

Penjelasan tugas redo/undo

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;
```

- Baris pertama hingga ketiga mendefinisikan penggunaan pustaka standar C++ yang diperlukan untuk program ini.
- `using namespace std;` memungkinkan kita menggunakan semua fungsi dan objek dalam ruang nama `std` tanpa menuliskan `std::` sebelumnya.

```
class UndoRedo {
private:
    stack<string> undoStack;
    stack<string> redoStack;
public:
    void undo(const string& word);
    void redo();
    void displayUndoStack();
    void displayRedoStack();
};
```

- • Kelas `UndoRedo` memiliki dua tumpukan (`stack`) sebagai anggota pribadi: untuk menyimpan tindakan yang dapat dibatalkan, dan untuk menyimpan tindakan yang dapat diulangi.
- Terdapat empat metode publik: `undo`, `redo`, `displayUndoStack`, dan `displayRedoStack`, yang masing-masing bertanggung jawab atas operasi undo, redo, serta menampilkan isi tumpukan undo dan redo.

```

void UndoRedo::undo(const string& word) {
    undoStack.push(word);
}

void UndoRedo::redo() {
    if (undoStack.empty()) {
        cout << "Data di hapus: --" << endl;
    } else {
        string word = undoStack.top();
        redoStack.push(word);
        undoStack.pop();
        cout << "Data di hapus: " << word << endl;
    }
}

void UndoRedo::displayUndoStack() {
    // Implementasi...
}

void UndoRedo::displayRedoStack() {
    // Implementasi...
}

```

- Metode `undo` digunakan untuk menambahkan kata ke dalam tumpukan `undoStack`.
- Metode `redo` memindahkan kata dari tumpukan `undoStack` ke `redoStack`, dan mencetak kata yang dihapus dari `undoStack`.
- Metode `displayUndoStack` dan `displayRedoStack` bertanggung jawab atas menampilkan isi tumpukan `undo` dan `redo`.

-

```

int main() {
    UndoRedo ur;
    int choice;
    string word;

    while (true) {
        // Implementasi...
    }

    return 0;
}

```

- Fungsi `main` inialisasi objek `UndoRedo` dan mengatur loop tak terbatas untuk meminta input dari pengguna dan memanggil metode yang sesuai berdasarkan pilihannya

```

while (true) {
    cout << "1. undo" << endl;
    cout << "2. redo (hapus)" << endl;
    cout << "3. isEmptyUndo (tampilkan data)" << endl;
    cout << "4. isEmptyRedo (data yang di hapus)" << endl;
    cout << "Masukan pilihan: ";
    cin >> choice;

    switch (choice) {
        case 1:
            cout << "Masukan kata: ";
            cin >> word;
            ur.undo(word);
            break;

```

```

        case 2:
            ur.redo();
            break;
        case 3:
            ur.displayUndoStack();
            break;
        case 4:
            ur.displayRedoStack();
            break;
        default:
            cout << "Pilihan tidak valid" << endl;
    }
}

```

- Loop ini akan terus berjalan hingga program dihentikan.
- Pengguna diminta untuk memasukkan pilihan mereka (1 untuk undo, 2 untuk redo, dst.).
- Metode yang sesuai kemudian dipanggil berdasarkan pilihan pengguna menggunakan pernyataan `switch`.

