

1. Pengenalan Arsitektur Komputer

Ini adalah modul tentang organisasi komputer. Modul ini mendeskripsikan fungsi dan desain berbagai unit komputer digital yang menyimpan dan mengolah informasi. Modul ini juga berkaitan dengan unit komputer yang menerima informasi dari sumber eksternal dan mengirimkan hasil terkomputasi ke destinasi eksternal. Kebanyakan materi dalam modul ini ditujukan untuk hardware komputer dan arsitektur komputer. Hardware komputer terdiri dari sirkuit elektronik, display, media penyimpanan magnetik dan optik, perangkat elektromekanik, dan fasilitas komunikasi. Arsitektur komputer meliputi spesifikasi sekumpulan instruksi dan unit hardware yang melaksanakan instruksi tersebut. Dalam modul ini dibahas pula banyak aspek pemrograman dan komponen software dalam sistem komputer. Sangatlah penting mempertimbangkan aspek hardware dan software pada desain berbagai komponen komputer guna mencapai pemahaman yang baik pada suatu sistem komputer.

Bab ini memperkenalkan sejumlah konsep hardware dan software, menampilkan beberapa istilah umum, dan memberikan pandangan umum tentang aspek dasar subjek tersebut. Pembahasan yang lebih detil diberikan pada bab-bab selanjutnya.

1.1. TIPE KOMPUTER

Marilah pertama-tama kita mendefinisikan istilah komputer digital, atau cukup disebut komputer. Dalam istilah yang paling sederhana, suatu komputer kontemporer adalah mesin hitung elektronik cepat yang menerima informasi input terdigitalisasi, mengolahnya sesuai dengan daftar instruksi yang tersimpan secara internal dan memberikan informasi output hasil. Daftar instruksi itu disebut program komputer, dan penyimpanan internal-nya disebut memori komputer.

Terdapat banyak tipe komputer yang sangat bervariasi dalam hal ukuran, biaya, daya komputasi, dan tujuan penggunaan. Komputer yang paling umum adalah personal computer, yang banyak digunakan di rumah-rumah, sekolah-sekolah, dan kantor-kantor bisnis. Personal computer merupakan bentuk paling umum dari komputer desktop. Komputer desktop memiliki unit pengolahan dan penyimpanan, display visual dan unit output audio, dan keyboard yang dapat ditempatkan secara mudah di meja rumah dan kantor. Media penyimpanan tersebut termasuk harddisk,

CD-ROM, dan disket. Komputer notebook portable adalah versi ringkas dari personal computer dengan semua komponen-nya terpaket dalam unit tunggal seukuran koper tipis. Workstation dengan kemampuan input/output grafis resolusi tinggi, sekalipun masih tetap memakai dimensi komputer desktop, namun memiliki daya komputasi yang lebih signifikan daripada personal computer. Workstation seringkali digunakan dalam aplikasi engineering, terutama untuk pekerjaan desain interaktif.

Di atas workstation, terdapat suatu rentang sistem komputer yang luas dan sangat kuat yang di sebut sistem enterprise dan server pada rentang low-end, dan superkomputer pada high-end. Sistem enterprise, atau mainframe, digunakan untuk pengolahan data bisnis pada korporasi menengah hingga besar yang memerlukan lebih banyak daya komputasi dan kapasitas penyimpanan dari yang dapat disediakan oleh workstation. Server berisi unit penyimpanan basis data yang cukup besar dan mampu menangani banyak volume permintaan untuk mengakses data. Pada banyak kasus, server dapat diakses secara luas oleh komunitas pendidikan, bisnis, dan pengguna personal. Permintaan dan respons biasanya ditransportasikan melalui fasilitas komunikasi Internet. Tentu saja, Internet dan server yang berhubungan dengannya telah menjadi sumber segala tipe informasi yang dominan di seluruh dunia. Fasilitas komunikasi Internet terdiri dari suatu struktur kompleks hubungan backbone serat optik kecepatan tinggi yang terinterkoneksi dengan kabel broadcast dan koneksi telepon ke sekolah, perusahaan, dan rumahrumah.

Supercomputer digunakan untuk perhitungan numerik skala besar seperti perkiraan cuaca dan desain dan simulasi pesawat terbang. Dalam sistem enterprise, server, dan superkomputer, unit fungsionalnya, yang meliputi banyak prosesor, dapat terdiri dari sejumlah unit besar dan seringkali terpisah.

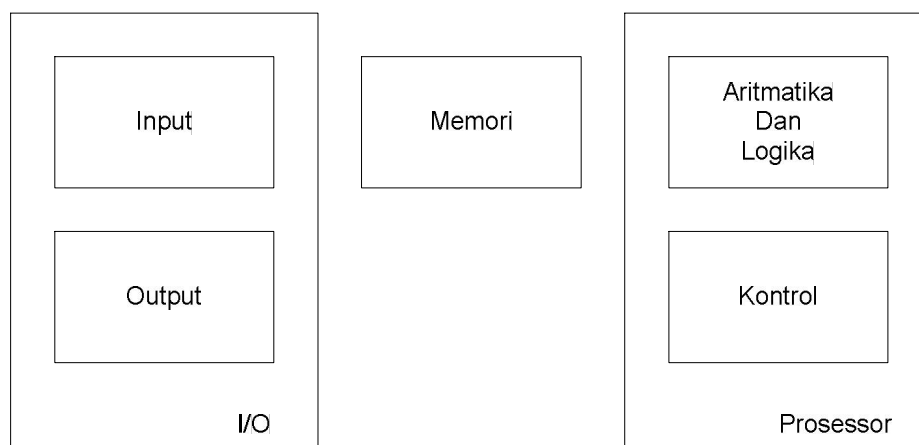
1.2 UNIT FUNGSIONAL

Suatu komputer terdiri dari lima bagian utama yang mandiri secara fungsional: unit input, memori, aritmatika dan logika, output, dan kontrol, sebagaimana ditampilkan pada Gambar 1.1. Unit input menerima informasi terkode dari operator manusia, dari peralatan elektromekanik seperti keyboard, atau dari komputer lain melalui jalur komunikasi digital. Informasi yang diterima disimpan dalam memori komputer untuk referensi selanjutnya atau segera digunakan oleh sirkuit aritmatika dan logika untuk melakukan operasi yang diinginkan. Langkah

pengolahan ditentukan oleh program yang tersimpan dalam memori. Akhirnya, hasil dikirim kembali ke dunia luar melalui unit output. Semua langkah ini dikoordinasikan oleh unit kontrol. Gambar 1.1 tidak menampilkan koneksi di antara unit fungsional. Koneksi tersebut, yang dapat dibuat dengan beberapa cara, dibahas secara menyeluruh di dalam modul ini. Kita mengacu pada sirkuit aritmatika dan logika, dalam hubungannya dengan sirkuit kontrol utama, yaitu prosesor dan peralatan input dan output yang seringkali secara kolektif disebut sebagai unit input-output (I/O).

Sekarang kita mengamati lebih detail informasi yang ditangani oleh suatu komputer. Sangat memudahkan untuk mengkategorikan informasi ini baik sebagai instruksi atau data. Instruksi, atau instruksi mesin, adalah perintah eksplisit yang

- Mengarahkan transfer informasi dalam komputer dan antar komputer dan peralatan I/O-nya
- Menetapkan operasi aritmatika dan logika yang akan dilaksanakan



Gambar 1.1 Unit fungsional dasar suatu komputer.

Daftar instruksi yang melakukan suatu tugas disebut program. Biasanya program tersebut disimpan dalam memori. Prosesor kemudian mengambil instruksi yang membentuk suatu program dari memori, satu demi satu, dan melaksanakan operasi yang diinginkan. Komputer sepenuhnya dikontrol oleh program yang tersimpan tersebut, kecuali adanya kemungkinan interupsi eksternal oleh operator atau oleh peralatan I/O yang terhubung ke mesin tersebut.

Data adalah angka dan karakter ter-encode yang digunakan sebagai operand oleh instruksi. Akan tetapi istilah data, sering digunakan untuk menyebut informasi digital apapun. Dalam definisi data tersebut, seluruh program (yaitu daftar instruksi) dapat dianggap sebagai data jika akan diolah oleh program lain.

Contohnya adalah tugas untuk mengkompilasi program source bahasa tingkat tinggi menjadi daftar instruksi mesin yang merupakan suatu program bahasa mesin, disebut program objek. Program source adalah data input ke program kompilasi yang mentranslasikan program source menjadi program bahasa mesin.

Informasi yang ditangani komputer harus di-encode dalam format yang sesuai. Kebanyakan hardware saat ini menggunakan sirkuit digital yang hanya memiliki dua kondisi stabil, ON dan OFF (lihat Apendiks A). Tiap bilangan, karakter, atau instruksi di-encode sebagai string bit yang disebut bit, masing-masing memiliki dua kemungkinan nilai, 0 atau 1. Bilangan biasanya dinyatakan dalam notasi biner posisi, sebagaimana yang akan dibahas secara detil dalam Bab 2. Kadang-kadang digunakan format binary-coded decimal (BCD), di mana tiap digit desimal di-encode menjadi empat bit.

Karakter alfanumerik juga dinyatakan dalam istilah kode biner. Telah dikembangkan beberapa skema pengkodean. Dua skema yang paling banyak digunakan adalah ASCII (American Standard Coded for Information Interchange), di mana tiap karakter dinyatakan sebagai kode 7bit, dan EBCDIC (Extended Binary-Coded Decimal Interchange Code), di mana digunakan 8 bit untuk menyatakan suatu karakter

1.2.1 UNIT INPUT

Komputer menerima informasi terkodekan melalui unit input, yang membaca data tersebut. Peralatan input yang paling terkenal adalah keyboard. Kapanpun suatu tombol ditekan, huruf atau digit yang sesuai secara otomatis ditranslasikan menjadi kode biner yang tepat dan ditransmisikan melalui suatu kabel ke memori atau ke prosesor.

Tersedia banyak jenis peralatan input lain, termasuk joystick, trackball, dan mouse. Peralatan tersebut seringkali digunakan sebagai peralatan input grafik dalam hubungan dengan display. Mikrofon dapat digunakan untuk menangkap input audio yang kemudian di-sample dan dikonversi menjadi kode digital untuk penyimpanan dan pengolahan. Pembahasan yang lebih detil tentang peralatan input dan operasinya terdapat dalam Bab 12.

1.2.2 UNIT MEMORI

Fungsi unit memori adalah untuk menyimpan program dan data. Terdapat dua kelas penyimpanan, primer dan sekunder.

Penyimpanan primer adalah memori cepat yang beroperasi pada kecepatan elektronik. Program harus disimpan dalam memori tersebut pada saat dieksekusi. Memori tersebut berisi sejumlah besar sel penyimpanan semikonduktor, yang masing-masing mampu menyimpan satu bit informasi. Set-sel tersebut jarang dibaca atau ditulis sebagai sel individual tetapi sebaliknya diolah dalam kelompok dengan ukuran tetap yang disebut word. Memori tersebut terorganisasi sedemikian sehingga isi satu word, yang terdiri dari n bit, dapat disimpan atau diambil dalam satu operasi dasar.

Untuk menyediakan akses yang mudah ke tiap word yang terdapat dalam memori, alamat yang berbeda dihubungkan ke tiap lokasi word. Alamat adalah bilangan yang menyatakan lokasi yang berurutan. Suatu word diakses dengan menyatakan alamatnya dan dengan menyatakan perintah kontrol yang memulai proses penyimpanan atau pengambilan tersebut.

Jumlah bit dalam tiap word sering disebut sebagai panjang word suatu komputer. Panjang word biasanya berkisar dari 16 hingga 64 bit. Kapasitas memori adalah salah satu faktor yang menentukan ukuran komputer. Mesin kecil biasanya hanya memiliki puluhan juta word, sedangkan mesin menengah dan besar biasanya memiliki puluhan atau ratusan juta word. Data biasanya diolah di dalam mesin dalam unit word, kelipatan word, atau pembagian word. Pada saat suatu memori diakses, biasanya hanya satu word data yang dibaca atau ditulis.

Program harus berada di memori selama eksekusi. Instruksi dan data dapat ditulis ke dalam memori atau dibaca dari memori di bawah kontrol prosesor. Sangatlah penting untuk dapat mengakses tiap lokasi word dalam memori secepat mungkin. Memori yang tiap lokasinya dapat dicapai dalam waktu cepat dan tertentu setelah ditetapkan alamatnya disebut random-access memory (RAM). Waktu yang diperlukan untuk mengakses satu word disebut memory access time (waktu akses memori). Waktu tersebut tetap, dan tidak tergantung pada lokasi word yang diakses. Biasanya berkisar dari beberapa nanosecond (ns) hingga sekitar 100 ns untuk unit RAM modern. Memori suatu komputer biasanya diimplementasikan sebagai hierarki memori dari tiga atau empat tingkat unit RAM semikonduktor dengan kecepatan

dan ukuran yang berbeda. Unit RAM yang cepat dan kecil disebut cache. Cache tersebut terangkai erat dengan prosesor dan seringkali termuat pada chip sirkuit terintegrasi yang sama untuk mencapai performa tinggi. Unit yang terbesar dan terlamban disebut sebagai memori utama. Kami akan memberikan deskripsi singkat tentang bagaimana informasi diakses dalam hierarki memori pada bagian berikutnya dari bab ini. Bab 11 membahas aspek operasi dan performa dari memori komputer secara detail.

Sekalipun penyimpanan primer sangat penting, namun cenderung mahal. Jadi penyimpanan sekunder tambahan yang lebih murah digunakan pada saat sejumlah besar data dan banyak program harus disimpan, terutama untuk informasi yang jarang diakses. Terdapat banyak pilihan peralatan penyimpanan sekunder, termasuk disk magnetik dan tape dan disk optikal (CDROM). Peralatan tersebut juga dideskripsikan dalam Bab 15.

1.2.3 UNIT ARITMATIKA DAN LOGIKA

Kebanyakan operasi komputer dieksekusi dalam unit aritmatika dan logika (ALU: arithmetic and logic unit) pada prosesor. Perhatikanlah suatu contoh umum: Misalkan dua bilangan yang berada dalam memori ditambahkan. Bilangan tersebut di bawa ke prosesor, dan penambahan yang sesungguhnya dilakukan oleh ALU. Jumlah tersebut kemudian disimpan dalam memori atau tetap dalam prosesor untuk segera digunakan.

Operasi aritmatika atau logika yang lain, misalnya, perkalian, pembagian, atau perbandingan bilangan, diawali dengan membawa operand yang diperlukan ke prosesor, di mana operasi tersebut dilakukan oleh ALU. Pada saat operand dibawa ke prosesor, operand tersebut disimpan dalam elemen penyimpanan kecepatan tinggi yang disebut register. Tiap register dapat menyimpan satu word data. Waktu akses ke register lebih cepat daripada waktu akses ke unit cache tercepat dalam hierarki memori.

Unit kontrol dan unit aritmatika dan logika jauh lebih cepat daripada peralatan lain yang terhubung ke sistem komputer. Jadi memungkinkan satu prosesor tunggal mengendalikan sejumlah peralatan eksternal seperti keyboard, display, disk magnetik dan optikal, sensor, dan kontroler mekanik.

1.2.4 UNIT OUTPUT

Unit output adalah pasangan unit input. Fungsinya untuk mengirimkan hasil yang telah diproses ke dunia luar. Contoh yang paling umum dari peralatan tersebut adalah printer. Printer menggunakan mechanical head impact, inkjet stream, atau teknik fotokopi, seperti dalam printer laser. untuk melakukan pencetakan. Sangat mungkin untuk menghasilkan tinta yang dapat mencetak sebanyak 10.000 baris per menit. Kecepatan ini luar biasa untuk peralatan mekanik tetapi masih sangat lambat jika dibandingkan dengan kecepatan elektron pada unit prosesor.

Beberapa unit, seperti display grafik, menyediakau fungsi output dan fungsi input. Peran ganda unit tersebut merupakan alasan penggunaan istilah tunggal unit I/O dalam banyak hal.

1.2.5 UNIT KONTROL

Unit memori, aritmatika dan logika, dan input dan output menyimpan dan mengolah informasi dan melakukan operasi input dan output. Operasi unit-unit tersebut harus dikoordinasi dengan beberapa cara. Kooordinasi adalah tugas dari unit kontrol. Unit kontrol secara efektif merupakan pusat saraf yang mengirim sinyal kontrol ke unit lain dan mengetahui keadaan unit tersebut.

Transfer I/O yang terdiri dari operasi input dan output, dikontrol oleh instruksi program I/O yang mengidentifikasi peralatan yang terlibat dan informasi yang ditransfer. Akan tetapi sinyal timing aktual yang mengatur transfer dibangkitkan oleh sirkuit kontrol. Sinyal timing adalah sinyal yang menentukan kapan suatu aksi tertentu dilakukan. Transfer data antara proses set dan memori juga dikontrol oleh unit kontrol melalui sinyal timing. Sangat beralasan untuk memikirkan suatu unit kontrol sebagai unit yang terpisah secara fisik dan terdefinisi dengan jelas yang berinteraksi dengan bagian lain dari mesin. Akan tetapi pada prakteknya, hal tersebut jarang terjadi. Sejumlah besar jalur kontrol (kabel) membawa sinyal yang digunakan untuk timing dan sinkronisasi kejadian disemua unit.

Operasi suatu komputer dapat diringkas sebagai berikut:

- Komputer menerima informasi dalam bentuk program dan data melalui unit input dan menyimpannya dalam memori.

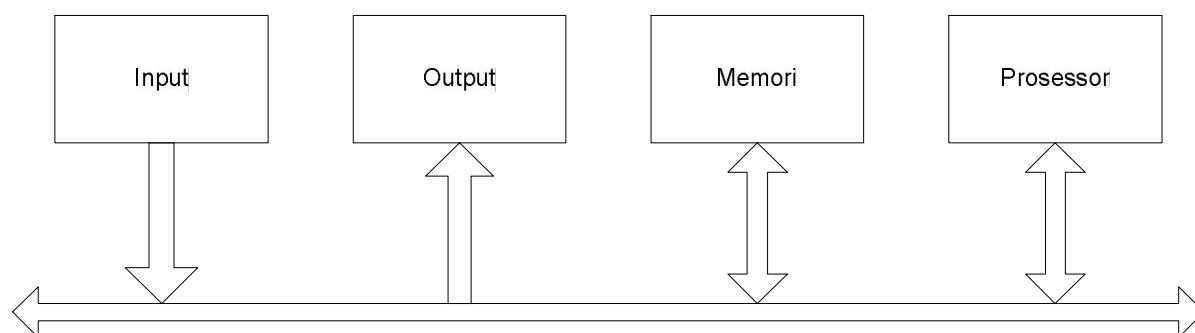
- Informasi yang disimpan dalam memori diambil, di bawah kontrol program, ke unit aritmatika dan logika, di mana informasi tersebut diproses.
- Informasi yang terproses meninggalkan komputer melalui unit output.
- Semua kegiatan di dalam mesin tersebut diarahkan oleh unit kontrol.

1.4 STRUKTUR BUS

Sejauh ini, kita telah membahas fungsi bagian-bagian individu dari suatu komputer. Untuk membentuk suatu sistem operasional, maka bagian-bagian tersebut harus dihubungkan dengan beberapa cara yang terorganisasi. Terdapat banyak cara untuk melakukan hal ini. Disini, kita akan membahas cara yang paling sederhana dan paling umum.

Untuk mencapai kecepatan operasi yang sesuai, komputer harus diorganisasi sehingga semua unitnya dapat menangani satu word data penuh pada waktu tertentu. Pada saat suatu word data ditransfer antar unit, semua bit-nya ditransfer secara paralel, yaitu bit tersebut ditransfer secara simultan melalui banyak kabel, atau jalur, satu bit per jalur. Sekelompok jalur yang berfungsi sebagai jalan penghubung untuk beberapa peralatan disebut bus. Selain jalur yang membawa data, bus harus memiliki jalur untuk alamat dan keperluan kontrol.

Cara yang paling sederhana untuk menginterkoneksi unit fungsional adalah dengan menggunakan bus tunggal, sebagaimana yang ditampilkan dalam Gambar 1.2. Semua unit dikoneksikan ke bus ini. Karena bus tersebut hanya dapat digunakan untuk satu transfer pada satu waktu, maka hanya dua unit yang dapat secara aktif menggunakan bus tersebut pada tiap waktu tertentu. Jalur kontrol bus digunakan untuk mempertimbangkan banyak permintaan terhadap penggunaan bus. Sifat utama struktur bus tunggal adalah biaya rendah dan fleksibilitasnya pada pemasangan peralatan perifer. Sistem yang terdiri dari banyak bus mencapai konkurensi yang lebih dalam operasi dengan memungkinkan dua atau lebih transfer dilakukan pada waktu yang sama. Hal ini menuju kepada performa yang lebih baik tetapi dengan biaya yang lebih besar.



Gambar 1.2. Struktur bus tunggal

Peralatan yang terhubung ke bus sangat bervariasi dalam kecepatan operasinya. Beberapa peralatan elektromekanik, seperti keyboard dan printer, relatif lambat. Peralatan lain, seperti disk magnetik dan optik, dianggap lebih cepat. Memori dan unit prosesor beroperasi pada kecepatan elektronik, menjadikannya sebagai bagian tercepat dalam komputer. Karena semua peralatan tersebut harus berkomunikasi satu sama lain melalui bus, maka diperlukan mekanisme transfer efisien yang tidak dibatasi oleh peralatan yang lambat dan yang dapat digunakan untuk memperkecil perbedaan timing antar prosesor, memori, dan peralatan eksternal.

Pendekatan yang umum adalah dengan menyertakan register buffer pada peralatan yang menyimpan informasi selama transfer. Untuk mengilustrasikan teknik tersebut, perhatikanlah transfer karakter ter-encode dari prosesor ke printer karakter. Karena buffer adalah register elektronik, maka transfer tersebut memerlukan waktu yang relatif sebentar. Pada saat buffer telah di-load, printer dapat mulai mencetak tanpa intervensi lebih lanjut dari prosesor. Bus dan prosesor tidak lagi diperlukan dan dapat dibebaskan untuk aktifitas yang lain. Printer terus mencetak karakter yang terdapat dalam buffer-nya dan tidak tersedia untuk transfer selanjutnya hingga proses ini selesai. Jadi, register buffer memperhalus perbedaan timing antar prosesor, memori, dan peralatan I/O. Register buffer tersebut mencegah prosesor kecepatan tinggi terhalangi oleh peralatan I/O yang lambat selama suatu rangkaian transfer data. Hal ini memungkinkan prosesor untuk berpindah dengan cepat dari satu peralatan ke peralatan yang lain, merangkaikan aktifitas pengolahannya dengan transfer data yang melibatkan beberapa peralatan I/O.

1.5. SOFTWARE

Agar user dapat memasukkan dan menjalankan program aplikasi, maka komputer harus sudah berisi beberapa software sistem dalam memori-nya. Software sistem adalah kumpulan program yang dieksekusi seperlunya untuk menjalankan fungsi seperti

- Menerima dan menginterpretasikan perintah user
- Memasukkan dan mengedit program aplikasi dan menyimpannya sebagai file dalam peralatan penyimpanan sekunder
- Mengatur penyimpanan dan pengambilan file dalam peralatan penyimpanan sekunder
- Menjalankan program aplikasi standar seperti word processor, spreadsheet, atau game, dengan data yang disediakan oleh user
- Mengontrol unit I/O untuk menerima informasi input dan menghasilkan output
- Mentranslasikan program dari bentuk source yang disediakan oleh user menjadi bentuk objek yang berisi instruksi mesin
- Menghubungkan dan menjalankan program aplikasi user-written dengan rutin library standar yang ada, seperti paket komputasi numerik

Software sistem-lah yang bertanggungjawab untuk koordinasi semua aktifitas dalam sistem komputasi. Tujuan bagian ini adalah untuk memperkenalkan beberapa aspek dasar software sistem.

Program aplikasi biasanya ditulis dalam bahasa pemrograman tingkat tinggi seperti C, C++, Java, atau Fortran, di mana programmer yang menentukan operasi matematis atau pengolahan teks-nya. Operasi tersebut dideskripsikan dalam format yang lepas dari penggunaan komputer tertentu untuk mengeksekusi program. Seorang programmer yang menggunakan bahasa tingkat tinggi tidak perlu mengetahui detail instruksi program mesin. Suatu software sistem yang disebut kompilator mentranslasikan bahasa pemrograman tingkat tinggi menjadi program bahasa mesin yang sesuai yang berisi instruksi seperti instruksi Add dan Load.

Program sistem penting lainnya yang digunakan semua programmer adalah editor teks. Program tersebut digunakan untuk memasukkan dan mengedit program aplikasi. User program ini secara interaktif mengeksekusi perintah yang

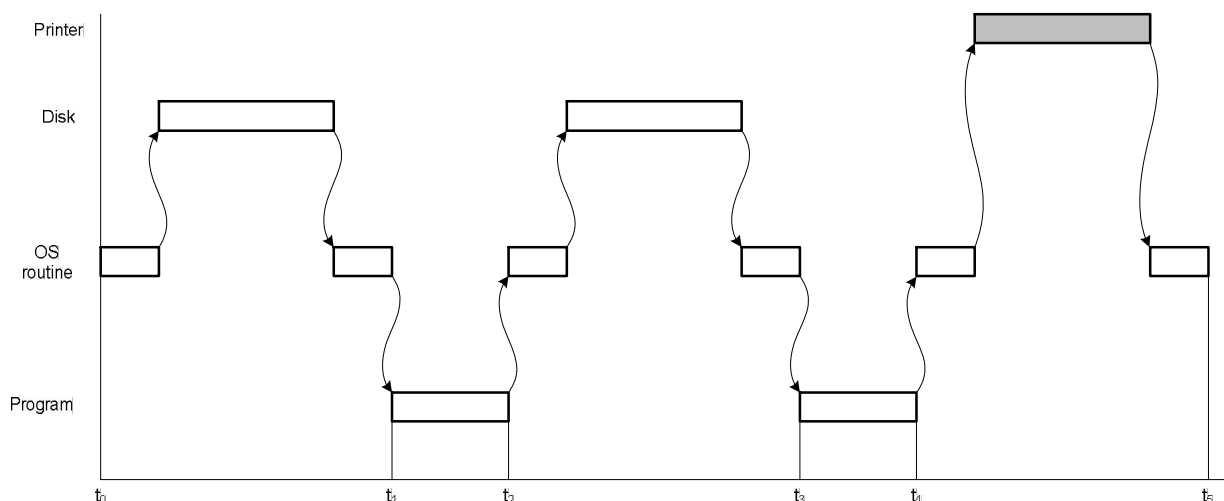
mengijinkan statement program source yang dirnasukkan melalui keyboard diakumulasikan dalam suatu file. File secara sederhana adalah rangkaian karakter alfanumerik atau data biner yang disimpan dalam memori atau dalam penyimpanan sekunder. Suatu file dapat disebut dengan nama yang dipilih oleh user.

Kita tidak membahas detil compiler, editor, atau sistem file dalam modul ini, tetapi marilah kita memperhatikan lebih cermat pada komponen software sistem utama yang disebut sistem operasi (OS: operating system). Ini adalah program yang besar, atau sebenarnya kumpulan rutin, yang digunakan untuk mengontrol pembagian dan interaksi di antara berbagai unit komputer pada saat mereka mengeksekusi program aplikasi. Rutin OS menjalankan tugas yang diperlukan untuk menetapkan resource komputer bagi program aplikasi individu. Tugas-tugas tersebut termasuk menetapkan ruang memori dan disk magnetik untuk program dan file data, memindahkan data antara memori dan unit disk, dan menangani operasi UO.

Untuk memahami sistem operasi dasar, marilah kita membahas suatu sistem dengan Satu prosesor, satu disk, dan satu printer. Pertama-tama kita membahas langkah-langkah yang terlibat dalam menjalankan satu program aplikasi. Setelah kita menjelaskan langkah tersebut, kita akan dapat memahami bagaimana sistem operasi mengatur eksekusi lebih dari satu program aplikasi pada satu waktu. Asumsikan bahwa program aplikasi telah dikompilasi dari bentuk bahasa tingkat-tinggi ke bentuk bahasa mesin dan disimpan dalam disk. Langkah pertama adalah mentransfer file tersebut ke dalam memori. Pada saat transfer selesai, eksekusi program dimulai. Asumsikan bahwa bagian dari tugas program ternasuk membaca file data dari disk ke memori, melakukan beberapa komputasi pada data tersebut, dan mencetak hasilnya. Pada saat eksekusi program mencapai titik di mana file data diperlukan, maka program meminta sistem operasi untuk mentransfer file data dari disk ke memori. OS menjalankan tugas ini dan mengembalikan kontrol eksekusi ke program aplikasi, yang kemudian melanjutkan melakukan komputasi yang diminta. Pada saat komputasi telah selesai dan hasilnya telah siap dicetak, maka program aplikasi mengirim lagi permintaan ke sistem operasi. Rutin OS kemudian dieksekusi agar printer mencetak hasil tersebut.

Kita telah melihat bagaimana kontrol eksekusi diserahkan di antara program aplikasi dan rutin OS. Cara yang mudah untuk mengilustrasikan pembagian

waktu eksekusi processor ini adalah dengan diagram berbasis waktu, seperti yang ditunjukkan pada Gambar 1.3.



Gambar 1.3. Sharing program user dan rutin OS pada processor

Selama periode waktu t_0 hingga t_1 , suatu rutin OS menginisiasi loading program aplikasi dari disk ke memori, menunggu hingga transfer tersebut selesai, dan kemudian menyerahkan kontrol eksekusi ke program aplikasi. Pola aktifitas yang mirip terjadi selama periode t_2 hingga t_3 dan periode t_4 hingga t_5 , pada saat sistem operasi mentransfer file data dari disk dan mencetak hasilnya. Pada t_5 , sistem operasi dapat me-load dan mengeksekusi program aplikasi yang lain.

Sekarang, marilah kita memperhatikan suatu cara di mana resource komputer dapat digunakan lebih efisien jika beberapa program aplikasi diproses. Perhatikanlah bahwa disk dan prosesor diam selama sebagian besar periode waktu t_0 hingga t_5 . Sistem operasi dapat me-load program berikutnya yang akan dieksekusi ke dalam memori dari disk pada saat printer beroperasi. Serupa pula dengan itu, selama t_0 hingga t_1 , sistem operasi dapat mengatur untuk mencetak hasil program sebelumnya pada saat program berikutnya di-load dari disk. Jadi sistem operasi mengatur eksekusi konkuren dari beberapa program aplikasi untuk menghasilkan penggunaan resource komputer terbaik yang dimungkinkan. Pola eksekusi konkuren ini disebut multiprogramming atau multitasking.

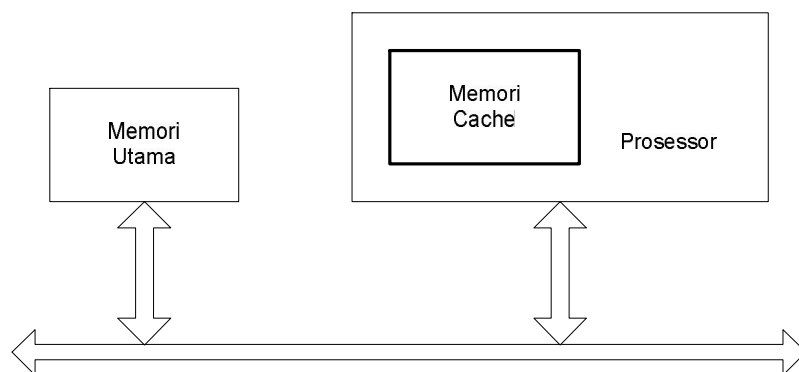
1.6. Performa

Pengukuran performa komputer yang paling penting adalah seberapa cepat komputer tersebut dapat mengeksekusi program. Kecepatan komputer mengeksekusi

program dipengaruhi oleh desain hardware dan instruksi bahasa mesin-nya. Karena program biasanya ditulis dalam bahasa tingkat tinggi, maka performa juga dipengaruhi oleh compiler yang mentranslasikan program ke dalam bahasa mesin. Untuk performa terbaik, perlu untuk mendesain compiler, set instruksi mesin, dan hardware dengan cara yang terkoordinasi. Kita tidak mendeskripsikan detail desain compiler dalam modul ini. Kita berfokus pada desain set instruksi dan hardware.

Pada Bagian 1.5, kita mendeskripsikan bagaimana sistem operasi menumpangtindihkan pengolahan, transfer disk, dan pencetakan untuk beberapa program agar menghasilkan penggunaan terbaik dari resource yang tersedia. Waktu total yang diperlukan untuk mengeksekusi program dalam Gambar 1.4 adalah $t_s - t_o$. Elapsed time ini adalah ukuran performa seluruh sistem komputer. Waktu tersebut dipengaruhi oleh kecepatan prosesor, disk, dan printer. Untuk membahas performa prosesor, kita sebaiknya hanya memperhatikan periode selama prosesor aktif. Periode tersebut adalah periode yang berlabel Program dan Rutin OS pada Gambar 1.3. Kita akan mengacu pada jumlah periode tersebut sebagai waktu prosesor yang diperlukan untuk mengeksekusi program. Selanjutnya kita akan mengidentifikasi beberapa parameter utama yang mempengaruhi waktu prosesor dan mengacu pada bab yang membahas persoalan yang relevan.

Seperti halnya elapsed time untuk eksekusi program tergantung pada semua unit dalam sistem komputer, maka waktu prosesor tergantung pada hardware yang terlibat dalam eksekusi instruksi mesin individu. Hardware tersebut meliputi prosesor dan memori, yang biasanya dihubungkan dengan bus, sebagaimana yang ditunjukkan pada Gambar 1.2. Bagian yang berhubungan dengan Gambar ini diulang dalam Gambar 1.4, termasuk memori cache sebagai bagian dari unit prosesor. Marilah kita memperhatikan aliran instruksi program dan data antara memori dan prosesor. Pada awal eksekusi, semua instruksi program dan data yang diperlukan disimpan di memori utama. Selama eksekusi berjalan, instruksi diambil satu demi satu melalui bus ke dalam prosesor, dan copy-annya diletakkan di cache. Pada saat eksekusi suatu instruksi meminta data yang berada dalam memori utama, data tersebut diambil dan copy-annya ditempatkan di cache. Selanjutnya, jika instruksi atau item data yang sama diperlukan untuk kedua kalinya, maka akan langsung dibaca dari cache.



Gambar 1.4. Cache Processor

Prosesor dan memori cache yang relatif kecil dapat dibuat pada chip sirkuit terintegrasi tunggal. Kecepatan internal untuk melaksanakan langkah dasar pengolahan instruksi pada chip semacam itu sangat tinggi dan dianggap lebih cepat daripada kecepatan pengambilan instruksi dan data dari memori utama. Suatu program akan dieksekusi lebih cepat jika perpindahan instruksi dan data antara memori utama dan prosesor diminimalisasi, yang dicapai dengan menggunakan cache. Misalnya, suatu instruksi dieksekusi berulang kali selama periode waktu yang singkat, sebagaimana yang terjadi pada loop program. Jika instruksi ini tersedia dalam cache maka dapat diambil dengan cepat selama periode pengulangan penggunaan. Hal yang sama diterapkan pada data yang digunakan berulang kali. Persoalan desain, operasi, dan performa untuk memori utama dan cache dibahas dalam Bab 11.

1.6.1 CLOCK PROSESOR

Sirkuit prosesor dikontrol oleh sinyal timing yang disebut clock. Clock menetapkan interval, waktu reguler, yang disebut siklus clock. Untuk mengeksekusi instruksi mesin, prosesor membagi tindakan yang akan dilakukan ke dalam rangkaian langkah dasar, sehingga tiap langkah dapat diselesaikan dalam satu siklus clock. Panjang P dari satu siklus clock adalah parameter penting yang mempengaruhi performa prosesor. Kebalikannya adalah clock rate, $R = 1/P$, yang diukur dalam siklus per detik. Prosesor yang digunakan dalam personal computer dan workstation saat ini memiliki clock rate yang berada dalam rentang beberapa ratus juta hingga lebih dari milyaran siklus per detik. Dalam terminologi teknik elektro standar, istilah "siklus per detik" disebut hertz (Hz). Istilah "juta" ditunjukkan oleh awalan Mega (M). Dan "milyar" ditunjukkan oleh awalan Giga (G).

Karena itu 500 juta siklus per detik biasanya disingkat menjad 500 Megahertz (MHz), dan 1250 juta siklus per detik disingkat menjadi 1,25 Gigahertz (GHZ). Periode clock yang sesuai masing-masing adalah 2 dan 0,8 nanosecond (ns).

1.6.2 PERSAMAAN PERFORMA DASAR

Sekarang kita memusatkan perhatian pada komponen waktu prosesor dari elapsed time total. Misalkan T adalah waktu prosesor yang diperlukan untuk mengeksekusi suatu program yang telah dipersiapkan dalam beberapa bahasa tingkat tinggi. Compiler menghasilkan program objek bahasa mesin yang sesuai dengan program source. Asumsikan bahwa eksekusi lengkap dari program memerlukan N instruksi bahasa mesin. Jumlah N adalah jumlah aktual eksekusi instruksi, dan tidak harus setara dengan jumlah eksekusi instruksi mesin dalam program objek. Beberapa instruksi dapat dieksekusi lebih dari sekali, yaitu untuk instruksi yang berada di dalam loop. Instruksi yang lain mungkin tidak dieksekusi sama sekali, tergantung data input yang digunakan. Misalkan jumlah langkah dasar rata-rata yang diperlukan untuk mengeksekusi satu instruksi mesin adalah S , di mana tiap langkah dasar diselesaikan dalam satu siklus clock. Jika clock rate adalah R siklus per detik, maka waktu eksekusi program dinyatakan sebagai berikut

$$T = \frac{N \times S}{R}$$

Rumusan ini sering disebut sebagai persamaan performa dasar.

Parameter performa T untuk program aplikasi jauh lebih penting bagi user daripada nilai individu parameter N , S , atau R . Untuk mencapai performa tinggi, desainer komputer harus mencari cara untuk mengurangi nilai T , yang berarti mengurangi N dan S , dan meningkatkan R . Nilai N berkurang jika program source dikompilasi menjadi instruksi mesin yang lebih sedikit. Nilai S berkurang jika instruksi memiliki jumlah langkah dasar yang lebih kecil untuk dilaksanakan atau jika eksekusi instruksi ditumpangtindihkan. Menggunakan clock frekuensi tinggi meningkatkan nilai tersebut atau R , yang berarti bahwa waktu yang diperlukan untuk menyelesaikan langkah eksekusi dasar berkurang.

Kita harus menekankan bahwa N , S , dan R bukanlah parameter bebas; perubahan pada salah satunya dapat mempengaruhi yang lain. Memperkenalkan

fitur baru dalam desain suatu prosesor akan menuju pada peningkatan performa hanya jika hasil keseluruhannya mengurangi nilai T Prosesor yang diiklankan memiliki clock 900Mhz belum tentu memberikan performa yang lebih baik daripada prosesor 700Mhz karena prosesor tersebut mungkin memiliki nilai S yang berbeda.

1.6.3 OPERASI PIPELINING DAN SUPERSCALAR

Dalam pembahasan di atas, kita mengasumsikan bahwa instruksi dieksekusi satu demi satu. Karena itu, nilai S adalah jumlah total langkah dasar, atau siklus clock, yang diperlukan untuk mengeksekusi suatu instruksi. Peningkatan yang substansial pada performa dapat dicapai dengan menumpangtindihkan eksekusi instruksi yang berurutan, menggunakan teknik yang disebut pipelining. Misalkan suatu instruksi

Add R1, R2, R3

yang menambahkan isi register R1 dan R2, dan menempatkan jumlahnya dalam R3. Isi R1 dan R2 mula-mula ditransfer ke input ALU. Setelah operasi penambahan dilakukan, jumlahnya ditransfer ke R3. Prosesor dapat membaca instruksi selanjutnya dari memori sementara operasi penambahan dilakukan. Kemudian jika instruksi tersebut juga menggunakan ALU, operandnya dapat ditransfer ke input ALU pada waktu yang sama dengan hasil instruksi Add ditransfer ke R3. Pada kasus ideal, jika semua instruksi ditumpangtindihkan ke derajat maksimum yang mungkin, maka eksekusi dilanjutkan pada kecepatan penyelesaian satu instruksi dalam tiap siklus clock. Instruksi individual masih memerlukan beberapa siklus clock untuk penyelesaian. Tetapi, untuk tujuan perhitungan T , maka nilai efektif S adalah 1.

Pipelining dibahas secara detil dalam Bab 8. sebagaimana yang akan kita lihat, nilai ideal $S = 1$ tidak dapat dicapai dalam praktek karena berbagai alasan. Akan tetapi, pipelining meningkatkan kecepatan eksekusi instruksi secara signifikan dan menyebabkan nilai efektif S mendekati 1.

Derajat konkurensi yang lebih tinggi dapat dicapai jika banyak pipeline instruksi diterapkan pada prosesor. Hal ini berarti digunakannya banyak unit fungsional, menciptakan jalur paralel di mana berbagai instruksi yang berbeda dapat dieksekusi secara paralel. Dengan pengaturan tersebut, maka dimungkinkan untuk memulai beberapa instruksi pada tiap siklus clock. Mode operasi ini disebut

eksekusi superscalar. Jika mode ini dapat bertahan dalam waktu lama selama eksekusi program, maka nilai efektif S dapat dikurangi hingga kurang dari satu. Tentu saja, eksekusi paralel harus mempertahankan kebenaran logika program, sehingga hasil yang diperoleh harus sama dengan hasil dari eksekusi serial instruksi program. Banyak dari prosesor performa tinggi saat ini didesain untuk bekerja dengan cara tersebut.

1.6.4 CLOCK RATE

Terdapat dua kemungkinan untuk meningkatkan clock rate, R . Pertama, meningkatkan teknologi integrated-circuit (IC) menjadikan sirkuit logika yang lebih cepat, sehingga mengurangi waktu yang diperlukan untuk menyelesaikan suatu langkah dasar. Hal ini memungkinkan periode clock, P , dikurangi dan clock rate, R , ditingkatkan. Kedua, mengurangi periode clock, P . Akan tetapi jika tindakan yang harus dilakukan oleh suatu instruksi tetap sama, maka jumlah langkah dasar yang diperlukan dapat bertambah.

Peningkatan nilai R yang sepenuhnya disebabkan oleh peningkatan teknologi IC mempengaruhi secara seimbang seluruh aspek operasi prosesor dengan pengecualian pada waktu yang diperlukan untuk mengakses memori utama. Dengan adanya cache, persentase akses ke memori utama menjadi kecil. Karena itu kebanyakan peningkatan performa yang diharapkan dari penggunaan teknologi yang lebih cepat dapat direalisasikan. Nilai T akan berkurang dengan faktor yang sama dengan peningkatan nilai R karena S dan N tidak terpengaruh. Pengaruh pada performa dari perubahan cara pembagian instruksi menjadi langkah dasar lebih sulit untuk diperkirakan. Persoalan ini dibahas dalam Bab 8.

1.6.5. SET INSTRUKSI: CISC DAN RISC

Instruksi sederhana memerlukan eksekusi sejumlah kecil langkah dasar. Instruksi kompleks melibatkan sejumlah besar langkah. Untuk prosesor yang hanya memiliki instruksi sederhana, sejumlah besar instruksi mungkin diperlukan untuk menjalankan suatu tugas pemrograman tertentu. Hal ini dapat menuju pada nilai N yang besar dan nilai S yang kecil. Sebaliknya, jika instruksi individu melaksanakan operasi yang lebih kompleks, maka diperlukan instruksi yang lebih sedikit, menuju

pada nilai N yang lebih rendah dan nilai S yang lebih besar. Tidak tampak jelas bahwa satu pilihan lebih baik dari pilihan yang lain.

Pertimbangan utama dalam memperbandingkan dua pilihan adalah penggunaan pipelining, Kita telah menjelaskan sebelumnya bahwa nilai efektif S dalam prosesor ter-pipeline mendekati 1 sekalipun jumlah langkah dasar per instruksinya mungkin dapat dianggap lebih besar. Hal ini tampaknya menyatakan secara tidak langsung bahwa instruksi kompleks dikombinasikan dengan pipelining akan mencapai performa terbaik. Akan tetapi, lebih mudah untuk mengimplementasikan pipelining yang efisien dalam prosesor dengan set instruksi yang sederhana. Kesesuaian set instruksi tersebut untuk eksekusi pipeline adalah pertimbangan yang penting dan seringkali menentukan.

Desain set instruksi suatu prosesor dan pilihan yang tersedia dibahas dalam Bab 2. Manfaat relatif prosesor dengan instruksi sederhana dan prosesor dengan instruksi yang lebih kompleks telah dipelajari secara luas. Pendahulunya disebut Reduced Instruction Set Computers (RISC), dan yang terbaru disebut Complex Instruction Set Computers (CISC). Sekalipun menggunakan istilah RISC dan CISC agar kompatibel dengan deskripsi kontemporer, kami memperingatkan pembaca untuk tidak berasumsi bahwa keduanya secara nyata mendefinisikan kelas-kelas prosesor. Suatu desain prosesor tertentu adalah hasil dari banyak pertukaran. Istilah RISC dan CISC mengacu pada prinsip dan teknik desain di beberapa bagian dalam modul

1.6.6 COMPILER

Compiler mentranslasikan bahasa pemrograman tingkat tinggi menjadi rangkaian instruksi mesin. Untuk mengurangi N , kita perlu memiliki set instruksi mesin yang sesuai dan compiler yang dapat menggunakannya dengan baik. Suatu optimizing compiler memanfaatkan berbagai fitur prosesor target untuk mengurangi hasil kali $N \times S$, yang merupakan jumlah total siklus clock yang diperlukan untuk mengeksekusi suatu program. Kita akan melihat pada Bab 8 bahwa jumlah siklus tidak hanya tergantung pada pilihan instruksi, tetapi juga pada urutan instruksi tersebut muncul dalam program. Compiler dapat menata ulang instruksi program untuk mencapai performa yang lebih baik. Tentu saja, perubahan semacam itu tidak harus mempengaruhi hasil komputasi.

Dari luar, compiler tampak sebagai entitas terpisah dari prosesor yang digunakannya dan mungkin bahkan berasal dari vendor yang berbeda. Akan tetapi, compiler kualitas tinggi harus dihubungkan dengan erat pada arsitektur prosesor. Compiler dan prosesor seringkali didesain pada waktu yang sama, dengan banyak interaksi antara desainer untuk mencapai hasil terbaik. Tujuan akhirnya adalah untuk mengurangi jumlah total siklus clock yang diperlukan untuk melakukan tugas pemrograman yang diminta.

1.6.7 PENGUKURAN PERFORMA

Sangat penting untuk dapat memperkirakan perfonna suahr komputer. Desainer komputer menggunakan perkiraan perfonna untuk mengevaluasi keefektifan fitur barn. Produsen menggunakan indikator performa dalarn proses pemasaran. Pembeli menggunakan data tersebut untuk memilin di antara model komputer yang tersedia.

Pembahasan sebelumnya menyarankan bahwa parameter satu-satunya yang mendeskripsikan secara tepat performa suatu komputer adalah waktu eksekusi, T , untuk program tertentu. Meskipun Persamaan 1.1 sederhana secara konseptual, namun menghitung nilai T tidak sederhana. Lagi pula, parameter seperti kecepatan clock dan berbagai fitur arsitektural bukan indikator yang handal dari performa yang diharapkan.

Untuk alasan tersebut, komunitas komputer mengadopsi ide pengukuran performa komputer menggunakan program benchmark. Untuk memungkinkan perbandingan tersebut, maka harus dibuat program yang terstandarisasi. Pengukuran perfonna adalah waktu yang diperlukan suatu komputer untuk mengeksekusi benchmark tertentu. Pada awalnya, beberapa pendekatan dibuat untuk menghasilkan program artifisial yang dapat digunakan sebagai benchmark standar. Tetapi program sintetik tidak dapat memprediksi perfonna yang diperoleh dengan tepat pada saat program aplikasi riil dijalankan.

Praktek yang diterima saat ini adalah menggunakan pilihan program aplikasi riil yang disetujui untuk mengevaluasi performa. Suatu organisasi nirlaba yang disebut System Performance Evaluation Corporation (SPEC) memilih dan mempublikasikan program aplikasi yang representatif untuk domain aplikasi yang berbeda, bersama dengan hasil tes untuk banyak komputer yang tersedia secara

komersial. Untuk komputer general-purpose, sekelompok program benchmark dipilih pada tahun 1989. Program tersebut dimodifikasi dan dipublikasikan pada tahun 1995 dan dimodifikasi lagi pada tahun 2000.

Program-program yang dipilih berada dalam rentang dari mulai memainkan game, compiler, dan aplikasi basis data hingga program yang intensif secara numerik dalam astrofisika dan kimia kuantum. Pada tiap kasus, program tersebut dikompilasi untuk computer under test, dan dilakukan pengukuran running time-nya pada komputer riil. Program yang sama juga dikompilasi dan dijalankan pada satu komputer yang dipilih sebagai referensi. Untuk SPEC95, referensinya adalah SUN SPARCstation 10/40. Untuk SPEC2000, komputer referensinya adalah workstation Ultra-SPAKCIO dengan prosesor 300-MHz UltraSPARC-III. SPEC rating dihitung sebagai berikut

$$SPEC = \frac{\text{Runningtime pada komputer referensi}}{\text{Runningtime pada komputer under test}}$$

Jadi SPEC rating 50 berarti computer under test 50 kali lebih cepat dari UltraSPARC10 untuk benchmark tersebut. Tes tersebut diulang untuk semua program dalam kelompok SPEC, dan dilakukan perhitungan rata-rata geometrik dari hasil tersebut. Misalkan SPEC adalah rating untuk program i dalam kelompok tersebut. Keseluruhan SPEC rating untuk komputer tersebut dinyatakan

$$SPEC \text{ rating} = \left(\prod_{i=1}^n SPEC_i \right)^{\frac{1}{n}}$$

Dimana n adalah jumlah program dalam kelompok tersebut

Karena waktu eksekusi aktual adalah terukur, maka SPEC rating adalah ukuran dari efek gabungan semua faktor yang mempengaruhi performa, termasuk compiler, system operasi, prosesor, dan memori computer yang sedang diuji.

1.7. MULTIPROSESOR DAN MULTIKOMPUTER

Sejauh ini, kita telah membahas komputer dengan satu prosesor. Sistem komputer besar dapat berisi sejumlah unit prosesor, yang disebut sistem multiprosesor. Sistem tersebut mengeksekusi sejumlah tugas eksekusi yang berbeda secara paralel, atau mengeksekusi sub tugas dari suatu tugas besar tunggal secara paralel. Semua prosesor biasanya memiliki akses ke semua memori dalam sistem semacam itu, dan istilah sistem multiprosesor shared-memory sering digunakan

untuk memperjelas sistem ini. Performa tinggi sistem tersebut muncul bersama dengan peningkatan kompleksitas dan biaya. Selain banyak prosesor dan unit memori, biaya meningkat karena adanya kebutuhan jaringan interkoneksi yang lebih kompleks.

Berbeda dengan sistem multiprosesor, dimungkinkan penggunaan kelompok komputer lengkap yang terinterkoneksi untuk mencapai daya komputasi total yang tinggi. Komputer tersebut biasanya hanya memiliki akses ke unit memornya sendiri. Pada saat tugas yang dieksekusinya perlu untuk mengkomunikasikan data, maka komputer tersebut melakukannya dengan mengirimkan pesan melalui jaringan komunikasi. Sifat ini membedakannya dari multiprosesor shared-memory, dengan nama multikomputer message-passing.