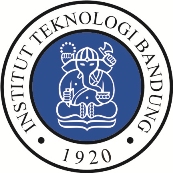
**13512036**

**[If3230]**

**Laporan Praktikum Pemrograman CUDA**



**oleh :**

**Riva Syafri Rachmatullah / 13512036**

**Daniar Heri Kurniawan / 13512064**

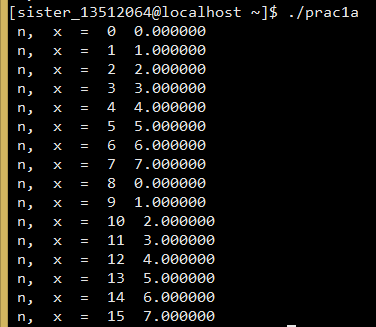
**Teknik Informatika**

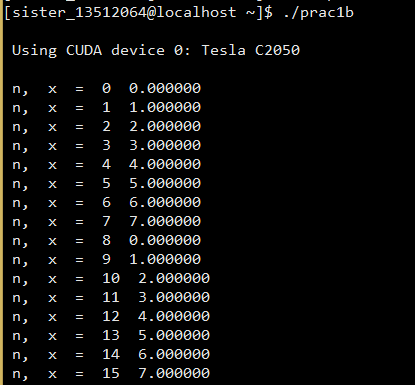
**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2015**

**PENGAMATAN**



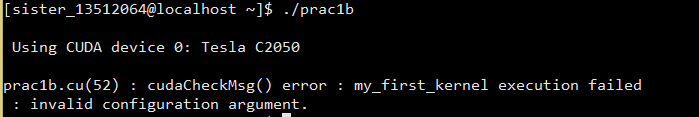
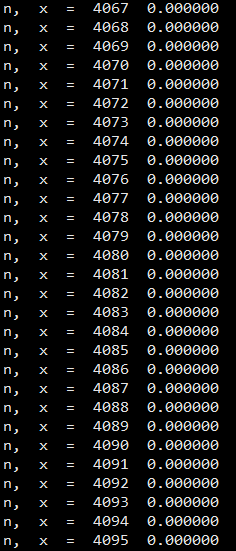


**1. Mengubah prac1a.cu dan prac1b.cu sehingga terdapat error, misalnya:**

**i. gunakan nthreads = 2048. Berapakah maksimum threads per block yang dapat**

**digunakan? Apa yang terjadi jika jumlah threads per block di atas batas yang**

**dibolehkan?**



**Analisa : Jumlah maksimum threads perblock yang boleh digunakan adalah 1024. Jika jumlah threads melebihi batas maka program tidak akan berfungsi dengan baik karena eksekusi kernel ada yang gagal. Pada program prac1b.cu terdapat penanganan error, sehingga ketika ada thread yang gagal akan ditampilkan pesan error**

**ii. dengan mengalokasikan memori berlebihan. Berapakah maksimum memory**

**yang dapat dialokasikan? Berapakah ukuran jumlah nblocks yang dibolehkan?**

**Hint: untuk membuat thread yang lebih banyak lagi agar jumlah memorynya**

**melebihi kapasitas, mungkin anda harus menggunakan ukuran blok lebih dari 1**

**dimensi (e.g. menggunakan 2 atau 3 dimensi).**

**Analisa : Jumlah maksimum alokasi memori adalah 65535 x 1024 x 4 B = 268.431.360 B = 268,4 MB. Jumlah maksimum block adalah 65535 buah.**

**iii. gunakan nblocks = 0**

**Analisa: Program tidak bisa berjalan dengan normal karena ada error.**

**2. Tambahkan sebuah statement printf pada kernel my\_first\_kernel, untuk menampilkan nilai tid. Compile dan jalankan.**

**Analisa: Program harus dicompile dengan perintah khusus yaitu : nvcc -arch=sm\_20 prac1b.cu -run**

**Threads dalam satu block akan dieksekusi bersamaan namun urutan block yang dieksekusi tidak tentu (random).**

1. **Buatlah program prac1c.cu dengan berbasis prac1b.cu, sehingga program tersebut menambahkan 2 buah vector yang diinisialisasi di host, dan mengirimkan hasilnya kembali ke host**