

# Pemrograman Berorientasi Object

S1 Informatika UNS

## Program Multithreading

Program 1 – file *threadProc.cpp*

```
#include <thread>
#include <iostream>
using namespace std;

void threadProc() {
    for( int count=0; count<3; ++count ) {
        cout << "Message ⇒ "
            << count
            << " from "
            << this_thread::get_id()
            << endl;
    }
}

int main() {
    thread thread1 ( threadProc );
    thread thread2 ( threadProc );
    thread thread3 ( threadProc );
    thread1.join();
    thread2.join();
    thread3.join();
    return 0;
}
```

simpan source code di atas dengan nama file **threadProc.cpp**

compile dan jalankan program

\*kalau pakai compiler **gcc** : compile program dengan menambahkan **-lpthread**

**g++ cobaThread.cpp -lpthread**

untuk compiler selain gcc menyesuaikan (*silakan googling, intinya supaya source dengan implementasi thread tersebut bisa tercompile*)

1. Tunjukkan output dari source code threadProc.cpp
2. Ada berapa thread di dalam program threadProc.cpp
3. Jelaskan kegunaan dari fungsi join() pada baris

```
thread1.join();
thread2.join();
thread3.join();
```

## Program 2

→ File **Thread.h**

```
#include <iostream>
#include <thread>
using namespace std;
```

```
class Thread {
private:
    thread *pThread;
    bool stopped;
    void run();
```

```
public:
    Thread();
    ~Thread();
```

```

    void start();
    void stop();
    void join();
    void detach();
};

```

Simpan source code dengan nama file ***Thread.h***

4. Jelaskan maksud dari baris ***~Thread()***; pada source code Thread.h

5. Jelaskan kegunaan file ***.h*** pada C++

→ File ***Thread.cpp***

```

#include "Thread.h"

```

```

Thread::Thread() {
    pthread = NULL;
    stopped = false;
}

```

```

Thread::~~Thread() {
    delete pthread;
    pthread = NULL;
}

```

```

void Thread::run() {
    while ( ! stopped ) {
        cout << "\n"<< this_thread::get_id() << endl;
        this_thread::sleep_for ( 1s );
    }
    cout << "\nThread " << this_thread::get_id()
    << " stopped as requested." << endl;
    return;
}

```

```

void Thread::stop() {
    stopped = true;
}

```

```

void Thread::start() {
    pthread = new thread( &Thread::run, this );
}

```

```

void Thread::join() {
    pthread→join();
}

```

```

void Thread::detach() {
    pthread→detach();
}

```

Simpan source code dengan nama ***Thread.cpp***, simpan pada folder yang sama dengan file Thread.h

→ File ***main.cpp***

```

#include "Thread.h"

```

```

int main() {
    Thread thread1, thread2, thread3;
    thread1.start();
    thread2.start();
    thread3.start();
    this_thread::sleep_for ( 10s );
}

```

```

    thread1.stop();
    thread2.stop();
    thread3.stop();
    this_thread::sleep_for ( 3s );
    return 0;
}

```

Simpan source code dengan nama **main.cpp**, simpan pada folder yang sama dengan file Thread.h compile dan jalankan program

\*kalau pakai compiler **gcc** : compile program dengan menambahkan **-lpthread**

**g++ Thread.cpp main.cpp -std=c++17 -o threads.exe -lpthread**

jalankan program dengan mengeksekusi program hasil compile, apabila mengikuti command sebelumnya untuk compile pakai **gcc**, hasil compile program adalah file **threads.exe** (anda bisa menggantinya dengan nama yang lain)

untuk compiler selain gcc menyesuaikan (*silakan googling, intinya supaya source dengan implementasi thread tersebut bisa tercompile*)

6. Jelaskan kegunaan dari fungsi start() pada baris

```

thread1.start();
thread2.start();
thread3.start();

```

7. Jelaskan maksud dari baris berikut

```

this_thread::sleep_for ( 10s );

```

8. Jelaskan apa yang terjadi setelah fungsi stop pada baris berikut dieksekusi

```

thread1.stop();
thread2.stop();
thread3.stop();

```

*source code diambil dari buku Mastering C++ Programming : Modern C++ 17 at your fingertips, Jeganathan Swaminathan, 2017*