TUGAS PERTEMUAN SEBELAS MATA KULIAH KOMPUTASI PARALEL: TASK DECOMPOSITION LEAF STRATEGY

*Oleh: Fauzan Zacky Firmansyah (23106050047)*

A

Leaf Strategy Menggunakan OpenMP

|  |
| --- |
| **Code(jpg):** |
| **Code(word):**  #include <stdio.h>  #include <omp.h>  // Fungsi untuk menghitung jumlah elemen dalam array  int sumArray(int\* array, int start, int end) {      // Jika array memiliki 1 elemen (basis rekursi)      if (start == end) {          return array[start];      }      // Jika ada lebih dari satu elemen, pecah tugas (decompose task)      int mid = (start + end) / 2; // Tentukan titik tengah array      int leftSum = 0, rightSum = 0;      #pragma omp task shared(leftSum) // Task untuk menghitung bagian kiri      leftSum = sumArray(array, start, mid);      #pragma omp task shared(rightSum) // Task untuk menghitung bagian kanan      rightSum = sumArray(array, mid + 1, end);      #pragma omp taskwait // Tunggu hingga semua task selesai      return leftSum + rightSum; // Gabungkan hasil dari kedua task  }  int main() {      // Inisialisasi array contoh      int array[] = {1, 2, 3, 4, 5, 6, 7, 8};      int n = sizeof(array) / sizeof(array[0]);      // Hasil akhir penjumlahan      int totalSum = 0;      // Bagian paralel      #pragma omp parallel      {          #pragma omp single // Satu thread memulai tugas utama          totalSum = sumArray(array, 0, n - 1);      }      // Cetak hasil      printf("Total Sum = %d\n", totalSum);      return 0;  } |
| **Hasil Run:** |

**Pertanyaan dan Jawaban:**

Pertanyaan 1: Apa fungsi #pragma omp task dalam kode ini?

Jawaban: Direktif ini digunakan untuk membuat tugas paralel baru yang akan dijalankan oleh thread yang tersedia di lingkungan OpenMP. Dalam kode, tugas paralel digunakan untuk menghitung jumlah elemen di bagian kiri dan kanan array secara terpisah.

Pertanyaan 2: Apa tujuan dari #pragma omp taskwait?

Jawaban**:** Direktif ini memastikan bahwa semua tugas paralel yang telah dibuat selesai sebelum melanjutkan ke langkah berikutnya (penggabungan hasil). Tanpa ini, program dapat melanjutkan sebelum semua tugas selesai, sehingga menghasilkan hasil yang salah