BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi

Dalam penelitian virus Zika ini, metodologi penelitian yang digunakan antara lain sebagai berikut.

1. Studi Pustaka

   Dalam proses ini, dijabarkan teori-teori yang berhubungan dengan penelitian virus Zika, yaitu virus Zika, nukleotida, penjajaran sekuen, algoritma Needleman-Wunsch, algoritma Smith-Waterman, National Center for Biotechnology Information (NCBI), Basic Local Alignment Search Tool (BLAST), dan algoritma *k-means clustering*.

2. Ekstraksi Data

   Pada tahap ini, dilakukan pengambilan data virus Zika dengan cara melakukan pencarian dan pengunduhan data dari situs National Center for Biotechnology Information (NCBI). Data GenBank virus Zika akan menjadi sumber data utama yang akan diproses dan digunakan dalam penelitian ini.

3. Perancangan Sistem

   Pada tahap ini, dilakukan perancangan atas sistem yang ingin dibuat, yang terdiri atas *flowchart* dan desain antarmuka. Pembuatan *flowchart* bertujuan untuk mendefinisikan alur kerja sistem sebagai dasar implementasi yang akan dilakukan. Pembuatan desain antarmuka bertujuan untuk merancang sistematika halaman penelitian agar dapat lebih intuitif dan informatif untuk digunakan.

4. Pemrograman Sistem

   Pemrograman sistem dilakukan dengan Jupyter Notebook versi 4.04 yang menggunakan bahasa pemrograman Python. Pada tahap ini, data GenBank virus Zika dari National Center for Biotechnology Information (NCBI) akan disiapkan agar dapat dijajarkan. Setelah itu, dilakukan penjajaran sekuen baik secara global menggunakan algoritma Needleman-Wunsch maupun secara lokal menggunakan algoritma Smith-Waterman. Hasil penjajaran ini akan digunakan dalam tahap analisis dan evaluasi.

5. Analisis dan Evaluasi

   Pada tahap ini, data hasil penjajaran akan dianalisis untuk mendapatkan informasi. Hasil analisis data disajikan menggunakan tabel dan grafik. Setelah itu, data analisis akan divalidasi, yaitu dengan melakukan perbandingan antara hasil keluaran dari sistem dan hasil keluaran menggunakan Basic Local Alignment Search Tool (BLAST).

3.2 Perancangan Sistem

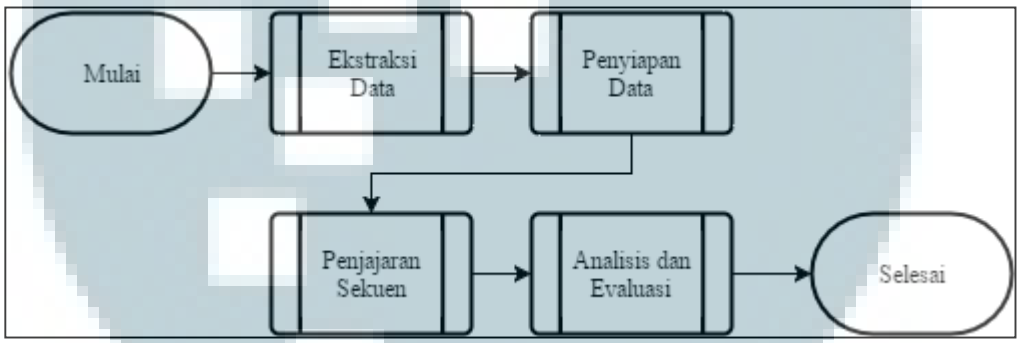
Perancangan yang dilakukan pada sistem ini terdiri atas dua bagian, yaitu pembuatan *flowchart* dan pembuatan desain antarmuka. *Flowchart* dibuat untuk mendefinisikan alur kerja sistem sebagai dasar implementasi yang akan dilakukan. Desain antarmuka dibuat untuk merancang sistematika halaman penelitian agar dapat lebih intuitif dan informatif untuk digunakan.

3.2.1 Flowchart

Flowchart dibuat untuk mendefinisikan alur kerja sistem sebagai dasar implementasi yang akan dilakukan. Berikut adalah *flowchart* yang dibuat dalam penelitian ini.

A. Flowchart Rangka Kerja Implementasi

Alur kerja implementasi secara umum dalam penelian ini dapat dilihat pada Gambar 3.1.

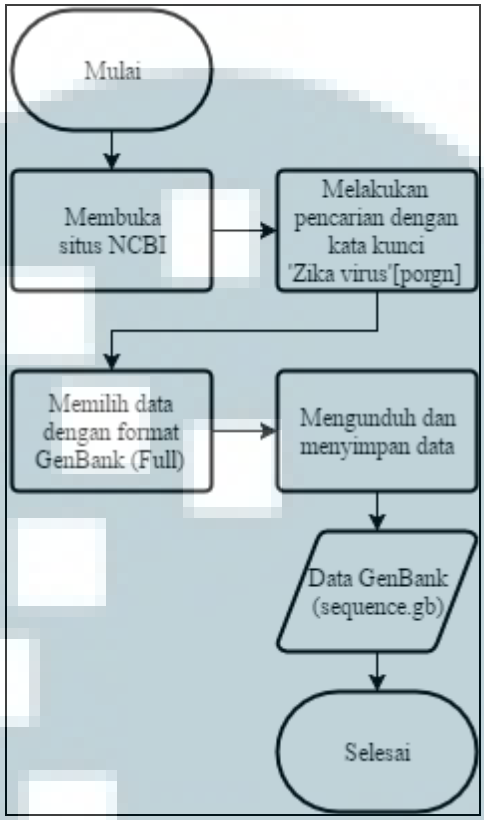


Gambar 3.1 *Flowchart* Rangka Kerja Implementasi

Tahapan kerja penelitian dimulai dengan ekstraksi data yang dilakukan dari situs National Center for Biotechnology Information (NCBI). Selanjutnya, data akan di-*parse* dan disiapkan untuk dijajarkan. Data yang telah disiapkan ini akan diproses oleh program untuk mendapatkan hasil berupa informasi hasil penjajaran. Informasi ini kemudian akan dianalisis untuk mendapatkan *insight* dari penelitian yang dilakukan.

B. Flowchart Detil Proses Ekstraksi Data

Alur kerja detil proses ekstraksi data dapat dilihat Gambar 3.2.

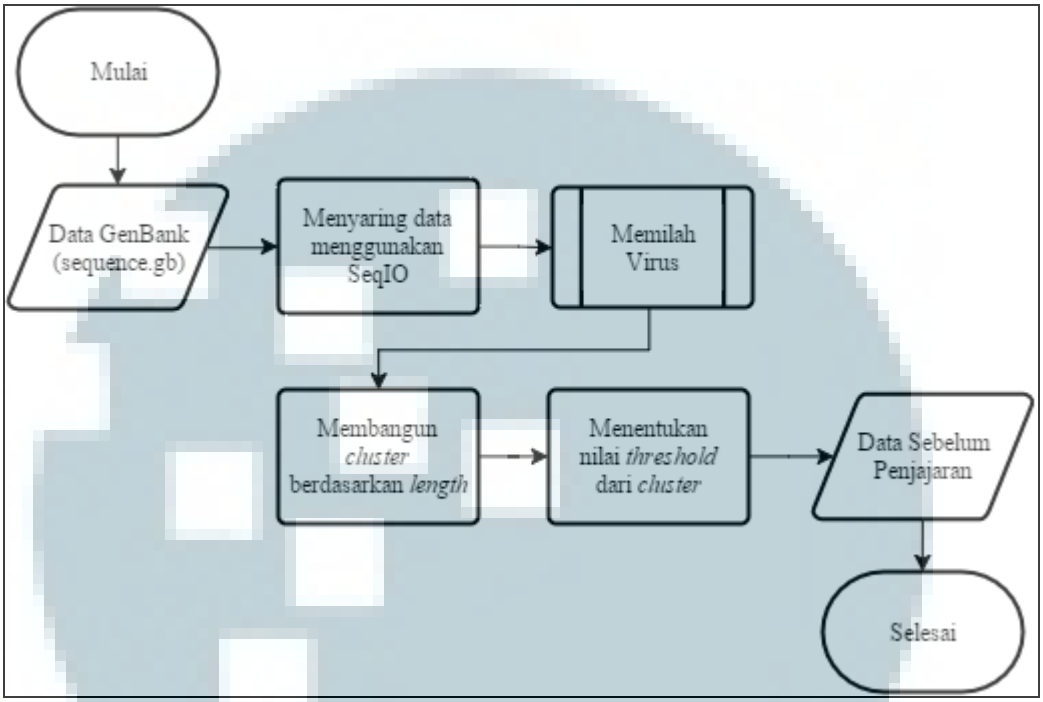


Gambar 3.2 *Flowchart* Detil Proses Ekstraksi Data

Pengambilan data dimulai dengan membuka situs NCBI dan masuk ke halaman pencarian nukleotida. Setelah itu, pencarian dilakukan pada mesin pencari dengan menggunakan kata kunci “Zika virus”[porgn] untuk mendapatkan seluruh data virus Zika yang menjadi organisme primer. Setelah itu, akan terdapat pilihan format data. Format data yang akan digunakan dalam penelitian ini adalah bentuk GenBank (Full). Setelah memilih format, data diunduh dan disimpan pada direktori penelitian. Hasil proses ini adalah data GenBank (Full) yang terdapat pada file ‘sequence.gb’.

C. Flowchart Detil Proses Penyiapan Data

Alur kerja detil proses penyiapan data dapat dilihat pada Gambar 3.3.

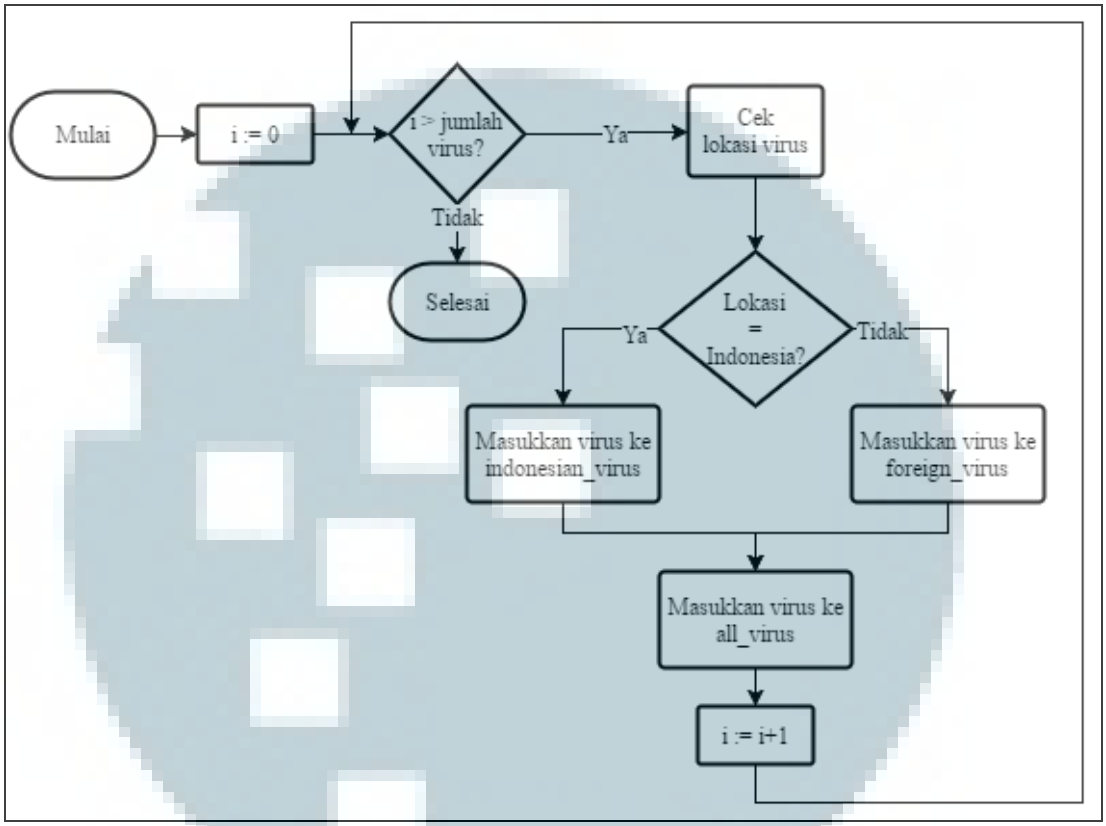


Gambar 3.3 *Flowchart* Detil Proses Penyiapan Data

Proses penyiapan data menerima input berupa data GenBank 'sequence.gb' dari proses ekstraksi data. Data GenBank ini dapat di-*parse* menggunakan *library* SeqIO. Kemudian, data virus akan dibagi menjadi kelompok virus Indonesia dan kelompok virus asing melalui proses Memilah Virus, agar dapat dibandingkan pada penjajaran sekuen. Setelah itu, data virus akan dibagi menjadi dua *cluster* berdasarkan panjang sekuen menggunakan *library* scikit-learn. Tujuan dari *clustering* ini adalah untuk mendapatkan nilai *threshold*, berdasarkan rerata dari dua *centroid cluster*, yang akan digunakan untuk menentukan jenis penjajaran. Hasil akhir dari tahap ini adalah data yang telah siap untuk dijajarkan.

D. Flowchart Detil Proses Memilah Virus

Alur kerja detil proses memilah virus dapat dilihat pada Gambar 3.4.

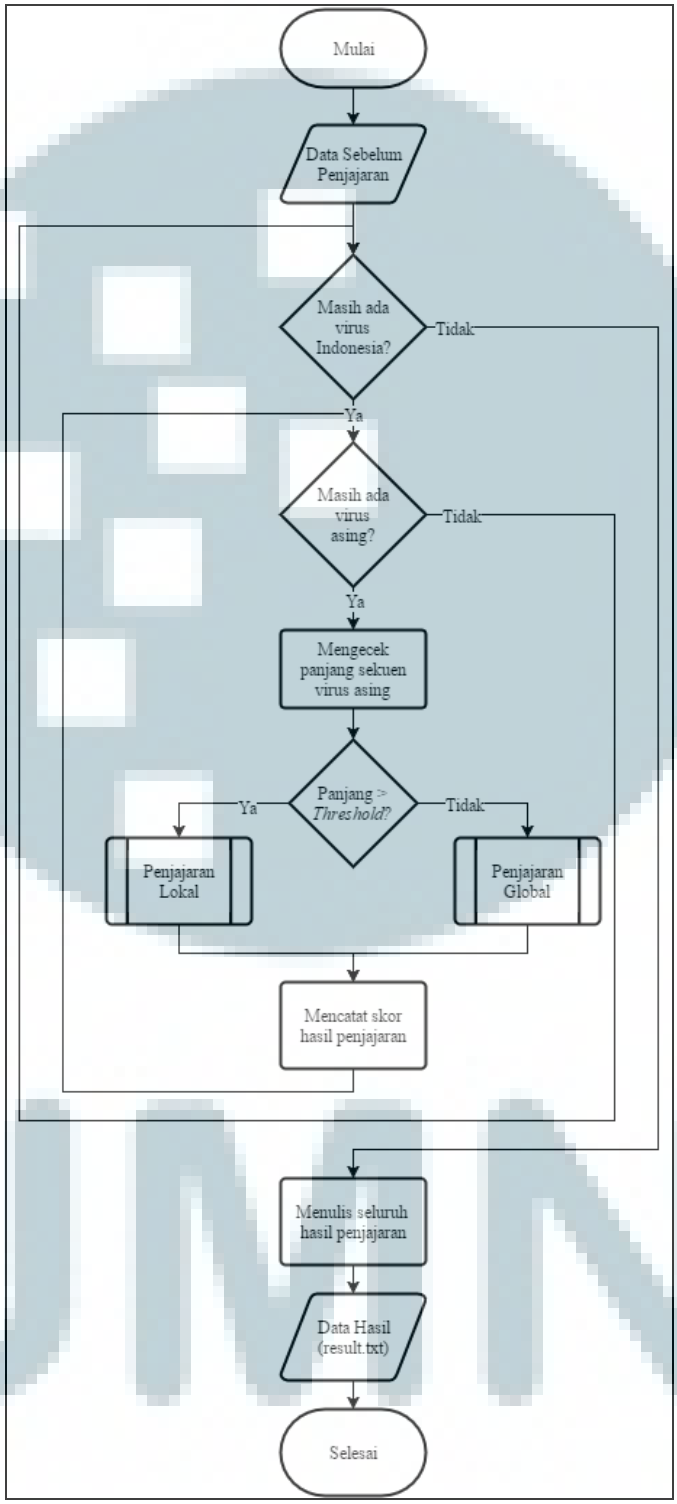


Gambar 3.4 *Flowchart* Detil Proses Memilah Virus

Proses memilah virus dimulai dengan mengecek apakah masih ada virus yang akan diperiksa. Jika sudah tidak ada virus yang tersisa, maka proses akan berakhir. Jika masih ada virus, maka lokasi virus ini akan diperiksa. Jika lokasi virus berasal dari Indonesia, maka virus akan dimasukkan ke *list indonesian_virus.* Jika lokasi virus bukan berasal dari Indonesia, maka virus akan dimasukkan ke *list foreign_virus.* Setelah itu, virus akan dimasukkan ke *list all_virus*, yang akan digunakan dalam tahap membangun *cluster.* Proses ini akan dilakukan hingga seluruh virus telah masuk ke dalam *list* yang sesuai.

E. Flowchart Detil Proses Penjajaran Sekuen

Alur kerja detil proses penjajaran sekuen dapat dilihat pada Gambar 3.5.

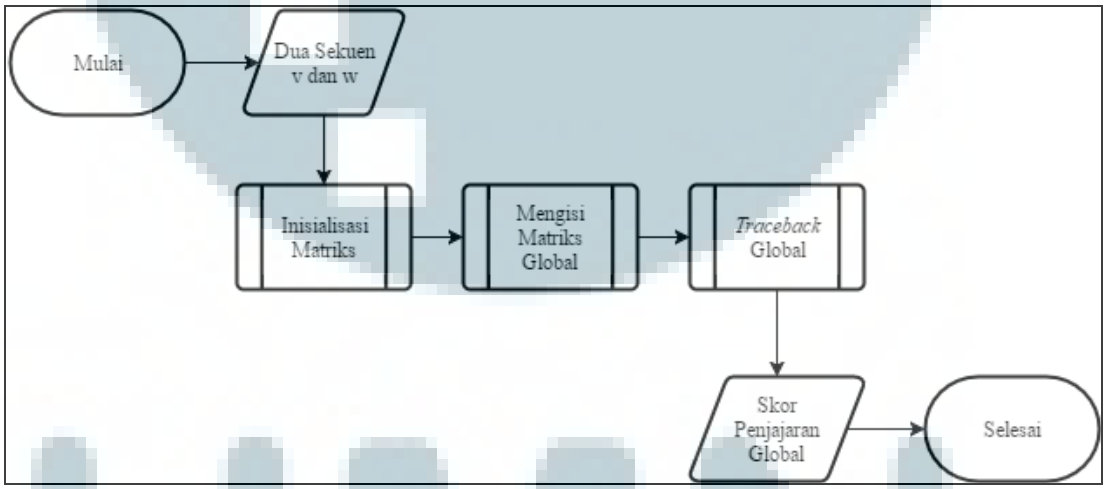


Gambar 3.5 *Flowchart* Detil Proses Penjajaran Sekuen

Proses penjajaran sekuen menerima input berupa data sebelum penjajaran dari proses penyiapan data. Selanjutnya, seluruh virus Indonesia akan dibandingkan dengan virus asing. Panjang virus asing akan dibandingkan dengan *threshold*. Jika panjang virus asing lebih besar daripada *threshold*, maka kedua virus akan dijajarkan secara lokal. Jika tidak, maka kedua virus akan dijajarkan secara global. Skor hasil setiap penjajaran ini akan dicatat. Jika seluruh virus Indonesia dan virus asing telah selesai dibandingkan, maka seluruh hasil penjajaran akan dituliskan ke 'result.txt'. Hasil ini akan digunakan dalam analisis dalam bentuk tabel dan grafik.

F. Flowchart Detil Proses Penjajaran Global

Alur kerja detil proses penjajaran global dapat dilihat pada Gambar 3.6.

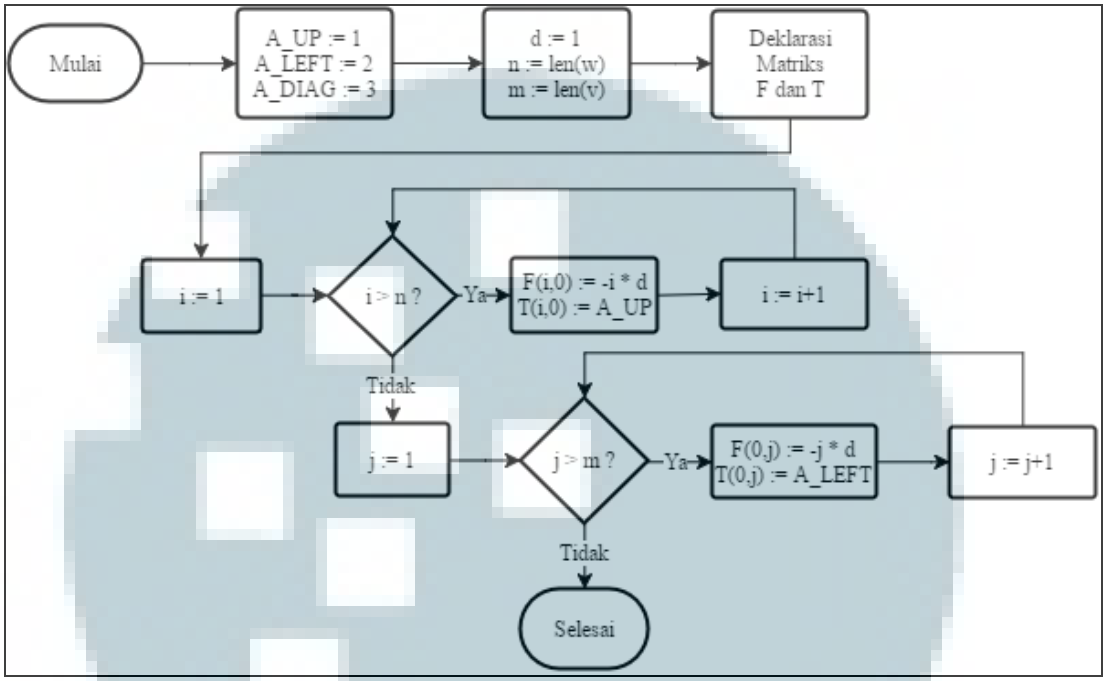


Gambar 3.6 *Flowchart* Detil Proses Penjajaran Global

Proses penjajaran global menerima data dua sekuen (*v* dan *w*) yang ingin dijajarkan secara global. Secara umum, proses ini terbagi menjadi tiga tahap, yaitu inisialisasi matriks, mengisi matriks global, dan *traceback* global. Hasil akhir proses ini adalah skor penjajaran global yang akan digunakan dalam analisis.

G. Flowchart Detil Proses Inisialisasi Matriks

Alur kerja detil proses inisialisasi matriks dapat dilihat pada Gambar 3.7.

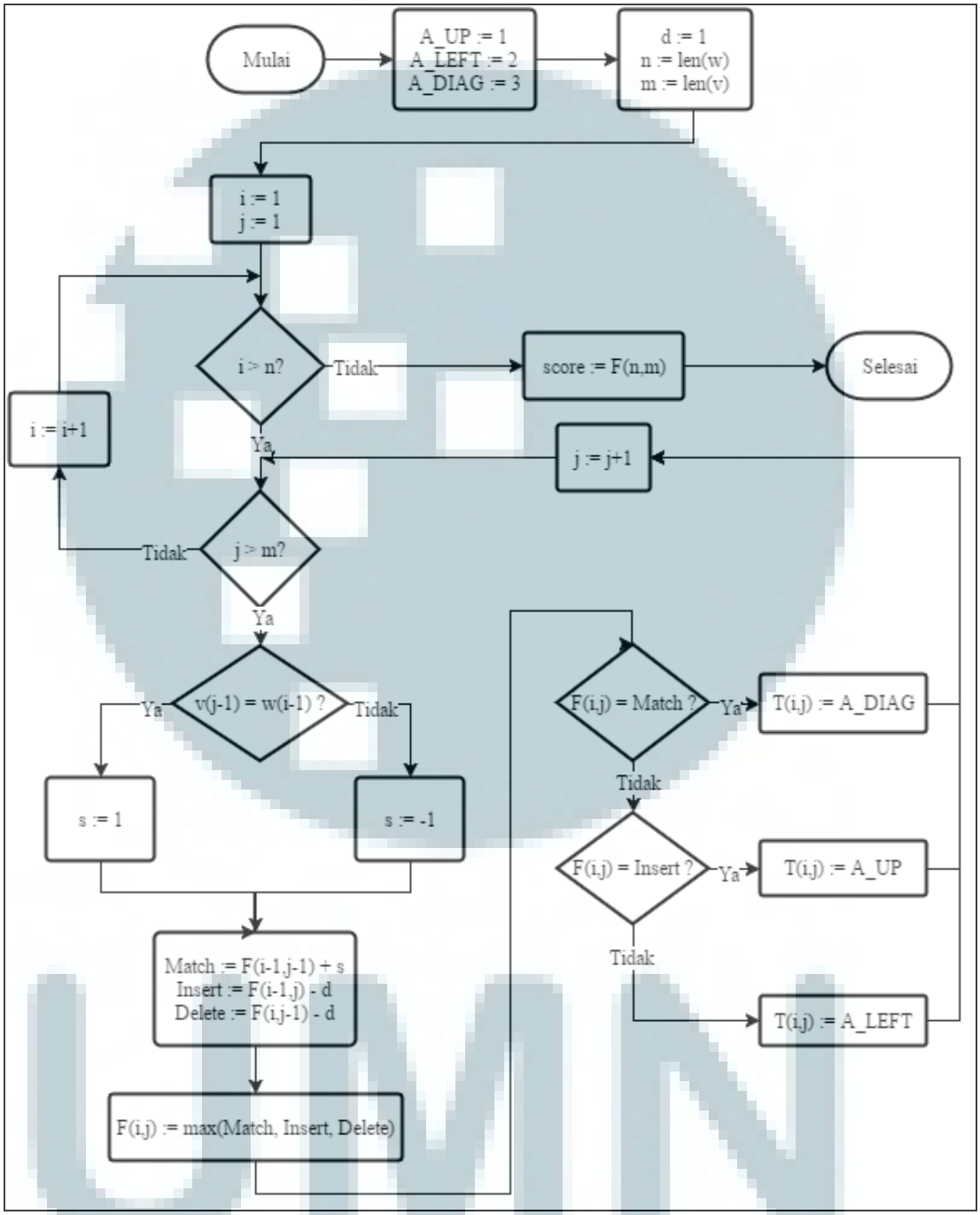


Gambar 3.7 *Flowchart* Detil Proses Inisialisasi Matriks

A_UP, *A_LEFT*, dan *A_DIAG* merupakan enumerasi yang melambangkan arah *traceback.* Variabel *d* merupakan *gap, m* merupakan panjang sekuen *v*, dan *n* merupakan panjang sekuen *w*. Matriks *F* (matriks penjajaran) dan *T* (matriks *traceback*) dideklarasi dalam bentuk *list* dua dimensi, dengan ukuran *n* baris dan *m* kolom (*n* x *m*). Pada Matriks *F*, untuk setiap baris *i* pada kolom pertama diisi dengan nilai *–i*d* dan untuk setiap kolom *j* pada baris pertama diisi dengan nilai *–j*d.* Kemudian, pada Matriks *T*, untuk setiap baris *i* pada kolom pertama diisi dengan nilai *A_UP* dan untuk setiap kolom *j* pada baris pertama diisi dengan nilai *A_LEFT.*

H. Flowchart Detil Proses Mengisi Matriks Global

Alur kerja detil proses mengisi matriks global dapat dilihat pada Gambar 3.8.

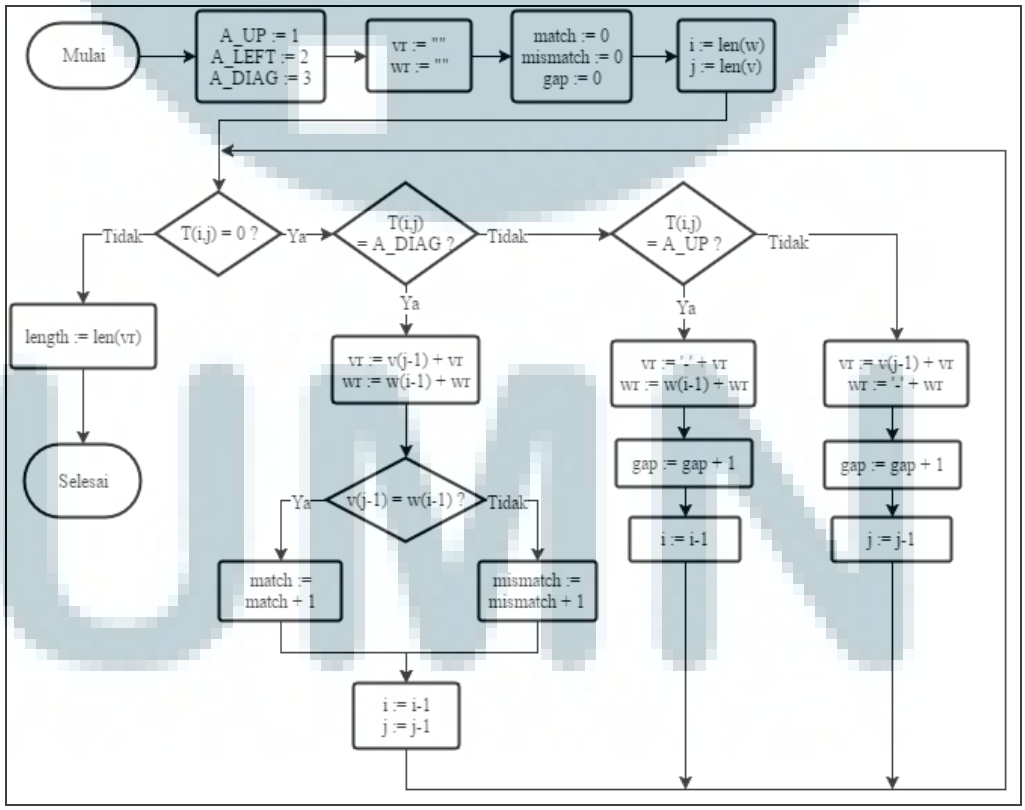


Gambar 3.8 *Flowchart* Detil Proses Mengisi Matriks Global

A_UP, *A_LEFT*, dan *A_DIAG* merupakan enumerasi yang melambangkan arah *traceback*. Variabel *d* merupakan *gap*, *m* merupakan panjang sekuen *v*, dan *n* merupakan panjang sekuen *w*. Untuk setiap baris *i* kolom *j*, hitung nilai *s*, yaitu nilai perbandingan antara *v*(*j*-1) dan *w*(*j*-1). Jika sama (*match*), maka nilai *s* adalah 1. Jika berbeda (*mismatch*), maka nilai *s* adalah -1. Kemudian, isi *F*(*i*,*j*) dengan nilai maksimum dari *Match*, *Insert*, dan *Delete*. Jika nilai *F*(*i*,*j*) sama dengan *Match*, maka nilai *T*(*i*,*j*) adalah *A_DIAG*. Jika nilai *F*(*i*,*j*) sama dengan *Insert*, maka nilai *T*(*i*,*j*) adalah *A_UP*. Jika nilai *F*(*i*,*j*) sama dengan *Delete*, maka nilai *T*(*i*,*j*) adalah *A_LEFT*. Proses ini akan menghasilkan nilai *score*, yang akan terisi pada *F*(*n*,*m*).

I. Flowchart Detil Proses Traceback Global

Alur kerja detil proses *traceback* global dapat dilihat pada Gambar 3.9.

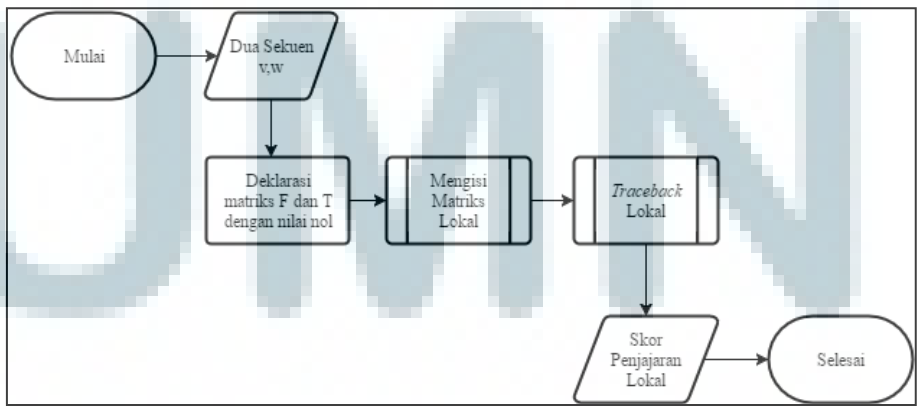


Gambar 3.9 *Flowchart* Detil Proses *Traceback* Global

A_UP, *A_LEFT*, dan *A_DIAG* merupakan enumerasi yang melambangkan arah *traceback.* Variabel *vr* dan *wr* adalah *string* hasil penjajaran untuk sekuen *v* dan *w.* Proses *traceback* dimulai dari sel baris terakhir kolom terakhir pada matriks *T.* Untuk setiap pengulangan, jika nilai *T*(*i*,*j*) adalah *A_DIAG*, maka tambahkan karakter *v*(*j*-1) pada bagian depan *string vr* dan tambahkan karakter *w*(*i*-1) pada bagian depan *string wr*. Jika nilai kedua karakter ini sama, maka nilai *match* bertambah satu. Jika tidak, maka nilai *mismatch* bertambah satu. Kemudian, pengecekan akan bergeser ke kolom kiri atasnya. Selanjutnya, jika nilai *T*(*i*,*j*) adalah *A_UP*, maka tambahkan karakter *dash* (-) pada bagian depan *string vr*, tambahkan karakter *w*(*i*-1) pada bagian depan *string wr,* dan nilai *gap* bertambah satu. Jika nilai *T*(*i*,*j*) adalah *A_LEFT*, maka tambahkan karakter karakter *v*(*j*-1) pada bagian depan *string vr*, tambahkan karakter *dash* (-) pada bagian depan *string wr*, dan nilai *gap* bertambah satu. Proses ini berlanjut hingga nilai *T(i,j)* adalah nol, yaitu sel di baris pertama kolom pertama. Proses ini akan menghasilkan nilai *length* (panjang sekuen yang dijajarkan), *match*, *mismatch*, dan *gap.*

J. Flowchart Detil Proses Penjajaran Lokal

Alur kerja detil proses penjajaran lokal dapat dilihat pada Gambar 3.10.

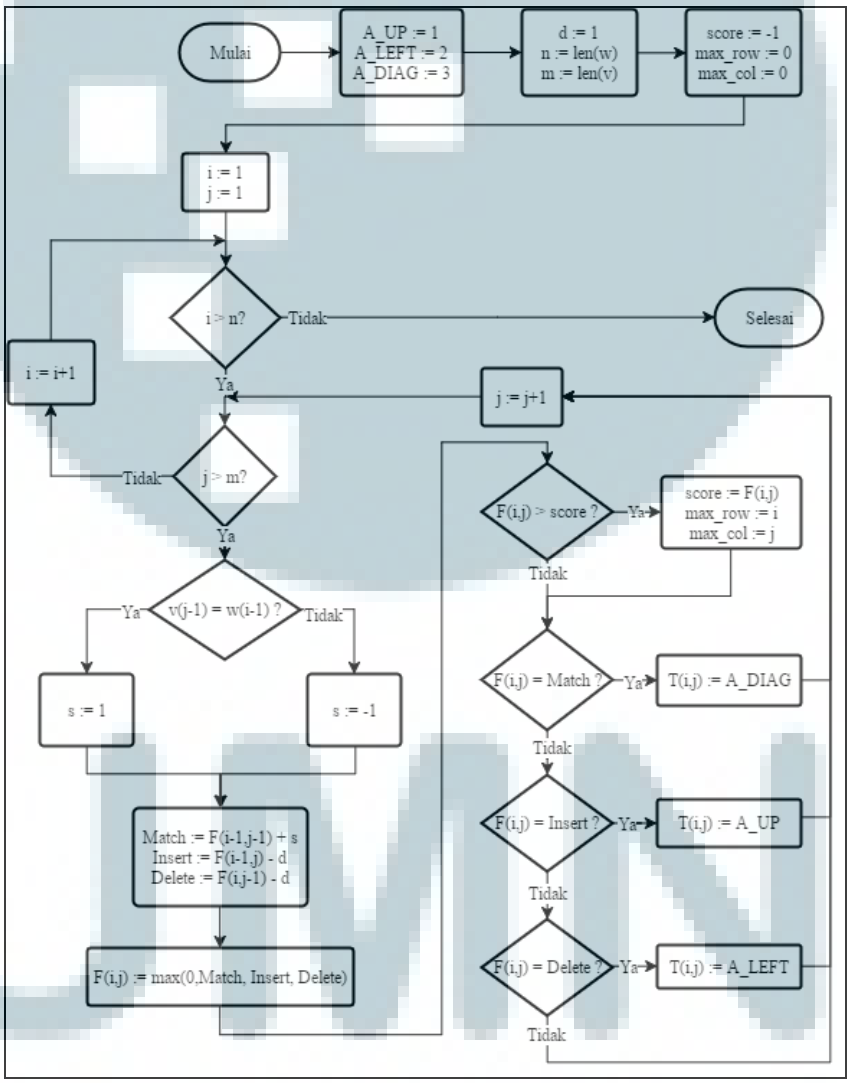


Gambar 3.10 *Flowchart* Detil Proses Penjajaran Lokal

Proses penjajaran lokal menerima data dua sekuen (*v* dan *w*) yang ingin dijajarkan secara lokal. Secara umum, proses ini terbagi menjadi tiga tahap, yaitu deklarasi matriks *F* dan *T* dengan nilai nol, mengisi matriks lokal, dan *traceback* lokal. Hasil akhir proses ini adalah skor penjajaran lokal yang akan digunakan dalam analisis.

K. Flowchart Detil Proses Mengisi Matriks Lokal

Alur kerja detil proses mengisi matriks lokal dapat dilihat pada Gambar 3.11.



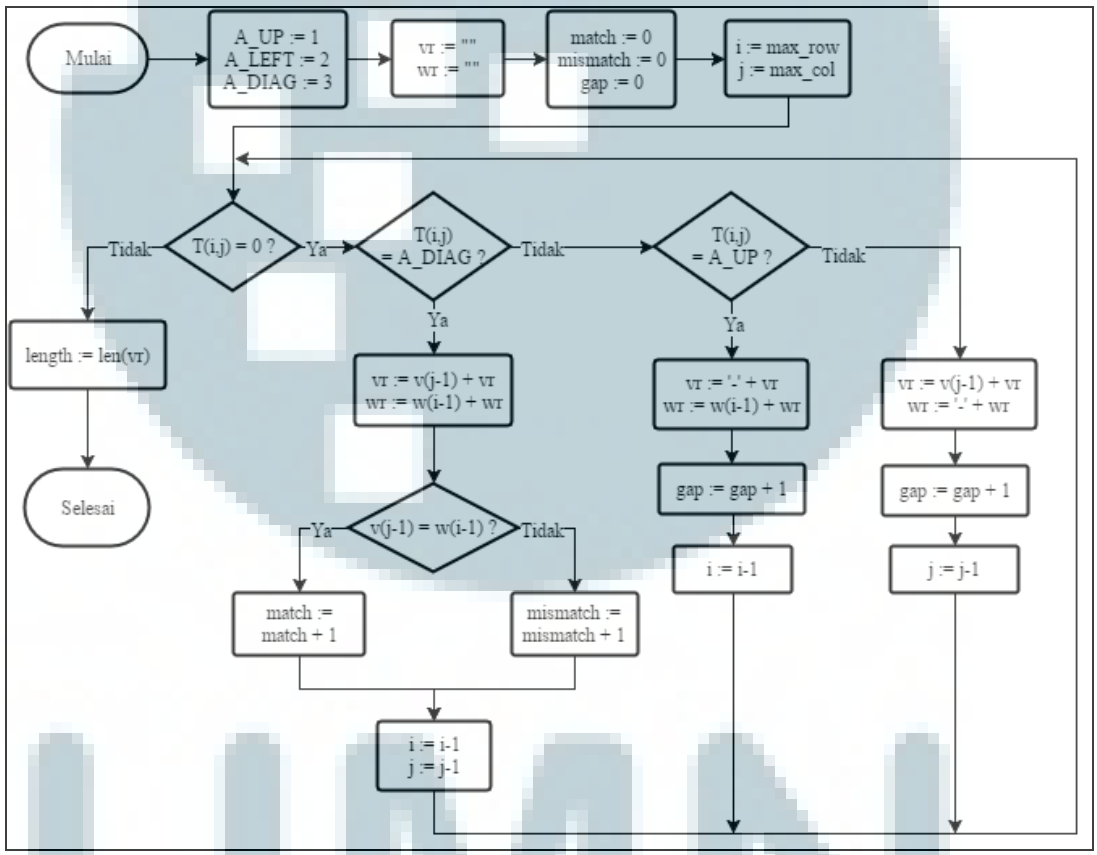
Gambar 3.11 *Flowchart* Detil Proses Mengisi Matriks Lokal

A_UP, *A_LEFT*, dan *A_DIAG* merupakan enumerasi yang melambangkan arah *traceback.* Variabel *d* merupakan *gap, m* merupakan panjang sekuen *v*, dan *n* merupakan panjang sekuen *w*. Variabel *score, max_row*, dan *max_col* digunakan untuk menentukan nilai maksimum dan lokasi sel. Untuk setiap baris *i* kolom *j*, hitung nilai *s*, yaitu nilai perbandingan antara *v*(*j*-1) dan *w*(*j*-1). Jika sama (*match*), maka nilai *s* adalah 1. Jika berbeda (*mismatch*), maka nilai *s* adalah -1. Kemudian, isi *F*(*i*,*j*) dengan nilai maksimum dari 0, *Match*, *Insert*, dan *Delete.* Jika nilai *F*(*i*,*j*) lebih besar daripada *score,* maka *update* nilai *F*(*i*,*j*) dengan nilai *score,* kemudian catat lokasi baris dan kolom saat ini ke variabel *max_row* dan *max_col*. Jika nilai *F*(*i*,*j*) sama dengan *Match,* maka nilai *T*(*i*,*j*) adalah *A_DIAG.* Jika nilai *F*(*i*,*j*) sama dengan *Insert,* maka nilai *T*(*i*,*j*) adalah *A_UP*. Jika nilai *F*(*i*,*j*) sama dengan *Delete,* maka nilai *T*(*i*,*j*) adalah *A_LEFT*. Proses ini akan menghasilkan nilai *score* maksimum dan lokasi sel *score* maksimum pada *max_row* dan *max_col*.

L. Flowchart Detil Proses Traceback Lokal

Alur kerja detil proses *traceback* lokal dapat dilihat pada Gambar 3.12. *A_UP*, *A_LEFT*, dan *A_DIAG* merupakan enumerasi yang melambangkan arah *traceback.* Variabel *vr* dan *wr* adalah *string* hasil penjajaran untuk sekuen *v* dan *w*. Proses *traceback* dimulai dari sel dengan nilai maksimum, yang indeksnya terdapat pada *max_row* dan *max_col*. Untuk setiap pengulangan, jika nilai *T*(*i*,*j*) adalah *A_DIAG*, maka tambahkan karakter *v*(*j*-1) pada bagian depan *string vr* dan tambahkan karakter *w*(*i*-1) pada bagian depan *string wr*. Jika nilai kedua karakter ini sama, maka nilai *match* bertambah satu. Jika tidak, maka nilai *mismatch* bertambah satu. Kemudian, pengecekan akan bergeser ke kolom kiri atasnya. Selanjutnya, jika nilai *T*(*i*,*j*) adalah

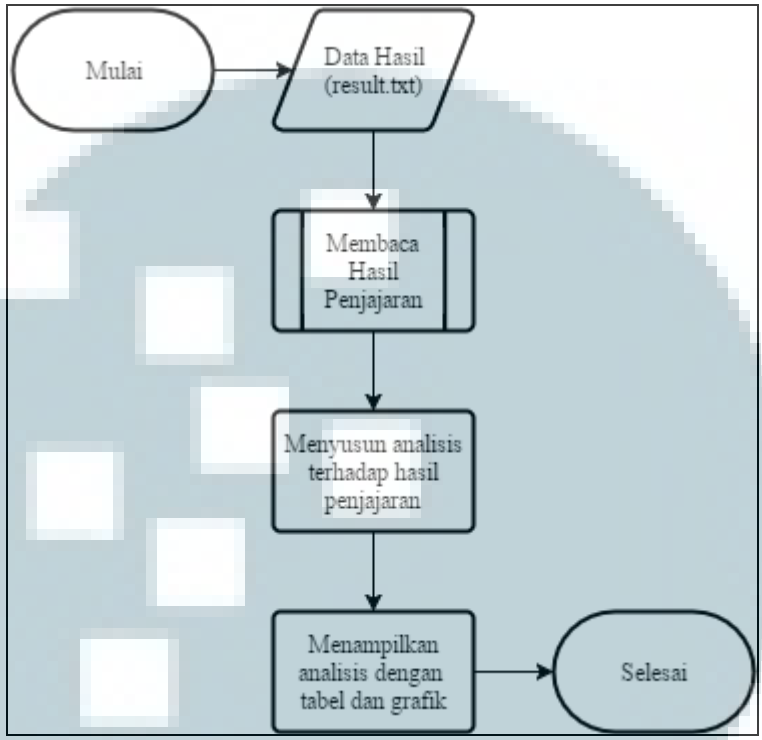
A_UP, maka tambahkan karakter *dash* (-) pada bagian depan *string vr*, tambahkan karakter *w*(*i*-1) pada bagian depan *string wr*, dan nilai *gap* bertambah satu. Jika nilai *T*(*i*,*j*) adalah *A_LEFT*, maka tambahkan karakter karakter *v*(*j*-1) pada bagian depan *string vr*, tambahkan karakter *dash* (-) pada bagian depan *string wr*, dan nilai *gap* bertambah satu. Proses ini berlanjut hingga nilai *T(i,j)* adalah nol. Proses ini akan menghasilkan nilai *length* (panjang sekuen yang dijajarkan), *match*, *mismatch*, dan *gap*.



Gambar 3.12 *Flowchart* Detil Proses Traceback Lokal

M. Flowchart Detil Proses Analisis dan Evaluasi

Alur kerja detail proses analisis dan evaluasi dapat dilihat pada Gambar 3.13.

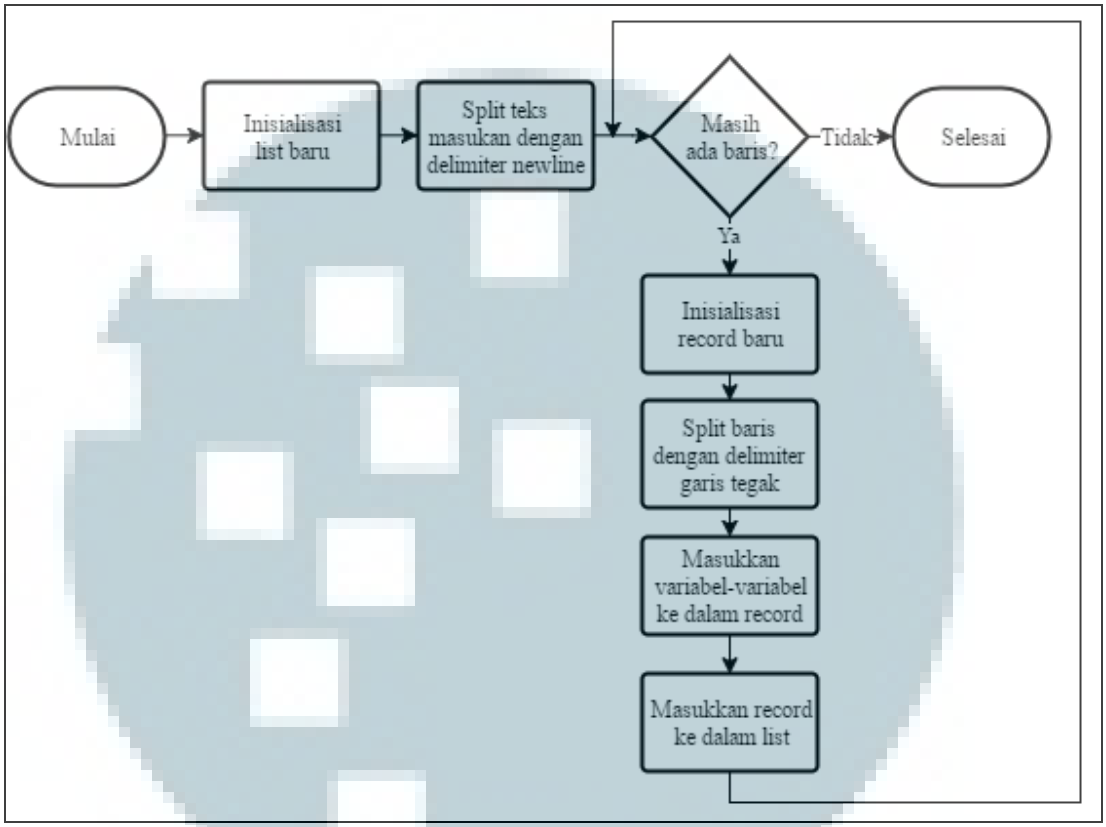


Gambar 3.13 *Flowchart* Detil Proses Analisis dan Evaluasi

Proses ini menerima masukan berupa data hasil penjajaran yang tersimpan pada 'result.txt'. Kemudian, data hasil ini akan dibaca oleh proses membaca hasil penjajaran untuk mendapatkan informasi mengenai hasil penjajaran yang telah dilakukan. Setelah data disiapkan oleh proses membaca hasil penjajaran, langkah selanjutnya adalah menyusun analisis terhadap hasil penjajaran. Analisis ini bertujuan untuk memperoleh *insight* mengenai hasil penelitian yang telah dilakukan. Terdapat beberapa analisis yang dilakukan dalam penelitian ini, di antaranya analisis pengaruh panjang sekuen terhadap waktu eksekusi algoritma, perbandingan *score* algoritma penjajaran global dan penjajaran lokal, dan penentuan *score* serta *identity* terbaik. Analisis yang telah disusun akan disajikan dalam bentuk tabel atau bentuk grafik.

N. Flowchart Detil Proses Membaca Hasil Penjajaran

Alur kerja detail membaca hasil penjajaran dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Flowchart* Detil Proses Membaca Hasil Penjajaran

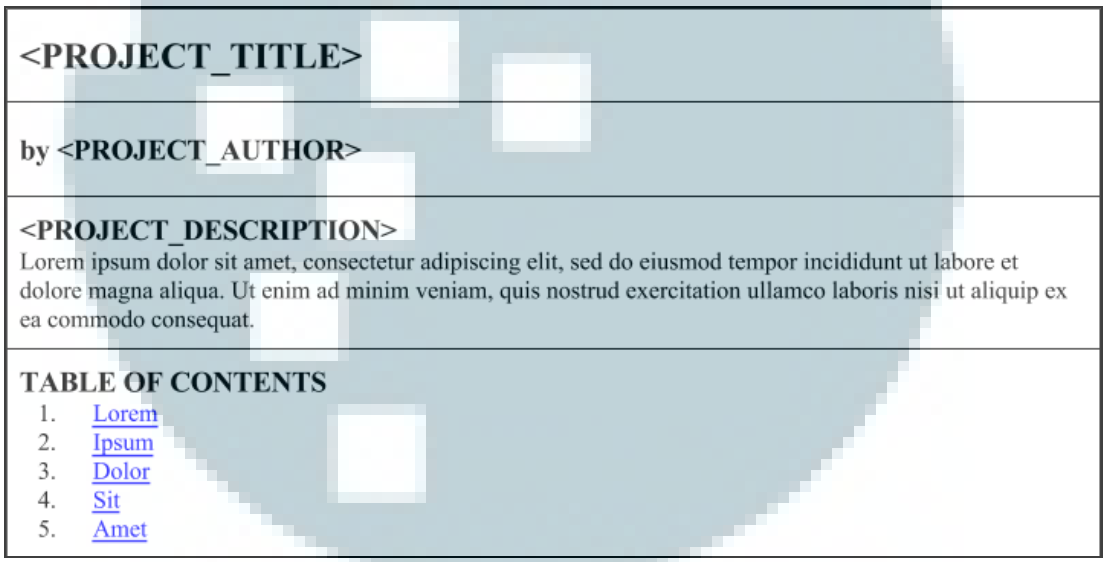
Proses ini dimulai dengan inisialisasi list baru. Kemudian, teks masukan di-*split* dengan *delimiter newline* (‘\n’) untuk mendapatkan baris-baris *record.* Setiap baris *record* ini akan dimasukkan ke dalam *list*, dengan langkah-langkah sebagai berikut. Pertama, dilakukan inisialisasi terhadap *record* baru. Kedua, teks dalam baris tersebut akan di-*split* dengan *delimiter* garis tegak (‘|’). Hasil *split* ini adalah variabel-variabel dalam baris tersebut. Variabel-variabel ini kemudian dimasukkan ke dalam *record* yang telah dibuat. Terakhir, *record* tersebut akan dimasukkan ke dalam *list.* Proses ini akan berakhir setelah seluruh baris *record* telah dimasukkan ke dalam *list.*

3.2.2 Desain Antarmuka

Desain antarmuka dibuat untuk merancang sistematika halaman penelitian agar dapat lebih intuitif dan informatif untuk digunakan. Berikut ini adalah desain antarmuka yang dilakukan dalam penelitian ini.

A. Bagian Header

Desain antarmuka bagian *header* dapat dilihat pada Gambar 3.15.

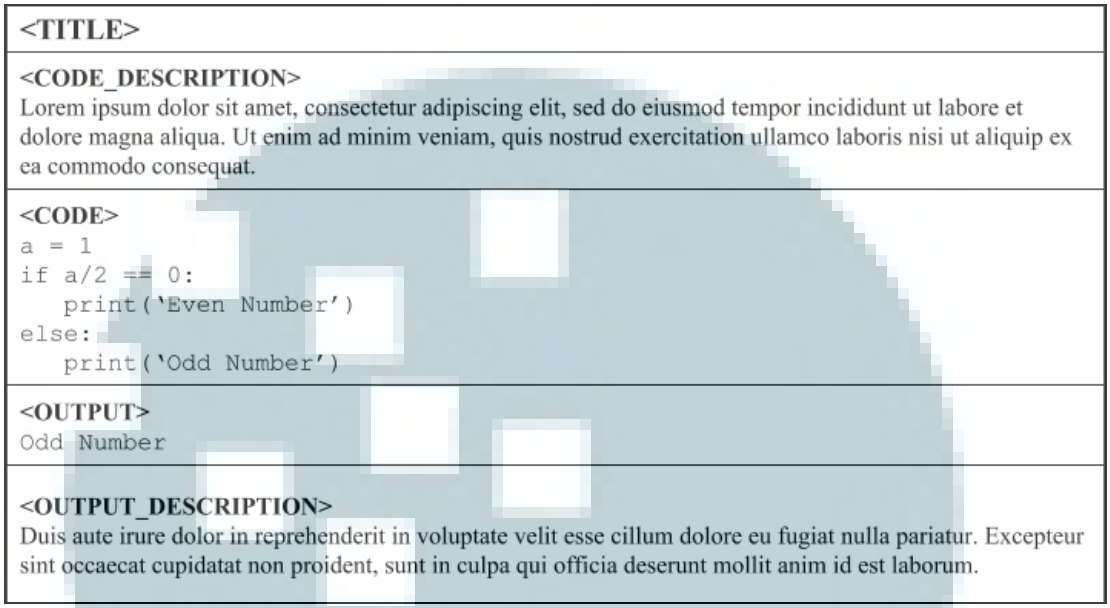


Gambar 3.15 Desain antarmuka bagian *header*

Bagian *header* terdiri atas empat bagian utama, yaitu *title*, *author*, *description*, dan *table of contents*. Bagian *title* merupakan judul dari penelitian yang dilakukan. Bagian *author* berisi nama peneliti yang melakukan penelitian. Bagian *description* berisi deskripsi singkat mengenai penelitian yang dilakukan. Bagian *table of contents* berisi daftar isi dari halaman penelitian, yang dapat diklik untuk langsung menuju bagian penelitian yang ingin dilihat.

B. Bagian Kode

Desain antarmuka bagian kode dapat dilihat pada Gambar 3.16.



Gambar 3.16 Desain antarmuka bagian kode

Bagian kode terdiri atas lima bagian utama, yaitu *title, code description, code, output* dan *output description.* Bagian *title* merupakan judul dari bagian kode yang dituliskan. Bagian *code description* merupakan deskripsi singkat atas kode yang dituliskan. Bagian *code* berisi kode yang dituliskan dengan bahasa pemrograman Python. Bagian *output* berisi keluaran setelah kode dieksekusi. Keluaran ini dapat berupa teks, tabel, maupun grafik. Bagian *output description* berisi penjelasan singkat atas keluaran yang dihasilkan. Jika potongan kode yang dituliskan tidak menghasilkan keluaran, maka bagian *output* dan *output description* tidak akan muncul.