

TUGAS: FINAL

PEMROGRAMAN BERORIENTASI OBJEK

“ OO ANALISIS DAN OO DESIGN “

(TEMA, OOPERATOR)



DISUSUN OLEH:

HESTI SULISTIAWATI (21916069)

SELFY (21916080)

PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

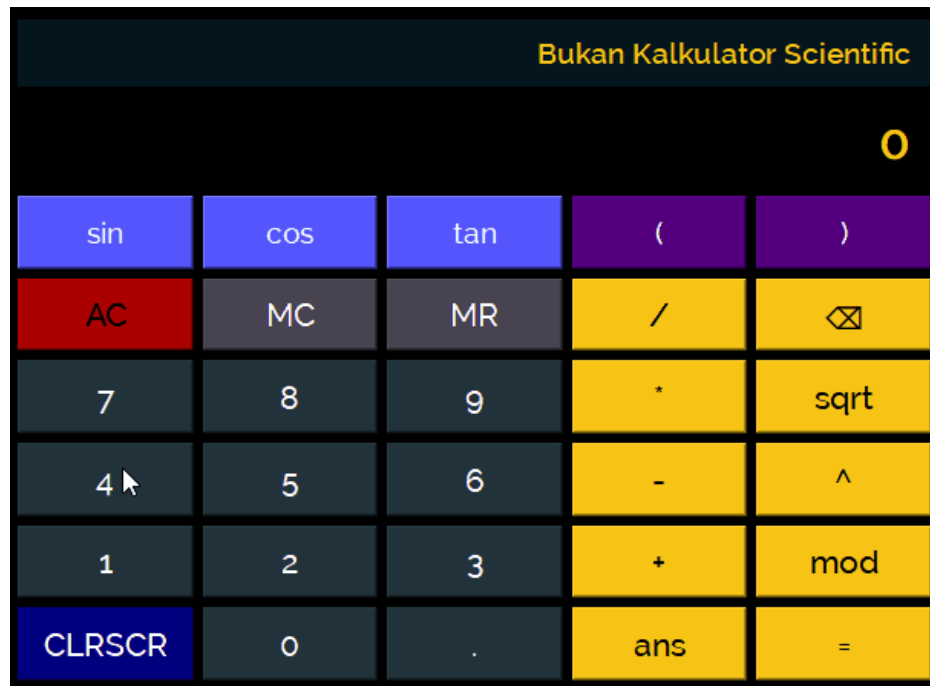
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

UNIVERSITAS MUHAMMADIYAH KENDARI

KENDARI

2021

Kalkulator sederhana yang memanfaatkan konsep OOP dalam implementasinya. Proyek ini merupakan Tugas Final.



Kalkulator ini memiliki fitur dasar kalkulator pada umumnya seperti:

- Perhitungan operasi biner (tambah, kurang, kali, bagi)
- Perhitungan operasi uner (akar)
- Fitur memory ans dan queue (MC dan MR)
- Exception handling
- Evaluasi ekspresi biner sederhana (A opr B)

dan tentu saja, OOperator adalah kalkulator yang mempunyai, sehingga memiliki fitur tambahan seperti berikut

- Perhitungan operasi biner lanjutan (pow, mod)
- Perhitungan operasi uner lanjutan (sin, cos, tan)
- Menangani ekspresi kurung

- Evaluasi ekspresi lanjutan, contoh: $\sqrt{182 * 100 + 25} + 1 - \cos(0)$

Instruksi berikut adalah bagaimana cara menyiapkan enviroment proyek ini untuk keperluan pengembangan maupun pengujian. Di dalam proyek ini juga sudah tercantum file executable yang sudah tidak perlu di build lagi.

Berikut adalah keperluan enviroment dari proyek ini

- C++ 11
- Qt Creator

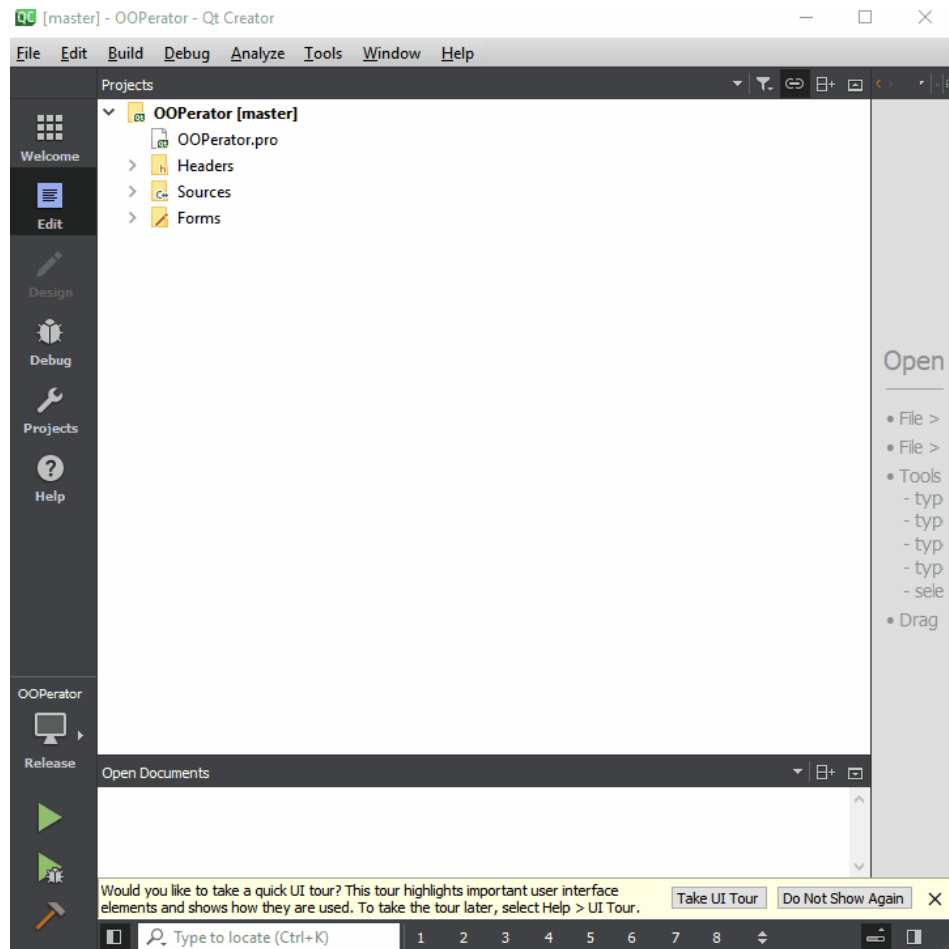
Installing

Installer Qt Creator dapat diunduh melalui tautan [berikut](#).

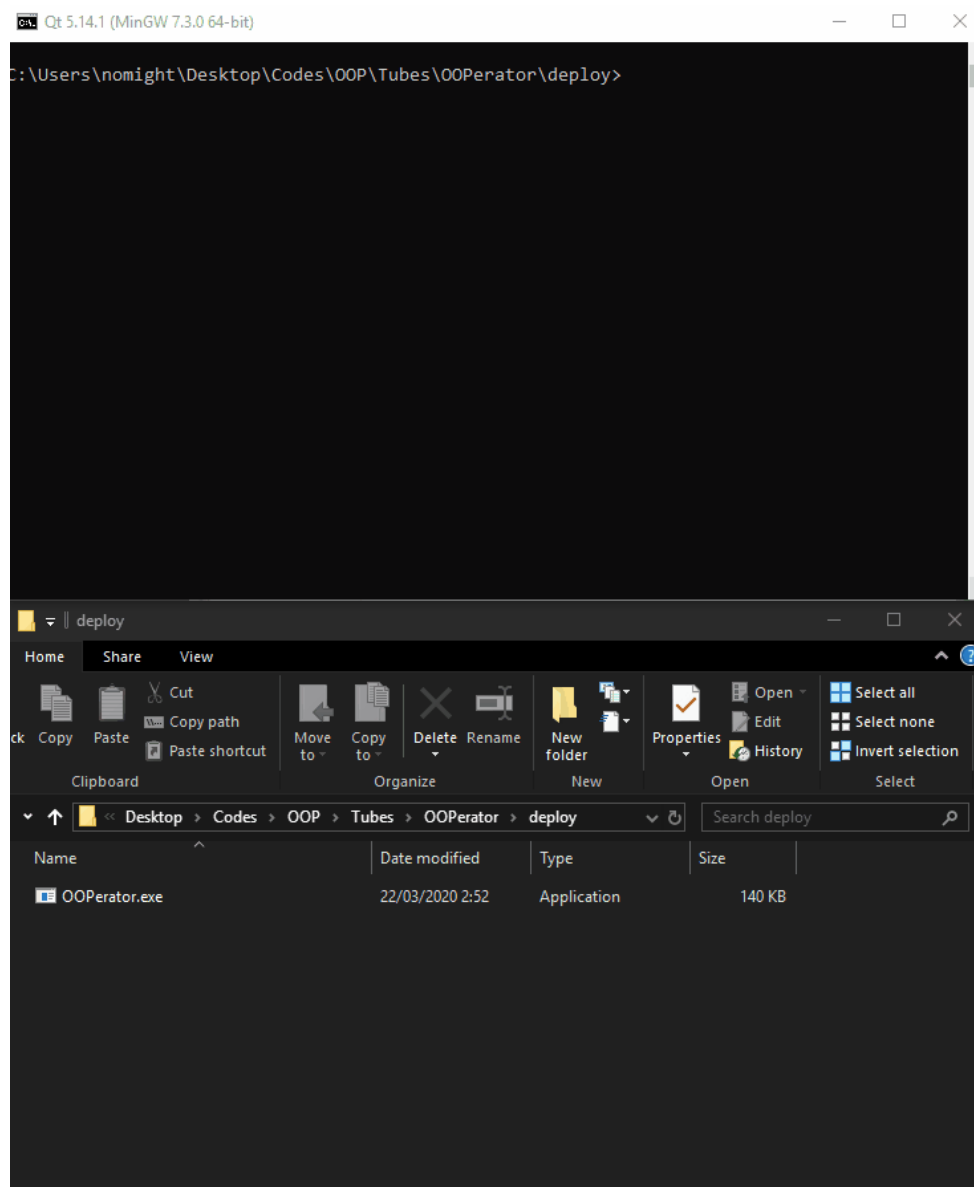
Pada tahap instalasi, silakan memilih komponen-komponen berikut

- ▼ ☒ Developer and Designer Tools
 - Qt Creator 4.11.1
 - ☒ Qt Creator 4.11.1 CDB Debugger Support
 - ☒ Debugging Tools for Windows
 - ☐ Qt Creator 4.11.1 Debug Symbols
 - ☐ Qt Creator 4.11.1 Plugin Development
 - ☐ Qt 3D Studio 2.6.0
 - > ☐ Qt 3D Studio OpenGL Runtime 2.6.0 for Qt 5.12.6
 - ☒ MinGW 7.3.0 32-bit
 - ☒ MinGW 7.3.0 64-bit

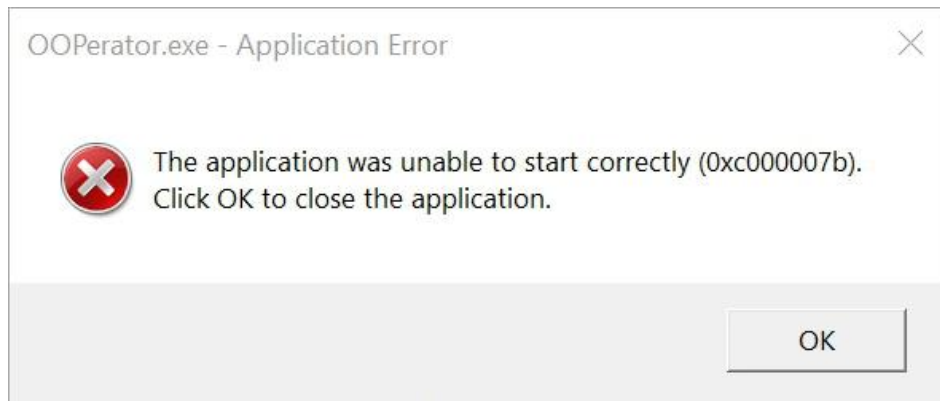
Untuk dapat membangun aplikasi ini secara keseluruhan pada sistem Anda, Anda hanya perlu membuka OOperator.pro dengan Qt Creator dan melakukan proses build dari situ.



Dalam proyek ini juga telah diberikan file executable yang sudah di build dengan dependencynya pada folder deploy. Jika anda ingin mendeploy dari awal, dapat menggunakan perintah berikut pada Qt windeployqt.exe --quick --no-translations .



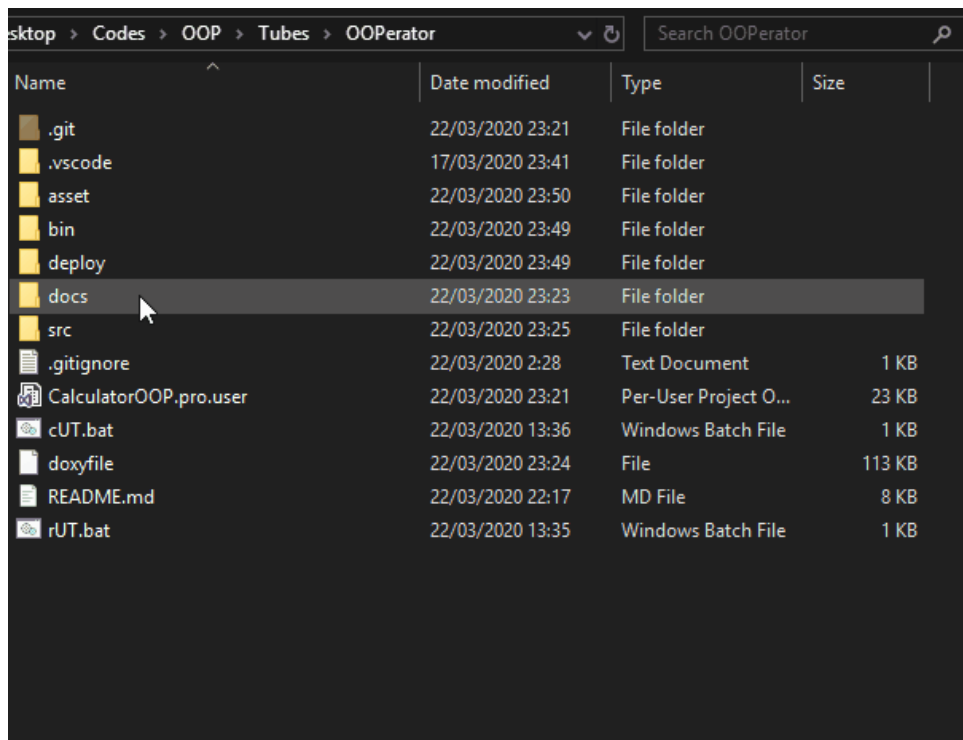
Jika terdapat error seperti berikut.



Maka silahkan mencoba build menggunakan Mingw32 di Qt Creator, dan deploy pada tahap sebelumnya menggunakan terminal Qt 32-bit.

Running

File executable yang sudah di build berada pada folder deploy



Unit Testing

Unit testing dilakukan menggunakan framework Catch2. Testing dilakukan untuk setiap Expression, dan beberapa fungsi-fungsi kelas Parser.

Untuk melakukan kompilasi unit test, dapat menjalankan script compile Unit Test (cUT.bat)

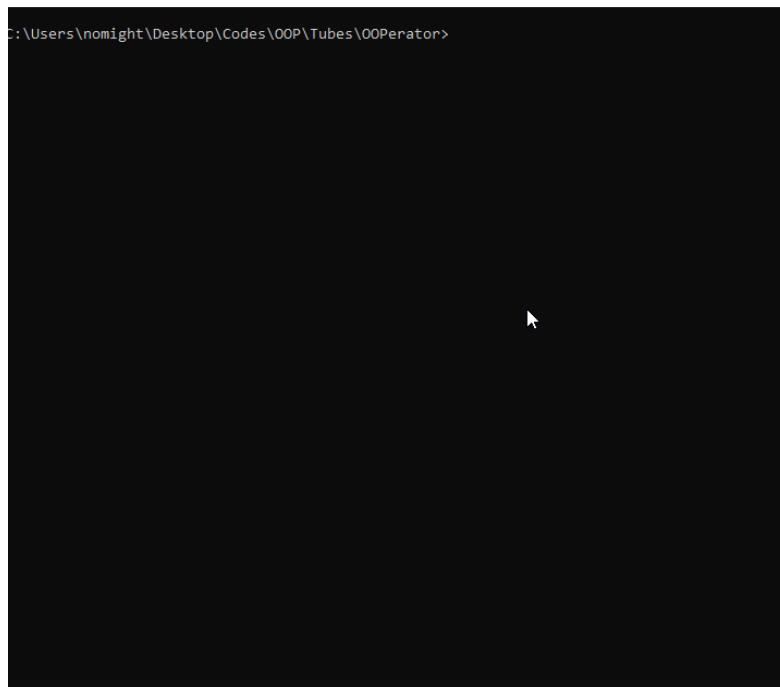
cUT

Untuk menjalankan unit test untuk semua modul, maka jalankan run Unit Test (rUT.bat)

rUT

jika, ingin menjalankan unit test untuk suatu modul yang diinginkan, maka jalankan script rUT dengan memberikan argumen berupa nama modul yang ingin dicoba Sebagai contoh

rUT Divide



Anda juga dapat memberikan lebih dari satu argumen untuk menguji lebih dari satu modul

rUT Add Divide

Mekanisme test untuk modul-modul Expression adalah sebagai berikut: akan diperiksa method solve() dari tiap ekspresi operator Sebagai contoh:

```
AddExpression      a(new      TerminalExpression(1),      new
TerminalExpression(3)))
REQUIRE(a.solve() == 3);
```

Untuk mempersingkat kode, make telah dibuat method public yang ada di class Parser

// Private method, digunakan untuk mengevaluasi string ekspresi secara keseluruhan

```
Expression* Parser::evalUnaryOp(Expression* a, string unaryOp) {
    Expression* ret;
    if (unaryOp == "sin") {
        ret = new SinExpression(a);
    } else if (unaryOp == "cos") {
        ret = new CosExpression(a);
    } else if (unaryOp == "tan") {
        ret = new TanExpression(a);
    } else if (unaryOp == "sqrt") {
        ret = new SqrtExpression(a);
    }
    return ret;
}
```

// Public method, digunakan oleh unit test

```
double Parser::evalUnaryOp(double a, string unaryOp) {
```



```

    return evalUnaryOp(new TerminalExpression(a), unaryOp)->solve();
}

```

// Private method, digunakan untuk mengevaluasi string ekspresi secara keseluruhan

```

Expression* Parser::evalBinaryOp(Expression* a, Expression* b, string
binaryOp) {
    Expression* ret;
    if (binaryOp == "+") {
        ret = new AddExpression(a, b);
    } else if (binaryOp == "-") {
        ret = new SubtractExpression(a, b);
    } else if (binaryOp == "*") {
        ret = new MultiplyExpression(a, b);
    } else if (binaryOp == "/") {
        ret = new DivideExpression(a, b);
    } else if (binaryOp == "^") {
        ret = new PowExpression(a, b);
    } else if (binaryOp == "mod") {
        ret = new ModExpression(a, b);
    }
    return ret;
}

```

// Public method, digunakan oleh unit test

```

double Parser::evalBinaryOp(double a, double b, string binaryOp) {
    return      evalBinaryOp(new      TerminalExpression(a),      new
TerminalExpression(b), binaryOp)->solve();
}

```

Berikut adalah list modul-modul yang diujikan

- Add
- Substract
- Divide
- Multiply
- Pow
- Mod
- Sin
- Tan
- Cos
- Sqrt
- Queue
- Parser

Documentation

Dokumentasi dapat diakses di <docs/html/index.html>, dokumentasi ini dibuat menggunakan Doxygen.

Project Structure

Berikut adalah struktur folder dari proyek ini

```
+---bin
+---deploy
+---docs
|  +---html
|  +---latex
|
\---src
```

```
| main.cpp
| OOperator.pro
| OOperator.pro.user
|
+---Calculator
|   Calculator.cpp
|   Calculator.hpp
|   calculator.ui
|
+---Exception
|   BaseException.hpp
|   DivideWithZeroException.hpp
|   Exception.hpp
|   ImbalancedParanthesesException.hpp
|   InvalidDecimalException.hpp
|   ModException.hpp
|   MoreThanOneOperatorException.hpp
|   NegativeSqrtException.hpp
|   NoParanthesisInUnaryException.hpp
|   UndefinedOperatorException.hpp
|
+---Expression
| | Expression.cpp
| | Expression.hpp
| | TerminalExpression.cpp
| | TerminalExpression.hpp
| |
| +---Binary
| |   AddExpression.cpp
| |   AddExpression.hpp
```

```
| | BinaryExpression.cpp
| | BinaryExpression.hpp
| | DivideExpression.cpp
| | DivideExpression.hpp
| | ModExpression.cpp
| | ModExpression.hpp
| | MultiplyExpression.cpp
| | MultiplyExpression.hpp
| | PowExpression.cpp
| | PowExpression.hpp
| | SubtractExpression.cpp
| | SubtractExpression.hpp
| |
| +---Other
| | ExpressionMain.cpp
| |
| \---Unary
|   CosExpression.cpp
|   CosExpression.hpp
|   NegativeExpression.cpp
|   NegativeExpression.hpp
|   SinExpression.cpp
|   SinExpression.hpp
|   SqrtExpression.cpp
|   SqrtExpression.hpp
|   TanExpression.cpp
|   TanExpression.hpp
|   UnaryExpression.cpp
|   UnaryExpression.hpp
|
```

```
+---Parser
|   Parser.cpp
|   Parser.hpp
|
\---Test
    AddTest.cpp
    CosTest.cpp
    DivideTest.cpp
    MainTest.cpp
    ModTest.cpp
    MultiplyTest.cpp
    NegativeTest.cpp
    ParserTest.cpp
    PowTest.cpp
    QueueTest.cpp
    SinTest.cpp
    SqrtTest.cpp
    SubtractTest.cpp
    TanTest.cpp
    all.hpp
    catch.hpp
```

Built With

- C++ - Bahasa pemrograman
- Qt - GUI toolkit
- Catch2 - Unit test framework
- Doxygen - Documentation generator

Authors

Bukan Kalkulator Scientific				
0				
sin	cos	tan	()
AC	MC	MR	/	⌫
7	8	9	*	sqrt
4	5	6	-	^
1	2	3	+	mod
CLRSCR	0	.	ans	=

Fitur-fitur dasar dan tambahan tidak terlepas dari bug, terutama fitur-fitur yang merupakan spesifikasi *bonus*, oleh karena itu kami minta maaf atas ketidak sempurnaan program.