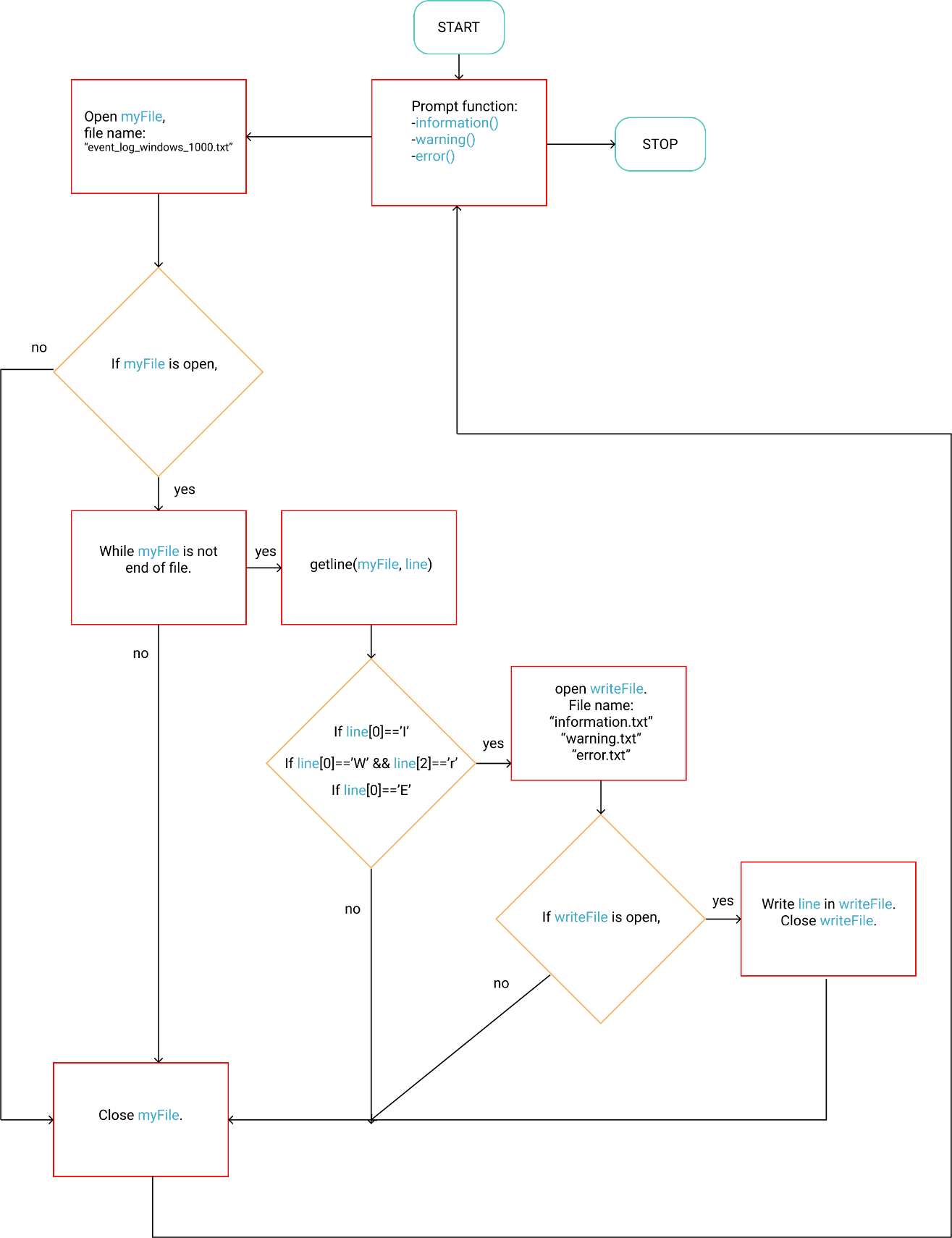
Tugas Pengolahan File Teks

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 memasukkan nya kedalam 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 fuction 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 dibilang 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 *funtion 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 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.

1. Source code
2. #include<iostream>
3. #include<fstream>
4. using namespace std;
5. void information(string \*word);
6. void warning(string \*word);
7. void error(string \*word);
8. int main()
9. {


13. string sentence;
14. information(&sentence);
15. warning(&sentence);
16. error(&sentence);


20. return 0;
21. }
22. void information(string \*word)
23. {
24. ifstream myFile;
25. ofstream writeFile;
26. string line;
27. int offset;
29. myFile.open("event\_log\_windows\_1000.txt");
30. if(myFile.is\_open())
31. {
32. while(!myFile.eof())
33. {
35. getline(myFile, line);
36. if(line[0] == 'I')
37. {
39. writeFile.open("information.txt", ios::app);
40. if(writeFile.is\_open())
41. {
42. writeFile<<line<<endl;
43. writeFile.close();
44. }
46. }
48. }
50. }
52. myFile.close();
54. }
55. void warning(string \*word)
56. {
57. ifstream myFile;
58. ofstream writeFile;
59. string line;
60. int offset;
62. myFile.open("event\_log\_windows\_1000.txt");
63. if(myFile.is\_open())
64. {
65. while(!myFile.eof())
66. {
68. getline(myFile, line);
69. if(line[0] == 'W' && line[2] == 'r')
70. {
72. writeFile.open("warning.txt", ios::app);
73. if(writeFile.is\_open())
74. {
75. writeFile<<line<<endl;
76. writeFile.close();
77. }
79. }
81. }
83. }
85. myFile.close();
86. }
87. void error(string \*word)
88. {
89. ifstream myFile;
90. ofstream writeFile;
91. string line;
92. int offset;
94. myFile.open("event\_log\_windows\_1000.txt");
95. if(myFile.is\_open())
96. {
97. while(!myFile.eof())
98. {
100. getline(myFile, line);
101. if(line[0] == 'E')
102. {
104. writeFile.open("error.txt", ios::app);
105. if(writeFile.is\_open())
106. {
107. writeFile<<line<<endl;
108. writeFile.close();
109. }
111. }
113. }
115. }
117. myFile.close();
119. }