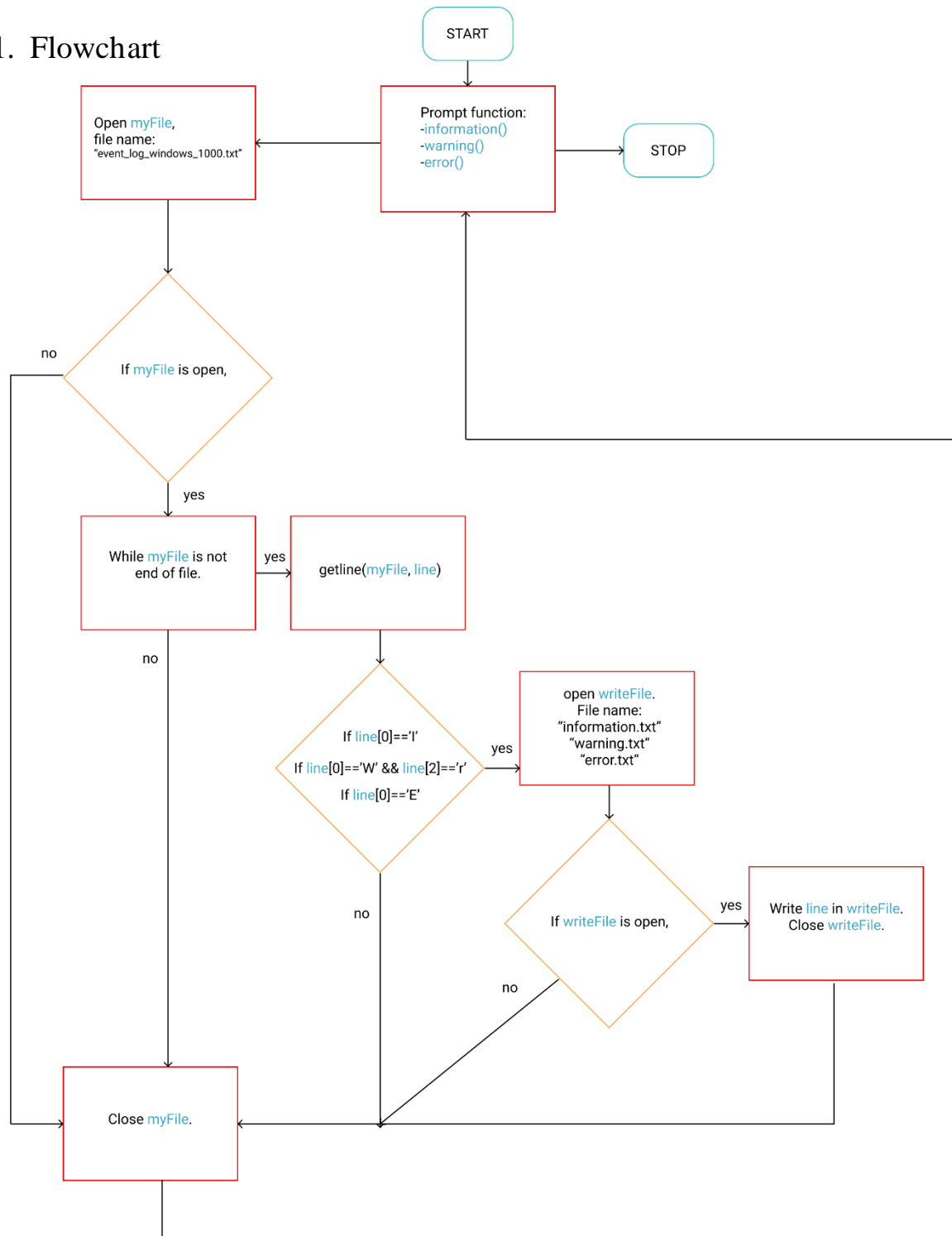


Tugas Pengolahan File Teks

Qornain Aji

21/481757/TK/53170

1. Flowchart



2. Pembahasan

Program yang saya buat adalah program untuk memilah data log “event_log_windows_1000.txt” dan memasukkannya ke dalam tiga file, yaitu, “information.txt”, “warning.txt”, “error.txt”. Saya menggunakan pendekatan *modular programming* sehingga dapat menghemat memori dari komputer menghindari error dengan penggunaan variabel lokal.

Program dimulai dengan pendefinisian modul function seperti void `information(string *word)`, void `warning(string *word)`, void `error(string *word)` dan dilanjutkan dengan memanggil modul function tersebut di dalam `int main()`. Penggunaan variabel pointer diatas hanya sebagai parameter penghubung saja dan tidak ada fungsi yang lain. Algoritma dari fungsi modular yang saya buat bisa terbilang sama, tetapi hanya berbeda pada huruf yang diambil sebagai *conditional statement* pada fungsi if-nya. Fungsi dimulai dengan membuka file “event_log_windows_1000.txt” dan di *assign* pada variabel `ifstream myFile`. Selanjutnya, jika `myFile` terbuka, maka akan lanjut pada proses iterasi `while`. Jika tidak, `myFile` akan ditutup dan proses akan selesai dan tidak terjadi apa-apa. Langkah ini bertujuan agar kita dapat memastikan ada tidaknya file bernama “event_log_windows_1000.txt”.

Proses iterasi `while` berjalan selama `myFile` yang dibaca belum sampai pada akhir dari file (EOF). Jika file tidak EOF, maka akan berlanjut ke proses dalam iterasi. Jika file EOF, maka `myFile` akan ditutup. Proses di dalam berlanjut dengan mengambil satu baris pada file “event_log_windows_1000.txt” dan meng-*assign*-kan pada variabel lokal `string line`. Perintah berikutnya adalah jika index huruf pertama pada `line` tersebut sesuai dengan persyaratan kategori file misal, ‘I’ untuk file “information.txt”, maka baris yang terdapat huruf I pada huruf pertamanya akan di masukkan ke dalam file “information.txt”. Perintah ini juga berlaku pada *function modul* `warning()` dan `error()`. Jika tidak memenuhi syarat, maka akan di lewati baris tersebut dan akan dilanjutkan dengan proses iterasi kedua.

Proses memasukkan `line` ke dalam file “information.txt” dilakukan dengan membuka file “information.txt” dan di-assign pada variabel lokal `ofstream writeFile`. Jika `writeFile` terbuka, maka baris tersebut akan dicetak ke dalam `writeFile` dan `writeFile` akan ditutup untuk proses iterasi pertama. Penutupan `writeFile` dilakukan agar program dapat melakukan perintah selanjutnya. Jika tidak, program akan terhenti hanya pada 1 baris awal saja. Setelah baris berhasil di

masukkan hingga ke baris yang terakhir dari file “event_log_windows_1000.txt”, program akan exit dari iterasi dan menutup myFile. Setelah modul function berhasil dijalankan, program akan return 0 sebagai output dari program dan program telah selesai dijalankan.

3. Source code

```
4. #include<iostream>
5. #include<fstream>
6.
7. using namespace std;
8.
9. void information(string *word);
10. void warning(string *word);
11. void error(string *word);
12.
13. int main()
14. {
15.
16.
17.
18.
19.     string sentence;
20.     information(&sentence);
21.     warning(&sentence);
22.     error(&sentence);
23.
24.
25.
26.
27.
28.     return 0;
29. }
30.
31. void information(string *word)
32. {
33.     ifstream myFile;
34.     ofstream writeFile;
35.     string line;
36.     int offset;
37.
```

```

38.     myFile.open("event_log_windows_1000.txt");
39.     if(myFile.is_open())
40.     {
41.         while(!myFile.eof())
42.         {
43.
44.             getline(myFile, line);
45.             if(line[0] == 'I')
46.             {
47.
48.                 writeFile.open("information.txt", ios::app);
49.                 if(writeFile.is_open())
50.                 {
51.
52.                     writeFile<<line<<endl;
53.                     writeFile.close();
54.                 }
55.
56.             }
57.
58.         }
59.     }
60. }
61.
62. myFile.close();
63.
64.
65. }
66. void warning(string *word)
67. {
68.     ifstream myFile;
69.     ofstream writeFile;
70.     string line;
71.     int offset;
72.
73.     myFile.open("event_log_windows_1000.txt");
74.     if(myFile.is_open())
75.     {
76.         while(!myFile.eof())
77.         {
78.
79.             getline(myFile, line);
80.             if(line[0] == 'W' && line[2] == 'r')
81.             {
82.

```

```
83.         writeFile.open("warning.txt", ios::app);
84.         if(writeFile.is_open())
85.         {
86.
87.             writeFile<<line<<endl;
88.             writeFile.close();
89.
90.         }
91.
92.     }
93.
94. }
95.
96. }
97.
98.
99. myFile.close();
100.
101. }
102. void error(string *word)
103. {
104.     ifstream myFile;
105.     ofstream writeFile;
106.     string line;
107.     int offset;
108.
109.     myFile.open("event_log_windows_1000.txt");
110.     if(myFile.is_open())
111.     {
112.         while(!myFile.eof())
113.         {
114.
115.             getline(myFile, line);
116.             if(line[0] == 'E')
117.             {
118.
119.                 writeFile.open("error.txt", ios::app);
120.                 if(writeFile.is_open())
121.                 {
122.
123.                     writeFile<<line<<endl;
124.                     writeFile.close();
125.                 }
126.
127.             }
```

```
128.  
129.         }  
130.  
131.     }  
132.  
133.  
134.     myFile.close();  
135.  
136.  
137. }
```