

## Bab 1 Pendahuluan

Perangkat komputer yang memiliki kualitas baik berarti sistem di dalamnya bekerja sesuai dengan fungsinya, sesuai dengan rentang operasional yang dirancang atau dapat dikatakan bekerja sesuai kriteria yang diinginkan. Analisa kualitas dalam sistem dapat digolongkan menjadi tiga yaitu:

- a. Kualitas Desain, yaitu penilaian terhadap desain atau fitur yang ada dalam suatu perancangan sistem. Misal jika itu sistem komputer maka berarti disain komponen-komponen arsitektur dari sistem tersebut, atau jika itu sistem pengukuran berarti komponen alat ukur, komunikasi dan penyimpanan yang dipilih
- b. Kualitas Conformity, yaitu penilaian mutu sesuai dengan acuan standar yang harus dipenuhi. Misal pada kondisi soft realtime, suatu mesin anjungan tunai mandiri (ATM) saat memproses pembayaran kartu kredit memberikan waktu terlama suatu proses pembayaran adalah 2 menit setelah memasukkan nomor rekening pembayaran menurut standar transaksi online ATM. Jika sistem ATM memenuhi nilai 2 menit ini berarti sistem sudah memproses sesuai standar transaksi online untuk ATM
- c. Kualitas Kinerja, yaitu penilaian kualitas berdasarkan beban yang diberikan. Jika suatu sistem dapat berfungsi dengan baik pada beban yang dirancang berarti kinerjanya baik. Konsistensi suatu kinerja sistem terhadap beban yang diberikan disebut dengan kehandalan (reliabilitas)

Pada matakuliah ini komponen kualitas yang dibahas adalah komponen kinerja. Dalam bab-bab berjalan akan diceritakan bagaimana sistem memiliki kinerja yang teruji oleh waktu dan beban yang diberikan. Beban atau dalam bahasa Inggris disebut load adalah rentang input yang dirancang untuk mampu diterima oleh sistem dengan sistem tetap berfungsi sebagaimana seharusnya.

Misal suatu sistem server komputer dirancang untuk web server sistem perwalian online dengan sekali perwalian, sekitar 1-300 orang dapat login dan melakukan registrasi perwalian dengan kinerja optimal. Jika jumlah mahasiswa yang perwalian dalam satu waktu melebihi dari 300 orang, sistem akan memiliki kinerja tidak sempurna atau tidak berfungsi penuh bahkan malah mati sama sekali.

### 1.1. Mengapa perlu mengetahui kinerja sistem?

Jika ditanya kepentingan apa suatu sistem kinerjanya harus diketahui atau dimonitoring setiap saat, tentunya jawabannya berdasarkan suatu kondisi yang ingin dijaga atau dipelihara. Prinsip dari mengetahui, memantau kinerja sistem adalah terdapat kondisi sistem yang tidak boleh terjadi, jadi alasannya bisa berupa konsistensi sistem, keamanan, keterkontrolan, kepuasan layanan bagi pengguna.

Jika dirangkum alasan mengapa suatu sistem diukur kinerjanya, yaitu:

- a. Mengetahui konsistensi sistem
- b. Mengetahui batas toleransi sistem
- c. Mengetahui kondisi optimum sistem
- d. Mengukur ketahanan sistem terhadap perubahan yang disebabkan oleh lingkungan di luar sistem
- e. Menentukan umur perawatan atau pemeliharaan dari sistem
- f. Mengetahui tingkat keamanan sistem
- g. Menjamin kehandalan dan availabilitas sistem

Dalam kondisi dilematis mana yang mau dipilih, sistem dengan kinerja yang tidak konsisten atau sistem yang kinerjanya konsisten pada kualitas rendah? Sebagai contoh jika anda memiliki modem, mana yang akan dipilih modem dengan kecepatan selalu bervariasi, kadang tinggi dan

kadang rendah atau modem tanpa variasi tetapi memiliki kecepatan konsisten di suatu nilai yang rendah di bawah standar?

Pelapukan atau dalam ilmu sistem komputer disebut dengan penurunan kinerja sistem karena usia alat perlu selalu diamati. Tidak ada alat elektronik, mekanik atau hidrolis yang tidak berubah kinerjanya terhadap waktu. Selain dari usia alat waktu lingkungan tempatnya bekerja juga mempengaruhi. Komputer dengan sistem virtual memory membutuhkan memori bebas pada harddisk sesuai dengan yang dirancangan. Karena pemakaian, memori bebas di harddisk secara alami akan selalu berkurang. Pada saat terjadi proses swapping yaitu proses memindahkan data dari memory RAM ke harddisk maka diperlukan alokasi memori yang sesuai. Jika memori bebas di harddisk tidak cukup atau pemetaan virtual harddisknya tidak baik. Maka proses penyelesaian suatu program akan terhambat.

## 1.2. Mekanisme Mengukur Kinerja Sistem

Sebelum melakukan analisa kinerja dari suatu sistem perlu kita mengetahui komponen mengukur kinerja sistem yang berupa besaran yang diukur, alat ukur, mekanisme pengukuran, mekanisme pencatatan. Besaran yang diukur sering disebut dengan parameter kinerja nama lainnya adalah metriks. Biasanya metriks merupakan keluaran dari suatu sistem. Misalnya pada sistem pengiriman data matrik yang diukur bisa berupa jumlah data terkirim, laju data yang dikirim, lama pengiriman. Jika suatu metriks berupa hasil dari suatu sistem yang diukur berdasarkan lajunya biasanya disebut dengan *throughput*.

Alat ukur bisa berupa algoritma yang dipasang untuk mengukur kejadian yang terjadi, misal untuk mengukur sistem pengiriman data dibuat sistem clock yang menghitung jumlah data yang sukses dikirim tanpa meminta pengiriman ulang (resend).

Mekanisme pengukuran kinerja adalah mekanisme yang menceritakan ide memasang alat ukur, memilih metriks dan bagaimana analisa kinerja akan dilakukan. Ada mekanisme pengukuran kinerja yang mengharuskan adanya stimulasi perubahan beban yang sengaja diciptakan untuk menguji kinerja sistem. Ada juga mekanisme yang tidak perlu menciptakan stimulasi beban untuk menguji atau mengukur sistem. Mekanisme ini juga merancang strategi pengukuran kinerja, tahapan, pewaktu dan perlindungan dari pengaruh diluar sistem yang bisa mengganggu proses pengukuran kinerja. Besaran statistik terkadang diperlukan pada saat melakukan analisa dari metriks yang telah diperoleh.

Pada komputer jaman dulu, saat komputer dinyalakan (bootstrap), prosesor mengecek status valid periferalnya seperti mengecek memori RAM secara sekuensial, keberadaan perangkat I/O. Jika ada yang tidak memberikan status ready maka program di BIOS akan menyatakan adanya ketidak valid-an atau kesiapan alat. Ini merupakan proses pengukuran kinerja yang didasari oleh kesiapan dan kevaliditasan perangkat.

Mekanisme pencatatan dari metriks yang diukur juga perlu diperhatikan seperti bentuk data, blok data yang akan disimpan, cara menyimpannya apakah langsung di Memori SD atau di flash disk atau dikirim lewat internet ke server data, sampling atau waktu pengambilan per data.

Bentuk pengamatan atau monitoring mempengaruhi mekanisme pencatatan, beberapa perangkat komputer ada yang bisa mencatat dan memonitoring secara otomatis.

## 1.3. Beban Sistem

Beban kerja suatu sistem atau yang disebut workload merupakan batasan dari rentang input suatu sistem yang menjadi dasar sistem tersebut dikatakan kinerjanya baik. Beban sendiri sebenarnya memiliki beberapa jenis ada beban operasional, beban berlebih, beban kontinu, beban sesaat. Harapan dari perancang sistem adalah saat diberikan beban operasional, sistem akan bekerja dan menghasilkan output sesuai dengan rancangan. Jika beban berlebihan maka sistem akan terganggu, kinerja bisa tidak sesuai. Ilmu yang mempelajari beban dan

perencanaan beban agar terhindar dari kinerja sistem yang buruk disebut dengan Perencanaan Kapasitas atau Capacity Planning. Terkadang untuk mengetahui kinerja sistem, sistem sengan diberi input uji atau disebut beban uji. Misal sistem sengaja diberikan beban sebesar 99% beban penuh, dan kemudian dilakukan monitoring apakah kinerja sistem masih dalam ambang toleransi kesalahan (Fault Tolerance) atau sudah diluar ambang ini.

#### **1.4. Sistem yang perlu di analisa**

Sistem yang dianalisa pada Teknik Komputer adalah sistem perangkat keras seperti server, jaringan komputer, prosesor, alat ukur, alat kontrol, alat komunikasi. Pada intinya semua peralatan yang kinerjanya perlu dipertahankan dan dipantau adalah sistem yang tepat untuk dianalisa. Kemampuan analisa yang diperlukan adalah kemampuan dalam membaca grafik output, mengartikan besaran statistik dari metriks.

#### **1.5. Aktivitas dalam melakukan analisa kinerja sistem**

Terdapat beberapa aktivitas yang dilakukan saat menganalisa kinerja sistem yaitu:

- a. Melakukan spesifikasi atau kriteria dari kinerja yang diperlukan
- b. Melakukan evaluasi kinerja berdasarkan beban dan usia
- c. Membandingkan dua atau lebih kinerja sistem
- d. Menentukan kinerja optimal
- e. Melakukan identifikasi *bottleneck*
- f. Menentukan beban kerja operasional
- g. Merencanakan manajemen kapasitas
- h. Memprediksi kinerja sistem

##### **a. Melakukan spesifikasi atau kriteria dari kinerja yang diperlukan**

Saat mempergunakan sistem perlu ditentukan kriteria kinerja yang sesuai dengan kebutuhan, penggunaan sistem dan kondisi yang menghambat kinerja tersebut. Sebagai contoh peralatan komputer yang dipergunakan untuk bergerak (mobile) tentunya tidak sama kriterianya untuk peralatan yang diam. Iklim juga mempengaruhi pemilihan perangkat komputer. Komputer untuk bermain game online tentunya memiliki kriteria yang berbeda dengan komputer yang hanya digunakan untuk mengetik surat menyurat. Dibutuhkan VGA Card, memory yang besar dan prosesor yang cepat untuk mengatasi proses rendering dari game online.

##### **b. Melakukan evaluasi kinerja berdasarkan beban dan usia**

Komputer yang digunakan untuk mengerjakan beberapa tugas (multi tasking) tentunya akan lebih terbebani karena proses alokasi sumber daya yang perlu dilakukan untuk proses multi tasking tersebut. Server data akan sangat sibuk jika melayani beberapa permintaan proses transaksi secara simultan, program perwalian online yang digunakan oleh 20 mahasiswa akan berbeda bebannya terhadap server jika digunakan oleh 2000 mahasiswa secara bersamaan.

Pada peralatan yang kinerjanya dipengaruhi oleh usia penggunaan, beban yang sama akan memberikan kinerja yang berbeda untuk sistem yang sudah lama. Suatu laptop yang sudah lama digunakan memiliki beberapa program yang membebani seperti malware, program yang mengecek register, program yang menetap di memori RAM. Sehingga laptop yang sudah lama tanpa pengorganisasian memori akan lebih lambat saat penyalaan komputer (bootstrap) maupun saat keluar dari sistem.

Melalui evaluasi akan diperoleh beban yang memberikan kinerja maksimum, minimum. Beban yang memberikan kinerja maksimum disebut dengan beban optimum, beban ini dapat dicari melalui percobaan dengan cara membebani sistem secara perlahan mulai dari nilai terkecil sampai nilai terbesar atau sebaliknya. Dalam sistem pengukuran jika kurva beban naik dan beban turun berbeda maka disebut terjadi histerisis. Proses mempertahankan kinerja maksimum dengan mengatur beban optimum disebut dengan penyeimbangan beban (*Load Balancing*). Istilah Load Balancing banyak ditemui pada matakuliah pemrosesan paralel, yaitu saat sistem operasi membagi input pada dua prosesor disesuaikan dengan beban optimum masing-masing prosesor.

### **c. Membandingkan dua atau lebih kinerja sistem**

Salah satu pekerjaan dari seorang Teknik Komputer adalah melakukan perbandingan kinerja dari dua sistem sejenis untuk menilai sistem mana yang lebih baik dan sesuai. Jika perbandingan kinerja tersebut dilakukan terhadap sistem baku atau standar, maka proses ini disebut dengan *benchmarking*. Benchmarking biasanya membandingkan dua metrik keluaran menggunakan beban yang sama, beberapa asosiasi komputer dan perusahaan komputer memberikan hasil perbandingan dalam bentuk score atau level.

### **d. Menentukan kinerja optimal**

Kinerja optimal adalah kinerja terbaik dari sistem pada saat diberikan beban dan berada pada lingkungan yang terkontrol. Istilah kinerja optimal biasanya diberikan pada saat sistem dalam keadaan tertekan oleh beban yang tinggi atau kondisi luar yang sangat mempengaruhi parameter sistem dan cenderung kondisi tersebut mengurangi kinerja sistem. Pada beban normal, kinerja optimal juga ditentukan oleh parameter sistem yaitu variabel internal di dalam sistem yang mempengaruhi keluaran atau metrik. Proses tuning di dalam komputer adalah proses pencarian dan penentuan parameter sistem komputer yang terbaik, misalnya untuk sistem operasi DOS masih diperlukan pengaturan memori buffer, cache software, jumlah files yang ditampung di dalam proses multitasking di RAM, program-program apa saja yang menetap di konvensional memori.

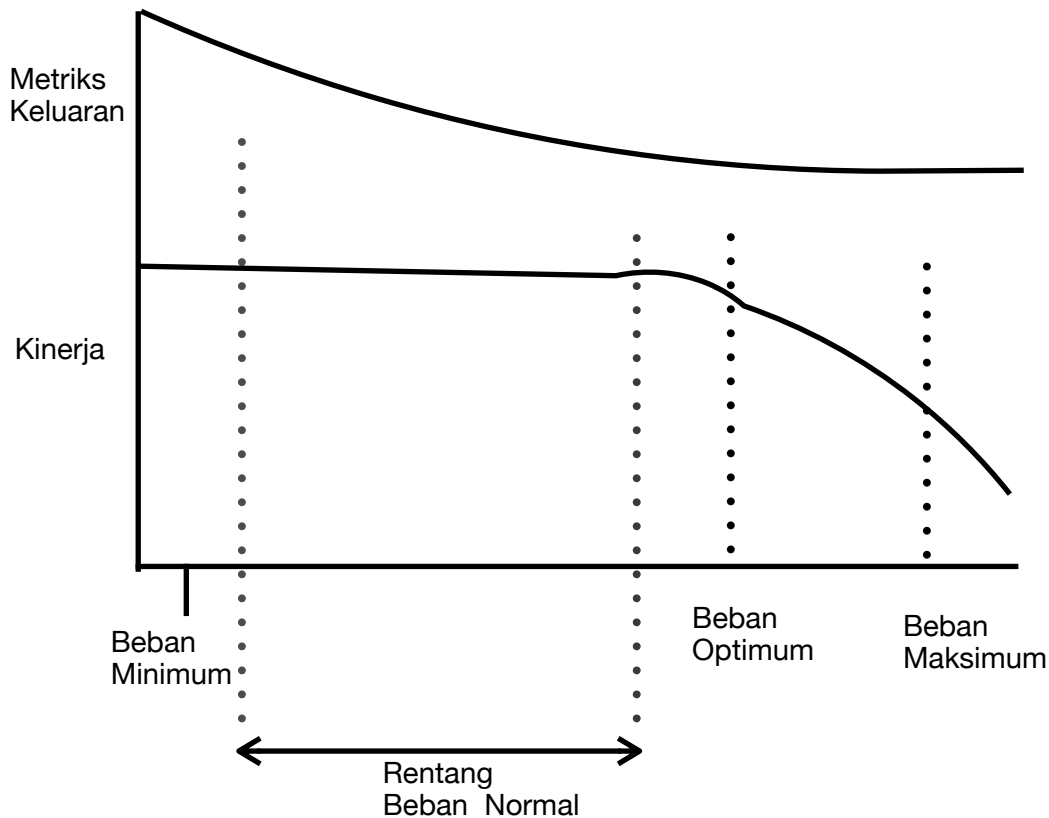
### **e. Melakukan identifikasi bottleneck**

Bottleneck adalah kinerja yang disebabkan oleh adanya antrian sumber daya yang tidak dapat disela (non pre-emptive). Pada kondisi bottleneck, sistem memiliki kinerja yang sukar untuk dicari beban optimumnya, kinerja sistemnya relatif konstan dan tidak tinggi dikarenakan adanya antrian sumber daya tersebut. Pada matakuliah sistem operasi suatu sumber daya yang diperlukan jika belum berstatus *Release* maka tidak dapat dipergunakan oleh program lain. Jika sumberdaya tersebut memiliki siklus Ready-Used-Released yang berdekatan atau dengan kata lain penggunaan sumber daya tersebut sangat sering maka sumber daya tersebut menjadi antrian bagi program-program yang menunggunya. Kondisi ini mirip air yang dikeluarkan dari botol, karena permukaan botol yang kecil maka keluaran atau air tersebut akan perlahan-lahan dan kita tidak bisa melakukan apapun untuk mengubahnya selain menunggunya. Contoh dari sistem ini adalah Port Printer, meskipun beberapa printer menggunakan memori di RAM yaitu teknik spooling, namun proses pemindahan dari memori ke port printer tetap harus melalui proses antrian, dan antriannya tidak bisa disela (non pre-emptive) oleh data printer sebelumnya.

### **f. Menentukan beban kerja operasional**

Setelah melakukan analisa kinerja sistem, biasanya suatu sistem akan bekerja dalam tiga kinerja yaitu kinerja normal, kinerja terbebani dan kinerja tidak terbebani. Kinerja ini berdasarkan beban atau input yang diberikannya. Kinerja normal adalah kondisi sistem yang dibebani oleh beban normal, yaitu rentang input yang masih membuat kinerja sistem dapat dikontrol dan terprediksi. Kinerja sistem tidak normal terjadi saat terdapat beban yang memberikan kinerja sistem tidak terprediksi. Batas dari kinerja normal adalah saat kinerjanya turun mendekati kinerja optimum. Kinerja terbebani adalah pada saat sistem diberikan beban yang besar sehingga sistem selalu berada di dekat kinerja optimum. Sedangkan kinerja tidak terbebani adalah kondisi kinerja maksimum karena beban yang diberikan sangat kecil.

Beban kerja operasional adalah beban yang memberikan sistem tetap berada pada kinerja normal. Batas beban kerja operasional pada beberapa sistem ditetapkan berada setelah beban minimum dan sebelum beban optimum agar sistem tidak mudah rusak dan usianya lebih panjang



Gambar 1.1. Beban kerja dan kinerja

#### g. Merencanakan manajemen kapasitas

Untuk alasan agar sistem tidak sering mati atau kinerja sistem tetap terkontrol maka perlu penjadwalan beban disesuaikan dengan kehandalan sistem. Kehandalan sistem adalah kemampuan sistem untuk tetap tersedia atau tidak mati. Kemampuan sistem bertahan pada kondisi optimum tanpa sistem mati merupakan batas ekstrim dari manajemen kapasitas. Manajemen kapasitas juga mengendalikan agar sistem tidak selalu berada pada kondisi beban kerja minimum sehingga keluaran yang dihasilkan sistem selalu kecil.

Sistem Back-up sering ditambahkan untuk mempertahankan ketersediaan sistem.

#### h. Memprediksi kinerja sistem

Prediksi atau perkiraan kinerja sistem adalah proses penentuan kinerja sistem tanpa melakukan pengujian beban. Misal untuk mengetahui beban maksimum, tidak perlu sistem diuji dengan beberapa input yang diberikan menaik sampai kinerja tidak berubah, namun dengan metoda matematik seperti ekstrapolasi, least square kondisi maksimum dapat diprediksi.

Prediksi kinerja dilakukan untuk mengetahui beberapa kondisi yang ekstristik misal saat perubahan beban yang drastis, kondisi terbebani yang terlalu lama, penentuan kehandalan sistem. Sebagai contoh untuk mengetahui kinerja sistem server dalam melayani transaksi online dalam sistem *Online Transaction Process* (OLTP) dapat dilakukan dengan menguji dan memprediksi. Pengujian dilakukan dengan memberikan beban menaik, mulai dari beban minimum sampai beban optimum, tetapi tidak bisa sampai beban maksimum. Karena pada saat beban maksimum tercapai saat server tidak bisa menghasilkan keluaran lagi atau malah sistem mati. Pengujian ini berbahaya bisa merusak server. Cara terbaik adalah dengan memprediksi, yaitu dengan menguji sistem pada beban diatas sedikit dari beban normal, kemudian dengan menggunakan metoda matematik kondisi maksimal dan optimal dapat diprediksi.

## 1.6. Penentuan Metriks

Metriks atau variabel kinerja biasanya berupa variabel output yang dipengaruhi oleh variabel input (beban). Satu variabel input bisa menghasilkan beberapa metriks tergantung dari kinerja yang akan diamati. Misalnya dari jumlah data yang dikirim oleh sistem pengiriman komunikasi data maka dapat dipilih metriksnya adalah jumlah proses yang selesai (throughput), jumlah paket data yang tidak perlu dikirim ulang (resend), waktu seluruh data selesai diterima oleh server. Metriks pada dasarnya bisa berupa jumlah atau hasil perhitungan dari metriks lain. Berikut ini metriks untuk beberapa sistem

Tabel 1.1. Contoh Metriks

Sifat Metriks	Metriks
Jumlah	Throughput, Respon Waktu, Availabilitas,
Waktu	MTBF, MTTR, settling time,
Konsistensi	Fidelity, Linieritas, Reliabilitas
Acuan	Akurasi, presisi, error, bias, fault tolerance
Perhitungan metriks	bit error rate, kecepatan unduh
Beban	Hysterisis, Bandwidth

## 1.7. Tahapan Analisa Kinerja Sistem

Perlu suatu perencanaan dan tahapan yang cermat untuk melakukan proses analisa kinerja sistem. Pada saat suatu sistem yang sedang bekerja dilakukan pengukuran biasanya sistem sendiri akan terganggu oleh proses pengukuran tersebut sehingga perlu dibedakan pengukuran saat sistem bekerja (on going) dan pengukuran dengan sistem sedang tidak bekerja (statis). Sebagai contoh saat mengukur kinerja komputer yang sedang sibuk berbeda dengan saat mengukur kinerja komputer yang sedang tidak sibuk. Pengukuran delay dan throughput dari suatu sistem jaringan sebenarnya menambahkan suatu prosedur yang menyimpan besaran delay dan throughput tersebut ke media penyimpan dan setiap kali penyimpanan sebenarnya sudah membebani waktu eksekusi dari sistem yang diukur.

Namun secara umum tahapan analisa kinerja sistem dapat diurutkan dengan tahapan penentuan tujuan (goal), penetapan metriks (variabel ukur), Proses pengukuran dengan memberikan beban uji, pengukuran metriks akibat beban uji, proses penentuan kinerja (leveling) dan proses perbandingan kinerja (benchmarking). Pada gambar 1.2. diperlihatkan diagram tahapan analisa kinerja sistem.



Gambar 1.2. Tahapan Analisa Kinerja Sistem

Tahapan ini membentuk suatu siklus analisa kinerja sistem yang dilakukan untuk mengukur kinerja sistem di setiap periode yang ditetapkan. Jadi ditanya sampai kapan analisa kinerja sistem dilakukan maka jawabannya adalah selama sistem masih bekerja atau selama umur (*life time*) dari sistem.

## Bab 2 Jenis-jenis Metriks

Metriks atau yang disebut dengan variabel kinerja merupakan variabel atau besaran yang dipilih dan diukur sebagai representasi dari kinerja sistem. Tapi tidak semua kinerja sistem perlu diukur, kinerja sistem yang bersifat kewajaran tidak perlu diukur. Sebagai contoh komputer yang memiliki single cache memory dengan virtual harddisk yang kecil maka saat beberapa aplikasi dibuka sekaligus maka kecepatannya menjadi lambat, prosesoranya menjadi sibuk dan komputer terdiam. Ini suatu kewajaran yang tidak perlu diukur, kinerja yang seperti ini disebut kinerja bawaan (*inherited performance*). Komputer menjadi panas saat dipakai, inipun bukan suatu kinerja yang perlu diukur.

Ciri suatu variabel dapat dipilih sebagai metriks yaitu variabel yang merepresentasikan kinerja adalah :

- a. Saat beban sistem berubah maka metriks yang dipilih juga berubah
- b. Saat beban naik atau turun ada suatu pola kinerja yang bisa dimonitoring
- c. Secara terpisah (*independence*) atau bersama-sama perubahan metriks merepresentasikan besaran kualitas.

Memilih komputer berdasarkan kinerja merupakan hal yang penting saat memilih komputer yaitu dengan membanding metrik dari dua atau lebih komputer. Sudut pandang dari pengguna adalah komputer mana yang kinerjanya terbaik, harganya termurah dan perbandingan antara harga dan kinerjanya sesuai. Sedangkan dari perspektif perancang komputer yang dihadapkan dengan permasalahan perancangan berupa kinerja terbaik mana yang berhadil ditingkatkan, harga termurah dan juga perbandingan harga dan kinerja yang terbaik. Baik pengguna maupun perancangan melakukan hal yang sama yaitu perbandingan acuan (*benchmark*) dan evaluasi metriks (pengukuran kinerja).

Setiap arsitektur komputer yang berbeda biasa dianalisis berdasarkan dari biayanya dan kinerja yang dihasilkan. Terkadang kita perlu memahami :

- a. apakah perlu mengganti perangkat keras atau sebenarnya hanya mengganti sistem operasinya untuk meningkatkan kinerja
- b. serta mengapa pada arsitektur yang berbeda kinerjanya juga berbeda saat menjalankan program yang sama;
- c. Bagaimana intruksi mesin mempengaruhi kinerja.

### 2.1. Metriks sebagai penentu Kinerja Sistem Komputer

Metriks yang sering digunakan untuk mengevaluasi kinerja komputer adalah waktu respon (*Response Time*) dan *Throughput*. Waktu respon atau waktu eksekusi adalah waktu dari awal suatu program mulai sampai selesai. Pada multi tasking mungkin saja suatu program dipanggil-panggil menjadi beberapa task dan eksekusi masing-masing task dipengaruhi oleh penjadwalannya. Sedangkan *throughput* adalah jumlah pekerjaan yang selesai dalam kurun waktu tertentu. Biasanya manajer data lebih peduli terhadap *throughput* daripada waktu respon. seorang manajer data ingin agar *throughput*nya meningkat atau dengan kata lain jumlah pekerjaan yang selesai lebih banyak untuk suatu kurun waktu yang sama. Sedangkan waktu respon diharapkan lebih berkurang atau semakin cepat selesai tasknya. Sebagai ilustrasi pada pabrik mobil, diperlukan 4 jam untuk membuat mobil, ini adalah waktu respon; dan 6 mobil per jam yang dihasilkan , ini adalah *throughput*.

#### 2.1.1. Waktu Eksekusi

Terdapat beberapa metriks yang terkait dengan waktu eksekusi dalam Teknik Komputer yaitu *elapsed Time* (*Response Time*) dan *CPU Time*. *Elapsed Time* digunakan untuk menghitung waktu respon hampir di setiap perangkat komputer seperti untuk menghitung pada Disk, akses memori, alat I/O. Metriks yang berguna namun tidak cocok untuk dijadikan perbandingan saat terjadi multitasking. Sedangkan *CPU Time* tidak menghitung waktu yang diselesaikan oleh perangkat I/



O, sesuai dengan namanya ini adalah waktu yang diperlukan oleh komputer untuk melaksanakan setiap instruksi. CPU Time dapat dibagi menjadi waktu sistem (System Time) untuk sistem operasi dan Waktu Pengguna (User Time) untuk program.

Untuk satu program yang dijalankan di komputer X, kinerja komputer X dapat dinyatakan sebagai :

$$Kinerja_x = 1 / Waktu Eksekusi_x$$

Komputer X lebih cepat n kali terhadap Y jika: Speedup = n

$$Speedup = Kinerja_x / Kinerja_y = Waktu Eksekusi_y / Waktu Eksekusi_x$$

Mesin komputer dikatakan memiliki kinerja yang lebih baik jika total waktu eksekusi untuk programnya lebih pendek dibandingkan komputer lain.

Sebagai contoh:

Sebuah program yang sama dikerjakan oleh dua mesin komputer yang berbeda, komputer A dan komputer B. Waktu eksekusi untuk komputer A adalah 1 detik, sedangkan waktu eksekusi untuk komputer B adalah 10 detik. Maka

$$\frac{Kinerja_A}{Kinerja_B} = \frac{Waktu Eksekusi_B}{Waktu Eksekusi_A} = \frac{10}{1} = 10$$

Jadi kinerja komputer A lebih cepat 10 kali dari komputer B

### 2.1.2. Analisis CPU Time

CPU Time tergantung pada program yang dieksekusi, karena program merupakan kumpulan instruksi, maka CPU Time tergantung pada jumlah dan jenis instruksi-instruksi yang berada di dalam program tersebut. Pewaktuan komputer menggunakan clock rate dengan satuan Hz. Clock Rate ini merupakan waktu dari periode siklus yang disebut dengan clock cycle time (detik). Beberapa persamaan yang terkait dengan perhitungan CPU Time dapat dilihat pada tabel 2.1.

Tabel persamaan terkait perhitungan CPU Time

No.	Persamaan
1.	$clock\ cycle\ time = \frac{1}{clock\ rate}$
2.	$CPU\ Time = \frac{CPU\ clock\ cycle}{CPU\ clock\ rate} = CPU\ clock\ cycle \times clock\ cycle\ time$
3.	$CPU\ Time = Jumlah\ Instruksi \times CPI \times clock\ cycle\ time$
4.	$CPI = \frac{Clock\ cycle\ program}{Instruction\ count}$
5.	$CPI = \sum_{i=1}^n CPI_i \times F_i = CPI_1 \times F_1 + CPI_2 \times F_2 + \dots + CPI_n \times F_n$

Biasanya dalam teknik komputer penggunaan eksekusi time dalam detik jarang digunakan untuk perhitungan waktu CPU tetapi lebih seringnya menggunakan siklus (cycle).

$$CPU\ Time = \frac{detik}{program} = \frac{siklus}{program} \times \frac{detik}{siklus}$$

Clock time : Detik per siklus adalah waktu diantara Clock Tick

Misal untuk komputer dengan clock rate 4 GHz maka clock timenya adalah  $1/4 \cdot 10^9$  detik atau 250 picodetik =  $(1/4 \cdot 10^9) \times 10^{12}$

### 2.1.3. Bagaimana meningkatkan kinerja komputer

Misal suatu program diselesaikan dalam 10 detik pada komputer A yang clock ratenya 4GHz. Agar dapat diselesaikan 6 detik pada komputer B, namun komputer B memiliki teknologi yang clock cyclenya 1,2 kali clock Cyle A. Maka clock rate B dapat ditentukan sebagai berikut:

$$CPU \ Time_A = \frac{CPU \ clock \ Cycle_A}{CPU \ clock \ rate_A}$$

$$10 \ detik = \frac{CPU \ clock \ cycle_A}{4 \times 10^9 \frac{cycle}{detik}}$$

$$\therefore CPU \ clock \ cycle_A = 40 \times 10^9 \ siklus$$

$$CPU \ Time_B = \frac{1,2 \times CPU \ clock \ cycle_A}{clock \ rate_B}$$

$$\therefore Clock \ rate_B = 8GHz$$

### 2.1.4. Mengukur waktu menggunakan Clock Cycle

Untuk menghitung clock cycle dipergunakan rata-rata clock cycle per intruksi yang disebut CPI (Clock cycle per Intruction) dengan persamaan sebagai berikut:

$$Clock \ cycle \ untuk \ satu \ program = Instruction \ count \ x \ CPI$$

Instruction count adalah jumlah instruksi di dalam program.

Sehinga waktu eksekusi CPU untuk suatu program = Instruction count x CPI x clock cycle time

$$CPU \ Time = \frac{Instruksi}{program} \times \frac{siklus}{instruksi} \times \frac{detik}{siklus}$$

$$CPU \ Time = \frac{detik}{program}$$

CPI bercerita mengenai arsitektur instruksi, implementasi dari arsitektur komputer dan program yang diukur. Clock cycle besarnya sudah terdapat pada spesifikasi komputer, Instruction cout diukur menggunakan simulator (software) atau menggunakan hardware counter pada register

Contoh:

Sebuah program dijalankan pada suatu komputer dengan parameter berikut:

Jumlah instruksi: 10.000.000 instruksi

Rata-rata CPI: 2,5 siklus/instruksi

CPU clock rate: 200 MHz

a. Berapakah waktu eksekusi untuk program ini:

$$CPU \ time = jumlah \ instruksi \ x \ CPI \ x \ Clock \ cycle$$

$$\begin{aligned}
&= 10.000.000 \times 2,5 \times 1 / \text{clock rate} \\
&= 10.000.000 \times 2,5 \times 5 \times 10^{-9} \\
&= 0,125 \text{ detik}
\end{aligned}$$

b. Menggunakan program yang sama pada compiler yang lain diperoleh

instruction count: 9.500.000,

CPI baru: 3 siklus/instruksi

CPU clock rate lebih cepat : 300 MHz

Berapa speedup terhadap compiler baru ini:

$$\begin{aligned}
\text{Speedup} &= (10.000.000 \times 2,5 \times 5 \times 10^{-9}) / (9.500.000 \times 3 \times 3,33 \times 10^{-9}) \\
&= 0,125 / 0,095 = 1,32
\end{aligned}$$

atau 32% lebih cepat dengan compiler baru.

### 2.1.5. Waktu yang diperlukan instruksi

Tergantung pada jenis instruksinya, waktu yang diperlukan untuk eksekusi sebuah intruksi bisa berbeda, bahkan pada mesin komputer yang berbeda bisa didapati waktu eksekusi yang berbeda. Sehingga tidak bisa diambil kesimpulan satu siklus memiliki jumlah instruksi yang sama.

- Instruksi perkalian membutuhkan waktu eksekusi lebih lama dari instruksi penjumlahan.
- Operasi floating point lebih lama dari operasi bilangan integer
- Akses pada memori lebih lama dari akses pada register
- Mengganti cycle time akan mengubah jumlah siklus yang diperlukan oleh instruksi. Sebagai contoh program lama yang di-compile oleh Turbo Pascal versi DOS yang diperuntukan untuk komputer dengan clock rate 800 MHz akan terlalu cepat jika dieksekusi pada komputer di era sekarang yang kecepatannya di atas 1,6 GHz

Contoh :

Terdapat dua implementasi ISA (*Instruction Set Architecture*) yang sama.

Mesin A memiliki clock cycle time 250 ps dan CPI 2,0

Mesin B memiliki clock cycle time 500 ps dan CPI 1,2

Mesin komputer mana yang paling cepat untuk program ini? Dan seberapa cepat?

Jawab:

$$\text{CPU clock cycle}_A = \text{Instruction} \times 2,0$$

$$\text{CPU clock cycle}_B = \text{Instruction} \times 1,2$$

$$\text{CPU time}_A = \text{CPU clock cycle}_A \times \text{clock cycle time}_A = \text{Instruction} \times 2,0 \times 250\text{ps} = \text{Instruction} \times 500\text{ps}$$

$$\text{CPU time}_B = \text{CPU clock cycle}_B \times \text{clock cycle time}_B = \text{Instruction} \times 1,2 \times 500\text{ps} = \text{Instruction} \times 600 \text{ ps}$$

$$\frac{\text{CPU time}_B}{\text{CPU time}_A} = 1,2$$

### 2.1.6. Frekuensi Instruksi dan CPI

Teknik analisa sistem komputer yang lain adalah menggunakan Frekuensi atau fraksi instruksi, pada teknik ini CPI merupakan nilai rata-rata untuk instruksi tertentu. Misal jika di dalam program ada dua jenis instruksi yaitu perkalian dan penjumlahan maka akan ada dua CPI yaitu CPI untuk instruksi perkalian dan CPI untuk instruksi penjumlahan.

Didefinisikan :

$C_i$  = Jumlah Instruksi untuk jenis ke-i

$CPI_i$  = CPI untuk intruksi jenis ke-i

$F_i$  = Frekuensi atau fraksi instruksi jenis ke-i =  $C_i / \text{Total instruksi} = C_i / I$

Maka:

$$CPI = \sum_{i=1}^n CPI_i \times F_i = CPI_1 \times F_1 + CPI_2 \times F_2 + \dots + CPI_n \times F_n$$

Contoh :

Seorang desainer Compiler mencoba untuk memutuskan antara dua urutan kode (code sequence) untuk suatu mesin. Berdasarkan dari implementasi perangkat keras terdapat tiga kelas instruksi: kelas A, kelas B dan kelas C dan masing-masing memerlukan satu, dua dan tiga siklus.

• Urutan kode pertama memiliki 5 instruksi: 2 dari A, 1 dari B dan 2 dari C

• Urutan kode kedua memiliki 6 instruksi : 4 dari A, 1 dari B dan 1 dari C

a. urutan kode mana yang lebih cepat? dan seberapa cepat?

b. Berapa CPI untuk masing-masing urutan kode

Jawab:

a.

$$\text{CPU clock cycle}_1 = \sum_{i=1}^n CPI_i \times C_i = (2 \times 1) + (1 \times 2) + (2 \times 3) = 10 \text{ siklus}$$

$$\text{CPU clock cycle}_2 = \sum_{i=1}^n CPI_i \times C_i = (4 \times 1) + (1 \times 2) + (1 \times 3) = 9 \text{ siklus}$$

$$\text{Speedup} = 10/9 = 1,11$$

b.

$$CPI_1 = 10/5 = 2$$

$$CPI_2 = 9/6 = 1,5$$

Contoh ini memperlihatkan betapa berbahayanya menganalisis komputer hanya dari satu metrik. Saat membandingkan dua komputer, harus ditinjau dari 3 metrik yaitu clock rate, jumlah dari instruksi dan CPI.

Contoh 2.

Tinjau penggunaan MIPS ISA dengan 500 MHz clock dan

• Setiap instruksi ALU memerlukan 3 clock cycle

• Setiap instruksi branch/jump memerlukan 2 clock cycle

• setiap intruksi sw memerlukan 4 clock cycle

• setiap instruksi lw memerlukan 5 clock cycle

Juga tinjau sebuah program selama eksekusinya melakukan:

• x = 200 juta instruksi ALU

- $y = 55$  juta instruksi branch/jump instruksi
- $z = 25$  juta instruksi sw (store word)
- $w = 20$  juta instruksi lw (load word)

Carilah CPU Time

a. Pendekatan Pertama:

$$\text{Clock Cycle untuk program} = (x.3 + y.2 + z.4 + w.5) = 910.10^6 \text{ clock cycle}$$

$$\text{CPU Time} = \text{Clock cycle untuk program} / \text{Clock rate} = 910.10^6 / 500.10^6 = 1,82 \text{ detik}$$

b. Pendekatan kedua:

$$\text{CPI} = \text{Clock cycle untuk program} / \text{Instruction count}$$

$$\text{CPI} = (x.3 + y.2 + z.4 + w.5) / (x + y + z + w) = 3,03 \text{ clock cycle/instruction}$$

$$\text{CPU Time} = \text{Instruction count} \times \text{CPI} / \text{Clock rate} = (x + y + z + w) \times 3,03 / 500.10^6$$

$$\text{CPU Time} = (200 + 55 + 25 + 20).10^6 \times 3,03 / 500.10^6 = 1,82 \text{ detik}$$

Contoh 3:

Tinjau penggunaan MIPS ISA dengan 1 GHz clock dan

- Setiap instruksi ALU memerlukan 4 clock cycle
- Setiap instruksi branch/jump memerlukan 3 clock cycle
- setiap intruksi sw (store word) memerlukan 5 clock cycle
- setiap instruksi lw (load word) memerlukan 6 clock cycle

Untuk program yang sama seperti contoh 2, carilah CPI dan CPU Time

Jawab:

$$\text{CPI} = (x.4 + y.3 + z.5 + w.6) / (x + y + z + w) = 4,03 \text{ clock cycle/instruction}$$

$$\text{CPU Time} = \text{Instnction count} \times \text{CPI} / \text{Clock rate} = (x + y + z + w) \times 4,03 / 1000.10^6$$

$$\text{CPU Time} = 300 . 10^6 \times 4,03 / 100.10^6 = 1,21 \text{ detik}$$

Contoh 4: Perhitungan CPI menggunakan frekuensi instruksi

Tabel di bawah ini memperlihatkan frekuensi dari beberapa instruksi yang dieksekusi pada suatu progam, serta jumlah siklus dari masing-masing instruksi

Tentukan CPI nya

Jenis Instruksi	Frekuensi	Siklus
Instruksi ALU	50%	4
Instruksi Load	30%	5
Instruksi Store	5%	4
Instruksi Branch	15%	2

Maka

$$\text{CPI} = 0,5 \times 4 + 0,3 \times 5 + 0,05 \times 4 + 0,15 \times 2 = 4 \text{ cycle/instruction}$$

Contoh 5:

Dari tabel dibawah ini tentukan

- seberapa cepat komputer jika penambahan data cachernya mengurangi waktu pembebanan rata-rata (average load time) menjadi 2 siklus
- bagaimana jika dibandingkan dengan prediksi instruksi branch yang memangkas waktu siklus instruksi branch setengahnya
- Bagaimana jika dua instruksi ALU dapat dikerjakan dengan sekali eksekusi
- Seberapa cepat CPU Time sekarang untuk soal a sampai c

Jenis Instruksi	Frekuensi	Siklus	Frekuensi x Siklus	Kasus a (Siklus load =2)	Kasus b (Siklus Branch=1)	Kasus c (Siklus ALU = 0,5)
ALU	50%	1	0,5	0,5	0,5	<b>0,25</b>
Load	20%	5	1	0,4	1	1
Store	10%	3	0,3	0,3	0,3	0,3
Branch	20%	2	0,4	0,4	<b>0,2</b>	0,4
$CPI = \sum frekuensi_i \times siklus_i$			2,2	1,6	2,0	1,95

Jenis Instruksi	Frekuensi	Siklus	Frekuensi x Siklus
ALU	50%	1	0,5
Load	20%	5	1
Store	10%	3	0,3
Branch	20%	2	0,4
$CPI = \sum frekuensi_i \times siklus_i$			2,2

Jawab:

Untuk nomor a - c

d.

Kasus	$\frac{CPU\ Time}{CPU\ Time\ baru} = \frac{CPI}{CPI\ baru}$
a	2,2 / 1,6 = 1,375
b	2,2 / 2,0 = 1,1
b	2,2 / 1,95 = 1,128

Berarti untuk kasus a lebih cepat 37,5%, kasus b lebih cepat 10% dan untuk kasus c lebih cepat 12,8%

## 2.2. MIPS (*Millions Instruction per Second*) Rating

Sebagai catatan jika membicarakan MIPS berarti pada Mikroprosesor tidak terjadi kondisi pipeline terkunci (interlocking pipeline stages). MIPS merupakan alternatif dari metrik yang mengacu pada waktu. Untuk suatu program, MIPS rating atau laju MIPS berarti berapa juta instruksi yang dieksekusi per detik.

Jlka :

- waktu yang diperlukan untuk satu instruksi =  $CPI * CT$
- berarti instruksi yang dieksekusi per detik =  $1 / (CPI * CT)$

maka

$$MIPS = 1 / (CPI * CT * 10^6) = CR / (CPI * 10^6) = IC / (Execution Time * 10^6)$$

Terdapat tiga masalah saat menggunakan MIPS sebagai metrik dalam analisa kinerja sistem komputer

1. MIPS tidak mewakili kapabilitas instruksi, tidak dapat membanding dua komputer dengan kumpulan instruksi yang berbeda menggunakan MIPS karena instruction count-nya akan berbeda
2. MIPS bervariasi nilainya untuk beberapa program pada komputer yang sama, komputer tidak memiliki nilai tunggal MIPS untuk semua program
3. MIPS bisa bervariasi nilainya dan berbanding terbalik dengan kinerja. Nilai MIPS yang besar tidak berarti kinerjanya lebih besar atau waktu eksekusinya lebih baik. Semua tergantung compiler yang digunakan

Namun eksekusi yang cepat berarti MIPS ratingnya besar.

Contoh :

Dua compiler yang berbeda diuji pada mesin 4 GHz dengan tiga kelas instruksi yang berbeda: Kelas A, Kelas B dan Kelas C yang masing-masing memerlukan 1, 2 dan 3 siklus.

Kedua compiler digunakan untuk menghasilkan kode untuk sebuah bagian besar dari perangkat lunak.

- Kode dari Compiler pertama memerlukan 5 juta instruksi kelas A, 1 juta instruksi kelas B dan 1 juta instruksi kelas C
- Kode dari Compiler kedua memerlukan 10 juta instruksi kelas A, 1 juta instruksi kelas B dan 1 juta instruksi kelas C

a Berdasarkan waktu eksekusi urutan mana yang tercepat?

b. Berdasarkan MIPS urutan (sequence) mana yang tercepat?

Jawab:

$$a. \text{ Execution Time} = \frac{\text{CPU clock cycle}}{\text{Clock rate}}$$

$$\text{CPU clock cycle}_1 = \sum CPI_i \times C_i = [(5 \times 1) + (1 \times 2) + (1 \times 3)] \times 10^6 = 10 \times 10^6$$

$$\text{CPU clock cycle}_2 = \sum CPI_i \times C_i = [(10 \times 1) + (1 \times 2) + (1 \times 3)] \times 10^6 = 15 \times 10^6$$

$$\text{Execution Time}_1 = 10 \cdot 10^6 / 4 \cdot 10^9 = 2,5 \text{ ms}$$

$$\text{Execution Time}_2 = 15 \cdot 10^6 / 4 \cdot 10^9 = 3,75 \text{ ms}$$

$$b. \text{ MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

$$\text{MIPS}_1 = \frac{7 \times 10^6}{2,5 \times 10^{-3} \times 10^6} = 2800$$

$$\text{MIPS}_2 = \frac{12 \times 10^6}{3,7 \times 10^{-3} \times 10^6} = 3200$$

Berarti kode dari compiler 2 memiliki MIPS rating terbesar namun compiler 1 lebih cepat

### 2.2.1 Kinerja Lup MIPS

Untuk lup berikut:

```
for (i=0; i<1000; i=i+1){
    x[i] = x[i] + s; }
```

maka dalam kode assemblynya:

lw \$3, 8(\$1)	# load s in \$3
addi \$6, \$2, 4000	# \$6 = address of last element + 4
loop: lw \$4, 0(\$2)	# load x[i] in \$4
add \$5, \$4, \$3	# \$5 has x[i] + s
sw \$5, 0(\$2)	# store computed x[i]
addi \$2, \$2, 4	# increment \$2 to point to next x[ ] element
bne \$6, \$2, loop	# last loop iteration reached?

kode MIPS dieksekusi pada CPU dengan spesifikasi 500 MHz (clock cycle rate 2ns) dengan jenis instruksi CPI sebagai berikut:

Jenis Instruksi	CPI
ALU	4
Load	5
Store	7
Branch	3

untuk kode MIPS yang dijalankan pada CPU ini tentukan

- Fraksi instruksi terhadap total instruksi untuk masing-masing jenis instruksi
- Total jumlah siklus CPU (CPU cycle)
- CPI rata-rata
- fraksi dari waktu eksekusi untuk masing-masing instruksi



- e. Waktu Eksekusi.
- f. MIPS Rating

Jawab:

a. dari program bahasa mesin dapat dihitung jumlah instruksi

Jenis Instruksi	Jumlah Instruksi	Fraksi instruksi
ALU	1999	1999/4997=40%
Load	1000	1000/4997=20%
Store	999	999/4997=20%
Branch	999	999/4997=20%
Total	4997	

b. Total CPU Clock cycle=  $[(0,4*4) + (0,2*5) + (0,2*7) + (0,2*3)]x10^6 = 4,6 \times 10^6$  siklus

c. CPI rata-rata =  $\frac{CPU \ Clock \ cycle}{instruction \ count} = \frac{4,6x10^6}{4997} = 921 \ \text{siklus/instruksi}$

d. Fraksi dari waktu eksekusi

Jenis Instruksi	Fraksi instruksi	CPI	Fraksi x CPI	Fraksi waktu
ALU	40%	4	1,6	1,6/4,6=35%
Load	20%	5	1	1/4,6=22%
Store	20%	7	1,4	1,4/4,6=30%
Branch	20%	3	0,6	0,6/4,6=13%
Total			4,6	

e. Waktu Eksekusi =  $\frac{CPU \ Clock \ cycle}{CPU \ clock \ rate} = \frac{4,6 \times 10^6}{500x10^6} = 0,0092 \ \text{detik}$

f. MIPS rating

=  $\frac{Instruction \ Count}{Execution \ time \ x \ 10^6} = \frac{4997}{0,0092x10^6} = 0,54 \ \text{juta \ instruksi/detik}$

### 2.3. Metriks pada Analisa Kinerja Sistem Jaringan yang memiliki multitasking

Sebagai contoh untuk menganalisis kualitas layanan suatu jaringan (Quality of Services, QoS) dipilih metriks sebagai berikut

- a. Waktu tanggap (*response time*)
- b. Throughput
- c. Availabilitas
- d. Reliabilitas
- e. Security
- f. Scalabilitas
- g. Extensibilitas

Penjelasan dari masing-masing metriks dan penggunaannya adalah sebagai berikut:

#### Waktu Tanggap

Waktu suatu task menanggapi perintah awal penugasan. Pada jaringan misal ingin di amati waktu tanggap sebelum kondisi jaringan sibuk maka dipilih tiga proses utama (thread):

1. Waktu browser (browser time)
2. Waktu Jaringan (Network Time)
3. Waktu server e-commerce

Maka masing-masing task yang diamati waktu tanggapnya sesuai dengan thread ditunjukkan pada tabel 2.2 berikut

Tabel 2.2. Pemilahan Waktu Respon

Waktu Browser		Waktu jaringan			Waktu Server e-commerce		
Pemrosesan oleh Prosesor	Sumber daya I/O	Waktu Browser ke ISP	Waktu internet	Waktu ISP ke server	Pemrosesan oleh Prosesor	Sumber daya I/O	Jaringan

Ketiga proses utama dapat dijadikan metriks, jika ingin lebih detail masing-masing tugas (task) yang berada di proses utama bisa dijadikan metriks. Ada variabel yang tidak bisa dijadikan metriks untuk kasus kemacetan (congestion) jaringan yaitu waktu layanan (service time) karena waktu layanan oleh server tidak ditentukan oleh jumlah beban. Namun semakin banyak browser yang dibuka maka kemacetan jaringan akan semakin cepat terjadi karena itu dapat dikatakan kemacetan jaringan bergantung pada beban (*load dependent*)

#### Throughput

Throughput dalam sistem komputer seringkali tidak diterjemahkan ke dalam bahasa Indonesia, ada yang menerjemahkan menjadi jumlah keluaran, jumlah produksi atau jumlah capaian. Throughput sendiri merupakan nilai yang merepresentasikan jumlah tugas yang selesai diproses untuk kurun waktu yang dipilih. Karena satuannya per waktu maka throughput secara ilmu fisika merupakan suatu laju (rate). Untuk kasus ini throughput yang dipilih adalah :

- a. Jumlah sumberdaya yang digunakan (I/O per sec)
- b. Jumlah halaman browser yang di download (Page per sec)
- c. Jumlah permintaan http (http request per sec)
- d. Jumlah tugas yang selesai (task per sec)
- e. Jumlah transaksi yang selesai (transaction per sec, tps)

### Contoh Soal 2.1.

Sebuah pengerjaan I/O pada harddisk pada saat sistem transaksi OLTP (*Online Transaction Process*) berlangsung rata-rata memerlukan waktu 10 ms. Untuk 1 menit proses hitung:

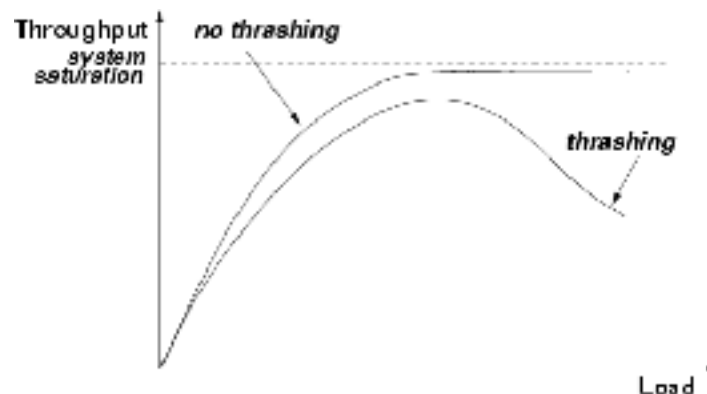
- Berapa throughput maksimum dari harddisk
- Jika permintaan I/O pada laju 80 permintaan per detik berapa throughput yang dihasilkan

Jawab:

Asumsi tidak ada proses pembuangan task (thrashing) pada manajemen jaringan

- Throughput maksimum adalah 1 menit / 10 ms = 60/0,1 = 600 throughput
- Throughput = 60/80 = 0,75 karena dibawah satu maka belum ada proses yang selesai atau throughputnya adalah nol

Hasil ini akan berbeda jika ada manajemen jaringan (Gambar 2.1)



Gambar 2.1 Throughput yang dihasilkan untuk peningkatan beban.

Dari gambar 2.1 ada yang menarik yaitu sistem yang sedang dianalisa memiliki batas saturasi. Batas ini menunjukkan untuk beban yang terus meningkat server ada throughput maksimum yang bisa dihasilkan. Sedangkan pada kurva dengan manajemen memori (thrashing), saat beban puncak maka sistem jaringan akan mengintruksikan prosesor untuk memilih thread yang mana yang perlu ditunda dulu, dan memberikan prioritas pada thread yang harus didahulukan. Manajemen jaringan ini membuat jumlah antrian thread menjadi lebih sedikit yang perlu diselesaikan dan juga menyebabkan jumlah throughput yang diselesaikan lebih sedikit. Pada kurva terlihat penurunan setelah beban puncak.

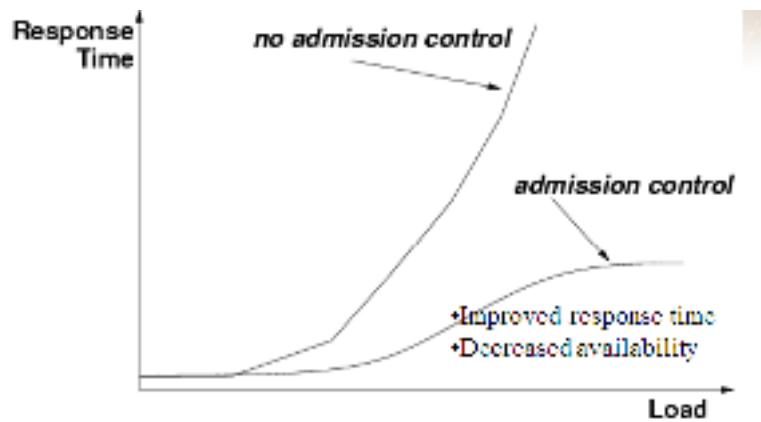
### Availabilitas

Availabilitas adalah metrik yang menyatakan kinerja sistem siap untuk digunakan. Nilai Availabilitas dinyatakan dengan 0% sampai 100%. Jika sistem siap sempurna maka availabilitasnya adalah 100%. Adanya interupsi pada layanan membuat beberapa saat sistem tidak siap bekerja. Interupsi yang melampaui batas maksimal dari availabilitas yang ditentukan bisa menyebabkan kehilangan suatu ongkos yang sangat besar bahkan jiwa manusia bisa hilang.

Sistem dengan availabilitas tinggi misalnya 99,99% selama 30 hari kerja berarti sistem memiliki kondisi tidak siap sebesar  $(1-0.9999) \times 30 \times 24 \times 60 = 4,32$  menit. Misalnya pernahkah saat membuka suatu browser ada tampilan error yang menyatakan browser sedang sibuk atau erros?

Dalam ilmu jaringan komputer tidak selamanya diperlukan server yang selalu siap, karena pada saat beban bertambah waktu antrian atau dalam hal ini waktu responnya menjadi lebih lama dan umur server menjadi berkurang karena waktu hidupnya dipercepat. Salah satu cara adalah dengan menerapkan algoritma kemacetan yang disebut *admission control*. Task dengan kebutuhan waktu layanan yang lebih cepat akan diprioritaskan sehingga jumlah *throughput* lebih banyak dan waktu tanggap dari suatu task menjadi lebih kecil.

Sebagai ilustrasi, algoritma First Come First Served (FCFS) pada sistem kemacetan jaringan akan membuat waktu tanggap atau antrian yang lama suatu Task harus menunggu Task lain yang memerlukan sumber daya yang lama. Sedangkan algoritma Short Remaining Time First Scheduling (SRTFS) akan memberikan jumlah Task yang selesai lebih banyak dan waktu tanggap atau antrian menjadi lebih kecil. Karena total waktu tanggap menjadi lebih kecil maka availabilitas secara total juga berkurang (Gambar 2.2.)

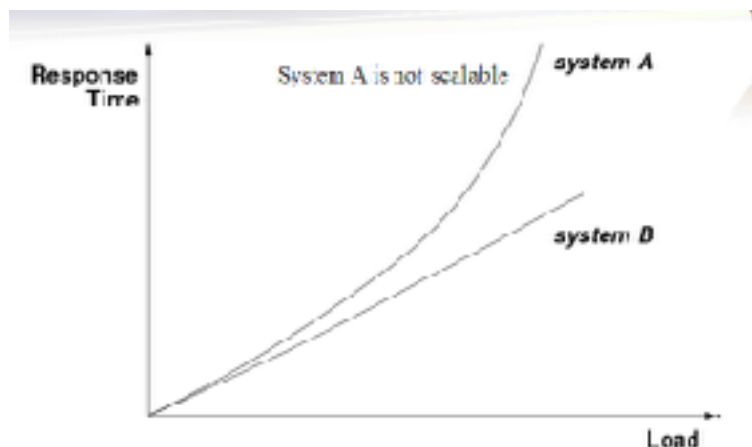


Gambar 2.2. Pengontrolan pembagian beban dengan menambah server (admission control)

Diskusikan : Bagaimana dengan algoritma yang membatasi waktu eksekusi pada nilai tertentu atau yang dikenal dengan algoritma Round Robin

**Scalabilitas**

Scalabilitas merupakan kemampuan sistem untuk mengubah kualitasnya sesuai dengan beban. Tujuannya untuk memperpanjang umur hidup dan mengefisiensikan ongkos operaional. Biasanya beberapa sistem mengurangi parameter normalnya untuk beban yang kecil dan akan mengubah parameternya seiring dengan penambahan beban. Misalkan pada saat beban jaringan masih normal maka bandwidth jaringannya juga normal dan tidak perlu ada penambahan server pembantu. Sedangkan saat jaringan tersebut bebannya bertambah maka secara bersamaan ada mekanisme yang mengubah bandwidth jaringan menjadi bertambah sesuai dengan penambahan beban tersebut. Jikalau perlu ada penambahan server untuk mempersingkat waktu tanggap. (Gambar 2.3)



Gambar 2.3. Pengaruh Proses Scalabilitas pada waktu tanggap

Pada gambar 2.3. terlihat bahwa dengan adanya scalabilitas pada jaringan, sistem B lebih terkontrol kenaikan waktu tanggapnya dan waktu tanggapnya menjadi linier terhadap beban.

## **Extensibilitas**

Ekstensibilitas adalah kemampuan sistem menambah kualitasnya dengan memperluas cakupan daerah kontrolnya terhadap perubahan beban.

Dalam ekstensibilitas terdapat kemampuan dari sistem untuk memantau kinerja dirinya sendiri dan secara otomatis mengubah parameter di dalam sistemnya untuk membuat kualitas kinerja tetap normal. Diperlukan kemampuan kecerdasan sistem untuk mengetahui kinerja sistem di luar dari kinerja operasional normal.

*Contoh scalabilitas dan Admission Control* merupakan salah satu ekstensibilitas di dalam sistem jaringan. Nama lain ekstensibilitas adalah *Autonomic Computing, self-managing system, self-recovery and mitigation system*