

Implementasi dan Analisis Kinerja Algoritma Boyer-Moore Untuk Deteksi Motif Epigenetik MTBC Dalam Penyakit TBC

Jonathan Alveraldo Bangun - 13524120

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: alveraldo.jonathan@gmail.com , 13524120@std.stei.itb.ac.id

Abstract—Identifikasi motif metilasi DNA epigenetik sangat krusial untuk memetakan virulensi dan resistensi obat pada galur *Mycobacterium tuberculosis complex* (MTBC). Namun, penelusuran motif ini pada data genomik berskala besar menuntut beban komputasi yang tinggi jika menggunakan metode pencarian linear. Makalah ini mengimplementasikan algoritma pencocokan string Boyer-Moore untuk menelusuri motif epigenetik MTBC pada 1.000.000 karakter data genomik sintetis. Kinerja algoritma dievaluasi dengan membandingkan waktu eksekusinya terhadap algoritma Naïve (Brute-Force) pada variasi panjang motif (6 hingga 150 karakter nukleotida)

Keywords—Boyer-Moore, Data Genomik, Epigenetik, Metilasi DNA, MTBC, Pencocokan String.

I. PENDAHULUAN

Tuberkulosis (TBC) merupakan salah satu krisis kesehatan masyarakat global yang paling mematikan. Agen penyebab penyakit ini, *Mycobacterium tuberculosis*, merupakan bagian dari kelompok bakteri yang sangat berkerabat erat yang dikenal sebagai *Mycobacterium tuberculosis complex* (MTBC). Studi genomik mengungkapkan bahwa anggota MTBC memiliki tingkat kesamaan nukleotida mencapai ~99%.

Meskipun secara genetik hampir identik, galur-galur MTBC (yang terbagi ke dalam setidaknya 7 lineage utama) menunjukkan heterogenitas fenotipik yang luar biasa dalam hal tingkat virulensi, respons terhadap inang, dan kemampuan resistensi obat. Penelitian terbaru yang memanfaatkan teknologi Next-Generation Sequencing (NGS) dan Single-Molecule Real-Time (SMRT) mengungkapkan bahwa sifat "unik" antar galur MTBC tidak hanya dipengaruhi oleh mutasi genetik, melainkan juga oleh mekanisme regulasi epigenetik dan modifikasi transkriptom. Salah satu mekanisme epigenetik terpenting adalah metilasi DNA.

Bakteri MTBC memanfaatkan enzim metiltransferase (MTase) untuk memetilasi motif string DNA spesifik, yang secara langsung memengaruhi regulasi genetik metabolisme, kemampuan bertahan hidup dalam kondisi hipoksia, dan kerentanan terhadap antibiotik. Untuk memetakan diversitas epigenetik ini secara klinis, para peneliti bioinformatika harus memindai jutaan pasangan basa DNA pasien guna menemukan

motif metilasi spesifik galur tersebut. Proses pencarian (string matching) ini memerlukan algoritma dengan efisiensi tinggi, karena pendekatan linear seperti algoritma Naïve akan memicu bottleneck komputasional.

Makalah ini mengimplementasikan Algoritma Boyer-Moore yang memiliki karakteristik heuristik pelompatan karakter sub-linear untuk menelusuri secara cepat keberadaan motif-motif epigenetik MTBC klinis di dalam rantai data genomik berskala masif.

II. DASAR TEORI

A. Tuberkulosis (TBC)

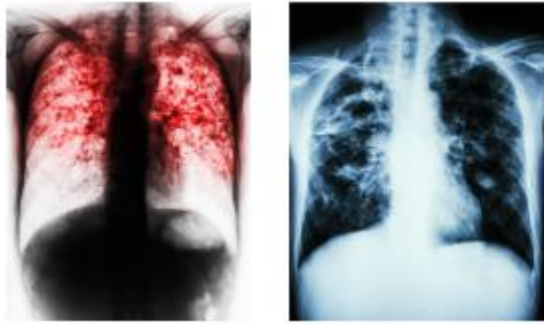
Tuberkulosis (TBC) adalah penyakit infeksi menular dan mematikan yang utamanya menyerang jaringan paru-paru (TBC paru), meskipun dalam beberapa kasus dapat menyebar ke luar paru-paru seperti kelenjar getah bening, tulang, hingga selaput otak (TBC ekstra-paru).

Penyakit ini menular secara aerogen melalui percikan dahak atau droplet nuclei di udara. Ketika penderita aktif batuk, bersin, atau berbicara. Penderita yang terinfeksi bakteri TBC sering kali menunjukkan gejala utama berupa batuk berdarah yang berlangsung lebih dari dua hingga tiga minggu, yang kerap kali disertai dengan bercak darah.

Gejala penyerta klinis lainnya meliputi demam ringan yang berkepanjangan, penurunan berat badan secara drastis tanpa sebab yang jelas, kelelahan kronis (malaise), nyeri dada, serta keringat berlebih pada malam hari (night sweats) meskipun tanpa melakukan aktivitas fisik berat.

Secara epidemiologis, penyakit ini masih menjadi salah satu penyebab utama mortalitas global akibat agen infeksi tunggal, dengan kawasan negara berkembang termasuk Indonesia sebagai salah satu penyumbang beban kasus terbesar di dunia.

Deteksi dini dan pengobatan yang komprehensif sangat krusial, karena penundaan diagnosis sering kali berujung pada kerusakan jaringan paru yang parah dan memicu penyebaran transmisi yang masif di tengah masyarakat.



Gambar 1. Pemeriksaan bakteriologis penyakit TBC
Sumber: <https://yki4tbc.org/tuberkulosis-tbc/>

Tantangan terbesar dalam penanganan TBC modern adalah munculnya fenomena resistensi obat, seperti MDR-TB (Multi-Drug Resistant Tuberculosis) dan XDR-TB (Extensively Drug-Resistant Tuberculosis). Kompleksitas ini memaksa dunia medis untuk beralih dari sekadar pemeriksaan mikroskopis konvensional berupa tes BTA (Basil Tahan Asam) menuju diagnosis molekuler tingkat lanjut. Pendekatan analisis genomik seperti Next-Generation Sequencing (NGS) dan sekuensing SMRT kini digunakan secara luas untuk memetakan (sequencing) profil DNA secara utuh dari bakteri penyebab TBC.

Melalui penelusuran algoritmik dan komputasional terhadap untaian DNA ini, praktisi medis dapat mengetahui secara presisi galur patogen, membedakan tingkat virulensinya, hingga mendeteksi genotipe spesifik yang menandakan kekebalan terhadap antibiotik sejak awal pemeriksaan

B. *Mycobacterium tuberculosis complex (MTBC)*

Bakteri penyebab TBC tergabung dalam sebuah kelompok yang sangat berkerabat erat yang dikenal sebagai *Mycobacterium tuberculosis complex (MTBC)*. Kelompok ini secara komprehensif mencakup *M. tuberculosis* yang merupakan agen infeksius utama pada inang manusia, beserta galur lain seperti *M. bovis*, *M. canetti*, *M. africanum*, hingga galur yang beradaptasi pada hewan. Secara genetik, seluruh galur di dalam kompleks ini dikonfirmasi sangat identik, dengan persentase kemiripan nukleotida mencapai ~99%.

Namun, terlepas dari identitas DNA komparatif tersebut, para ahli biologi evolusioner dan medis mengidentifikasi heterogenitas fenotipik yang luar biasa yang memengaruhi tingkat penyebaran (transmisibilitas), tingkat keparahan penyakit (virulensi), hingga adaptasi terhadap obat. Galur MTBC secara filogenetik dibagi ke dalam klade modern dan klade purba (ancient) yang mencakup setidaknya 9 galur utama, meliputi:

- Indo-Oceanic (Lineage 1),
- East-Asian (Lineage 2, mencakup famili Beijing yang dikenal hipervirulen di tingkat global),
- East-African Indian (Lineage 3),
- Euro-American (Lineage 4),
- *M. africanum* West Africa 1 (Lineage 5),

- *M. africanum* West Africa 2 (Lineage 6),
- Ethiopia (Lineage 7)

Methylation motif	Function	Strain/Lineage	Consequence of MTBC diversity
CTCCAG	Metabolic regulatory proteins	Lineage 4	Down regulation of gene expression
CTCCAG	Role in transcription	Lineage 2	Decreased gene expression and hypoxia survival
CACGCAG	Enrichment of fatty acid and Polyketide synthesis pathways	Lineage 1	Increased virulence
GATNNNNRTAC	Enrichment of cell and plasma membrane associated genes	Lineage 4	Decreased antibiotic susceptibility
GTAYN4ATC	Respiration and drug resistance-related genes	Strain 11,826	Altered drug resistances and susceptibility
CTCGAG	Cell wall biosynthesis and modification	<i>M. smegmatis</i> strain mc ² 155	Acquisition of resistance to anti-mycobacterial factors

Gambar 2. Pola metilasi dari berbagai garis keturunan MTBC
Sumber:

<https://www.sciencedirect.com/science/article/pii/S2405844023098195>

Gambar 2 diatas merupakan pemetaan biologis yang menguraikan hubungan sebab-akibat antara pola susunan DNA spesifik (motif metilasi) dengan perilaku fenotipik berbagai galur bakteri *Mycobacterium tuberculosis complex (MTBC)*. Kolom motif metilasi berisi untaian sekuens nukleotida yang bertindak sebagai target pencarian esensial di dalam sistem bioinformatika, sedangkan kolom lainnya merincikan galur bakteri yang membawa motif tersebut beserta dampak klinisnya secara nyata. Pemetaan penanda biologis (*biomarker*) ini mendasari urgensi penggunaan algoritma komputasi pencocokan pola yang efisien, mengingat praktisi medis harus memindai jutaan pasangan basa DNA pada data genomik pasien untuk menemukan motif spesifik galur tersebut tanpa memicu hambatan komputasional (*bottleneck*).

Berdasarkan rincian pada gambar 2, identifikasi motif-motif genetik tersebut memberikan informasi klinis yang krusial terkait tingkat keparahan infeksi dan adaptasi patogen terhadap penanganan medis. Sebagai contoh, penemuan motif spesifik CACGCAG pada galur Lineage 1 menandakan bahwa bakteri tersebut memperkaya jalur sintesis asam lemak yang berdampak pada peningkatan virulensi atau keparahan penyakit. Di sisi lain, keberhasilan sistem mendeteksi motif yang lebih panjang seperti GATNNNNRTAC pada galur Lineage 4 menjadi peringatan dini bahwa bakteri tersebut mengalami penurunan kerentanan terhadap antibiotik.

Pemetaan motif-motif metilasi yang sangat spesifik ini di dalam data sampel genomik pasien adalah fondasi krusial bagi sistem bioinformatika untuk melacak perilaku dan evolusi molekuler bakteri TBC secara cepat dan akurat

C. *Algoritma Naïve (Brute-Force)*

Algoritma Naive atau Brute-Force merupakan metode paling fundamental dan intuitif di dalam domain pencocokan pola (pattern matching). Pendekatan ini memeriksa setiap posisi probabilitas di dalam teks (T) untuk menguji apakah pola sasaran (P) dapat ditemukan. Secara mekanis, algoritma

ini beroperasi dengan mensejajarkan karakter pertama dari pola tepat di indeks awal teks, lalu melakukan evaluasi komparasi karakter satu per satu secara sekuensial dari arah kiri ke kanan. Jika seluruh karakter pola terkonfirmasi sama persis dengan potongan teks, maka indeks kemunculan tersebut dikembalikan sebagai hasil temuan.

Akan tetapi, kelemahan utama algoritma ini terletak pada strategi penanganan kegagalannya (mismatch handling) dan ketiadaan memori internal. Apabila pada iterasi pemeriksaan terjadi satu saja ketidakcocokan karakter meskipun ketidakcocokan itu terjadi di karakter paling akhir dari pola setelah rentetan panjang karakter yang cocok algoritma ini secara kaku akan membuang semua informasi pencocokan yang sudah dilakukan. Algoritma sekadar menggeser jendela pencarian sejauh tepat satu indeks ke kanan pada teks (bergeser 1 posisi setiap kalinya), lalu mengulangi kembali proses perbandingan karakter secara membabi buta dari awal pola.

Secara performa, algoritma brute force dapat beroperasi dengan cukup wajar jika himpunan alfabet penyusun teks berukuran besar (misalnya alfabet Latin A..Z, a..z, atau kombinasi alfanumerik panjang), karena probabilitas ketidakcocokan dini sangat tinggi sehingga iterasi dapat cepat dihentikan. Namun, metode ini menjadi sangat lambat dan tidak proporsional saat memproses teks dengan ragam alfabet yang sempit.

Dalam bioinformatika, data genomik hanya memiliki ruang lingkup empat karakter nukleotida murni, yakni himpunan $\Sigma = \{A, C, G, T\}$. Kombinasi karakter yang repetitif memperbesar peluang terjadinya "kecocokan parsial palsu" (false partial match). Algoritma akan sering kali terjebak memvalidasi banyak huruf sebelum menemui kegagalan di tengah jalan, memicu redundansi operasional komputasi yang parah.

Kompleksitas waktu penyelesaian (time complexity) algoritma Naive pada kasus terbaik (best case) adalah $O(n)$, pada kasus terburuk (worst case), total operasi perbandingan akan meledak menuju $m(n-m+1) = O(nm)$.

Misalkan terdapat teks "janganmaumasukelefem" dan target pencarian "masukel", maka:

```
janganmaumasukelefem
|
masukel
masukel
masukel
masukel
masukel
masukel
masukel
masukel
masukel
masukel
```

Pola ditemukan.

D. Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan salah satu penemuan komputasional fundamental paling revolusioner di dalam penyelesaian kompleksitas masalah pencocokan pola mutakhir.

Algoritma spesifik ini diformulasikan secara kolaboratif oleh ilmuwan komputer Robert S. Boyer dan J. Strother Moore dan resmi dipublikasikan pada jurnal saintifik Communications of the ACM tahun 1977 di bawah naungan publikasi "A Fast String Searching Algorithm".

Lahir dan dirancang sebagai kerangka pemecahan (solvers) fundamental guna secara radikal mempercepat rentang waktu iterasi kalkulatif ekstraksi komputasi informasi teks berskala data masif, pendekatan analitik Boyer-Moore sukses mengubah tata struktur paradigma konvensional sistem scanning linear algoritma terdahulu, menumbangkan batasan-batasan kecepatan metode pencocokan yang konvensional.

Algoritma pencocokan pola Boyer-Moore didasarkan pada dua teknik utama. Kunci utama efisiensi algoritma ini dipecah menjadi dua tahapan, yakni fase pra-pemrosesan dan fase pencarian.

1. The looking-glass technique (Fase Pencarian) Algoritma ini menemukan pola P di dalam teks T dengan bergerak mundur (dari kanan ke kiri) melalui P, dimulai pada bagian akhirnya.
2. The character-jump technique (Fase Pra-pemrosesan dan Pelompatan) Ketika terjadi ketidakcocokan pada $T[i] \neq x$, di mana karakter pada pola $P[j]$ tidak sama dengan $T[i]$, terdapat 3 kemungkinan kasus yang dicoba secara berurutan:
 - Kasus 1: Jika P mengandung x di suatu tempat, maka coba geser P ke kanan untuk mensejajarkan kemunculan terakhir x di P dengan $T[i]$.
 - Kasus 2: Jika P mengandung x di suatu tempat, tetapi menggeser ke kanan pada kemunculan terakhir tidak memungkinkan, maka geser P ke kanan sejauh 1 karakter ke $T[i+1]$.
 - Kasus 3: Jika kasus 1 dan 2 tidak berlaku (x sama sekali tidak ada di dalam P), maka geser P untuk mensejajarkan $P[0]$ dengan $T[i+1]$.

Untuk mengimplementasikan teknik ini, algoritma melakukan pra-pemrosesan pola P dan alfabet A untuk membangun fungsi kemunculan terakhir (last occurrence function), $L()$.

Fungsi ini memetakan semua huruf ke dalam bilangan bulat (indeks terbesar i sedemikian sehingga $P[i] = x$, atau -1 jika tidak ada). $L()$ umumnya disimpan sebagai array atau tabel.

Misalkan terdapat teks “janganmaumasukelefem” dan pola sasaran “masukel”, maka:

Langkah 1:

```
janganmaumasukelefem
      |
masukel
```

Dengan fungsi last occurrence, pola bergeser sejauh 6 karakter

Langkah 2:

```
janganmaumasukelefem
      |
      masukel
```

Dengan fungsi last occurrence, pola bergeser sejauh 3 karakter

Langkah 3:

```
janganmaumasukelefem
      |
      masukel
```

Pola ditemukan.

III. IMPLEMENTASI

Rancangan eksperimen komputasional diimplementasikan pada perangkat lunak pengujian untuk mengukur kinerja wall-clock time dari rutinitas pencarian sekuens epigenetik MTBC. Tidak ada kode sumber yang dilampirkan dalam makalah ini sebagai pemenuhan kriteria teknis spesifikasi tugas.

A. Pemodelan Data Genomik dan Motif Sasaran

Sebuah rantai Teks (T) berukuran 1.000.000 nukleotida dibangkitkan untuk mensimulasikan lingkungan sekuensing genomik utuh. Sasaran pencarian (P) divariasikan untuk mencakup motif metilasi dasar yang pendek hingga sekuens regulasi atau penyandi MTase yang jauh lebih panjang. Target ini secara sengaja diinjeksikan pada posisi data ke-850.000 guna menghasilkan ruang penelusuran (search space) yang cukup signifikan dalam perekaman selisih waktu performa algoritma.

B. Algoritma Naive (Brute-Force)

Algoritma Brute-Force diimplementasikan menggunakan dua buah struktur perulangan (*loop*) bersarang:

- Perulangan luar (i)
Bertugas menggeser jendela pencarian sepanjang teks (mulai dari indeks ke-0 hingga indeks $n - m$).
- Perulangan dalam (j)
Untuk setiap posisi i , algoritma mulai mencocokkan karakter dari pola dengan teks secara berurutan dari kiri ke kanan.

• Logika Kegagalan

Jika ditemukan karakter teks $\text{Text}[i + j]$ yang tidak sama dengan karakter pola $\text{Pattern}[j]$, perulangan dalam seketika dihentikan. Hal ini memicu perulangan luar untuk menambah nilai i sebesar 1. Artinya, pola digeser secara kaku sejauh tepat satu karakter ke kanan tanpa memedulikan informasi kegagalan sebelumnya. Jika variabel j berhasil mencapai nilai panjang pola (m), berarti seluruh karakter telah cocok secara sempurna.

```
procedure BruteForceSearch(Text, Pattern)
  n ← length(Text)
  m ← length(Pattern)
  for i ← 0 to (n - m) do
    j ← 0
    while (j < m) and (Text[i + j] == Pattern[j]) do
      j ← j + 1
    end while
    if j == m then
      return i // Posisi kecocokan ditemukan
    end if
  end for
  return -1 // Tidak ditemukan
end procedure
```

C. Algoritma Boyer-Moore

Fungsi BuildLast merupakan fungsi pra-pemrosesan (sebelum pencarian dimulai) yang bertugas membentuk tabel rujukan cepat.

- Fungsi ini mengalokasikan sebuah larik (*array*) bernama last berukuran 128/256 (merekpresentasikan nilai karakter ASCII standar). Pada awalnya, seluruh sel di dalam larik diisi dengan nilai konstanta -1.
- Algoritma melakukan satu kali iterasi berjalan dari indeks 0 hingga akhir pola ($m-1$). Setiap karakter yang dijumpai akan memperbarui nilai indeksinya di dalam larik last.
- Karena iterasi berjalan dari kiri ke kanan, jika terdapat karakter ganda, larik secara otomatis akan menimpa dan menyimpan indeks posisi kemunculannya yang paling kanan. Hasil akhirnya adalah sebuah rujukan instan $O(1)$ untuk mencari tahu di mana sebuah karakter terakhir kali muncul di dalam pola.

```
procedure BuildLast(Pattern, out last)
  for i ← 0 to 127 do
```

```

    last[i] ← -1
end for
for i ← 0 to length(Pattern) - 1 do
    last[Pattern[i]] ← i
end for
end procedure

```

```

    j ← j - 1
end if
else // (Pergeseran berbasis tabel)
    lo ← last[Text[i]]
    i ← i + m - min(j, 1 + lo)
    j ← m - 1
end if
while i ≤ n - 1
    return -1 // Tidak ditemukan
end procedure

```

Fungsi pencarian utama BoyerMooreSearch menerapkan teknik:

- Jendela pencarian disandingkan di awal teks, namun indeks pembanding (variabel i pada teks dan j pada pola) diatur berada di paling kanan pola (indeks m - 1). Pemeriksaan berjalan mundur ke kiri.
- Saat karakter tidak cocok (terjadi *mismatch*), algoritma akan melihat karakter pada teks yang menyebabkan ketidakcocokan tersebut (*bad character*). Algoritma mengambil posisi kemunculan terakhir karakter itu dari tabel last yang sudah dibuat (yakni $lo = last[Text[i]]$).
- Indeks teks i akan digeser ke kanan menggunakan formula: $i + m - \min(j, 1 + lo)$. Penggunaan fungsi minimum (min) di sini sangat vital. Fungsi ini menjamin bahwa algoritma akan memilih nilai terkecil antara indeks ketidakcocokan (j) dengan letak kemunculan terakhir karakter ditambah satu ($1 + lo$). Perlindungan ini memastikan posisi pergeseran tidak pernah bernilai negatif (pola tidak akan pernah bergerak mundur melintasi teks yang sudah diperiksa).

```

procedure BoyerMooreSearch(Text, Pattern)
    declare last as array of integer
    BuildLast(Pattern, last)
    n ← length(Text)
    m ← length(Pattern)
    i ← m - 1
    if i > n - 1 then
        return -1;
    end if
    j ← m - 1
    do
        if Pattern[j] == Text[i] then
            if j == 0 then
                return i // Posisi kecocokan ditemukan
            else // (Mundur ke kiri)
                i ← i - 1
            end if
        end if
        j ← j - 1
    until j < 0
end procedure

```

IV. ANALISIS

Waktu pencarian pada kasus terburuk (worst case) untuk algoritma Boyer-Moore diproyeksikan pada $O(nm+A)$. Secara teoretis, literatur menyebutkan algoritma ini berjalan lebih lambat Ketika berhadapan dengan alfabet yang sempit.

Mengingat set data genomik hanya dibangun oleh 4 jenis alfabet pembentuk basa nukleotida ({A, C, G, T}), pengujian empiris ini bertujuan untuk menginvestigasi apakah algoritma Boyer-Moore tetap relevan dan mampu mempertahankan keunggulan performa pelompatan indeksnya dibandingkan pendekatan Naive. Evaluasi dieksekusi menggunakan bahasa Java pada teks sepanjang $n = 1.000.000$ karakter dengan memvariasikan ukuran panjang pola (m).

A. Rekapitulasi Kinerja Komputasi

Tabel 1. Tabel Rekapitulasi Kinerja Komputasi

Data Genomik (n)	Target Epigenetik CTCCAG (m)	Waktu Naïve (Brute-Force)	Waktu Boyer-Moore	Kelipatan Kecepatan
1.000.000	6	0.00019	0.00019	1.0x
1.000.000	30	0.01699	0.01241	1.4x
1.000.000	60	0.00769	0.00313	2.5x
1.000.000	150	0.01250	0.00275	4.6x

B. Reduksi Komputasi

Data empiris pada Tabel 1 memberikan validasi komputasional yang sangat menarik mengenai perilaku kedua algoritma ketika dihadapkan pada alfabet biner buatan alam. Pada scenario pencarian motif epigenetik primer yang sangat pendek ($m = 6$), waktu komputasi antara algoritma Naive dan Boyer-Moore bersaing ketat dan menyentuh angka yang sama (0.00019 detik).

Hal ini sangat wajar terjadi karena biaya **overhead** (waktu tambahan) yang dibutuhkan oleh Boyer-Moore untuk menginisialisasi dan membangun Last Occurrence Table di

awal eksekusi menyeimbangkan keuntungan kecepatan dari pelompatan karakter yang rentangnya masih sangat sempit.

Akan tetapi, manifestasi sejati dari sifat efisiensi sub-linear terlihat jelas ketika kompleksitas ukuran motif genetik terus diperbesar. Ketika rentang pola (m) bertambah menjadi 30, 60, hingga mencapai 150 karakter, faktor percepatan (speedup) yang dihasilkan algoritma Boyer-Moore meningkat secara agresif, beranjak dari 1,4x, menuju 2,5x, hingga memuncak pada rasio 4,6x lipat lebih cepat. Pada pencarian sekuens sepanjang 150 karakter, waktu eksekusi Boyer-Moore secara konsisten merosot tajam menjadi hanya 0.00275 detik.

Anomali performa ini membuktikan bahwa panjangnya target pencarian diolah menjadi sebuah keuntungan besar bagi fungsi heuristik; algoritma memanfaatkannya untuk melompati ratusan blok nukleotida yang tidak relevan secara dinamis dan drastis.

Eksperimen ini menegaskan bahwa meskipun bekerja pada ekosistem ber-alfabet sempit, strategi pencarian mundur dari kanan-ke-kiri merupakan arsitektur sequence alignment yang baik untuk bioinformatika modern.

V. KESIMPULAN

Eksperimen ini memberikan validasi empiris bahwa algoritma Boyer-Moore adalah solusi yang tangguh dalam penelusuran sekuens nukleotida berskala tinggi. Penerapannya pada deteksi penanda metilasi epigenetik kelompok *Mycobacterium tuberculosis* complex (seperti galur Lineage 4 pembawa motif CTCCAG) menunjukkan fleksibilitas algoritma ini; berjalan efisien pada pencarian biomarker pendek, dan memanifestasikan keunggulan performa mutlak pada identifikasi struktur genetik fungsional panjang.

Karakteristik komputasi yang sub-linear ini memungkinkan sistem perangkat lunak untuk menekan beban sequence alignment secara eksponensial (mencapai percepatan 4,6x lipat pada target berukuran besar), sehingga sangat direkomendasikan untuk menunjang infrastruktur sistem diagnosis genomik secara tangkas dan akurat di masa depan.

VI. LAMPIRAN

Link Github:

<https://github.com/jonvernal/Makalah-StiMa>

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa
2. Bapak Dr. techn. Muhammad Zuhri Catur Candra, S.T, M.T., Ibu Tricya Esterina Widagdo, S.T., M.Sc., Ibu Dr. Fazat Nur Azizah, S.T., M.Sc. selaku dosen-dosen pengampu mata kuliah Strategi Algoritma
3. Keluarga Bangun
4. Keluarga Sapiipa2
5. Keluarga KMP Pencemooh
6. Keluarga Medik KAT ITB "Nithvara Padyaseva"
7. Keluarga Basety
8. Keluarga Besar Nawla
9. Keluarga LFM ITB "Krumunan"
10. Keluarga Kesenatoran HMIF ITB "MenataMakna"

REFERENCES

- [1] X. He dan S. Wang, "DNA consensus sequence motif for binding response regulator PhoP, a virulence regulator of *Mycobacterium tuberculosis*," *Biochemistry*, vol. 53, no. 51, pp. 8008-8020, 2014. [Online]. <https://pubmed.ncbi.nlm.nih.gov/25434965/> [Diakses: 18 Juni 2026]
- [2] R. Munir, "Pencocokan string," slide mata kuliah Strategi Algoritma, Program Studi Teknik Informatika, Institut Teknologi Bandung, 2026. [Online]. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/23-Pencocokan-string-\(2026\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/23-Pencocokan-string-(2026).pdf) [Diakses: 17 Juni 2026]
- [3] Yasa Karsa Insani, "Apa itu Tuberkulosis / TBC?" [Online]. <https://yki4tbc.org/tuberkulosis-tbc/> [Diakses: 18 Juni 2026]
- [4] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge, UK: Cambridge University Press, 1997.
- [5] N. E. Mvubu, "Mycobacterium tuberculosis complex molecular networks and their regulation: Implications of strain heterogeneity on epigenetic diversity and transcriptome regulation," *Heliyon*, 2023. [Online]. <https://www.sciencedirect.com/science/article/pii/S2405844023098195> [Diakses: 19 Juni 2026]
- [6] GeeksforGeeks, "Boyer Moore Algorithm for Pattern Searching," 2025. [Online]. <https://www.geeksforgeeks.org/dsa/boyer-moore-algorithm-for-pattern-searching/> [Diakses: 17 Juni 2026]
- [7] Z. Ditse, M. H. Lamers, dan D. F. Warner, "DNA Replication in *Mycobacterium tuberculosis*," *Microbiology Spectrum*, vol. 5, no. 2, 2017. [Online]. <https://journals.asm.org/doi/10.1128/microbiolspec.tb12-0027-2016> [Diakses: 18 Juni 2026]
- [8] N. C. Jones dan P. A. Pevzner, *An Introduction to Bioinformatics Algorithms*. Cambridge, MA: MIT Press, 2004.
- [9] D. Gusfield, "The Boyer-Moore Algorithm," catatan mata kuliah CS224, University of California, Davis, 2011. [Online]. <https://www.cs.ucdavis.edu/~gusfield/cs224f11/bnotes.pdf> [Diakses: 19 Juni 2026]
- [10] D. Bespyatykh, J. Bespyatykh, I. Mokrousov, dan E. Shitikov, "A Comprehensive Map of *Mycobacterium tuberculosis* Complex Regions of Difference," *mSphere*, vol. 6, no. 4, 2021. [Online]. <https://journals.asm.org/doi/10.1128/msphere.00535-21> [Diakses: 19 Juni 2026]

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.

Bandung, 19 Juni 2026



Jonathan Alveraldo Bangun 13524120