yang akan digunakan adalah data sekunder dengan memperoleh data dari *NationalCenter for Biotechnology Information* (NCBI).NCBI merupakan salah satu bank data gen, protein, dan literatur khususnya dibidang kesehatan yang terlengkap dan diacu oleh para peneliti dunia [5].Situs NCBI dapat di akses pada:*www.ncbi.nlm.nih.gov*.

*C. Metode Pengembangan Sistem*

Metode pengembangan sistem yang digunakan dalam pembuatan aplikasi pencarian kesamaan DNA dalam tugas akhir ini adalah model *waterfall*. Adapun langkah-langkah yang dilakukan dalam pengembangan sistem ini secara garis besar adalah sebagai berikut :

1. Analisis Kebutuhan

   Adapun analisis kebutuhan aplikasi yang akan dibuat adalah sebagai berikut :

   a. Kebutuhan data masukan

      Data masukan yang dibutuhkan dalam aplikasi ini adalah nukleutida dari kanker hati yang digunakan dalam pencarian kesamaan DNA.

   b. Kebutuhan data keluaran

      Adapun data keluaran yang dibutuhkan adalah tingkat persentase kesamaan DNA kanker hati yang telah diinputkan.

   c. Kebutuhan antarmuka

      Kebutuhan antarmuka pada aplikasi adalah kemudahan dan kenyamanan pengguna saat mengakses aplikasi sesuai dengan permasalahan yang ada.

2. Perancangan Aplikasi

   Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui sistem yang ada untuk meningkatkan kinerja sistem itu sendiri, agar dapat memenuhi hasil yang diinginkan. Diagram yang digunakan dalam perancangan aplikasi ini adalah *Data Flow Diagram* (DFD).

3. Implementasi

   Dalam pembuatan aplikasi, tahap ini merupakan tahapan nyata dimana tahap desain secara teknis nantinya dikerjakan oleh penulis yaitu proses *coding*.

4. Pengujian

   Proses pengujian dilakukan dengan mengujicoba semua fungsi-fungsi *software* agar *software* bebas dari error, dan hasilnya sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Proses pengujian yang dilakukan pada aplikasi yang dibuat menggunakan dua algoritma pengujian yaitu *white box testing* dan *black box testing*.

5. Penggunaan dan Pemeliharaan

   Pemeliharaan aplikasi dapat berupa pengembangan fungsional.Pengembangan diperlukan ketika adanya perubahan dari eksternal pengguna seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

## IV. ANALISIS DAN PERANCANGAN

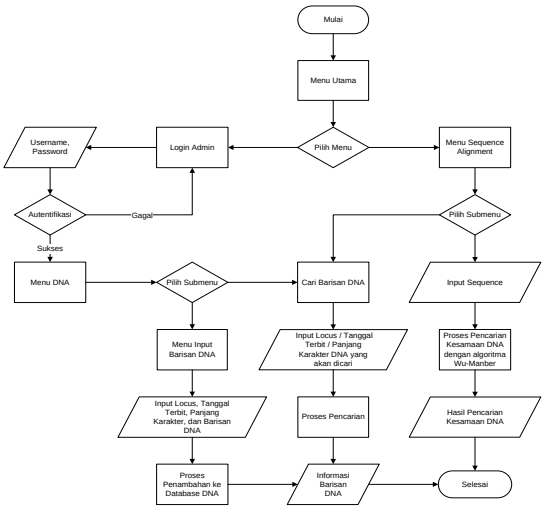
*A. Identifikasi Masalah*

Aplikasi pencarian kesamaan pola DNA yang telah ada saat ini pada umumnya menggunakan metode pencarian yang hanya mendukung problema *single pattern matching* atau *string matching* biasa. Penggunaan algoritma *single pattern matching* atau *string matching* biasa tidak efisien apabila diterapkan untuk mencari beberapa pola sekaligus dan seringkali terdapat kemungkinan untuk mengabaikan sekumpulan pola dengan porsi besar pada saat pencarian.

Pada sistem ini, aplikasi akan menggunakan metode yang dapat digunakan pada problema *multi pattern matching* yang dapat mendukung pencarian banyak pola, sehingga hasil yang diperoleh akan jauh lebih mendekati tingkat kesamaan yang akurat. Selain itu, algoritma *multi pattern matching* juga merupakan algoritma yang berkonsentrasi pada pencarian biasa daripada keadaan kasus terburuk (*worst case*), sehingga diyakini bahwa algoritma ini jauh lebih efisien daripada algoritma lainnya pada implementasinya.

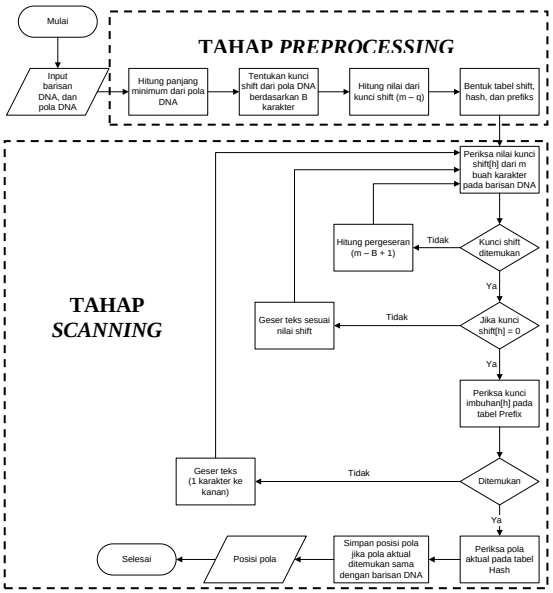
*B. Alur Kerja Sistem*

Perancangan alur kerja sistem atau *flowchart* merupakan langkah prosedur penyelesaian masalah yang diekspresikan dengan serangkaian simbol grafis yang baku dan lebih mudah digunakan, sehingga akan terhindar sedini mungkin timbulnya kesalahan interpretasi bagi orang lain terhadap suatu prosedur yang dikembangkan.Secara garis besar tahapan alur kerja sistem yang akan dibangun dapat dilihat pada Gambar 1.




Gambar 1. Diagram Alir Sistem

Tahapan alur algoritma Wu-Manber dapat dilihat pada Gambar 2.




Gambar 2. Diagram Alir Algoritma

*C. Analisis Kebutuhan*

Tujuan analisis kebutuhan adalah sebagai batasan dari sistem yang akan dibuat, menentukan kemampuan dan fungsi sistem sesuai dengan kebutuhan pengguna, dan fasilitas-fasilitas yang merupakan nilai tambah yang ada pada sistem yang dibangun.

1. Kebutuhan Fungsional

Analisis kebutuhan fungsional berarti melakukan analisis fungsi-fungsi fitur pada sistem yang akan dibangun. Berikut ini merupakan beberapa fitur yang ada pada sistem yang akan dibangun.

a. Mampu melakukan pencarian kesamaan DNA kanker hati dengan mengimplementasikan metode Wu-Manber berdasarkan *multi pattern matching*.
b. Mampu melakukan penambahan barisan DNA beserta hasil kesamaannya yang telah dicari pada *database* sistem.

2. Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional berarti melakukan analisis yang berhubungan dengan

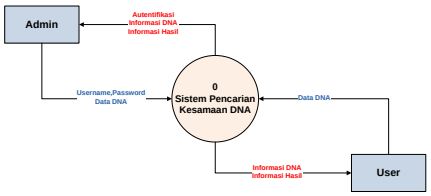
sistem diluar fitur-fitur yang akan dibangun dalam sistem berupa perangkat lunak dan perangkat keras yang mendukung dalam pembangunan sistem.

*D. Perancangan Data Flow Diagram*

Perancangan *Data Flow Diagram* (DFD) dalam penelitian ini dibuat dalam tiga bagian level, yaitu : diagram konteks, diagram level 1 dan diagram level 2.

1. Diagram Konteks

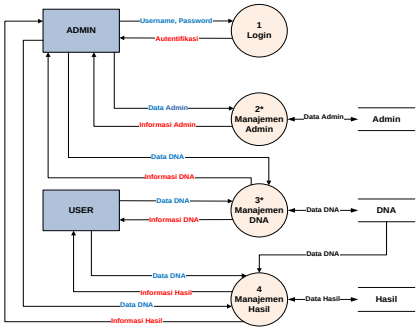
Diagram konteks merupakan diagram tertinggi yang ada di *Data Flow Diagram* (DFD). Tujuan dari pembuatan diagram konteks ini adalah memberikan pandangan tentang aplikasi yang akan dibangun secara umum. Diagram konteks dari aplikasi atau sistem yang akan dibangun, dapat dilihat pada Gambar 3 berikut ini.




Gambar 3. Diagram Konteks atau Diagram Level 0 Sistem Pencarian Kesamaan DNA

2. Diagram Level 1

Diagram level 1 merupakan pemecahan dari diagram konteks. Pada diagram level 1 ini terdapat 4 proses yang menggambarkan sistem pencarian kesamaan DNA yang akan dibangun. Diagram level 1 dari sistem pencarian kesamaan DNA ini dapat dilihat pada Gambar 4.



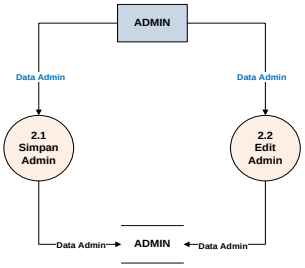

Gambar 4. Diagram Level 1 Sistem Pencarian Kesamaan DNA

3. Diagram Level 2

Diagram level 2 merupakan diagram pecahan dari proses yang ada pada diagram level 1, dimana proses-proses tersebut akan dijelaskan secara lebih rinci lagi. Berdasarkan diagram sebelumnya, maka terdapat dua buah proses yang akan dipecah secara rinci, yaitu proses 2 manajemen admin dan proses 3 manajemen DNA.

a. Diagram Level 2 Proses 2 Manajemen Admin

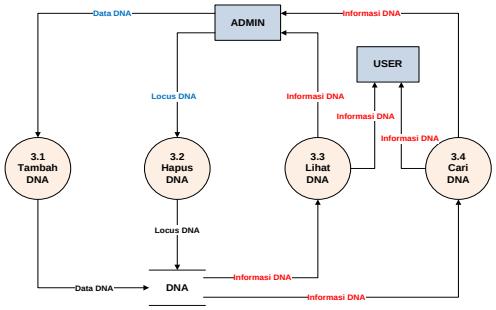
Pada diagram level 2 proses 2, manajemen Admin dipecah lagi menjadi 2 buah proses yang dapat dilihat pada Gambar 5 berikut.




Gambar 5. Diagram Level 2 Proses 2 Manajemen Admin

b. Diagram Level 2 Proses 3 Manajemen DNA

Diagram level 2 berikutnya adalah diagram level 2 proses 3 manajemen DNA. Proses manajemen DNA dipecah menjadi 4 buah proses yang dapat dilihat pada Gambar 6 berikut.




Gambar 6. Diagram Level 2 Proses 3 Manajemen DNA

## V. PEMBAHASAN

*A. Implementasi Antarmuka*

1. Halaman Utama

Halaman utama merupakan halaman yang pertama muncul ketika aplikasi dijalankan.Halaman utama terdiri dari menu DNA, Admin, dan *Help*.Masing-masing terdiri dari beberapa submenu.Menu DNA terdiri dari submenu Halaman DNA dan *Sequence Alignment* DNA.Menu Admin terdiri dari submenu *Login*, Halaman Admin, dan *Logout*. Menu *Help* terdiri dari submenu *Help Contents* dan *About*. Implementasi antarmuka halaman utama dari aplikasi pencarian DNA dapat dilihat pada Gambar 7.

Gambar 7. Halaman Utama

Aplikasi pencarian kesamaan DNA ini terdiri dari 2 jenis pengguna (*user)*, yaitu *user* biasa dan *user* admin.*User* admin memiliki hak akses penuh terhadap sistem dengan melakukan *login* terlebih dahulu. Sedangkan *user* biasa hanya dapat mengakses submenu *Sequence Alignment* DNA, *Help Contents*, dan *About*.

2. Halaman DNA

Halaman DNA hanya dapat diakses oleh admin. Halaman ini digunakan oleh admin dalam mengelola *database* DNA. Admin dapat menambah, menghapus, dan mengubah data DNA dengan menggunakan tombol yang telah disediakan yaitu tombol Tambah, Hapus, dan *Edit*.Implementasi antarmuka halaman DNA dapat dilihat pada Gambar 8.

Gambar 8. Halaman DNA

3. Halaman *Sequence Alignment* DNA

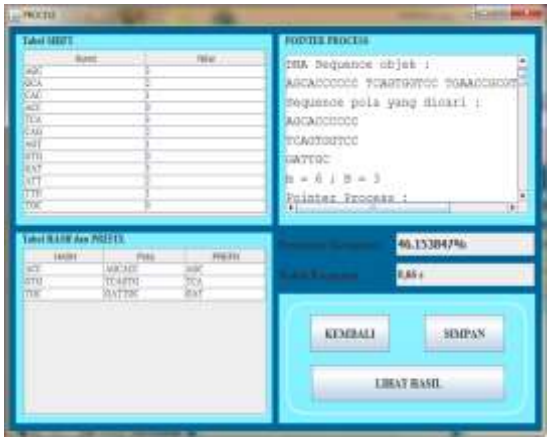
Halaman *Sequence Alignment* DNA merupakan halaman implementasi dari metode Wu-Manber.Pada halaman *Sequence Alignment* DNA ini, pengguna diminta untuk menginputkan *sequence* yang ingin pengguna cari kesamaannya, yaitu *sequence* DNA sebagai objek pencarian, dan *sequence* DNA pola sebagai *sequence* pembanding. Kedua *sequence* tersebut dapat pengguna pilih dari tabel DNA yang sama yang telah disediakan. Implementasi antarmuka *Sequence Alignment*DNA dapat dilihat pada Gambar 9.

Gambar 9. Halaman *Sequence Alignment* DNA

Pengguna wajib mengisi *Block Size* pada *combo box* yang telah disediakan.Pengguna dapat memilih 2 atau 3.*Block Size* merupakan nilai ketentuan yang berguna dalam proses pergeseran.

Pilih tombol Proses untuk memproses data yang telah pengguna inputkan untuk dicari kesamaannya.Ketika pengguna pilih tombol

Proses, maka akan muncul *JPanel Process*. Halaman proses *Sequence Alignment* DNA dapat dilihat pada Gambar 10.




Gambar 10. Halaman Proses *Sequence Alignment* DNA

Halaman proses menampilkan 3 buah bentukan tabel sebagai proses *preprocessing*, yaitu tabel *Shift*, *Hash*, dan *Prefix*. *Pointer Process* berfungsi untuk menampilkan tahap *scanning* yaitu proses pergeseran ketika dilakukan pencarian kesamaan DNA. Selain itu, halaman ini juga menampilkan persentase kesamaan dan waktu kecepatan / *runtime* sebagai *output* dari proses pencarian kesamaan DNA.

Pilih tombol Simpan untuk menyimpan hasil kesamaan DNA yang telah dicari pada database hasil.Kemudian pilih tombol Lihat Hasil untuk menampilkan *report* berupa tabel hasil. Tampilan *report* tabel hasil pencarian kesamaan DNA dapat dilihat pada Gambar 11.

Gambar 11. *Report* Tabel Hasil Pencarian Kesamaan DNA

4. Halaman *Login*

Halaman *Login* merupakan halaman yang digunakan khusus bagi admin untuk memperoleh hak akses penuh atas sistem. Pada halaman ini, admin diminta untuk menginputkan *username* dan *password*.Implementasi antarmuka halaman *Login* dapat dilihat pada Gambar 12.

Gambar 12. Halaman *Login*

5. Halaman Admin

Halaman Admin merupakan halaman yang digunakan admin untuk mengelola data dirinya sendiri seperti mengubah *username* dan atau *password*.Implementasi antarmuka halaman Admin dapat dilihat pada Gambar 13.

Gambar 13. Halaman Admin

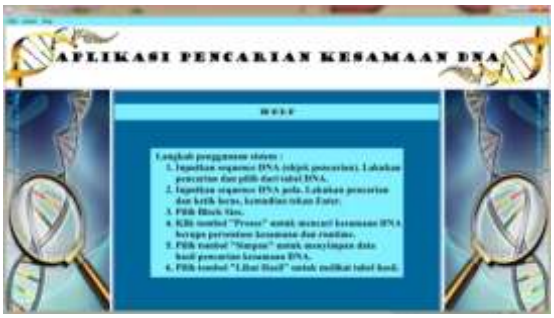
6. Submenu *Logout*

Submenu *Logout* berfungsi bagi admin yang ingin keluar dari hak aksesnya.

7. Halaman *Help Contents*

Halaman *Help Contents*berisi langkah-langkah penggunaan sistem, diharapkan dapat membantu pengguna agar lebih mudah menggunakan aplikasi pencarian kesamaan DNA ini, sehingga tidak

terjadi kekeliruan dalam penggunaannya.Implementasi antarmuka halaman *Help Contents* dapat dilihat pada Gambar 14.




Gambar 14. Halaman *Help*

8. Halaman *About*

Halaman *About*merupakan halaman yang dapat digunakan untuk mendeskripsikan aplikasi serta dapat berfungsi untuk mengetahui nama aplikasi dan karya pembuatnya. Implementasi antarmuka halaman *About*dapat dilihat pada Gambar 15.

Gambar 15. Halaman *About*

## VI. PENUTUP

### *A. Kesimpulan*

Berdasarkan analisis perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa:

1. Semakin besar jumlah pola yang ditemukan dan semakin besar *minimum length* serta semakin pendek karakter *sequence* pola, maka persentase kesamaan DNA yang dihasilkan akan semakin besar.
2. Semakin pendek karakter *sequence* DNA (objek pencarian) yang diujikan, maka waktu kecepatan / *runtime*akan semakin cepat.
3. Algoritma *multi pattern matching* pada metode Wu-Manber mampu diterapkan pada pencarian kesamaan DNA dengan contoh studi kasus DNA kanker hati, dimana pada pengujiannya dapat menghasilkan persentase mencapai 85% dalam pencarian kesamaan *sequence* AB073612 jenis kanker hati *Hepablastoma* dengan *sequence* AY080393 jenis kanker hati *Hepatocellular Carcinoma*.

### *B. Saran*

Berdasarkan analisis perancangan sistem, implementasi, dan pengujian sistem, maka untuk pengembangan penelitian selanjutnya penulis menyarankan sebagai berikut :

1. Aplikasi ini hanya dapat melakukan pencarian kesamaan dua buah *sequence*. Diharapkan pada penelitian Wu-Manber selanjutnya dapat melakukan pencarian kesamaan lebih dari dua *sequence* secara langsung.
2. Pada pengembangan sistem selanjutnya, diharapkan adanya aplikasi yang dapat mendeteksi apakah seorang manusia tersebut terjangkit penyakit kanker hati atau tidak yaitu membandingkan DNA manusia yang ingin dideteksi dengan DNA yang positif terjangkit penyakit kanker hati.
3. Pada aplikasi ini, data yang ada pada *database* diperoleh dari situs GenBank yang kemudian diinputkan dengan cara*copy-paste* oleh admin. Diharapkan untuk pengembangannya, aplikasi ini dapat langsung terhubung ke *database* DNA (GenBank), sehingga pengguna akan lebih

leluasa untuk mencari DNA yang ingin dicari kesamaannya.

## REFERENSI


[1] Hadi, M. (2008). *Soliton dan DNA*. [Online]. Tersedia:http://www.nano.lipi.go.id/. [diakses pada : 13 Desember 2014]

[2] Harjana, D. (2013). *Gejala Kanker Hati, Penyebab dan Cara Pencegahan*. [Online]. Tersedia:http://gejalapenyakitmu.blogspot.com/2013/08/gejala-kanker-hati-penyebab-dan-cara.html. [diakses pada : 25Januari 2015]

[3] Indrajani. (2011). *Perancangan Basis Data dalam All in 1*. Jakarta: PT Alex Media Komputindo.

[4] Irsan, I. C. (2014). Penggunaan Algoritma String Matching Untuk Mendeteksi Kanker.

[5] Miftakhunnafisah, W. (2012). Pengenalan NCBI Untuk Analisis, Protein dan Senyawa Kimia.

[6] Noprisson, H. (2013). Implementasi Algoritma Rabin Karp Untuk Menentukan Keterkaitan Antar Publikasi Penelitian Dosen tahun 2013 (Studi Kasus : Website Lembaga Penelitian Universitas Bengkulu).

[7] Rosa, & Salahuddin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.

[8] Usman, I. (2013). *Kanker Hati*. [Online]. Tersedia:http://ifhausman.blogspot.com/2013/06/kanker-hati.html. [diakses pada : 25Januari 2015]

[9] Utama, A. (2003). Peranan Bioinformatika dalam Dunia Kedokteran.

[10] Yudi, & Chandra, T. (2010). Simulasi Pencarian Pola Kata Pada File Teks Berdasarkan Multi Pattern Matching Dengan Metode Wu-Manber.