

DOKUMENTASI PROGRAM KALKULATOR SAKTI

TUGAS BESAR I

[IF2210] Pemrograman Berorientasi Objek

Asisten

Sonny Lazuardi H (13511029)

Kelompok /* */

Luqman Arifin Siswanto (13513024)


Muhammad Aodyra Khaidir (13513063)

Wiwit Rifai (13513073)

Ahmad Darmawan (13513096)

Version 1.0

19 Maret 2015

| | | | | |
|---|---|---------------|---|---------------|
|  | Sekolah Teknik Elektro dan Informatika ITB | Nomor Dokumen | | Halaman |
| | | IF2210-TB-01 | | 95 |
| | | Revisi | - | 19 Maret 2015 |

Hierarchical Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

| | |
|-------------------------|----|
| ClassController | 4 |
| EquationException | 16 |
| Expression | 19 |
| Equation | 10 |
| Extension | 22 |
| Log | 24 |
| Logger | 27 |
| Reader..... | 61 |
| Saver | 64 |
| stack< T > | 68 |
| stack< Token * > | |
| Token..... | 72 |
| Logic | 33 |
| Number..... | 40 |
| NumberArab..... | 48 |
| NumberRomawi | 55 |
| vector< T > | 77 |
| vector< Log > | |

File Index

File List

Here is a list of all documented files with brief descriptions:

| | |
|------------------------------------|------------------------------|
| Command/ClassController.h | Error! Bookmark not defined. |
| Command/Log.h | Error! Bookmark not defined. |
| Command/Logger.h | Error! Bookmark not defined. |
| Command/Reader.h | Error! Bookmark not defined. |
| Command/Saver.h | 82 |
| Equation/Equation.cpp | 83 |
| Equation/Equation.h | 84 |
| Equation/EquationException.h | 85 |
| Equation/Expression.cpp | 86 |
| Equation/Expression.h | 87 |
| Extension/Extension.h | Error! Bookmark not defined. |
| Stack/stack.h | 88 |
| Token/Logic.cpp | 89 |
| Token/Logic.h | 90 |
| Token/Number.h | 91 |
| Token/NumberArab.h | 93 |
| Token/NumberRomawi.h | 94 |
| Token/Token.h | 95 |
| Vector/vector.h | Error! Bookmark not defined. |

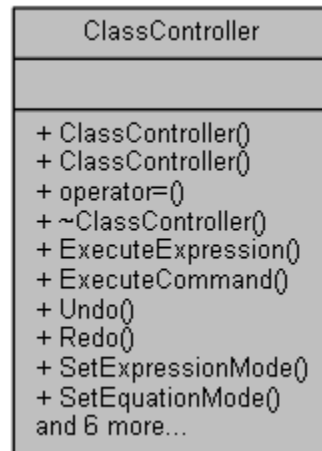
Class Documentation

ClassController Class Reference

Kelas **ClassController** bertugas untuk mengatur kehidupan dan kematian kelas-kelas lain.

#include <ClassController.h>

Collaboration diagram for ClassController:



Public Member Functions

- **ClassController ()**
*Konstruktor kelas **ClassController**.*
- **ClassController (const ClassController &)**
*Copy Constructor kelas **ClassController** dengan parameter.*
- **ClassController & operator= (const ClassController &)**
*Operator assignment kelas **ClassController**.*
- **~ClassController ()**
*Destruktor kelas **ClassController**.*
- void **ExecuteExpression** (string &)
Mengeksekusi ekspresi yang dimasukkan oleh user.
- void **ExecuteCommand** (string)
Mengeksekusi command yang dimasukkan oleh user.
- void **Undo** (int)
Mengembalikan perintah yang telah dimasukkan user.
- void **Redo** (int)
Mengembalikan perintah yang telah di-undo user.
- void **SetExpressionMode** (int)
Mengubah mode ekspresi (prefiks - infiks - postfix)
- void **SetEquationMode** (int)
Mengubah mode equation (bilangan - logika)
- void **SetNumberMode** (int)

Mengubah mode number (romawi - arab)

- void **ResetSetting** ()
Mengembalikan setting ke mode default.
- void **ShowMem** (int)
Menampilkan n perintah terakhir yang dimasukkan user.
- void **ShowMemAll** ()
Menampilkan semua perintah yang dimasukkan user.
- void **Help** ()
Menampilkan 'help' yang berisi daftar command yang berlaku dalam program.
- void **ViewSetting** ()
Menampilkan mode setting terkini.

Detailed Description

Kelas **ClassController** bertugas untuk mengatur kehidupan dan kematian kelas-kelas lain.

Author:

Luqman A. Siswanto (13513024)

Version:

1.0

Description

Constructor & Destructor Documentation

ClassController::ClassController (const **ClassController** & *man*)

Copy Constructor kelas **ClassController** dengan parameter.

Parameters:

| | |
|------------------------|--------------------|
| <i>ClassController</i> | yang akan di-copy. |
|------------------------|--------------------|

Member Function Documentation

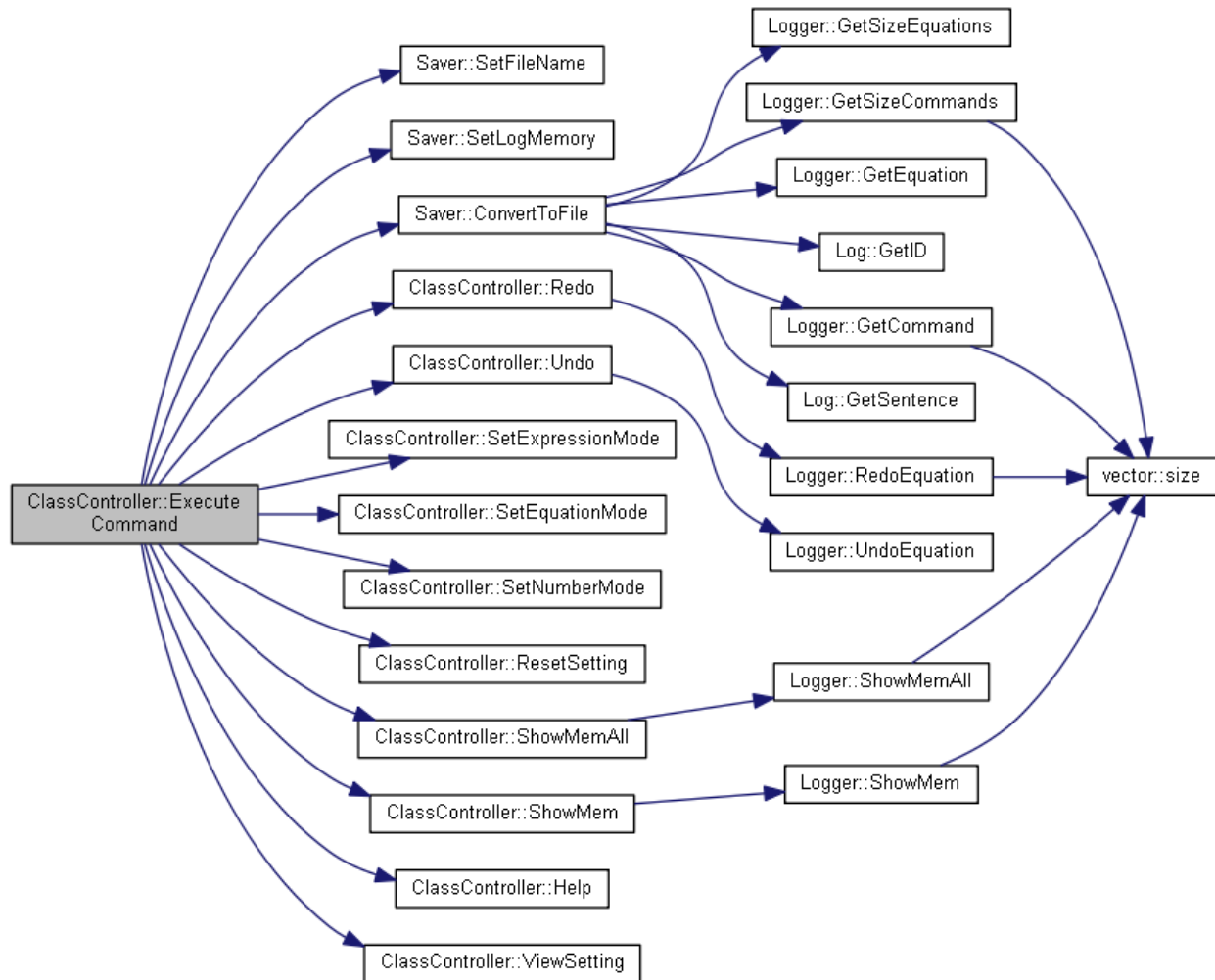
void ClassController::ExecuteCommand (string *buffer*)

Mengeksekusi command yang dimasukkan oleh user.

Parameters:

| | |
|---------------|--|
| <i>string</i> | - masukan string command yang akan dieksekusi. |
|---------------|--|

Here is the call graph for this function:



Here is the caller graph for this function:



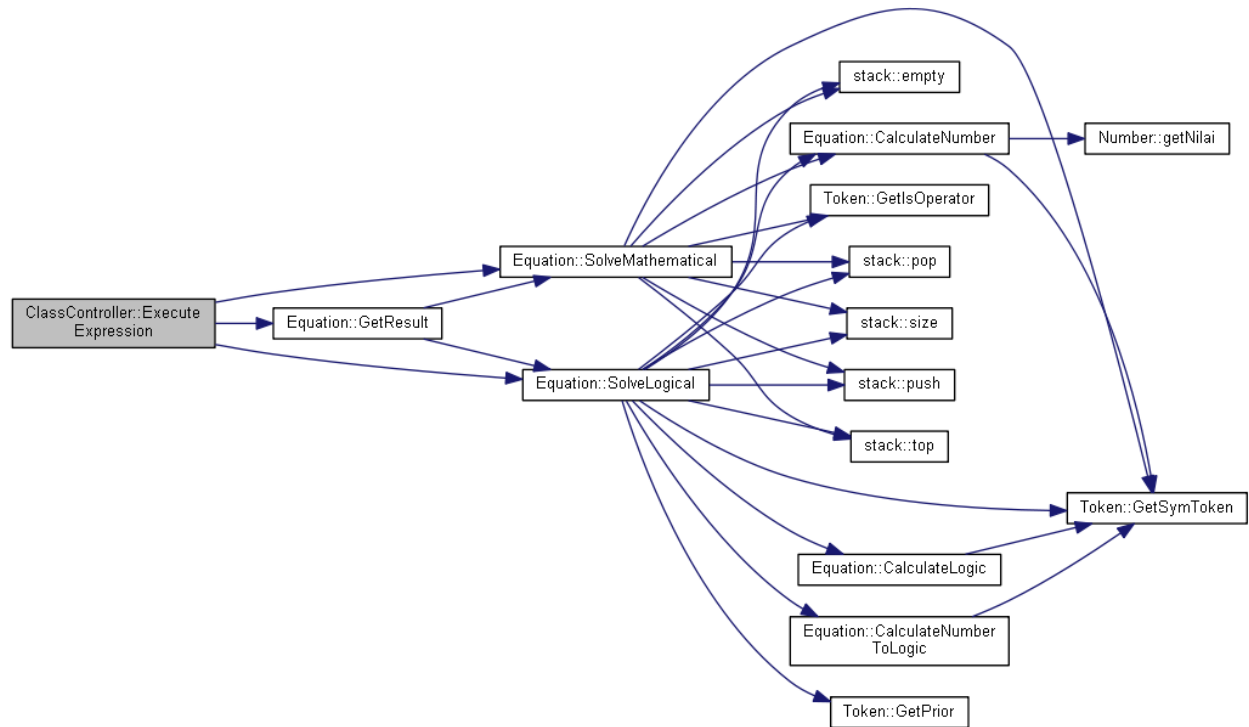
void ClassController::ExecuteExpression (string & *buffer*)

Mengeksekusi ekspresi yang dimasukkan oleh user.

Parameters:

| | |
|---------------|---|
| <i>string</i> | - masukan string ekspresi yang akan dieksekusi. |
|---------------|---|

Here is the call graph for this function:



Here is the caller graph for this function:



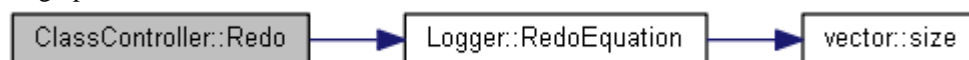
void ClassController::Redo (int n)

Mengembalikan perintah yang telah di-undo user.

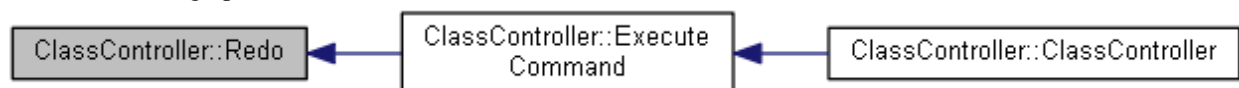
Parameters:

| | |
|-----|--------------------|
| int | - banyak perintah. |
|-----|--------------------|

Here is the call graph for this function:



Here is the caller graph for this function:



void ClassController::SetEquationMode (int equationMode)

Mengubah mode equation (bilangan - logika)

Parameters:

| | |
|------------|---|
| <i>int</i> | - nomor setting (ada di kelas Extension). |
|------------|---|

Here is the caller graph for this function:



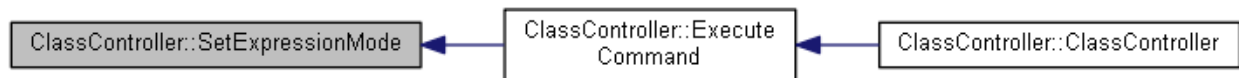
void ClassController::SetExpressionMode (int *expressionMode*)

Mengubah mode ekspresi (prefiks - infiks - postfiks)

Parameters:

| | |
|------------|---|
| <i>int</i> | - nomor setting (ada di kelas Extension). |
|------------|---|

Here is the caller graph for this function:



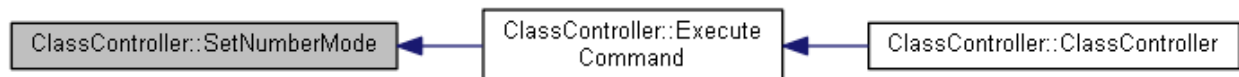
void ClassController::SetNumberMode (int *numberMode*)

Mengubah mode number (romawi - arab)

Parameters:

| | |
|------------|---|
| <i>int</i> | - nomor setting (ada di kelas Extension). |
|------------|---|

Here is the caller graph for this function:



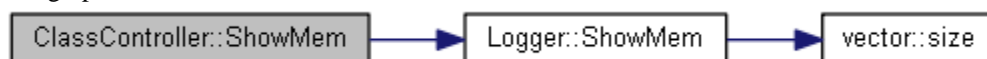
void ClassController::ShowMem (int *n*)

Menampilkan n perintah terakhir yang dimasukkan user.

Parameters:

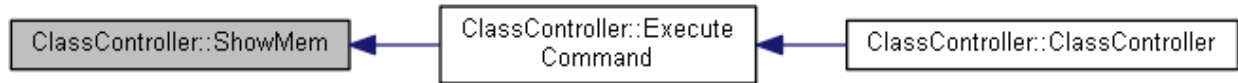
| | |
|----------|-------------------------|
| <i>n</i> | - int. Banyak perintah. |
|----------|-------------------------|

Here is the call graph for this function:



Here is the caller graph for this function:

| | | |
|---|--------------|---------------------------|
| STEI- ITB | IF2210-TB-01 | Halaman 8 dari 96 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |



void ClassController::Undo (int *n*)

Mengembalikan perintah yang telah dimasukkan user.

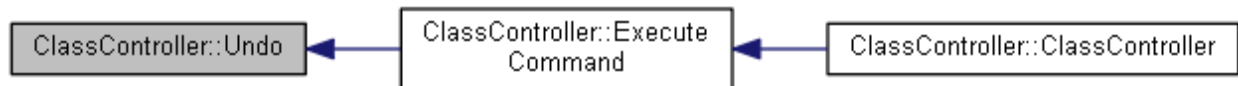
Parameters:

`int` - banyak perintah.

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

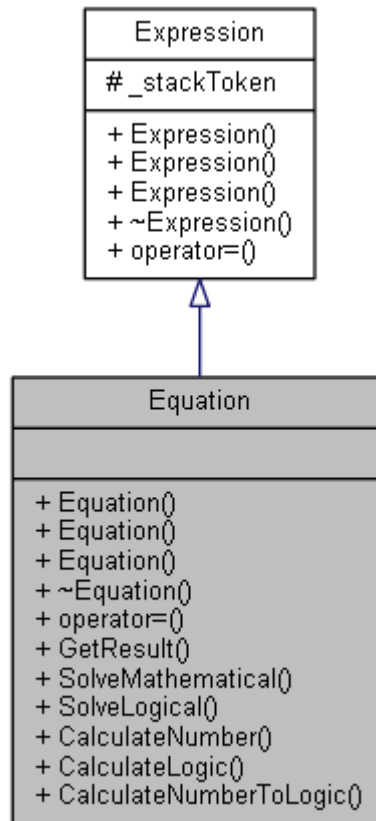
- Command/ClassController.h
- Command/ClassController.cpp

Equation Class Reference

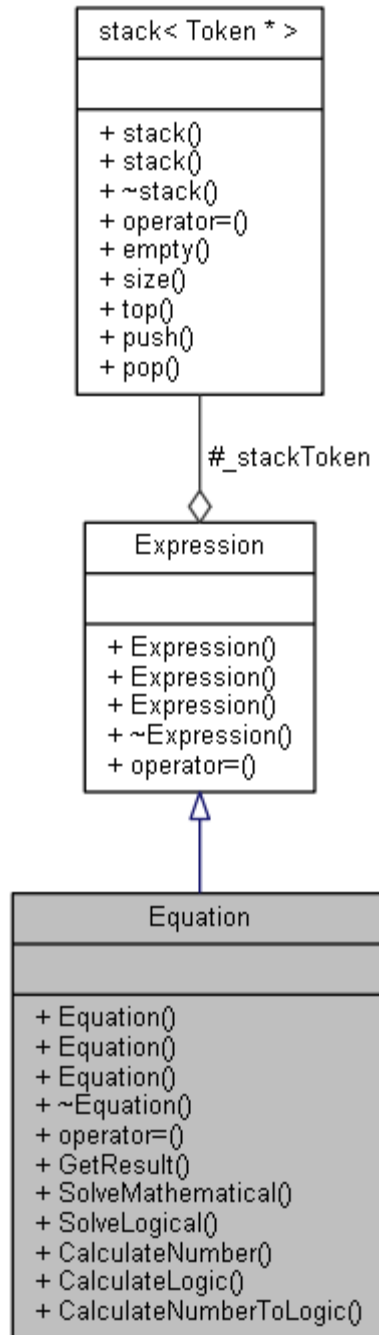
Kelas **Equation** bertanggung jawab dalam perhutingan hasil ekspresi matematika ataupun logika dalam bentuk prefix. Kelas **Equation** merupakan inheritent dari kelas **Expression**.

```
#include <Equation.h>
```

Inheritance diagram for Equation:



Collaboration diagram for Equation:



Public Member Functions

- **Equation ()**
*Konstruktor kelas **Equation**.*
- **Equation** (string strEqin, int modeExpress, int modeEquation, int modeNumber)
*Konstruktor kelas **Equation** dengan parameter.*
- **Equation** (const **Equation** &)
*Copy Constructor kelas **Equation**.*

- virtual **~Equation** ()
*Destructor kelas **Equation**.*
- **Equation & operator=** (const **Equation** &)
*operator assignment kelas **Equation***
- string **GetResult** ()
Getter hasil perhitungan dari ekspresi.
- void **SolveMathematical** ()
*Menghitung hasil ekspresi matematika yang ada dalam stack<token * > _stackToken.*
- void **SolveLogical** ()
*Menghitung hasil ekspresi logika yang ada dalam stack<token * > _stackToken.*
- **Number * CalculateNumber** (**Number** *opn1, **Token** *opr, **Number** *opn2)
Menghitung operasi bilangan yaitu untuk operator (+), (-), (), dan (/).*
- **Logic * CalculateLogic** (**Logic** *opn1, **Token** *opr, **Logic** *opn2)
Menghitung operasi logika yaitu untuk operator (&), (/), dan (^).
- **Logic * CalculateNumberToLogic** (**Number** *opn1, **Token** *opr, **Number** *opn2)
Menghitung operasi relasional pada bilangan yaitu untuk operator (<), (<=), (>), (>=), (=) dan (!=).

Additional Inherited Members

Detailed Description

Kelas **Equation** bertanggung jawab dalam perhutingan hasil ekspresi matematika ataupun logika dalam bentuk prefix. Kelas **Equation** merupakan inheritent dari kelas **Expression**.

Author:

Wiwit Rifa'i (13513073)

Version:

1.0

Description

Constructor & Destructor Documentation

Equation::Equation (string *strEqin*, int *modeExpress*, int *modeEquation*, int *modeNumber*)

Konstruktor kelas **Equation** dengan parameter.

Parameters:

| | |
|------------|---|
| <i>int</i> | strEqin - String yang merepresentasikan ekspresi matematika atau logika yang akan dikelola. |
| <i>int</i> | modeExpress - kode integer yang menentukan apakah ekspresi dalam string tersebut dalam bentuk prefix, infix, ataupun postfix. |

| | | |
|---|---------------------|-----------------------------------|
| STEI- ITB | IF2210-TB-01 | Halaman 12 dari 96 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

| | |
|------------|---|
| <i>int</i> | modeEquation - kode integer yang menentukan jenis operasi yang digunakan yaitu operais pada bilangan ataukah logika. |
| <i>int</i> | modeNumber - kode integer yang menentukan jenis operan bilangan yang digunakan yaitu bilangan arab ataupun bilangan romawi. |

Member Function Documentation

Logic * Equation::CalculateLogic (Logic * *opn1*, Token * *opr*, Logic * *opn2*)

Menghitung operasi logika yaitu untuk operator (&), (|), dan (^).

Parameters:

| | |
|--------------|---|
| Logic | * opn1 - pointer pada operan logika pertama |
| Token | * opr - pointer pada operator logika |
| Logic | * opn2 - pointer pada operan logika kedua |

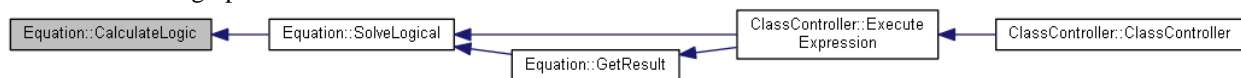
Returns:

Logic * ret - pointer pada hasil logika dari operasi

Here is the call graph for this function:



Here is the caller graph for this function:



Number * Equation::CalculateNumber (Number * *opn1*, Token * *opr*, Number * *opn2*)

Menghitung operasi bilangan yaitu untuk operator (+), (-), (*), dan (/).

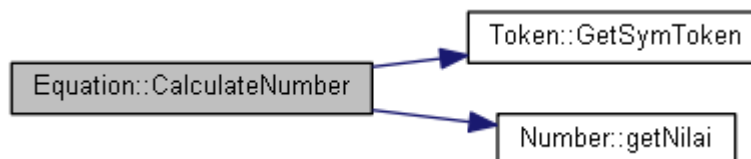
Parameters:

| | |
|---------------|---|
| Number | * opn1 - pointer pada operan bilangan pertama |
| Token | * opr - pointer pada operator bilangan |
| Number | * opn2 - pointer pada operan bilangan kedua |

Returns:

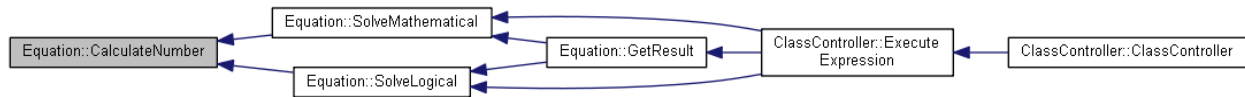
Number * ret - pointer pada bilangan hasil dari operasi

Here is the call graph for this function:



Here is the caller graph for this function:

| | | |
|---|---------------------|-----------------------------------|
| STEI- ITB | IF2210-TB-01 | Halaman 13 dari 96 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |



Logic * Equation::CalculateNumberToLogic (Number * *opn1*, Token * *opr*, Number * *opn2*)

Menghitung operasi relasional pada bilangan yaitu untuk operator (<), (<=), (>), (>=), (=) dan (!=).

Parameters:

| | |
|---------------|---|
| Number | * opn1 - pointer pada operan bilangan pertama |
| Token | * opr - pointer pada operator relasional |
| Number | * opn2 - pointer pada operan bilangan kedua |

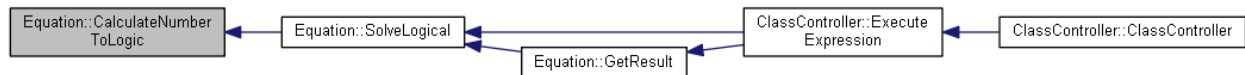
Returns:

Logic * ret - pointer pada hasil logika dari operasi

Here is the call graph for this function:



Here is the caller graph for this function:



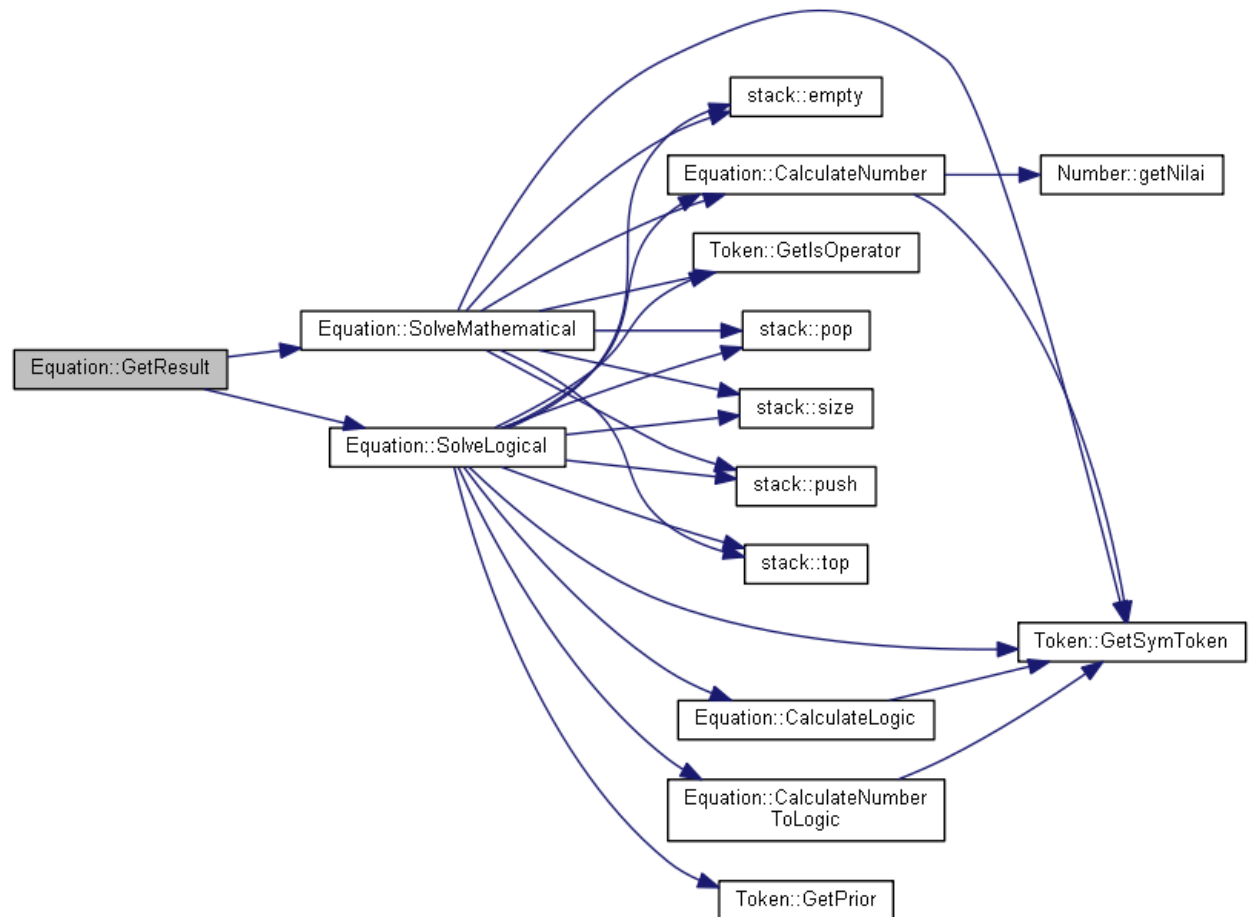
string Equation::GetResult ()

Getter hasil perhitungan dari ekspresi.

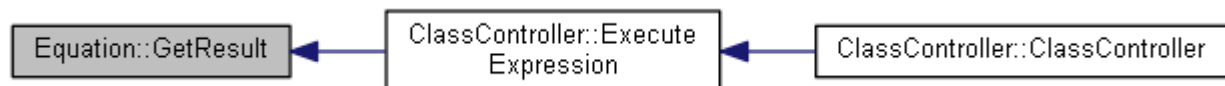
Returns:

string _result - string yang merepresentasikan hasil dari perhitungan

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

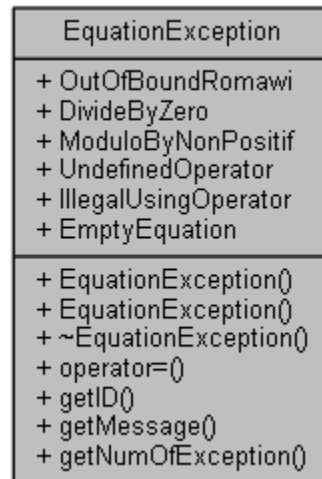
- Equation/Equation.h
- Equation/Equation.cpp

EquationException Class Reference

kelas **EquationException** merupakan kelas mengurus exception pada equation.

#include <EquationException.h>

Collaboration diagram for EquationException:



Public Member Functions

- **EquationException** (int)
*Konstruktor kelas **EquationException**.*
- **EquationException** (const **EquationException** &)
*Copy Constructor kelas **EquationException**.*
- **~EquationException** ()
*Destruktor kelas **EquationException**.*
- **EquationException** & **operator=** (const **EquationException** &)
*Operator assignment kelas **EquationException**.*
- const int **getID** ()
Getter nilai ID kesalahan.
- string **getMessage** ()
Getter pesan kesalahan.

Static Public Member Functions

- static int **getNumOfException** ()
Getter banyaknya exception yang telah terjadi.

Static Public Attributes

- static const int **OutOfBoundRomawi** = 0
konstanta yang menandakan nilai romawi diluar batas [1..3999]
- static const int **DivideByZero** = 1
konstanta yang menandakan pembagian dengan nol.

- static const int **ModuloByNonPositif** = 2
konstanta yang menandakan modulo dengan bilangan yang tak positif.
- static const int **UndefinedOperator** = 3
konstanta yang menandakan operator yang tidak diperbolehkan.
- static const int **IllegalUsingOperator** = 4
konstanta yang menandakan penggunaan operator yang tidak sesuai.
- static const int **EmptyEquation** = 5
*konstanta yang menandakan program mencoba menyelesaikan **Equation** kosong.*

Detailed Description

kelas **EquationException** merupakan kelas mengurus exception pada equation.

Author:

Wiwit Rifa'i (13513073)

Version:

1.0

Description

Constructor & Destructor Documentation

EquationException::EquationException (int *id*)

Konstruktor kelas **EquationException**.

Parameters:

| | |
|------------|--|
| <i>int</i> | - menyatakan jenis error yang terjadi. |
|------------|--|

EquationException::EquationException (const EquationException & *s*)

Copy Constructor kelas **EquationException**.

Parameters:

| | |
|--------------|---|
| <i>const</i> | EquationException & - reference objek yang akan di copy; |
|--------------|---|

Member Function Documentation

EquationException & EquationException::operator= (const EquationException &)

Operator assignment kelas **EquationException**.

Parameters:

| | |
|-------------------------------|--------------------------------------|
| <i>EquationException</i> & | - reference objek yang akan di-copy. |
|-------------------------------|--------------------------------------|

The documentation for this class was generated from the following files:

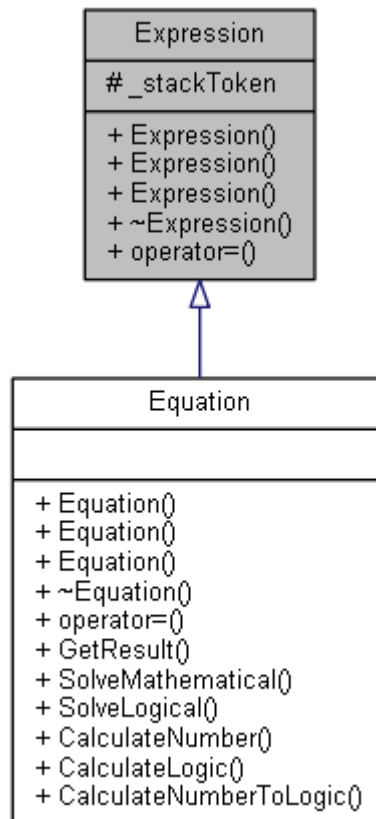
- Equation/**EquationException.h**
- Equation/EquationException.cpp

Expression Class Reference

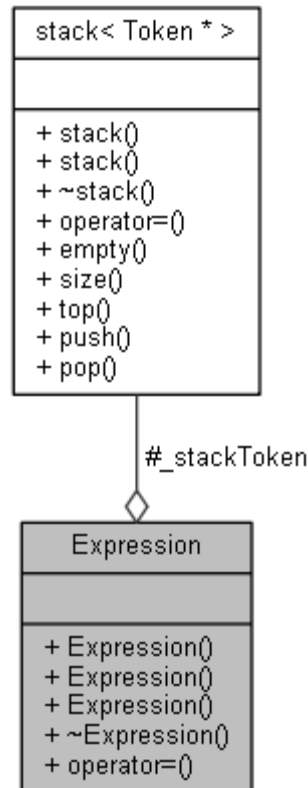
Kelas **Expression** bertanggung jawab dalam mengolah ekspresi matematika atau logika seperti mengubah string menjadi ekspresi matematika dan mengubah bentuk infix atau postfix menjadi prefix agar bisa dihitung di class **Equation**.

```
#include <Expression.h>
```

Inheritance diagram for Expression:



Collaboration diagram for Expression:



Public Member Functions

- **Expression ()**
*Konstruktur kelas **Expression**.*
- **Expression** (const string &strExp, int modeExpression, int modeEquation, int modeNumber)
*Konstruktur kelas **Expression** dengan parameter.*
- **Expression** (const **Expression** &)
*Copy Constructor kelas **Expression**.*
- virtual **~Expression ()**
*Destruktor kelas **Expression**.*
- **Expression & operator=** (const **Expression** &)
*Operator assignment kelas **Expression**.*

Protected Attributes

- **stack< Token * > _stackToken**

Detailed Description

Kelas **Expression** bertanggung jawab dalam mengolah ekspresi matematika atau logika seperti mengubah string menjadi ekspresi matematika dan mengubah bentuk infix atau postfix menjadi prefix agar bisa dihitung di class **Equation**.

Author:

Wiwit Rifa'i (13513073)

Version:

1.0

Description

Constructor & Destructor Documentation

Expression::Expression (const string & strExp, int modeExpression, int modeEquation, int modeNumber)

Konstruktor kelas **Expression** dengan parameter.

Parameters:

| | |
|---------------|--|
| <i>string</i> | strExp - String yang merepresentasikan ekspresi matematika atau logika yang akan dikelola. |
| <i>int</i> | modeExpression - kode integer yang menentukan apakah ekspresi dalam string tersebut dalam bentuk prefix, infix, ataupun postfix. |
| <i>int</i> | modeEquation - kode integer yang menentukan jenis operasi yang digunakan yaitu operasi pada bilangan ataukah logika. |
| <i>int</i> | modeNumber - kode integer yang menentukan jenis operan bilangan yang digunakan yaitu bilangan arab ataupun bilangan romawi. |

The documentation for this class was generated from the following files:

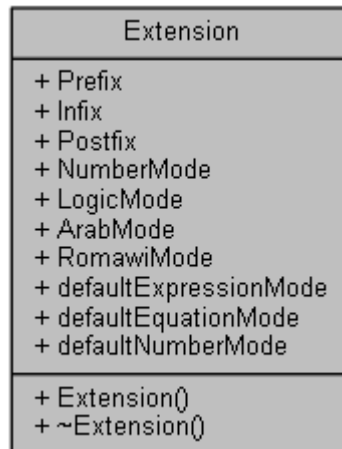
- Equation/Expression.h
- Equation/Expression.cpp

Extension Class Reference

Kelas **Extension** berisi konstanta yang dibutuhkan dalam program.

#include <Extension.h>

Collaboration diagram for Extension:



Public Member Functions

- **Extension ()**
*Konstruktor kelas **Extension**.*
- **~Extension ()**
*Desktruktor kelas **Extension**.*

Static Public Attributes

- static const int **Prefix** = 0
- static const int **Infix** = 1
- static const int **Postfix** = 2
- static const int **NumberMode** = 0
- static const int **LogicMode** = 1
- static const int **ArabMode** = 0
- static const int **RomawiMode** = 1
- static const int **defaultExpressionMode** = Infix
- static const int **defaultEquationMode** = NumberMode
- static const int **defaultNumberMode** = ArabMode

Detailed Description

Kelas **Extension** berisi konstanta yang dibutuhkan dalam program.

Author:

Luqman A. Siswanto (13513024)

Version:

1.0

Description

The documentation for this class was generated from the following files:

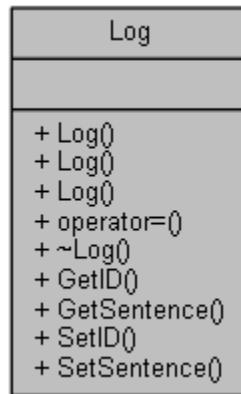
- Extension/Extension.h
- Extension/Extension.cpp

Log Class Reference

Kelas **Log** adalah abstract data type untuk log command.

```
#include <Log.h>
```

Collaboration diagram for Log:



Public Member Functions

- **Log ()**
*Konstruktur kelas **Log**.*
- **Log (int, string)**
*Konstruktur kelas **Log** dengan parameter.*
- **Log (const Log &)**
*Copy Constructor kelas **Log**.*
- **Log & operator= (const Log &)**
*Operator assignment kelas **Log**.*
- **~Log ()**
*Destruktor kelas **Log**.*
- **int GetID ()**
Getter untuk ID.
- **string GetSentence ()**
Getter untuk kalimat log.
- **void SetID (int)**
Setter untuk ID.
- **void SetSentence (string)**
Setter untuk sentence.

Detailed Description

Kelas **Log** adalah abstract data type untuk log command.

Author:

Luqman A. Siswanto (13513024)

Version:

1.0

Description

Constructor & Destructor Documentation

Log::Log (int *id*, string *sentence*)

Konstruktor kelas **Log** dengan parameter.

Parameters:

| | |
|---------------|---------------------------------------|
| <i>int</i> | - ID. |
| <i>string</i> | - command pada log yang akan disimpan |

Log::Log (const Log & *log*)

Copy Constructor kelas **Log**.

Parameters:

| | |
|------------|--------------------|
| <i>Log</i> | yang akan di-copy. |
|------------|--------------------|

Member Function Documentation

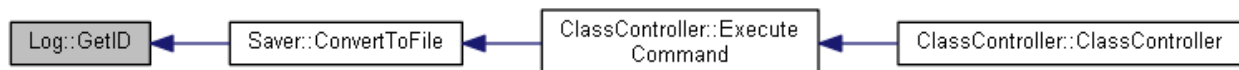
int Log::GetID ()

Getter untuk ID.

Returns:

int - ID pada log.

Here is the caller graph for this function:



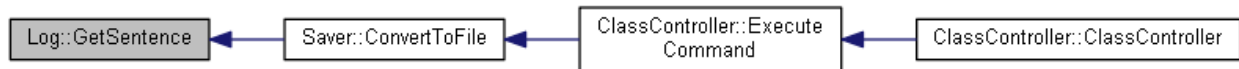
string Log::GetSentence ()

Getter untuk kalimat log.

Returns:

string - sentence.

Here is the caller graph for this function:

**Log & Log::operator= (const Log & log)**

Operator assignment kelas **Log**.

Parameters:

| | |
|------------|--------------------|
| <i>Log</i> | yang akan di-copy. |
|------------|--------------------|

void Log::SetID (int id)

Setter untuk ID.

Parameters:

| | |
|------------|-----------|
| <i>int</i> | - ID log. |
|------------|-----------|

void Log::SetSentence (string sentence)

Setter untuk sentence.

Parameters:

| | |
|---------------|-------------|
| <i>string</i> | - sentence. |
|---------------|-------------|

The documentation for this class was generated from the following files:

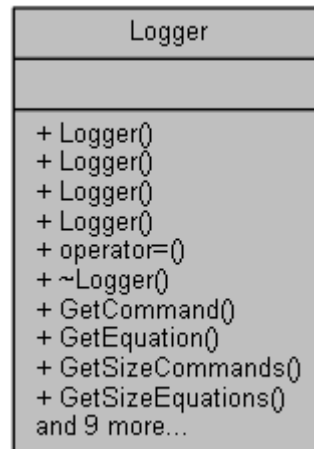
- Command/Log.h
- Command/Log.cpp

Logger Class Reference

Kelas **Logger** adalah abstract data type untuk log command.

```
#include <Logger.h>
```

Collaboration diagram for Logger:



Public Member Functions

- **Logger ()**
Konstruktur kelas logger.
- **Logger (vector< Log >, vector< Log >)**
Konstruktur kelas logger dengan parameter.
- **Logger (vector< Log >, vector< Log >, int)**
Konstruktur kelas logger dengan parameter.
- **Logger (const Logger &)**
Copy constructor kelas logger.
- **Logger & operator= (const Logger &)**
*Operator assignment kelas **Logger**.*
- **~Logger ()**
*Destruktor kelas **Logger**.*
- **Log GetCommand (int)**
*Getter **Log** Commands dengan indeks tertentu.*
- **Log GetEquation (int)**
*Getter **Log** Equations dengan indeks tertentu.*
- **int GetSizeCommands ()**
Getter ukuran commands yang valid.
- **int GetSizeEquations ()**
Getter ukuran equations yang valid.
- **void Clear ()**
Menghilangkan seluruh commands dan equations di memori.
- **void ClearCommands ()**

Menghilangkan commands di memori.

- void **ClearEquations** ()
Menghilangkan commands di equations.
- void **AddCommand (Log)**
Menambahkan log pada list of commands.
- void **AddEquation (Log)**
Menambahkan log pada list of equations.
- int **UndoEquation** (int)
Membatalkan equation yang telah ditambah.
- int **RedoEquation** (int)
Melakukan kembali equation yang telah di-undo.
- void **ShowMem** (int)
Menampilkan memori kembali ke layar.
- void **ShowMemAll** ()
Menampilkan semua memori kembali ke layar.

Detailed Description

Kelas **Logger** adalah abstract data type untuk log command.

Author:

Luqman A. Siswanto (13513024)

Version:

1.0

Description

Rule

Semua dalam vector commands dijamin valid (tidak ada operasi undo/redo) Seluruh isi vector equations belum tentu valid karena bisa jadi hasil undo

Constructor & Destructor Documentation

Logger::Logger (vector< Log > commands, vector< Log > equations)

Konstruktor kelas logger dengan parameter.

Parameters:

| | |
|--------------------------|-------------|
| vector<Log> | - commands |
| vector<Log> | - equations |

Here is the call graph for this function:

| | | |
|---|---------------------|-----------------------------------|
| STEI- ITB | IF2210-TB-01 | Halaman 28 dari 96 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |



Logger::Logger (vector< Log > *commands*, vector< Log > *equations*, int *sizeEquations*)

Konstruktor kelas logger dengan parameter.

Parameters:

| | |
|--------------------------|-------------------------------|
| <i>vector<Log></i> | - commands |
| <i>vector<Log></i> | - equations |
| <i>int</i> | - ukuran equations yang valid |

Logger::Logger (const Logger & *logger*)

Copy constructor kelas logger.

Parameters:

| | |
|---------------|----------------------------|
| <i>Logger</i> | - logger yang akan di-copy |
|---------------|----------------------------|

Member Function Documentation

void Logger::AddCommand (Log *log*)

Menambahkan log pada list of commands.

Parameters:

| | |
|------------|----------------------|
| <i>Log</i> | - yang akan ditambah |
|------------|----------------------|

Here is the call graph for this function:



Here is the caller graph for this function:



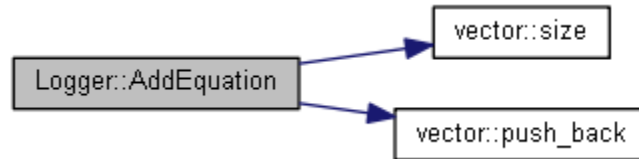
void Logger::AddEquation (Log *log*)

Menambahkan log pada list of equations.

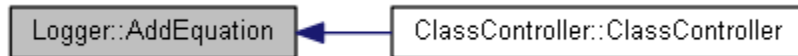
Parameters:

| | |
|------------|----------------------|
| <i>Log</i> | - yang akan ditambah |
|------------|----------------------|

Here is the call graph for this function:



Here is the caller graph for this function:

**Log Logger::GetCommand (int i)**

Getter **Log** Commands dengan indeks tertentu.

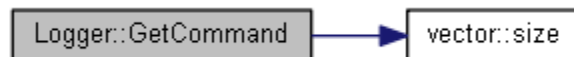
Parameters:

| | |
|------------|----------|
| <i>int</i> | - indeks |
|------------|----------|

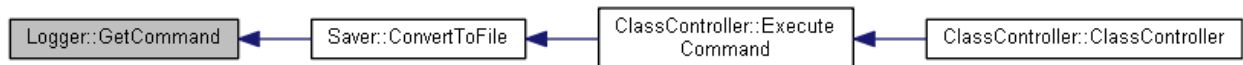
Returns:

Log - **Log** Command dengan indeks tertentu

Here is the call graph for this function:



Here is the caller graph for this function:

**Log Logger::GetEquation (int i)**

Getter **Log** Equations dengan indeks tertentu.

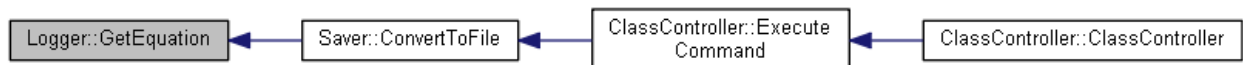
Parameters:

| | |
|------------|----------|
| <i>int</i> | - indeks |
|------------|----------|

Returns:

Log - **Log Equation** dengan indeks tertentu

Here is the caller graph for this function:



int Logger::GetSizeCommands ()

Getter ukuran commands yang valid.

Returns:

int - commands size

Here is the call graph for this function:



Here is the caller graph for this function:



int Logger::GetSizeEquations ()

Getter ukuran equations yang valid.

Returns:

int - equations size

Here is the caller graph for this function:



Logger & Logger::operator= (const Logger & logger)

Operator assignment kelas **Logger**.

Parameters:

| | |
|---------------|----------------------------|
| Logger | - logger yang akan di-copy |
|---------------|----------------------------|

int Logger::RedoEquation (int n)

Melakukan kembali equation yang telah di-undo.

Parameters:

| | |
|------------|---------------------------------|
| int | - berapa item redo yang diminta |
|------------|---------------------------------|

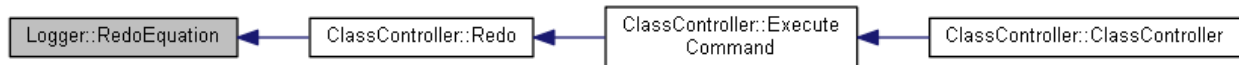
Returns:

int - berapa item redo yang berhasil dilakukan

Here is the call graph for this function:



Here is the caller graph for this function:



void Logger::ShowMem (int *n*)

Menampilkan memori kembali ke layar.

Parameters:

int - berapa item yang akan ditampilkan ke layar

Here is the call graph for this function:



Here is the caller graph for this function:



int Logger::UndoEquation (int *n*)

Membatalkan equation yang telah ditambah.

Parameters:

int - berapa item undo yang diminta

Returns:

int - berapa item undo yang berhasil dilakukan

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

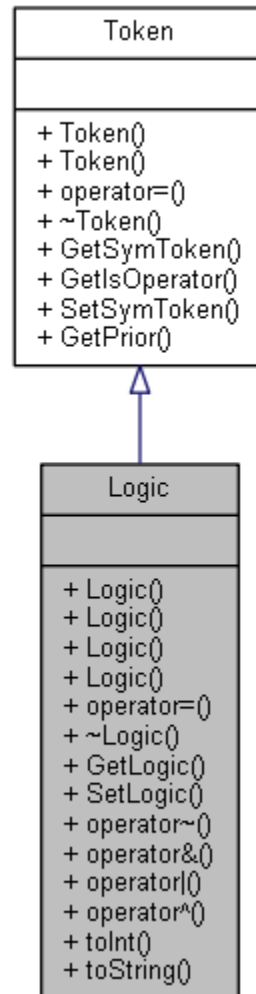
- Command/Logger.h
- Command/Logger.cpp

Logic Class Reference

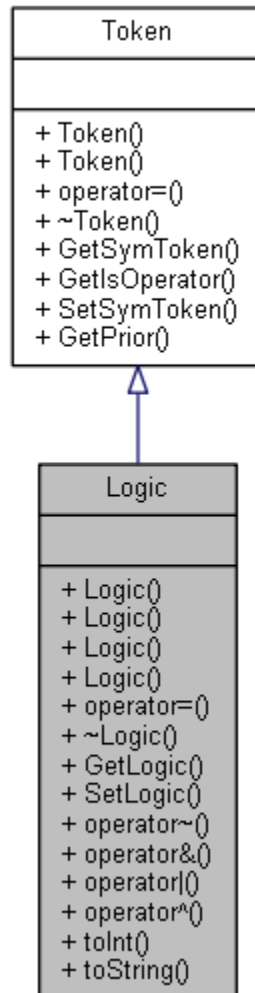
Kelas **Logic** mengatur perhitungan operasi logika (~, &, ^, |)

```
#include <Logic.h>
```

Inheritance diagram for Logic:



Collaboration diagram for Logic:



Public Member Functions

- **Logic ()**
*Konstruktor kelas **Logic**.*
- **Logic (string s)**
*Konstruktor kelas **Logic** dengan parameter string.*
- **Logic (int i)**
*Konstruktor kelas **Logic** dengan parameter integer.*
- **Logic (const Logic &L)**
*Copy Konstruktor kelas **Logic**.*
- **Logic & operator= (const Logic &L)**
*Assignment kelas **Logic**.*
- **~Logic ()**
*Destruktor kelas **Logic**.*
- **int GetLogic ()**
Getter untuk mendapatkan nilai logic.
- **void SetLogic (int i)**

Setter untuk mengubah nilai logika.

- **Logic & operator~ ()**
Melakukan proses "not" pada nilai logika ($\sim L$).
- **Logic & operator& (const Logic &L)**
Melakukan proses "and" pada nilai logika ($L1 \& L2$).
- **Logic & operator| (const Logic &L)**
Melakukan proses "or" pada nilai logika ($L1 | L2$).
- **Logic & operator^ (const Logic &L)**
Melakukan proses "xor" pada nilai logika ($L1 \wedge L2$).
- **int toInt (string)**
Mengubah nilai logika dalam bentuk string (TRUE atau FALSE) menjadi bentuk integer (1 & 0).
- **string toString (int)**
Mengubah nilai logika dalam bentuk bentuk integer (1 & 0) menjadi string (TRUE atau FALSE).

Detailed Description

Kelas **Logic** mengatur perhitungan operasi logika (\sim , $\&$, \wedge , $|$)

Author:

Ahmad Darmawan (13513096)

Version:

1.0

Constructor & Destructor Documentation

Logic::Logic (string s)

Konstruktor kelas **Logic** dengan parameter string.

Parameters:

| | |
|----------|---|
| <i>s</i> | - String berisi nilai logika (TRUE atau FALSE). |
|----------|---|

Here is the call graph for this function:



Logic::Logic (int i)

Konstruktor kelas **Logic** dengan parameter integer.

Parameters:

| | |
|----------|---|
| <i>i</i> | - Integer berisi nilai logika (0 atau 1). |
|----------|---|

Logic::Logic (const Logic & L)

Copy Konstruktor kelas **Logic**.

Parameters:

| | |
|----------|----------------------------------|
| <i>L</i> | - Variabel Logic masukan. |
|----------|----------------------------------|

Returns:

Li - Variabel **Logic** yang tercopy.

Member Function Documentation

int Logic::GetLogic ()

Getter untuk mendapatkan nilai logic.

Returns:

i - integer nilai logic (0 atau 1).

Logic & Logic::operator& (const Logic & L)

Melakukan proses "and" pada nilai logika (L1 & L2).

Parameters:

| | |
|-----------|-------------------------|
| <i>L1</i> | - Objek Logika pertama. |
| <i>L2</i> | - Objek Logika kedua. |

Returns:

L' = L1 & L2.

Here is the call graph for this function:



Logic & Logic::operator= (const Logic & L)

Assignment kelas **Logic**.

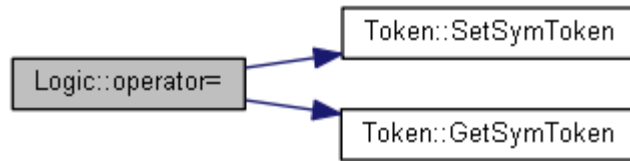
Parameters:

| | |
|----------|----------------------------------|
| <i>L</i> | - Variabel Logic masukan. |
|----------|----------------------------------|

Returns:

Li - Variabel **Logic** yang terassign.

Here is the call graph for this function:



Logic & Logic::operator^ (const Logic & L)

Melakukan proses "xor" pada nilai logika ($L1 \wedge L2$).

Parameters:

| | |
|-----------|-------------------------|
| <i>L1</i> | - Objek Logika pertama. |
| <i>L2</i> | - Objek Logika kedua. |

Returns:

$L' = L1 \wedge L2$.

Here is the call graph for this function:



Logic & Logic::operator| (const Logic & L)

Melakukan proses "or" pada nilai logika ($L1 \vee L2$).

Parameters:

| | |
|-----------|-------------------------|
| <i>L1</i> | - Objek Logika pertama. |
| <i>L2</i> | - Objek Logika kedua. |

Returns:

$L' = L1 \vee L2$.

Here is the call graph for this function:



Logic & Logic::operator~ ()

Melakukan proses "not" pada nilai logika ($\sim L$).

Parameters:

| | |
|----------|----------------------|
| <i>L</i> | - Objek Logika awal. |
|----------|----------------------|

Returns:

$\sim L$ - Objek Logika akhir.

Here is the call graph for this function:



void Logic::SetLogic (int i)

Setter untuk mengubah nilai logic.

[I.S] integer i terdefinisi (0 atau 1). [F.S] _logic diassign dengan nilai i.

Parameters:

| | |
|----------|-------------------------------------|
| <i>S</i> | - nilai logic yang akan dimasukkan. |
| <i>S</i> | - nilai logic yang akan dimasukkan. |

int Logic::toInt (string s)

Mengubah nilai logika dalam bentuk string (TRUE atau FALSE) menjadi bentuk integer (1 & 0).

Prekondisi : string terdefinisi (TRUE atau FALSE)

Parameters:

| | |
|----------|------------------------------|
| <i>S</i> | - nilai logika dalam string. |
|----------|------------------------------|

Returns:

i - nilai logika dalam integer.

Parameters:

| | |
|----------|------------------------------|
| <i>S</i> | - nilai logika dalam string. |
|----------|------------------------------|

Returns:

i - nilai logika dalam integer.

Here is the caller graph for this function:



string Logic::toString (int n)

Mengubah nilai logika dalam bentuk integer (1 & 0) menjadi string (TRUE atau FALSE).

Prekondisi : int terdefinisi (0 atau 1)

Parameters:

| | |
|----------|-------------------------------|
| <i>i</i> | - nilai logika dalam integer. |
|----------|-------------------------------|

Returns:

S - nilai logika dalam string.

Parameters:

| | |
|----------|-------------------------------|
| <i>i</i> | - nilai logika dalam integer. |
|----------|-------------------------------|

Returns:

S - nilai logika dalam string.

The documentation for this class was generated from the following files:

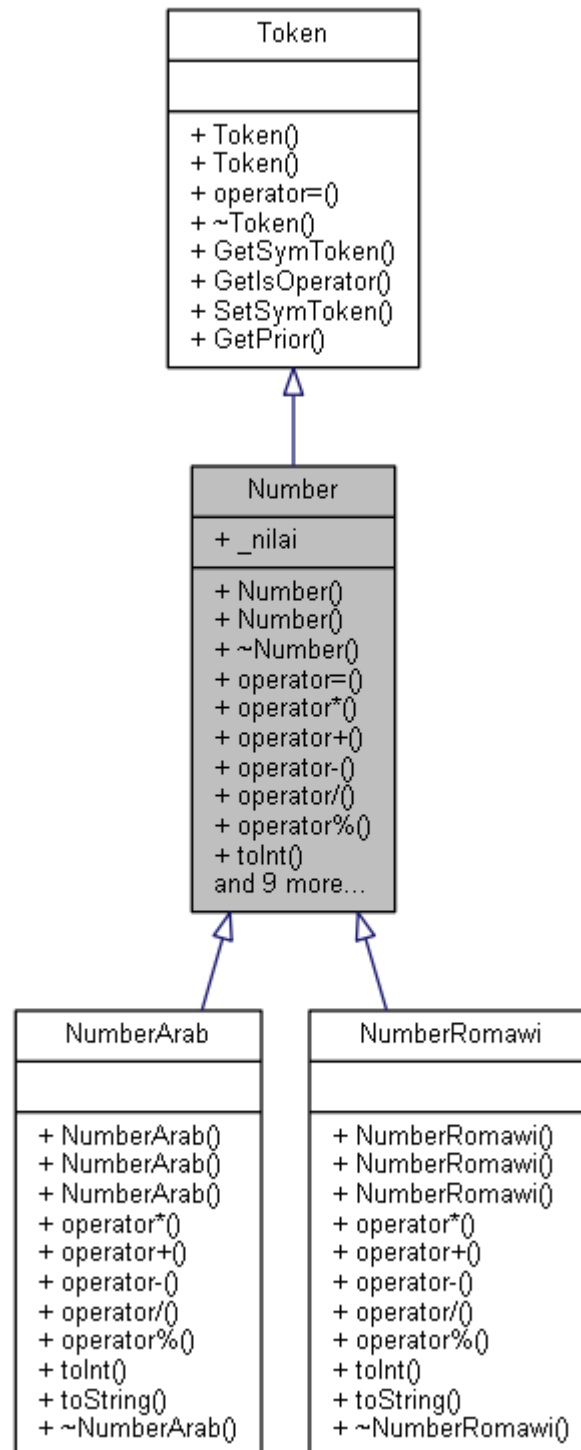
- Token/Logic.h
- Token/Logic.cpp

Number Class Reference

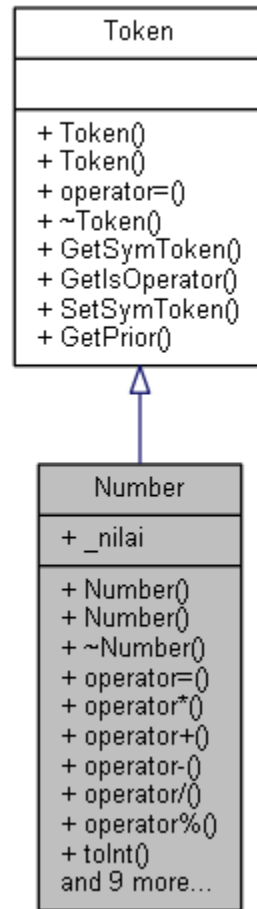
Kelas **Number** bertugas untuk mengelola operasi bilangan.

```
#include <Number.h>
```

Inheritance diagram for Number:



Collaboration diagram for Number:



Public Member Functions

- **Number ()**
*Konstruktor kelas **Number**.*
- **Number (string s)**
*Konstruktor kelas **Number** dengan parameter.*
- **virtual ~Number ()**
*Destruktor kelas **Number**.*
- **virtual Number & operator= (Number &n)**
*operator= untuk menyamakan nilai dari **Number** n.*
- **virtual Number & operator* (const Number &)=0**
operator untuk mengali nilai objek dengan nilai dari **Number** n.*
- **virtual Number & operator+ (const Number &)=0**
*operator+ untuk menambahkan nilai objek dengan nilai dari **Number** n.*
- **virtual Number & operator- (const Number &)=0**
*operator- untuk mengurangi nilai objek dengan nilai dari **Number** n.*
- **virtual Number & operator/ (const Number &)=0**

*operator/ untuk membagi nilai objek dengan nilai dari **Number** n.*

- virtual **Number** & **operator%** (const **Number** &)=0
*operator% untuk modulo nilai objek dengan nilai dari **Number** n.*
- virtual int **toInt** (string)=0
fungsi yang merubah string menjadi integer
- virtual string **toString** (int)=0
fungsi yang merubah integer menjadi string
- **Logic** & **operator<** (const **Number** &)
prosedur membandingkan nilai lebih kecil dengan nilai N
- **Logic** & **operator<=** (const **Number** &)
prosedur membandingkan nilai lebih kecil sama dengan nilai N.
- **Logic** & **operator>** (const **Number** &)
prosedur membandingkan nilai lebih dari dengan nilai N.
- **Logic** & **operator>=** (const **Number** &)
prosedur membandingkan nilai lebih dari sama dengan _nilai N.
- **Logic** & **operator==** (const **Number** &)
prosedur membandingkan _nilai sama dengan _nilai N.
- **Logic** & **operator!=** (const **Number** &)
prosedur membandingkan _nilai tidak sama dengan _nilai dari N.
- int **getNilai** ()
prosedur mendapatkan _nilai.
- void **setNilai** (int _n)
prosedur assign _nilai

Public Attributes

- int **_nilai**

Detailed Description

Kelas **Number** bertugas untuk mengelola operasi bilangan.

Author:

Muhammad Aodyra Khaidir (13513063)

Version:

1.0

Description

Constructor & Destructor Documentation

Number::Number (string s)

| STEI- ITB | IF2210-TB-01 | Halaman 42 dari 96 halaman |
|---|--------------|----------------------------|
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

Konstruktor kelas **Number** dengan parameter.

Parameters:

| | |
|----------|---|
| <i>s</i> | - String yang akan menjadi parameter Token . |
| <i>S</i> | - String yang akan menjadi nama file. |

Member Function Documentation

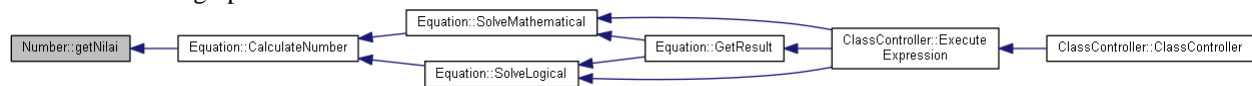
int Number::getNilai ()

prosedur mendapatkan *_nilai*.

Returns:

i - integer dari *_nilai*

Here is the caller graph for this function:



Logic & Number::operator!= (const Number & N)

prosedur membandingkan *_nilai* tidak sama dengan *_nilai* dari *N*.

Parameters:

| | |
|----------|--|
| <i>N</i> | - Number yang nilainya akan dibandingkan. |
|----------|--|

Returns:

$L' = L1 \neq L2$.

virtual Number& Number::operator% (const Number &) [pure virtual]

operator% untuk modulo nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya menjadi modulator. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implemented in **NumberRomawi** (p.58), and **NumberArab** (p.51).

virtual Number& Number::operator* (const Number &) [pure virtual]

operator* untuk mengali nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dikali. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implemented in **NumberRomawi** (p.58), and **NumberArab** (p.51).

virtual Number& Number::operator+ (const Number &) [pure virtual]

operator+ untuk menambahkan nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan ditambah. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implemented in **NumberRomawi** (p.59), and **NumberArab** (p.52).

virtual Number& Number::operator- (const Number &) [pure virtual]

operator- untuk mengurangi nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dikurang. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implemented in **NumberRomawi** (p.59), and **NumberArab** (p.52).

virtual Number& Number::operator/ (const Number &) [pure virtual]

operator/ untuk membagi nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dibagi. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implemented in **NumberRomawi** (p.59), and **NumberArab** (p.52).

Logic & Number::operator< (const Number & N)

prosedur membandingkan nilai lebih kecil dengan nilai *N*

Parameters:

| | |
|----------|--|
| <i>N</i> | - Number yang nilainya akan dibandingkan. |
|----------|--|

Returns:

$L' = L1 < L2.$

Logic & Number::operator<= (const Number & *N*)

prosedur membandingkan nilai lebih kecil sama dengan nilai *N*.

Parameters:

| | |
|----------|--|
| <i>N</i> | - Number yang nilainya akan dibandingkan. |
|----------|--|

Returns:

$L' = L1 <= L2.$

Number & Number::operator= (Number & *n*)[virtual]

operator= untuk menyamakan nilai dari **Number** *n*.

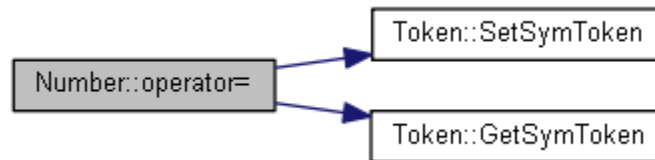
Parameters:

| | |
|----------|---|
| <i>n</i> | - Number yang nilainya akan disamakan. |
|----------|---|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Here is the call graph for this function:

**Logic & Number::operator== (const Number & *N*)**

prosedur membandingkan *_nilai* sama dengan *_nilai* *N*.

Parameters:

| | |
|----------|--|
| <i>N</i> | - Number yang nilainya akan dibandingkan. |
|----------|--|

Returns:

$L' = L1 == L2.$

Logic & Number::operator> (const Number & *N*)

prosedur membandingkan nilai lebih dari dengan nilai *N*.

Parameters:

| | |
|----------|--|
| <i>N</i> | - Number yang nilainya akan dibandingkan. |
|----------|--|

Returns:

$L' = L1 > L2.$

Logic & Number::operator>= (const Number & *N*)

prosedur membandingkan nilai lebih dari sama dengan _nilai *N*.

Parameters:

| | |
|----------|--|
| <i>N</i> | - Number yang nilainya akan dibandingkan. |
|----------|--|

Returns:

$L' = L1 >= L2.$

void Number::setNilai (int *_n*)

prosedur assign _nilai

Parameters:

| | |
|----------|---|
| <i>n</i> | - nilai yang nantinya akan di assign ke _nilai. |
|----------|---|

virtual int Number::toInt (string) [pure virtual]

fungsi yang merubah string menjadi integer

Parameters:

| | |
|----------|--|
| <i>s</i> | - string yang nantinya akan dirubah menjadi integer. |
|----------|--|

Returns:

i - integer dari string *n*.

Implemented in **NumberRomawi** (p.60), and **NumberArab** (p.53).

virtual string Number::toString (int) [pure virtual]

fungsi yang merubah integer menjadi string

Parameters:

| | |
|----------|--|
| <i>n</i> | - nilai yang nantinya akan dirubah menjadi string. |
|----------|--|

Returns:

s - string dari integer *n*.

Implemented in **NumberRomawi** (p.60), and **NumberArab** (p.53).

The documentation for this class was generated from the following files:

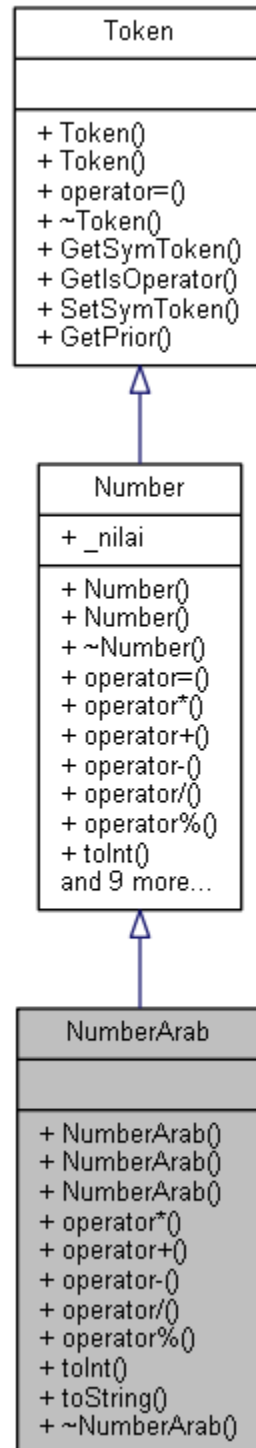
- Token/Number.h
- Token/Number.cpp

NumberArab Class Reference

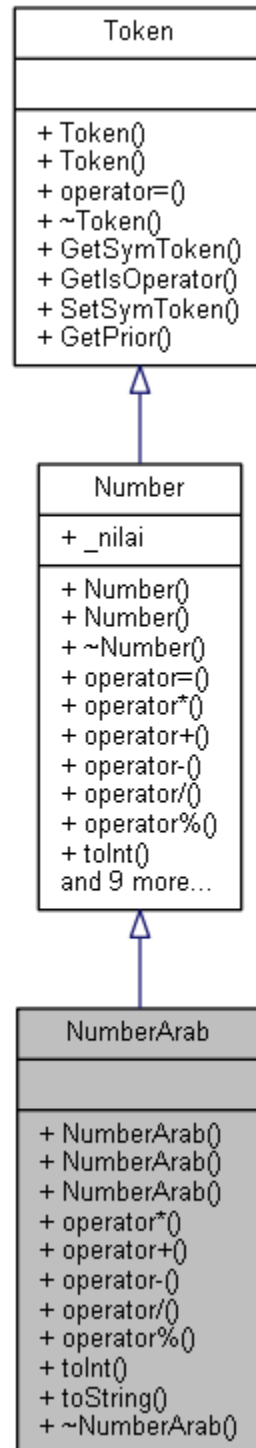
Kelas NumberArav bertugas untuk mengelola operasi bilangan arab.

#include <NumberArab.h>

Inheritance diagram for NumberArab:



Collaboration diagram for NumberArab:



Public Member Functions

- NumberArab ()

Konstruktor kelas **NumberArab**.

- **NumberArab** (string s)
*Konstruktor kelas **NumberArab** dengan parameter.*
- **NumberArab** (int _n)
*Konstruktor kelas **NumberArab** dengan parameter.*
- **Number & operator*** (const **Number** &)
operator untuk mengali nilai objek dengan nilai dari **Number** n.*
- **Number & operator+** (const **Number** &)
*operator+ untuk menambahkan nilai objek dengan nilai dari **Number** n.*
- **Number & operator-** (const **Number** &)
*operator- untuk mengurangi nilai objek dengan nilai dari **Number** n.*
- **Number & operator/** (const **Number** &)
*operator/ untuk membagi nilai objek dengan nilai dari **Number** n.*
- **Number & operator%** (const **Number** &)
fungsi yang merubah string menjadi integer
- int **toInt** (string s)
fungsi yang merubah string menjadi integer
- string **toString** (int n)
fungsi yang merubah integer menjadi string
- **~NumberArab** ()
*Destructor kelas **NumberArab**.*

Additional Inherited Members

Detailed Description

Kelas **NumberArab** bertugas untuk mengelola operasi bilangan arab.

Author:

Muhammad Aodyra Khaidir (13513063)

Version:

1.0

Description

Constructor & Destructor Documentation

NumberArab::NumberArab (string s)

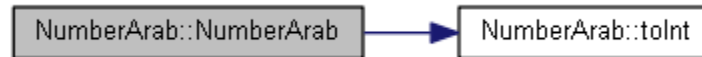
Konstruktor kelas **NumberArab** dengan parameter.

| STEI- ITB | IF2210-TB-01 | Halaman 50 dari 96 halaman |
|---|--------------|----------------------------|
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

Parameters:

| | |
|----------------|--|
| <code>s</code> | - String yang akan menjadi parameter Number |
|----------------|--|

Here is the call graph for this function:

**NumberArab::NumberArab (int `_n`)**

Konstruktor kelas **NumberArab** dengan parameter.

Parameters:

| | |
|-----------------|--|
| <code>_n</code> | - integer yang akan di assign ke <code>_nilai</code> |
|-----------------|--|

Member Function Documentation**Number & NumberArab::operator% (const Number & `M`)[virtual]**

fungsi yang merubah string menjadi integer

operator% untuk modulo nilai objek dengan nilai dari **Number** `n`.

Parameters:

| | |
|----------------|--|
| <code>s</code> | - string yang nantinya akan dirubah menjadi integer. |
|----------------|--|

Returns:

`i` - integer dari string `n`.

Parameters:

| | |
|----------------|--|
| <code>n</code> | - Number yang nilainya menjadi modulator. |
|----------------|--|

Implements **Number** (p.43).

Here is the call graph for this function:

**Number & NumberArab::operator* (const Number & `M`)[virtual]**

operator* untuk mengali nilai objek dengan nilai dari **Number** `n`.

Parameters:

| | |
|----------------|--|
| <code>n</code> | - Number yang nilainya akan dikali. |
|----------------|--|

Returns:

`n'` - Objek **Number** yang telah diassign dari `n`.

Implements **Number** (p.43).

Here is the call graph for this function:



Number & NumberArab::operator+ (const Number & *M*)[virtual]

operator+ untuk menambahkan nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan ditambah. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implements **Number** (p.44).

Here is the call graph for this function:



Number & NumberArab::operator- (const Number & *M*)[virtual]

operator- untuk mengurangi nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dikurang. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implements **Number** (p.44).

Here is the call graph for this function:



Number & NumberArab::operator/ (const Number & *M*)[virtual]

operator/ untuk membagi nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dibagi. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implements **Number** (p.44).

Here is the call graph for this function:



int NumberArab::toInt (string s)[virtual]

fungsi yang merubah string menjadi integer

Parameters:

| | |
|----------|--|
| <i>s</i> | - string yang nantinya akan dirubah menjadi integer. |
|----------|--|

Returns:

i - integer dari string n.

Implements **Number** (p.46).

Here is the caller graph for this function:



string NumberArab::toString (int n)[virtual]

fungsi yang merubah integer menjadi string

Parameters:

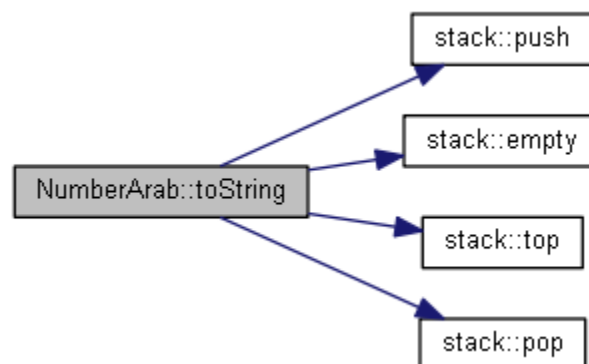
| | |
|----------|--|
| <i>n</i> | - nilai yang nantinya akan dirubah menjadi string. |
|----------|--|

Returns:

s - string dari integer n.

Implements **Number** (p.46).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

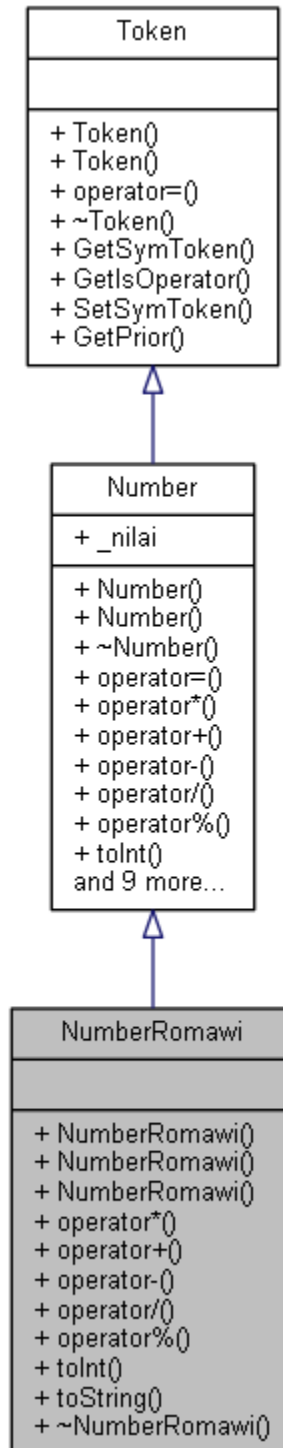
- Token/NumberArab.h
- Token/NumberArab.cpp

NumberRomawi Class Reference

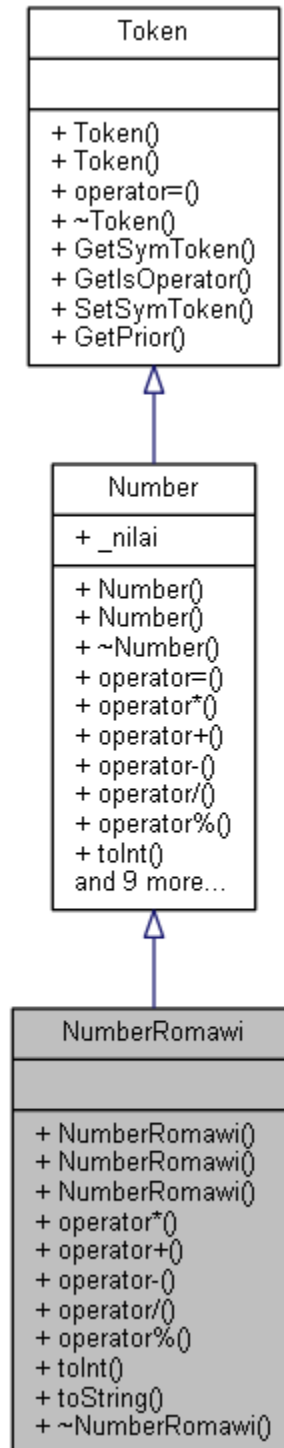
Kelas **Number** bertugas untuk mengelola operasi bilangan romawi.

#include <NumberRomawi.h>

Inheritance diagram for NumberRomawi:



Collaboration diagram for NumberRomawi:



Public Member Functions

- NumberRomawi ()

Konstruktor kelas **NumberRomawi**.

- **NumberRomawi** (string s)
*Konstruktor kelas **Number** dengan parameter.*
- **NumberRomawi** (int _n)
*Konstruktor kelas **Number** dengan parameter.*
- **Number & operator*** (const **Number** &)
operator untuk mengali nilai objek dengan nilai dari **Number** n.*
- **Number & operator+** (const **Number** &)
*operator+ untuk menambahkan nilai objek dengan nilai dari **Number** n.*
- **Number & operator-** (const **Number** &)
*operator- untuk mengurangi nilai objek dengan nilai dari **Number** n.*
- **Number & operator/** (const **Number** &)
*operator/ untuk membagi nilai objek dengan nilai dari **Number** n.*
- **Number & operator%** (const **Number** &)
*operator% untuk modulo nilai objek dengan nilai dari **Number** n.*
- int **toInt** (string s)
fungsi yang merubah string menjadi integer
- string **toString** (int n)
fungsi yang merubah integer menjadi string
- **~NumberRomawi** ()
*Destructor kelas **NumberRomawi**.*

Additional Inherited Members

Detailed Description

Kelas **Number** bertugas untuk mengelola operasi bilangan romawi.

Author:

Muhammad Aodyra Khaidir (13513063)

Version:

1.0

Description

Constructor & Destructor Documentation

NumberRomawi::NumberRomawi (string s)

Konstruktor kelas **Number** dengan parameter.

| STEI- ITB | IF2210-TB-01 | Halaman 57 dari 96 halaman |
|---|--------------|----------------------------|
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

Parameters:

| | |
|----------------|--|
| <code>s</code> | - String yang akan menjadi parameter Number . |
|----------------|--|

Here is the call graph for this function:

**NumberRomawi::NumberRomawi (int `_n`)**

Konstruktor kelas **Number** dengan parameter.

Parameters:

| | |
|-----------------|--|
| <code>_n</code> | - integer yang akan di assign ke <code>_nilai</code> |
|-----------------|--|

Member Function Documentation**Number & NumberRomawi::operator% (const Number & `M`)[virtual]**

operator% untuk modulo nilai objek dengan nilai dari **Number** `n`.

Parameters:

| | |
|----------------|--|
| <code>n</code> | - Number yang nilainya menjadi modulator. |
|----------------|--|

Returns:

`n'` - Objek **Number** yang telah diassign dari `n`.

Implements **Number** (p.43).

Here is the call graph for this function:

**Number & NumberRomawi::operator* (const Number & `M`)[virtual]**

operator* untuk mengali nilai objek dengan nilai dari **Number** `n`.

Parameters:

| | |
|----------------|--|
| <code>n</code> | - Number yang nilainya akan dikali. |
|----------------|--|

Returns:

`n'` - Objek **Number** yang telah diassign dari `n`.

Implements **Number** (p.43).

Here is the call graph for this function:



Number & NumberRomawi::operator+ (const Number & *N*)[virtual]

operator+ untuk menambahkan nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan ditambah. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implements **Number** (p.44).

Here is the call graph for this function:



Number & NumberRomawi::operator- (const Number & *N*)[virtual]

operator- untuk mengurangi nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dikurang. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implements **Number** (p.44).

Here is the call graph for this function:



Number & NumberRomawi::operator/ (const Number & *N*)[virtual]

operator/ untuk membagi nilai objek dengan nilai dari **Number** *n*.

Parameters:

| | |
|----------|--|
| <i>n</i> | - Number yang nilainya akan dibagi. |
|----------|--|

Returns:

n' - Objek **Number** yang telah diassign dari *n*.

Implements **Number** (p.44).

Here is the call graph for this function:



int NumberRomawi::toInt (string s)[virtual]

fungsi yang merubah string menjadi integer

Parameters:

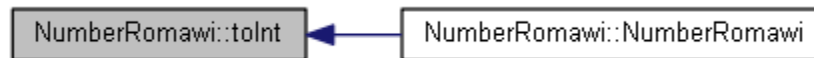
| | |
|----------|--|
| <i>s</i> | - string yang nantinya akan dirubah menjadi integer. |
|----------|--|

Returns:

i - integer dari string n.

Implements **Number** (p.46).

Here is the caller graph for this function:



string NumberRomawi::toString (int n)[virtual]

fungsi yang merubah integer menjadi string

Parameters:

| | |
|----------|--|
| <i>n</i> | - nilai yang nantinya akan dirubah menjadi string. |
|----------|--|

Returns:

s - string dari integer n.

Implements **Number** (p.46).

The documentation for this class was generated from the following files:

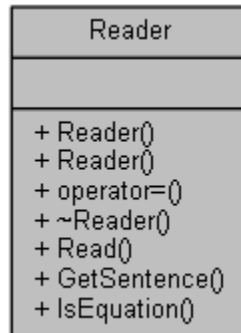
- Token/NumberRomawi.h
- Token/NumberRomawi.cpp

Reader Class Reference

Kelas **Reader** bertugas menerima input dari user kemudian mengkategorikan input tersebut termasuk command atau ekspresi.

```
#include <Reader.h>
```

Collaboration diagram for Reader:



Public Member Functions

- **Reader ()**
*Konstruktur kelas **Reader**.*
- **Reader (const Reader &)**
Copy constructor kelas logger.
- **Reader & operator= (const Reader &)**
Copy constructor kelas logger.
- **~Reader ()**
Destruktor kelas logger.
- string **Read ()**
Membaca perintah dari user sekaligus meng-update data member isEquation.
- string **GetSentence ()**
Mengambil kalimat dari input yang telah dibaca.
- bool **IsEquation ()**
Mengembalikan predikat apakah sebuah string merupakan equation.

Detailed Description

Kelas **Reader** bertugas menerima input dari user kemudian mengkategorikan input tersebut termasuk command atau ekspresi.

Author:

Luqman A. Siswanto (13513024)

Version:

1.0

Description

Constructor & Destructor Documentation

Reader::Reader (const Reader & r)

Copy constructor kelas logger.

Parameters:

| | |
|---------------|---------------------|
| <i>Reader</i> | : yang akan di-copy |
|---------------|---------------------|

Member Function Documentation

string Reader::GetSentence ()

Mengambil kalimat dari input yang telah dibaca.

Returns:

string - string yang berhasil dibaca

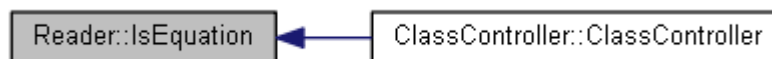
bool Reader::IsEquation ()

Mengembalikan predikat apakah sebuah string merupakan equation.

Returns:

bool - jika true, maka string adalah equation

Here is the caller graph for this function:



Reader & Reader::operator= (const Reader & r)

Copy constructor kelas logger.

Parameters:

| | |
|---------------|---------------------|
| <i>Reader</i> | : yang akan di-copy |
|---------------|---------------------|

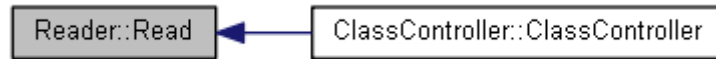
string Reader::Read ()

Membaca perintah dari user sekaligus meng-update data member isEquation.

Returns:

string - string yang berhasil dibaca

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

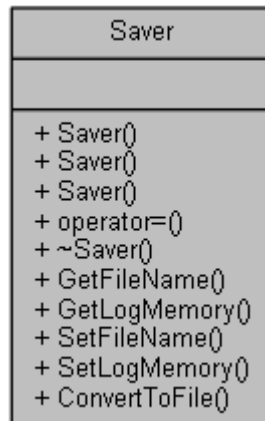
- Command/Reader.h
- Command/Reader.cpp

Saver Class Reference

Kelas **Saver** bertugas untuk mengelola penyimpanan instruksi ke dalam file eksternal.

#include <Saver.h>

Collaboration diagram for Saver:



Public Member Functions

- **Saver ()**
Konstruktur kelas saver.
- **Saver (string S, Logger L)**
*Konstruktur kelas saver dengan parameter. [I.S] string S dan **Logger** L terdefinisi. [F.S] Elemen **Saver** telah terassign dengan string S dan **Logger** L.*
- **Saver (const Saver &S)**
Copy Konstruktur kelas saver.
- **Saver & operator= (const Saver &S)**
Assignment kelas saver.
- **~Saver ()**
Destruktor kelas saver.
- string **GetFileName ()**
Getter untuk mendapatkan nama file.
- **Logger GetLogMemory ()**
Getter untuk mendapatkan variabel logger.
- void **SetFileName (string)**
Setter untuk mengubah nama file.
- void **SetLogMemory (Logger)**
*Setter untuk mengubah **Logger**.*
- void **ConvertToFile ()**
Prosedur program membuat file eksternal.

Detailed Description

Kelas **Saver** bertugas untuk mengelola penyimpanan instruksi ke dalam file eksternal.

Author:

Ahmad Darmawan (13513096)

Version:

1.0

Constructor & Destructor Documentation

Saver::Saver (string S, Logger L)

Konstruktor kelas saver dengan parameter. [I.S] string S dan **Logger** L terdefinisi. [F.S] Elemen **Saver** telah terassign dengan string S dan **Logger** L.

Konstruktor kelas saver dengan parameter.

Parameters:

| | |
|----------|---|
| <i>S</i> | - String yang akan menjadi nama file. |
| <i>L</i> | - Logger berisi instruksi (ekspresi dan perintah) yang disimpan. |

Saver::Saver (const Saver & S)

Copy Konstruktor kelas saver.

Parameters:

| | |
|----------|------------------------|
| <i>S</i> | - Objek Saver . |
|----------|------------------------|

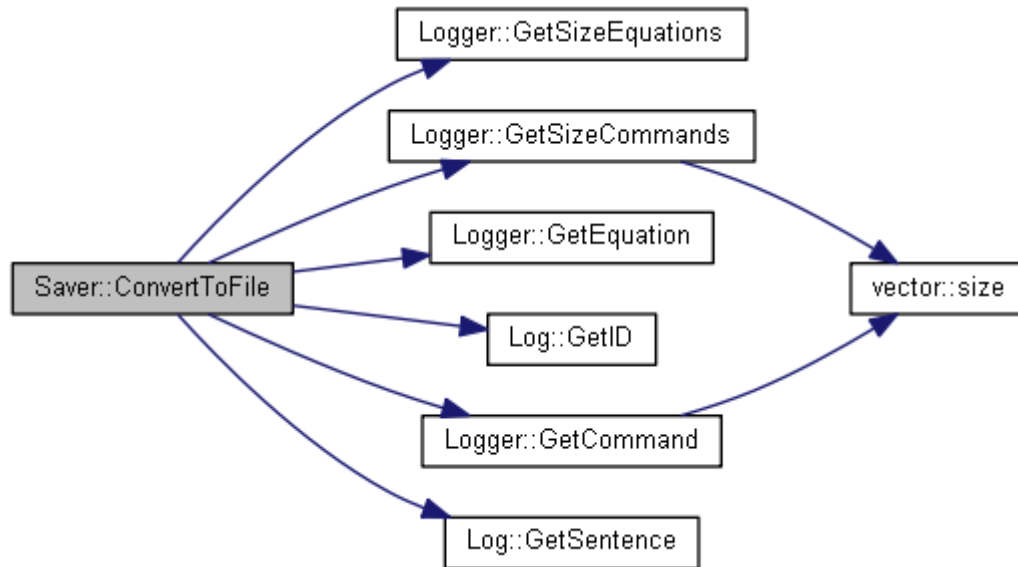
Member Function Documentation

void Saver::ConvertToFile ()

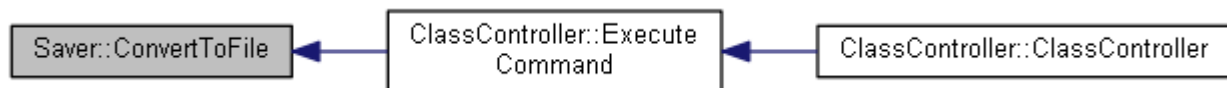
Prosedur program membuat file eksternal.

[I.S] String `_filename` dan **Logger** `_logMemory` terdefinisi. [F.S] Tercipta file eksternal bernama `_filename` dengan masukan dari `_logMemory`.

Here is the call graph for this function:



Here is the caller graph for this function:



string Saver::GetFileName ()

Getter untuk mendapatkan nama file.

Returns:

S - string nama file.

Logger Saver::GetLogMemory ()

Getter untuk mendapatkan variabel logger.

Returns:

L - variabel logger.

Saver & Saver::operator= (const Saver & S)

Assignment kelas saver.

Parameters:

| | |
|---|----------------|
| S | - Objek Saver. |
|---|----------------|

Returns:

S' - Objek **Saver** yang telah diassign dari S.

void Saver::SetFileName (string filename)

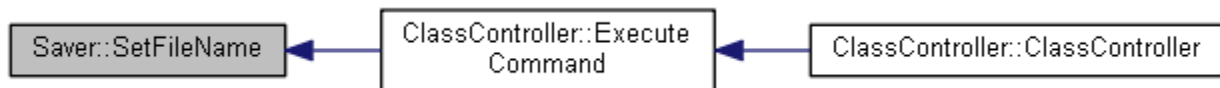
Setter untuk mengubah nama file.

[I.S] String terdefinisi yang dapat dijadikan nama file. [F.S] _filename terassign dengan parameter string.

Parameters:

| | |
|----------|--------------|
| <i>S</i> | - nama file. |
| <i>S</i> | - nama file. |

Here is the caller graph for this function:



void Saver::SetLogMemory (Logger logMemory)

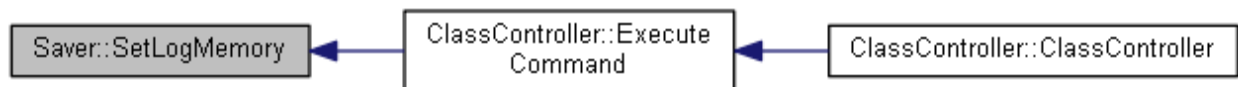
Setter untuk mengubah **Logger**.

[I.S] **Logger** terdefinisi. [F.S] _logMemory terassign dengan parameter **Logger**.

Parameters:

| | |
|----------|-------------------|
| <i>L</i> | - Logger . |
| <i>L</i> | - Logger . |

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

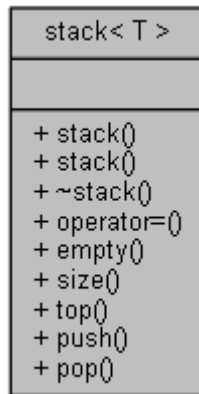
- Command/Saver.h
- Command/Saver.cpp

stack< T > Class Template Reference

kelas stack adalah implementasi vector yang ekuivalen stack STL C++

```
#include <stack.h>
```

Collaboration diagram for stack< T >:



Public Member Functions

- **stack ()**
Konstruktor kelas stack.
- **stack (const stack< T > &)**
Copy constructor kelas stack.
- **~stack ()**
Destruktor kelas stack.
- **stack< T > & operator= (const stack< T > &)**
Operator assignment kelas stack.
- **bool empty ()**
Mengembalikan predikat apakah stack kosong.
- **int size ()**
Mengembalikan ukuran stack.
- **T & top ()**
Mengembalikan elemen teratas stack.
- **void push (const T &)**
Menambahkan item pada paling atas stack.
- **void pop ()**
Menghapus item pada paling atas stack.

Detailed Description

template<class T>class stack< T >

kelas stack adalah implementasi vector yang ekuivalen stack STL C++

Wiwit Rifa'i (13513073)

Version:

1.0

Description

Constructor & Destructor Documentation

```
template<class T> stack< T >::stack (const stack< T > &  S)
```

Copy constructor kelas stack.

Parameters:

| | |
|--------------|----------------------|
| <i>stack</i> | : yang akan di-copy. |
|--------------|----------------------|

Member Function Documentation

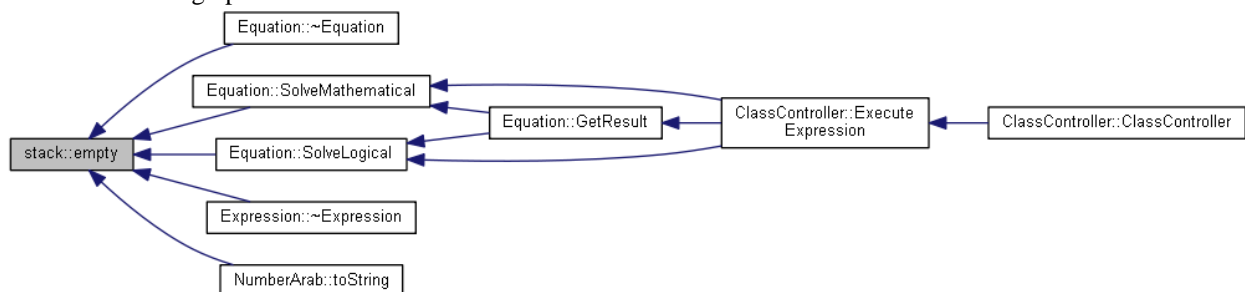
```
template<class T > bool stack< T >::empty ()
```

Mengembalikan predikat apakah stack kosong.

Returns:

bool : predikat kosong stack.

Here is the caller graph for this function:



```
template<class T> stack< T > & stack< T >::operator= (const stack< T > & S)
```

Operator assignment kelas stack.

Parameters:

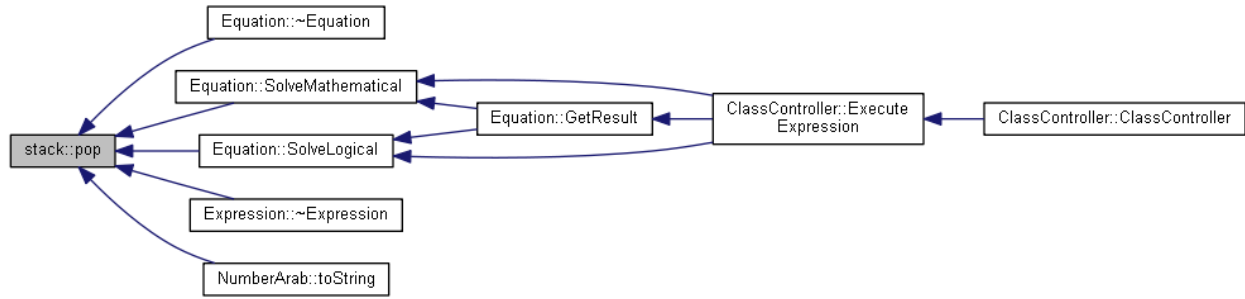
| | |
|--------------|----------------------|
| <i>stack</i> | : yang akan di-copy. |
|--------------|----------------------|

template<class T > void stack< T >::pop ()

Menghapus item pada paling atas stack.

I. S. stack tidak kosong.

Here is the caller graph for this function:



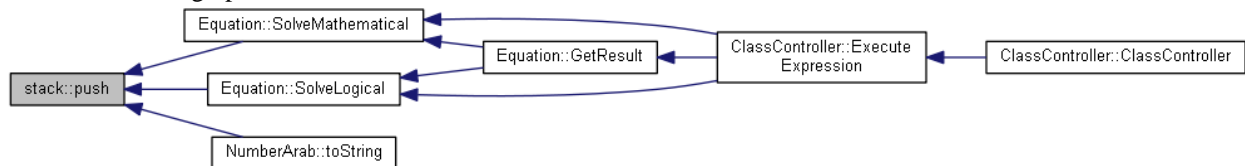
template<class T> void stack< T >::push (const T & _new)

Menambahkan item pada paling atas stack.

Parameters:

| | |
|--------------|--------------------------------|
| <i>class</i> | T : item yang akan ditambahkan |
|--------------|--------------------------------|

Here is the caller graph for this function:



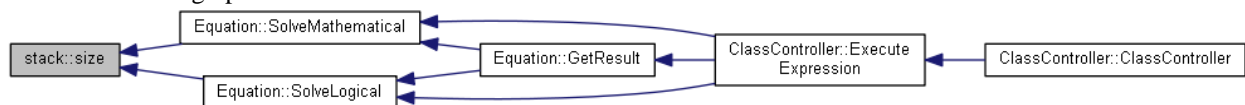
template<class T > int stack< T >::size ()

Mengembalikan ukuran stack.

Returns:

int - ukuran stack.

Here is the caller graph for this function:



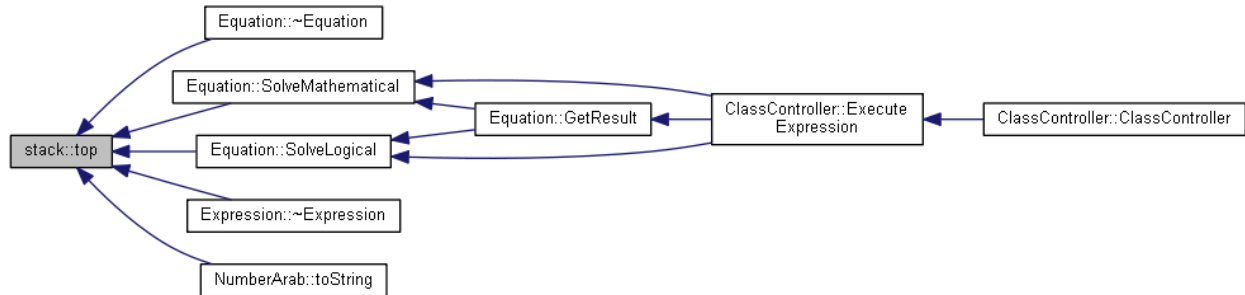
template<class T > T & stack< T >::top ()

Mengembalikan elemen teratas stack.

Returns:

reference class T : elemen teratas

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

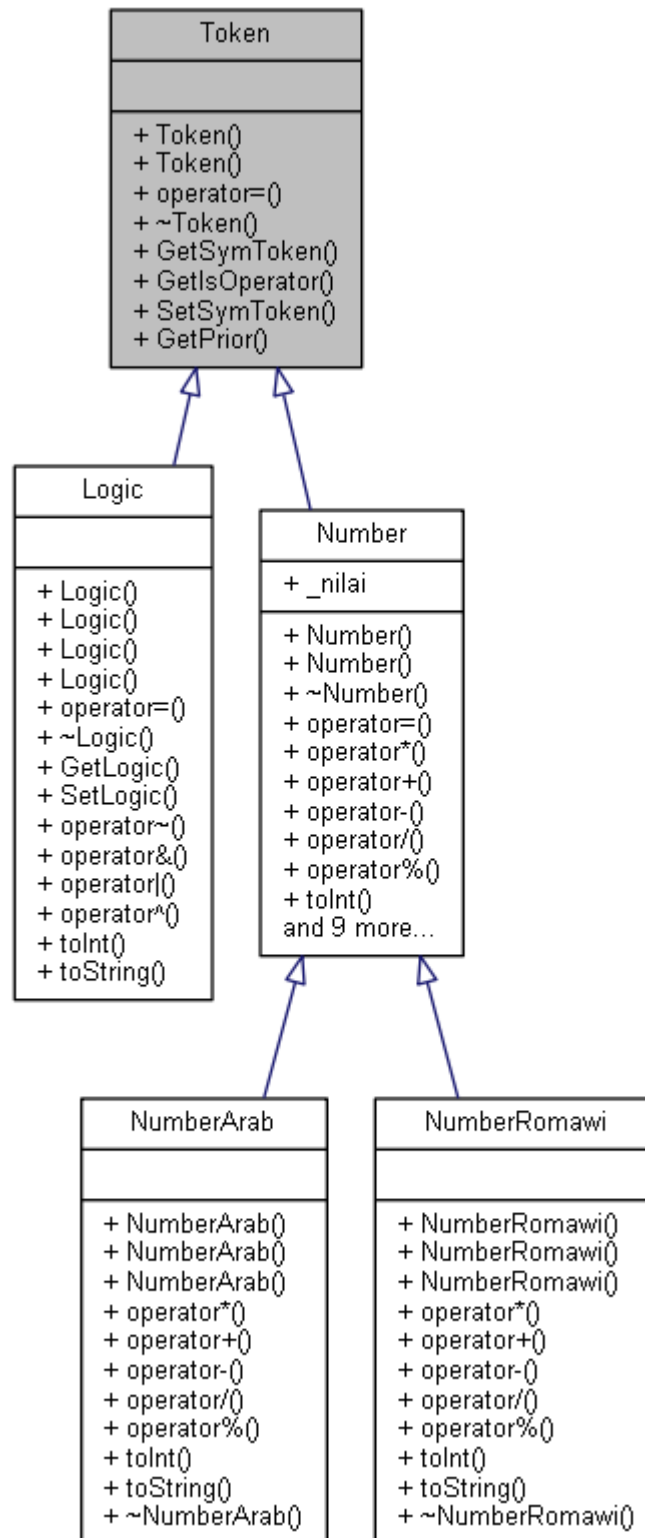
- `Stack/stack.h`

Token Class Reference

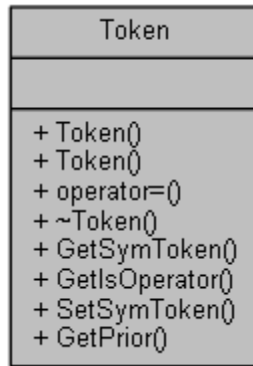
Kelas **Token** menyimpan operator dan operan.

```
#include <Token.h>
```

Inheritance diagram for Token:



Collaboration diagram for Token:



Public Member Functions

- **Token** (string S)
*Konstruktur kelas **Token** dengan Parameter.*
- **Token** (const **Token** &T)
*Copy Konstruktur kelas **Token**.*
- **Token & operator=** (const **Token** &T)
*Assignment kelas **Logic**.*
- **~Token** ()
*Destruktor kelas **Token**.*
- string **GetSymToken** () const
Getter untuk mendapatkan nilai `_symToken`.
- bool **GetIsOperator** ()
Getter untuk mendapatkan nilai `_isOperator`.
- void **SetSymToken** (string S)
Setter untuk mengubah nilai `_symToken`.
- int **GetPrior** ()
Getter untuk mendapatkan nilai `_prior`.

Detailed Description

Kelas **Token** menyimpan operator dan operan.

Author:

Ahmad Darmawan (13513096)

Version:

1.0

Constructor & Destructor Documentation

Token::Token (string S)

| STEI- ITB | IF2210-TB-01 | Halaman 74 dari 96 halaman |
|---|--------------|----------------------------|
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

Konstruktor kelas **Token** dengan Parameter.

Parameters:

| | |
|----------|--------------------------------------|
| <i>S</i> | - String yang menjadi anggota token. |
|----------|--------------------------------------|

Token::Token (const Token & *T*)

Copy Konstruktor kelas **Token**.

Parameters:

| | |
|----------|----------------------------------|
| <i>T</i> | - Variabel Token masukan. |
|----------|----------------------------------|

Returns:

Ti - Variabel **Token** yang tercopy.

Member Function Documentation

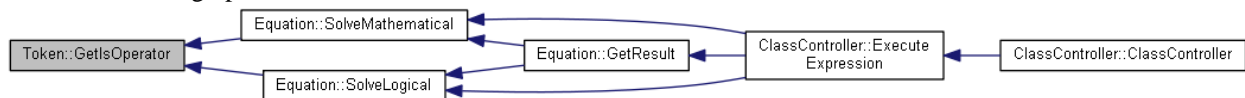
bool Token::GetIsOperator ()

Getter untuk mendapatkan nilai `_isOperator`.

Returns:

bool - boolean nilai `_isOperator`.

Here is the caller graph for this function:



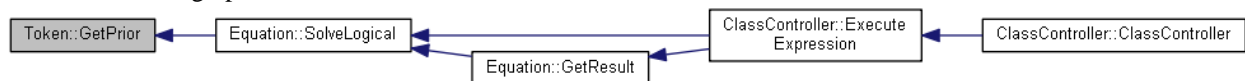
int Token::GetPrior ()

Getter untuk mendapatkan nilai `_prior`.

Returns:

i - integer nilai `_prior`.

Here is the caller graph for this function:



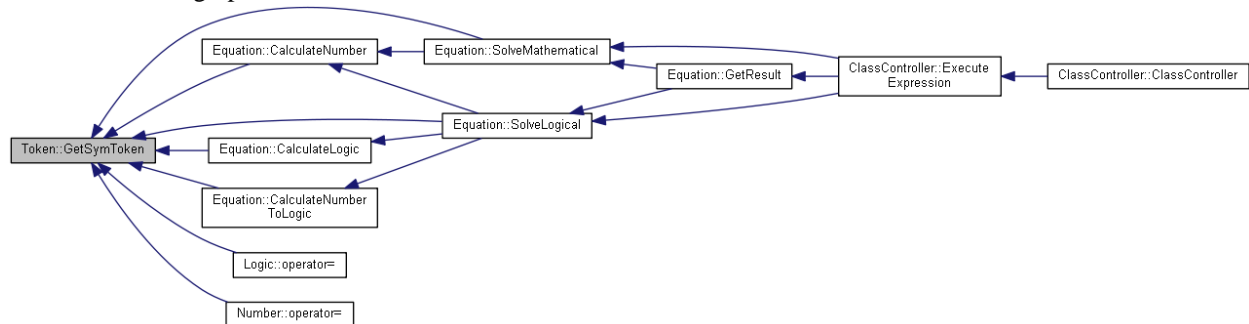
string Token::GetSymToken () const

Getter untuk mendapatkan nilai `_symToken`.

Returns:

S - string nilai _symToken.

Here is the caller graph for this function:

**Token & Token::operator= (const Token & T)**

Assignment kelas **Logic**.

Parameters:

| | |
|----------|----------------------------------|
| <i>T</i> | - Variabel Token masukan. |
|----------|----------------------------------|

Returns:

Ti - Variabel **Token** yang terassign.

void Token::SetSymToken (string S)

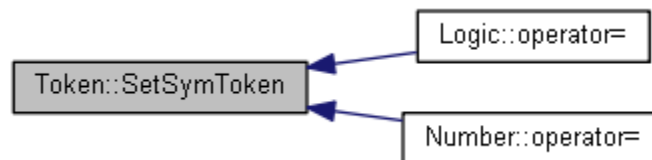
Setter untuk mengubah nilai _symToken.

[I.S] String S merupakan simbol token yang valid. [F.S] Simbol token akan terset dengan S.

Parameters:

| | |
|----------|---|
| <i>S</i> | - nilai _symToken yang akan dimasukkan. |
|----------|---|

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

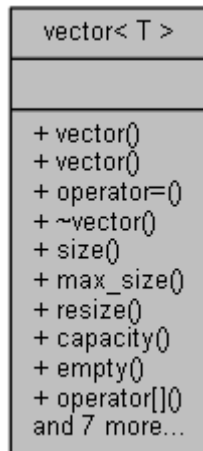
- Token/**Token.h**
- Token/**Token.cpp**

vector< T > Class Template Reference

Vector adalah implementasi vector yang ekuivalen vector STL C++.

#include <vector.h>

Collaboration diagram for vector< T >:



Public Member Functions

- **vector ()**
Konstruktorkelas vector.
- **vector (const vector< T > &)**
Copy constructor kelas vector.
- **vector< T > & operator= (vector< T >)**
Operator assignment kelas vector.
- **~vector ()**
Destruktorkelas vector.
- **int size ()**
Mengembalikan ukuran vector.
- **int max_size ()**
Mengembalikan ukuran maksimal vector saat ini sebelum alokasi kembali.
- **void resize (int)**
Mengubah ukuran vector.
- **int capacity ()**
Mengembalikan ukuran maksimal vector saat ini sebelum alokasi kembali.
- **bool empty ()**
Mengembalikan predikat apakah vector kosong.
- **T & operator[] (int)**
Mengembalikan isi kontainer pada indeks tertentu.
- **T & at (int)**
Mengembalikan isi kontainer pada indeks tertentu.
- **T & front ()**

Mengembalikan isi kontainer paling awal.

- **T & back ()**
Mengembalikan isi kontainer paling akhir.
- **void push_back (T)**
Memasukkan item pada akhir kontainer.
- **void pop_back ()**
Melepaskan item paling belakang vector.
- **void swap (vector< T > &)**
Menukar kontainer vector beserta atributnya dengan vector lain.
- **void clear ()**
Mengosongkan isi vector.

Detailed Description

template<class T>class vector< T >

Vector adalah implementasi vector yang ekuivalen vector STL C++.

Author:

Luqman A. Siswanto (13513024)

Version:

1.0

Description

Constructor & Destructor Documentation

template<class T> vector< T >::vector (const vector< T > & v)

Copy constructor kelas vector.

Parameters:

| | |
|---------------|---------------------|
| <i>vector</i> | : yang akan di-copy |
|---------------|---------------------|

Member Function Documentation

template<class T > T & vector< T >::at (int i)

Mengembalikan isi kontainer pada indeks tertentu.

Parameters:

| | |
|------------|-----------------|
| <i>int</i> | - indeks vector |
|------------|-----------------|

Returns:

reference class T : item pada indeks tertentu

template<class T > T & vector< T >::back ()

Mengembalikan isi kontainer paling akhir.

Returns:

reference class T : item pada indeks paling belakang

template<class T > int vector< T >::capacity ()

Mengembalikan ukuran maksimal vector saat ini sebelum alokasi kembali.

Returns:

int - ukuran max vector

template<class T > bool vector< T >::empty ()

Mengembalikan predikat apakah vector kosong.

Returns:

bool : predikat kosong vector

template<class T > T & vector< T >::front ()

Mengembalikan isi kontainer paling awal.

Returns:

reference class T : item pada indeks terawal

template<class T > int vector< T >::max_size ()

Mengembalikan ukuran maksimal vector saat ini sebelum alokasi kembali.

Returns:

int - ukuran max vector

template<class T> vector< T > & vector< T >::operator= (vector< T > v)

Operator assignment kelas vector.

| | | |
|---|---------------------|-----------------------------------|
| STEI- ITB | IF2210-TB-01 | Halaman 79 dari 96 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

Parameters:

| | |
|---------------|---------------------|
| <i>vector</i> | : yang akan di-copy |
|---------------|---------------------|

template<class T > T & vector< T >::operator[] (int i)

Mengembalikan isi kontainer pada indeks tertentu.

Parameters:

| | |
|------------|-----------------|
| <i>int</i> | - indeks vector |
|------------|-----------------|

Returns:

reference class T : item pada indeks tertentu

template<class T > void vector< T >::pop_back ()

Melepaskan item paling belakang vector.

I. S. vector tidak kosong

template<class T> void vector< T >::push_back (T e)

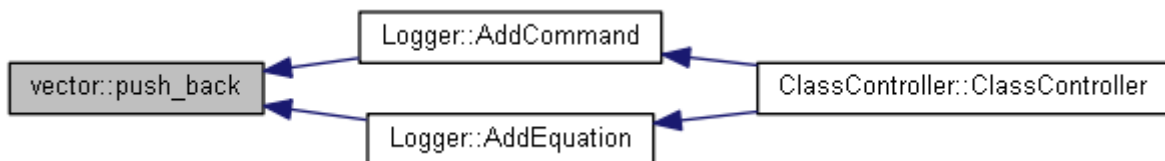
Memasukkan item pada akhir kontainer.

Bila vector penuh, maka mengalokasikan memori tambahan sebesar default Size

Parameters:

| | |
|--------------|-------------------------------|
| <i>class</i> | T : item yang akan dimasukkan |
|--------------|-------------------------------|

Here is the caller graph for this function:



template<class T > void vector< T >::resize (int n)

Mengubah ukuran vector.

Parameters:

| | |
|---------------|---------------|
| <i>ukuran</i> | vector tujuan |
|---------------|---------------|

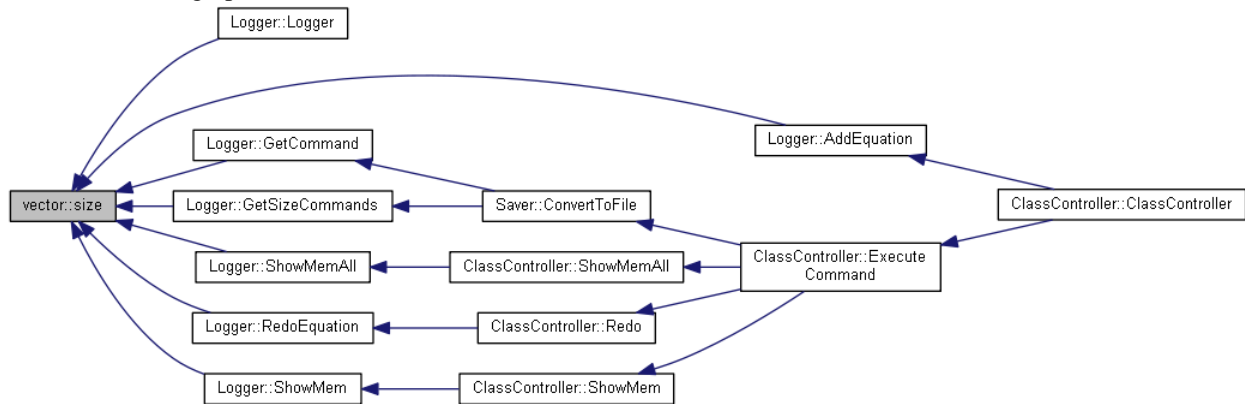
template<class T > int vector< T >::size ()

Mengembalikan ukuran vector.

Returns:

int - ukuran vector

Here is the caller graph for this function:



```
template<class T> void vector< T >::swap (vector< T > & v)
```

Menukar kontainer vector beserta atributnya dengan vector lain.

Parameters:

vector

: vector yang akan ditukar dengan object this

The documentation for this class was generated from the following file:

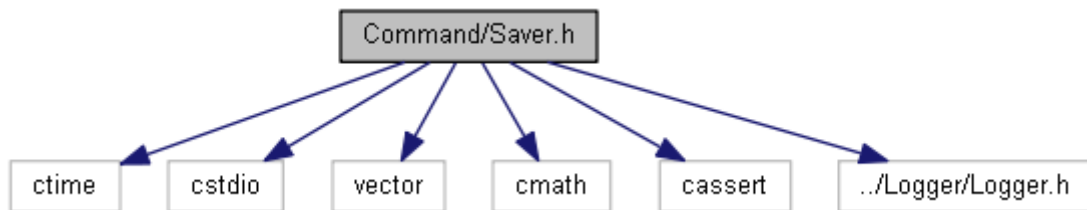
- Vector/vector.h

File Documentation

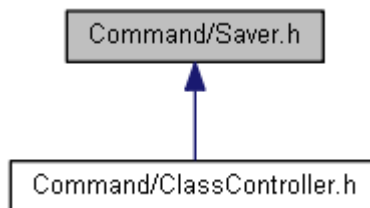
Command/Saver.h File Reference

```
#include <ctime>
#include <cstdio>
#include <vector>
#include <cmath>
#include <cassert>
#include "../Logger/Logger.h"
```

Include dependency graph for Saver.h:



This graph shows which files directly or indirectly include this file:



Classes

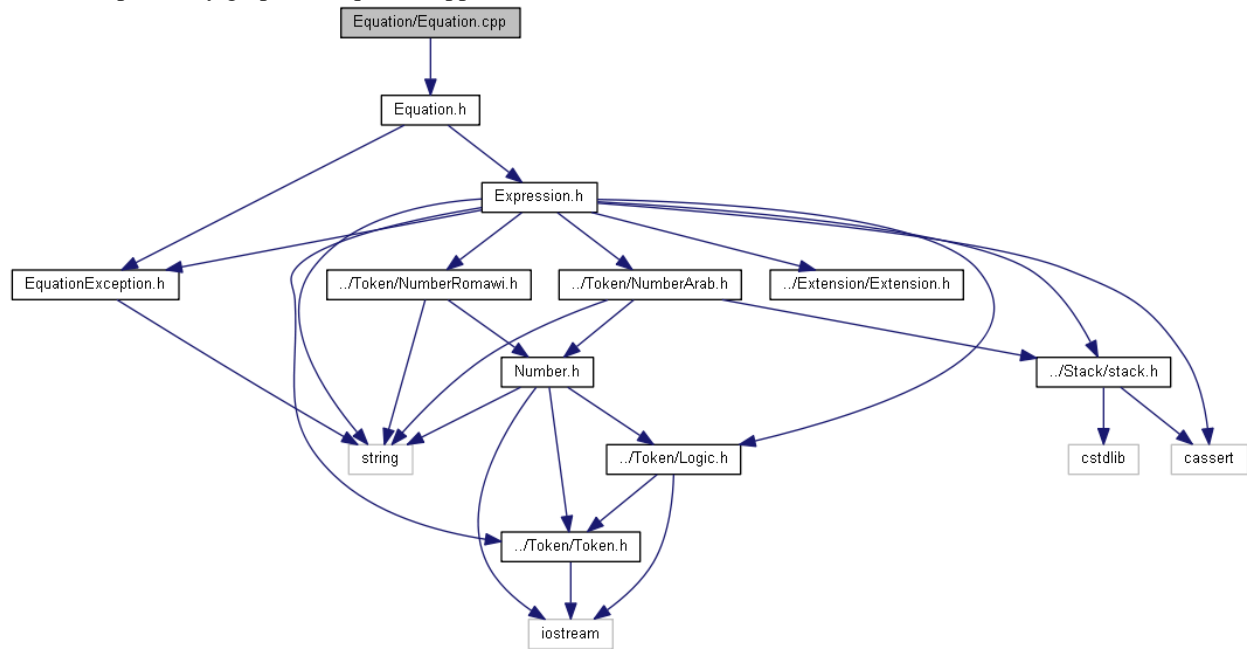
- class **Saver**

Kelas **Saver** bertugas untuk mengelola penyimpanan instruksi ke dalam file eksternal.

Equation/Equation.cpp File Reference

```
#include "Equation.h"
```

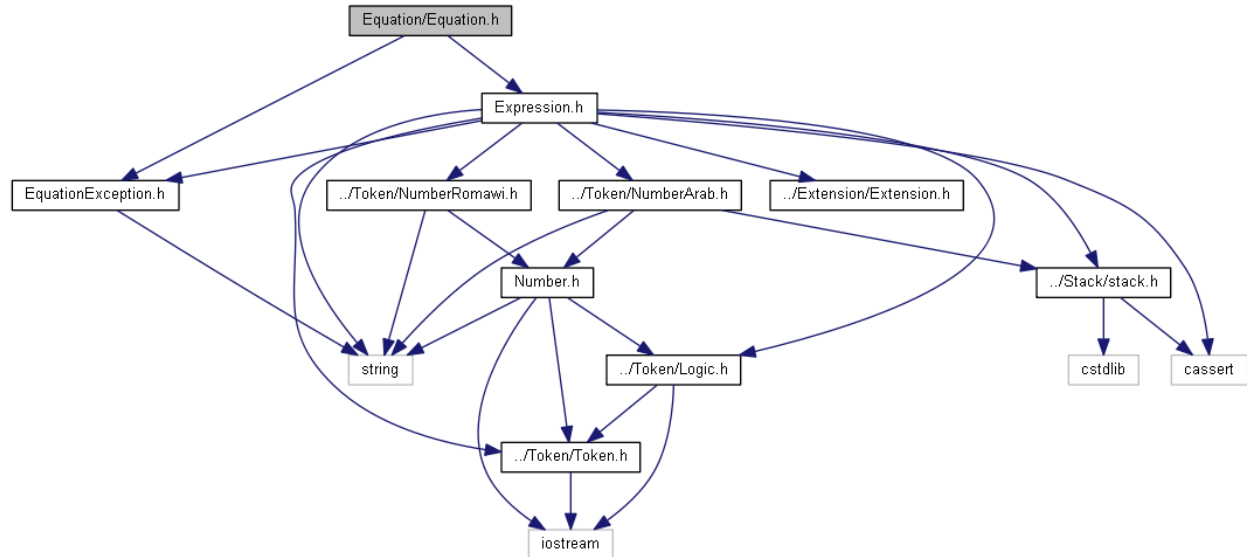
Include dependency graph for Equation.cpp:



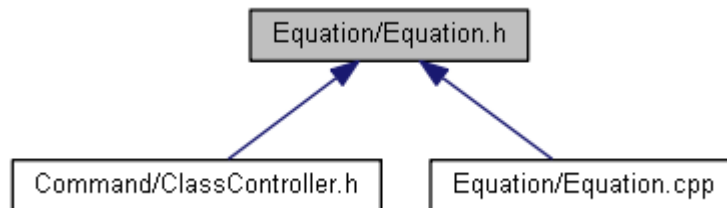
Equation/Equation.h File Reference

```
#include "Expression.h"
#include "EquationException.h"
```

Include dependency graph for Equation.h:



This graph shows which files directly or indirectly include this file:



Classes

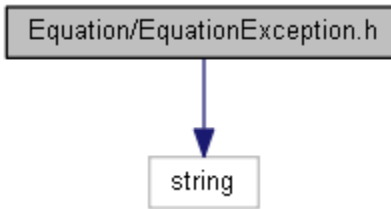
- class **Equation**

Kelas **Equation** bertanggung jawab dalam perhutingan hasil ekspresi matematika ataupun logika dalam bentuk prefix. Kelas **Equation** merupakan inheritent dari kelas **Expression**.

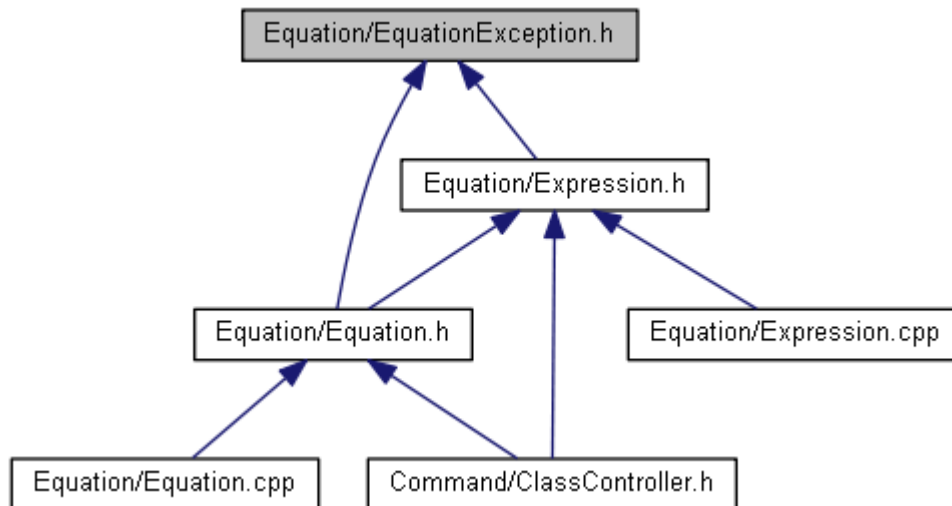
Equation/EquationException.h File Reference

```
#include <string>
```

Include dependency graph for EquationException.h:



This graph shows which files directly or indirectly include this file:



Classes

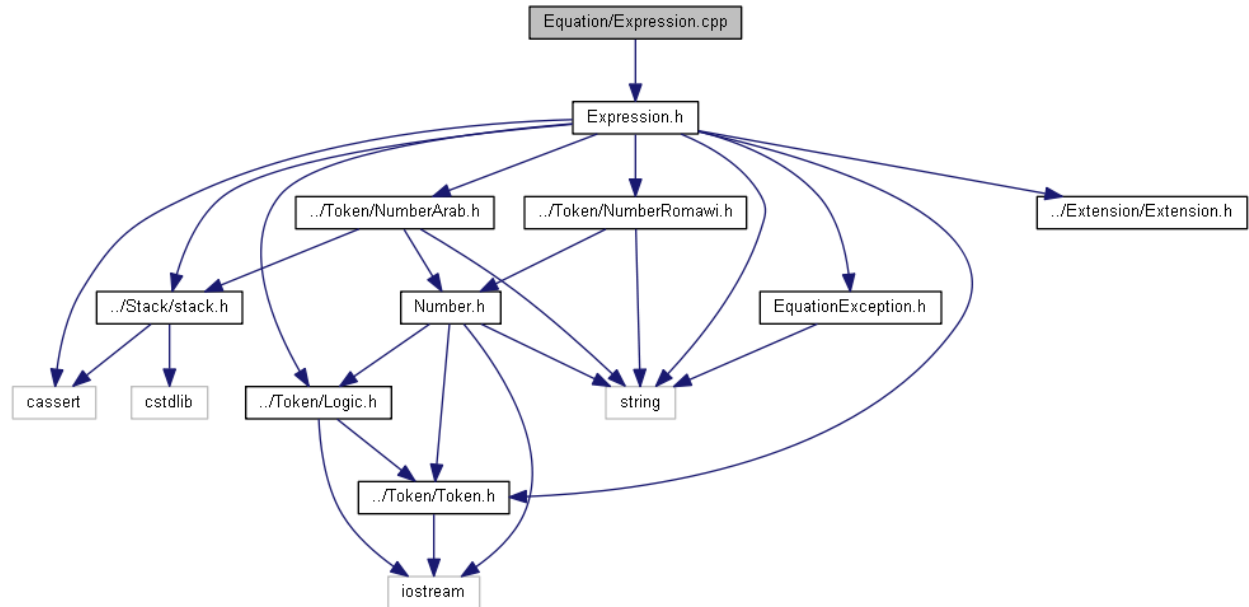
- class **EquationException**

*kelas **EquationException** merupakan kelas mengurus exception pada equation.*

Equation/Expression.cpp File Reference

```
#include "Expression.h"
```

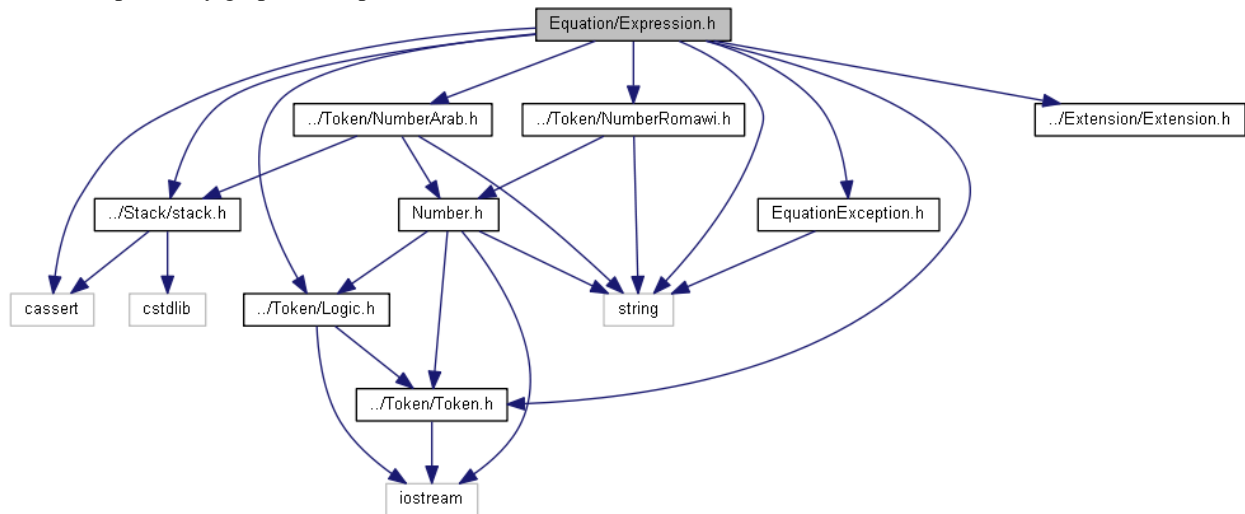
Include dependency graph for Expression.cpp:



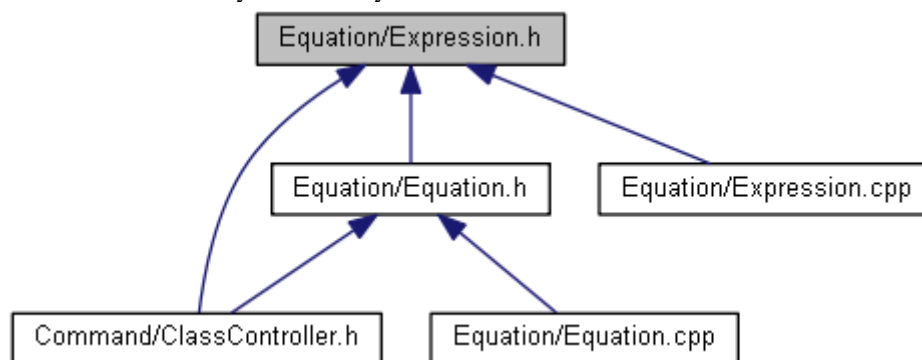
Equation/Expression.h File Reference

```
#include "EquationException.h"
#include <string>
#include <cassert>
#include "../Stack/stack.h"
#include "../Token/Logic.h"
#include "../Token/Token.h"
#include "../Token/NumberArab.h"
#include "../Token/NumberRomawi.h"
#include "../Extension/Extension.h"
```

Include dependency graph for Expression.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **Expression**

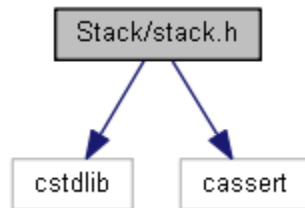
Kelas **Expression** bertanggung jawab dalam mengolah ekspresi matematika atau logika seperti mengubah string menjadi ekspresi matematika dan mengubah bentuk infix atau postfix menjadi prefix agar bisa dihitung di class **Equation**.

Stack/stack.h File Reference

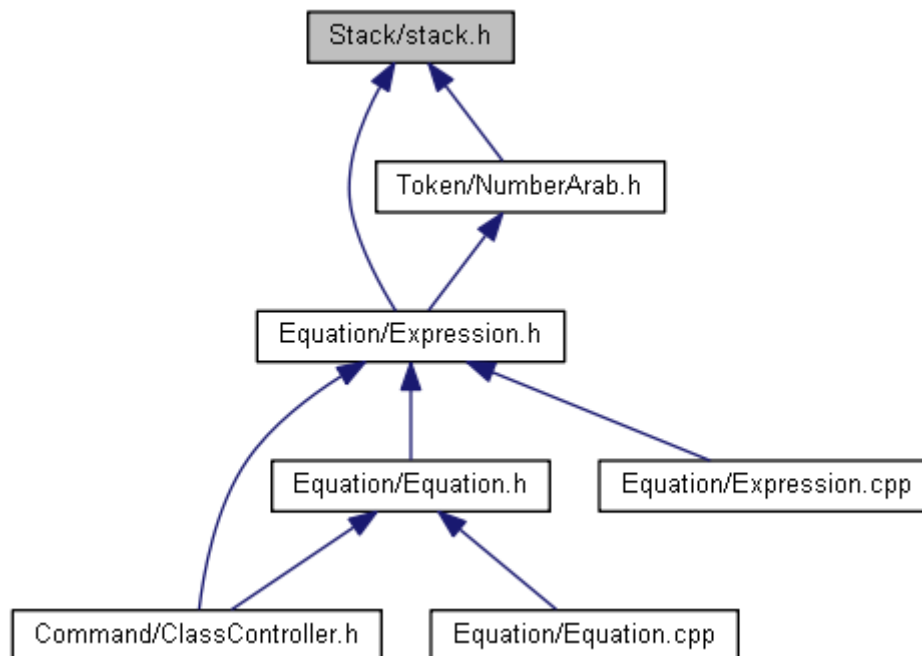
```
#include <cstdlib>
```

```
#include <cassert>
```

Include dependency graph for stack.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **stack**< T >

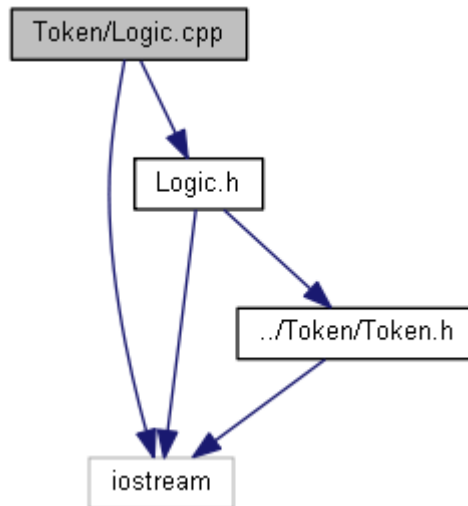
kelas stack adalah implementasi vector yang ekuivalen stack STL C++

Token/Logic.cpp File Reference

```
#include <iostream>
```

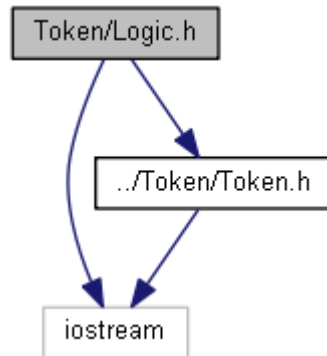
```
#include "Logic.h"
```

Include dependency graph for Logic.cpp:

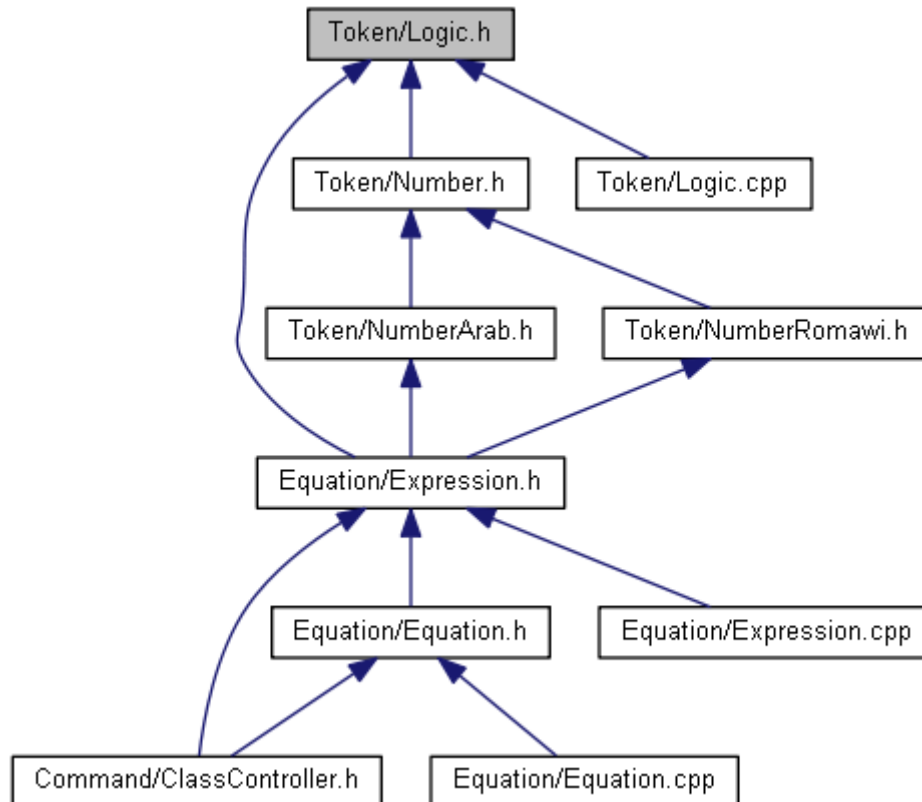


Token/Logic.h File Reference

```
#include <iostream>
#include "../Token/Token.h"
Include dependency graph for Logic.h:
```



This graph shows which files directly or indirectly include this file:



Classes

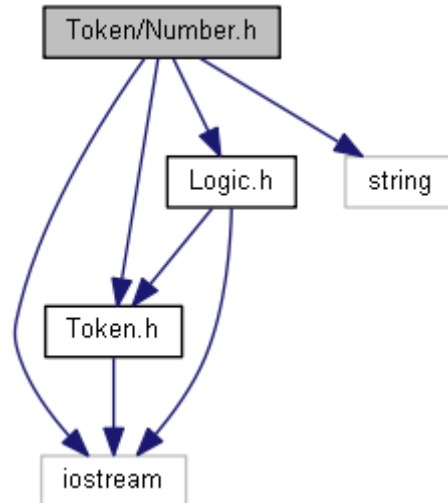
- class **Logic**

Kelas **Logic** mengatur perhitungan operasi logika (`~`, `&`, `^`, `/`)

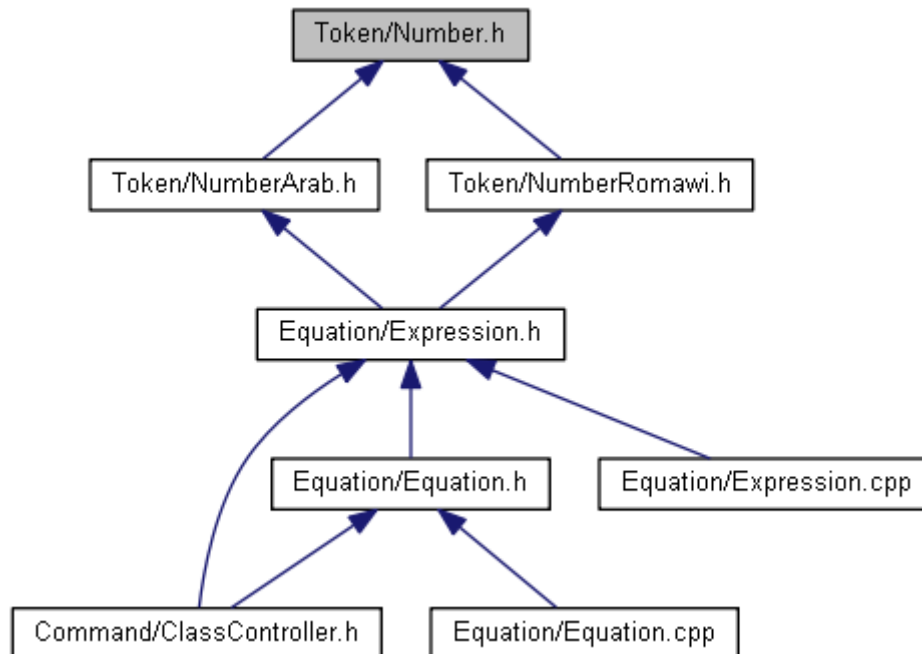
Token/Number.h File Reference

```
#include <iostream>
#include "Token.h"
#include "Logic.h"
#include <string>
```

Include dependency graph for Number.h:



This graph shows which files directly or indirectly include this file:



Classes

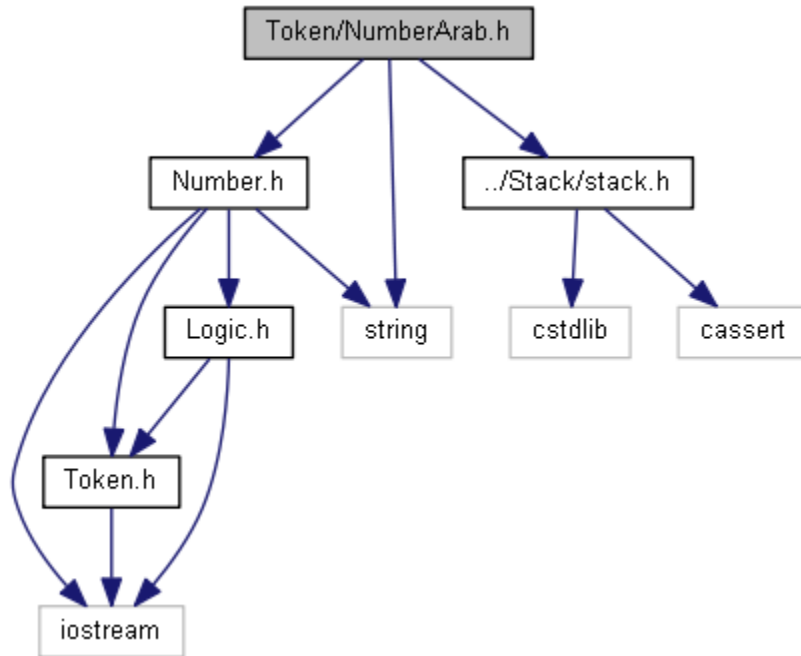
- class **Number**

*Kelas **Number** bertugas untuk mengelola operasi bilangan.*

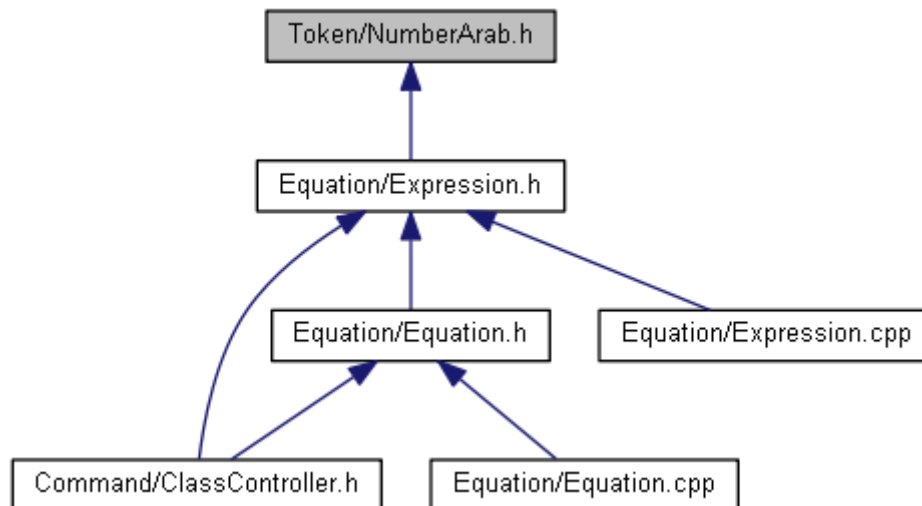
| | | |
|---|----------------------------|-----------------------------------|
| STEI- ITB | <i>IF2210-TB-01</i> | Halaman 92 dari 96 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

Token/NumberArab.h File Reference

```
#include "Number.h"
#include "../Stack/stack.h"
#include <string>
Include dependency graph for NumberArab.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class **NumberArab**

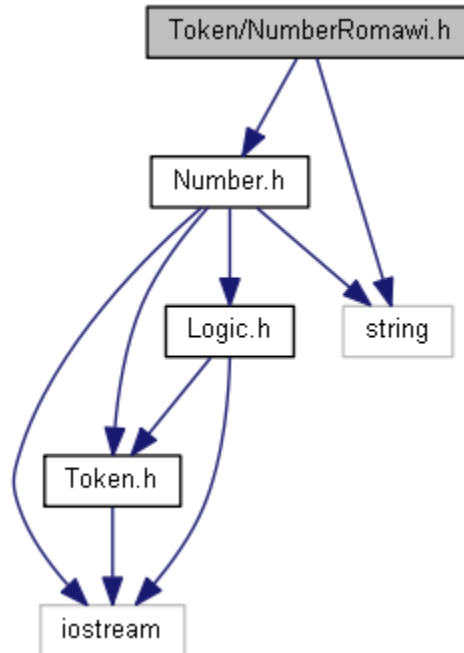
Kelas NumberArav bertugas untuk mengelola operasi bilangan arab.

Token/NumberRomawi.h File Reference

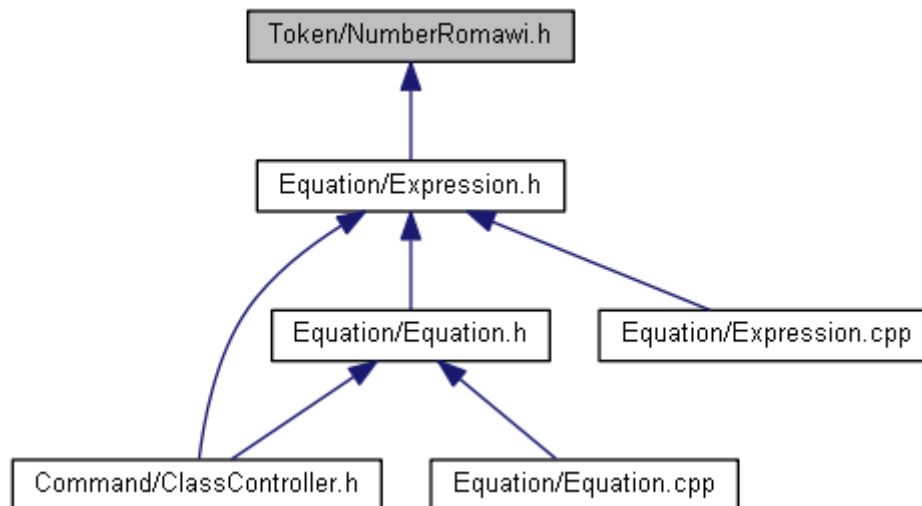
```
#include "Number.h"
```

```
#include <string>
```

Include dependency graph for NumberRomawi.h:



This graph shows which files directly or indirectly include this file:



Classes

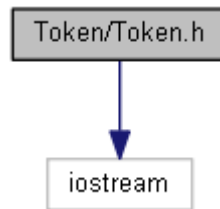
- class **NumberRomawi**

Kelas **Number** bertugas untuk mengelola operasi bilangan romawi.

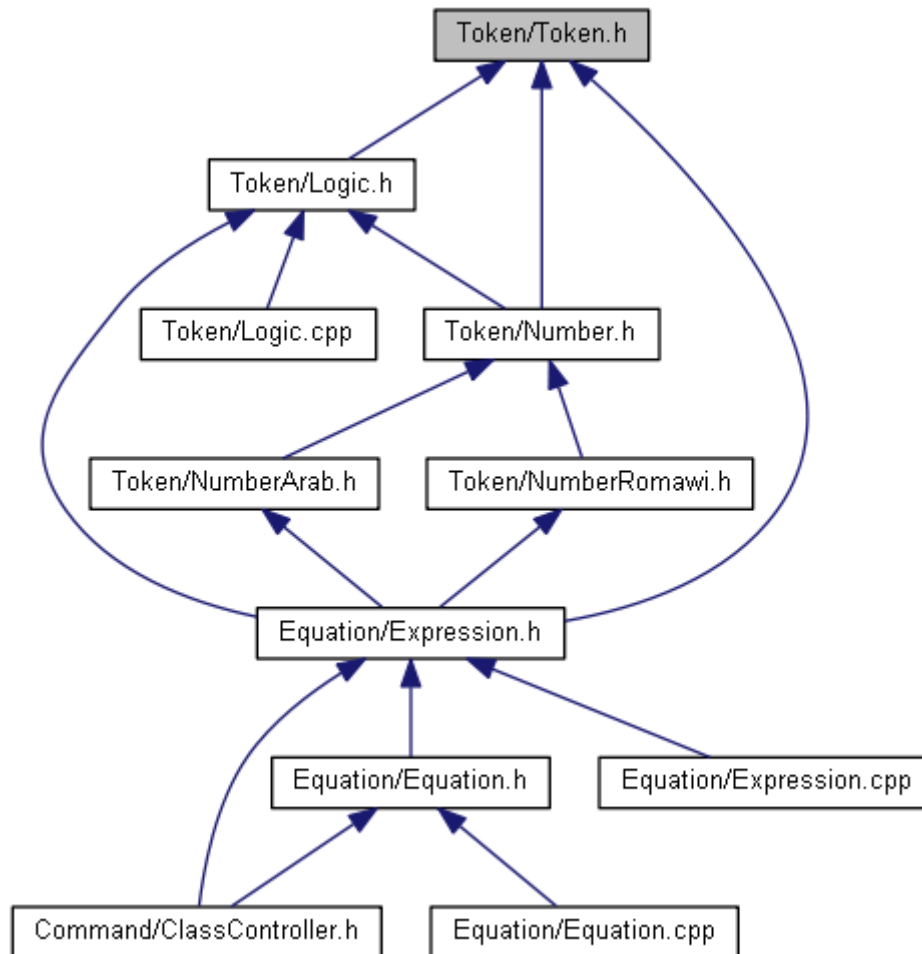
Token/Token.h File Reference

```
#include <iostream>
```

Include dependency graph for Token.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **Token**

Kelas **Token** menyimpan operator dan operand

