Nomor Kelompok : 32

Anggota : Luqman A. Siswanto (13513024)

Muhammad Aodyra K. (13513063)

Wiwit Rifa’i (13513073)

Ahmad Darmawan (13513096)

Mentor : Sonny Lazuardi Hermawan (13511029)

# Konfirmasi Spesifikasi

Mahasiswa harus membuat spesifikasi yang jelas dari deskripsi TB yang diberikan.

Pada bagian ini diberikan sebagian di antaranya

|  |  |  |
| --- | --- | --- |
| **No** | **Item Yang diperjelas** | **Spesifikasi (detailnya boleh sebagai lampiran)** |
|  | String | Memakai STL C++ |
|  | Command | - Terdiri dari Set, Redo, Undo, Save, Show All Memory, Show <n>  - Masukkan selain itu diberikan pesan kesalahan (sudah diantisipasi).  - Case Sensitive |
|  | Redo | - Bila Redo <n> dengan n lebih dari jumlah operasi yang telah di Undo maka seluruh operasi yang di Undo untuk terakhir kalinya di-Redo.  - Redo tidak termasuk operasi yang di-Undo/Redo.  - Redo hanya bisa dilakukan pada operasi yang di-Undo sebelum melakukan operasi lain, Sehingga jika sudah dilakukan operasi lain maka Undo yang sebelumnya tidak bisa di-Redo kembali. |
|  | Undo | - Bila Undo <n> dengan n lebih dari jumlah operasi yang sekarang telah dilakukan, maka seluruh operasi di-Undo  - Undo tidak termasuk operasi yang di-Undo/Redo |
|  | Save | - Save tidak termasuk operasi yang di-Undo/Redo |
|  | Angka Romawi | - Dalam operasinya, angka tidak melebihi 3000 dan selalu positif  - Penulisan bilangan romawi dalam upper-case  - Masukan romawi selalu benar dan bisa ada ekuivalennya dalam bilangan arab |
|  | Ekspresi | - Sintaks (aturan penulisan) setiap jenis ekspresi, prefix, infix, postfix  - Penggunaan tanda dalam kurung hanya ada dalam setting ekspresi infiks.  - Ekspresi masukan pengguna diasumsikan selalu benar |
|  | Bilangan | - Tidak ada operasi yang menghasilkan bilangan tidak terdefinisi (contoh : pembagian dengan 0)  - Masukkan dan hasil harus berupa integer yang mampu ditampung dalam signed integer 32 bit. |
|  | Operator | - Operator bagi dilambangkan dengan( “/ “).  - Operator modulo dilambangkan dengan(“%”)  - Operator kali dilambangkan dengan (“\*”)  - Operator tambah dilambangkan dengan (“+”)  - Operator kurang dilambangkan dengan (“-”)  - Operator and dilambangkan dengan (“&”)  - Operator or dilambangkan dengan (“|”)  - Operator xor dilambangkan dengan (“^”)  - Operator not dilambangkan dengan (“~”) |
| 10. | Setting Mode | - Setting mode tidak termasuk operasi yang di-Undo/Redo |
| 11. | Bilangan logik | - Representasi bilangan logik adalah angka (0, 1) bukan true false |

# Daftar Fungsi Pengguna yang diimplementasi

Tuliskan nama fungsi ditawarkan ke Pengguna dan parameternya (jika ada)

*Catatan : Yang dimaksud fungsi adalah “fungsi” yang ditawarkan ke pengguna (end user), bukan fungsi sebagai method.*

|  |  |  |
| --- | --- | --- |
| **No** | **Fungsi &param** | **Deskripsi** |
|  | Save | Menyimpan hasil operasi yang telah dimasukkan oleh user ke dalam file kesternal |
|  | Redo <n> | Mengulang n perintah terkahir |
|  | Undo <n> | Menghapus n buah perintah yang disimpan, dan menampilkan pesan penghapusan berhasil |
|  | Set <Mode Ekspresi> | Menentukan jenis ekspresi (infix, prefix, postfix) yang ingin digunakan dalam masukkan |
|  | Set <Equation> (bilangan / logika) | Menentukan pola persamaan yang ingin digunakan dalam perhitungan |
|  | Set <Jenis Bilangan> | Menentukan jenis bilangan (romawi/bilangan arab) yang ingin digunakkan dalam masukkan |
|  | Reset | Mengembalikan setting kembali ke default setting |
|  | Show Mem <n> | Menampilkan ekspresi atau perintah sebanyan n terakhir yang disimpan dalam memori |
|  | Show Mem All | Menampilkan semua ekspresi atau perintah semua yang tersimpan di memori |
| 10. | Help | Menjelaskan semua fungsi yang ditawarkan ke pengguna, beserta parameternya |
| 11 | View | Mengeluarkan setting sekarang |
| 12 | Exit | Keluar dari command prompt |

# Daftar Kelas yang akan diimplementasi

Jenis kelas : ADT/Mesin/Proses

## Nama Package : Command

Isi kelas dalam Package :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas (Unit)** | **Jenis \*)** | **Deskripsi** | **Method**  **(signature saja)** |
|  | Manager | Proses | Mengelola kelas kelas yang lain, menghidupkan kelas lain ketika dibutuhkan, menyimpan setting mode dalam data member | void SetMode(int);  int GetMode(); |
|  | Reader | Mesin | Membaca ekspresi / perintah dalam command | void ReadCommand();  string GetEquation(char\*); |
|  | Logger | Mesin | Mengurusi operasi redo / undo dan menyimpan ekspresi / command yang sudah dimasukkan user | void Redo()  void Undo()  void ShowMemAll() |
|  | Saver | Mesin | Menyimpan ekspresi/perintah yang telah dikerjakan ke dalam file eksternal | void SaveMem() |
|  |  |  |  |  |

## Nama Package : Equation

Isi kelas dalam Package :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas (Unit)** | **Jenis \*)** | **Deskripsi** | **Method**  **(signature saja)** |
| 1. | Equation | Proses | Menyelesaikan persamaan perhitungan yang dimasukkan pengguna, mengganti proses | Equation()  int GetHasil()  void SolveMathematical()  void SolveLogical() |
| 2. | Equation Exception | Proses | Memberikan keluran kepada user, jika ada masukan yang salah | EquationException()  void DisplayError() |
|  |  |  |  |  |

## Nama Package : Token

Isi kelas dalam Package :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas (Unit)** | **Jenis \*)** | **Deskripsi** | **Method**  **(signature saja)** |
| 1. | Token | ADT | Membaca setiap persamaan yang dimasukkan pengguna, dan metranslasikan kedalam bilangan atau operator | char GetToken()  void SetToken() |
| 2. | Logics | ADT | Merupakan class dalam operasi lojik, merupakan derived dari class Token | bool operator|  bool operator&  bool operator~  bool opertator^ |
| 3. | Number | ADT | Merupakan abstract class yang berisi pure virtual void/function yang belum diimplementasi, merupakan derived dari class Token | Number& operator\*  Number& operator+  Number& operator-  Number& operator/  bool operator<  int toInt |
| 4. | NumberRomawi | ADT | Merupakan inherited dari class Number |  |
| 5. | NumberArab | ADT | Merupakan inherited dari class Number |  |

## Nama Package : vector

Isi kelas dalam Package :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas (Unit)** | **Jenis \*)** | **Deskripsi** | **Method**  **(signature saja)** |
| 1. | vector<template T> | ADT | Tipe abstrak data berupa larik yang tiap elemennya dapat diakses secara langsung. | void push\_back(template T)  void pop\_back()  T back()  T& operator[]  int size()  bool empty() |
|  |  |  |  |  |
|  |  |  |  |  |

## Nama Package : stack

Isi kelas dalam Package :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas (Unit)** | **Jenis \*)** | **Deskripsi** | **Method**  **(signature saja)** |
| 1. | stack<template T> | ADT | Tipe abstrak data berupa larik dengan pengaksesan datanya hanya bisa pada elemen terakhir. | void push(template T)  void pop()  T top()  bool empty()  int size() |
|  |  |  |  |  |
|  |  |  |  |  |

**4 Lampiran tambahan keterangan**

1. Batasan kesalahan yang ditangani oleh EquationException adalah
2. Hasil romawi yang di luar batas [1..3999]
3. Pembagian dengan 0
4. Untuk membedakan token dalam equation, setiap token harus dipisahkan dengan spasi

Contoh : Pada pola infix penulisan 3+4-2(1\*8) harus dibuat menjadi 3 + 4 - 2 ( 1 \* 8 )

1. Stack dan Vektor dibuat generik karena akan diimplementasikan untuk tipe string, token, dan bilangan.
2. Mode default ketika awal membuka program adalah
3. Bilangan : Bilangan Arab
4. Ekspresi : Infix
5. Operator : Aritmatika ( +, -, \*, /)