PENGEMBANGAN ALGORITMA SMITH WATERMAN MENGGUNAKAN PENDEKATAN METODE BISECTION DAN BACKTRACKING UNTUK MENGIDENTIFIKASI KEMIRIPAN PROTEIN

SKRIPSI

TASNIA AKIL

H13116301

PROGRAM STUDI ILMU KOMPUTER

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2020

PENGEMBANGAN ALGORITMA SMITH WATERMAN MENGGUNAKAN PENDEKATAN METODE BISECTION DAN BACKTRACKING UNTUK MENGIDENTIFIKASI KEMIRIPAN PROTEIN

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains pada Program Studi Ilmu Komputer Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

TASNIA AKIL

H13116301

PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

NOVEMBER 2020

LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

PENGEMBANGAN ALGORITMA SMITH WATERMAN MENGGUNAKAN PENDEKATAN METODE BISECTION DAN BACKTRACKING UNTUK MENGIDENTIFIKASI KEMIRIPAN PROTEIN

adalah benar hasil karya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 24 November 2020

Tasnia Akil
NIM. H13116301

PENGEMBANGAN ALGORITMA SMITH WATERMAN MENGGUNAKAN PENDEKATAN METODE BISECTION DAN BACKTRACKING UNTUK MENGIDENTIFIKASI KEMIRIPAN PROTEIN

Disetujui oleh:

Pembimbing Utama	**Pembimbing Pertama**
Dr. Eng. Armin Lawi, S.Si., M.Eng.	**Supri Bin Hj.Amir S.Si., M.Eng.**
NIP. 19720423 199512 1 001	**NIP. 19880504 201903 1 012**

Pada 24 November 2020

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama	: Tasnia Akil
NIM	: H13116301
Program Studi	: Ilmu Komputer
Judul Skripsi	: Pengembangan Algoritma Smith Waterman Menggunakan Pendekatan Metode *Bisection* dan *Backtracking* Untuk Mengidentifikasi Kemiripan Protein

Telah berhasil dipertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda Tangan
1. Ketua	: Dr.Eng. Armin Lawi, S.Si., M.Eng.	( )
2. Sekretaris	: Supri Bin Hj.Amir, S.Si., M.Eng.	( )
3. Anggota	: Dr. Muhammad Hasbi, M.Sc.	( )
4. Anggota	: Dr. Hendra, S.Si., M.Kom.	( )

Ditetapkan di : Makassar

Tanggal : 24 November 2020

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS HASANUDDIN DEPARTEMEN MATEMATIKA FAKULTAS MIPA

KATA PENGANTAR

Bismillahirrahmanirrahim, segala puji bagi Allah *Subhanahu Wa ta'ala*, Tuhan alam semesta yang telah memberikan nikmat kesempatan, kesehatan dan kemampuan sehingga penulisan skripsi ini bisa selesai. Shalawat serta salam senantiasa tercurah kepada *Rasulullah* Muhammad *Shallallahu Alaihi Wasallam* dan kepada para keluarga serta sahabat beliau, yang merupakan teladan dalam menjalankan kehidupan di dunia.

Alhamdulillah, skripsi dengan Judul **"Pengembangan Algoritman Smith Waterman Menggunakan Pendekatan Metode Bisection dan Backtracking untuk Mengidentifikasi Kemiripan Protein"** yang disusun sebagai salah satu syarat akademik untuk meraih gelar Sarjana Sains pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin ini dapat diselesaikan. Tentunya, dalam penulisan skripsi ini, penulis mampu menyelesaikan tepat waktu berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, ucapan terima kasih dan apresiasi yang tak terhingga kepada kedua orang tua penulis, Ayahanda **Ambo Ala** dan Ibunda **Rahma** yang tak kenal lelah dalam memanjatkan doa serta memberikan nasihat, motivasi dan dukungan kepada penulis untuk menggapai cita-cita sehingga penulis dapat menyelesaikan pendidikan di perguruan tinggi. Tugas akhir ini hanya setitik kebahagiaan kecil yang bisa penulis persembahkan. Juga kepada keluarga penulis, Nenek **Senabe** dan adik-adik penulis **Surya Nasrullah Akil**, **Naivah Akil**, dan **Azisah Syahra Akil**, terima kasih atas doa , dukungan dan semangatnya.

Terima kasih juga penulis ucapkan kepada:

1. Ibu Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu** beserta jajarannya. Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuna Alam (FMIPA), **Dr.Eng. Amiruddin** beserta jajarannya.
2. Bapak **Dr. Nurdin, S.Si., M.Si.** sebagai Ketua Departemen Matematika FMIPA Unhas. Bapak **Dr. Muhammad Hasbi, M.Sc** sebagai Ketua Program Studi Ilmu Komputer. Penulis juga berterima kasih atas dedikasi

dosen-dosen pengajar, serta Dapartemen atas ilmu dan bantuan yang bermanfaat.

3. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku pembimbing utama sekaligus ketua tim peguji, atas segala ilmu, nasehat dan kesabaran dalam membimbing penulis serta meluangkan waktu di sela-sela rutinitas yang begitu padat hingga skripsi ini dirampungkan.
4. Bapak **Supri Bin Hj. Amir, S.Si., M.Eng.** sebagai dosen pembimbing pertama sekaligus sekteraris tim penguji atas ilmu yang diberikan selama proses perkuliahan dan bimbingan, serta segala bentuk bantuan yang telah diberikan dalam penyusunan skripsi ini.
5. Bapak **Dr. Muhammad Hasbi, M.Sc** sebagai anggota tim penguji atas segala ilmu yang telah diberikan selama proses perkuliahan serta berbagai masukan dan kritik yang membangun dalam proses penyusunan skripsi ini.
6. Bapak **Dr. Hendra, S.Si., M.Si.** sebagai anggota tim penguji atas segala ilmu yang telah diberikan selama proses perkuliahan serta berbagai masukan dan kritik yang membangun dalam proses penyusunan skripsi ini.
7. Ibu **Nur Hilal A Syahrir, S.Si., M.Si.** yang pertamakali memperkenalkan bidang *Bioinformatika* disaat kebingungan mencari judul tugas akhir.
8. Saudara **Rio Mukhtarom** dan **Fatur Rahman**, yang telah meluangkan waktunya dalam membagi ilmu dan selalu memberi dukungan, motivasi dan semangat kepada penulis, terima kasih untuk semuanya.
9. Saudara – saudara ku **St. Hestiana Kadir, Rizka Syahfitri, Suci Rahmadana Anwar, Ainun Mardiyah Istiqamah, Berlian Adriani Putri, Nurmayulina, Marselia Ghanyyu Wahdini, , Nirwana Sari Hamka, Nisrina Syadza Dewanty, dan ILMU KOMPUTER 2016,** atas kebersamaan, kepedulian, suka-duka, canda tawa yang telah kita lewati selama ini.
10. Keluarga besar **ILMU KOMPUTER 2014, 2015, 2017, 2018, dan 2019.**
11. Keluarga besar **Fakultas MIPA** dan **HIMATIKA** terutama **A16ORITMA 2016** atas segala bentuk dukungan dan bantuan selama proses perkuliahan. Semoga kesuksesan selalu kita dapatkan dalam setiap langkah-langkah kita.

12. Saudari ku **kak Firsya Natasya Achmad dan Astuti Hardianti,** atas segala bentuk dukungan dan semangat serta kebersamaan bersama penulis selama ini.
13. Keluarga besar **KKN Tematik Luwu Utara** yang secara ikhlas dan tulus bersama penulis untuk mengabdi pada masyarakat.
14. Seluruh pihak yang tidak dapat disebutkan satu per satu atas segala bentuk konstribusi, partisipasi, serta motivasi yang diberikan kepada penulis selama ini. Semoga apa yang kita berikan, dilipatgandakan oleh Allah Subahana Wata'ala.

Semoga segala bantuan yang dengan tulus ditujukan kepada penulis mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa masih banyak kekurangan dalam tugas akhir ini, untuk itu dengan segala kerendahan hati penulis memohon maaf. Akhir kata, semoga tulisan ini memberikan manfaat kepada semua pihak yang membutuhkan dan terutama untuk penulis.

Makassar, 24 November 2020

Tasnia Akil

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama	: Tasnia Akil
NIM	: H13116301
Programa Studi	: Ilmu Komputer
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul:

"PENGEMBANGAN ALGORITMA SMITH WATERMAN MENGGUNAKAN PENDEKATAN METODE BISECTION DAN BACKTRACKING UNTUK MENGIDENTIFIKASI KEMIRIPAN PROTEIN"

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 24 November 2020

Yang menyatakan

(Tasnia Akil)

ABSTRAK

Protein merupakan polipeptida yang tersusun atas asam amino, urutan basa yang berbeda akan menghasilkan asam amino yang berbeda pada setiap protein. Penelitian mengenai kemiripan antar protein dibutuhkan sehingga dapat dibandingkan kemiripan asam amino yang menyusun protein. Salah satu pendekatan bioinformatika adalah perbandingan kedua sekuens dengan mensejajarkan sekuens. Dibutuhkan biaya yang mahal pada saat pensejajaran lokal terutama untuk waktu perhitungan apabila menggunakan data sekuens dengan ukuran yang besar. Pengukuran kemiripan antara urutan asam amino dapat dilakukan dengan mengimplementasikan pensejajaran lokal menggunakan Algoritma *Smith Waterman*. Dalam penelitian ini dilakukan pengembangan pada Algoritma *Smith Waterman* menggunakan pendekatan metode *Bisection* dan *Backtracking* untuk mensejajarkan urutan dan mengoptimalkan waktu eksekusi. Data yang digunakan pada penelitian ini adalah data protein target pada obat. Hasil penelitian menunjukkan bahwa semakin pendek sekuens asam amino yang dibandingkan maka semakin cepat waktu komputasi pada Algoritma *Smith Waterman* dan jika semakin panjang sekuens asam amino yang dibandingkan maka lebih cepat waktu komputasi pada Pengembangan Algoritma *Smith Waterman.*

Kata Kunci : Algoritma *Smith Waterman, Bisection, Backtracking,* Sekuens Asam Amino, Waktu Komputasi.

ABSTRACT

Proteins are polypeptides composed of amino acids, different base sequences will produce different amino acids in each protein. Research on the similarity between proteins is needed so that the similarity of the amino acids that make up proteins can be compared. One of the bioinformatics approaches is the comparison of the two sequences by aligning the sequences. It takes a high cost at the time of the local alignment, especially for computation time when using the sequence data with a large size. Measurement of similarity between amino acid sequences can be done by implementing a local alignment using the Smith Waterman algorithm. In this study, the development of the Smith Waterman Algorithm using the Bisection dan Backtracking method approach to aligning sequences and optimizing execution time. The data used in this study is data on the drug target protein. The results showed that the shorter the amino acid sequences were compared faster the computation time on the Smith Waterman Algorithm and if the longer the amino acid sequence is compared, faster the computation time in the Smith Waterman Algorithm Development.

Keywords : *Smith Waterman Algorithm, Bisection, Backtracking, amino acid sequence, computation time.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Bioinformatika merupakan ilmu perpaduan antara ilmu biologi dan ilmu informatika untuk penyimpanan data (*storage*), pencarian informasi (*retrieval*), manipulasi data, dan distribusi informasi yang direlasikan dengan makromolekul biologi, seperti DNA, RNA, dan protein (Xiong, 2006). Protein adalah makromolekul polipeptida yang tersusun dari sejumlah n-asam amino yang dihubungkan oleh ikatan peptida. Suatu molekul protein disusun oleh sejumlah asam amino dengan susunan tertentu dan bersifat turunan. Asam amino terdiri atas unsur-unsur karbon, hidrogen, oksigen, dan nitrogen. Pada tingkat selular, transkripsi DNA menjadi RNA pembawa pesan (mRNA) menghasilkan cetakan untuk sintesis protein di ribosom. Urutan basa yang berbeda pada mRNA akan menghasilkan asam amino yang berbeda pada setiap protein (Probosari, 2019).

Salah satu topik dalam bioinformatika adalah perbandingan kemiripan urutan (Nugroho & Kusuma, 2018). Pencarian basis data urutan dapat diterapkan untuk menemukan kesamaan antara urutan kueri (urutan yang akan dianalisis) dan urutan subjek (urutan terkenal) dalam *database* urutan. Informasi tersebut juga dapat digunakan untuk mengidentifikasi hubungan evolusi dalam suatu spesies (Bustamam, dkk., 2014). *Sequence alignment* (penjajaran barisan) adalah cara mengatur urutan deoxyribonucleic acid (DNA), ribonucleic acid (RNA), atau protein untuk mengidentifikasi daerah kesamaan yang mungkin menjadi konsekuensi dari hubungan fungsional, struktural, atau evolusi antara urutan (Singh, dkk., 2011).

Pada tahun 1970, Needleman dan Wuncsh memperkenalkan metode dengan pemrograman dinamis yang mensejajarkan urutan secara global yang dikenal dengan Algoritma *Needleman Wuncsh (NW)*. Kemudian pada tahun 1981, Smith dan Waterman mengembangkan metode pensejajaran urutan secara lokal yang dikenal dengan Algoritma *Smith Waterman (SW)* (Liu, dkk., 2015). Dalam pensejajaran urutan protein, Algoritma *Smith Waterman* membandingkan setiap

residu antar dua urutan dan menemukan area dengan tingkat kesamaan yang tinggi berdasarkan nilai kesamaan antara kedua urutan tertinggi.

Beberapa penelitian yang menggunakan urutan nukleotida atau asam amino protein juga telah dilakukan untuk menemukan skor kemiripan seperti yang telah dilakukan oleh Bu'ulölö dkk (2010) yaitu membangun sebuah *prototipe* aplikasi berbasis dekstop untuk melakukan penyelarasan urutan menggunakan Algoritma *Smith Waterman*. Penelitian lainnya dilakukan oleh Nugroho dkk (2018) untuk mengukur Kemiripan Protein pada Ijah Webserver dengan Algoritma *Smith Waterman* Berbasis Komputasi Paralel Cuda GPU. Budiman (2009) yang melakukan penelitian penyejajaran Lokal Barisan DNA dengan menggunakan metode Smith Waterman. Penelitian lainnya juga dilakukan oleh Himawan F.A (2013) yang juga membahas tentang penjajaran lokal sekuen DNA menggunakan Algoritma *Smith Waterman*.

Algoritma yang efisien adalah algoritma yang meminimumkan kebutuhan waktu dan ruang. Besaran yang digunakan untuk menjelaskan pengukuran waktu dan ruang adalah kompleksitas algoritma. Kompleksitas dari suatu algoritma adalah ukuran jumlah komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah (Azizah, 2013). Kompleksitas algoritma *Smith Waterman* adalah O(mn) di mana m dan n merupakan panjang dari kedua urutan yang dicari, sehingga membutuhkan biaya yang mahal terutama untuk waktu perhitungan (Liu, dkk., 2015). Dalam penelitian ini, akan dilakukan pengembangan pada Algoritma *Smith Waterman* menggunakan metode *bisection* dan *backtracking* untuk mengetahui skor kemiripan antar dua urutan dan mengoptimalkan waktu komputasi.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana mengembangkan Algoritma *Smith Waterman* untuk mengoptimalkan waktu komputasi pada pensejajaran urutan asam amino protein?
2. Bagaimana mengimplemantasikan Algoritma *Smith Waterman* dan pengembangan algoritma *Smith Waterman* untuk mencari kemiripan pada sekuens protein?

3. Bagaimana mengukur kinerja Algoritma *Smith Waterman* dan pengembangan Algoritma *Smith Waterman* berdasarkan tingkat kemiripan hasil algoritma dan waktu komputasi?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah :

1. Mengembangkan Algoritma *Smith Waterman* untuk mengoptimalkan waktu komputasi pada pensejajaran urutan dengan menggunakan metode *bisection* dan *backtracking*.
2. Mengetahui kemiripan pada sekuens protein menggunakan Algoritma *Smith Waterman* dan pengembangan Algoritma *Smith Waterman.*
3. Mengetahui kinerja Algoritma *Smith Waterman* dan pengembangan Algoritma *Smith Waterman* berdasarkan tingkat kemiripan hasil algoritma dan waktu komputasi.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat berguna untuk hal-hal berikut :

1. Pemanfaatan ilmu komputer dalam bidang Biologi terkhusus makromolekul dapat dimanfaatkan oleh peneliti lain sebagai referensi.
2. Mempermudah bidang lainnya seperti farmasi dalam menentukan protein target dari obat dan ilmu gizi dalam menentukan protein target dari makanan.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Penelitian ini hanya akan membahas tentang kemiripan antar urutan asam amino dari protein.
2. Data yang digunakan berasal dari protein target pada obat di *website DrugBank*.
3. Skor kemiripan yang diperoleh hanya berlaku untuk data yang digunakan pada penelitian ini.

1.6 Organisasi Skripsi

Organisasi pada penulisan skripsi ini adalah sebagai berikut :

Bab I Pendahuluan membahas mengenai latar belakang masalah dari penelitian yang akan dilakukan, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta organisasi skripsi.

Pada Bab II Tinjauan Pustaka membahas mengenai landasan teori yang merupakan konsep dasar yang mendasari pokok permasalahan dalam penulisan skripsi, *state of the art* yang menjadi acuan penulis dari penelitian sebelumnya dan kerangka konseptual dari skripsi.

Pada Bab III Metodologi Penelitian membahas tentang hal-hal untuk mencapai hasil dari penelitian, antara lain waktu dan tempat penelitian, sumber data, dan istrumen penelitian.dan tahapan penelitian.

Pada Bab IV Hasil dan Pembahasan penulis menjelaskan tentang hasil penelitian dan pembahasannya yang telah dicapai dari rumusan masalah.

Pada Bab V Kesimpulan dan Saran berisikan tentang kesimpulan dari penelitian dan saran yang berguna untuk penelitian lebih lanjut dari skripsi ini.

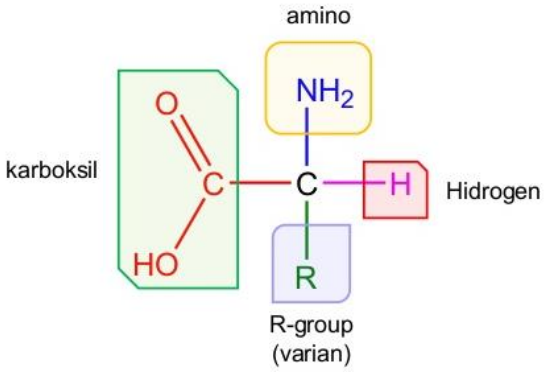
BAB II
TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Protein

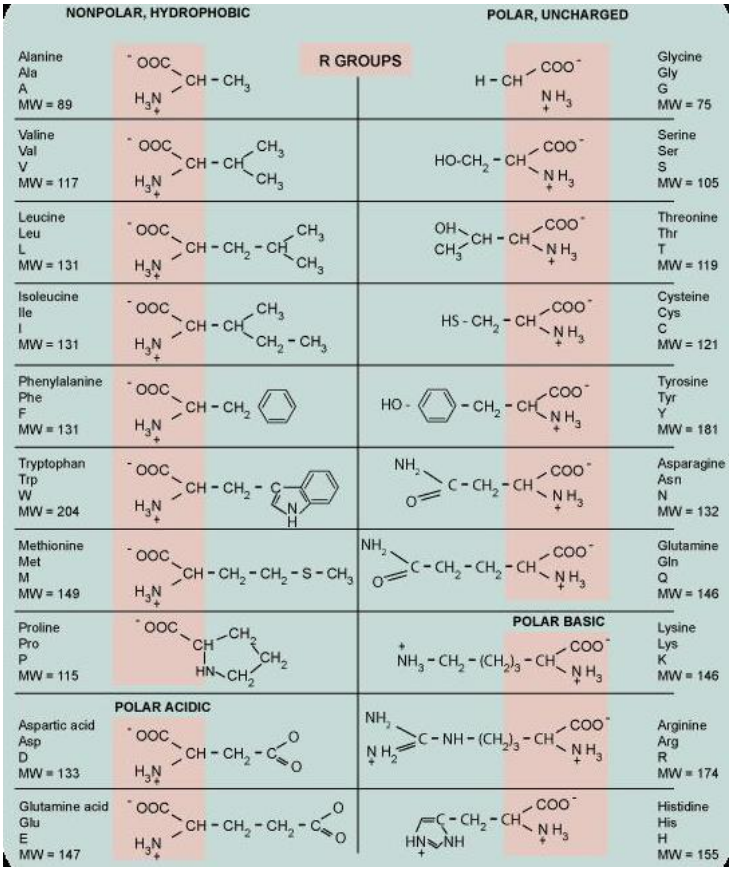
Protein (akar kata *protos* dari bahasa Yunani yang berarti "yang paling utama") adalah senyawa organik kompleks berbobot molekul tinggi yang merupakan polimer dari monomer-monomer asam amino yang dihubungkan satu sama lain dengan ikatan peptida. Molekul protein mengandung karbon, hidrogen, oksigen, nitrogen dan sulfur serta fosfor. Protein berperan penting dalam struktur dan fungsi semua sel makhluk hidup dan virus. Protein merupakan salah satu dari biomolekul raksasa, selain polisakarida, lipid, dan polinukleotida, yang merupakan penyusun utama makhluk hidup. Selain itu, protein merupakan salah satu molekul yang paling banyak diteliti dalam biokimia. Protein ditemukan oleh Jöns Jakob Berzelius pada tahun 1838 (Al-Mahmudattussa'adah, -).

Gambar 2.1 merupakan struktur asam amino yang secara umum adalah satu atom C yang mengikat empat gugus yaitu gugus amina (NH2), gugus karboksil (COOH), atom hidrogen (H), dan satu gugus sisa (R, dari *residue*) atau disebut juga gugus atau rantai samping yang membedakan satu asam amino dengan asam amino lainnya. Atom C pusat dinamai atom Cα (*C-alfa*) sesuai dengan penamaan senyawa bergugus karboksil, yaitu atom C yang berikatan langsung dengan gugus karboksil. Oleh karena gugus amina juga terikat pada atom Cα ini, senyawa tersebut merupakan *α-amino acid*.



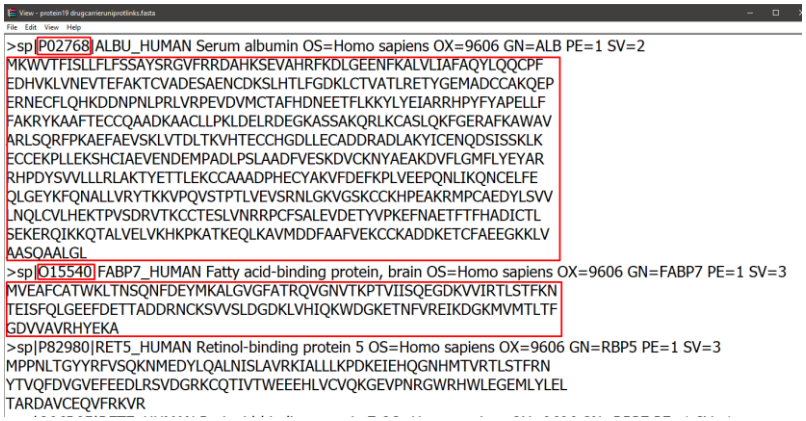
Gambar 2.1. Struktur Asam Amino

Semua organisme menggunakan 20 asam amino yang sama sebagai unit pembangun suatu molekul protein. Kedua puluh asam amino ini adalah asam amino normal yang terdapat pada protein alami. Asam amino pertama yang ditemukan adalah asparagin pada tahun 1806, sedangkan asam amino yang terakhir ditemukan adalah treonin yang belum teridentifikasi sampai tahun 1938. Protein alami terdiri dari kombinasi kedua puluh asam amino. Kedua puluh asam amino ini adalah *α-amino acid*. Sembilan belas dari dua puluh asam amino yang umumnya diisolasi dari protein alami mempunyai struktur umum yang sama dengan amina primer pada *α-carbon*. Asam amino lainnya adalah prolin yang merupakan amina sekunder (Azhar, 2016).



Gambar 2.2. 20 Jenis Asam Amino

Pada gambar 2.2 ditunjukkan 20 jenis asam amino, asam amino ini biasanya diklasifikasikan berdasarkan sifat kimia rantai samping tersebut menjadi empat kelompok. Rantai samping dapat membuat asam amino bersifat asam lemah, basa lemah, hidrofilik jika polar, dan hidrofobik jika nonpolar.



Gambar 2.3. Ilustrasi data sekuens Asam Amino
eksistensi .fasta

Ilustrasi pada data sekuens asam amino ada pada gambar 2.3, P02768 adalah Id Protein target, dan urutan huruf seperti MKWT...dst adalah sekuens asam amino dari id protein P02768.

2.1.2 Pensejajaran Sekuen Protein

Karakteristik Protein ditentukan oleh asam amino yang akan mengendalikan sifat protein. Setiap asam amino memiliki karakteristik yang khas sehingga terdapat perbedaan komposisi pada masing-masing asam amino. Perbedaan komposisi asam amino menggambarkan perbedaan sifat pada masing-masing protein dengan kecenderungan bahwa terdapat kemiripan sifat pada protein dengan kekerabatan yang besar. Kemiripan sifat antar protein disebabkan karena terdapat kandungan jenis asam amino yang sama pada protein yang berbeda (Arifin A. Y., 2004).

Pensejajaran protein dapat dilakukan antar dua barisan (*pairwise Alignment*) dan lebih dari dua barisan (*multiple Alignment*). Penyejajaran dua urutan asam amino adalah memberikan nilai kecocokan kedua urutan tersebut. Semakin tinggi nilainya, berarti kedua protein tersebut semakin mirip satu sama lain, dan sebaliknya semakin rendah nilainya berarti kedua protein tersebut semakin tidak mirip (Budiman, 2009).

Suatu penyejajaran barisan (*sequence alignment*) dapat didefinisikan dengan ketentuan sebagai berikut :

1. Karakter-karakter dari suatu barisan DNA atau protein dipasangkan dengan karakter-karakter dari barisan DNA atau protein lainnya.
2. Adanya kemungkinan penyisipan spasi di depan, di belakang atau di tengah barisan karakter sehingga barisan-barisan tersebut memiliki panjang yang sama.
3. Urutan karakter pada barisan DNA atau protein yang telah disejajarkan sama dengan urutan karakter pada barisan semula.
4. Tidak terjadi pemasangan seluruh karakter berupa spasi pada barisan-barisan tersebut.

Pada penyejajaran barisan dipasangkan masing-masing karakter pada suatu barisan dengan karakter atau spasi pada barisan lain. Setiap pemasangan tersebut akan diberikan sebuah skor pemasangan. Skor pemasangan apabila dua karakter sama haruslah ditetapkan sebagai suatu nilai positif. Hal ini disebabkan kedua barisan dikatakan semakin mirip jika skor penyejajarannya tinggi, sementara skor penyejajaran kedua barisan tinggi apabila skor setiap pemasangan karakter yang sama bernilai tinggi. Sebaliknya, nilai ketidakcocokan ditetapkan sebagai nilai negatif atau nol. Hal yang sama juga berlaku untuk pemasangan karakter dengan spasi/*gap*. Penyejajaran yang optimum adalah yang memiliki total skor penyejajaran yang paling besar dari semua penyejajaran yang mungkin (Budiman, 2009). Pensejajaran urutan dapat dilakukan dengan dua cara yaitu pensejajaran global digunakan untuk melihat kemiripan dari dua barisan DNA atau protein secara keseluruhan, dan pensejajaran lokal digunakan untuk mencari pasangan segmen dari dua barisan dengan skor kemiripan yang paling besar, gambar 2.4 menunjukkan urutan asam amino yang disejajarkan pada pensejajaran lokal.

Gambar 2.4. Ilustrasi pensejajaran lokal

2.1.3 Algoritma Smith Waterman

Algoritma *Smith-Waterman* ditemukan pada tahun 1981 oleh TF Smith dan MS Waterman. Algoritma ini digunakan untuk menemukan penjajaran lokal yang memiliki nilai optimal lokal dari dua buah sekuen. Algoritma *Smith-Waterman* menghitung semua informasi yang terdapat pada dua sekuen sehingga jika sekuen *query* berukuran m, dan sekuen *reference* berukuran n maka waktu kompleksitas saat proses inisialisasi adalah O(m+n). Hal ini karena waktu inisialisasi hanya membutuhkan pengisian vertikal dan horizontal saja. Sedangkan waktu kompleksitas pengisian matriks adalah O(mn) karena harus mengakses semua bagian sekuen, dan saat penentuan nilai maksimum adalah O(mn) karena harus menyimpan informasi sekuen baris sekuen sebelumnya, sehingga perhitungan waktu kompleksitas Algoritma seperti pada persamaan 1 (Chan, 2004) :

$$O(m+n) + O(mn) + O(mn) = O(mn) \quad (1)$$

Misalnya diberikan $X = x_1, x_2 \ldots x_n$ dan $Y = y_1, y_2 \ldots y_m$ sebagai urutan yang harus disejajarkan, n dan m adalah panjang dari X dan Y. Algoritma Smith Waterman sebagai berikut :

1. Definisikan Matriks Skor $s(x_i, y_j)$ dan nilai *gap* .
   a. $s(x_i, y_j)$ adalah skor pemasangan untuk kesamaan/ kecocokan dan skor ketidakcocokan elemen yang membentuk dua urutan. Apabila terjadi kecocokan diberikan nilai skor positif, sedangkan apabila terjadi ketidakcocokan mendapat skor yang lebih rendah atau negatif.
   b. Nilai *gap* didefinisikan untuk penyisipan (*insertion*) dan penghapusan (*deletion*) mutasi. Skor yang didefinisikan biasanya nilai negatif.
2. Bangun matriks penilaian H dan inisiasi baris pertama dan kolom pertama. Ukuran dari matriks penilaian adalah $(m+1) * (n+1)$. Matriks menggunakan pengindeksan berbasis 0, ditunjukkan pada gambar 2.5.

$$H_{i0} = H_{0j} = 0 \quad \text{untuk} \quad 0 \leq i \leq m \text{ and } 0 \leq j \leq n \quad (2)$$

Inisialisasi pada metode *Smith Waterman* (Budiman, 2009) :

$H_{0,0}$ (Skor maksimum sebelum adanya pemasangan karakter) = 0

H_{i0} (pemasangan i karakter awal pada barisan A dengan spasi) = 0

H_{0j} (pemasangan karakter awal pada barisan A dengan spasi) = 0

	0	y	y_2	. . . .	y_{n-1}	y_n
0	$H_{0,0}$	$H_{0,1}$	$H_{0,2}$	. . . .	$H_{0,n-1}$	$H_{0,n}$
x_1	$H_{1,0}$	$H_{1,1}$	$H_{1,2}$	. . . .	$H_{1,n-1}$	$H_{1,n}$
x_2	$H_{2,0}$	$H_{2,1}$	$H_{2,2}$	. . . .	$H_{2,n-1}$	$H_{2,n}$
: :	: :	: :	: :	: :	: :	: :
x_{m-1}	$H_{m-1,0}$	$H_{m-1,1}$	$H_{m-1,2}$	. . . .	$H_{m-1,n-1}$	$H_{m-1,n}$
x_m	$H_{m,0}$	$H_{m,1}$	$H_{m,2}$	. . . .	$H_{m,n-1}$	$H_{m,n}$

Gambar 2.5. Contoh matriks pengindeksan

3. Isi matriks penilaian dengan persamaan berikut :

$$H_{ij} = max \begin{cases} H_{i-1,j-1} + s(x_i, y_j), \\ H_{i-1,j} + gap \\ H_{i,j-1} + gap \\ 0 \end{cases} \qquad (1 \le i \le m,\ 1 \le j \le n) \qquad (3)$$

Formula tersebut terjadi dikarenakan pada pensejajaran yang berakhir pada x_i dari barisan X dan y_j dari barisan Y terdapat empat kemungkinan ;

a. $H_{i-1,j-1} + s(x_i, y_j)$ adalah skor pemasangan *match* atau *mismatch* pada x_i dan y_j. Skor pemasangan *match* atau cocok harus ditetapkan sebagai nilai positif, hal ini disebabkan dua urutan dikatakan semakin mirip jika nilai kecocokannya tinggi, sementara nilai kecocokan kedua urutan tinggi apabila nilai kecocokan tiap karakter/residu tinggi. Skor pemasangan mismatch atau tidak cocok sebagai nilai negatif atau nol karena ketidakcocokan karakter/residu mengurangi kemiripan dua urutan.
b. $H_{i-1,j} + gap$ adalah skor jika x_i melakukan penyisipan elemen (*insertion*). Nilai *gap* atau penalti terjadi apabila salah satu karakter dari kedua urutan yang di sejajarkan digeser sehingga diganti dengan karakter celah kosong juga harus bernilai negatif. Hal ini disebabkan ada upaya tambahan yang diperlukan untuk menggeser karakter-karakter setelah celah disisipkan (Puspitaningrum, dkk., 2014).

c. $H_{i,\,j-1} + gap$ adalah skor jika y_j melakukan penghapusan elemen (*deletion*).
d. $H_{ij} = 0$, apabila ketiga pilihan di atas tidak lebih besar dari nilai 0 maka pensejajaran segmen dinyatakan tidak memiliki kemiripan. Agar pasangan segmen tersebut tidak mempengaruhi skor pensejajaran segmen yang lain maka skor pensejajaran pasangan segmen tersebut diberi nilai 0.

4. Penelusuran kembali (*traceback*)

H_{ij} merupakan skor penjajaran maksimum dari pemasangan dua segmen dari dua barisan sehingga semakin besar nilai H_{ij} maka kemiripan dari dua segmen tersebut semakin besar. Misalkan terdapat pasangan segmen dengan skor H_{ij} tertinggi maka segmen berakhir pada pasangan x_i dan y_j sehingga penelusuran kembali dimulai dari H_{ij}. Selanjutnya mencari asal pasangan segmen tersebut dimulai. Penelusuran kembali ini dilakukan sampai pada nilai 0 sebagai penanda bahwa pasangan segmen H_{ij} dimulai. Segmen yang bernilai 0 menandakan bahwa segmen tersebut tidak memiliki kemiripan sehingga pasangan segmen tersebut tidak termasuk dalam pasangan segmen H_{ij} yang memiliki kemiripan tertinggi karena hanya akan membuat skor kemiripan dari pasangan segmen H_{ij} berkurang.

Contoh 1:

Penerapan algoritma *Smith Waterman* pada pensejajaran dua urutan protein:

1. Diberikan dua urutan asam amino x dan y dengan panjang masing-masing n dan m :

$$x = \text{T A G A A L};\ \ y = \text{A G G L A L A}$$

Definisikan *Scoring matrix* dan nilai *gap penalty* :

Untuk *match* / cocok diberikan skor +5. Digunakan nilai +5 karena nilai ini tidak terlalu besar dan tidak terlalu kecil sehingga bisa memberi perbedaan yang jelas pada saat di proses pensejajaran dua sekuens terjadi kecocokan, apabila terjadi kecocokan nilai kemiripan bertambah dan semakin tinggi.

Untuk *mismatch* / tidak cocok diberikan skor -3. Apabila terjadi ketidakcocokan dari pensejajaran dua sekuens nilai kemiripannya berkurang, sehingga nilai ketidakcocokannya terlihat.

Untuk *Gap* diberikan skor -4. Penggunaan nilai gap -4 yang lebih besar dari nilai mismatch -3 karena upaya yang dibutuhkan ketika terjadi pergeseran karakter lebih besar , sehingga kemungkinan kemiripannya berkurang lebih besar.

2. Bangun matriks H dengan ukuran $(n+1) * (m+1)$. pengindeksan dimulai dari 0 dan Inisiasi baris dan kolom pertama menggunakan persamaan (2), gambar 2.6 adalah pengindeksan dan inisialisasi pada matriks H .

			j = 0	j = 1	j = 2	j = 3	j = 4	j = 5	j = 6	j = 7
				A	G	G	L	A	L	A
i = 0		0	0	0	0	0	0	0	0	0
i = 1	T	0								
i = 2	A	0								
i = 3	G	0								
i = 4	A	0								
i = 5	A	0								
i = 6	L	0								

Gambar 2.6. Matriks H pengindeksan dan inisialisasi

3. Isi matriks penilaian dengan persamaan (3) sesuai dengan *Scoring matrix* dan nilai *Gap*. Gambar 2.7 ditunjukkan matriks skor penilaian setelah matriks diisi.

		A	G	G	L	A	L	A
	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0
A	0	5	1	0	0	5	1	5
G	0	1	10	6	2	1	2	1
A	0	5	6	7	3	7	3	7
A	0	5	2	3	3	8	4	8
L	0	1	2	0	8	4	13	9

Gambar 2.7. Matriks dengan skor penilaian

4. Penelusuran Kembali (*traceback*)

Penelusuran kembali dimulai dari skor tertinggi dan berakhir pada nilai 0. Gambar 2.8 menunjukkan hasil penelusuran kembali matriks H.

		A	G	G	L	A	L	A
	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0
A	0	5	1	0	0	5	1	5
G	0	1	10	6	2	1	2	1
A	0	5	6	7	3	7	3	7
A	0	5	2	3	3	8	4	8
L	0	1	2	0	8	4	13	9

Gambar 2.8. Penelusuran kembali (traceback)

Setelah dilakukan penelusuran kembali diperoleh hasil sebagai berikut :

+5 -4 +5 -3 +5 +5

A	G	G	L	A	L
|	|	|	|	|	|
A	-	G	A	A	L

Gambar 2.9 *local alignment* terbaik

Berdasarkan gambar 2.9 dapat disimpulkan bahwa *local Alignment* terbaik yaitu 5 - 4 + 5 - 3 + 5 + 5 = 13.

Untuk mengukur kemiripan sequence alignment menggunakan persamaan berikut:

$$S_p(P_1, P_2) = \frac{SW(P_1,P_2)}{\sqrt{SW(P_1,P_1)} \times \sqrt{SW(P_2,P_2}} \times 100\% \quad (4)$$

Persamaan (4) adalah persamaan untuk mengetahui normalisasi nilai pada skor kemiripan yang diperoleh menggunakan algoritma Smith Waterman dengan

$S_p(P_1, P_2)$ adalah normalisasi nilai untuk skor pada pasangan protein P_1 dan P_2, $SW(P_1, P_2)$ adalah skor kemiripan pada pasangan protein P_1 dan P_2, $\sqrt{SW(P_1, P_1)}$ adalah akar dari skor kemiripan pada pasangan protein P_1 dan P_1, dan $\sqrt{SW(P_2, P_2}$ adalah akar dari skor kemiripan pada pasangan protein P_2 dan P_2 (Keum, dkk., 2015).

2.1.4 Pemrograman Dinamis

Program dinamis (*dynamic programming*) merupakan salah satu metode pemecahan masalah dengan menguraikan masalah yang rumit menjadi sub masalah yang lebih sederhana. Pemecahan sub masalah dilakukan sekali saja dan menyimpan solusinya. Sehingga apabila terdapat sub masalah yang sama, alih – alih melakukan komputasi ulang untuk mencari solusinya, seseorang hanya perlu mencari solusi yang sebelumnya pernah dipecahkan. Teknik menyimpan solusi dari sub masalah disebut *memorization* (Arifin M. A., 2018).

Konsep dasar dan karakteristik masalah pemrograman dinamis antara lain (Budiman, 2009):

1. Suatu masalah dibagi ke dalam beberapan tahapan, dengan keputusan kebijakan diperlukan pada tiap tahap.
2. Tiap tahap mempunyai sejumlah keadaan yang saling berhubungan
3. Pengaruh keputusan ditiap tahap adalah merubah keadaan sekarang menjadi keadaan yang terhubung dengan tahap berikutnya.
4. Prosedur penyelesaiannya dirancang untuk menemukan keputusan kebijakan dari keseluruhan masalah.
5. Saat keadaan sekarang, suatu keputusan kebijakan untuk tahap yang belum dilalui tidak bergantung dari kebijakan yang telah diambil pada tahap sebelumnya.
6. Prosedur penyelesaian ditandai dengan menemukan keputusan kebijakan pada tahap terakhir.
7. Adanya suatu hubungan rekursif.

Pada program dinamis, rangkaian keputusan yang optimal dibuat menggunakan prinsip optimalitas. Prinsip optimalitas menyatakan bahwa jika solusi total optimal, maka bagian solusi sampai tahap ke-*k* juga optimal. Dengan

prinsip optimalitas ini akan menjamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap – tahap selanjutnya. Hal tersebut akan berdampak pada pengurangan rangkaian keputusan yang tidak mengarah ke solusi optimum secara drastis. Pada metode *greedy* hanya satu rangkaian keputusan yang dihasilkan, sedangkan pada metode program dinamis lebih dari satu rangkaian keputusan. Hanya rangkaian keputusan yang memenuhi prinsip optimalitas yang akan dihasilkan (Arifin Y. , 2019).

Terdapat dua pendekatan dalam menyelesaikan permasalahan dengan menggunakan algoritma program dinamis, yaitu:

1. Program dimanis maju (*forward* atau *up-down*).

Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya hingga tahap n. Rentetan peubah keputusan adalah $x_1, x_2, \dots, x_n$.

2. Program dinamis mundur (*backward* atau *bottom-up*)

Program dinamis bergerak mulai dari tahap n, terus mundur ke tahap $n-1$, $n-2$, dan seterusnya hingga tahap 1. Rentetan peubah keputusan adalah $x_n, x_{n-1}, \dots, x_1$.

Secara umum, terdapat empat langkah yang perlu dilakukan dalam mengembangkan algoritma program dinamis :

1. Karakteristikkan struktur dari solusi optimal.
2. Mendefinisikan nilai solusi optimal secara rekursif.
3. Hitung nilai solusi optimal secara maju atau mundur.
4. Membangun solusi optimal dari informasi yang dihitung.

2.1.5 Algoritma *Bisection*

Salah satu metode numerik sederhana untuk pencarian akar persamaan yang telah banyak dikenal adalah Metode Bagi-Dua (*Bisection*). Metode Bagi-Dua didasarkan pada Teorema Nilai Antara untuk fungsi kontinu, yaitu bahwa suatu selang [a,b] harus memuat suatu titik nol f (akar persamaan f) bila f(a) < 0, f(b) > 0. dan berlawanan tanda, misalnya . Hal ini menyarankan metode pengulangan pembagi-duaan selang dan dalam setiap langkah mengambil setengah selang yang juga memenuhi persyaratan tersebut.

Metode Bagi-Dua memerlukan dua nilai sebagai tebakan awal, sebut a dan b, dimana a < b, yang harus memenuhi f (a) · f (b) < 0 b sehingga selang memuat satu

akar riil. Mula-mula ditentukan titik tengah selang (a,b) sebut titik tengahnya c. Diantara dua selang baru yang diperoleh yakni (a,c) dan (c,b), salah satu diantaranya pasti memuat akar. Berikutnya yang ditinjau adalah selang yang memuat akar tersebut. Proses pembagi-duaan selang ini diulang dan dilanjutkan sampai lebar yang ditinjau cukup kecil atau dengan kata lain untuk memperoleh taksiran/hampiran yang diperhalus.

Penentuan selang yang mengandung akar dilakukan dengan memeriksa tanda dari hasil kali f(a) · f(c) atau f(c) · f(b). Dalam algoritma Metode Bagi-Dua digunakan peubah-peubah: a sebagai ujung kiri selang, b sebagai ujung kanan selang, dan c sebagai titik tengah. Karena metode ini selalu menghasilkan akar maka dikatakan bahwa metode ini selalu konvergen. Besarnya epsilon tergantung pada ketelitian yang diinginkan, semakin kecil epsilon akan semakin teliti taksiran/hampiran akar yang diperoleh (Insani, 2006).

2.1.6 Algoritma *Backtracking*

Teknik runut balik (*backtracking*) merupakan salah satu teknik dalam penyelesaian masalah secara umum (*General Problem Solving*). Adapun dasar dari teknik ini adalah suatu teknik pencarian (Teknik *Searching*). Teknik pencarian ini digunakan dalam rangka mendapatkan himpunan penyelesaian yang mungkin. Dari himpunan penyelesaian yang mungkin ini akan diperoleh solusi optimal atau memuaskan (Rivai, 2011).

Runut balik (*backtracking*) adalah algoritma yang berbasis pada *Depth First Search* (DFS) untuk mencari solusi persoalan secara lebih mangkus. Runut balik yang merupakan perbaikan dari algoritma *brute-force*, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode runut balik, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Akibatnya, waktu pencarian dapat dihemat. Saat ini algoritma runut balik banyak diterapkan untuk program permainan seperti permainan tic-tac-toe, menemukan jalan keluar dalam sebuah labirin, catur, *crossword puzzle*, sudoku dan masalah-masalah pada bidang kecerdasan buatan (*artificial intelligence*) (Munir, 2013).

Dalam pencarian solusi algoritma *backtracking*, ada beberapa langkah yang dilakukan untuk mencapai solusi tersebut, yaitu sebagai berikut:

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti aturan pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup (*live node*). Simpul hidup yang sedang diperluas dinamakan simpul-E (*Expand-node)*.
2. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut "dibunuh" sehingga menjadi simpul mati (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan fungsi pembatas (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi.
3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak yang lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut balik ke simpul hidup terdekat (simpul orang tua). Selanjutnya simpul ini menjadi simpul-E yang baru.

Pencarian dihentikan bila kita telah menemukan solusi atau tidak ada lagi simpul hidup untuk runut balik (Rivai, 2011).

2.2 *State of the Art*

Penyusunan skripsi ini mengambil beberapa referensi penelitian sebelumnya termasuk jurnal-jurnal yang berhubungan dengan penelitian ini. Dalam hasil penelitian Smith and Waterman memperluas ide analisis sekuens modern yang dimulai dengan *the heuristic homology algorithm* oleh Needleman and Wuncsh (1970). Menemukan sepasang segmen dari masing-masing dua urutan panjang, sehingga tidak ada pasangan segmen lainnya dengan kesamaan yang lebih besar (homologi). Ukuran kesamaan yang digunakan disini memungkinkan untuk penghapusan dan penyisipan pada urutan. Pada algoritma ini tidak hanya menempatkan pencarian untuk maksimal kemiripan pada pasangan segmen secara

matematis tetapi juga dapat diprogram secara efisien dan sederhana di komputer (Smith & Waterman, 1981).

Penelitian selanjutnya membangun sebuah *prototipe* aplikasi berbasis *dekstop* yang dapat melakukan *sequence alignment* pada sekuen protein dengan algortima *Smith-Waterman*. Penelitian ini dilakukan untuk mengetahui tingkat kesamaan dari satu sekuen protein terhadap satu sekuen protein lainnya dengan menggunakan algoritma *Smith-Waterman*. Pada prosesnya hanya dilakukan analisis pada format *file* Fasta dari GenBank (NCBI) sebagai penyimpanan sekuen protein. Pada prototipe aplikasi *SeqAP*, sekuen protein dibaca dari *file* dengan format Fasta. Nilai konstanta pada matriks BLOSUM50 digunakan dalam menentukan nilai sepasang residu yang disejajarkan. Hasil keluaran *SeqAP* tersebut yaitu nilai *similarity* yang diurutkan secara *descending*. Panjang sekuen yang dibandingkan berbeda-beda, sehingga dilakukan modifikasi pada perhitungan nilai optimal yaitu pada hasil pensejajaran sekuen ditampilkan *id* sekuen dengan nilai paling tinggi dalam *database*, Sehingga dapat disimpulkan bahwa perhitungan nilai optimal tidak tergantung pada panjang pendeknya karakter sekuen karena *local alignment* hanya menghitung nilai optimal dari setiap segmen yang memiliki kesamaan (Bu'ulölö, dkk., 2010).

Algoritma Smith Waterman memiliki kompleksitas kuadratik. Dalam penelitian lainnya algoritma *Smith Waterman* akan dijalankan secara paralel dengan teknologi dari sebuah GPU yang dinamakan dengan *CUDA core* dan model *inter-task parallelization* dengan skema *rowwise alignment* atau pengisian matriks berbasis baris. Model *inter-task parallelization* yaitu sebuah model dimana setiap *thread* akan menjalankan tepat satu tugas dan masing-masing *thread* akan dijalankan secara paralel. Metode ini digunakan karena panjang sekuens pada data Ijah webserver sebanyak 98% memiliki ukuran kurang dari 3000 residu, namun dilakukan penjajaran secara keseluruhan data, dalam artian setiap data akan dihitung dengan seluruh data yang ada. Sehingga apabila mempunyai i data maka akan melakukan perhitungan sebanyak i^2, sedangkan algoritma Smith Waterman memiliki kompleksitas O(mn) yang mana m dan n merupakan panjang sekuens yang dijajarkan, sehingga keseluruhan penjajaran sekuens pada Ijah webserver memiliki kompleksitas $O(i^2 * mn)$. Data yang digunakan pada penelitian ini adalah

data protein pada basis data Ijah webserver sebanyak 3334 uniprot_id. Pada penelitian ini disimpulkan bahwa penggunaan GPU pada data yang kecil akan mengalami penurunan *speed up* dikarenakan biaya yang digunakan tidak sebanding dengan data yang diproses, begitu pula sebaliknya semakin besar data yang diproses maka *speed up* yang dihasilkan akan semakin tinggi. Namun aplikasi ini membutuhkan *memory* yang cukup besar untuk memproses data yang besar, karena setiap data yang dibutuhkan akan dialokasikan terlebih dahulu sebelum diproses. Penelitian ini menghasilkan *speed up* sebesar *2,29 fold* dengan 30 data yang diproses (Nugroho & Kusuma, 2018).

Pada penelitian lainnya, penulis mencari pasangan segmen dari dua DNA yang telah disejajarkan yang memiliki tingkat kemiripan paling tinggi dari pasangan segmen lainnya menggunakan metode *Smith Waterman*. Dalam proses mencocokan dua atau lebih DNA tidak mungkin menemukan kesamaan 100% karena DNA bersifat unik. Namun, karena panjang DNA (jumlah rangkaian gugus karbon) dua atau lebih organisme tidak selalu sama, maka proses yang digunakan bukanlah 'pembandingan', melainkan 'pensejajaran'. Suatu barisan DNA dapat dinyatakan sebagai barisan karakter dari adenin (dilambangkan A), sitosin (C, dari *cytosine*), guanin (G), dan timin (T). Hal yang sama juga diterapkan dalam proses pensejajaran protein. Pada penelitian ini hanya menghitung kemiripan antara dua barisan DNA. Kelebihan pada metode *Smith-Waterman* adalah dapat menentukan pasangan segmen dari dua barisan yang memiliki kemiripan paling tinggi tanpa harus memperhitungkan seluruh kemungkinan pasangan segmen yang mungkin terjadi. Metode ini juga dapat mengurutkan pasangan segmen berdasarkan dari kemiripannya (Budiman, 2009).

Penelitian lainnya juga difokuskan kepada pensejajaran lokal dengan algoritma *Smith-Waterman*. Algoritma ini mencoba menemukan sebanyak mungkin kesamaan dari sepasang sekuen, dengan cara memberikan nilai negatif pada *base pair* yang tidak sama (*mismatch*), dan nilai positif pada *base pair* yang sama (*match*). Sehingga nantinya akan didapatkan nilai positif maksimum sebagai akhir dari pensejajaran, dan nilai minimum sebagai awal pensejajaran. Data yang digunakan pada penelitian ini adalah data berformat Fasta berasal dari *GenBank* yang diunduh dari situs resmi *National Centre for Biotechnology Information*

(NCBI). Pada data pertama sekuen DNA *Ancylostoma duodenale* (NC_003415.1) dan *Necator americanus* (NC_003416.2), hasil pensejajaran menunjukkan similaritas 84%, dengan *gap* 3%. Data kedua sekuen DNA *Human papillomavirus type* 134 (NC_014956.1) dan *Human papillomavirus type* 132 (NC_014955.1), hasil pensejajaran menunjukkan similaritas 62.1%, dengan *gap* 15.9%. Data ketiga sekuen DNA *Chaetoceros lorenzianus* DNA virus (NC_015211.1) dan *Chaetoceros tenuissimus* DNA virus (NC_014748.1), hasil pensejajaran menunjukkan similaritas 53.2% dengan *gap* 26.2%. Data terakhir sekuen DNA *Thermoproteus tenax spherical virus* 1 (NC_006556.1) dan *Vesicular stomatitis indiana virus* (NC_001560.1), hasil pensejajaran menunjukkan similaritas 40.8%, dengan *gap* 47.6%. Hal ini menunjukkan bahwa pensejajaran menggunakan algoritma *Smith-Waterman* menghasilkan beragam nilai kemiripan yang tergantung dari panjang sekuen, dan bentuk sekuennya (Himawan, 2013).

Pasangan sekuen dengan panjang yang sama akan memakan waktu yang lebih lama ketimbang panjang pasangan yang berbeda, hal ini dikarenakan waktu kompleksitas yang menjadi kuadratik saat sekuen *query* dan *reference* memiliki panjang yang sama, sedangkan jika terjadi perbedaan panjang *query* dan *reference*, salah satu sekuen akan menjadi lebih panjang dari satu lainnya, sehingga $m \neq n$, yang mengakibatkan berkurangnya waktu *eksekusi program.* Pada penelitian ini mengaplikasikan metode *Bisection* dan *backtracking* diharapkan mampu mengoptimalkan kemampuan maksimum panjang sekuen yang disejajarkan serta waktu eksekusi dari Algoritma *Smith Waterman*.

2.3 Kerangka Konseptual

Pendahuluan : Protein merupakan polipeptida yang tersusun atas asam amino. Protein dibentuk melalui proses transkripsi dan translasi. Urutan basa yang berbeda pada mRNA akan menghasilkan asam amino yang berbeda pada setiap protein.

Latar Belakang : Penelitian mengenai kemiripan antar protein dibutuhkan sehingga dapat dibandingkan kemiripan asam amino yang menyusun protein. Salah satu pendekatan bioinformatika yang umum dilakukan yaitu perbandingan kemiripan urutan dengan mensejajarkan urutan untuk mengidentifikasi atau menemukan daerah kesamaan antara dua urutan baik itu urutan DNA maupun protein.

Masalah : Pengukuran kemiripan antara urutan asam amino dilakukan menggunakan Algoritma *Smith Waterman*. Kompleksitas waktu yang digunakan pada algoritma ini adalah O(mn) di mana m dan n merupakan panjang dari kedua urutan yang dicari, sehinga membutuhkan biaya yang mahal terutama untuk waktu perhitungan. Maka dari itu dibutuhkan metode yang tepat untuk lebih mengoptimalkan waktu komputasi.

Solusi : Dalam penelitian ini, akan dilakukan pengembangan pada algoritma *Smith Waterman* untuk meminimalkan dan mengoptimalkan waktu komputasi

Tujuan yang diharapkan : Dengan adanya penelitian ini diharapkan dapat ditemukan metode yang tepat dalam pensejajaran urutan dengan waktu komputasi yang lebih cepat dan nilai kesamaan yang tinggi

Gambar 2.10. Kerangka Konseptual