

Aplikasi Web Untuk Analisis Sekuens Fasta Berbasis Python

Pebi Mina Husania ¹, Tengku Syahvina Rival Dini ², Rani Chantika ³, Puji Sri Alhirani ⁴

Program Studi Ilmu Komputer, Universitas Islam Negeri Sumatera Utara ^{1,2,3,4}

Corresponding Author: febimina13@gmail.com¹, tengkusyahfina1004@gmail.com²,
chantikarani399@gmail.com³, srialhiranipuji@gmail.com⁴

Info Artikel

Submitted: 03 Desember 2025

Revised : 08 Januari 2026

Accepted: 20 Januari 2026

Published: 26 Februari 2026

Keywords: Bioinformatics,
FASTA, Web Application

Kata Kunci: Bioinformatika,
FASTA, Aplikasi Web.

Abstract

The rapid development of sequencing technology has generated large volumes of DNA, RNA, and protein sequence data, commonly stored in FASTA format. Although FASTA is simple and widely adopted, many existing bioinformatics tools are still desktop-based or rely on command line interfaces, which can be challenging for beginners. This study aims to develop a simple Python-based web application for basic FASTA sequence analysis. The application is built using the Flask framework and the BioPython library to analyze sequence length, nucleotide composition, and GC content through a user-friendly web interface. The system development follows the Waterfall model, including requirement analysis, system design, implementation, and testing. The results show that the application can process FASTA files accurately and efficiently without requiring additional software installation on the user side. This web-based application is expected to support bioinformatics learning activities and facilitate preliminary sequence analysis for students and researchers in molecular biology.

Abstrak

Perkembangan teknologi sekuensing telah menghasilkan data sekuens DNA, RNA, dan protein dalam jumlah besar yang umumnya disimpan dalam format FASTA. Meskipun format ini sederhana dan banyak digunakan, perangkat lunak bioinformatika yang tersedia masih didominasi oleh aplikasi berbasis desktop atau command line interface yang kurang ramah bagi pengguna pemula. Penelitian ini bertujuan untuk mengembangkan aplikasi web sederhana berbasis Python untuk analisis dasar file FASTA. Aplikasi dibangun menggunakan framework Flask dan pustaka BioPython untuk melakukan analisis panjang sekuens, komposisi basa nukleotida, serta persentase GC melalui antarmuka web yang mudah digunakan. Metode pengembangan sistem yang digunakan adalah model Waterfall, meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Hasil pengujian menunjukkan bahwa aplikasi mampu memproses file FASTA secara akurat dan cepat tanpa memerlukan instalasi perangkat lunak tambahan di sisi pengguna. Aplikasi ini diharapkan dapat menjadi media pembelajaran bioinformatika serta mendukung analisis awal sekuens biologis bagi mahasiswa dan peneliti.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Publisher: Lembaga Penerbit Penelitian Nusantara

PENDAHULUAN

Perkembangan ilmu biologi molekuler dan teknologi sekuensing dalam beberapa dekade terakhir telah menghasilkan data sekuens DNA, RNA, maupun protein dalam jumlah yang sangat besar. Data tersebut umumnya disimpan dalam format standar FASTA, yaitu format teks sederhana yang berisi urutan basa nukleotida atau asam amino. Keunggulan utama dari format FASTA terletak pada kesederhanaannya sehingga mudah dibaca oleh komputer maupun berbagai perangkat lunak analisis. Oleh karena itu, format FASTA telah menjadi standar penyimpanan data genetika digital yang dapat diakses secara luas dalam berbagai analisis bioinformatika, baik di tingkat nasional maupun internasional (Famuji et al., 2023).

Seiring dengan meningkatnya volume data biologis yang tersedia, kebutuhan akan perangkat yang mampu mengolah data FASTA secara cepat, ringan, dan mudah diakses menjadi semakin mendesak. Kondisi ini terutama dirasakan oleh mahasiswa dan peneliti pemula yang membutuhkan sarana analisis dasar sebelum melakukan analisis lanjutan. Namun demikian, sebagian besar perangkat lunak bioinformatika yang tersedia saat ini masih berbasis desktop atau command line interface (CLI). Model penggunaan seperti ini sering kali menimbulkan hambatan bagi pengguna yang tidak memiliki latar belakang teknis dalam instalasi perangkat lunak maupun penggunaan terminal. Selain itu, beberapa perangkat lunak bioinformatika bersifat berbayar sehingga aksesibilitasnya menjadi terbatas bagi kalangan akademisi tertentu.

Bioinformatika sebagai disiplin ilmu yang mengintegrasikan biologi, ilmu komputer, dan matematika hadir untuk menjawab tantangan analisis data biologis berskala besar. Akan tetapi, kompleksitas perangkat lunak yang ada sering kali tidak sebanding dengan kebutuhan pengguna pemula yang hanya memerlukan analisis sederhana terhadap data sekuens. Penelitian “Rancang Bangun Aplikasi Bioinformatika BLOKIBA” menegaskan bahwa Python dan pustaka BioPython efektif digunakan dalam membangun aplikasi bioinformatika, namun antarmuka yang tersedia masih relatif sulit diakses oleh pengguna non-teknis (Hadinata, 2021).

Dalam praktiknya, mahasiswa maupun peneliti di bidang biologi molekuler sering kali hanya memerlukan analisis dasar terhadap sekuens, seperti menghitung panjang sekuens, melihat komposisi basa nukleotida, atau menentukan persentase GC. Analisis dasar ini berperan penting sebagai langkah awal sebelum memasuki analisis lanjutan, seperti multiple sequence alignment, analisis filogenetik, maupun prediksi fungsi protein. Tanpa pemahaman karakteristik dasar sekuens, kualitas data yang digunakan pada tahap analisis lanjutan berpotensi tidak optimal. Beberapa penelitian nasional yang memanfaatkan data FASTA untuk analisis protein dan genom menekankan pentingnya tahap awal berupa identifikasi panjang dan

komposisi sekuens sebagai dasar validasi kualitas data. Artikel mengenai implementasi algoritma Needleman–Wunsch dalam analisis sekuens juga menegaskan bahwa identifikasi karakteristik dasar sekuens merupakan tahapan krusial sebelum data diproses lebih lanjut (Ngovangari et al., 2024).

Meskipun analisis dasar tersebut relatif sederhana, keterbatasan perangkat lunak yang praktis dan mudah digunakan sering kali menyebabkan proses analisis menjadi terhambat. Pengguna dihadapkan pada pilihan antara menggunakan perangkat lunak berbayar atau harus melakukan instalasi dan konfigurasi yang rumit. Kondisi ini menunjukkan adanya kesenjangan antara kebutuhan pengguna dan ketersediaan perangkat analisis yang sesuai, khususnya bagi pengguna pemula di lingkungan akademik.

Salah satu solusi yang dapat ditawarkan untuk menjawab permasalahan tersebut adalah pengembangan aplikasi web sederhana berbasis Python yang mampu melakukan analisis dasar file FASTA. Aplikasi ini memanfaatkan pustaka BioPython untuk pemrosesan sekuens biologis serta framework Flask sebagai kerangka kerja pengembangan web. Dengan pendekatan ini, pengguna dapat melakukan analisis langsung melalui browser tanpa perlu melakukan instalasi perangkat lunak yang kompleks. Aplikasi berbasis web juga bersifat lebih inklusif karena dapat diakses dari berbagai perangkat, baik komputer maupun gawai, sehingga mendukung fleksibilitas penggunaan di lingkungan pendidikan dan penelitian.

Aplikasi ini diharapkan dapat menjadi media pembelajaran bioinformatika yang mudah digunakan dalam kegiatan praktikum. Selain itu, aplikasi ini dapat membantu pengguna melakukan analisis cepat terhadap sekuens DNA, RNA, maupun protein sebelum memasuki tahapan analisis lanjutan. Aplikasi ini juga berpotensi mendukung riset genetika, analisis sekuens virus atau bakteri, serta studi mutasi dasar dalam konteks pengembangan obat atau vaksin. Dengan menyediakan alternatif aplikasi bioinformatika yang gratis, ringan, dan dapat diakses dari berbagai perangkat, pengembangan aplikasi ini diharapkan mampu menjangkau lebih banyak kalangan akademisi.

Berbagai penelitian terdahulu telah menunjukkan potensi pengembangan aplikasi bioinformatika berbasis web. Hadinata (2021) mengembangkan aplikasi BLOKIBA berbasis web yang mengintegrasikan BioPython untuk pemrosesan sekuens DNA dan RNA. Tian et al. (2022) mengembangkan ASAP 2 sebagai pipeline dan web server untuk analisis data sekuens amplicon secara otomatis. Studi nasional terbaru juga menegaskan bahwa ketersediaan aplikasi bioinformatika yang user-friendly berperan penting dalam meningkatkan efektivitas pembelajaran dan penelitian di perguruan tinggi (Sari et al., 2024).

Berdasarkan uraian tersebut, pengembangan aplikasi web sederhana berbasis Python untuk analisis dasar file FASTA menjadi relevan dan penting. Aplikasi ini tidak hanya menjawab kebutuhan praktis pengguna pemula, tetapi juga memberikan kontribusi dalam pengembangan perangkat lunak bioinformatika yang mudah diakses, efisien, dan mendukung kegiatan pendidikan serta penelitian di bidang biologi molekuler.

TINJAUAN PUSTAKA

A. Bioinformatika

Bioinformatika merupakan penggabungan beberapa disiplin ilmu, seperti matematika, statistika, komputer, biokimia, genetika, dan biologi molekuler. Istilah bioinformatika dan komputasi biologi sering digunakan bergantian, tetapi dua bidang ini memiliki fokus berbeda. Bioinformatika lebih berhubungan dengan pengembangan software, koleksi dan penyimpanan informasi biologi (database), dan metode visualisasi, sedangkan komputasi biologi lebih condong kepada pengembangan algoritma, model matematika dan statistika dalam menganalisa data-data biologi melalui bantuan komputer (Edi, 2020).

Bioinformatika dapat dijabarkan sebagai sistem dari teknologi komputer untuk mengelola informasi biologi. Bioinformatika mempunyai tujuan utama untuk menunjang proses biologi. Selain itu bioinformatika juga dapat digunakan untuk menyelesaikan masalah yang berkaitan dengan manusia (Famuji et al., 2023).

B. FASTA

FASTA dan FASTQ adalah format dasar dan umum untuk menyimpan urutan nukleotida dan protein. Manipulasi umum berkas FASTA/Q meliputi konversi, pencarian, penyaringan, deduplikasi, pemisahan, pengacakan, dan pengambilan sampel. Alat yang ada hanya mengimplementasikan sebagian dari manipulasi ini, dan tidak terlalu efisien, dan beberapa hanya tersedia untuk sistem operasi tertentu (Shen et al., 2016).

Nama FASTA berasal dari paket perangkat lunak yang ditulis pada pertengahan tahun 1980-an yang mencari dengan cepat melalui kumpulan besar atau sekuens – perangkat lunak tersebut disebut FASTA, tetapi juga disebut FAST-N(nukleotida) dan FAST-P (protein). Format FASTA minimal harus memiliki header (selalu diawali dengan tanda ">") di baris pertama berkas, dan urutannya dimulai di baris kedua. Header tersebut memuat beberapa informasi minimal tentang urutan tersebut. Contoh *file* FASTA, dapat direpresentasikan sebagai berikut:

>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCCTCTTTTCTTATCATTTGACATTTAACTCTGGGGCAGGTCCTCGCGTAGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACGCCCCGGGCTCCGGCCCCGGCCGGCTCGGGGCCCCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGCTCTTCTGCAAAAGTAGCAAAATGTTCCACTCCTAAGAGTGGACTTCCAGTCCGGCCCT
CAGCTCCGACTAGCCGGCAGCACTGCTGCTGCTGCTGCTAAGCAAGCACTCCGAGCCGCAAAATCGA

Sumber: <https://compgenomr.github.io/book/fasta-and-fastq-formats.html>

Ini adalah contoh **file FASTA**, yaitu format standar yang digunakan untuk menyimpan **data sekuens DNA, RNA, atau protein** dalam bioinformatika. Baris pertama (**header**) menjelaskan identitas gen dan sumber data. Baris berikutnya adalah **urutan DNA** dari gen **PAX6** manusia. *File* ini bisa dianalisis lebih lanjut untuk penelitian genetika, deteksi mutasi, maupun pembelajaran bioinformatika. Setiap *file* FASTA selalu dimulai dengan tanda > yang menunjukkan awal dari header, diikuti oleh informasi seperti ID sekuens, nama organisme, dan deskripsi gen. Baris setelahnya berisi urutan nukleotida atau asam amino yang ditulis tanpa angka, sehingga memudahkan pemrosesan data oleh perangkat lunak bioinformatika.

Kelebihan FASTA adalah sederhana mudah dibaca manusia, didukung secara luas, kompatibel dengan banyak alat dan basis data bioinformatika. Sedangkan, Kekurangan FASTA adalah Informasi terbatas sehingga tidak menyimpan skor kualitas atau data penyelarasan dan ukuran berkas dapat menjadi besar untuk kumpulan data yang luas.

C. BioPython

BioPython adalah pustaka open-source berbasis Python yang dikembangkan secara komunitas untuk mendukung analisis data biologis. Pustaka ini menyediakan berbagai modul untuk membaca, menulis, dan memproses format *file* biologis seperti FASTA, FASTQ, GenBank, hingga PDB. Modul Seq dan SeqRecord memudahkan representasi serta manipulasi sekuens DNA, RNA, maupun protein, sedangkan Bio.SeqIO dan Bio.AlignIO mendukung konversi lintas format sekuens maupun alignment. Yang pada dasarnya, tujuan dari Biopython adalah untuk mempermudah dalam menggunakan Bahasa pemrograman Python untuk penerapannya dalam bioinformatika dengan membuat modul dan kelas berkualitas tinggi yang dapat dimanfaatkan (Ngovangari et al., 2024).

BioPython memiliki sejumlah kelebihan yang menjadikannya populer dalam analisis bioinformatika. Pustaka ini mudah digunakan karena berbasis Python dengan sintaks sederhana sehingga dapat diakses oleh mahasiswa maupun peneliti pemula.

D. Aplikasi Web Bioinformatika

Perkembangan teknologi informasi telah mendorong munculnya berbagai aplikasi bioinformatika berbasis web yang dapat diakses melalui browser tanpa perlu instalasi software yang rumit. Aplikasi web ini menjadi solusi bagi peneliti dan mahasiswa yang ingin melakukan analisis data biologis secara cepat dan praktis. Salah satu contohnya adalah ASAP 2, yaitu pipeline sekaligus web server yang dirancang untuk memproses data sekuens marker (amplicon) secara otomatis. ASAP 2 dapat melakukan tahapan impor data, demultiplexing, trimming, penghilangan chimera, hingga klasifikasi taksonomi dan analisis keanekaragaman (alpha/beta) melalui antarmuka web yang ramah pengguna (Tian et al., 2022).

E. Relevansi Penelitian Terdahulu dengan Proyek Ini

Penelitian ASAP 2 relevan karena menunjukkan bahwa pipeline untuk analisis data sekuens biologis dapat diotomatisasi dan disajikan melalui web server sehingga pengguna yang tidak memiliki keahlian bioinformatika tetap dapat menggunakannya. Proyek ini akan memanfaatkan ide tersebut, namun dengan cakupan yang lebih sederhana, yaitu analisis *file* FASTA dasar berbasis Python.

Sementara itu, Genealyzer relevan dari aspek antarmuka pengguna dan kemudahan akses web untuk analisis data biologis meskipun fokusnya pada ekspresi gen. Proyek ini dapat mengambil inspirasi dari cara Genealyzer menyajikan visualisasi hasil analisis, sistem upload data, serta dukungan berbagai format input melalui browser. Dengan menggabungkan pendekatan web dan kemudahan penggunaan yang ditunjukkan oleh kedua aplikasi tersebut, proyek “Aplikasi Web untuk Analisis Sekuens FASTA Berbasis Python” diharapkan dapat menjadi versi yang lebih ringan dan fokus khusus untuk analisis dasar *file* FASTA, seperti panjang sekuens, komposisi basa, dan persentase GC. Aplikasi ini juga diharapkan bermanfaat sebagai media pembelajaran bioinformatika dan sebagai alat bantu penelitian awal di bidang genetika dan kesehatan.

METODE PENELITIAN

A. Metode Pengembangan Sistem

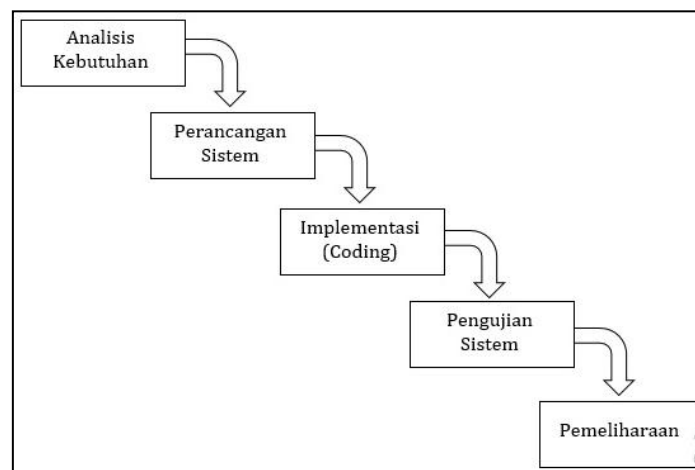
Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Waterfall. Metode ini merupakan salah satu model pengembangan perangkat lunak tertua dan paling umum digunakan karena bersifat sistematis, terstruktur, dan memiliki tahapan yang jelas. Setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya.

Keunggulan dari metode ini adalah dokumentasi yang lengkap di setiap proses pengembangan, sehingga memudahkan proses evaluasi dan perbaikan apabila diperlukan.

Model Waterfall dipilih karena sesuai dengan kebutuhan penelitian yang memiliki tujuan jelas dan ruang lingkup yang terdefinisi sejak awal. Pada pengembangan Aplikasi Web untuk Analisis Sekuens FASTA Berbasis Python, seluruh fungsi dan kebutuhan sistem telah ditentukan berdasarkan studi literatur dan analisis kebutuhan pengguna. Dengan demikian, pengembangan sistem dapat dilakukan secara bertahap mulai dari analisis kebutuhan hingga pengujian dan implementasi akhir.

Secara umum, metode Waterfall terdiri dari lima tahapan utama: analisis kebutuhan, perancangan sistem, implementasi atau coding, pengujian sistem, dan pemeliharaan. Setiap tahapan akan menghasilkan output berupa dokumen atau produk yang menjadi dasar untuk tahap selanjutnya.

Model Waterfall memiliki alur kerja yang berjalan secara linear dari tahap pertama hingga tahap akhir. Setiap tahapan harus diselesaikan sebelum berlanjut ke tahapan berikutnya. Berikut adalah diagram alur pengembangan sistem menggunakan metode Waterfall:



Gambar 1. Flowchart Model Waterfall

Gambar di atas menunjukkan bahwa proses pengembangan sistem dilakukan secara berurutan dan sistematis. Model ini dipilih karena sesuai dengan karakteristik proyek yang memiliki kebutuhan jelas sejak awal dan tidak mengalami banyak perubahan selama pengembangan.

1. Analisis Kebutuhan Sistem

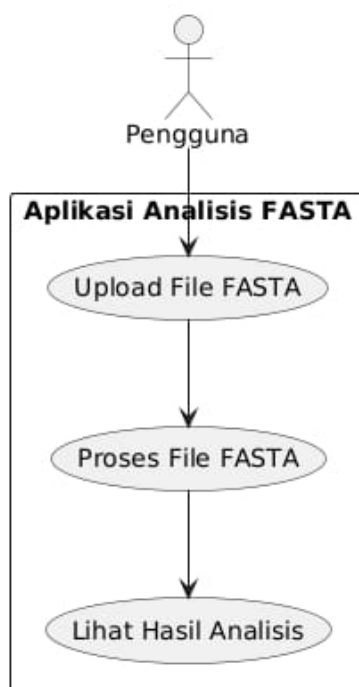
Analisis kebutuhan bertujuan untuk mendefinisikan fungsi-fungsi yang akan disediakan oleh sistem dan batasan-batasan yang harus dipenuhi. Analisis ini dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan non-fungsional.

2. Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan bagaimana sistem akan dibangun, mencakup pemodelan proses bisnis, interaksi pengguna dengan sistem, struktur data FASTA, serta arsitektur teknis aplikasi berbasis Flask dan BioPython. Perancangan sistem bertujuan memastikan bahwa aplikasi dikembangkan sesuai kebutuhan pengguna dan fungsi sistem dapat berjalan secara optimal.

3. Use Case Diagram

Use Case Diagram menggambarkan hubungan antara aktor (pengguna) dengan sistem. Pada aplikasi ini, aktor utama adalah user yang berinteraksi langsung dengan sistem melalui browser. Use case utama mencakup melakukan upload *file* FASTA, memproses file, dan melihat hasil analisis. Diagram ini bertujuan untuk menjelaskan fungsi utama yang dapat dilakukan pengguna dan bagaimana sistem merespon tindakan tersebut. Dengan adanya Use Case Diagram, alur interaksi sistem menjadi jelas dan dapat dijadikan acuan saat implementasi.



Gambar 2. Use Case Diagram

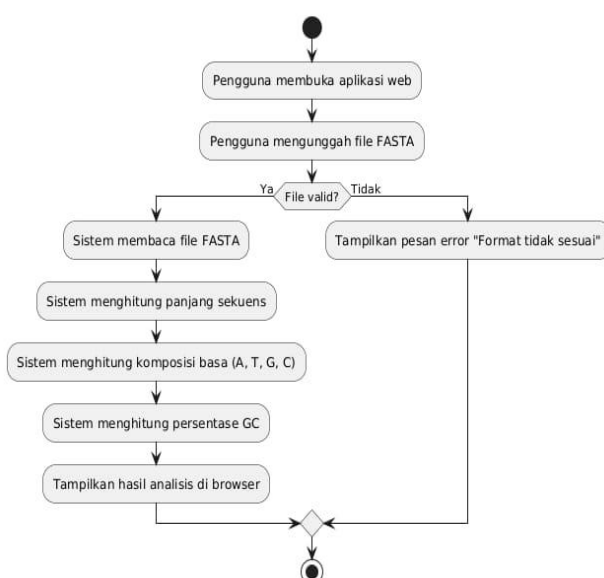
Pada diagram di atas, aktor utama adalah pengguna yang berinteraksi dengan sistem melalui browser web. Pengguna dapat melakukan tiga aktivitas utama, yaitu:

1. Mengunggah *file* FASTA, yaitu *file* berisi sekuens DNA atau RNA yang akan dianalisis.
2. Memproses *file* FASTA, di mana sistem akan membaca isi *file* menggunakan library BioPython dan melakukan analisis.
3. Melihat hasil analisis, yaitu tampilan hasil yang berisi informasi panjang sekuens, komposisi basa, serta persentase GC.

Diagram ini menggambarkan bahwa sistem hanya memiliki satu aktor eksternal (pengguna) dan seluruh proses dijalankan secara otomatis setelah *file* diunggah.

4. Activity Diagram

Activity Diagram menjelaskan alur aktivitas sistem dari awal hingga akhir. Alur dimulai ketika pengguna membuka aplikasi dan mengunggah *file* FASTA. Setelah itu, sistem akan melakukan validasi format file. Jika *file* sesuai, sistem akan memproses data dan menampilkan hasil analisis. Jika *file* tidak valid, sistem akan memberikan pesan kesalahan. Diagram ini menunjukkan proses berjalan secara berurutan dan adanya alur keputusan (decision flow). Activity Diagram berfungsi untuk memastikan bahwa setiap tahapan proses berjalan secara logis dan sistematis.



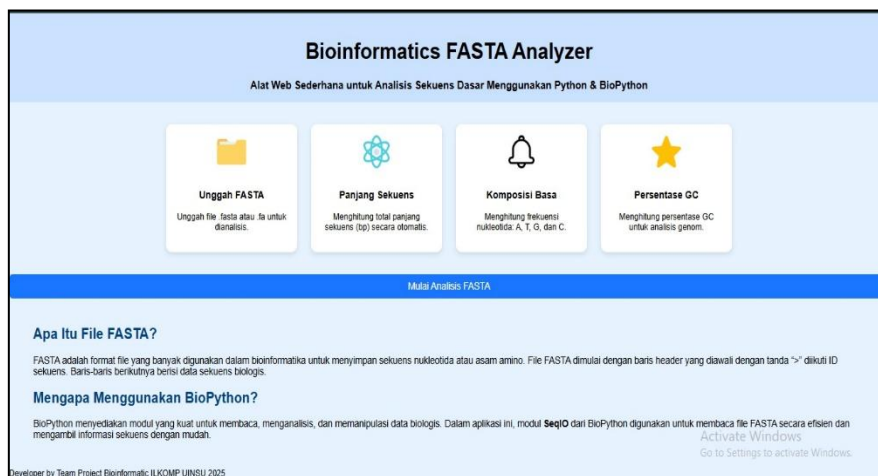
Gambar 3. Activity Diagram

Pada diagram di atas, proses dimulai ketika pengguna membuka aplikasi web dan mengunggah *file* FASTA untuk dianalisis. Setelah *file* diunggah, sistem akan melakukan validasi format file. Jika *file* tidak sesuai format, sistem menampilkan pesan kesalahan dan meminta pengguna mengunggah ulang file. Namun, jika *file* valid, sistem memproses isi *file* menggunakan library BioPython. Proses analisis mencakup penghitungan panjang sekuens, komposisi basa (A, T, G, C), serta persentase GC. Setelah perhitungan selesai, hasil analisis ditampilkan secara langsung di browser. Diagram ini menunjukkan bahwa aplikasi berjalan secara berurutan, dengan satu percabangan keputusan di bagian validasi file.

HASIL DAN PEMBAHASAN

A. Tampilan dan Fitur Aplikasi

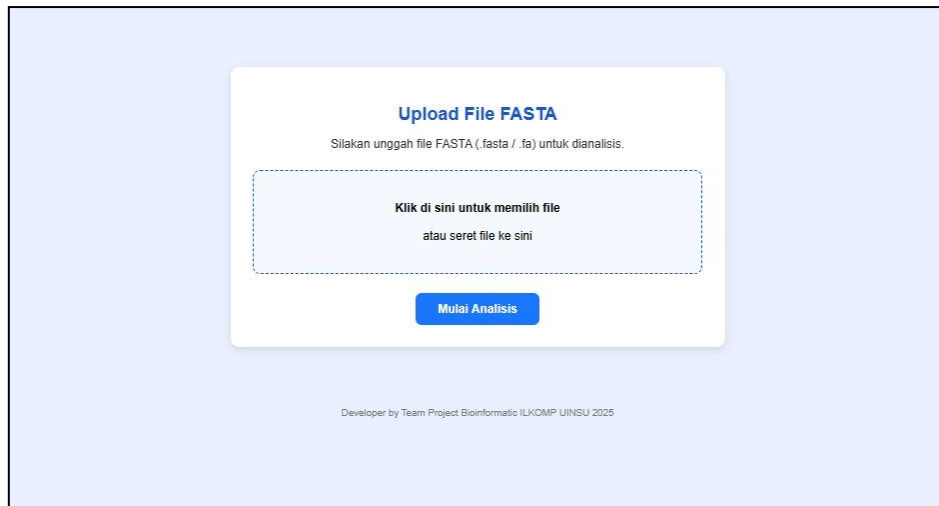
A. Tampilan Halaman *Dashboard*



Gambar 4 Halaman *Dashboard*

Tampilan *dashboard* merupakan halaman awal yang muncul ketika pengguna membuka aplikasi *FASTA Analyzer*. Halaman ini dirancang sebagai pusat navigasi utama yang memberikan gambaran umum mengenai fungsi dan fitur dari aplikasi. Desain *dashboard* menggunakan tema warna biru dan putih untuk menampilkan kesan sederhana, bersih, dan profesional sehingga mudah dipahami oleh pengguna, khususnya mahasiswa atau peneliti pemula yang baru mengenal analisis sekuens. Pada bagian atas *dashboard* terdapat judul aplikasi yang menegaskan bahwa sistem ini berfungsi sebagai alat *web* sederhana untuk analisis sekuens dasar berbasis Python dan BioPython. Di bawahnya, pengguna akan melihat empat kartu fitur yang masing-masing menjelaskan kemampuan inti aplikasi, yaitu mengunggah *file* FASTA, melihat panjang sekuens, menghitung komposisi basa A, T, G, C, serta mengetahui persentase GC. Setiap kartu dilengkapi ikon dan deskripsi singkat untuk membantu pengguna memahami fungsi aplikasi secara visual. Tombol “Mulai Analisis FASTA” ditempatkan di tengah halaman sebagai akses utama menuju proses pengunggahan *file*.

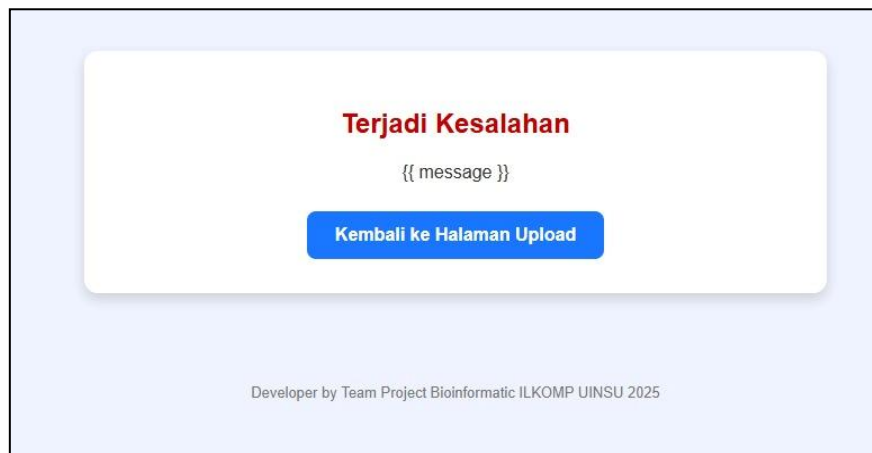
B. Tampilan Halaman Upload *File* FASTA



Gambar 5 Halaman *Upload File* FASTA

Halaman *upload file* merupakan halaman yang digunakan pengguna untuk mengunggah *file* FASTA yang akan dianalisis oleh aplikasi. Tampilan halaman ini dirancang dengan sederhana namun tetap informatif agar mudah dipahami oleh pengguna yang mungkin baru pertama kali bekerja dengan data sekuens. Pada halaman ini, terdapat sebuah kotak *upload* yang berfungsi sebagai area interaktif tempat pengguna dapat memilih *file* dengan mengklik bagian tersebut atau menyeret *file* FASTA langsung ke dalam kotak. Desainnya dibuat cukup besar dan berada di bagian tengah halaman agar mudah terlihat dan memberikan fokus utama pada fungsi *upload*. Selain itu, terdapat keterangan yang menjelaskan bahwa aplikasi hanya menerima *file* dengan format *.fasta* atau *.fa*, sehingga mengurangi kesalahan pengguna dalam memilih *file* yang tidak sesuai. Setelah *file* dipilih, pengguna dapat menekan tombol “Mulai Analisis” untuk memproses *file* tersebut. Halaman ini dirancang untuk memberikan pengalaman unggah yang sederhana, responsif, dan intuitif, sehingga proses analisis dapat dilakukan dengan cepat tanpa membingungkan pengguna. Secara keseluruhan, halaman *upload file* berfungsi sebagai pintu masuk utama untuk memulai proses analisis sekuens dan memastikan bahwa *file* yang diterima sistem merupakan data biologis yang *valid* untuk diproses.

C. Tampilan Halaman Error



Gambar 6 Halaman *Error*

Halaman error merupakan tampilan yang disediakan untuk memberi tahu pengguna ketika terjadi kesalahan selama proses pengunggahan atau pembacaan *file* FASTA. Halaman ini muncul secara otomatis apabila pengguna mengunggah *file* dengan format yang tidak sesuai, *file* kosong, nama *file* tidak valid, atau ketika terjadi *error* teknis saat aplikasi membaca struktur *file* FASTA menggunakan BioPython. Tampilan halaman *error* dirancang sangat sederhana dengan pesan yang jelas dan informatif, sehingga pengguna dapat langsung memahami penyebab masalah tanpa kebingungan. Pesan *error* ditampilkan di bagian tengah halaman dalam bentuk teks yang mudah dibaca, disertai warna penanda seperti merah atau elemen visual lain yang menunjukkan bahwa terjadi kegagalan proses. Selain itu, halaman ini juga menyediakan tombol untuk kembali ke halaman upload sehingga pengguna dapat langsung memperbaiki kesalahan dan mencoba mengunggah ulang *file* yang benar.

D. Tampilan Halaman Hasil Analisis

Hasil Analisis File FASTA

Nama File:
{{ filename }}

Nama Spesies:
{{ species }}

Panjang Sekuens:
{{ length }} basa

Komposisi Basa:
A: {{ count_A }}
T: {{ count_T }}
G: {{ count_G }}
C: {{ count_C }}

Persentase GC:
{{ gc_percent }} %

Preview Sekuens:
{{ preview }}

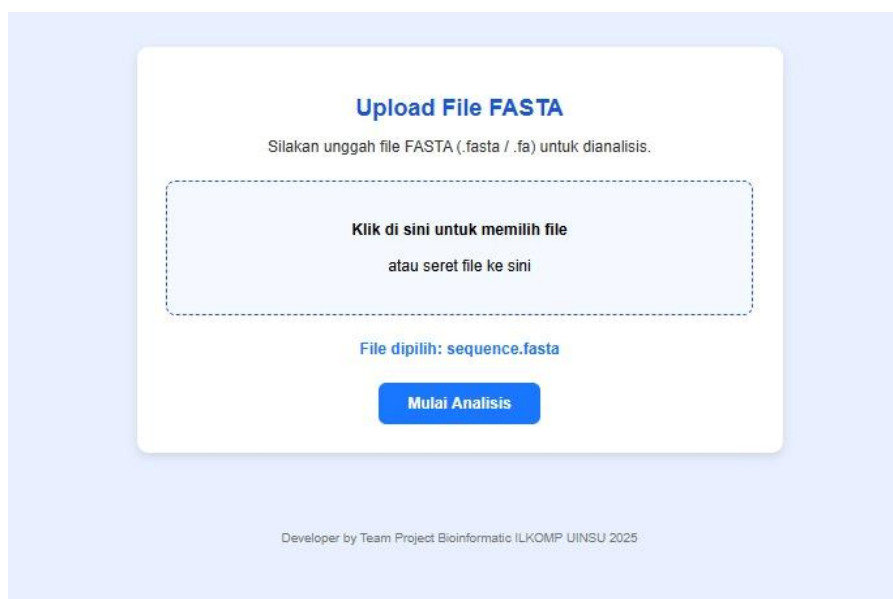
Analisis File Lain

Developer by Team Project Bioinformatik: IKOMF UINSU 2025

Gambar 7 Halaman Analisis *File*

Tampilan halaman hasil analisis merupakan halaman yang menampilkan seluruh output dari proses komputasi yang dilakukan terhadap file sekuens yang telah diunggah oleh pengguna. Pada halaman ini, sistem menampilkan informasi dasar terkait file, seperti nama file dan identitas spesies yang berhasil terdeteksi dari sekuens yang dianalisis. Informasi tersebut ditampilkan secara jelas di bagian atas untuk memudahkan pengguna memastikan bahwa file yang diproses adalah file yang benar. Selain itu, halaman ini juga menyajikan hasil analisis alignment atau pemrosesan bioinformatika lain dalam bentuk visualisasi tabel atau grafik, sehingga pengguna dapat melihat pola, skor, atau karakteristik sekuens yang dianalisis dengan lebih informatif. Keseluruhan tampilan dirancang sederhana dan ringkas agar hasil analisis dapat dibaca dengan mudah tanpa gangguan elemen yang tidak diperlukan. Halaman ini menjadi bagian inti dari aplikasi karena menyajikan output utama yang menjadi tujuan penggunaan sistem.

B. Hasil Pengujian Aplikasi



Gambar 8. Halaman Upload File Setelah Pemilihan File FASTA

Gambar ini menunjukkan tampilan halaman upload setelah pengguna berhasil memilih file FASTA yang akan dianalisis. Pada tahap ini, sistem menampilkan informasi nama file yang telah diunggah, yaitu *sequence.fasta*, sebagai indikator bahwa file telah terbaca dan tersimpan sementara di sistem. Penampilan nama file ini berfungsi sebagai bentuk umpan balik langsung kepada pengguna agar dapat memastikan bahwa file yang dipilih sudah benar sebelum proses analisis dilanjutkan.

Selain itu, halaman ini juga menunjukkan bahwa proses validasi dasar telah berhasil dijalankan, di mana sistem hanya menerima file dengan format *.fasta* atau *.fa* sesuai standar data sekuens biologis. Apabila file tidak sesuai format, sistem akan menolak dan meminta pengguna mengganti file terlebih dahulu. Dengan demikian, halaman ini menjadi bukti bahwa fungsi input dan pengecekan format bekerja dengan baik.

Tahap ini merupakan bagian penting dari alur kerja aplikasi karena memastikan integritas data sebelum masuk ke proses analisis. Setelah file berhasil diunggah, pengguna dapat menekan tombol “Mulai Analisis” untuk menjalankan pemrosesan yang meliputi pembacaan sekuens, perhitungan panjang nukleotida, distribusi basa (A, T, G, C), serta persen GC. Keberhasilan tahap ini menunjukkan bahwa antarmuka dan backend telah terhubung dengan baik, sehingga aplikasi siap menjalankan fungsi utamanya dalam analisis data biologis berbasis file FASTA.

[illegible]

Gambar 9. Tampilan Output Analisis File FASTA

Gambar ini merupakan tampilan output hasil analisis setelah file FASTA berhasil diproses oleh aplikasi. Pada halaman ini, sistem menampilkan informasi karakteristik sekuens secara detail, meliputi nama file yang dianalisis, panjang total sekuens, jumlah masing-masing basa nukleotida (A, T, G, dan C), serta persentase GC. Informasi tersebut ditampilkan dalam format teks yang terstruktur dan mudah dibaca, sehingga pengguna dapat langsung memahami hasil analisis tanpa perlu melakukan perhitungan manual.

Tampilan ini menunjukkan bahwa aplikasi tidak hanya berhasil membaca file FASTA, tetapi juga melakukan parsing sekuens secara benar menggunakan metode perhitungan yang sesuai dengan standar bioinformatika. Nilai panjang sekuens dan persentase GC yang dihasilkan juga dapat digunakan sebagai parameter awal dalam analisis komputasional lebih lanjut, seperti identifikasi gen, penentuan stabilitas DNA, ataupun pemetaan karakteristik organisme tertentu.

Selain itu, tampilan hasil analisis ini menjadi bukti bahwa fungsi *frontend* dan *backend* berjalan dengan baik secara terintegrasi. Proses analisis dilakukan pada sisi server, kemudian hasilnya dikirim kembali dan ditampilkan melalui antarmuka web secara responsif. Dengan

adanya tampilan ini, pengguna dapat melakukan evaluasi cepat terhadap kualitas dan struktur sekuens yang dianalisis tanpa memerlukan perangkat lunak bioinformatika tambahan. Hal ini membuktikan bahwa aplikasi mampu menyediakan analisis dasar FASTA secara otomatis, efisien, dan ramah pengguna.

KESIMPULAN

A. Kesimpulan

Pengembangan “Aplikasi Web untuk Analisis Sekuens FASTA Berbasis Python” bertujuan untuk menyediakan solusi komputasi bioinformatika yang sederhana, ringan, dan mudah digunakan oleh mahasiswa maupun peneliti pemula. Melalui pemanfaatan framework Flask dan pustaka BioPython, aplikasi mampu melakukan analisis dasar terhadap file FASTA meliputi pembacaan sekuens, perhitungan panjang nukleotida, komposisi basa (A, T, G, C), serta persentase GC secara otomatis melalui antarmuka web. Berdasarkan hasil implementasi dan pengujian sistem, aplikasi telah berjalan dengan baik dan mampu memproses data secara akurat dalam waktu singkat tanpa memerlukan instalasi perangkat lunak tambahan di sisi pengguna. Hal ini menunjukkan bahwa pengembangan aplikasi web bioinformatika dapat menjadi alternatif solusi pembelajaran maupun riset dasar di bidang biologi molekuler dan genetika.

Meskipun demikian, aplikasi yang dikembangkan masih memiliki beberapa keterbatasan, khususnya dalam dukungan multi-sekuens, analisis lanjutan, dan fitur penyimpanan hasil secara permanen. Oleh karena itu, sistem saat ini lebih optimal digunakan sebagai media pembelajaran dan eksplorasi awal dalam memahami karakteristik sekuens biologis sebelum melangkah ke analisis tingkat lanjut seperti alignment, anotasi gen, atau pemodelan struktur protein.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa teknologi pengembangan aplikasi web berbasis Python, khususnya dengan integrasi BioPython, dapat memberikan kontribusi nyata dalam mempermudah akses dan pemrosesan data sekuens biologis. Aplikasi yang dihasilkan bukan hanya memberi pengalaman praktis dalam penerapan konsep bioinformatika, tetapi juga membuka kesempatan untuk pengembangan sistem yang lebih luas dan canggih di masa mendatang sesuai kebutuhan penelitian dan perkembangan kurikulum pendidikan.

B. Saran

Berdasarkan hasil implementasi dan keterbatasan yang ada, beberapa saran pengembangan di masa mendatang antara lain:

1. Menambahkan dukungan analisis untuk file FASTA dengan lebih dari satu sekuens (multi-entry) sehingga pengguna dapat melakukan perbandingan atau pemrosesan batch secara sekaligus.
2. Mengembangkan fitur analisis lanjutan seperti identifikasi ORF, perhitungan motif nukleotida, prediksi gen, atau integrasi dengan algoritma alignment seperti BLAST atau Clustal.
3. Menambahkan fitur ekspor hasil analisis ke format PDF, CSV, atau JSON agar pengguna dapat menyimpan dan membagikan laporan hasil secara praktis.
4. Meningkatkan antarmuka pengguna dengan dokumentasi penggunaan, indikator progres pemrosesan, serta visualisasi grafik agar output lebih informatif dan mudah dipahami.
5. Mengembangkan sistem menjadi platform multi-user yang dilengkapi autentikasi dan penyimpanan berbasis database jika aplikasi digunakan dalam lingkungan akademik atau penelitian berskala lebih besar.

Dengan pengembangan berkelanjutan, aplikasi diharapkan tidak hanya menjadi media pembelajaran, tetapi juga mampu berkembang menjadi perangkat bioinformatika yang bermanfaat dalam penelitian komputasi biologis pada tingkat perguruan tinggi maupun institusi penelitian.

REFERENSI

- Budiono, A., Adam, C., & Arini, D. (2022). *Biopython: Comparing the DNA Polymerase I (polA) Gene of Thermophilic, Mesophilic, and Psychrophilic Bacteria*. BiosciED Journal, 3(1), 1–8. Universitas Palangka Raya.
- Edi, E. (2020). Bioinformatika: Komputer + Statistika + Matematika + Biologi. *Jurnal TIMES*, 6(1), 23–25.
- Famuji, T. S., Herman, & Sunardi. (2023). Proses Implementasi Bioinformatika Pada Digitalisasi Data Genetika Manusia. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 14(1), 1–12. <https://doi.org/10.24176/simet.v14i1.9064>
- Genbinesia. (2023). *Bioinformatika Dasar dengan Python 3*. Genbinesia Press. <https://genbinesia.or.id/wp-content/uploads/2023/05/Bioinformatika-Dasar-dengan-Python-3.pdf>

- Hadinata, E. (2021). Rancang Bangun Aplikasi Bioinformatika Biokiba Menggunakan Biopython Dan Framework Laminas. *Jurnal Abdi Ilmu*, 14(1), 117–134. <https://journal.pancabudi.ac.id/index.php/abdiilmu/article/view/3946>
- Lietz, K., Saremi, B., & Wiese, L. (2023). Genealyzer: web application for the analysis and comparison of gene expression data. *BMC Bioinformatics*, 24(1), 1–17. <https://doi.org/10.1186/s12859-023-05266-4>
- Montecuolo, F., Schmid, G., & Tagliaferri, R. (2017). E2FM: An encrypted and compressed full-text index for collections of genomic sequences. *Bioinformatics*, 33(18), 2808–2817. <https://doi.org/10.1093/bioinformatics/btx313>
- Ngovangari, R. H., Rorimpandey, G. C., Kembuan, O., & Sumual, H. M. (2024). Implementasi Algoritma Needleman-Wunsch dalam Pengujian Tingkat Kemiripan DNA Babi Sulawesi Utara. *JOINTER: Journal of Informatics Engineering*, 5(01), 20–26. <https://doi.org/10.53682/jointer.v5i01.32>
- Phelan, J., Deelder, W., Ward, D., Campino, S., Hibberd, M. L., & Clark, T. G. (2022). COVID-profiler: a webserver for the analysis of SARS-CoV-2 sequencing data. *BMC Bioinformatics*, 23(1), 1–5. <https://doi.org/10.1186/s12859-022-04632-y>
- Razsanjani, M. L., Nasution, H., & Novriando, H. (2024). Analisis Performa Web server Python dan Go Pada Protokol Http (Hypertext Transfer Protocol). *Juara, Jurnal Aplikasi Dan Riset Informatika*, 02(2), 95–103. <https://doi.org/10.26418/juara.v2i2.82059>
- Rifani Khairani Pohan, Juan Dini, Mutiarani Mutiarani, M. Iqbal, & Fatur Rahman. (2024). Pencegahan Penyakit Kanker Payudara dengan Bioinformatika: Membangun Masa Depan Kesehatan. *Jurnal Sistem Informasi Dan Ilmu Komputer*, 3(1), 129–134. <https://doi.org/10.59581/jusiik-widyakarya.v3i1.4489>
- Sardi, A. (2022). Bioinformatics: Challenges in Integrating Biological Information. *Jurnal Biologi Tropis*, 22(4), 1297–1301. <https://doi.org/10.29303/jbt.v22i4.4346>
- Sari, I. J., R. Ahmad Zaky El, Suparman, S., & Lutfi, A. (2024). Pemanfaatan Bioinformatics Tools dalam Pembelajaran Genetika Bagi Dosen Pendidikan Biologi di Salah Satu Universitas Di Ternate. *Jurnal Pengabdian Dinamika*, 11(2), 45–61.
- Shen, W., Le, S., Li, Y., & Hu, F. (2016). SeqKit: A cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PLoS ONE*, 11(10), 1–10. <https://doi.org/10.1371/journal.pone.0163962>
- Shendre, A., Mehta, N. K., Rathore, A. S., Kumar, N., Patiyal, S., & Raghava, G. P. S. (2025). MAP Format for Representing Chemical Modifications, Annotations, and Mutations in Protein Sequences: An Extension of the FASTA Format. *ArXiv Preprint*

ArXiv:2505.03403, 302.

- Tian, R., & Imanian, B. (2022). ASAP 2: a pipeline and web server to analyze marker gene amplicon sequencing data automatically and consistently. *BMC Bioinformatics*, 23(1), 1–12. <https://doi.org/10.1186/s12859-021-04555-0>
- Zhou, D., Tran, Y., Abou Elela, S., & Scott, M. S. (2022). SAPFIR: A webserver for the identification of alternative protein features. *BMC Bioinformatics*, 23(1), 1–13. <https://doi.org/10.1186/s12859-022-04804-w>.