

PENJELASAN TUGAS STRUKTUR DATA

NAMA = SATRIA ADI WIJAYA

NIM = 20230801102

Kelas `TextEditor`

Kelas ini mengelola teks yang sedang diedit dan menyimpan operasi undo dan redo menggunakan dua stack: `undoStack` dan `redoStack`.

Atribut

- `std::string text`: Menyimpan teks saat ini.
- `std::stack<std::string> undoStack`: Menyimpan sejarah teks untuk mendukung operasi undo.
- `std::stack<std::string> redoStack`: Menyimpan sejarah teks yang dapat di-redo setelah operasi undo.

Metode

- `void type(const std::string& newText)`
 - Menambahkan teks baru ke dalam dokumen.
 - Menyimpan keadaan teks saat ini ke `undoStack` sebelum menambahkan teks baru.
 - Mengosongkan `redoStack` karena ada operasi baru yang dilakukan.
- `void undo()`
 - Mengembalikan teks ke keadaan sebelumnya jika `undoStack` tidak kosong.
 - Menyimpan keadaan teks saat ini ke `redoStack` sebelum mengembalikan keadaan sebelumnya.
 - Jika `undoStack` kosong, menampilkan pesan bahwa stack undo kosong.
- `void redo()`
 - Mengembalikan teks ke keadaan setelah undo jika `redoStack` tidak kosong.
 - Menyimpan keadaan teks saat ini ke `undoStack` sebelum mengembalikan keadaan setelah undo.
 - Jika `redoStack` kosong, menampilkan pesan bahwa stack redo kosong.
- `bool isEmptyUndo() const`
 - Mengembalikan true jika `undoStack` kosong.
- `bool isEmptyRedo() const`
 - Mengembalikan true jika `redoStack` kosong.
- `void printText() const`
 - Menampilkan teks saat ini.

Fungsi `displayMenu`

- Menampilkan menu pilihan kepada pengguna, dengan opsi untuk mengetik teks, melakukan undo, redo, memeriksa apakah stack undo atau redo kosong, dan keluar dari program.

Fungsi `main`

- Mengelola logika utama dari program.
- Menginisialisasi objek `TextEditor`.
- Menggunakan loop tak hingga untuk menampilkan menu dan menerima input pengguna.
- Menggunakan struktur `switch` untuk menangani pilihan menu pengguna:
 - Pilihan 1 (**ketik**): Meminta pengguna memasukkan teks dan menambahkannya ke editor.
 - Pilihan 2 (**undo**): Melakukan operasi undo dan menampilkan teks saat ini.
 - Pilihan 3 (**redo**): Melakukan operasi redo dan menampilkan teks saat ini.
 - Pilihan 4 (**IsEmptyUndo**): Memeriksa apakah stack undo kosong.
 - Pilihan 5 (**IsEmptyRedo**): Memeriksa apakah stack redo kosong.
 - Pilihan 6 (**exit**): Keluar dari program.
- Menangani input yang tidak valid dengan menampilkan pesan kesalahan.

Contoh Penggunaan

1. jika memilih 1 untuk mengetik teks, masukkan kalimat apa saja saya berikan contoh "Bejo Kikir".
2. jika memilih 2 untuk melakukan undo, teks kembali ke keadaan sebelumnya. Yaitu "Bejo"
3. jika memilih 3 untuk melakukan redo, teks kembali ke "Bejo Kikir".
4. jika memilih 4 untuk **memeriksa apakah stack undo kosong**.
5. jika memilih 5 untuk **memeriksa apakah stack redo kosong**.
6. jika memilih 6 untuk keluar dari program.