

# Dungeon Generator (BSP Tree & recursive bactracking) c++

1.00.0

Dibangkitkan oleh Doxygen 1.8.14



# Daftar Isi

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>dungeon generator (BSP Tree &amp; recursive backtracking)</b> | <b>1</b> |
| <b>2</b> | <b>Indeks Kelas</b>  | <b>3</b> |
| 2.1      | Daftar Kelas . . . . .   | 3        |
| <b>3</b> | <b>Indeks File</b>   | <b>5</b> |
| 3.1      | Daftar File . . . . .  | 5        |
| <b>4</b> | <b>Dokumentasi Kelas</b>   | <b>7</b> |
| 4.1      | Referensi Kelas BSPTree . . . . .                                | 7        |
| 4.1.1    | Keterangan Lengkap . . . . .                                     | 8        |
| 4.1.2    | Dokumentasi Konstruktor & Destruktor . . . . .                   | 8        |
| 4.1.2.1  | BSPTree() . . . . .  | 8        |
| 4.1.2.2  | ~BSPTree() . . . . .   | 8        |
| 4.1.3    | Dokumentasi Anggota: Fungsi . . . . .                            | 8        |
| 4.1.3.1  | addLeaf() . . . . .  | 8        |
| 4.1.3.2  | getListLeaf() . . . . .  | 9        |
| 4.1.3.3  | getListLeafValue() . . . . .                                     | 9        |
| 4.1.3.4  | getListOrder() . . . . .   | 10       |
| 4.1.3.5  | getListOrderValue() . . . . .                                    | 10       |
| 4.1.3.6  | preOrder() . . . . .   | 11       |
| 4.1.3.7  | printInOrder() . . . . .   | 11       |
| 4.1.3.8  | split() . . . . .  | 11       |
| 4.2      | Referensi Kelas cell . . . . .                                   | 12       |
| 4.2.1    | Keterangan Lengkap . . . . .                                     | 13       |

|          |  |    |
|----------|--|----|
| 4.2.2    | Dokumentasi Konstruktor & Destruktor . . . . . | 13 |
| 4.2.2.1  | cell() . . . . .                               | 13 |
| 4.2.2.2  | ~cell() . . . . .                              | 13 |
| 4.2.3    | Dokumentasi Anggota: Data . . . . .            | 13 |
| 4.2.3.1  | color . . . . .                                | 13 |
| 4.2.3.2  | display . . . . .                              | 14 |
| 4.2.3.3  | value . . . . .                                | 14 |
| 4.2.3.4  | visited . . . . .                              | 14 |
| 4.2.3.5  | x . . . . .                                    | 14 |
| 4.2.3.6  | y . . . . .                                    | 14 |
| 4.3      | Referensi Kelas dungeon . . . . .              | 15 |
| 4.3.1    | Keterangan Lengkap . . . . .                   | 16 |
| 4.3.2    | Dokumentasi Konstruktor & Destruktor . . . . . | 16 |
| 4.3.2.1  | dungeon() . . . . .                            | 16 |
| 4.3.2.2  | ~dungeon() . . . . .                           | 16 |
| 4.3.3    | Dokumentasi Anggota: Fungsi . . . . .          | 17 |
| 4.3.3.1  | execute() . . . . .                            | 17 |
| 4.3.3.2  | save() . . . . .                               | 17 |
| 4.3.3.3  | show() . . . . .                               | 18 |
| 4.3.4    | Dokumentasi Anggota: Data . . . . .            | 19 |
| 4.3.4.1  | bsptree . . . . .                              | 19 |
| 4.3.4.2  | inOut . . . . .                                | 19 |
| 4.3.4.3  | lebar . . . . .                                | 19 |
| 4.3.4.4  | listRuanganManual . . . . .                    | 19 |
| 4.3.4.5  | maxLebar . . . . .                             | 20 |
| 4.3.4.6  | maxPanjang . . . . .                           | 20 |
| 4.3.4.7  | minLebar . . . . .                             | 20 |
| 4.3.4.8  | minPanjang . . . . .                           | 20 |
| 4.3.4.9  | panjang . . . . .                              | 20 |
| 4.3.4.10 | proses . . . . .                               | 20 |

|          |  |    |
|----------|--|----|
| 4.4      | Referensi Kelas gridm . . . . .                | 21 |
| 4.4.1    | Keterangan Lengkap . . . . .                   | 22 |
| 4.4.2    | Dokumentasi Konstruktor & Destruktor . . . . . | 22 |
| 4.4.2.1  | gridm() [1/2] . . . . .                        | 22 |
| 4.4.2.2  | gridm() [2/2] . . . . .                        | 23 |
| 4.4.2.3  | ~gridm() . . . . .                             | 23 |
| 4.4.3    | Dokumentasi Anggota: Fungsi . . . . .          | 23 |
| 4.4.3.1  | clearing() . . . . .                           | 23 |
| 4.4.3.2  | get_height() . . . . .                         | 24 |
| 4.4.3.3  | get_width() . . . . .                          | 24 |
| 4.4.3.4  | getCell() . . . . .                            | 25 |
| 4.4.3.5  | getColor() . . . . .                           | 25 |
| 4.4.3.6  | getDisplay() . . . . .                         | 26 |
| 4.4.3.7  | getValue() . . . . .                           | 26 |
| 4.4.3.8  | getVisited() . . . . .                         | 26 |
| 4.4.3.9  | printing() . . . . .                           | 28 |
| 4.4.3.10 | resize() . . . . .                             | 28 |
| 4.4.3.11 | save() . . . . .                               | 29 |
| 4.4.3.12 | set_region() . . . . .                         | 29 |
| 4.4.3.13 | set_region_color() . . . . .                   | 30 |
| 4.4.3.14 | set_region_display() . . . . .                 | 31 |
| 4.4.3.15 | set_region_visited() . . . . .                 | 31 |
| 4.4.3.16 | setCell() . . . . .                            | 32 |
| 4.4.3.17 | setColor() . . . . .                           | 32 |
| 4.4.3.18 | setDisplay() . . . . .                         | 33 |
| 4.4.3.19 | setValue() . . . . .                           | 34 |
| 4.4.3.20 | setVisited() . . . . .                         | 34 |
| 4.4.4    | Dokumentasi Anggota: Data . . . . .            | 35 |
| 4.4.4.1  | map . . . . .                                  | 35 |
| 4.5      | Referensi Kelas labyrinth . . . . .            | 35 |

|         |  |    |
|---------|--|----|
| 4.5.1   | Keterangan Lengkap . . . . .                   | 36 |
| 4.5.2   | Dokumentasi Konstruktor & Destruktor . . . . . | 36 |
| 4.5.2.1 | labyrinth() . . . . .                          | 36 |
| 4.5.2.2 | ~labyrinth() . . . . .                         | 37 |
| 4.5.3   | Dokumentasi Anggota: Fungsi . . . . .          | 37 |
| 4.5.3.1 | getMaze() . . . . .                            | 37 |
| 4.6     | Referensi Kelas rectangle . . . . .            | 38 |
| 4.6.1   | Keterangan Lengkap . . . . .                   | 39 |
| 4.6.2   | Dokumentasi Konstruktor & Destruktor . . . . . | 39 |
| 4.6.2.1 | rectangle() [1/2] . . . . .                    | 39 |
| 4.6.2.2 | rectangle() [2/2] . . . . .                    | 39 |
| 4.6.2.3 | ~rectangle() . . . . .                         | 40 |
| 4.6.3   | Dokumentasi Anggota: Fungsi . . . . .          | 40 |
| 4.6.3.1 | midX() . . . . .                               | 40 |
| 4.6.3.2 | midY() . . . . .                               | 41 |
| 4.6.3.3 | randomRange() . . . . .                        | 41 |
| 4.6.3.4 | reDefine() . . . . .                           | 41 |
| 4.6.4   | Dokumentasi Anggota: Data . . . . .            | 43 |
| 4.6.4.1 | color . . . . .                                | 43 |
| 4.6.4.2 | height . . . . .                               | 43 |
| 4.6.4.3 | large . . . . .                                | 44 |
| 4.6.4.4 | width . . . . .                                | 44 |
| 4.6.4.5 | x . . . . .                                    | 44 |
| 4.6.4.6 | y . . . . .                                    | 44 |

|   |           |
|---|-----------|
| <b>5 Dokumentasi File</b>   | <b>45</b> |
| 5.1 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/BSPTree.cpp .    | 45        |
| 5.2 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/BSPTree.h . .    | 45        |
| 5.3 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/cell.cpp . . . . | 46        |
| 5.4 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/cell.h . . . . . | 47        |
| 5.4.1 Keterangan Lengkap . . . . .  | 48        |
| 5.5 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/dungeon.cpp .    | 48        |
| 5.6 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/dungeon.h . .    | 49        |
| 5.6.1 Keterangan Lengkap . . . . .  | 50        |
| 5.7 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/example.cpp .    | 50        |
| 5.7.1 Dokumentasi Fungsi . . . . .  | 51        |
| 5.7.1.1 main() . . . . .  | 51        |
| 5.8 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/gridm.cpp . .    | 52        |
| 5.9 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/gridm.h . . . .  | 52        |
| 5.9.1 Keterangan Lengkap . . . . .  | 53        |
| 5.10 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/labyrinth.cpp . | 53        |
| 5.11 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/labyrinth.h . . | 54        |
| 5.11.1 Dokumentasi Makro Definisi . . . . .   | 55        |
| 5.11.1.1 down . . . . .   | 55        |
| 5.11.1.2 left . . . . .   | 55        |
| 5.11.1.3 right . . . . .  | 56        |
| 5.11.1.4 up . . . . .   | 56        |
| 5.12 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/rectangle.cpp   | 56        |
| 5.13 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/rectangle.h . . | 56        |
| 5.13.1 Keterangan Lengkap . . . . .   | 57        |
| <b>Indeks</b>   | <b>59</b> |





## Bab 1

# dungeon generator (BSP Tree & recursive backtracking)

Class untuk membuat dungeon secara otomatis

dungeon yang di desain otomatis menggunakan penggabungan algoritma BSP Tree dan Recursive Backtracker. yang dimana BSP Tree berfungsi untuk membuat ruangan dalam dungeon (ruangan juga dapat dibuat secara manual) .dan Recursive Backtracker (atau juga dikenal sebagai Deep First Search) untuk membuat labirin nya.



## Bab 2

# Indeks Kelas

### 2.1 Daftar Kelas

Berikut ini daftar kelas, struct, union, dan interface, dengan penjelasan singkat:

|                  |  |    |
|------------------|--|----|
| <b>BSPTree</b>   | Merupakan objek berbentuk binary tree yang dimana nilai child-nya adalah hasil pembagian dari nilai parent-nya . . . . .                 | 7  |
| <b>cell</b>      | Merupakan objek berupa <b>cell(layaknya ruang di lemari)</b> yang menyimpan beberapa attribut . .  | 12 |
| <b>dungeon</b>   | Merupakan satu satunya objek yang di #include untuk membuat dungeon menggunakan fungsi dan mengatur attributnya . . . . .                | 15 |
| <b>gridm</b>     | Merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat x,y seperti cell pada microsof excel . . . . . | 21 |
| <b>labyrinth</b> | Merupakan objek maze(labirin) yang dibuat di dalam grid map dengan menggunakan algoritma deep first search . . . . .                     | 35 |
| <b>rectangle</b> | Merupakan objek berbentuk kotak . . . . .  | 38 |



## Bab 3

# Indeks File

### 3.1 Daftar File

Berikut ini daftar seluruh file, dengan penjelasan singkat:

|   |    |
|---|----|
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/BSPTree.cpp . . . . .   | 45 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/BSPTree.h . . . . .   | 45 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/cell.cpp . . . . .  | 46 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/cell.h  |    |
| Merupakan objek berupa <a href="#">cell(layaknya ruang di lemari)</a> yang menyimpan beberapa attribut . .                                    | 47 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/dungeon.cpp . . . . .   | 48 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/dungeon.h   |    |
| Merupakan objek untuk menggenerate duneon dalam bentuk matriks char . . . . .   | 49 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/example.cpp . . . . .   | 50 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/gridm.cpp . . . . .   | 52 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/gridm.h   |    |
| Merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat<br>x,y seperti cell pada microsoft excell . . . . . | 52 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/labyrinth.cpp . . . . .   | 53 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/labyrinth.h . . . . .   | 54 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/rectangle.cpp . . . . .   | 56 |
| C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/rectangle.h   |    |
| Merupakan objek berbentuk kotak . . . . .   | 56 |



## Bab 4

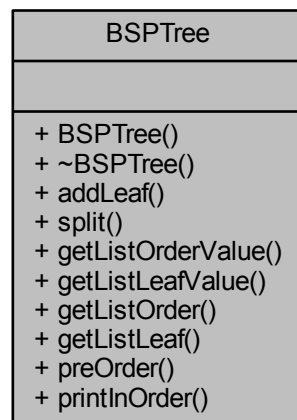
# Dokumentasi Kelas

### 4.1 Referensi Kelas BSPTree

merupakan objek berbentuk binary tree yang dimana nilai child-nya adalah hasil pembagian dari nilai parent-nya

```
#include <BSPTree.h>
```

Diagram kolaborasi untuk BSPTree:



### Fungsi Anggota Publik

- `BSPTree ()`  
*Construct a new BST::BST object.*
- virtual `~BSPTree ()`  
*Destroy the BST::BST object.*
- void `addLeaf (rectangle area)`

- prosedur emembuat daun baru dengan memanggil createLeaf*
- void `split` (int minWidth, int minHeight, int maxWidth, int maxHeight)
  - prosedur membagi data dari tree menggunakan fungsi splitPrivate dengan ketentuan jika melebihi ukuran max nya maka dibagi dan hanya area yang di atas ukuran minimalnya yang akan di jadikan ruangan*
- vector< `rectangle` > `getListOrderValue` ()
  - mendapatkan nilai data pohon secara preorder dengan memanggil fungsi `getListOrder()`*
- vector< `rectangle` > `getListLeafValue` ()
  - fungsi untuk mendapatkan nilai data dari daun terbawah dari pohon*
- void `getListOrder` ()
  - memasukkan data pohon secara preorder dengan memanggil fungsi `getListOrderPrivate()` kedalam var. `listOrder`*
- void `getListLeaf` ()
  - fungsi untuk memuat data daun terbawah dari pohon dengan memanggil fungsi `getListLeaf()`*
- void `preOrder` ()
  - prosedur untuk mencetak data secara pre order dengan memanggil `preOrderPrivate`*
- void `printlnOrder` ()
  - prosedur untuk mencetak data secara inOrder dengan memanggil `printlnOrderPrivate()`*

#### 4.1.1 Keterangan Lengkap

merupakan objek berbentuk binary tree yang dimana nilai child-nya adalah hasil pembagian dari nilai parent-nya

#### 4.1.2 Dokumentasi Konstruktor & Destruktor

##### 4.1.2.1 BSPTree()

```
BSPTree::BSPTree ( )
```

Construct a new BST::BST object.

##### 4.1.2.2 ~BSPTree()

```
BSPTree::~~BSPTree ( ) [virtual]
```

Destroy the BST::BST object.

Lihat juga

```
removeSubTree()
```

#### 4.1.3 Dokumentasi Anggota: Fungsi

##### 4.1.3.1 addLeaf()

```
void BSPTree::addLeaf (
    rectangle area )
```

prosedur emembuat daun baru dengan memanggil createLeaf



**Parameter**

|             |   |
|-------------|---|
| <i>area</i> | data rectangle yang akan disimpan di simpul pohon(daun) |
|-------------|---|

**Lihat juga**

`createLeaf()`

**4.1.3.2 `getListLeaf()`**

```
void BSPTree::getListLeaf ( )
```

fungsi untuk memuat data daun terbawah dari pohon dengan memanggil fungsi [getListLeaf\(\)](#)

**Lihat juga**

`getListLeafPrivate()`

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:

**4.1.3.3 `getListLeafValue()`**

```
vector< rectangle > BSPTree::getListLeafValue ( )
```

fungsi untuk mendapatkan nilai data dari daun terbawah dari pohon

**Mengembalikan**

`vector<rectangle>`

Lihat juga

[getListLeaf\(\)](#)

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.1.3.4 `getListOrder()`

```
void BSPTree::getListOrder ( )
```

memasukkan data pohon secara preorder dengan memanggil fungsi `getListOrderPrivate()` kedalam var. `listOrder`

Lihat juga

`getListOrderPrivate()`

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.1.3.5 `getListOrderValue()`

```
vector< rectangle > BSPTree::getListOrderValue ( )
```

mendapatkan nilai data pohon secara preorder dengan memanggil fungsi [getListOrder\(\)](#)

Mengembalikan

`vector<rectangle>`

Lihat juga

[getListOrder\(\)](#)

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.1.3.6 preOrder()

```
void BSPTree::preOrder ( )
```

prosedur untuk mencetak data secara pre order dengan memanggil preOrderPrivate

Lihat juga

preOrderPrivate()pre order untuk keperluan debug

#### 4.1.3.7 printInOrder()

```
void BSPTree::printInOrder ( )
```

prosedur untuk mencetak data secara inOrder dengan memanggil printInOrderPrivate()

Lihat juga

printInOrderPrivate()

#### 4.1.3.8 split()

```
void BSPTree::split (
    int minWidth,
    int minHeight,
    int maxWidth,
    int maxHeight )
```

prosedur membagi data dari tree menggunakan fungsi splitPrivate dengan ketentuan jika melebihi ukuran max nya maka dibagi dan hanya area yang di atas ukuran minimalnya yang akan di jadikan ruangan

**Parameter**

|                  |                        |
|------------------|------------------------|
| <i>minWidth</i>  | [int] lebar minimal    |
| <i>minHeight</i> | [int] panjang minimal  |
| <i>maxWidth</i>  | [int] lebar maksimal   |
| <i>maxHeight</i> | [int] panjang maksimal |

**Lihat juga**

splitPrivate()

Dokumentasi untuk kelas ini dibangkitkan dari file-file berikut:

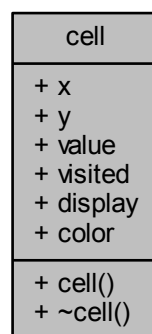
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[BSPTree.h](#)
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[BSPTree.cpp](#)

**4.2 Referensi Kelas cell**

merupakan objek berupa [cell](#)(*layaknya ruang di lemari*) yang menyimpan beberapa attribut

```
#include <cell.h>
```

Diagram kolaborasi untuk cell:

**Fungsi Anggota Publik**

- [cell](#) ()  
*Construct a new [cell::cell](#) object.*
- virtual [~cell](#) ()  
*Destroy the [cell::cell](#) object.*

### Atribut Publik

- int `x`  
*koordinat x objek cell*
- int `y`  
*koordinat y objek cell*
- int `value`  
*nilai yang disimpan objek cell*
- bool `visited`  
*menyatakan apakah cell sudah di kunjungi atau belum*
- string `display`  
*nilai yang dimunculkan cell di output (std::cout)*
- int `color`  
*warna tampilan dari variable display*

#### 4.2.1 Keterangan Lengkap

merupakan objek berupa `cell`(*layaknya ruang di lemari*) yang menyimpan beberapa attribut

#### 4.2.2 Dokumentasi Konstruktor & Destruktor

##### 4.2.2.1 `cell()`

```
cell::cell ( )
```

Construct a new `cell::cell` object.

##### 4.2.2.2 `~cell()`

```
cell::~~cell ( ) [virtual]
```

Destroy the `cell::cell` object.

#### 4.2.3 Dokumentasi Anggota: Data

##### 4.2.3.1 `color`

```
int cell::color
```

warna tampilan dari variable display

#### 4.2.3.2 display

```
string cell::display
```

nilai yang dimunculkan cell di output (std::cout)

#### 4.2.3.3 value

```
int cell::value
```

nilai yang disimpan objek cell

#### 4.2.3.4 visited

```
bool cell::visited
```

menyatakan apakah cell sudah di kunjungi atau belum

#### 4.2.3.5 x

```
int cell::x
```

koordinat x objek cell

#### 4.2.3.6 y

```
int cell::y
```

koordinat y objek cell

Dokumentasi untuk kelas ini dibangkitkan dari file-file berikut:

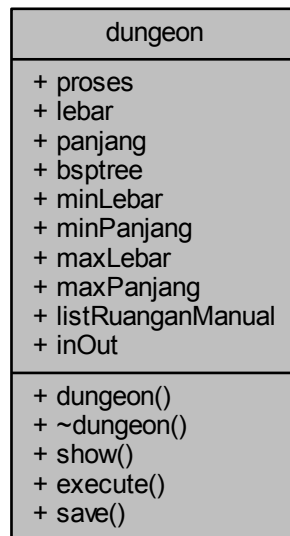
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[cell.h](#)
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[cell.cpp](#)

## 4.3 Referensi Kelas dungeon

merupakan satu satunya objek yang di #include untuk membuat dungeon menggunakan fungsi dan mengatur attributnya.

```
#include <dungeon.h>
```

Diagram kolaborasi untuk dungeon:



### Fungsi Anggota Publik

- `dungeon ()`  
*Construct a new dungeon object.*
- virtual `~dungeon ()`  
*Destroy the dungeon object.*
- void `show ()`  
*fungsi untuk menampilkan hasil dungeon yang dibuat*
- void `execute ()`  
*mfungsi untuk men generate ulang dungeon dengan nilai attribut yang telah dirubah*
- void `save (string name)`  
*fungsi untuk menyimpandungeon ke dalam file .txt*

### Atribut Publik

- bool `proses`  
*true untuk melihat proses pembuatan*
- int `lebar`
- int `panjang`

- lebar dan panjang dungeon
- bool `bsptree`  
true jika ingin membuat ruangan dengan algoritma *BSPTree*
- int `minLebar`
- int `minPanjang`
- int `maxLebar`
- int `maxPanjang`  
parameter yang digunakan oleh algoritma *BSPTree*
- vector< `rectangle` > `listRuanganManual`  
list yang berisikan definisi ruangan yang di inisiasi secara manual
- bool `inOut`  
bernilai true jika menginginkan dungeon dengan jalan masuk dan keluar

### 4.3.1 Keterangan Lengkap

merupakan satu satunya objek yang di #include untuk membuat dungeon menggunakan fungsi dan mengatur attributnya.

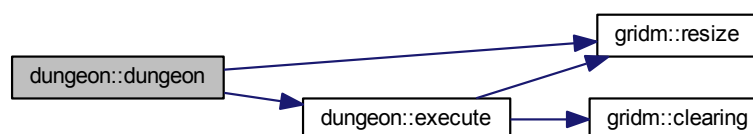
### 4.3.2 Dokumentasi Konstruktor & Destruktor

#### 4.3.2.1 `dungeon()`

```
dungeon::dungeon ( )
```

Construct a new dungeon object.

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.3.2.2 `~dungeon()`

```
dungeon::~~dungeon ( ) [virtual]
```

Destroy the dungeon object.



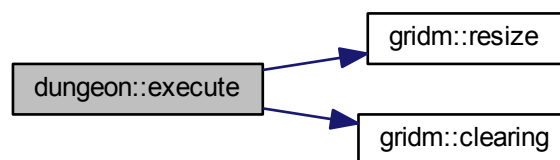
### 4.3.3 Dokumentasi Anggota: Fungsi

#### 4.3.3.1 execute()

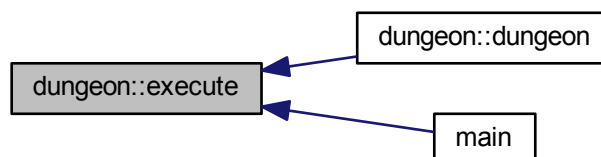
```
void dungeon::execute ( )
```

mfungsi untuk men generate ulang dungeon dengan nilai attribut yang telah dirubah

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.3.3.2 save()

```
void dungeon::save (
    string name )
```

fungsi untuk menyimpandungeon ke dalam file .txt

**Parameter**

|             |                             |
|-------------|-----------------------------|
| <i>name</i> | : nama file yang diinginkan |
|-------------|-----------------------------|

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



Berikut ini bagan fungsi-pemanggil untuk fungsi ini:

**4.3.3.3 show()**

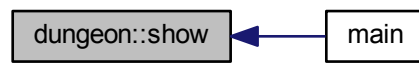
```
void dungeon::show ( )
```

fungsi untuk menampilkan hasil dungeon yang dibuat

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.3.4 Dokumentasi Anggota: Data

##### 4.3.4.1 bsptree

```
bool dungeon::bsptree
```

true jika ingin membuat ruangan dengan algoritma [BSPTree](#)

##### 4.3.4.2 inOut

```
bool dungeon::inOut
```

bernilai true jika menginginkan dungeon dengan jalan masuk dan keluar

##### 4.3.4.3 lebar

```
int dungeon::lebar
```

##### 4.3.4.4 listRuanganManual

```
vector<rectangle> dungeon::listRuanganManual
```

list yang berisikan definisi ruangan yang di inisiasi secara manual

#### 4.3.4.5 maxLebar

```
int dungeon::maxLebar
```

#### 4.3.4.6 maxPanjang

```
int dungeon::maxPanjang
```

parameter yang digunakan oleh algoritma [BSPTree](#)

#### 4.3.4.7 minLebar

```
int dungeon::minLebar
```

#### 4.3.4.8 minPanjang

```
int dungeon::minPanjang
```

#### 4.3.4.9 panjang

```
int dungeon::panjang
```

lebar dan panjang dungeon

#### 4.3.4.10 proses

```
bool dungeon::proses
```

true untuk melihat proses pembuatan

Dokumentasi untuk kelas ini dibangkitkan dari file-file berikut:

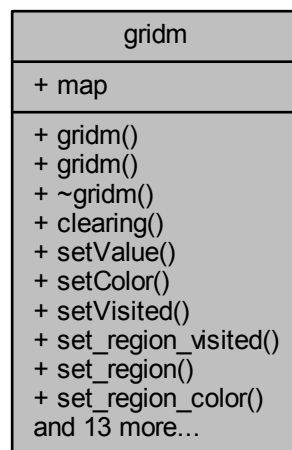
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[dungeon.h](#)
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[dungeon.cpp](#)

## 4.4 Referensi Kelas gridm

merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat x,y seperti cell pada microsofst excell

```
#include <gridm.h>
```

Diagram kolaborasi untuk gridm:



### Fungsi Anggota Publik

- `gridm ()`  
*Construct a new `gridm::gridm` object.*
- `gridm (int width, int height)`  
*Construct a new `gridm::gridm` object membuat gridmap berisikan cell. sebanyak lebar \* panjang = banyaknya cell.*
- virtual `~gridm ()`  
*Destroy the `gridm::gridm` object.*
- void `clearing (int value)`  
*menginisiasi nilai setiap cell pada grid map*
- void `setValue (int x, int y, int value)`  
*setter : untuk menentukan value(nilai) cell pada koordinat x,y*
- void `setColor (int x, int y, int color)`  
*setter : untuk menentukan color(warna) cell pada koordinat x,y*
- void `setVisited (int x, int y, bool visited)`  
*setter : untuk menentukan apakah suatu cell sudah dikunjungi pada koordinat x,y*
- void `set_region_visited (int x1, int y1, int x2, int y2, bool value)`  
*setter : untuk menentukan sekumpulan cell sudah dikunjungi atau belum pada koordinat x,y awal hingga x,y akhir*
- void `set_region (int x1, int y1, int x2, int y2, int value)`  
*setter : untuk menentukan value(nilai) sekumpulan cell pada koordinat x,y awal hingga x,y akhir*
- void `set_region_color (int x1, int y1, int x2, int y2, int color)`

- setter : untuk menentukan color(warna) sekumpulan cell pada koordinat x,y awal hingga x,y akhir
- void `setCell` (int x, int y, `cell` Cell)
  - setter : untuk menentukan objek cell pada koordinat x,y
- void `setDisplay` (int x, int y, string value)
  - setter : untuk menentukan display(string yang ditampilkan) cell pada koordinat x,y
- void `set_region_display` (int x1, int y1, int x2, int y2, string value)
  - setter : untuk menentukan display(string yang ditampilkan) cell pada koordinat x,y awal hingga x,y akhir
- void `resize` (int height, int width)
  - prosedur untuk mengubah panjang dan lebar grid map
- int `get_width` ()
  - getter : untuk mendapatkan nilai width(lebar) grid map
- int `get_height` ()
  - getter : untuk mendapatkan nilai height(panjang) grid map
- int `getValue` (int x, int y)
  - getter : untuk mendapatkan nilai cell pada koordinat x,y
- int `getColor` (int x, int y)
  - getter : untuk mendapatkan color(warna) cell pada koordinat x,y
- bool `getVisited` (int x, int y)
  - getter : untuk mengetahui apakah cell pada koordinat x,y sudah dikunjungi
- `cell` `getCell` (int x, int y)
  - getter : untuk mendapatkan cell pada koordinat x,y
- string `getDisplay` (int x, int y)
  - getter : untuk mendapatkan display(string yang ditampilkan) cell pada koordinat x,y
- void `printing` ()
  - untuk mencetak(menampilkan) seluruh cell pada grid map
- void `save` (string name)
  - untuk menyimpan seluruh cell pada grid map

## Atribut Publik

- `vector< vector< cell > > map`
  - list 2 dimensi yang berisikan cell

### 4.4.1 Keterangan Lengkap

merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat x,y seperti cell pada microsoft excel

### 4.4.2 Dokumentasi Konstruktor & Destruktor

#### 4.4.2.1 `gridm()` [1/2]

```
gridm::gridm ( )
```

Construct a new `gridm::gridm` object.

## 4.4.2.2 gridm() [2/2]

```
gridm::gridm (
    int width,
    int height )
```

Construct a new `gridm::gridm` object membuat gridmap berisikan cell. sebanyak lebar \* panjang = banyaknya cell.

## Parameter

|               |                   |
|---------------|-------------------|
| <i>width</i>  | [int] lebar map   |
| <i>height</i> | [int] panjang map |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



## 4.4.2.3 ~gridm()

```
gridm::~~gridm ( ) [virtual]
```

Destroy the `gridm::gridm` object.

## 4.4.3 Dokumentasi Anggota: Fungsi

## 4.4.3.1 clearing()

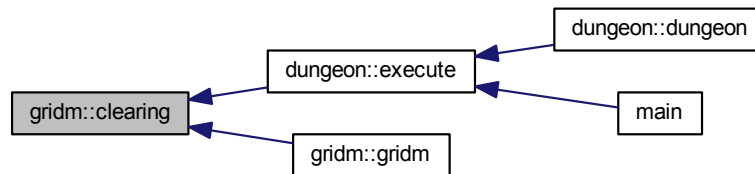
```
void gridm::clearing (
    int value )
```

menginisiasi nilai setiap cell pada grid map

## Parameter

|              |  |
|--------------|--|
| <i>value</i> | [int] nilai inisiasi untuk setiap cell |
|--------------|--|

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.3.2 `get_height()`

```
int gridm::get_height ( )
```

getter : untuk mendapatkan nilai height(panjang) grid map

**Mengembalikan**

int

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.3.3 `get_width()`

```
int gridm::get_width ( )
```

getter : untuk mendapatkan nilai width(lebar) grid map



Mengembalikan

int

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.3.4 getCell()

```
cell gridm::getCell (
    int x,
    int y )
```

getter : untuk mendapatkan cell pada koordinat x,y

Parameter

|   |                   |
|---|-------------------|
| x | [int] koordinat x |
| y | [int] koordinat y |

Mengembalikan

cell

#### 4.4.3.5 getColor()

```
int gridm::getColor (
    int x,
    int y )
```

getter : untuk mendapatkan color(warna) cell pada koordinat x,y

Parameter

|   |                   |
|---|-------------------|
| x | [int] koordinat x |
| y | [int] koordinat y |

**Mengembalikan**

int

**4.4.3.6 getDisplay()**

```
string gridm::getDisplay (
    int x,
    int y )
```

getter : untuk mendapatkan display(string yang ditampilkan) cell pada koordinat x,y

**Parameter**

|   |                   |
|---|-------------------|
| x | [int] koordinat x |
| y | [int] koordinat y |

**Mengembalikan**

string

**4.4.3.7 getValue()**

```
int gridm::getValue (
    int x,
    int y )
```

getter : untuk mendapatkan nilai cell pada koordinat x,y

**Parameter**

|   |                   |
|---|-------------------|
| x | [int] koordinat x |
| y | [int] koordinat y |

**Mengembalikan**

int

**4.4.3.8 getVisited()**

```
bool gridm::getVisited (
    int x,
    int y )
```

getter : untuk mengetahui apakah cell pada koordinat x,y sudah dikunjungi

**Parameter**

|          |                   |
|----------|-------------------|
| <i>x</i> | [int] koordinat x |
| <i>y</i> | [int] koordinat y |

**Mengembalikan**

true jika sudah dikunjungi  
false jika belum dikunjungi

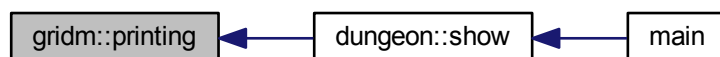
Berikut ini bagan fungsi-pemanggil untuk fungsi ini:

**4.4.3.9 printing()**

```
void gridm::printing ( )
```

untuk mencetak(menampilkan) seluruh cell pada grid map

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:

**4.4.3.10 resize()**

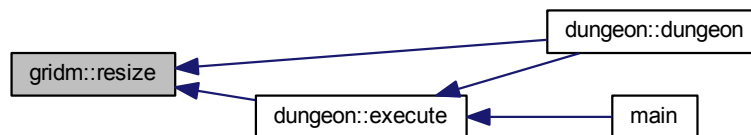
```
void gridm::resize (
    int height,
    int width )
```

prosedur untuk mengubah panjang dan lebar grid map

## Parameter

|               |                                  |
|---------------|----------------------------------|
| <i>width</i>  | [int] lebar grid map yang baru   |
| <i>height</i> | [int] panjang grid map yang baru |

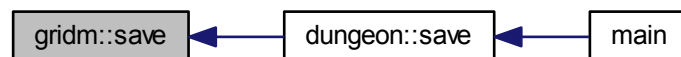
Berikut ini bagan fungsi-pemanggil untuk fungsi ini:

4.4.3.11 `save()`

```
void gridm::save (
    string name )
```

untuk menyimpan seluruh cell pada grid map

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:

4.4.3.12 `set_region()`

```
void gridm::set_region (
    int x1,
    int y1,
    int x2,
    int y2,
    int value )
```

setter : untuk menentukan value(nilai) sekumpulan cell pada koordinat x,y awal hingga x,y akhir

**Parameter**

|              |                                      |
|--------------|--------------------------------------|
| <i>x1</i>    | [int] koordinat x awal               |
| <i>y1</i>    | [int] koordinat y awal               |
| <i>x2</i>    | [int] koordinat x akhir              |
| <i>y2</i>    | [int] koordinat y akhir              |
| <i>value</i> | [int] nilai yang diberikan pada cell |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:

**4.4.3.13 set\_region\_color()**

```

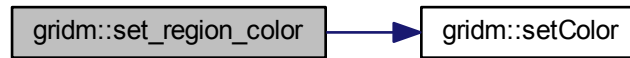
void gridm::set_region_color (
    int x1,
    int y1,
    int x2,
    int y2,
    int color )
  
```

setter : untuk menentukan color(warna) sekumpulan cell pada koordinat x,y awal hingga x,y akhir

**Parameter**

|              |   |
|--------------|---|
| <i>x1</i>    | [int] koordinat x awal                    |
| <i>y1</i>    | [int] koordinat y awal                    |
| <i>x2</i>    | [int] koordinat x akhir                   |
| <i>y2</i>    | [int] koordinat y akhir                   |
| <i>color</i> | [int] kode warna yang diberikan pada cell |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.4.3.14 set\_region\_display()

```

void gridm::set_region_display (
    int x1,
    int y1,
    int x2,
    int y2,
    string value )
  
```

setter : untuk menentukan display(string yang ditampilkan) cell pada koordinat x,y awal hingga x,y akhir

##### Parameter

|              |  |
|--------------|--|
| <i>x1</i>    | [int] koordinat x awal                                       |
| <i>y1</i>    | [int] koordinat y awal                                       |
| <i>x2</i>    | [int] koordinat x akhir                                      |
| <i>y2</i>    | [int] koordinat y akhir                                      |
| <i>value</i> | [string] string yang diberikan untuk cell pada koordinat x,y |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.4.3.15 set\_region\_visited()

```

void gridm::set_region_visited (
    int x1,
  
```

```

    int y1,
    int x2,
    int y2,
    bool value )

```

setter : untuk menentukan sekumpulan cell sudah dikunjungi atau belum pada koordinat x,y awal hingga x,y akhir

#### Parameter

|                |   |
|----------------|---|
| <i>x1</i>      | [int] koordinat x awal  |
| <i>y1</i>      | [int] koordinat y awal  |
| <i>x2</i>      | [int] koordinat x akhir   |
| <i>y2</i>      | [int] koordinat y akhir   |
| <i>visited</i> | [bool] boolean yang diberikan pada variable visited di cell x,y |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.4.3.16 setCell()

```

void gridm::setCell (
    int x,
    int y,
    cell Cell )

```

setter : untuk menentukan objek cell pada koordinat x,y

#### Parameter

|             |  |
|-------------|--|
| <i>x</i>    | [int] koordinat x                              |
| <i>y</i>    | [int] koordinat y                              |
| <i>Cell</i> | [cell] cell yang diberikan untuk koordinat x,y |

#### 4.4.3.17 setColor()

```

void gridm::setColor (
    int x,

```



```
int y,
int color )
```

setter : untuk menentukan color(warna) cell pada koordinat x,y

#### Parameter

|              |   |
|--------------|---|
| <i>x</i>     | [int] koordinat x                         |
| <i>y</i>     | [int] koordinat y                         |
| <i>color</i> | [int] kode warna yang diberikan pada cell |

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.3.18 setDisplay()

```
void gridm::setDisplay (
    int x,
    int y,
    string value )
```

setter : untuk menentukan display(string yang ditampilkan) cell pada koordinat x,y

#### Parameter

|              |  |
|--------------|--|
| <i>x</i>     | [int] koordinat x  |
| <i>y</i>     | [int] koordinat y  |
| <i>value</i> | [string] string yang diberikan untuk cell pada koordinat x,y |

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.3.19 setValue()

```
void gridm::setValue (
    int x,
    int y,
    int value )
```

setter : untuk menentukan value(nilai) cell pada koordinat x,y

##### Parameter

|              |                                      |
|--------------|--------------------------------------|
| <i>x</i>     | [int] koordinat x                    |
| <i>y</i>     | [int] koordinat y                    |
| <i>value</i> | [int] nilai yang diberikan pada cell |

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.3.20 setVisited()

```
void gridm::setVisited (
    int x,
    int y,
    bool visited )
```

setter : untuk menentukan apakah suatu cell sudah dikunjungi pada koordinat x,y

##### Parameter

|                |   |
|----------------|---|
| <i>x</i>       | [int] koordinat x   |
| <i>y</i>       | [int] koordinat y   |
| <i>visited</i> | [bool] boolean yang diberikan pada variable visited di cell x,y |

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.4.4 Dokumentasi Anggota: Data

##### 4.4.4.1 map

```
vector<vector<cell> > gridm::map
```

list 2 dimensi yang berisikan cell

Dokumentasi untuk kelas ini dibangkitkan dari file-file berikut:

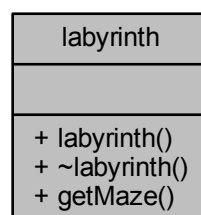
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[gridm.h](#)
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[gridm.cpp](#)

## 4.5 Referensi Kelas labyrinth

merupakan objek maze(labirin) yang dibuat di dalam grid map dengan menggunakan algoritma deep first search

```
#include <labyrinth.h>
```

Diagram kolaborasi untuk labyrinth:



## Fungsi Anggota Publik

- `labyrinth` (`gridm` `graph`, `bool` `progress`)  
*Construct a new labyrinth::labyrinth object membuat maze pada grid map dengan memanggil fungsi generate.*
- virtual `~labyrinth` ()  
*Destroy the labyrinth::labyrinth object.*
- `gridm` `getMaze` ()  
*merupakan getter untuk mendapatkan grid map yang terdapat maze di dalamnya*

### 4.5.1 Keterangan Lengkap

merupakan objek maze(labirin) yang dibuat di dalam grid map dengan menggunakan algoritma deep first search

### 4.5.2 Dokumentasi Konstruktor & Destruktor

#### 4.5.2.1 `labyrinth()`

```
labyrinth::labyrinth (
    gridm graph,
    bool progress )
```

Construct a new labyrinth::labyrinth object membuat maze pada grid map dengan memanggil fungsi generate.

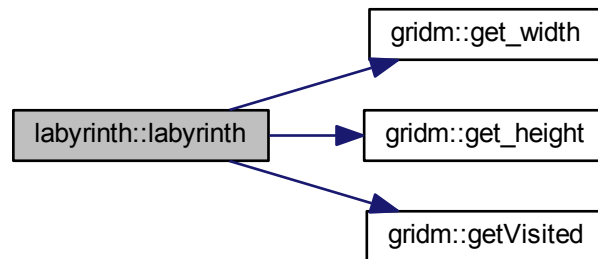
#### Parameter

|                 |  |
|-----------------|--|
| <i>graph</i>    | grid map tujuan yang akan di buat maze di dalamnya |
| <i>progress</i> | true jika ingin melihat proses pembuatan maze      |

#### Lihat juga

`generate()`

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



#### 4.5.2.2 ~labyrinth()

```
labyrinth::~~labyrinth ( ) [virtual]
```

Destroy the labyrinth::labyrinth object.

### 4.5.3 Dokumentasi Anggota: Fungsi

#### 4.5.3.1 getMaze()

```
gridm labyrinth::getMaze ( )
```

merupakan getter untuk mendapatkan grid map yang terdapat maze di dalamnya

#### Mengembalikan

gridmvariable yang mengembalikan hasil labyrinth

Dokumentasi untuk kelas ini dibangkitkan dari file-file berikut:

- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[labyrinth.h](#)
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[labyrinth.cpp](#)

## 4.6 Referensi Kelas rectangle

merupakan objek berbentuk kotak

```
#include <rectangle.h>
```

Diagram kolaborasi untuk rectangle:

| rectangle   |
|---|
| + x<br>+ y<br>+ width<br>+ height<br>+ color<br>+ large   |
| + rectangle()<br>+ rectangle()<br>+ ~rectangle()<br>+ midX()<br>+ midY()<br>+ randomRange()<br>+ reDefine() |

### Fungsi Anggota Publik

- `rectangle ()`  
*Construct a new rectangle object.*
- `rectangle (int x, int y, int width, int height)`  
*Construct a new `rectangle::rectangle` object membuat objek kotak.*
- `virtual ~rectangle ()`  
*Destroy the `rectangle::rectangle` object.*
- `int midX ()`  
*fungsi untuk mendapatkan posisi tengah horizontal*
- `int midY ()`  
*fungsi untuk mendapatkan posisi tengah vertical*
- `int randomRange (int start, int size_index)`  
*fungsi untuk mendapatkan angka secara acak*
- `void reDefine (int x, int y, int width, int height)`  
*menginisiasi ulang objek*

### Atribut Publik

- int `x`  
*koordinat x*
- int `y`  
*koordinat y*
- int `width`  
*lebar kotak*
- int `height`  
*tinggi kotak*
- int `color`  
*warna kotak*
- int `large`  
*luas kotak*

#### 4.6.1 Keterangan Lengkap

merupakan objek berbentuk kotak

#### 4.6.2 Dokumentasi Konstruktor & Destruktor

##### 4.6.2.1 `rectangle()` [1/2]

```
rectangle::rectangle ( )
```

Construct a new rectangle object.

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



##### 4.6.2.2 `rectangle()` [2/2]

```
rectangle::rectangle (
    int x,
    int y,
    int width,
    int height )
```

Construct a new `rectangle::rectangle` object membuat objek kotak.

**Parameter**

|               |                |
|---------------|----------------|
| <i>x</i>      | [int] posisi x |
| <i>y</i>      | [int] posisi y |
| <i>width</i>  | [int] lebar    |
| <i>height</i> | [int] tinggi   |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:

**4.6.2.3 ~rectangle()**

```
rectangle::~rectangle ( ) [virtual]
```

Destroy the `rectangle::rectangle` object.

**4.6.3 Dokumentasi Anggota: Fungsi****4.6.3.1 midX()**

```
int rectangle::midX ( )
```

fungsi untuk mendapatkan posisi tengah horizontal

**Mengembalikan**

int



#### 4.6.3.2 midY()

```
int rectangle::midY ( )
```

fungsi untuk mendapatkan posisi tengah vertical

Mengembalikan

int

#### 4.6.3.3 randomRange()

```
int rectangle::randomRange (
    int start,
    int size_index )
```

fungsi untuk mendapatkan angka secara acak

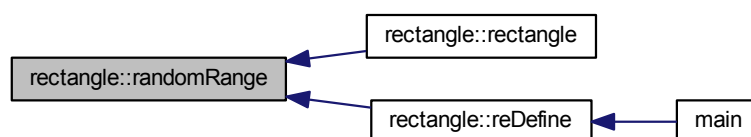
Parameter

|                   |                              |
|-------------------|------------------------------|
| <i>start</i>      | nilai minimum angka acak     |
| <i>size_index</i> | panjang jangkauan nilai acak |

Mengembalikan

int nilai acak antara start hingga start + size\_index

Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.6.3.4 reDefine()

```
void rectangle::reDefine (
    int x,
```

```
int y,  
int width,  
int height )
```

menginisiasi ulang objek

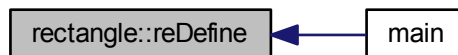
**Parameter**

|               |                |
|---------------|----------------|
| <i>x</i>      | [int] posisi x |
| <i>y</i>      | [int] posisi y |
| <i>width</i>  | [int] lebar    |
| <i>height</i> | [int] tinggi   |

Berikut ini bagan fungsi-terpanggil untuk fungsi ini:



Berikut ini bagan fungsi-pemanggil untuk fungsi ini:



#### 4.6.4 Dokumentasi Anggota: Data

##### 4.6.4.1 color

```
int rectangle::color
```

warna kotak

##### 4.6.4.2 height

```
int rectangle::height
```

tinggi kotak

#### 4.6.4.3 large

```
int rectangle::large
```

luas kotak

#### 4.6.4.4 width

```
int rectangle::width
```

lebar kotak

#### 4.6.4.5 x

```
int rectangle::x
```

koordinat x

#### 4.6.4.6 y

```
int rectangle::y
```

koordinat y

Dokumentasi untuk kelas ini dibangkitkan dari file-file berikut:

- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[rectangle.h](#)
- C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/[rectangle.cpp](#)

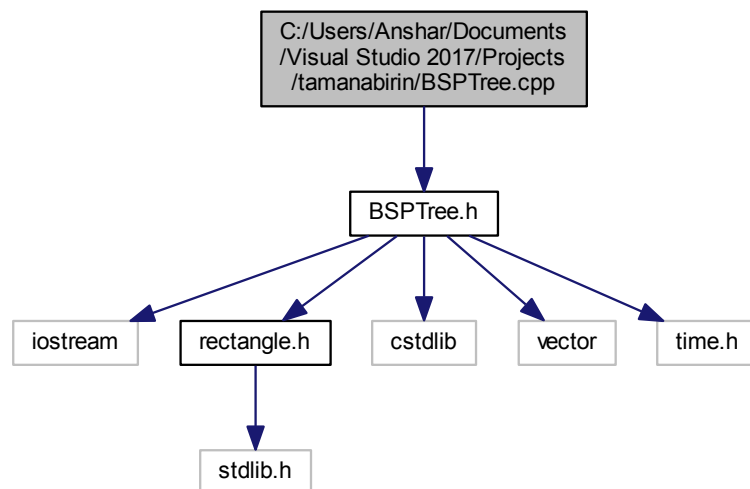
## Bab 5

# Dokumentasi File

### 5.1 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/↵ BSPTree.cpp

```
#include "BSPTree.h"
```

Bagan kebergantungan pemuatan untuk BSPTree.cpp:

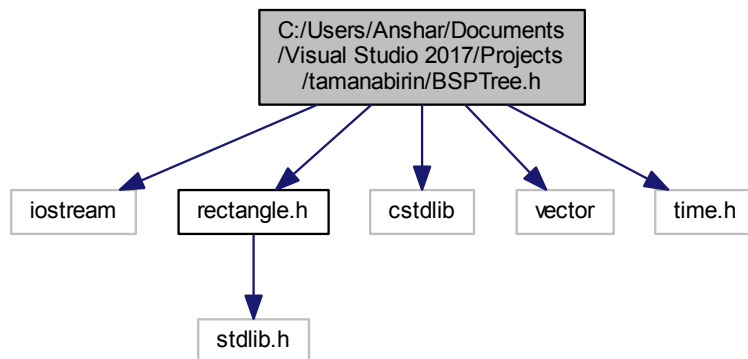


### 5.2 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/↵ BSPTree.h

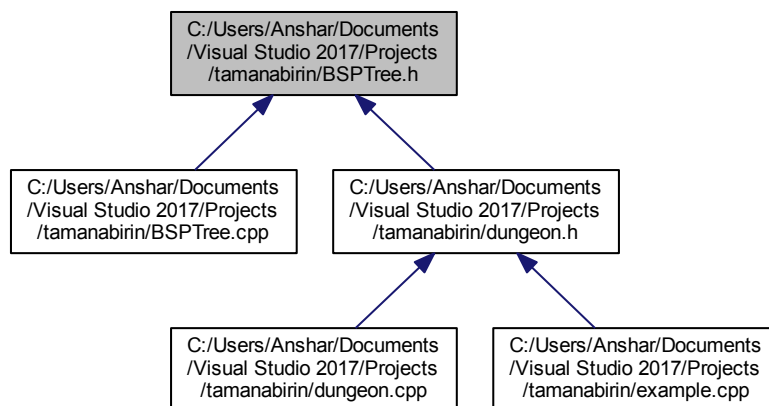
```
#include <iostream>
#include "rectangle.h"
#include <cstdlib>
#include <vector>
```

```
#include <time.h>
```

Bagan kebergantungan pemuatan untuk BSPTree.h:



Bagan ini menunjukkan file-file yang memuat atau menggunakan file ini baik secara langsung maupun tidak langsung:



## Kelas

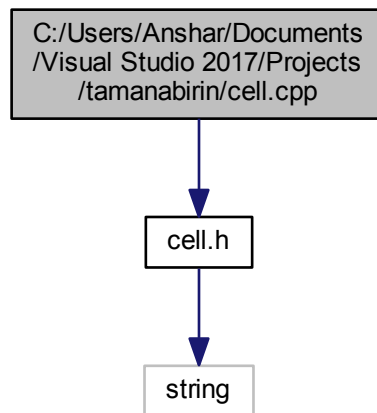
- class **BSPTree**

*merupakan objek berbentuk binary tree yang dimana nilai child-nya adalah hasil pembagian dari nilai parent-nya*

## 5.3 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/cell.cpp

```
#include "cell.h"
```

Bagan kebergantungan pemuatan untuk cell.cpp:

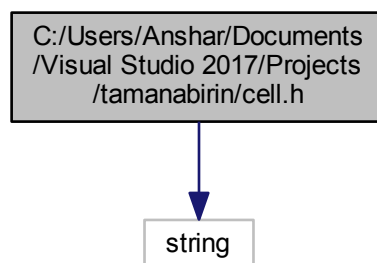


## 5.4 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/cell.h

merupakan objek berupa [cell\(layaknya ruang di lemari\)](#) yang menyimpan beberapa attribut

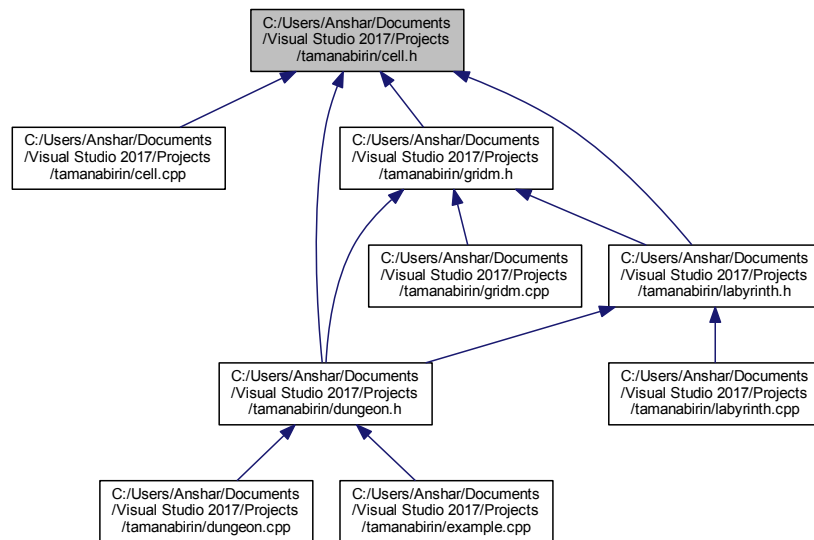
```
#include <string>
```

Bagan kebergantungan pemuatan untuk cell.h:



Bagan ini menunjukkan file-file yang memuat atau menggunakan file ini baik secara langsung maupun tidak

langsung:



## Kelas

- class `cell`

merupakan objek berupa `cell`(layaknya ruang di lemari) yang menyimpan beberapa attribut

### 5.4.1 Keterangan Lengkap

merupakan objek berupa `cell`(layaknya ruang di lemari) yang menyimpan beberapa attribut

Penulis

Muhammad Anshar

Tanggal

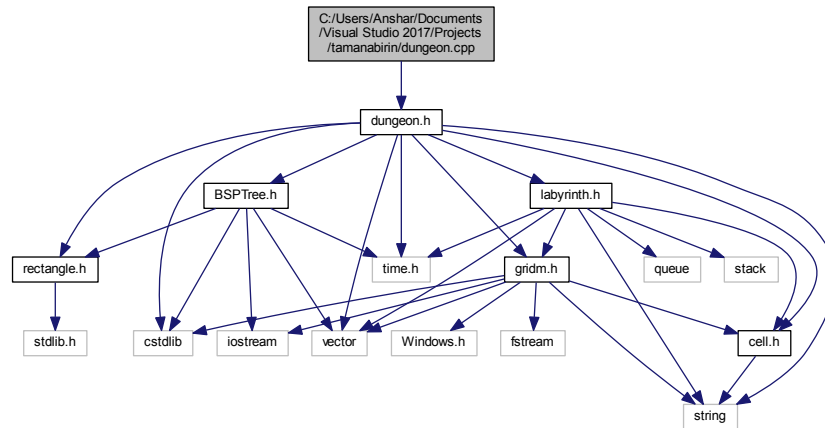
2018-05-06

## 5.5 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/dungeon.cpp

```
#include "dungeon.h"
```



Bagan kebergantungan pemuatan untuk dungeon.cpp:



## 5.6 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/dungeon.h

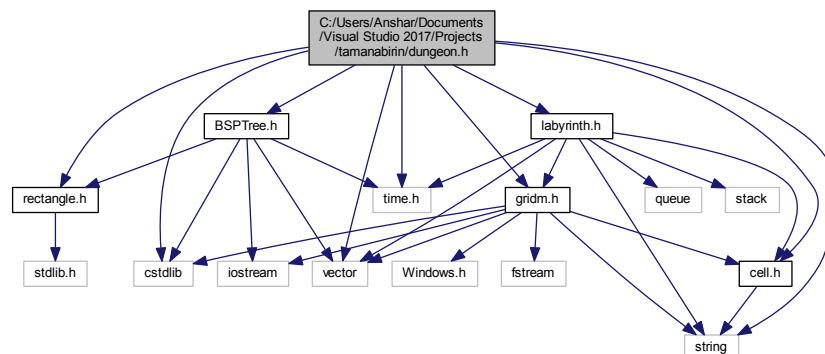
merupakan objek untuk menggenerate duneon dalam bentuk martriks char

```

#include <cstdlib>
#include <time.h>
#include <vector>
#include "rectangle.h"
#include "cell.h"
#include "gridm.h"
#include "BSPTree.h"
#include "labyrinth.h"
#include <string>

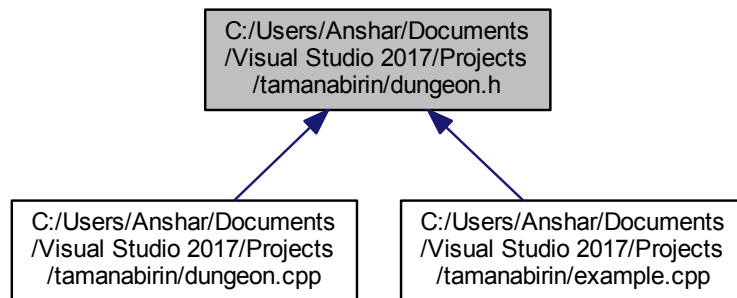
```

Bagan kebergantungan pemuatan untuk dungeon.h:



Bagan ini menunjukkan file-file yang memuat atau menggunakan file ini baik secara langsung maupun tidak

langsung:



## Kelas

- class `dungeon`

*merupakan satu satunya objek yang di #include untuk membuat dungeon menggunakan fungsi dan mengatur attributnya.*

### 5.6.1 Keterangan Lengkap

merupakan objek untuk menggenerate duneon dalam bentuk martriks char

#### Penulis

Muhammad Anshar

#### Tanggal

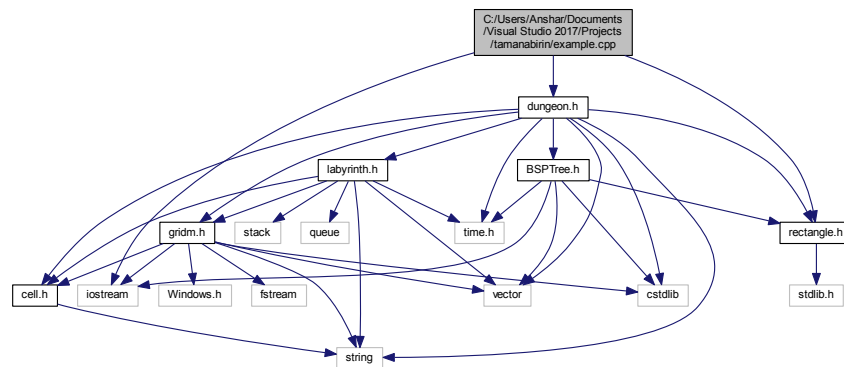
2018-06-02

## 5.7 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/example.cpp

```
#include <iostream>
#include "dungeon.h"
```

```
#include "rectangle.h"
```

Bagan kebergantungan pemuatan untuk example.cpp:



## Fungsi

- int `main` ()

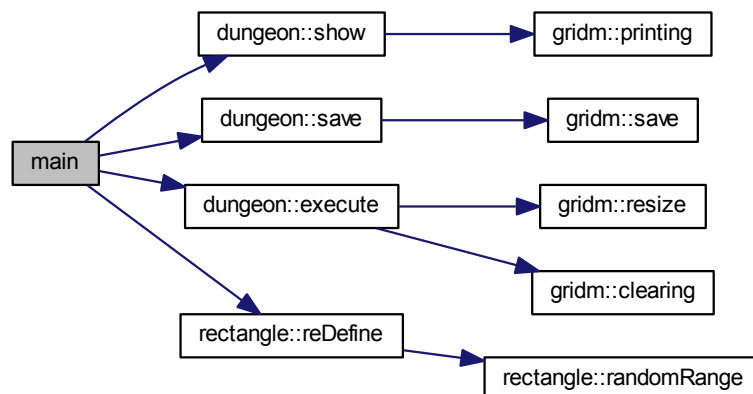
### 5.7.1 Dokumentasi Fungsi

#### 5.7.1.1 `main()`

```
int main ( )
```

dungeon yang menggunakan `BSPTree`

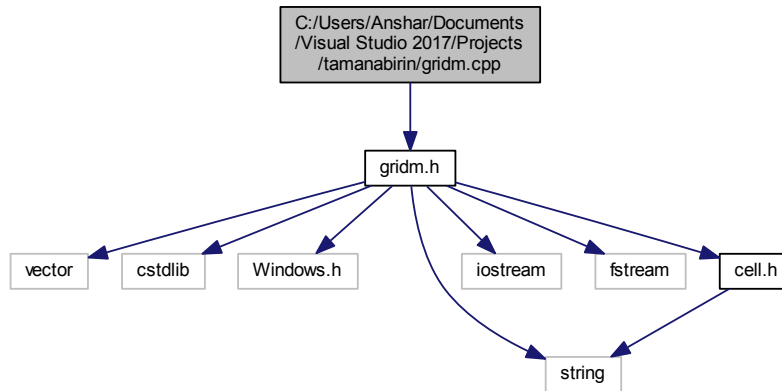
dungeon yang menggunakan deinisi ruangan secara manualBerikut ini bagan fungsi-terpanggil untuk fungsi ini:



## 5.8 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/gridm.cpp

```
#include "gridm.h"
```

Bagan kebergantungan pemuatan untuk gridm.cpp:

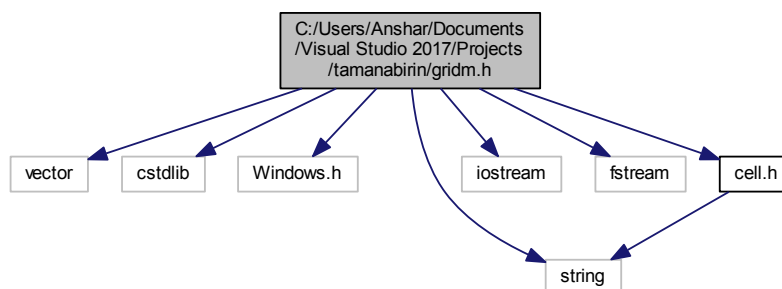


## 5.9 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/gridm.h

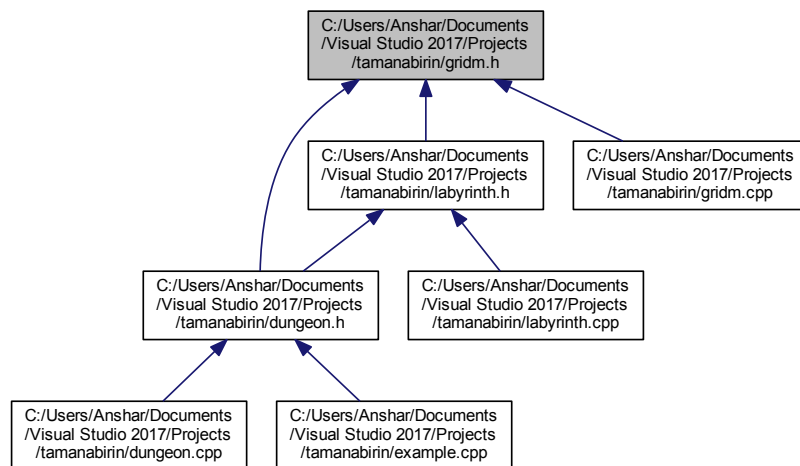
merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat x,y seperti cell pada microsofft excell

```
#include <vector>
#include <cstdlib>
#include <Windows.h>
#include <string>
#include <iostream>
#include <fstream>
#include "cell.h"
```

Bagan kebergantungan pemuatan untuk gridm.h:



Bagan ini menunjukkan file-file yang memuat atau menggunakan file ini baik secara langsung maupun tidak langsung:



## Kelas

- class `gridm`

*merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat x,y seperti cell pada microsoft excel*

### 5.9.1 Keterangan Lengkap

merupakan objek grid map. yaitu objek yang menampung sekumpulan objek cell pada koordinat x,y seperti cell pada microsoft excel

## Penulis

Muhammad Anshar

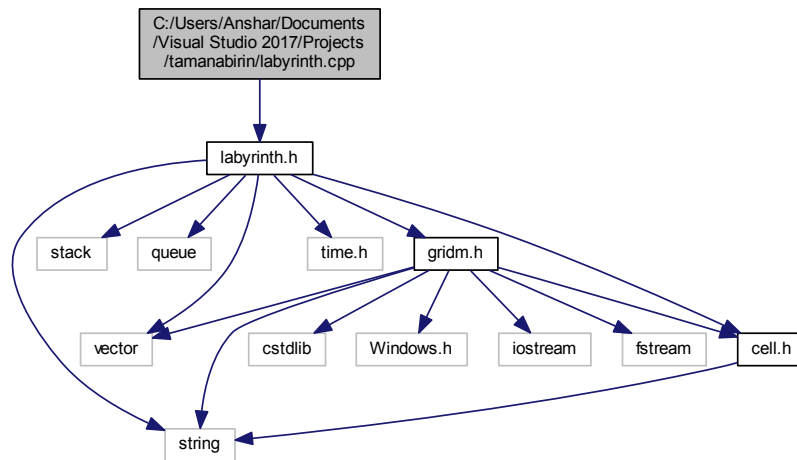
## Tanggal

2018-05-07

## 5.10 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/labyrinth.cpp

```
#include "labyrinth.h"
```

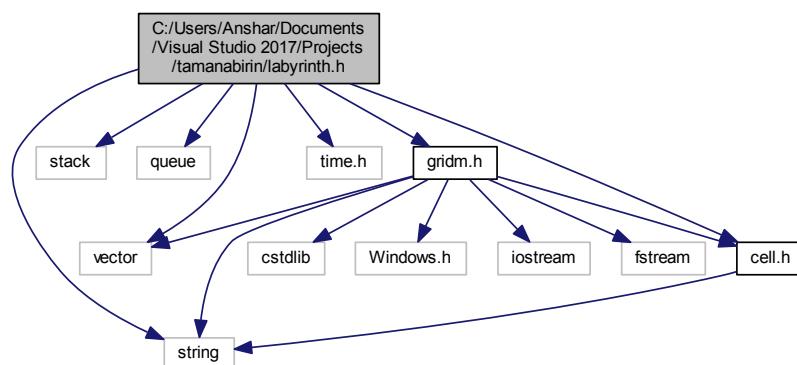
Bagan kebergantungan pemuatan untuk labyrinth.cpp:



## 5.11 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/labyrinth.h

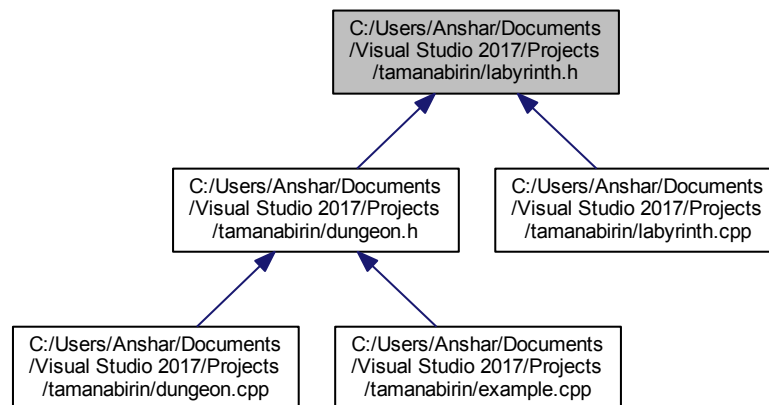
```
#include <string>
#include <stack>
#include <queue>
#include <vector>
#include <time.h>
#include "gridm.h"
#include "cell.h"
```

Bagan kebergantungan pemuatan untuk labyrinth.h:



Bagan ini menunjukkan file-file yang memuat atau menggunakan file ini baik secara langsung maupun tidak

langsung:



## Kelas

- class `labyrinth`

*merupakan objek maze(labirin) yang dibuat di dalam grid map dengan menggunakan algoritma deep first search*

## Makro Definisi

- `#define up 0`
- `#define down 1`
- `#define left 2`
- `#define right 3`

### 5.11.1 Dokumentasi Makro Definisi

#### 5.11.1.1 down

```
#define down 1
```

#### 5.11.1.2 left

```
#define left 2
```

#### 5.11.1.3 right

```
#define right 3
```

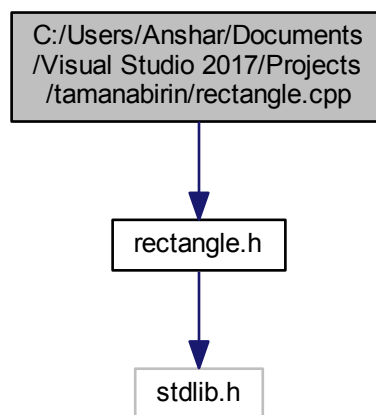
#### 5.11.1.4 up

```
#define up 0
```

### 5.12 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/rectangle.cpp

```
#include "rectangle.h"
```

Bagan kebergantungan pemuatan untuk rectangle.cpp:



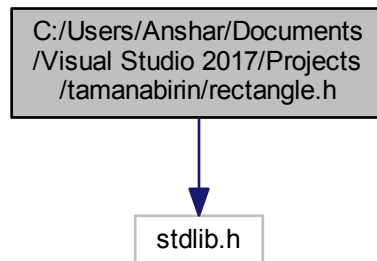
### 5.13 Referensi File C:/Users/Anshar/Documents/Visual Studio 2017/Projects/tamanabirin/rectangle.h

merupakan objek berbentuk kotak

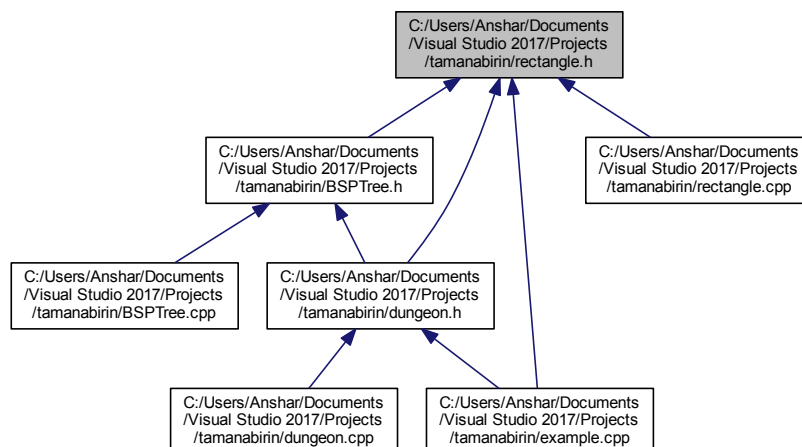


```
#include <stdlib.h>
```

Bagan kebergantungan pemuatan untuk rectangle.h:



Bagan ini menunjukkan file-file yang memuat atau menggunakan file ini baik secara langsung maupun tidak langsung:



## Kelas

- class `rectangle`  
*merupakan objek berbentuk kotak*

### 5.13.1 Keterangan Lengkap

merupakan objek berbentuk kotak

Penulis

Muhammad Anshar

Tanggal

2018-05-06

# Indeks

- ~BSTree
  - BSTree, 8
- ~cell
  - cell, 13
- ~dungeon
  - dungeon, 16
- ~gridm
  - gridm, 23
- ~labyrinth
  - labyrinth, 37
- ~rectangle
  - rectangle, 40
- addLeaf
  - BSTree, 8
- BSTree, 7
  - ~BSTree, 8
  - addLeaf, 8
  - BSTree, 8
  - getListLeaf, 9
  - getListLeafValue, 9
  - getListOrder, 10
  - getListOrderValue, 10
  - preOrder, 11
  - printlnOrder, 11
  - split, 11
- bsptree
  - dungeon, 19
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/BSTree.cpp, 45
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/BSTree.h, 45
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/cell.cpp, 46
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/cell.h, 47
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/dungeon.cpp, 48
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/dungeon.h, 49
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/example.cpp, 50
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/gridm.cpp, 52
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/gridm.h, 52
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/labyrinth.cpp, 53
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/labyrinth.h, 54
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/rectangle.cpp, 56
- C:/Users/Anshar/Documents/Visual Studio 2017/↵  
Projects/tamanabirin/rectangle.h, 56
- cell, 12
  - ~cell, 13
  - cell, 13
  - color, 13
  - display, 13
  - value, 14
  - visited, 14
  - x, 14
  - y, 14
- clearing
  - gridm, 23
- color
  - cell, 13
  - rectangle, 43
- display
  - cell, 13
- down
  - labyrinth.h, 55
- dungeon, 15
  - ~dungeon, 16
  - bsptree, 19
  - dungeon, 16
  - execute, 17
  - inOut, 19
  - lebar, 19
  - listRuanganManual, 19
  - maxLebar, 19
  - maxPanjang, 20
  - minLebar, 20
  - minPanjang, 20
  - panjang, 20
  - proses, 20
  - save, 17
  - show, 18
- example.cpp
  - main, 51
- execute
  - dungeon, 17
- get\_height
  - gridm, 24
- get\_width

- gridm, 24
- getCell
  - gridm, 25
- getColor
  - gridm, 25
- getDisplay
  - gridm, 26
- getListLeaf
  - BSPTree, 9
- getListLeafValue
  - BSPTree, 9
- getListOrder
  - BSPTree, 10
- getListOrderValue
  - BSPTree, 10
- getMaze
  - labyrinth, 37
- getValue
  - gridm, 26
- getVisited
  - gridm, 26
- gridm, 21
  - ~gridm, 23
  - clearing, 23
  - get\_height, 24
  - get\_width, 24
  - getCell, 25
  - getColor, 25
  - getDisplay, 26
  - getValue, 26
  - getVisited, 26
  - gridm, 22
  - map, 35
  - printing, 28
  - resize, 28
  - save, 29
  - set\_region, 29
  - set\_region\_color, 30
  - set\_region\_display, 31
  - set\_region\_visited, 31
  - setCell, 32
  - setColor, 32
  - setDisplay, 33
  - setValue, 34
  - setVisited, 34
- height
  - rectangle, 43
- inOut
  - dungeon, 19
- labyrinth, 35
  - ~labyrinth, 37
  - getMaze, 37
  - labyrinth, 36
- labyrinth.h
  - down, 55
  - left, 55
  - right, 55
  - up, 56
- large
  - rectangle, 43
- lebar
  - dungeon, 19
- left
  - labyrinth.h, 55
- listRuanganManual
  - dungeon, 19
- main
  - example.cpp, 51
- map
  - gridm, 35
- maxLebar
  - dungeon, 19
- maxPanjang
  - dungeon, 20
- midX
  - rectangle, 40
- midY
  - rectangle, 40
- minLebar
  - dungeon, 20
- minPanjang
  - dungeon, 20
- panjang
  - dungeon, 20
- preOrder
  - BSPTree, 11
- printlnOrder
  - BSPTree, 11
- printing
  - gridm, 28
- proses
  - dungeon, 20
- randomRange
  - rectangle, 41
- reDefine
  - rectangle, 41
- rectangle, 38
  - ~rectangle, 40
  - color, 43
  - height, 43
  - large, 43
  - midX, 40
  - midY, 40
  - randomRange, 41
  - reDefine, 41
  - rectangle, 39
  - width, 44
  - x, 44
  - y, 44
- resize
  - gridm, 28
- right

- labyrinth.h, [55](#)
- save
  - dungeon, [17](#)
  - gridm, [29](#)
- set\_region
  - gridm, [29](#)
- set\_region\_color
  - gridm, [30](#)
- set\_region\_display
  - gridm, [31](#)
- set\_region\_visited
  - gridm, [31](#)
- setCell
  - gridm, [32](#)
- setColor
  - gridm, [32](#)
- setDisplay
  - gridm, [33](#)
- setValue
  - gridm, [34](#)
- setVisited
  - gridm, [34](#)
- show
  - dungeon, [18](#)
- split
  - BSPTree, [11](#)
- up
  - labyrinth.h, [56](#)
- value
  - cell, [14](#)
- visited
  - cell, [14](#)
- width
  - rectangle, [44](#)
- x
  - cell, [14](#)
  - rectangle, [44](#)
- y
  - cell, [14](#)
  - rectangle, [44](#)