

| | |
|-------|---------------|
| NAMA | INGWE IANNICO |
| NBI | 1461600214 |
| KELAS | R |

JAWABAN NO 1.

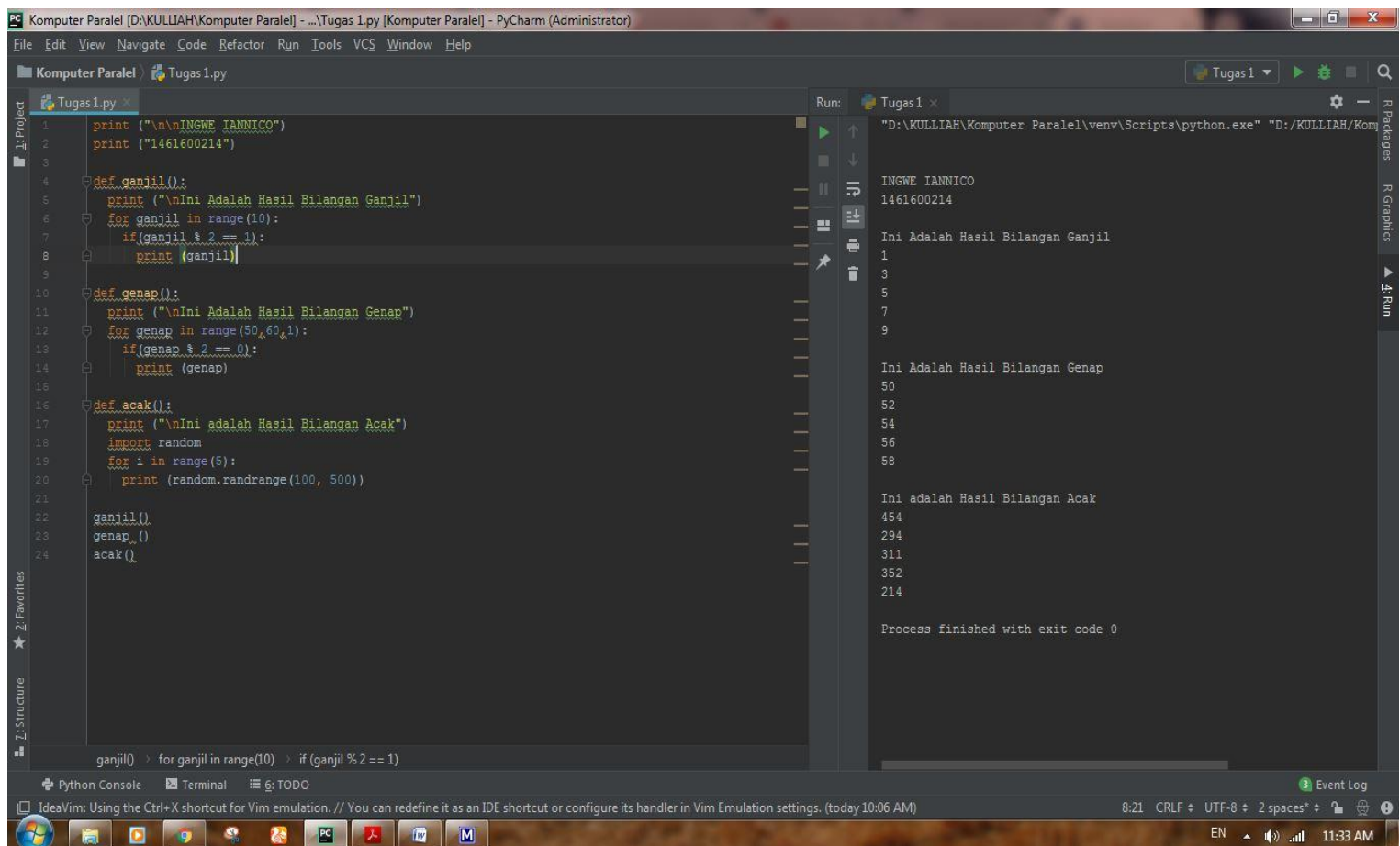
```
print ("\n\nINGWE IANNICO")
print ("1461600214")

def ganjil():
    print ("\nIni Adalah Hasil Bilangan Ganjil")
    for ganjil in range(10):
        if(ganjil % 2 == 1):
            print (ganjil)

def genap(b):
    print ("\nIni Adalah Hasil Bilangan Genap")
    for genap in range(50,b,1):
        if(genap % 2 == 0):
            print (genap)

def acak():
    print ("\nIni adalah Hasil Bilangan Acak")
    import random
    for i in range(5):
        print (random.randrange(100, 500))

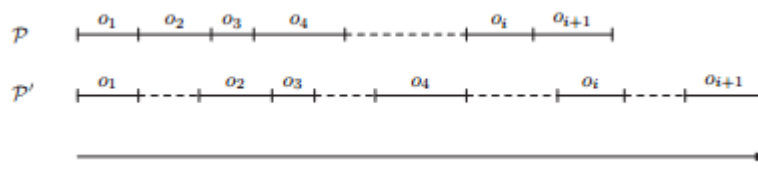
ganjil()
genap (60)
acak()
```



RANGKUMAN NO 2

Komputasi Paralel

program berurutan adalah urutan instruksi yang memecahkan masalah dengan mengubah data input menjadi data output. Diasumsikan bahwa program berurutan dijalankan oleh satu prosesor. Jika lebih banyak prosesor yang akan digunakan untuk menyelesaikan masalah, itu harus dipartisi menjadi sejumlah sub-masalah yang dapat diselesaikan secara paralel. Solusi untuk masalah asli adalah komposisi solusi untuk subproblem. Subproblem diselesaikan dengan komponen terpisah yang merupakan bagian dari program bersamaan. Setiap komponen adalah program berurutan tradisional yang disebut tugas komputasi. Program bersamaan terdiri dari sejumlah tugas yang menggambarkan perhitungan yang dapat dieksekusi secara paralel. Program bersamaan mendefinisikan bagaimana tugas bekerja sama satu sama lain menerapkan hasil perhitungan sebagian, dan bagaimana mereka menyinkronkan tindakan mereka. Tugas dieksekusi dalam komputer paralel di bawah pengawasan sistem operasi. Tugas tunggal dilakukan sebagai proses berurutan (serial), yaitu sebagai urutan operasi, oleh prosesor konvensional yang kita sebut prosesor virtual. Dalam proses berurutan, yang dihasilkan dari pelaksanaan urutan instruksi tunggal, operasi berikutnya dimulai hanya setelah penyelesaian operasi sebelumnya. Dengan demikian, urutan operasi didefinisikan dengan jelas.

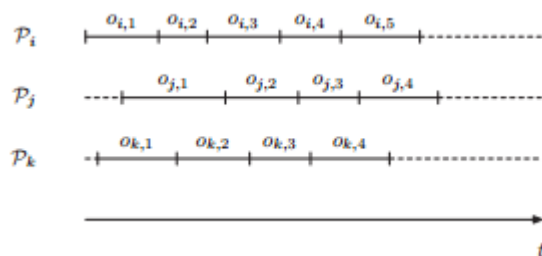


menunjukkan proses berurutan P dan P' yang berbeda dalam waktu dimulainya dan penghentian operasi $o_1, o_2, \dots, o_i, o_{i+1}$. Jika operasi proses identik, dalam hal argumen dan hasil mereka, hasil perhitungan P dan P' juga akan identik, meskipun waktu pelaksanaannya akan berbeda.

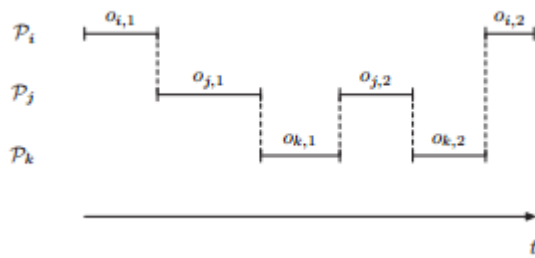
Pemrograman Paralel sendiri adalah teknik pemrograman komputer yang memungkinkan eksekusi perintah/operasi secara bersamaan. Bila komputer yang digunakan secara bersamaan tersebut dilakukan oleh komputer-komputer terpisah yang terhubung dalam satu jaringan komputer, biasanya disebut sistem terdistribusi. Bahasa pemrograman yang populer digunakan dalam pemrograman paralel adalah MPI (Message Passing Interface) dan PVM (Parallel Virtual Machine).

komputasi paralel berbeda dengan multitasking. Pengertian multitasking adalah komputer dengan processor tunggal mengeksekusi beberapa tugas secara bersamaan. Walaupun beberapa orang yang bergelut di bidang sistem operasi beranggapan bahwa komputer tunggal tidak bisa melakukan beberapa pekerjaan sekaligus, melainkan proses penjadwalan yang berlakukan pada sistem operasi membuat komputer seperti mengerjakan tugas secara bersamaan. Sedangkan komputasi paralel sudah dijelaskan sebelumnya, bahwa komputasi paralel menggunakan beberapa processor atau komputer. Selain itu komputasi paralel tidak menggunakan arsitektur Von Neumann.

Proses berurutan yang dilakukan secara simultan dan tidak sinkron, di mana pelaksanaan operasi dapat tumpang tindih dalam waktu, disebut proses bersamaan. Karena sinkronitas ada banyak kemungkinan implementasi atau skenario pelaksanaan proses bersamaan. Pengakuan urutan relatif eksekusi pelaksanaan operasi dalam berbagai implementasi berarti pada kenyataannya tidak ada asumsi tentang kecepatan prosesor virtual yang dibuat. Itu dibenarkan karena kecepatan ini tergantung pada kecepatan dan jumlah prosesor nyata yang tersedia di komputer di mana program bersamaan akan dilaksanakan. Jika jumlah aktual dari prosesor nyata setidaknya sama dengan jumlah prosesor virtual, yaitu jumlah proses, maka proses dapat dieksekusi secara paralel.



Dalam hal ini, setiap proses dilakukan oleh prosesor nyata yang sesuai. Ada juga kasus di mana hanya satu prosesor nyata yang tersedia. Kemudian, proses individu dilakukan dengan interleaving

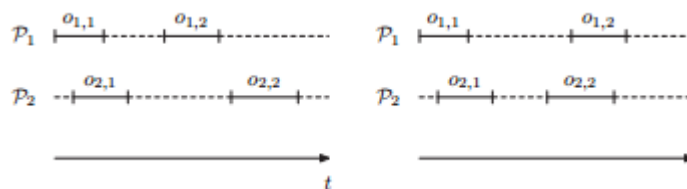


Jika program bersamaan dirancang dengan benar, maka kedua eksekusi ini, dengan satu atau lebih banyak prosesor, akan memberikan hasil yang sama.

Komunikasi antar Proses

Secara umum, proses konkuren dijalankan secara independen, tetapi mereka dapat berkomunikasi satu sama lain dalam titik waktu tertentu. Komunikasi proses dapat dilakukan

- Operasi sinkronisasi dapat memaksakan urutan parsial (serta total) di antara operasi yang dijalankan dalam proses.
- Ingat bahwa dalam setiap proses, operasi dijalankan secara berurutan.
- Di seluruh buku ini, oleh prosesor (nyata) yang kami maksud adalah uniprocessor atau inti sebagai perangkat fisik.



dicapai dengan dua cara: menggunakan memori bersama atau lewat pesan. Dalam cara pertama, diasumsikan bahwa memori bersama tersedia dan semua proses virtual yang melakukan komputasi bersamaan memiliki akses ke sana. Setiap prosesor dapat melakukan beberapa operasi pada sejumlah variabel yang terletak di memori bersama. Ini bisa berupa operasi sederhana baca atau tulis pada variabel bersama, tetapi juga yang lebih maju, seperti pertukaran, uji-dan-set, bandingkan-dan-tukar, ambil-dan-tambah. Dengan memanfaatkan variabel bersama proses transfer bersamaan satu sama lain hasil perhitungan, serta sinkronisasi tindakan mereka. Untuk tujuan sinkronisasi, penting bahwa operasi ini bersifat atomik). Dengan cara komunikasi kedua, prosesor virtual bertukar pesan melalui saluran komunikasi (memori bersama tidak tersedia). Setiap saluran biasanya menyediakan koneksi dua arah antara sepasang prosesor. Seperangkat saluran komunikasi membuat jaringan interkoneksi suatu sistem. Prosesor yang terletak di simpul jaringan memiliki kemampuan untuk melakukan perhitungan program bersamaan, serta untuk

mengirim dan menerima pesan dari prosesor tetangga. Dalam sebagian besar keadaan tidak ada asumsi yang dibuat mengenai latensi pengiriman pesan antara sumber dan prosesor target. Akibatnya, perhitungan sepenuhnya asinkron karena kedua titik yang tepat dalam waktu pelaksanaan operasi proses bersamaan, serta mengirim dan menerima pesan, tidak dapat diidentifikasi terlebih dahulu.

Program Bersamaan, Paralel, dan Terdistribusi

Proses bersamaan yang memecahkan masalah komputasi dapat diimplementasikan dalam tiga cara dasar:

- (i) proses dijalankan oleh satu prosesor dengan interleaving;
- (ii) setiap proses dijalankan oleh prosesor terpisah dan semua prosesor memiliki akses ke memori bersama;
- (iii) proses dijalankan oleh prosesor terpisah dan terdistribusi yang saling terhubung oleh saluran komunikasi. Kami berasumsi bahwa dua proses bersamaan, jika dimungkinkan untuk menjalankannya secara paralel. Proses-proses ini paralel, jika sewaktu-waktu keduanya dijalankan secara bersamaan. Dalam konteks ini, hanya proses dari kasus (ii) dan (iii) dapat dianggap sebagai paralel. Program bersamaan menentukan proses yang dapat dieksekusi secara paralel. Ini juga menjelaskan bagaimana masalah sinkronisasi dan komunikasi antarab proses diselesaikan.

Threads

merupakan komponen dari program paralel dieksekusi sebagai proses berurutan di bawah pengawasan sistem operasi. Dari sudut pandang implementasi, proses adalah entitas eksekusi yang dibuat, diawasi, dan dihancurkan oleh sistem operasi. Untuk mengeksekusi, sistem operasi mengalokasikan ke proses sumber daya tertentu, seperti waktu komputasi prosesor, ruang memori untuk menyimpan instruksi, data, dan tumpukan proses, set register, termasuk penghitung program dan penunjuk tumpukan. Suatu proses dapat dieksekusi oleh threads tunggal atau oleh kelompok threads yang bekerja sama. Threads adalah entitas eksekusi yang dapat menjalankan aliran instruksi secara mandiri. Beberapa threads juga dapat dijalankan secara paralel oleh beberapa prosesor atau prosesor multicore.

Model dari komputasi. Ada 4 model komputasi yang digunakan, yaitu:

- SISD
- SIMD
- MISD
- MIMD
- SISD

Yang merupakan singkatan dari Single Instruction, Single Data adalah satu-satunya yang menggunakan arsitektur Von Neumann. Ini dikarenakan pada model ini hanya digunakan 1 processor saja. Oleh karena itu model ini bisa dikatakan sebagai model untuk komputasi tunggal. Sedangkan ketiga model lainnya merupakan komputasi paralel yang menggunakan beberapa processor. Beberapa contoh komputer yang menggunakan model SISD adalah UNIVAC1, IBM 360, CDC 7600, Cray 1 dan PDP 1.

- **SIMD**

Yang merupakan singkatan dari Single Instruction, Multiple Data. SIMD menggunakan banyak processor dengan instruksi yang sama, namun setiap processor mengolah data yang berbeda. Sebagai contoh kita ingin mencari angka 27 pada deretan angka yang terdiri dari 100 angka, dan kita menggunakan 5 processor. Pada setiap processor kita menggunakan algoritma atau perintah yang sama, namun data yang diproses berbeda. Misalnya processor 1 mengolah data dari deretan / urutan pertama hingga urutan ke 20, processor 2 mengolah data dari urutan 21 sampai urutan 40, begitu pun untuk processor-processor yang lain. Beberapa contoh komputer yang menggunakan model SIMD adalah ILLIAC IV, MasPar, Cray X-MP, Cray Y-MP, Thinking Machine CM-2 dan Cell Processor (GPU).

- **MISD**

Yang merupakan singkatan dari Multiple Instruction, Single Data. MISD menggunakan banyak processor dengan setiap processor menggunakan instruksi yang berbeda namun mengolah data yang sama. Hal ini merupakan kebalikan dari model SIMD. Untuk contoh, kita bisa menggunakan kasus yang sama pada contoh model SIMD namun cara penyelesaian yang berbeda. Pada MISD jika pada komputer pertama, kedua, ketiga, keempat dan kelima sama-sama mengolah data dari urutan 1-100, namun algoritma yang digunakan untuk teknik pencariannya berbeda di setiap processor. Sampai saat ini belum ada komputer yang menggunakan model MISD.

- **MIMD**

Yang merupakan singkatan dari Multiple Instruction, Multiple Data. MIMD menggunakan banyak processor dengan setiap processor memiliki instruksi yang berbeda dan mengolah data yang berbeda. Namun banyak komputer yang menggunakan model MIMD juga memasukkan komponen untuk model SIMD. Beberapa komputer yang menggunakan model MIMD adalah IBM POWER5, HP/Compaq AlphaServer, Intel IA32, AMD Opteron, Cray XT3 dan IBM BG/L.

Proses program paralel bekerja sama satu sama lain melalui memori bersama dan mengeksekusi di komputer yang terintegrasi secara logis dan fisik. Kelompok kedua adalah program terdistribusi yang prosesnya berkomunikasi oleh mengirim pesan. Proses memiliki ruang alamat (pribadi) yang terpisah dan diimplementasikan dalam sistem terdistribusi, misalnya dalam kelompok yang terdiri dari komputer yang saling berhubungan oleh jaringan. Dalam konteks ini, program paralel dapat dianggap sebagai dirancang untuk mempercepat perhitungan, yang menghasilkan penyelesaian masalah yang rumit di waktu yang lebih singkat.