BAB I
PENDAHULUAN

1.1 Latar Belakang

Deoxyribonucleic Acid (DNA) adalah molekul yang membawa informasi genetik dan berperan penting dalam struktur, fungsi, dan regulasi sel-sel yang membentuk organisme hidup (Wilson & Hunt, 2015). Struktur DNA terdiri dari basa *nitrogen adenine* (A), *guanine* (G), *cytosine* (C), *thymine* (T) yang berpasangan dalam struktur heliks ganda (Watson & Crick, 1953). DNA *sequencing* adalah teknik fundamental dalam biologi molekuler yang memungkinkan penentuan urutan nukleotida yang tepat dalam molekul DNA. Hal ini melibatkan teknik-teknik seperti *Sanger Sequencing* dan *Next Generation Sequencing* (NGS), yang telah merevolusi bidang *genomic* dan berperan penting dalam berbagai terobosan ilmiah. *Sanger Sequencing*, juga dikenal sebagai *chain termination sequencing*, adalah metode pertama yang dikembangkan untuk DNA *sequencing* dan merupakan metode yang digunakan untuk mengurutkan *genomee* manusia yang pertama (Sanger et al., 1977). Di sisi lain, teknologi NGS, seperti Illumina *sequencing*, telah memungkinkan *sequencing* dengan kecepatan tinggi, yang secara drastis meningkatkan kecepatan dan volume pembuatan data (Mardis, 2008).

Dengan teknologi yang digunakan saat ini, sekuens DNA yang panjang sulit untuk dilakukan *sequencing* secara akurat. Contohnya adalah DNA manusia yang panjangnya 3.2 milyar nukleotida dan tidak dapat dibaca dalam sekali proses. Karena permasalahan inilah sekuens DNA yang panjang harus dibagi menjadi fragmen-fragmen kecil dalam proses *sequencing*-nya, proses ini disebut sebagai *shotgun sequencing*. Dengan menggunakan pendekatan ini potongan DNA yang besar tadi dibagi menjadi beberapa potongan yang cukup kecil yang dapat diproses secara otomatis oleh mesin. *De novo genome assembly* adalah proses menyambungkan satu set sekuens pendek *genome*, juga disebut *reads*, untuk menghasilkan urutan DNA yang lebih panjang, yang disebut *contigs*. *De novo genome assembly* digunakan untuk *genome* baru, yaitu ketika tidak ada *genome*

referensi yang ada. Ada kebutuhan yang besar dalam penggunaan *de novo assembly* karena jumlah spesies yang tersekuens sepenuhnya sangat kecil dibandingkan dengan perkiraan jumlah organisme yang ada (Kunin et al., 2008).

Proses *assembly* diperumit oleh beberapa faktor. Pertama, adanya sekuens yang berulang dalam *genome* dapat menyebabkan ambiguitas dalam perakitan, karena sulit untuk menentukan di mana di dalam *genome* pengulangan ini terjadi (Alkan et al., 2011). Kedua, kesalahan *sequencing* dapat menyebabkan kesalahan dalam *assembly* (Nagarajan & Pop, 2013). Terakhir, ukuran dan kompleksitas *genome* yang sedang dirakit juga dapat menimbulkan kesulitan. Sebagai contoh, merakit *genome* manusia, dengan ukuran sekitar 3 miliar pasangan basa, adalah tugas yang jauh lebih kompleks daripada merakit *genome* bakteri kecil. Terlepas dari tantangan-tantangan ini, kemajuan dalam teknologi *sequencing* dan metode komputasi terus meningkatkan akurasi dan efisiensi DNA *sequence assembly*.

Secara tradisional DNA *sequence assembly* telah diimplementasikan menggunakan tiga pendekatan yang berbeda. Yang pertama adalah metode *Overlap-Layout-Consensus* (OLC). Contoh *tools* yang menggunakan metode ini antara lain PASQUAL (Liu et al., 2013) dan MAP (Lai et al., 2012). Pendekatan OLC didasarkan pada menemukan *overlap* antara *reads* dan membangun graf di mana setiap *vertex* mewakili *reads* dan setiap *edge* mewakili kesamaan *overlap* antara *reads*. Kemudian *traversing* dan transformasi graf diterapkan untuk mengekstraksi *contig* panjang. Pendekatan kedua untuk mengimplementasikan DNA *sequence assembly* didasarkan pada algoritma *greedy* seperti SSAKE (Warren et al., 2007) dan SHARCGS (Dohm et al., 2007). Dalam hal ini, *reads* secara bertahap ber-*overlap* menggunakan *seed reads* untuk menghasilkan *contig* yang lebih panjang. Pendekatan ketiga untuk menerapkan DNA *sequence assembly* menggunakan *de Bruijn graph* (DBG) sebagai struktur datanya seperti Velvet (Zerbino & Birney, 2008), SPAdes (Bankevich et al., 2012), dan ABySS (Simpson et al., 2009). DNA *sequence assembly* berbasis DBG (Pevzner et al., 2001), beroperasi dengan memotong *reads* menjadi subsekuens berturut-turut dengan panjang k, yang disebut k-mers. Setiap k-mer diwakili oleh *vertex* dan *edge* antara dua simpul mewakili k-1 *overlap* (akhiran-awalan).

Optimasi adalah konsep dasar dalam ilmu komputer dan matematika, yang melibatkan pencarian solusi terbaik dari sekumpulan solusi yang mungkin untuk masalah tertentu. Solusi "terbaik" biasanya didefinisikan dalam bentuk fungsi objektif, yang mengukur kualitas solusi. Masalah optimasi dapat ditemukan di berbagai bidang, termasuk teknik, ekonomi, analisis data, dan bioinformatika, antara lain (Kochenderfer & Wheeler, 2019). Ada berbagai jenis masalah optimasi, seperti linier dan non-linier, diskrit dan kontinu, serta deterministik dan stokastik, masing-masing dengan karakteristik dan metode solusinya yang unik (D. Boyd & Crawford, 2012). Namun, banyak masalah optimasi di dunia nyata yang kompleks dan menantang untuk dipecahkan, sering kali melibatkan ruang solusi yang besar, berbagai tujuan yang saling bertentangan, dan lingkungan yang dinamis (Deb, 2011).

Untuk mengatasi tantangan-tantangan ini, algoritma *metaheuristic* telah dikembangkan. *Metaheuristic* adalah kerangka kerja algoritma tingkat tinggi yang tidak bergantung pada masalah yang menyediakan seperangkat panduan atau strategi untuk mengembangkan algoritma optimasi heuristik (Blum & Roli, 2003). *Metaheuristic* sering digunakan untuk memecahkan masalah optimasi yang kompleks di mana metode tradisional tidak efektif atau efisien. *Metaheuristic* dapat diklasifikasikan ke dalam dua kategori utama: *metaheuristic* dengan *single-solution*, seperti *Simulated Annealing* (Kirkpatrick et al., 1983) dan *Tabu Search* (Glover, 1986a), dan *population-based metaheuristic*, seperti *Genetic Algorithm* (Holland, 1992), *Particle Swarm Optimization* (Kennedy & Eberhart, 1995), *Ant Colony Optimization* (Dorigo et al., 1996), dan *Differential Evolution* (Storn & Price, 1997). Algoritma-algoritma tersebut telah banyak digunakan di berbagai bidang karena kemampuannya dalam mengeksplorasi ruang solusi secara efektif dan efisien, serta fleksibilitasnya untuk diadaptasikan ke berbagai jenis masalah (Yang & Suash Deb, 2009). Namun, setiap algoritma memiliki kelebihan dan kekurangan, dan pemilihan algoritma sering kali bergantung pada karakteristik spesifik dari masalah yang dihadapi (Karaboga & Akay, 2009).

DNA *sequence assembly* adalah masalah menantang karena kompleksitas dan ukuran data yang digunakan. Sekuens DNA merupakan hasil proses DNA

sequencing yang terdiri dari banyak fragmen pendek yang perlu dirangkai menjadi sebuah urutan yang lebih panjang yang membuat ruang solusi sangat luas untuk dijelajahi. DNA *sequence assembly* memerlukan optimasi kombinatorial yang melibatkan pendekatan komputasional yang canggih dan memiliki ruang solusi yang besar. Proses optimasi kombinatorial ini dibutuhkan algoritma untuk mencari permutasi urutan dari banyaknya fragmen pendek DNA sehingga menghasilkan total *overlap* yang besar. Algoritma tradisional tidak efisien atau bahkan tidak praktis untuk data sebesar ini. Karena itu dibutuhkan optimasi atau algoritma *metaheuristic* yang menawarkan solusi yang lebih cepat dan efisien dari segi sumber daya komputasi.

Banyak algoritma heuristik dan *metaheuristic*, yang mengikuti pendekatan OLC dan bertujuan untuk menghitung jalur *Hamiltonian* pada graf *overlap* telah dikembangkan untuk menangani masalah DNA *sequence assembly*, selain metode-metode yang didasarkan pada teori graf. Dalam masalah DNA *sequence assembly*, *metaheuristic* bertujuan untuk memaksimalkan *overlap* antara fragmen DNA. Dengan melakukan ini, algoritma memastikan bahwa fragmen-fragmen tersebut disusun dan diurutkan dengan benar. *Overlap* yang besar sangat penting untuk mendapatkan perakitan berkualitas tinggi yang dapat digunakan untuk analisis biologis selanjutnya. *Problem Aware Local Search* (PALS) adalah algoritma heuristik yang efisien dan terkenal dalam domain ini (Alba & Luque, 2007). *Hybrid metaheuristic* yang menggabungkan PALS telah dikembangkan sebelumnya (Alba & Luque, 2008; Dorronsoro et al., 2008, 2010; Minetti et al., 2014). Ada beberapa penelitian yang telah dilakukan untuk merancang dan menguji teknik DNA *sequence assembly* berdasarkan *Genetic Algorithm* (GA) (Bucur, 2017; Hughes et al., 2016; Nebro et al., 2008). Potensi algoritma berbasis *swarm intelligence* juga telah diteliti, dengan contohnya termasuk *Ant Colony Optimization* (Meksangsouy & Chaiyaratana, 2003), *Particle Swarm Optimization* (PSO) (K.-W. Huang et al., 2015; Mallen-Fullerton & Fernandez-Anaya, 2013; Rajagopal & Maheswari Sankareswaran, 2015; Verma & Kumar, 2012), *Bee Algorithms* (Zemali & Boukra, 2018), *Cuckoo Search Algorithm* (Indumathy et al., 2015), dan *Penguin Search Optimization Algorithm* (Gheraibia et al., 2016).

Penelitian ini bertujuan untuk membandingkan performa algoritma-algoritma *population-based metaheuristic* dalam permasalahan DNA *sequence assembly*. Studi komparatif bertujuan untuk memberikan analisis yang komprehensif dan mendalam tentang algoritma yang digunakan dalam DNA *sequence assembly*. Dengan membandingkan algoritma *population-based metaheuristic*, kelebihan dan kekurangan dari masing-masing algoritma dapat dievaluasi sesuai dengan metrik yang digunakan yaitu waktu komputasi, total *overlap*, dan total *contig*. Selain itu, dikarenakan *de novo sequence assembly* merakit sekuens DNA yang tidak memiliki *genome* referensi, hasil dari perakitan DNA yang benar tidak diketahui, jika mayoritas algoritma menghasilkan *contig* yang sama atau serupa, ini bisa dianggap sebagai indikator bahwa *contig* tersebut adalah hasil yang paling mendekati benar.

Penggunaan algoritma berbasis populasi dalam DNA *sequence assembly* menawarkan beberapa keuntungan yang membuatnya sangat cocok untuk masalah optimasi yang kompleks ini. Pertama, algoritma ini mengevaluasi beberapa solusi secara bersamaan. Ini sangat bermanfaat untuk DNA *sequence assembly*, di mana ruang pencarian sangat besar dan kompleks. Kedua, algoritma berbasis populasi sangat baik dalam menyeimbangkan eksplorasi dan eksploitasi. Eksplorasi merujuk pada kemampuan algoritma untuk mencari area baru dari ruang solusi, sementara eksploitasi berfokus pada penyempurnaan solusi yang sudah ditemukan (Gandomi et al., 2013). Ini sangat penting dalam perakitan urutan DNA, di mana menemukan permutasi urutan baru (eksplorasi) maupun mengoptimalkan yang sudah ada (eksploitasi) penting untuk mencapai perakitan berkualitas tinggi. Terakhir, algoritma ini kurang rentan terjebak dalam *local optima* dibandingkan dengan algoritma *single-based metaheuristic* (Beheshti & Shamsuddin, 2013).

Algoritma yang digunakan adalah *Particle Swarm Optimization* (PSO) (Kennedy & Eberhart, 1995), *Honey Badger Algorithm* (HBA) (Hashim et al., 2022), *Lévy Flight Distribution* (LFD) (Houssein et al., 2020) dan *African Vultures Optimization Algorithm* (AVOA) (Abdollahzadeh et al., 2021). Penelitian ini mengusulkan model komputasi untuk menyelesaikan masalah DNA *sequence*

assembly dan membandingkan kinerja masing-masing algoritma berdasarkan waktu komputasi, jumlah contigs, dan nilai *overlap*.

1.2 Rumusan Masalah

Rumusan masalah dari latar belakang masalah yang telah dipaparkan pada sub bab sebelumnya adalah sebagai berikut:

1. Bagaimana implementasi algoritma HBA, LFD, AVOA, dan PSO dalam kasus DNA *Sequence Assembly* menggunakan bahasa pemrograman R?
2. Bagaimana perbandingan performa dari beberapa algoritma HBA, LFD, AVOA, dan PSO dalam kasus DNA *Sequence Assembly*?
3. Bagaimana tingkat efektivitas algoritma HBA, LFD, AVOA, dan PSO dalam kasus DNA *Sequence Assembly*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang dijelaskan sebelumnya, makan tujuan dari penelitian ini adalah sebagai berikut

1. Merancang dan mengimplementasikan model komputasi untuk DNA *sequence assembly* menggunakan algoritma HBA, LFD, AVOA, dan PSO.
2. Membandingkan algoritma HBA, LFD, AVOA, dan PSO untuk masalah DNA *sequence assembly*.
3. Membuktikan dan mengevaluasi tingkat efektivitas algoritma HBA, LFD, AVOA, dan PSO untuk masalah DNA *sequence assembly*.

1.4 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini adalah sebagai berikut:

1. Menyediakan alternatif lain untuk DNA *sequence assembly* dengan menggunakan algoritma *population-based metaheuristic* yang terbaru seperti HBA, LFD, AVOA, dan PSO.
2. Membuat model komputasi yang dapat mempermudah peneliti pada bidang bioinformatika untuk melakukan *DNA sequence assembly*.
3. Melakukan model program yang dapat dikembangkan pada penelitian selanjutnya.

1.5 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

1. Sekuens DNA yang dikenali adalah nukleobasa (*adenine* (A), *guanine* (G), *cytosine* (C), *thymine* (T)).
2. Sekuens DNA yang digunakan diperoleh dari NCBI.
3. Data yang digunakan menggunakan masukan dengan format FASTA.
4. Data yang digunakan merupakan data sekuens DNA yang berasal dari *single-end sequencing*.

1.6 Sistematika Penulisan

Pada dokumen skripsi ini terdapat sistematika yang digunakan sebagai pedoman penulisan. Sistematika tersebut dibagi menjadi lima bab, yaitu:

BAB I PENDAHULUAN

Bab ini berisi mengenai penjelasan latar belakang permasalahan penelitian ini. Pada Bab ini juga dijelaskan mengenai rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dalam penelitian, dan sistematika penulisan skripsi.

BAB II KAJIAN PUSTAKA

Bab ini berisi mengenai penjelasan teori-teori yang digunakan dalam menyelesaikan permasalahan penelitian ini. Adapun teori yang digunakan yaitu DNA dan DNA *sequence assembly*, algoritma *string matching*, bahasa pemrograman R, optimasi, dan *metaheuristics*.

BAB III METODE PENELITIAN

Bab ini berisi metode dan langkah-langkah yang digunakan dalam penelitian. Pada bab ini juga menjelaskan mengenai instrumen yang digunakan, tahapan pengumpulan data yang dilakukan, hingga langkah-langkah analisis data yang dijalankan.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi penjelasan mengenai proses dari setiap tahap penelitian. Proses tersebut meliputi pengumpulan data, perancangan model komputasi,

implementasi pengembangan perangkat lunak, rancangan skenario eksperimen, hasil eksperimen, dan pembahasan hasil penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi mengenai kesimpulan yang dibuat berdasarkan hasil penelitian. Kemudian terdapat beberapa saran yang dapat digunakan di dalam penelitian selanjutnya.