Algoritma Penyejajaran Multi-Kriteria Berbasis Program Dinamis untuk Deteksi Varian Genetik DNA dalam Konteks Penyakit Herediter

Ferdinand Gabe Tua Sinaga - 13523051
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: ferdinandgts5@gmal.com , 13523051@std.stei.itb.ac.id

***Abstract*—Penerapan algoritma Needleman-Wunsch untuk penjajaran sekuens DNA, yang kemudian dimanfaatkan untuk mendeteksi dan mengklasifikasikan varian genetik. Algoritma ini dimodifikasi dengan mengintegrasikan matriks penskoran yang mempertimbangkan perbedaan biologis antara transisi dan transversi, serta penerapan affine gap penalty untuk penanganan insersi/delesi yang lebih realistis. Varian genetik yang teridentifikasi kemudian diklasifikasikan menggunakan database simulasi berdasarkan data dari ClinVar dan gnomAD. Hasil pengujian menunjukkan bahwa implementasi ini berhasil secara akurat mendeteksi berbagai jenis varian (substitusi, insersi, dan delesi) dan memberikan klasifikasi klinis awal. Meskipun terbukti efektif dalam akurasi, analisis komputasi mengungkap keterbatasan efisiensi waktu untuk pemrosesan sekuens DNA berukuran panjang.**

Kata Kunci— Needleman-Wunsch; penjajaran sekuens; varian genetik; bioinformatika; klasifikasi klinis

I. LATAR BELAKANG

Penyakit genetic merupakan kondisi yang timbul akibat adanya perubahan pada rantai DNA individu, perubahan tersebut dapat berupa mutase gen tunggal atau mutase pada kromosomnya. Perubahan kecil saja dalam rantai DNA dapat menimbulkan efek samping yang signifikan contoh nyatanya ada pada penyakit cystic fibrosis, yang disebabkan oleh mutasi pada gen *CFTR*,. dan sickle cell anemia, yang diakibatkan oleh substitusi asam amino tunggal pada protein hemoglobin, mengubah bentuk sel darah merah dan menyebabkan komplikasi serius. Kedua contoh diatas mengilustrasikan bagaimana satu gen yang cacat dapat mengganggu fungsi organ vital.

Kemajuan pesat teknologi telah membantu manusia untuk memahami, mengidentifikasi dan menangani penyakit genetic dari seseorang. Metode-metode pendeteksian tersebut telah berkembang secara signifikan, dimulai dari teknik sekuensing Sanger yang klasik hingga munculnya teknologi sekuensing generasi berikutnya (Next-Generation Sequencing/NGS) yang mampu memproses data genetik dalam jumlah besar dengan cepat dan akurat. Perkembangan tersebut memungkinkan para ilmuan dan tenaga untuk melakukan diagnosis penyakit genetik secara lebih dini dan tepat sasaran.

Dalam konteks ini, bioinformatika memegang peranan krusial dalam mendeteksi mutasi genetik serta mendukung perkembangan metode-metode analisis sekuens yang telah disebutkan sebelumnya. Dengan memanfaatkan algoritma *string matching*, bidang ini memungkinkan para ilmuwan untuk menganalisis sekuens DNA secara efisien, guna mengidentifikasi varian genetik yang bersifat normal maupun yang patogenik. Melalui proses perbandingan antara sekuens DNA individu yang memiliki kondisi tertentu dengan sekuens DNA referensi, bioinformatika membantu dalam mengungkap variasi protein atau mutasi yang menjadi penyebab penyakit herediter, baik yang bersifat jinak maupun yang berpotensi membahayakan.

Metode dasar untuk membandingkan sekuens DNA adalah penyejajaran sekuens DNA. Proses ini bertujuan untuk menemukan daerah homologi dan perbedaan antara dua atau lebih sekuens, yang secara visual direpresentasikan dengan menyelaraskan basa-basa yang cocok dan menyoroti perbedaan (*mismatch* atau *indel*). Algoritma penyejajaran standar, seperti Needleman-Wunsch tradisional, menggunakan pendekatan matriks untuk menemukan penyejajaran global optimal antara dua sekuens. Namun, algoritma dasar ini memiliki keterbatasan signifikan, terutama dalam konteks analisis varian genetik. Salah satu keterbatasan utamanya adalah kecenderungan untuk memperlakukan semua *mismatch* sama. Ini berarti substitusi G ke A (transisi) dinilai sama buruknya dengan substitusi G ke T (transversi), padahal secara biologis, mismatch biologis tidak sama. Transisi (purin ke purin atau pirimidin ke pirimidin) cenderung lebih sering terjadi dan seringkali memiliki dampak yang lebih kecil dibandingkan transversi (purin ke pirimidin atau sebaliknya).

Penyejajaran standar seringkali gagal membedakan antara varian jinak yang umum dalam populasi dan varian patogenik yang menyebabkan penyakit. Varian jinak mungkin tidak memiliki relevansi klinis, sementara varian patogenik dapat menjadi kunci diagnosis dan terapi. Algoritma dasar yang hanya memberikan "penalti" yang sama untuk semua ketidakcocokan tidak dapat mencerminkan kompleksitas biologis ini. Oleh karena itu, diperlukan metode yang lebih canggih untuk secara akurat mencerminkan bobot biologis yang berbeda dari berbagai jenis varian genetik.

Untuk mengatasi keterbatasan metode standar, pendekatan yang lebih fleksibel dan akurat diperlukan. Di sinilah program dinamis muncul sebagai solusi yang sangat efektif. Program dinamis adalah teknik algoritma kuat yang memecah masalah kompleks menjadi sub-masalah yang lebih kecil dan lebih mudah dikelola, kemudian menggabungkan solusi dari sub-masalah tersebut untuk membangun solusi akhir. Dalam bioinformatika, teknik ini telah menjadi dasar bagi algoritma penyejajaran sekuens yang sangat akurat, seperti Needleman-Wunsch untuk penyejajaran global dan Smith-Waterman untuk penyejajaran lokal.

Keunggulan utama program dinamis terletak pada kemampuannya untuk diadaptasi untuk fleksibilitas lebih lanjut. Alih-alih menerapkan penalti yang seragam untuk semua ketidakcocokan, kerangka kerja program dinamis memungkinkan penggunaan matriks skor yang disesuaikan. Matriks ini dapat menetapkan bobot atau "penalti" yang berbeda untuk jenis *mismatch* yang berbeda, misalnya, memberikan penalti lebih rendah untuk transisi dibandingkan transversi, atau bahkan membedakan antara substitusi asam amino yang konservatif versus non-konservatif. Fleksibilitas ini memungkinkan algoritma penyejajaran untuk lebih akurat mencerminkan realitas biologis dan membedakan antara varian genetik yang memiliki implikasi klinis yang berbeda, sehingga sangat penting untuk deteksi varian yang relevan secara klinis dalam genomika penyakit.

II. DASAR TEORI

A. Algoritma Needleman-Wunsch

Algoritma Needleman-Wunsh adalah algoritma program dinamis yang dipakai dalam bidang bioinformatika demi membantu ilmuan dan tenaga medis untuk melakukan penjajaran sekuens DNA. Algoritma ini akan secara rekursif mencari hasil pensejajaran paling baik dari dua sekuens DNA. Pada dasarnya untuk menjalan kan algoritma ini dibutuhkan 2 tahap utama yaitu:

1. Proses Pembuatan Matriks
   Sebuah matriks dua dimensi akan dibentuk untuk menyimpan solusi sementara, matriks ini sering disebut sebagai matriks score dalam dynamic programming. Setiap sel dalam matriks ini akan merepresentasikan nilai skor penyejajaran dari prefix sekuens pertama hingga karakter ke-i dan prefix sekuens kedua hingga karakter ke-j. Inisialisasi matriks ini akan dimulai dengan mengisi baris dan kolom pertamanya dengan nilai pinalti gap kumulatif dan mengisi sel(0,0) dengan nilai 0. Selanjutnya setiap sel didalam matriks akan diisi dengan cara rekursif dengan mempertimbangkan tiga kemungkinan operasi penyejajaran yang ada yaitu Match/Missmatch, Insert dan Delete.

   Setiap sel dari matriks tersebut akan diisi dengan rumus dibawah ini

$$S(i,j) = \max \begin{cases} S(i-1,j-1) + \text{skor}(seq1[i-1], seq2[j-1]) \\ S(i,j-1) + \text{gap\_penalty} \\ S(i-1,j) + \text{gap\_penalty} \end{cases}$$

2. Proses Traceback:
   Setelah seluruh elemen dalam matriks diisi, nilai penyejajaran global yang paling optimal akan ditemukan di sel paling kanan bawah (S(m,n), di mana m dan n merupakan panjang dari masing-masing sekuens). Untuk membentuk kembali hasil penyejajaran yang sebenarnya, algoritma melakukan penelusuran ulang dari sel ini menuju sel awal (0,0). Pada setiap langkah, algoritma mengikuti jalur yang memberikan skor tertinggi pada sel tersebut, dengan kemungkinan pergerakan secara diagonal untuk match atau missmatch, ke kiri untuk insert, atau ke atas delete. Dengan demikian, proses ini secara bertahap menyusun penyejajaran dari akhir menuju awal.

Untuk memperjelas bagaimana algoritma ini bekerja, berikut disajikan ilustrasi sederhana dari proses pensejajaran dua sekuens DNA pendek, yaitu “AG” dan “AT”. Skor yang digunakan dalam algoritma adalah: +1 untuk *match*, -1 untuk *mismatch*, dan -2 sebagai *gap penalty*. Hasil iterasi pengisian matriks dapat dilihat pada tabel di bawah ini, di mana skor optimal penyejajaran global akan ditemukan pada sel paling kanan bawah:

	-	A	T
-	0	-2	-4
A	-2	1	-1
G	-4	-1	0

Untuk iterasi pertama kita diharuskan untuk mencari 3 nilai yaitu nilai diagonal, kiri dan atas. Berdasarkan rumus yang sudah kami jelaskan diatas, perhitungan nilai Diagonal akan melibatkan nilai diagonal dari S(i-1,j-1) + skor match atau mismatch. Dalam konteks ini nilai dari S(i-1,j-1) = 0 dan karena “A” pada seq 1 match dengan “A” pada seq 2 maka nilai diagonal akan 0 + 1 = 1. Lalu untuk nilai kiri kita akan menambahkan nilai dikiri dari S(i,j-1) = -2 dengan gap pinalti yaitu -2 sehingga nilai kiri = -4. Sama juga untuk nilai atas kita akan diminta untuk menghitung S(i-1,j) dimana untuk konteks ini nilainya -2 lalu ditambah dengan gap pinaltinya -2 sehingga nilai finalnya -4. Setelah mendapatkan semua nilai untuk iterasi pertama yang i dan j nya = 1, kita harus mencari maximum dari ketiga nilai tersebut, yang pada iterasi ini maxnya adalah 1. Itulah contoh ilustrasi iterasinya, ilustrasi ini menjelaskan mengapa pada matriks S(1,1) nilai nya adalah 1.

B. Jarak Edit

Skor akhir yang dihasilkaan oleh algoritma Needleman-Wunsch merepresentasikan seberapa mirip kedua sekuens

yang diberikan. Jika semakin tinggi skornya maka semakin mirip keduanya dan sebaliknya, jika semakin kecil artinya mengindikasikan semakin tidak miripnya kedua sekuens tersebut. Dalam matematika konsep tersebut dinamakan sebagai Levenshtein distance. Pada dasarnya Levenshtein distance atau jarak edit digunakan untuk menghitung jumlah operasi minimum seperti substitusi, insersi dan atau delesi yang diperlukan untuk mengubah 1 sekuens menjadi sekuens yang lain.

Dalam algoritma Needleman, banyaknya operasi untuk insersi atau delesi itu dihitung dengan memanfaatkan gap penalty. Penentuan *gap penalty* yang baik sebenarnya sangat dibutuhkan, terutama untuk menemukan dan menghasilkan hasil yang dapat diandalkan dalam dunia medis. Ini karena dalam dunia medis, mutasi satu nukleotida saja terkadang bobotnya bisa lebih besar daripada insersi/delesi kecil yang tidak memberikan dampak signifikan. Namun, seringkali satu kejadian *gap* tunggal (misalnya, insersi atau delesi satu nukleotida) dalam biologi mungkin merupakan peristiwa genetik tunggal yang sama disruptifnya dengan banyak insersi/delesi kecil secara independen.

Dalam konteks biologis, asumsi bahwa semua *mismatch* memiliki nilai penalti yang sama adalah penyederhanaan yang kurang akurat. Mutasi genetik tidak terjadi secara acak dengan probabilitas yang sama
beberapa jenis mutasi lebih sering terjadi atau bisa saja memiliki dampak fungsional yang lebih kecil daripada yang lain. Oleh karena itu, skor *match/mismatch* tidak harus selalu +1/-1. Untuk mencerminkan realitas biologis ini, digunakan matriks penskoran atau substitusi yang disesuaikan.
Matriks ini memberikan skor yang berbeda untuk setiap kemungkinan pasangan *match* atau *mismatch*, membedakan antara jenis substitusi yang berbeda. Misalnya, dalam DNA, substitusi dapat dibagi menjadi dua kategori utama:

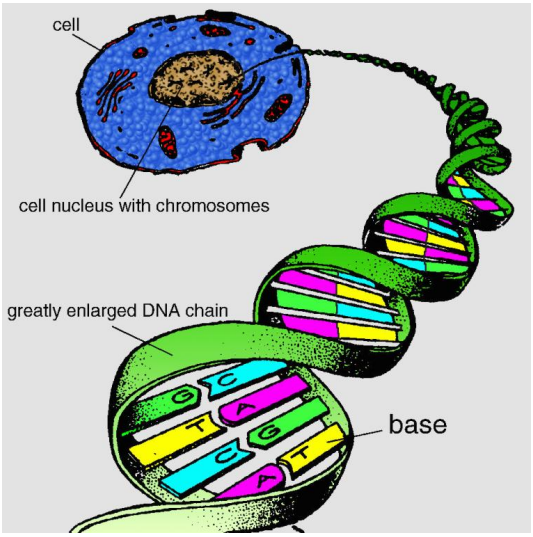
- Transisi: Perubahan dari purin ke purin (A ↔ G) atau pirimidin ke pirimidin (C ↔ T). Transisi adalah jenis substitusi yang lebih sering terjadi di alam dan cenderung kurang disruptif terhadap struktur atau fungsi protein jika terjadi dalam daerah pengkode.
- Transversi: Perubahan dari purin ke pirimidin (A ↔ C, A ↔ T, G ↔ C, G ↔ T) atau pirimidin ke purin (C ↔ A, C ↔ G, T ↔ A, T ↔ G). Transversi lebih jarang terjadi dan berpotensi lebih disruptif karena melibatkan perubahan kelas basa yang lebih fundamental.

Penggunaan matriks penskoran yang mempertimbangkan perbedaan biologis ini memungkinkan algoritma Needleman-Wunsch untuk menghasilkan penyejajaran yang lebih akurat dan bermakna secara biologis, terutama dalam konteks identifikasi varian genetik yang relevan dengan penyakit.

C. Pentingnya DNA dalam Analisis Varian Genetik

Deoxyribonucleic Acid atau yang biasa kita kenal sebagai DNA adalah molekul paling dasar yang membentuk makhluk hidup. Ia memiliki peran penting untuk bertindak sebagai blueprint genetik yang menyimpan seluruh informasi yang diperlukan untuk membangun dan menjalankan sel-sel, jaringan, serta organ tubuh. Oleh karena itu, perubahan sedikit saja padanya tentu akan berdampak besar pada fungsionalitas organ yang ada dalam individu tersebut.

DNA sendiri menyimpan informasi *blueprint* tersebut dalam bentuk urutan basa nukleotida (Adenin/A, Timin/T, Guanin/G, Sitosin/C). Urutan spesifik dari basa-basa inilah yang membentuk 'kode' genetik, yang kemudian akan ditranskripsi menjadi RNA dan selanjutnya ditranslasi menjadi protein. Molekul-molekul protein itulah yang nantinya akan menjalankan sebagian besar fungsi dalam sel.



Gambar Rantai DNA adalam Sel

Dalam konteks bioinformatika dan medis, perubahan kecil pada sekuens DNA, yang sering disebut sebagai mutasi gen atau varian genetik, dapat mengakibatkan dampak signifikan. Klasifikasi dampak tersebut dapat dipetakan berdasarkan jenis perubahan pada rantai nukleotida tersebut. Varian-varian ini dapat berupa:

- Substitusi (SNV - Single Nucleotide Variant): Perubahan satu basa dengan basa lain (misalnya A menjadi T). Konsep Subtitusi ini sebenarnya masih bisa dibagi lagi klasifikasinya yaitu berdasarkan perubahannya apakah jenis basa tujuannya sama atau tidak.
- Insersi: Penambahan satu atau lebih basa ke dalam sekuens.
- Delesi: Penghapusan satu atau lebih basa dari sekuens.

Varian-varian ini bisa bersifat jinak atau benign yang artinya tidak memiliki efek merugikan, tidak pasti (variant of uncertain significance/VUS), yang dampaknya belum sepenuhnya dipahami atau patogenik, yang secara langsung menyebabkan atau berkontribusi pada suatu penyakit.

Algoritma Needleman-Wunsch memainkan peran krusial dalam identifikasi varian ini. Dengan melakukan penjajaran sekuens DNA pasien terhadap sekuens referensi yang diketahui, algoritma ini mampu secara presisi mendeteksi lokasi dan jenis varian yang ada. Informasi ini kemudian sangat penting bagi ilmuwan untuk:

1. Memahami variansi genetik penyakit: Mengidentifikasi varian patogenik yang terkait dengan kondisi kesehatan tertentu.
2. Mendiagnosis penyakit: Menggunakan profil varian genetik pasien untuk diagnosis yang lebih akurat.
3. Mengembangkan terapi yang ditargetkan: Memanfaatkan informasi varian untuk merancang pengobatan yang lebih efektif dan personal (pengobatan presisi).
4. Penelitian genetik: Menganalisis variasi dalam populasi untuk memahami evolusi dan keragaman genetik.

III. IMPLEMENTASI DAN PENGUJIAN

Proses implementasi sistem analisis varian genetik ini dimulai dengan pengembangan inti algoritma penjajaran sekuens DNA, dilanjutkan dengan modul untuk identifikasi dan klasifikasi varian berdasarkan database yang disimulasikan. Seluruh fungsionalitas ini dikembangkan menggunakan bahasa pemrograman Python.

Tahap pertama dalam implementasi adalah pembuatan algoritma Needleman-Wunsch. Dalam program yang dibuat algoritma tersebut di implementasikan dalam fungsi needleman_wunsch yang menerima dua parameter berupa seq_a, seq_b, params. Dalam implementasinya, kami membangun dua matriks utama menggunakan NumPy yang berukuran (n+1)×(m+1). Salah satu dari matriks tersebut akan dimanfaatkan untuk menyimpan skor penjajaran (F) dan satu lagi untuk merekam jalur penelusuran balik (P).

Proses inisialisasi matriks F dilakukan dengan mengisi baris dan kolom pertamanya menggunakan penalti *gap* kumulatif.

```python
def needleman_wunsch(seq_a, seq_b, params):
    n = len(seq_a)
    m = len(seq_b)
    F = np.zeros((n + 1, m + 1))
    P = np.zeros((n + 1, m + 1), dtype=int)

    # Inisialisasi baris pertama dan kolom pertama
    for i in range(n + 1):
        F[i, 0] = params['gap_open'] + (i - 1) * params['gap_extend'] if i > 0 else 0
        P[i, 0] = 1
    for j in range(m + 1):
        F[0, j] = params['gap_open'] + (j - 1) * params['gap_extend'] if j > 0 else 0
        P[0, j] = 2
```

Gambar Proses Inisialisasi Matriks F

Setelah inisialisasi, pengisian matriks F dilakukan secara iteratif. Setiap sel F[i, j] dihitung berdasarkan nilai maksimum dari tiga kemungkinan yaitu diagonal untuk *match* atau *mismatch*, atas untuk delesi di seq_b, dan kiri untuk insersi di seq_a.

```python
# Isi matriks F dan P
for i in range(1, n + 1):
    for j in range(1, m + 1):
        char_a = seq_a[i - 1]
        char_b = seq_b[j - 1]

        # Ambil skor dari matriks penskoran yang baru
        if char_a in params['dna_matrix'] and char_b in params['dna_matrix'][char_a]:
            match_score = params['dna_matrix'][char_a][char_b]
        else:
            match_score = -1

        # Hitung skor dari 3 kemungkinan arah
        score_diag = F[i - 1, j - 1] + match_score
        score_up = F[i - 1, j] + (params['gap_extend'] if P[i - 1, j] == 1 else params['gap_open'])
        score_left = F[i, j - 1] + (params['gap_extend'] if P[i, j - 1] == 2 else params['gap_open'])

        # Pilih skor maksimum
        F[i, j] = max(score_diag, score_up, score_left)

        # Catat arah (untuk traceback)
        if F[i, j] == score_diag:
            P[i, j] = 0 # Diagonal
        elif F[i, j] == score_up:
            P[i, j] = 1 # Atas
        else:
            P[i, j] = 2 # Kiri
```

Gambar Algoritma pengisian Matriks F

Demi memenuhi kebutuhan analisis dua sekuens DNA yang berbeda algoritma ini sedikit diperluas dengan pembobotan score matrix yang disesuaikan dengan bobot perubahan nukelotida. Pembobotan untuk *match* dan *mismatch* diimplementasikan secara *hardcoded* dalam sebuah matriks penskoran DNA yang merupakan bagian dari parameter params. Matriks ini secara spesifik memberikan skor yang lebih tinggi untuk *match* (+1), penalti sedang (+0.5) untuk transisi seperti perubahan purin ke purin atau pirimidin ke pirimidin, dan penalti yang lebih besar (-2) untuk transversi yang bisa berupa perubahan purin ke pirimidin atau sebaliknya. Pendekatan ini memungkinkan algoritma untuk menghasilkan penjajaran yang lebih akurat secara biologis, mengingat bahwa beberapa jenis mutasi lebih sering terjadi atau memiliki dampak fungsional yang berbeda.

```python
dna_scoring_matrix = {
    'A': {'A': 1,   'C': -2,  'G': 0.5, 'T': -2},
    'C': {'A': -2,  'C': 1,   'G': -2,  'T': 0.5},
    'G': {'A': 0.5, 'C': -2,  'G': 1,   'T': -2},
    'T': {'A': -2,  'C': 0.5, 'G': -2,  'T': 1},
}
```

Gambar Pembuatan Score Matrix untuk Rantai Nukelotida

Setelah matriks F terisi penuh, algoritma akan memasuki tahapan kedua yaitu proses *traceback.* Proses ini dimulai dari sel paling kanan bawah matriks F, merepresentasikan skor penjajaran global optimal dari kedua sekuens. Kemudian algoritma akan memanfaatkan matriks P untuk membangun kembali kedua sekuens. Pada setiap langkah *traceback*, pergerakan yang terekam dalam matriks P (diagonal, atas, atau kiri) dicatat untuk secara bertahap merekonstruksi dua sekuens

yang telah dijajarkan, yaitu aligned_ref (sekuens referensi yang sudah dijajarkan) dan aligned_pat (sekuens pasien yang sudah dijajarkan).

Dalam implementasi *traceback* itu, kami melakukan sedikit modifikasi yang memanfaatkan matriks P demi mengidentifikasi posisi-posisi varian (seperti *match*, *mismatch*, insersi, atau delesi) langsung dari hasil penjajaran. Jika karakter pada aligned_ref dan aligned_pat berbeda namun keduanya adalah basa, maka itu menandakan adanya mismatch atau Varian Nukleotida Tunggal (SNV).
Jika ada *gap* ('-') di sekuens pasien (aligned_pat) pada posisi tertentu, sementara sekuens referensi (aligned_ref) memiliki basa, ini mengindikasikan adanya delesi pada sekuens pasien.
Sebaliknya, jika ada *gap* ('-') di sekuens referensi (aligned_ref) pada posisi tertentu, sementara sekuens pasien (aligned_pat) memiliki basa, ini mengindikasikan adanya insersi pada sekuens pasien.

```
# Traceback untuk mendapatkan aligned sequences
aligned_a = ""
aligned_b = ""
i, j = n, m
while i > 0 or j > 0:
    if P[i, j] == 0: # Diagonal
        aligned_a = seq_a[i - 1] + aligned_a
        aligned_b = seq_b[j - 1] + aligned_b
        i -= 1
        j -= 1
    elif P[i, j] == 1: # Atas (gap di seq_b)
        aligned_a = seq_a[i - 1] + aligned_a
        aligned_b = "-" + aligned_b
        i -= 1
    else: # Kiri (gap di seq_a)
        aligned_a = "-" + aligned_a
        aligned_b = seq_b[j - 1] + aligned_b
        j -= 1
```

Gambar algoritma traceback dalam fungsi Needleman

Setelah varian berhasil diidentifikasi dari sekuens yang dijajarkan, langkah selanjutnya adalah **generasi dan penggunaan database simulasi** untuk klasifikasi klinis. Database ini disimulasikan berdasarkan informasi yang relevan dari platform seperti ClinVar, yang menyimpan data tentang patogenisitas varian.

Untuk menjaga simplisitas dari databasenya kami hanya mengambil beberapa contoh nyata dari varian yang kami selidiki dari database yang disediakan oleh ClinVar. Representasi database tersebut diimplementasikan dalam kamus local bernama simulated_clinvar_data_local. Kamus ini berisi informasi minimal yang krusial untuk klasifikasi varian, di mana setiap entri menggunakan string format VCF minimal. Format ini tersusun atas elemen CHROM-POS-REF-ALT sebagai kunci unik untuk mengidentifikasi varian. Dimana CHROM berfungsi sebagai ID dari varian, POS sebagai posisi awal varian relative terhadap sekuens, REF sebagai basa asli yang ada pada posisi awal tersebut dan ALT adalah perubahannya pada pattern DNA pasien. Sebagai contoh, 11-5225480-CA-C mengindikasikan varian pada kromosom 11 di posisi 5225480, di mana sekuens referensi 'CA' berubah menjadi 'C' (sebuah delesi). Nilai tersebut akan digunakan sebagai kunci unik untuk mengidentifikasi varian, dan nilai yang diasosiasikan adalah klasifikasi klinisnya seperti, "Pathogenic", "Likely pathogenic", "Uncertain significance", dan "Benign". Melengkapi informasi ini, kami juga menyertakan kamus simulated_gnomad_data_local, yang memiliki struktur kunci varian serupa (CHROM-POS-REF-ALT), namun nilainya adalah frekuensi varian tersebut dalam populasi umum yang disimulasikan. Informasi frekuensi ini sangat penting dalam menilai signifikansi klinis suatu varian, mengingat varian patogenik yang serius cenderung memiliki frekuensi yang sangat rendah dalam populasi sehat. Kedua kamus simulasi ini kemudian diimplementasikan oleh fungsi classify_variant, yang dirancang untuk membandingkan varian yang terdeteksi pada sekuens pasien dengan entri yang ada di kedua database simulasi.

```
simulated_gnomad_data_local = {
    "11-5225480-CA-C": 0.00000001,
    "11-5225466-A-T": 0.15,
    "11-5225483-CTGTGTT-C": 0.00000001,
    "11-5225485-G-C": 0.00000001,
    "11-5225486-T-A": 0.05,
    "11-5225487-G-C": 0.00000001,
    "11-5225486-T-TC": 0.00000001,
}

simulated_clinvar_data_local = {
    "11-5225480-CA-C": "Pathogenic",
    "11-5225466-A-T": "Uncertain_significance",
    "11-5225483-CTGTGTT-C": "Pathogenic/Likely_pathogenic",
    "11-5225485-G-C": "Pathogenic/Likely_pathogenic",
    "11-5225486-T-A": "Uncertain_significance",
    "11-5225487-G-C": "Pathogenic",
    "11-5225486-T-TC": "Pathogenic",
}
```

Gambar Database Lokal yang digunakan untuk simulasi

Setelah mengintegrasikan semua aspek penting untuk analisis, ini saat yang tepat untuk menguji kelayakan program kami. kami menggunakan sekuens DNA HBB dari individu sehat sebagai dasar sekuens orang sehat

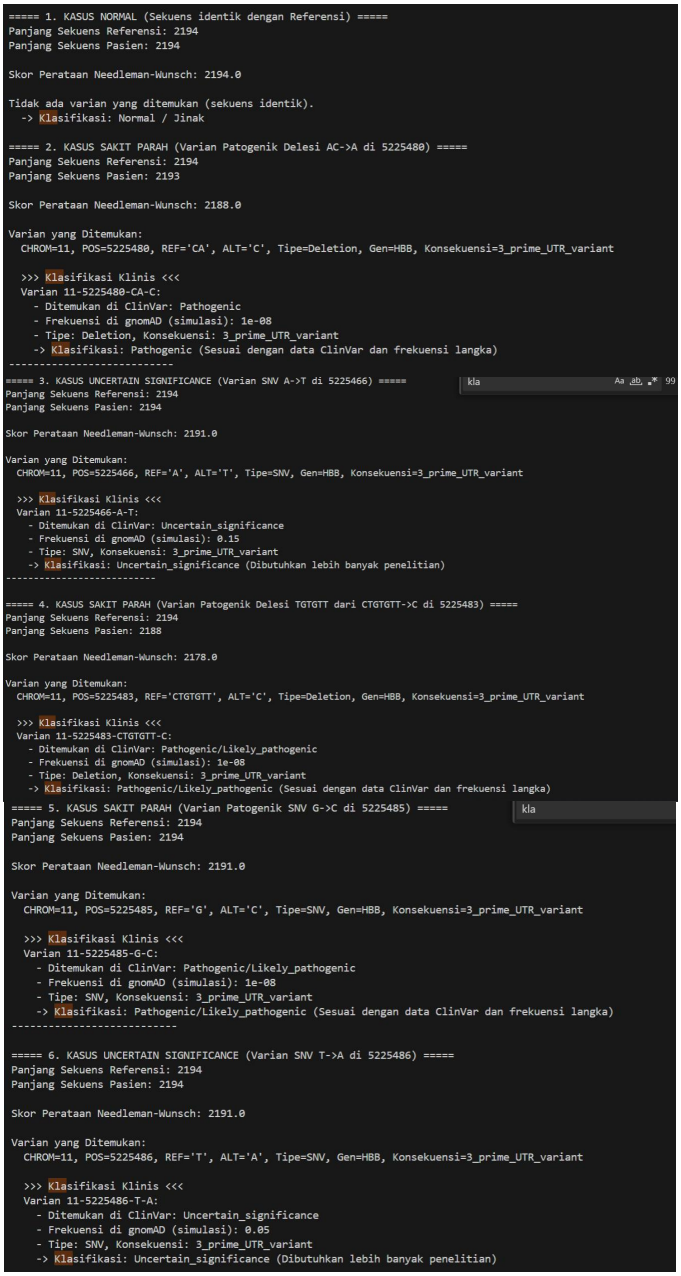
ACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGACACC
ATGGTGCATCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGG
CAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGTTGGTA
TCAAGGTTACAAGACAGGTTTAAGGAGACCAATAGAAACTGGGCATGTG
GAGACAGAGAAGACTCTTGGGTTTCTGATAGGCACTGACTCTCTCTGCCT
ATTGGTCTATTTTCCCACCCTTAGGCTGCTGGTGGTCTACCCTTGGACCC
AGAGGTTCTTTGAGTCCTTTGGGGATCTGTCCACTCCTGATGCTGTTATG
GGCAACCCTAAGGTGAAGGCTCATGGCAAGAAAGTGCTCGGTGCCTTTA
GTGATGGCCTGGCTCACCTGGACAACCTCAAGGGCACCTTTGCCACACT
GAGTGAGCTGCACTGTGACAAGCTGCACGTGGATCCTGAGAACTTCAGG
GTGAGTCTATGGGACGCTTGATGTTTTCTTTCCCCTTCTTTTCTATGGTTA
AGTTCATGTCATAGGAAGGGGATAAGTAACAGGGTACAGTTTAGAATGG
GAAACAGACGAATGATTGCATCAGTGTGGAAGTCTCAGGATCGTTTTAG
TTTCTTTTATTTGCTGTTCATAACAATTGTTTTCTTTTGTTTAATTCTTGCT
TTCTTTTTTTTTCTTCTCCGCAATTTTTACTATTATACTTAATGCCTTAACA
TTGTGTATAACAAAAGGAAATATCTCTGAGATACATTAAGTAACTTAAA
AAAAAACTTTACACAGTCTGCCTAGTACATTACTATTTGGAATATATGTG
TGCTTATTTGCATATTCATAATCTCCCTACTTTATTTTCTTTTATTTTTAAT
TGATACATAATCATTATACATATTTATGGGTTAAAGTGTAATGTTTTAAT
ATGTGTACACATATTGACCAAATCAGGGTAATTTTGCATTTGTAATTTTA
AAAAATGCTTTCTTCTTTTAATATACTTTTTTGTTTATCTTATTTCTAATAC
TTTCCCTAATCTCTTTCTTTCAGGGCAATAATGATACAATGTATCATGCCT
CTTTGCACCATTCTAAAGAATAACAGTGATAATTTCTGGGTTAAGGCAAT
AGCAATATCTCTGCATATAAATATTTCTGCATATAAATTGTAACTGATGT
AAGAGGTTTCATATTGCTAATAGCAGCTACAATCCAGCTACCATTCTGCT
TTTATTTTATGGTTGGGATAAGGCTGGATTATTCTGAGTCCAAGCTAGGC
CCTTTTGCTAATCATGTTCATACCTC
TTATCTTCCTCCCACAGCTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCC
CATCACTTTGGCAAAGAATTCACCCCACCAGTGCAGGCTGCCTATCAGA
AAGTGGTGGCTGGTGTGGCTAATGCCCTGGCCCACAAGTATCACTAAGC
TCGCTTTCTTGCTGTCCAATTTCTATTAAAGGTTCCTTTGTTCCCTAAGTC

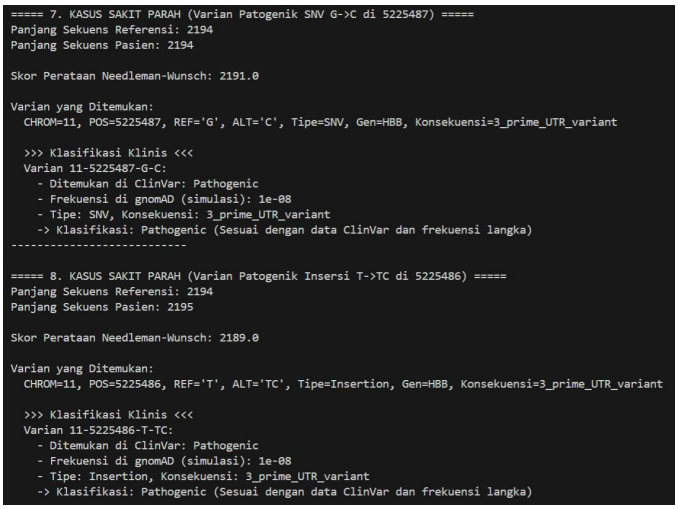
AAGGGCCTTGAGCATCTGGATTCTGCCTAATAAAAAACATTTATTTTCAT
TGCAA

Sekuens diatas kemudian dimodifikasi secara sengaja untuk menciptakan skenario-skenario yang mengandung varian spesifik, seperti insersi, delesi, atau substitusi. Tujuan dari modifikasi ini adalah untuk mengevaluasi apakah algoritma dapat secara berhasil mendeteksi dan memetakan varian-varian tersebut pada posisi yang sesuai. Berikut testcasenya:

1. Kasus Normal: Sekuens pasien identik dengan sekuens referensi.
2. Kasus Sakit Parah (Delesi AC->A): Sekuens pasien mengandung delesi satu basa (A) dari sekuens referensi 'CA' menjadi 'C' pada posisi genomik 5225480.
3. Kasus *Uncertain Significance* (SNV A->T): Sekuens pasien mengandung substitusi basa 'A' menjadi 'T' pada posisi genomik 5225466.
4. Kasus Sakit Parah (Delesi TGTGTT): Sekuens pasien mengandung delesi enam basa ('TGTGTT') dari sekuens referensi 'CTGTGTT' menjadi 'C' pada posisi genomik 5225483.
5. Kasus Sakit Parah (SNV G->C): Sekuens pasien mengandung substitusi basa 'G' menjadi 'C' pada posisi genomik 5225485.
6. Kasus *Uncertain Significance* (SNV T->A): Sekuens pasien mengandung substitusi basa 'T' menjadi 'A' pada posisi genomik 5225486.
7. Kasus Sakit Parah (SNV G->C): Sekuens pasien mengandung substitusi basa 'G' menjadi 'C' pada posisi genomik 5225487.
8. Kasus Sakit Parah (Insersi T->TC): Sekuens pasien mengandung insersi basa 'C' setelah 'T' pada posisi genomik 5225486.

Setelah melakukan testing menggunakan algoritma yang sudah diimplementasikan didapat hasil sebagai berikut:





Gambar Hasil Percobaan

Berdasarkan serangkaian pengujian pada delapan kasus yang telah dimodifikasi, dapat disimpulkan bahwa implementasi algoritma Needleman-Wunsch ini berhasil secara akurat mendeteksi dan memetakan semua varian genetik (Substitusi, Insersi, dan Delesi) yang disimulasikan. Keberhasilan ini mengindikasikan bahwa algoritma yang diterapkan dapat secara efektif mengetahui posisi terjadinya miss/match dan mengidentifikasi secara akurat variansi yang cocok dengan posisi miss/match tersebut. Selain itu algoritma ini juga mampu memberi informasi berupa seberapa buruk mutasi gen yang ada pada pasien yang ditandai dengan jauhnya nilai kedekatannya antara referensi dengan pattern DNA dari pasien itu sendiri. Namun, di balik keberhasilan deteksi varian, implementasi algoritma ini masih memiliki beberapa keterbatasan yang signifikan, terutama terkait efisiensi komputasi. Meskipun akurat, proses penjajaran untuk sekuens besar menunjukkan waktu komputasi yang relatif lambat untuk masing-masing kasus. Hal ini karena algoritma Needleman-Wunsch memiliki kompleksitas waktu O(nm), di mana n dan m adalah panjang dari kedua sekuens. Untuk sekuens yang sangat panjang, ini dapat menyebabkan peningkatan waktu pemrosesan yang substansial, menjadikannya kurang praktis untuk analisis genom skala besar.

IV. KESIMPULAN

Algoritma Needleman-Wunsch terbukti menjadi alat yang sangat berguna, terutama dalam bidang bioinformatika. Berdasarkan serangkaian percobaan yang telah dilakukan, algoritma ini mampu secara tepat mengidentifikasi setiap varian genetik pada sekuens pasien. Keberhasilan ini semakin diperkuat oleh modifikasi pada matriks penskoran, yang tidak lagi menggunakan bobot statis melainkan mempertimbangkan relevansi biologis. Dengan memberikan bobot yang berbeda untuk *match*, transisi, dan transversi, algoritma ini mampu memberikan informasi yang lebih akurat bagi penganalisis mengenai sejauh mana perubahan pada gen pasien.

Walaupun begitu, perlu diakui bahwa implementasi algoritma ini masih memerlukan pengembangan lebih lanjut, khususnya dalam hal efisiensi waktu untuk data berukuran besar. Kompleksitas waktu O(nm) dari Needleman-Wunsch menjadikannya kurang praktis untuk analisis genom skala penuh yang melibatkan jutaan hingga miliaran basa. Tidak bisa dipungkiri, implementasi ini hanyalah sebuah prototipe. Meski berhasil menunjukkan konsep dan fungsionalitasnya solusi ini belum sepenuhnya realistis jika dibandingkan dengan alat bioinformatika komersial atau riset yang dirancang untuk menangani volume data genomik yang masif dengan kecepatan tinggi.

REFERENCES

[1] Jiang, X., Fu, X., Dong, G., & Li, H. (2017). "Research on Pairwise Sequence Alignment Needleman-Wunsch Algorithm." *141(Icmmcce), 1041–1046.* https://doi.org/10.2991/icmmcce-17.2017.187. [Diakses 24 Juni 2025]

[2] National Center for Biotechnology Information (NCBI) – ClinVar. https://www.ncbi.nlm.nih.gov/clinvar/ [Diakses 24 Juni 2025]

[3] Genome Aggregation Database (gnomAD). https://gnomad.broadinstitute.org/ [Diakses 24 Juni 2025]

[4] Needleman, Saul B., and Wunsch, Christian D. (1970). "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins." *Journal of Molecular Biology*, 48(3), 443–453.Needleman_Wunsch_JMB_70_Global_alignment.pdf [Diakses Pada 24 Juli 2025]

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Ferdinand Gabe Tua Sinaga 13523051