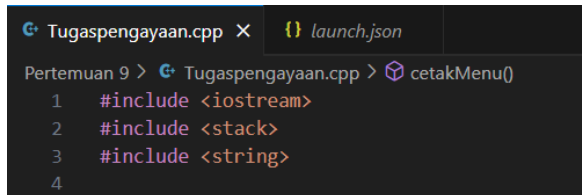


PENJELASAN CODINGAN PENGAYAAN STRUKTUR DATA

Nama : Faiz Izaz Fathoni

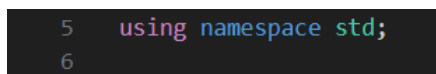
NIM : 20230801081



```
Tugaspengayaan.cpp X launch.json
Pertemuan 9 > Tugaspengayaan.cpp > cetakMenu()
1  #include <iostream>
2  #include <stack>
3  #include <string>
4
```

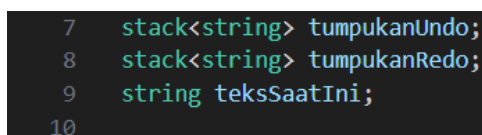
Terdapat beberapa Header yang digunakan pada program ini, yaitu

- iostream : Library ini digunakan agar tersedia fungsi-fungsi input dan output standar, seperti cin & cout
- stack : Library ini digunakan agar bisa mengakses fungsi stack
- string : Library ini digunakan agar bisa mengakses fungsi string



```
5  using namespace std;
6
```

Using namespace std digunakan pada C++ untuk menggunakan semua elemen dalam namespace 'std' secara default.



```
7  stack<string> tumpukanUndo;
8  stack<string> tumpukanRedo;
9  string teksSaatIni;
10
```

- Line 7: Sebuah stack 'tumpukanUndo' dengan tipe 'string' ini, digunakan untuk menyimpan keadaan teks sebelum perubahan terakhir untuk mendukung operasi undo.
- Line 8: Sebuah stack 'tumpukanRedo' dengan tipe 'string' ini, digunakan untuk menyimpan keadaan teks setelah sebuah undo dilakukan untuk mendukung operasi redo.
- Line 9: Sebuah variabel 'teksSaatIni' dengan tipe 'string' ini, digunakan untuk menyimpan teks saat ini yang sedang dikerjakan oleh user.

```

11 void tambahkanTeks(const string &teks) {
12     tumpukanUndo.push(teksSaatIni);
13     teksSaatIni += teks;
14     while (!tumpukanRedo.empty()) {
15         tumpukanRedo.pop();
16     }
17 }
18

```

Fungsi ‘tambahkanTeks’ digunakan agar user bisa menambahkan teks dengan cara menginput teks yang ingin ditambahkan.

```

19 void undo() {
20     if (!tumpukanUndo.empty()) {
21         tumpukanRedo.push(teksSaatIni);
22         teksSaatIni = tumpukanUndo.top();
23         tumpukanUndo.pop();
24     } else {
25         cout << "Tumpukan undo kosong.\n";
26     }
27 }
28

```

Fungsi ‘undo’ digunakan agar user bisa mengembalikan teks saat ini ke kondisi sebelum perubahan terakhir.

```

29 void redo() {
30     if (!tumpukanRedo.empty()) {
31         tumpukanUndo.push(teksSaatIni);
32         teksSaatIni = tumpukanRedo.top();
33         tumpukanRedo.pop();
34     } else {
35         cout << "Tumpukan redo kosong.\n";
36     }
37 }

```

Fungsi ‘redo’ digunakan agar user bisa mengulangi perubahan yang sebelumnya dibatalkan dengan undo.

```

38
39 bool tumpukanUndoKosong() {
40     return tumpukanUndo.empty();
41 }
42

```

Fungsi ‘tumpukanUndoKosong()’ digunakan untuk memeriksa apakah stack ‘tumpukanUndo’ kosong. Jika kosong akan mengembalikan nilai true, dan false jika tidak.

```

43 bool tumpukanRedoKosong() {
44     return tumpukanRedo.empty();
45 }
46

```

Fungsi ‘tumpukanRedoKosong()’ digunakan untuk memeriksa apakah stack ‘tumpukanRedo’ kosong. Jika kosong akan mengembalikan nilai true, dan false jika tidak.

```

47 void cetakTekssaatIni() {
48     cout << "\nTeks saat ini: " << teksSaatIni << endl;
49 }
50

```

Fungsi ‘cetakTekssaatIni()’ digunakan untuk mencetak seluruh teks yang telah ditambahkan oleh user (teksSaatIni) ke layar.

```

51 void cetakMenu() {
52     cout << "=====\n";
53     cout << "|          Menu          |\n";
54     cout << "=====\n";
55     cout << "| 1. Ketik teks          |\n";
56     cout << "| 2. Undo                |\n";
57     cout << "| 3. Redo                |\n";
58     cout << "| 4. IsEmptyUndo         |\n";
59     cout << "| 5. IsEmptyRedo         |\n";
60     cout << "| 6. Keluar              |\n";
61     cout << "=====\n";
62     cout << "Masukkan pilihan Anda: ";
63 }
64

```

Fungsi ‘cetakMenu’ digunakan untuk mencetak menu yang berisikan fungsi-fungsi yang tersedia untuk dipilih oleh user.

```

Tugaspengayaan.cpp x launch.json
Pertemuan 9 > Tugaspengayaan.cpp > cetakMenu()
65 void cetakTumpukanUndo() {
66     if (tumpukanUndoKosong()) {
67         cout << "Tumpukan undo kosong.\n";
68     } else {
69         cout << "Isi tumpukan undo:\n";
70         stack<string> temp = tumpukanUndo;
71         while (!temp.empty()) {
72             cout << "\"" << temp.top() << "\"\n";
73             temp.pop();
74         }
75     }
76 }
77

```

Fungsi ‘cetakTumpukanUndo()’ digunakan untuk mencetak isi dari stack ‘tumpukanUndo’. Jika kosong akan menampilkan keterangan tumpukan undo kosong, jika tidak kosong akan menampilkan setiap elemen dalam stack.

```

78 void cetakTumpukanRedo() {
79     if (tumpukanRedoKosong()) {
80         cout << "Tumpukan redo kosong.\n";
81     } else {
82         cout << "Isi tumpukan redo:\n";
83         stack<string> temp = tumpukanRedo;
84         while (!temp.empty()) {
85             cout << "\"" << temp.top() << "\"\n";
86             temp.pop();
87         }
88     }
89 }
90

```

Fungsi ‘cetakTumpukanRedo()’ digunakan untuk mencetak isi dari stack ‘tumpukanRedo’. Jika kosong akan menampilkan keterangan tumpukan undo kosong, jika tidak kosong akan menampilkan setiap elemen dalam stack.

```

91 int main() {
92     int pilihan;
93     string teks;
94

```

Pada fungsi utama di program ini, terdapat ‘int pilihan’ untuk menyimpan pilihan user dari menu, dan ‘string teks’ untuk menyimpan teks yang di input oleh user untuk ditambahkan.

```

95     do {
96         cetakTeksSaatIni();
97         cetakMenu();
98         cin >> pilihan;
99         cin.ignore();
100

```

Line 96 : syntax pada line ini akan memanggil fungsi ‘cetakTeksSaatIni’ yang berada di line 47-50.

Line 97: Syntax pada line ini akan memanggil fungsi ‘cetakMenu’ yang berada pada line 51 – 63.

Line 98-99: Syntax pada line ini ditujukan agar user dapat memasukkan angka sesuai dengan fungsi yang ingin dijalankan berdasarkan fungsi yang tersedia pada menu. Lalu user harus menekan enter untuk melanjutkan program.

```

101         switch (pilihan) {
102             case 1:
103                 cout << "Masukkan teks yang ingin ditambahkan: ";
104                 getline(cin, teks);
105                 tambahkanTeks(teks);
106                 cout << "\nTeks Berhasil Ditambahkan!! " << endl;
107                 cout << "Tekan Enter untuk melanjutkan...";
108                 cin.get();
109                 break;

```

Line 102 – 109: Syntax pada line ini akan berjalan jika user menginput 1 pada line 98, sehingga user bisa menginput teks yang ingin ditambahkan.

```

110             case 2:
111                 undo();
112                 cout << "Tekan Enter untuk melanjutkan...";
113                 cin.get();
114                 break;

```

Line 110 – 114: Syntax pada line ini akan berjalan jika user menginput 2 pada line 98, sehingga fungsi ‘undo’ pada line 19–28 akan terpanggil untuk dijalankan

```

115             case 3:
116                 redo();
117                 cout << "Tekan Enter untuk melanjutkan...";
118                 cin.get();
119                 break;

```

Line 115 – 119: Syntax pada line ini akan berjalan jika user menginput 3 pada line 98, sehingga fungsi ‘redo’ pada line 29–37 akan terpanggil untuk dijalankan.

```

120             case 4:
121                 cetakTumpukanUndo();
122                 cout << "Tekan Enter untuk melanjutkan...";
123                 cin.get();
124                 break;

```

Line 120 – 124: Syntax pada line ini akan berjalan jika user menginput 4 pada line 98, sehingga fungsi ‘cetakTumpukanUndo()’ pada line 65-76 akan terpanggil untuk dijalankan.

```

125             case 5:
126                 cetakTumpukanRedo();
127                 cout << "Tekan Enter untuk melanjutkan...";
128                 cin.get();
129                 break;

```

Line 125 – 129: Syntax pada line ini akan berjalan jika user menginput 5 pada line 98, sehingga fungsi ‘cetakTumpukanRedo()’ pada line 78-89 akan terpanggil untuk dijalankan.

```

130         case 6:
131             cout << "=====\n";
132             cout << "| Terima Kasih Telah Menggunakan Program Ini |\n";
133             cout << "=====\n";
134             break;

```

Line 130 – 134: Syntax pada line ini akan berjalan jika user menginput 6 pada line 98, sehingga program akan berhenti dan menampilkan pesan “Terima Kasih Telah Menggunakan Program Ini’

```

135         default:
136             cout << "Pilihan tidak valid. Silakan coba lagi.\n";
137             break;
138     }
139 } while (pilihan != 6);
140
141 return 0;
142 }

```

Line 135 – 142: Syntax pada line ini akan berjalan jika user menginput angka yang tidak terdapat di menu pada line 98. Maka user akan kembali pada proses pemilihan menu pada line 98.