Tugas Pemrograman Parallel

Oleh:

D121171519 - Glenn Claudio Ivan Petrus

Departemen Teknik Informatika

Fakultas Teknik

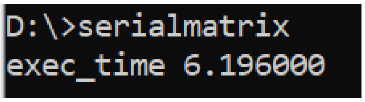
Universitas Hasanuddin

2020

Program Serial Matriks – Source Code

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

#define MATRIX_SIZE 1000

int main () {
      int i, j ,k;
      int row = 1000, col = 1000;
      int *A = (int *)malloc(row * col * sizeof(int)); // Alokasi Dinamis Array Dua
Dimensi Menggunakan Single Pointer
      int *B = (int *)malloc(row * col * sizeof(int)); // Alokasi Dinamis Array Dua
Dimensi Menggunakan Single Pointer
      int *C = (int *)malloc(row * col * sizeof(int)); // Alokasi Dinamis Array Dua
Dimensi Menggunakan Single Pointer
      long start, finish; // Clock
      double exec_time;

      // Membangkitkan Nilai Elemen untuk Matriks A dan Matriks B
      for(i=0;i<MATRIX_SIZE;i++){
            for(j=0;j<MATRIX_SIZE;j++){
                  *(A + i*col + j) = j*100;
                  *(B + i*col + j) = j*100;
            }
      }

      // Mengkalkulasi Perkalian Matriks
      start = clock(); // Mulai Clock Saat Kalkulasi
      for(i=0;i<MATRIX_SIZE;i++){
            for(j=0;j<MATRIX_SIZE;j++){
                  *(C + i*col + j) = 0;
                  for(k=0;k<MATRIX_SIZE;k++){
                        *(C + i*col + j) += ((*(A + k*col + j))*(*(B + j*col +
k)));
                  }
            }
      }
      finish = clock();

      // Mencetak Nilai Elemen Matriks C
//    for(i=0;i<MATRIX_SIZE;i++){
//          for(j=0;j<MATRIX_SIZE;j++){
//                printf("%d\n",      *(C + i*col + j));
//          }
//    }

      exec_time = (double) (finish - start)/CLOCKS_PER_SEC;
      printf("exec_time %1f\n", exec_time);

      return 0;
}
```

Program Serial Matriks – Execution Time

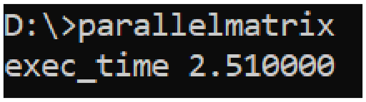


```
D:\>serialmatrix
exec_time 6.196000
```

Program Parallel Matriks – Source Code

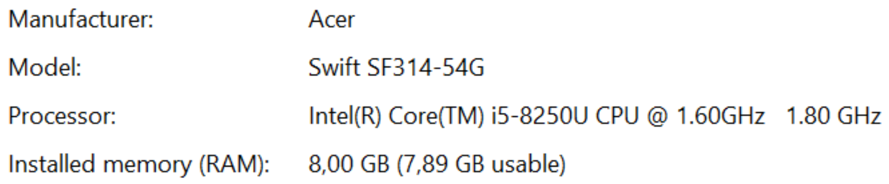
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MATRIX_SIZE 1000

int main () {
        int i, j ,k;
        int row = 1000, col = 1000;
        int *A = (int *)malloc(row * col * sizeof(int)); // Alokasi Dinamis Array Dua
Dimensi Menggunakan Single Pointer
        int *B = (int *)malloc(row * col * sizeof(int)); // Alokasi Dinamis Array Dua
Dimensi Menggunakan Single Pointer
        int *C = (int *)malloc(row * col * sizeof(int)); // Alokasi Dinamis Array Dua
Dimensi Menggunakan Single Pointer
        long start, finish; // Clock
        double exec_time;


        // Membangkitkan Nilai Elemen untuk Matriks A dan Matriks B
        for(i=0;i<MATRIX_SIZE;i++){
                for(j=0;j<MATRIX_SIZE;j++){
                        *(A + i*col + j) = j*100;
                        *(B + i*col + j) = j*100;
                }
        }

        // Mengkalkulasi Perkalian Matriks
        start = clock(); // Mulai Clock Saat Kalkulasi
        #pragma omp parallel shared(A,B,C) private(i,j,k)
        {
                #pragma omp for  schedule(static)
                for(i=0;i<MATRIX_SIZE;i++){
                        for(j=0;j<MATRIX_SIZE;j++){
                                *(C + i*col + j) = 0;
                                for(k=0;k<MATRIX_SIZE;k++){
                                        *(C + i*col + j) += ((*(A + k*col + j))*(*(B + j*col
+ k)));
                                }
                        }
                }
        }
        finish = clock();

        // Mencetak Nilai Elemen Matriks C
//      for(i=0;i<MATRIX_SIZE;i++){
//              for(j=0;j<MATRIX_SIZE;j++){
//                      printf("%d\n",      *(C + i*col + j));
//              }
//      }

        exec_time = (double) (finish - start)/CLOCKS_PER_SEC;
        printf("exec_time %1f\n", exec_time);

        return 0;
}
```

Program Parallel Matriks – Execution Time



```
D:\>parallelmatrix
exec_time 2.510000
```

Tabel Pengamatan



Manufacturer:	Acer
Model:	Swift SF314-54G
Processor:	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Installed memory (RAM):	8,00 GB (7,89 GB usable)

Ukuran Matriks	**Execution Time (Serial) sec**	**Execution Time (Parallel) sec**	**Speed-Up**
500 x 500	0,548000	0,235000	2,331910
1000 x 1000	6,196000	2,510000	2,468520