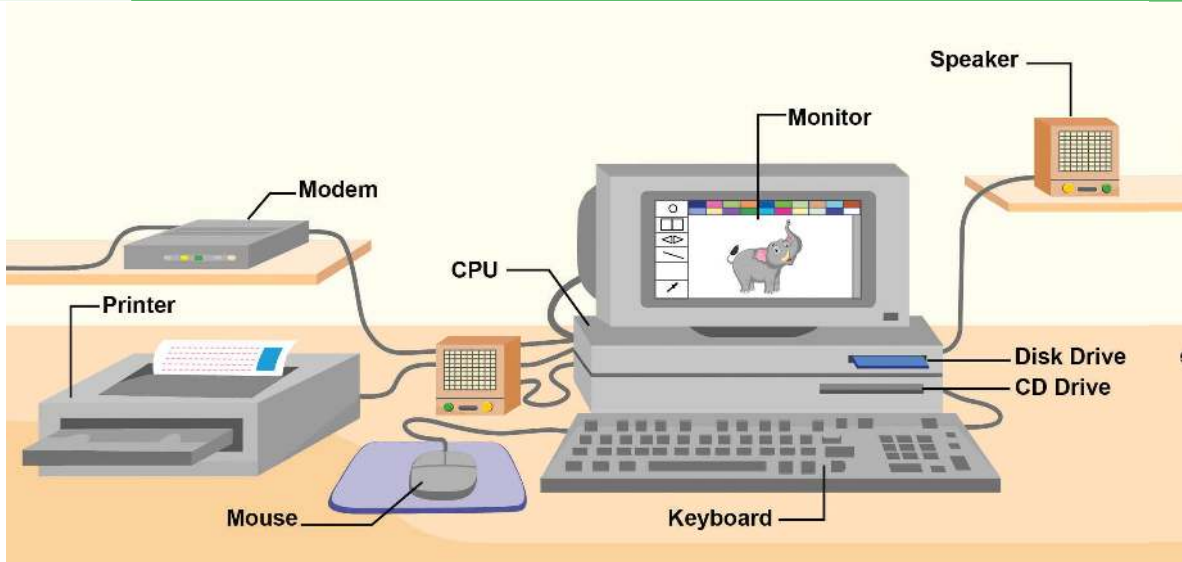


Sistem Komputer



Tujuan Pembelajaran

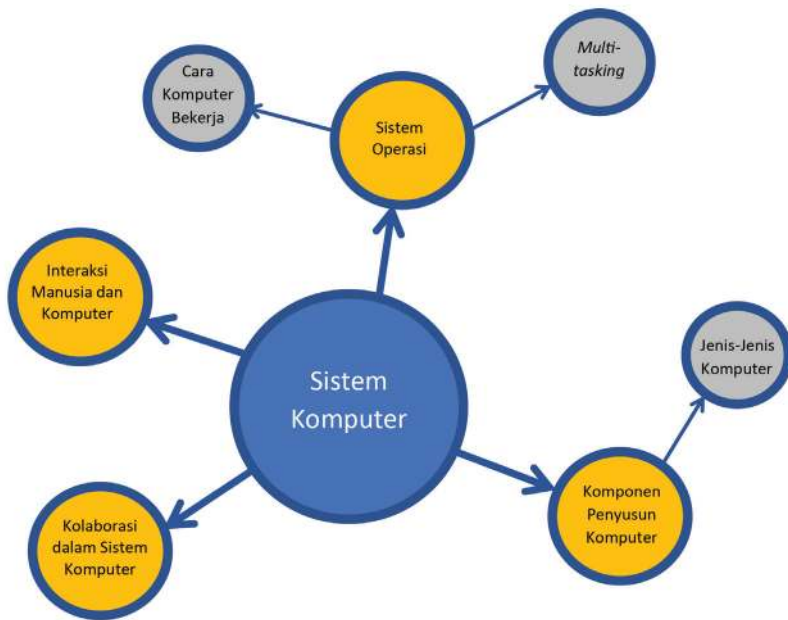
Setelah mempelajari bab ini, kalian mampu menjelaskan peran sistem operasi pada komputer, cara kerja komputer dalam memproses data, dan mekanisme internal yang terjadi pada interaksi antara perangkat keras, perangkat lunak, dan pengguna.



Pertanyaan Pemantik

Pada era digital saat ini, banyak kegiatan kita yang bergantung pada peralatan komputer. Tahukah kalian bahwa komputer bisa membantu kegiatan kita karena adanya sistem komputer yang dibentuk dari komponen-komponen? Bagaimana kerja sistem komputer?

Peta Konsep



Gambar 4.1 Peta Konsep Sistem Komputer
Sumber: Dokumen Kemendikbud, 2021

Apersepsi

Mungkin kalian pernah melihat bagaimana peranti elektronik bekerja, seperti mesin cuci, mesin pemotong rumput, dll. Komputer sebenarnya sama halnya dengan peranti lain, tetapi tampak lebih kompleks. Pekerjaan yang dilakukan oleh komputer dilakukan oleh banyak komponennya. Komputer juga memiliki arsitektur, yaitu tatanan komponen yang dirancang oleh pembuatnya. Pada materi ini, kalian akan belajar lebih dalam bagaimana sistem komputer bekerja.

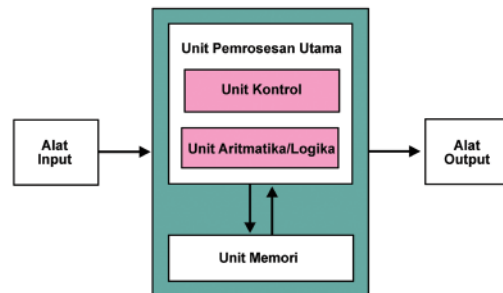
Kata Kunci

komponen komputer, CPU, ALU, perangkat lunak, perangkat keras, sistem operasi

Di jenjang SMP, kalian telah mengenal perangkat keras dan perangkat lunak. Perangkat tersebut dapat dijumpai sehari-hari di sekitar masyarakat yang sudah memanfaatkan TIK untuk membentuk sebuah sistem komputasi yang siap dimanfaatkan oleh pengguna. Selain itu, kalian juga telah memahami pentingnya sistem operasi sebagai perangkat lunak yang akan melayani perangkat keras dan perangkat lunak aplikasi yang dipakai oleh pengguna. Di kelas X, kalian diajak untuk menjenguk lebih dalam ke dalam “mesin” komputer. Gunanya untuk memahami komponen-komponen di dalam perangkat keras komputer secara lebih rinci, tetapi dalam bentuk abstraksi, dan bagaimana instruksi atau perintah yang ditulis sebagai program komputer dijalankan.

A. Komputer dan Komponen Penyusunnya

Secara umum, komputer adalah peralatan elektronik yang menerima masukan data, mengolah data, dan memberikan hasil keluaran dalam bentuk informasi, baik itu berupa gambar, teks, suara, maupun video. Secara sederhana, sebuah komputer menerima masukan dari peranti masukan, memproses masukan tersebut, dan menghasilkan output. Gambaran umum sebuah komputer tampak pada Gambar 4.2, di mana “data” (teks, suara, klik, sentuhan, atau lainnya) dari peranti masukan akan diproses oleh *Central Processing Unit* (CPU) untuk menghasilkan *output* berupa tampilan informasi, sinyal reaksi, dll. yang direpresentasikan melalui peranti keluaran.



Gambar 4.2 Diagram Blok Konseptual Komputer

Sumber: Dokumen Kemendikbud, 2021

1. Sistem Komputer

Sistem Komputer terdiri atas beberapa bagian berikut.

1. Perangkat Keras (*Hardware*)
2. Perangkat Lunak (*Software*)
3. Pengguna

Semua komponen tersebut saling mendukung sehingga komputer dapat beroperasi. Perangkat keras komputer membutuhkan perangkat lunak agar komputer bisa dihidupkan dan difungsikan. Jika hardware yang tidak disertai *software*, komputer hanyalah sebuah mesin yang tidak berguna. Hal ini

dikarenakan *software* tercipta untuk menulis fungsionalitas pada komputer tersebut sehingga terciptalah sebuah komputer yang memiliki fungsi untuk digunakan. *Hardware* saja ibarat tubuh manusia tanpa jiwa.

a. Perangkat Keras (*Hardware*)

Perangkat keras komputer (*hardware*) adalah komponen fisik pada komputer yang dapat disentuh, dilihat atau dipindahkan. Contoh perangkat keras ialah mouse, harddisk, processor, RAM, printer, scanner dll.

b. Perangkat Lunak (*Software*)

Perangkat lunak komputer (*software*) tidak terlihat secara fisik, tetapi berfungsi dan dapat dioperasikan oleh pengguna melalui antarmuka yang disediakan. Fungsinya ialah untuk menjembatani pengguna dengan perangkat keras. Perangkat lunak adalah kode-kode program yang dibuat menggunakan bahasa pemrograman. Kode-kode tersebut merupakan kumpulan perintah atau instruksi untuk menjalankan tugas tertentu sesuai dengan keinginan pengguna, atau untuk mengendalikan kerja perangkat keras. Jika sebuah sistem komputer diibaratkan manusia, perangkat keras adalah “otak” dan perangkat lunak adalah “pikiran”. Contoh perangkat lunak ialah sistem operasi, aplikasi (app), dll. Sistem operasi yang banyak dipakai saat ini ialah MS Windows, MacOS, dan Linux, sedangkan perangkat lunak lain yang menjadi sistem operasi ponsel pintar adalah Android.

c. Pengguna

Pengguna adalah orang yang menggunakan atau mengoperasikan komputer. Pemahaman lebih lanjut tentang pengguna, dapat ditemukan pada materi Dampak Sosial Informatika.

2. Jenis-Jenis Komputer

Berdasarkan ukurannya, komputer dibagi menjadi beberapa jenis antara lain seperti ditunjukkan pada penjelasan berikut.

Microcomputer (Komputer Mikro)

Komputer mikro merupakan komputer yang memiliki ukuran paling kecil dibandingkan dengan jenis komputer lainnya dan menggunakan microprocessor sebagai CPU atau unit pemrosesan utama. Contoh dari komputer mikro antara lain Ultrabook, permainan konsol, telepon pintar dan Tablet. Karena ukuran yang kecil dan harga yang lebih murah dibandingkan dengan jenis komputer lainnya, komputer mikro paling banyak digunakan dalam kehidupan sehari-hari.

Beberapa komputer bahkan dalam bentuk papan tunggal (*single board circuit*, SBC) yang berukuran kecil, misalnya yang populer ialah raspberry pi dan arduino.

Raspberry Pi, sering disingkat Raspi, ialah SBC yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris (<https://www.raspberrypi.org/>)



Gambar 4.3 Contoh Microcomputer

Arduino adalah *platform* elektronik open-source berdasarkan perangkat keras dan perangkat lunak yang mudah digunakan, ditujukan untuk membuat proyek interaktif. Papan arduino dapat membaca *input* dan menghasilkan sinyal output yang mengaktifasi motor, menyalakan LED, atau lainnya. Arduino dapat diprogram dengan mudah. Karena kemudahan dan harganya yang murah, arduino dapat ditemui mulai dari papan 8-bit sederhana hingga produk untuk aplikasi IoT, perangkat yang dapat dikenakan, pencetakan 3D, dan *embedded system*. (<https://www.arduino.cc/>)

Komputer Personal (PC, *Personal Computer*)

Komputer personal atau PC memiliki ukuran yang lebih besar dari komputer mikro dan memiliki kemampuan penyimpanan dan pengolahan data yang lebih besar dibandingkan dengan komputer mikro, dan dibuat untuk penggunaan personal. PC dapat berbentuk *desktop* PC (dirancang untuk ditaruh di meja), atau untuk dapat dijinjing dan dibawa-bawa (*laptop*).



Gambar 4.4 Contoh Personal Computer

Mini PC

Merupakan komputer “peralihan” dari komputer personal ke komputer mini yang dipakai di industri. Biasanya, dipakai untuk industri kecil atau personal untuk keperluan profesional atau industri kecil.



Gambar 4.5 Contoh Mini PC, Coffee Lake-U-based Bean Canyon Intel NUC8i5BEK2

Minicomputer

Berbeda dengan komputer personal, komputer mini berukuran lebih besar, dan mempunyai kapasitas memori maupun pemroses yang lebih besar. Komputer mini dipakai menunjang kebutuhan pengolahan informasi perusahaan skala menengah. Saat ini, komputer mini kurang populer dan makin sedikit digunakan karena perusahaan lebih praktis untuk menyewa komputer di *cloud* yang memudahkan pemeliharannya.



Gambar 4.6 Contoh Minicomputer PDP-8e Minicomputer System

Komputer Mainframe

Komputer *Mainframe* berukuran lebih besar dibandingkan dengan komputer dan biasanya digunakan oleh perusahaan-perusahaan besar sebagai *server* (peladen).



Gambar 4.7 Salah satu contoh Komputer Mainframe IBM z Systems z13.

Supercomputer

Dibandingkan dengan komputer lainnya, *supercomputer* memiliki ukuran yang paling besar dan memiliki kapasitas pengolahan data dan kinerja yang paling kuat. Super Komputer memiliki kemampuan untuk melakukan triliunan perintah atau instruksi per detik yang dapat dihitung dalam FLOPS (*Floating Point Operation per Second*). Sama seperti minicomputer dan *mainframe*, pengguna super komputer biasanya ialah perusahaan atau organisasi besar, misalnya NASA yang menggunakannya dalam meluncurkan dan mengendalikan pesawat dan roket.



Gambar 4.8 Sierra/ATS-2 Super Computer

Catatan: selain “komputer” yang disebutkan di atas, beberapa gawai (*gadget*), yaitu perangkat elektronik kecil yang berfungsi khusus, ada yang termasuk “komputer” karena terdiri atas perangkat keras, sistem operasi dan perangkat lunak. Contohnya telepon pintar (*smartphone*) dan tablet, yang makin populer dan menjadi perlengkapan sehari-hari di era digital ini. Selain fungsi utamanya untuk berkomunikasi, ponsel pintar bahkan sudah menjadi “asisten” pribadi.



Gambar 4.9 Abstraksi Interaksi Perangkat Keras, Perangkat Lunak, dan Pengguna

Sumber: Dokumen Kemendikbud, 2021

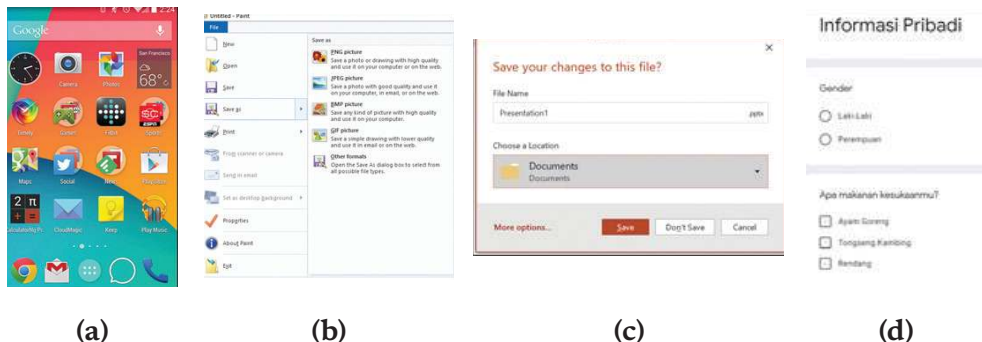
B. Interaksi Manusia dan Komputer

Sebuah sistem komputasi terdiri atas perangkat keras dan perangkat lunak yang saling berinteraksi. Interaksi antarmuka seringkali digunakan untuk menghubungkan perangkat satu dengan perangkat lainnya, mulai dari perangkat masukan dan keluaran yang dikendalikan oleh sistem operasi, dan kemudian dapat ditambahkan berbagai aplikasi yang dapat dipakai oleh pengguna.

Berbagai macam jenis interaksi dapat dilakukan oleh pengguna dengan sistem komputer, di antaranya seperti berikut.

1. Berbasis GUI (*Graphical User Interface*)

Merupakan antarmuka yang menggunakan menu grafis untuk memudahkan pengguna berinteraksi dengan komputer. GUI merupakan antarmuka pada sistem operasi komputer yang menggunakan menu grafis. Pengguna berinteraksi melalui ikon, menu, dialog dengan *button* dan *text box*, radio *button* (untuk satu pilihan), atau *checkbox* (untuk banyak Pilihan).

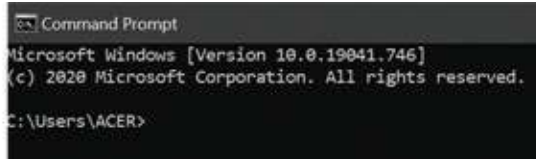


Gambar 4.10 Antarmuka Berbasis Grafis (a) Ikon pada ponsel, (b) Menu, (c) Dialog, textbox, dan button, (d) checkbox dan radio button

Sumber: Dokumen Kemendikbud, 2021

2. Antarmuka Berbasis Perintah (*Command Line Interface/CLI*)

Sistem operasi berbasis CLI merupakan tipe antarmuka melalui *text-terminal*. Pengguna menjalankan perintah dan program di sistem operasi tersebut dengan cara mengetikkan baris-baris tertentu.



Gambar 4.11 Contoh Antarmuka Berbasis Perintah (CLI)

Sumber: Dokumen Kemendikbud, 2021

3. Melalui Suara (*Audio*)

Antarmuka menggunakan suara memungkinkan pengguna mengucapkan sesuatu dan hasilnya akan direkam, dalam bentuk format audio. Antarmuka ini hanya dimungkinkan jika perangkat keras menyediakan perekam suara dan melalui aplikasi.

4. Melalui Gambar (*Video*)

Antarmuka menggunakan gambar hanya dimungkinkan jika perangkat keras menyediakan kamera. Kamera akan merekam gambar dan melalui aplikasi akan menyimpan gambar dalam format video.

5. Melalui Berbagai Piranti Masukan Lainnya

Selain melalui perangkat lunak, pengguna dapat berinteraksi langsung ke komputer melalui piranti masukan, seperti yang dijelaskan sebelumnya (*keyboard, joystick, mouse, touchpad, layar sentuh, keyboard virtual, dll*).

C. Kolaborasi dalam Sistem Komputer

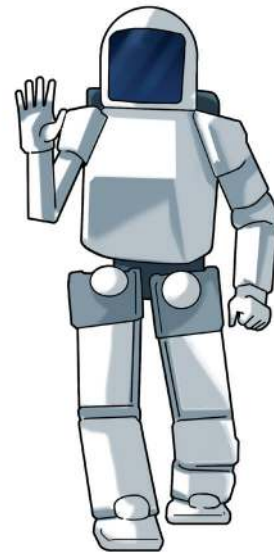
Dari semua uraian di atas, terlihat bahwa di dalam sebuah sistem komputer, perangkat keras berkolaborasi dengan perangkat lunak aplikasi atau piranti lain melalui sistem operasi. Sistem Operasi pun berkolaborasi dengan aplikasi untuk berinteraksi dengan pengguna. Kolaborasi itu menghasilkan sebuah sistem komputasi yang akan bermanfaat bagi pengguna. Komputer tunggal seperti seorang manusia, yang di dalam tubuhnya beroperasi sistem-sistem yang berfungsi sesuai peran dari setiap organ tubuh. Pikiran manusia akan menggerakkan anggota tubuh seperti tangan dan kaki karena adanya sistem syaraf. Indra penglihat, peraba, perasa akan memberikan signal kepada sistem syaraf untuk membuat manusia bereaksi, bergerak, dan melakukan tindakan. Inilah yang menjadi dasar manusia menciptakan robot-robot yang bertindak dan berperilaku sesuai program yang disimpan dalam robot tersebut.

Pada sistem-sistem otonom, sistem komputasi tidak hanya berkomunikasi dengan manusia, tetapi juga berinteraksi dengan alam sekitarnya. Perekam gambar, suara, dan signal lainnya akan dapat mengirimkan data ke sistem untuk diolah. Inilah dasar dari sistem IoT (*Internet of Things*) di mana di sekeliling manusia dipasang perangkat-perangkat yang mampu menangkap data dan mengirimkan ke komputer. Dalam sebuah komputer, komponen-komponennya saling berhubungan. Demikian juga antara satu komputer dan komputer lainnya akan berhubungan melalui jaringan, mulai dari jaringan lokal sampai jaringan global internet. Berikut penjelasan ringkas kolaborasi tersebut.

Bahwa dalam sistem komputer, terjadi interaksi dan kolaborasi. Ini dapat dijumpai dalam kehidupan sehari-hari, seperti software yang digunakan di sebuah toko, warnet, kantor, rumah sakit, dan sebagainya. Seperti ketika melakukan instalasi program aplikasi penggajian (*payroll*) di komputer atau laptop kantor yang datanya masih perlu diolah dengan pengolah lembar kerja. Contoh lainnya ialah ketika membeli *laptop* baru dan akan membuat laporan di dalamnya, perlu menginstal sistem operasi terlebih dulu, sebelum aplikasi pengolah kata. Hal ini menunjukkan bahwa kedua *software* di atas saling bekerja sama satu sama lainnya, sebuah *software* bergantung pada *software* lainnya.

Salah satu contoh dari interaksi *hardware* dengan sesama *hardware* ialah saat kita mencolokkan sebuah perangkat keras dengan perangkat keras lain. Misalnya, mencolokkan *flashdisk* atau *mouse* ke komputer. Saat kita mengambil *photo* menggunakan *webcam* yang ada pada komputer dan kemudian hasilnya akan dicetak menggunakan printer, apakah ini interaksi antarperangkat keras? Pada kasus ini, *webcam* akan mengirimkan gambar ke komputer, dan gambar yang akan dicetak ke *printer* dibuka oleh aplikasi lain. Interaksi *webcam* dan memori komputer dilakukan melalui sistem operasi yang dijalankan pada komputer. Interaksi antarperangkat keras ini membutuhkan perangkat lunak sebagai perantara.

Interaksi antarmuka *hardware* dan *software* terlihat sedikit lebih rumit karena *software* harus dirancang dan dibuat agar dapat mengenali atau dikenali oleh *hardware*. Contohnya, sebelum menggunakan *printer*, biasanya pengguna



Gambar 4.12 Valkyrie, Humanoid Buatan NASA

Sumber: Dokumen Kemendikbud, 2021

akan memasang sebuah program yang disebut *driver* agar komputer dapat untuk mengenali *printer*. *Driver* pada komputer adalah komponen system *software* yang berfungsi sebagai perangkat komunikasi antara sistem operasi dan *hardware*. *Driver* menggunakan kode biner untuk memerintah *hardware* melakukan perintah yang diberikan oleh sistem operasi dan mengambil data yang dikirimkan oleh *hardware*. Begitu juga selanjutnya, sistem operasi memberikan data ke *printer* agar *printer* bekerja mencetak dokumen. Program yang akan dipasang ialah termasuk *software*, dan *printer* yang dihubungkan dengan komputer atau *laptop* sebagai *hardware*. Dapat disimpulkan, telah terjadi hubungan yang memerlukan antarmuka antara *software*, komputer atau *laptop*, dan pengguna.

Contoh lainnya ialah ketika akan bermain musik. Kita dapat menggunakan alat tambahan menyerupai *flashdisk* (Makey Makey, misalnya pada modul https://brainsintheclouds.eu/?page_id=159) yang dapat ditancapkan ke komputer, kemudian dengan sebuah kabel, dapat disambungkan ke pisang, wortel, jeruk, alumunium, dan sebagainya. Untuk memainkannya, dapat membuka situs web atau aplikasi tertentu, lalu menyentuh barang yang disambungkan dengan kabel, dan musik pun dapat dimainkan. Ketika akan bermain aplikasi permainan menggunakan ponsel, pasti membutuhkan *software* aplikasi permainan tersebut sehingga terjadi interaksi antarmuka antara ponsel sebagai *hardware* dan *software* aplikasi permainan.

D. Sistem Operasi

Sistem operasi (OS) adalah perangkat lunak sistem yang mengelola perangkat keras komputer, sumber daya perangkat lunak, dan menyediakan layanan umum untuk program komputer. Sistem operasi berfungsi untuk mengendalikan, mengontrol atau memberikan koneksi antarperangkat keras komputer.

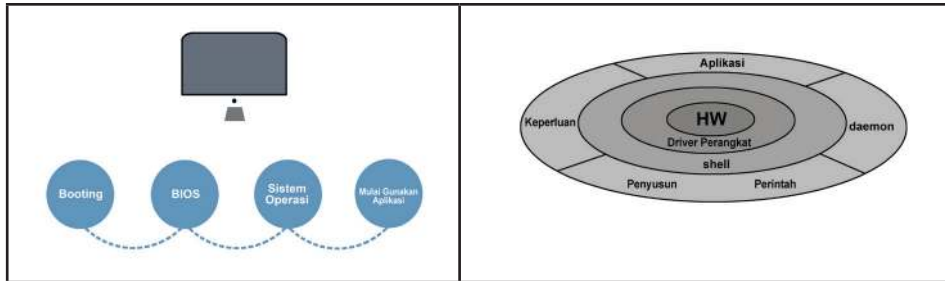
Contoh dari sistem operasi ialah MS Windows, Linux, MacOS, Android, iOS, dan sebagainya. Sistem operasi berfungsi untuk mengelola proses, memori utama, mengelola file, mengelola penyimpanan sekunder, serta melakukan proteksi dan keamanan.



Gambar 4.13 Beberapa Contoh Sistem Operasi

Sumber: Dokumen Kemendikbud, 2021

Setelah komputer yang sudah diinstal sistem operasi MS Windows dihidupkan dan proses *booting* (penyalan awal komputer) selesai, secara otomatis, sistem akan mengarahkan ke login MS Windows. Namun, jika baru untuk pertama kalinya dan tidak menggunakan password, tampilan login MS Windows akan dilewati dan berlanjut ke tampilan desktop.



Gambar 4.14 Alur Sistem Operasi pada Komputer dan “Lapisan Bawang” Sistem Komputer

Sumber: Dokumen Kemendikbud, 2021

Pada saat sebuah komputer dalam keadaan hidup, Sistem Operasi (SO) hidup di atas perangkat keras dan mengendalikan perangkat keras serta melayani perangkat lunak. Saat sebuah komputer dalam keadaan mati, tentu Sistem Operasi mati. Lalu, bagaimana Sistem Operasi mulai dihidupkan saat perangkat keras mulai dihidupkan? Pernahkah kalian memikirkan “*chicken and egg problem*” ini?

Semua program komputer termasuk Sistem Operasi dimuat dalam CPU saat dijalankan (hal ini akan dijelaskan pada bagian berikutnya). Pada saat sebuah komputer dihidupkan, CPU-nya belum mengandung program. Proses menyalakan komputer dan membuat Sistem Operasi mulai bekerja disebut *booting*. *Booting* adalah urutan *startup* (membangunkan) Sistem Operasi komputer saat dinyalakan. Pada saat *booting*, semua komponen perangkat keras dalam komputer dibangun. Ada 5 langkah penting saat *booting*: daya (*power*) dihidupkan, POST (*Power On Self Test*), memuat BIOS (*Basic Input Output System*), memuat sistem operasi, kemudian perangkat keras mengalihkan kontrol ke Sistem Operasi.

Fungsi Sistem Operasi secara lebih rinci dapat dijelaskan sebagai berikut.

1. Menyediakan antarmuka ke pengguna untuk berinteraksi dengan komputer. Tanpa antarmuka yang mudah digunakan, komputer akan sulit untuk digunakan.
2. Mengendalikan *input* dan *output*.
3. Mengelola perangkat keras dan periferal. Pada saat pengguna mengetik di *keyboard*, aplikasi yang dipakai akan bekerja sama dengan Sistem Operasi untuk menangani *signal* dari *keyboard* untuk diproses. Misalnya, saat menggunakan pemroses kata, *mouse* yang digerakkan

kursornya akan terlihat di layar, huruf-huruf yang diketikkan dari *keyboard* (perangkat *input*) akan muncul di layar monitor (perangkat keluaran). Pada modul AP, saat kalian membaca sebuah nilai variabel, aplikasi menunggu sampai pengguna mengetikkan data yang dibaca, dan program akan memroses data yang diketikkan sesuai dengan algoritma program kalian. Sistem Operasi akan menjadi perantara program aplikasi untuk berinteraksi dengan *keyboard* atau perangkat *input/output* lainnya.

4. Mengelola pemuatan perangkat lunak dan menjalankan perangkat lunak. SO menjalankan program aplikasi agar dapat berfungsi dengan baik. Program aplikasi dirancang untuk dapat berfungsi di atas sistem operasi tertentu. Hal ini memungkinkan program aplikasi mendapat keuntungan dari koneksi dan fungsi perangkat keras yang tersedia melalui sistem operasi karena aplikasi yang sama dapat dijalankan pada perangkat keras yang berbeda-beda.
5. Mengelola *file* (*copy, save, sort, delete*). Sistem Operasi mengelola sistem pengarsipan *file* yang dibuat untuk mengatur *file* dan direktori (*folder*). Adanya sistem *file* ini menyediakan cara yang konsisten untuk menyimpan dan mengambil data. Sistem Operasi juga bertanggung jawab atas penamaan, penyortiran, penghapusan, pemindahan, dan penyalinan *file/folder* (atas permintaan pengguna). Sistem Operasi mengelola sebuah Tabel pencarian yang digunakan untuk menghubungkan nama *file/folder* dan lokasi penyimpanan. Pengelolaan *file* dan *folder* ini dilakukan SO seperti kalian mengelola Buku Kerja Siswa.
6. Menangani interupsi dan kesalahan (*error*). Beberapa program dapat disimpan dalam RAM pada waktu bersamaan, tetapi prosesor hanya dapat memproses satu per satu. Melalui penggunaan sinyal interupsi yang reguler, Sistem Operasi dapat mengatur permintaan penggunaan prosesor dan menentukan prioritas program yang dijalankan. Hal ini membuat seakan-akan CPU menangani lebih dari satu program sekaligus (*multitasking*). Sistem Operasi juga dapat mendeteksi kesalahan yang biasanya ditangani tanpa perlu mengingatkan pengguna.
7. Mengelola prosesor. Sistem Operasi juga mengelola CPU. Saat sebuah program aplikasi diaktifkan, Sistem Operasi memuat program tersebut ke memori (RAM) dan CPU akan mengeksekusinya (penjelasan detail mengenai eksekusi program akan dijelaskan dalam pembahasan tentang mesin konseptual sederhana). CPU kemudian diinstruksi oleh Sistem Operasi untuk menjalankan program. Jadi, SO

bekerja sama dengan CPU untuk menjalankan sebuah program. Pada saat *multitasking* (menjalankan banyak program sekaligus), Sistem Operasi mengatur penggunaan waktu prosesor dengan menggilir pekerjaan. Bagian Sistem Operasi yang mengelola penggiliran CPU disebut *scheduler* (penjadwal).

8. Mengelola memori. Sistem operasi bertanggung jawab untuk mentransfer program ke memori dan dari memori. Sistem Operasi melacak penggunaan memori dan memutuskan berapa banyak yang harus diberikan untuk setiap program. Sistem Operasi juga memutuskan apa yang terjadi jika tidak ada cukup memori.
9. Mengelola keamanan. Sistem Operasi bertanggung jawab atas pembuatan akun pengguna dan kata sandi serta penerapannya. Sistem Operasi juga dilengkapi dengan banyak program utilitas, termasuk *firewall*.
10. Menangani komunikasi dengan jaringan.

Dari semua fungsi di atas, kalian dapat membayangkan, bahwa sebuah SO pun terdiri atas komponen-komponen dengan tugasnya masing-masing. Setiap komponen akan menjalankan tugas sesuai fungsinya yang spesifik. Pada kegiatan berikut ini, kalian akan lebih memahami salah satu fungsi Sistem Operasi dengan menjalankan tugas *scheduler*, yang mirip dengan tugas-tugas manusia menangani pekerjaan. Seperti halnya kalian berbagi peran dalam mengerjakan sebuah tugas, Sistem Operasi terdiri atas komponen-komponen dengan peran masing-masing yang disebutkan di atas, yang secara keseluruhan berfungsi sebagai Sistem Operasi.

1. Multitasking

Kalian tentu pernah melakukan “multitasking”, yaitu mengerjakan beberapa pekerjaan sekaligus, misalnya sambil merajut, nonton TV, bahkan sesekali menengok HP. Sambil menyapu, kalian mendengarkan musik bahkan ikut bernyanyi. Manusia mempunyai kemampuan *multitasking*, walaupun untuk beberapa kondisi, perlu dilakukan dengan hati-hati, misalnya sangat berbahaya menonton video sambil menyetir mobil. *Multitasking* yang tidak dikendalikan dengan baik, belum tentu menambah efisiensi dan hasilnya belum tentu baik. Misalnya, belajar sambil menonton sepak bola dapat memecah perhatian sehingga kalian tidak belajar dengan baik. Ketika kalian sedang belajar sambil mendengarkan lagu, kalian berhenti ketika ibu memanggil untuk makan malam (ini yang disebut interupsi).

Sebuah komputer yang sedang melakukan multitasking, misalnya saat penggunaanya sedang menjalankan aplikasi pengolah kata, aplikasi pengolah lembar kerja, aplikasi presentasi, dan aplikasi Paint untuk menggambar.

Pengguna memindahkan sepotong teks dari satu aplikasi ke lainnya lewat *clipboard* yang sebetulnya juga sebuah “aplikasi”. Diam-diam, jam yang tertulis di pojok layar juga sedang bekerja. Jika sedang terhubung ke jaringan, tiba-tiba email masuk dan komputer menampilkan pesan. Komputer dan SO sangat “sibuk”, bukan?



(Sumber: Dokumen Kemdikbud, 2021)

Seperti dijelaskan di atas, salah satu fungsi sistem operasi ialah menangani *multitasking*. Sistem Operasi tidak menangani *multitasking* seperti manusia karena komputer hanya mempunyai satu prosesor, dan prosesor itu yang menjalankan program (lihat aktivitas mesin super konseptual tentang bagaimana CPU menjalankan program dengan langkah sangat rinci).

Sistem Operasi dapat melakukan *multitasking* dengan menjalankan algoritma “*round robin*” (RR). Ya, sebuah algoritma karena Sistem Operasi adalah sebuah program juga. Prinsip dari algoritma penjadwalan *round robin* dijelaskan sebagai berikut.

Round-Robin (RR) ialah salah satu algoritma yang digunakan oleh penjadwal proses (*process scheduler*) dalam sebuah sistem operasi. Pada algoritma RR, ditentukan suatu slot waktu (*time slice*) yang akan dialokasi ke setiap proses dalam porsi yang sama dan dalam urutan melingkar, menangani semua proses tanpa prioritas. Penjadwalan RR sederhana, dan mudah diterapkan. Penjadwalan RR dapat diterapkan pada masalah penjadwalan lainnya, seperti penjadwalan paket data di jaringan komputer. Nama algoritma ini berasal dari prinsip *round-robin*, di mana setiap orang mengambil bagian yang sama dari sesuatu secara bergantian.

Agar proses dikerjakan secara adil, penjadwal RR memberikan setiap pekerjaan slot waktu atau penyisihan waktu CPU, dan menginterupsi pekerjaan belum selesai saat itu. Pekerjaan dilanjutkan saat slot waktu berikutnya ditetapkan bagi proses itu. Jika proses selesai atau mengubah statusnya menjadi menunggu selama slot waktu yang diberikan, penjadwal memilih proses pertama dalam antrian siap untuk dieksekusi. Dengan tidak ada banyak pekerjaan yang dilakukan, atau jika slot waktu relatif besar terhadap ukuran pekerjaan, proses yang menghasilkan pekerjaan besar akan lebih banyak dikerjakan daripada proses lainnya.

Misalnya, jika slot waktu ialah 100 milidetik (*mili second/ms*), dan job1 membutuhkan total waktu 250 ms untuk menyelesaikannya, penjadwal RR akan menangguhkan pekerjaan setelah 100 ms dan memberikan waktu pada pekerjaan lain di CPU. Setelah pekerjaan lain memiliki bagian yang sama (masing-masing 100 ms), job1 akan mendapatkan alokasi waktu CPU lain dan siklus akan berulang. Proses ini berlanjut hingga pekerjaan selesai dan tidak membutuhkan waktu lagi di CPU.

Job1 membutuhkan 250 ms untuk dapat diselesaikan, dengan slot 100 ms

Alokasi pertama 100 ms.

Alokasi kedua 100 ms.

Alokasi ke-3 100 ms tetapi job1 selesai dan diakhiri [ada 50 ms.

Jadi, waktu CPU untuk job1 = 250 ms.

Ada dua pendekatan algoritma untuk menyelesaikan RR *scheduler*.

Algoritma Pertama (dengan algoritma ini, CPU tidak pernah berhenti)

1. Selama periode satu slot waktu (kuantum): jika ada job selesai, hapus dari antrean, ambil berikutnya.
2. Di akhir satu kuantum: antrekan kembali, ambil giliran berikutnya

Algoritma Kedua

1. Selama Periode Satu Kuantum: Jika Ada Job Selesai, Hapus Dari Antrean, Tunggu Sampai Akhir Kuantum.
2. Di Akhir satu kuantum: antrekan kembali, ambil giliran berikutnya.

Tabel berikut menunjukkan waktu kedatangan dan waktu eksekusi dari beberapa proses, dengan slot 100 ms dan eksekusi dari proses-proses tersebut.

Tabel 4.1 Waktu Kedatangan dan Eksekusi Proses

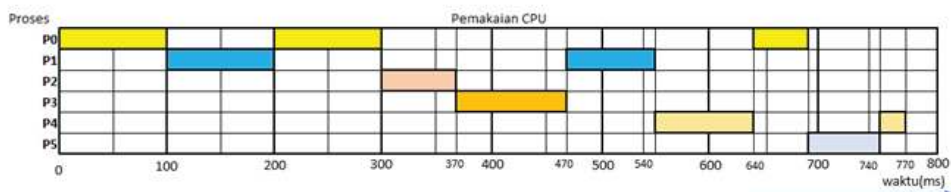
Kuantum = 100 ms

| Proses | Waktu Kedatangan | Waktu Eksekusi (ms) |
|--------|------------------|---------------------|
| P0 | 0 | 250 |
| P1 | 50 | 170 |
| P2 | 120 | 70 |
| P3 | 170 | 100 |
| P4 | 200 | 130 |
| P5 | 350 | 50 |
| Total | | 770 |

SIMULASI PENJADWALAN PROSES

| Waktu | Antrian | | | | | | Selesai | Deskripsi Kejadian |
|-------|---------|----|----|----|----|----|---------|--------------------------------------|
| 0 | P0 | | | | | | | P0 datang, diproses |
| 50 | P0 | P1 | | | | | | P1 datang masuk antrian |
| 100 | P1 | P0 | | | | | | P1 diproses, P0 diantrikan kembali |
| 120 | P1 | P0 | P2 | | | | | P2 datang masuk antrian |
| 170 | P1 | P0 | P2 | P3 | | | | P3 datang masuk antrian |
| 200 | P0 | P2 | P3 | P1 | P4 | | | P0 diproses, P4 datang masuk antrian |
| 300 | P2 | P3 | P1 | P4 | P0 | | | P2 diproses, P0 diantrikan kembali |
| 350 | P2 | P3 | P1 | P4 | P0 | P5 | | P5 datang masuk antrian |
| 370 | P3 | P1 | P4 | P0 | P5 | | P2 | P2 selesai, P3 diproses |
| 470 | P1 | P4 | P0 | P5 | | | P3 | P3 selesai, P1 diproses |
| 540 | P4 | P0 | P5 | | | | P1 | P1 selesai, P4 diproses |
| 640 | P0 | P5 | P4 | | | | | P0 diproses, P4 diantrikan kembali |
| 690 | P5 | P4 | | | | | P0 | P0 selesai, P5 diproses |
| 740 | P4 | | | | | | P5 | P5 selesai, P4 diproses |
| 770 | | | | | | | P4 | P4 selesai |

(a)



(b)

Gambar 4.15 (a) Simulasi Penjadwalan Proses, (b) Simulasi CPU

Sumber: Dokumen Kemendikbud, 2021

Nah, perancang sistem dapat mengatur kuantum yang optimal jika perilaku pekerjaan (*job*) yang harus ditangani oleh komputer diketahui. Ini membutuhkan berpikir komputasional. Pada komputer personal, hal ini tidak penting. Namun, dalam sebuah sistem komputer *mainframe* atau sistem komputasi dengan banyak mesin yang melayani banyak pengguna dengan karakteristik masing-masing, mengoptimasi pelaksanaan pekerjaan menjadi sangat penting.

Interpretasi Diagram!

Jawablah pertanyaan berikut ini.

1. Mengacu ke simulasi eksekusi tersebut, pekerjaan (proses) mana yang selesai terlebih dahulu dan mana yang selesai paling akhir?
2. Kesimpulan apa saja yang kalian dapatkan?
3. Dari dua algoritma yang dijelaskan di atas, algoritma mana yang dipilih?

Kalian juga dapat membuat diagram eksekusi yang lebih visual dengan sticky notes, untuk mempermudah interpretasinya.



Ayo Lakukan

Aktivitas Kelompok

Aktivitas SK-K10-01-U Simulasi Multitasking (*Unplugged*)

Pada aktivitas ini, kalian berlatih untuk menerapkan sistem *multitasking* dengan menggunakan simulasi dan permainan peran (*role play*).

Apa yang kalian perlukan?

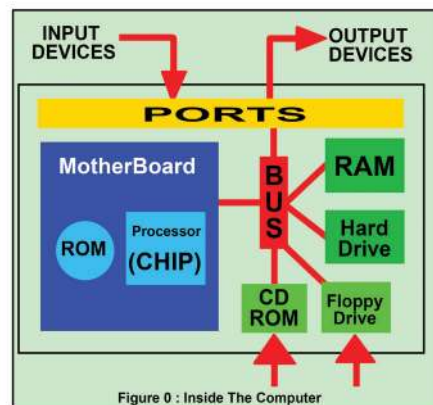
1. Papan tulis atau *whiteboard*
2. Spidol *boardmarker* berwarna atau kapur berwarna
3. *Sticky note* berwarna

Langkah-Langkah:

1. Setiap kelompok akan berbagi.
 - a. Berperan sebagai “*scheduler*” (penjadwal) yang merupakan bagian dari SO. *Scheduler* mengelola permintaan layanan dan antrean, mencatatnya dalam lembar catatannya. Setiap job yang datang akan memberi tahu sebelumnya berapa lama waktu eksekusi yang dibutuhkannya.
 - b. CPU (yang menjalankan job, menahannya selama mendapat giliran) dan melepaskannya saat slot waktu habis.
 - c. Bus yang akan membawa dan mengembalikan “job” ke antrean yang dikelola *scheduler* berikut statusnya (selesai, atau masih perlu berapa lama lagi).
2. Guru akan memberikan sebuah tabel yang isinya waktu kedatangan beberapa *job* untuk dilayani oleh SO.
3. Simulasikan pelaksanaan pekerjaan mulai kedatangan job yang pertama sampai semua *job* selesai dikerjakan.

2. Cara Komputer Bekerja

Sekarang, kita akan mempelajari mekanisme kerja pemroses karena pemroses pun memiliki banyak komponen yang harus melakukan interaksi satu sama lain. Secara ringkas, Gambar 4.16 menggambarkan diagram kotak arsitektur sederhana sebuah komputer di mana pemroses (kotak besar) menerima masukan dari piranti masukan dan menghasilkan



Gambar 4.16 Komponen Internal pada Komputer

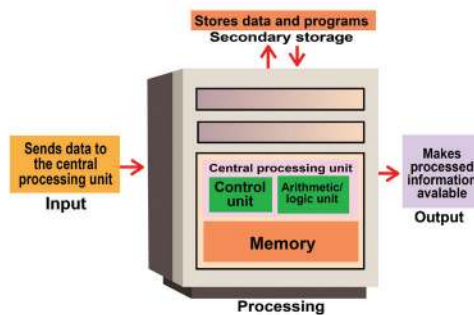
Sumber: Dokumen Kemendikbud, 2021

keluaran melalui *port*. Komponen pada pemroses (kotak besar) terdiri atas *processor* (CPU) yang berupa chip, ROM, RAM, Hard drive, CD ROM, dan *floppy drive* (sekarang tidak populer). CD ROM dan *Floppy drive* ialah alat yang terhubung dengan media penyimpanan eksternal, yaitu (CD dan *floppy disk*). Setiap komponen tersebut terhubung dengan BUS.

Berikut ini akan dijelaskan mengenai CPU, Bus, Controller, dan memori utama.

Central Processing Unit (CPU)

Sekarang, kita akan melihat apa yang dikerjakan oleh pemroses sebuah komputer yang tak terlihat, yaitu sebuah pusat pengontrol yang mengonversi data input menjadi output. Pengontrol ini disebut CPU (*Central Processing Unit*). CPU adalah sebuah sirkuit elektronik yang sangat kompleks, yang bertugas mengeksekusi instruksi yang disimpan sebagai program. Semua komputer, mulai yang kecil seperti ponsel pintar sampai yang besar dan canggih, harus mempunyai CPU. Seperti ditunjukkan pada Gambar 4.17, CPU terdiri atas dua bagian, yaitu CU (*control unit*) dan ALU (*Arithmetic/Logic Unit*). Setiap bagian mempunyai fungsinya masing-masing.



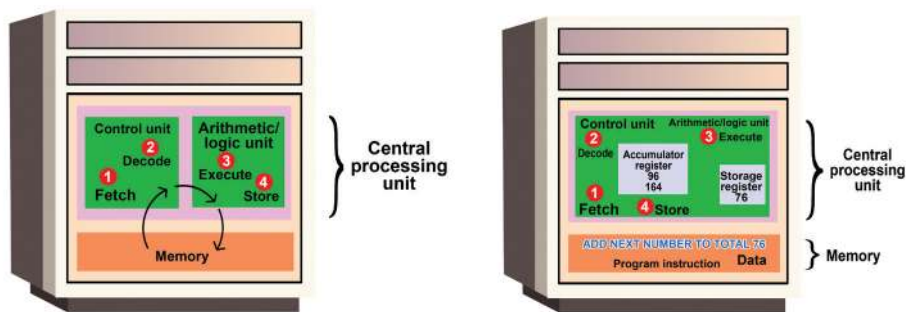
Gambar 4.17 Central Processing Unit
Sumber: Dokumen Kemendikbud, 2021

Sebelum kita membahas CU dan ALU secara rinci, kita perlu mempelajari penyimpanan data dan hubungannya dengan CPU. Komputer mempunyai dua jenis tempat penyimpanan: primer (*primary storage*) dan sekunder (*secondary storage*). CPU berinteraksi dengan tempat penyimpanan primer atau memori utama (*main memory*), mengacu ke data dan instruksi yang disimpan di dalamnya. Dengan alasan ini, memori dibahas terkait CPU, walaupun sebetulnya secara teknis, memori merupakan bagian terpisah dari CPU.

Ingat, bahwa memori utama komputer menyimpan data sementara saja, saat komputer mengeksekusi sebuah program. Memori sekunder menyimpan data permanen atau semi permanen dalam media lain, yaitu media magnetik atau optik. Disket dan CD-ROM yang sering kita lihat sebagai bagian dari PC adalah peranti sekunder, sama halnya dengan *hard disk*. Memori sekunder akan dibahas pada bagian berikutnya.

a. Bagaimana CPU Menjalankan Instruksi Program

Mari, kita periksa cara unit pengolah pusat (CPU), dalam hubungannya dengan memori, menjalankan program komputer. Kita akan melihat bagaimana hanya satu instruksi dalam program yang dijalankan. Faktanya, kebanyakan komputer saat ini hanya dapat menjalankan satu instruksi pada satu waktu, meskipun mereka menjalaninya dengan sangat cepat. Banyak komputer pribadi dapat menjalankan instruksi dalam waktu kurang dari sepersejuta detik. Komputer yang dikenal sebagai superkomputer dapat menjalankan instruksi dalam waktu kurang dari sepemiliar detik.



Gambar 4.18 Siklus Mesin pada Komputer Sebelum dan Setelah Bekerja

Sumber: Dokumen Kemendikbud, 2021

Sebelum instruksi dapat dieksekusi, instruksi program dan data harus ditempatkan ke dalam memori dari perangkat input atau perangkat penyimpanan sekunder. Seperti yang ditunjukkan Gambar 4.18 (sebelah kiri), setelah data dan instruksi yang diperlukan berada dalam memori, unit pemrosesan pusat melakukan empat langkah berikut untuk setiap instruksi.

1. Unit kontrol mengambil (mendapat) instruksi dari memori.
2. Unit kontrol menerjemahkan instruksi (memutuskan apa artinya) dan memerintahkan agar data yang diperlukan dipindahkan dari memori ke ALU (unit aritmatika/logika). Dua langkah pertama ini bersama-sama disebut waktu instruksi, atau waktu-I.
3. Unit aritmatika/logika menjalankan instruksi aritmatika atau logika. Artinya, ALU diberikan kendali dan melakukan operasi aktual pada data.
4. Unit aritmatika/logika menyimpan hasil operasi ini dalam memori atau register. Langkah 3 dan 4 bersama-sama disebut waktu eksekusi, atau waktu-E.

Unit kontrol selanjutnya memerintahkan memori untuk mengirimkan hasilnya ke perangkat keluaran atau perangkat penyimpanan sekunder. Kombinasi waktu-I dan waktu-E disebut siklus mesin. Gambar 4.18 (kanan) menunjukkan instruksi yang melalui siklus mesin.

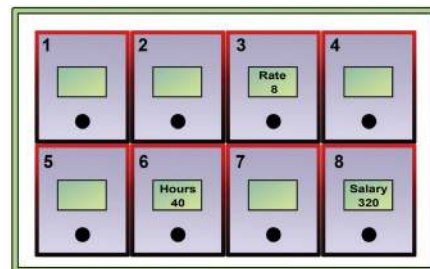
CPU memiliki *clock* internal yang menghasilkan detak (*pulse*) dengan kecepatan tetap untuk menyinkronkan semua operasi komputer. Sebuah instruksi siklus mesin tunggal dapat terdiri atas sejumlah besar sub-instruksi, yang masing-masing harus mengambil setidaknya satu siklus *clock*. Setiap jenis CPU dirancang untuk memahami sekumpulan instruksi tertentu yang disebut *instruction set*. Sama seperti, ada banyak bahasa berbeda yang dipahami orang, setiap jenis CPU memiliki *instruction set* yang dimengertinya. Oleh karena itu, CPU yang digunakan untuk PC merk Compaq tidak dapat memahami kumpulan instruksi dari CPU lain (misalnya Macintosh).

Pada memori, dapat tersimpan instruksi program dan juga data. Bagaimana *control unit* bisa membedakannya?

Lokasi dalam memori untuk setiap instruksi dan setiap bagian data diidentifikasi oleh sebuah alamat. Artinya, setiap lokasi memiliki nomor alamat, seperti *locker* (kotak penyimpanan) di perpustakaan atau di sekolah. Seperti kotak penyimpanan memiliki nomor yang tetap, isi kotak penyimpanan bisa berbeda di suatu waktu, bisa berisi tas, bisa berisi buku, atau berisi *tumbler* tempat minum.

Seperti hal kotak penyimpanan, memori bisa berisi instruksi atau data. Instruksi lama bisa diganti dengan instruksi baru, demikian juga data lama bisa diganti dengan data baru, tetapi memori tetap memiliki alamat yang sama. Tidak seperti kotak penyimpanan, lokasi memori hanya dapat menampung sejumlah data dalam ukuran byte.

Gambar 4.19 menunjukkan bagaimana program memanipulasi data dalam memori. Sebuah program penggajian, misalnya, dapat memberikan instruksi untuk meletakkan data tarif (gaji-per jam) di lokasi kotak 3 dan jumlah jam kerja di lokasi kotak 6. Untuk menghitung gaji karyawan, instruksi untuk komputer, yaitu mengalikan data di lokasi kotak 3 dengan data di lokasi kotak 6 dan pindahkan hasilnya ke lokasi kotak 8. Pemilihan lokasi dapat dilakukan di mana saja yang belum digunakan. Pemrogram yang menggunakan bahasa pemrograman tidak perlu tahu nomor alamat mesin yang sebenarnya, karena setiap alamat data disebut dengan nama yang menjadi alamat simbolis. Dalam contoh ini, nama alamat simbolis adalah Tarif, Jam, dan Gaji.



Gambar 4.19 Memory Address seperti Kotak Surat

sumber: Dokumen Kemendikbud, 2021

b. Mesin Konseptual Sederhana

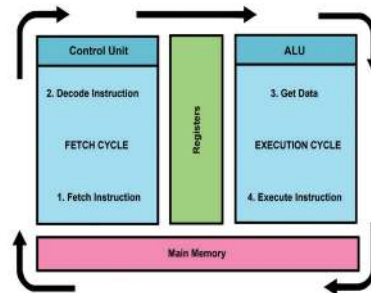
Tahukah kalian, bahwa kita bisa membuat abstraksi sebuah mesin komputer dengan menciptakan sebuah mesin konseptual sederhana?

Kalian diberi sebuah mesin konseptual sederhana ciptaan Mr. ALGO dan menyimulasikan cara kerjanya, yang seperti cara kerja sebuah komputer, tetapi dengan lebih sederhana. Pada bagian ini, kalian akan memahami bahwa instruksi program dalam bahasa yang lebih dekat ke manusia harus diterjemahkan menjadi instruksi dalam bahasa mesin untuk dapat dijalankan. Ingat bahwa dengan struktur komputer yang terdiri atas input, output, memori dan CPU, komputer hanya dapat membaca data dari perangkat masukan, menulis data ke perangkat keluaran, menyalin data dari CPU ke memori atau dari memori ke CPU, dan melakukan perhitungan aritmatika dan logika. Dengan kemampuan tersebut, kita dapat menuliskan program komputer yang beragam dan luar biasa daya gunanya!

Program komputer terdiri atas sekumpulan instruksi. Instruksi yang dijalankan oleh mesin harus dikenali oleh CPU yang disebut bahasa mesin. Bagaimana sebuah komputer menjalankan sebuah program dalam bahasa mesin? Komputer akan menjalankan (mengeksekusi) perhitungan dengan langkah yang disebut “*fetch execute cycle*” (siklus ambil dan jalankan). *Fetch execute cycle* ialah operasi yang paling mendasar dalam komputer, yang juga disebut “*fetch decode execute cycle*”. Selama *fetch execute cycle*, mesin komputer akan mengambil instruksi dari memori dan menjalankan instruksi tersebut sesuai dengan jenis instruksinya. Mengapa disebut *fetch execute cycle*? Siklus untuk *fetching*, *decoding*, dan *executing* sebuah instruksi ini akan diulang-ulang oleh CPU selama mesin komputer hidup!

Gambaran siklusnya ditunjukkan dalam Gambar 4.20 yang menunjukkan lebih detail bagian-bagian dari CPU.

1. CU akan mengambil instruksi (Fetch).
2. CU akan menerjemahkan instruksi tersebut harus melakukan apa, misalnya menyimpan data, menghitung, atau lainnya. Ingat, bahwa komputer hanya dapat melakukan perhitungan dan menyimpan/mengambil, atau mengirimkan data ke *input/output device*.

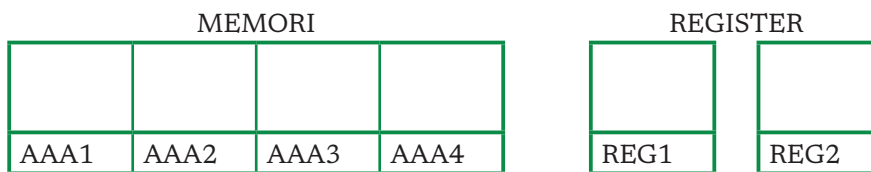


Gambar 4.20 Siklus Ambil dan Jalankan (Fetch Execute Cycle)

Sumber: Dokumen Kemendikbud, 2021

3. ALU akan mengambil data yang diperlukan untuk menjalankan instruksi, dan data yang sedang diproses disimpan dalam Register.
4. ALU menjalankan instruksi.

Berikut ini gambaran Memori dan Register dari sebuah mesin ciptaan Mr. ALGO tersebut. Alamat pada mesin ini dinyatakan dalam kode Heksadesimal. Memori dibagi-bagi menjadi kotak-kotak. Setiap kotak mempunyai alamat dan dapat berisi data. Misalnya, mesin ciptaan kalian memiliki kapasitas memori untuk menampung 4 data dengan alamat AAA1 s.d AAA4 dan CPU mempunyai 2 register dengan alamat REG1 dan REG2 yang ilustrasikan seperti pada gambar di bawah ini. Sebetulnya, semua data akan disimpan dalam bentuk biner, seperti yang telah dipelajari di jenjang SMP. Namun demikian, untuk kemudahan membaca, data semua ilustrasi, data tetap dituliskan dalam besaran desimal.



Sekarang, kalian akan mensimulasi *Fetch instruction cycle* tersebut untuk menjalankan instruksi perhitungan aritmatika sederhana sebagai berikut yang semula ditulis dalam bahasa yang lebih dipahami manusia (bahasa tingkat tinggi).

X=200
Y=100
Jumlah= X + Y
PRINT Jumlah

Mesin Konseptual Sederhana akan mengeksekusi perintah tersebut dalam beberapa langkah karena data disimpan dalam memori (disimpan dalam variabel X dan Y), sedangkan proses perhitungan penjumlahan harus dilakukan oleh ALU yang merupakan bagian dari CPU. Misalnya, nilai dari variabel X disimpan dalam alamat AAA1 dan nilai dari variabel Y disimpan dalam AAA2, dan nilai variabel Jumlah akan disimpan dalam alamat AAA4. Data harus dibawa ke Register untuk dijumlahkan.

Mesin Konseptual Sederhana menjalankan beberapa instruksi bahasa mesin sebagai berikut untuk menjalankan program di atas.

SIMPAN 100 AAA1
 SIMPAN 200 AAA2
 SALIN AAA1 REG1
 SALIN AAA2 REG2
 TAMBAH REG1 REG2
 SALIN REG2 AAA3
 PRINT AAA3

Eksekusi dari perintah-perintah tersebut secara berturut-turut ditunjukkan dengan ilustrasi sebagai berikut.

| Instruksi | Penjelasan | Peta Memori dan CPU |
|------------------|---|--|
| SIMPAN 100 AAA1 | Menyimpan data 100 ke dalam memori dengan alamat AAA1. | |
| SIMPAN 200 AAA2 | Menyimpan data 200 ke dalam memori dengan alamat AAA2. | |
| SALIN AAA1 REG1 | Menyalin data dari memori dengan alamat AAA1 ke register dengan alamat REG1. | |
| SALIN AAA2 REG2 | Menyalin data dari memori dengan alamat AAA2 ke register dengan alamat REG2. | |
| TAMBAH REG1 REG2 | Menambahkan data pada alamat REG1 dan data pada alamat REG2 dan mesin akan menyimpan hasilnya pada alamat REG2. *) Baca Penjelasan di bawah tabel. | |
| SALIN REG2 AAA3 | Menyalin data dari register REG2 ke memori AAA3. | |
| PRINT AAA3 | Mencetak data yang ada pada memori AAA3 ke layar monitor mesin untuk ditampilkan. | <div style="background-color: #4CAF50; color: white; border-radius: 15px; padding: 10px; display: inline-block;">300</div> |

Mesin Konseptual Sederhana tersebut hanya mempunyai 2 register (REG1 dan REG2), dan dirancang agar jika melakukan operasi aritmatika dua buah bilangan, hasilnya disimpan pada register yang menyimpan operan ke-2.

Mungkin saja, ada orang yang menciptakan mesin konseptual lain yang mempunyai 3 register. Seandainya, mesin mempunyai 3 register, bisa saja hasil penjumlahan disimpan pada register ke-3. Jumlah register dan bagaimana operasi dilakukan, serta hasil operasi aritmatika disimpan, itu ditentukan oleh penciptanya, merupakan spesifikasi mesin.

Kita juga dapat menciptakan mesin konseptual yang sangat sederhana, yang hanya mempunyai 1 register untuk menampung data, yang biasanya disebut AKUMULATOR karena semua operan dan hasil perhitungan disimpan di register tunggal. Perintah TAMBAH dilakukan dengan menambahkan data yang diambil dari memori ke satu-satunya register itu. Perintah Mesin misalnya:

| INSTRUKSI | HASIL |
|----------------------|---|
| LOAD <alamat> | Data yang disimpan pada alamat memori disimpan ke AKUMULATOR. |
| TAMBAH <alamat> | Data yang disimpan pada AKUMULATOR ditambah dengan data yang diambil dari alamat memori. |
| KURANG <alamat> | Data yang disimpan pada AKUMULATOR dikurangi dengan data yang diambil dari alamat memori. |
| KALI <alamat> | Data yang disimpan pada AKUMULATOR dikalikan dengan data yang diambil dari alamat memori. |
| BAGI <alamat> | Data yang disimpan pada AKUMULATOR dikurangi dengan data yang diambil dari alamat memori. |
| OUTPUT | Data AKUMULATOR dikirim ke perangkat keluaran. |
| INPUT <data><alamat> | Alamat diisi dengan data. |

Untuk menambahkan $100+200$, instruksi yang dilakukan oleh mesin dengan satu AKUMULATOR yang hanya mempunyai 4 alamat memori AAA1, AAA2, AAA3 dan AAA4 diberikan sebagai berikut.

| INSTRUKSI | ISI MEMORI dan CPU (Akumulator) | | | | | | | | | | | | |
|----------------|---|------|------|--|------------|--|--|------|------|------|------|--|------------|
| INPUT 100 AAA1 | <table><tr><td>100</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>AAA1</td><td>AAA2</td><td>AAA3</td><td>AAA4</td><td></td><td>AKUMULATOR</td></tr></table> | 100 | | | | | | AAA1 | AAA2 | AAA3 | AAA4 | | AKUMULATOR |
| 100 | | | | | | | | | | | | | |
| AAA1 | AAA2 | AAA3 | AAA4 | | AKUMULATOR | | | | | | | | |
| INPUT 200 AAA2 | <table><tr><td>100</td><td>200</td><td></td><td></td><td></td><td></td></tr><tr><td>AAA1</td><td>AAA2</td><td>AAA3</td><td>AAA4</td><td></td><td>AKUMULATOR</td></tr></table> | 100 | 200 | | | | | AAA1 | AAA2 | AAA3 | AAA4 | | AKUMULATOR |
| 100 | 200 | | | | | | | | | | | | |
| AAA1 | AAA2 | AAA3 | AAA4 | | AKUMULATOR | | | | | | | | |

| INSTRUKSI | ISI MEMORI dan CPU (Akumulator) | | | | |
|-------------|---------------------------------|------|------|------|------------|
| LOAD AAA1 | 100 | 200 | | | 100 |
| | AAA1 | AAA2 | AAA3 | AAA4 | AKUMULATOR |
| TAMBAH AAA2 | 100 | 200 | | | 300 |
| | AAA1 | AAA2 | AAA3 | AAA4 | AKUMULATOR |
| OUTPUT | 300 | | | | |

Jika kalian perhatikan, instruksi-instruksi di atas merupakan rangkaian perintah yang dilakukan untuk menghitung penjumlahan $100+200$. Instruksi yang diberikan bergantung pada spesifikasi mesinnya, dan mesin juga dapat mempunyai satu atau lebih register. Pencipta mesin yang akan merancang mesin-mesin komputer tersebut, dan mewujudkannya menjadi perangkat elektronik yang dapat berfungsi sesuai spesifikasi. Tentu saja, gambaran pelaksanaan instruksi di atas ialah penyederhanaan karena yang dilakukan oleh komputer sebenarnya jauh lebih rinci. Data 100 dan 200 dalam desimal pun akan direpresentasi dalam bentuk biner seperti yang kalian telah pelajari di tingkat SMP.

Unit pembelajaran ini memberikan gambaran bahwa ekspresi yang ditulis dalam bahasa yang dimengerti manusia harus diterjemahkan sangat detail agar dapat dieksekusi oleh mesin ciptaan kalian. Komputer yang selama ini kalian gunakan sebenarnya mirip dengan mesin konseptual sederhana tersebut, hanya kecepatan melaksanakan instruksi sangat cepat sekali.



Ayo Lakukan

Aktivitas Individu

Aktivitas Sk-K10-02-U Simulasi Eksekusi Perintah Dalam Mesin Ciptaan Mr. ALGO

Pada aktivitas ini, kalian akan bermain untuk memahami lebih detail lagi tentang bagaimana data disimpan dan diproses dalam sebuah komputer konseptual.

Apa yang kalian perlukan?

1. Sticky notes
2. Spidol
3. Kertas berukuran A0

Apa yang perlu kalian lakukan?

Guru kalian akan membuat kelompok yang terdiri atas 3-4 siswa.

Mr. ALGO menciptakan mesin konseptual sederhana dengan 5 alamat memori AEB1, AEB2, AEB3, AEB4, AEB5. CPU-nya memiliki 3 Register REG1, REG2 dan REG3. Spesifikasi instruksi-instruksinya sebagai berikut.

| Instruksi MESIN | Penjelasan |
|-----------------------------|---|
| SIMPAN <data><alamat> | Menyimpan data ke alamat memori yang dituju. |
| SALIN <alamat1><alamat2> | Menyalin data dari alamat1 untuk kemudian disimpan pada alamat2. Alamat dapat berupa alamat memori atau register. |
| TAMBAH REG1 REG2 | Melakukan penjumlahan data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3. |
| KALI REG1 REG2 | Melakukan perkalian data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3. |
| KURANG REG1 REG2 | Melakukan pengurangan data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3. |
| BAGI REG1 REG2 | Melakukan pembagian data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3. |
| PRINT <alamat> | Mencetak data yang ada pada alamat memori untuk ditampilkan pada monitor. |

Tugas kalian ialah menerjemahkan operasi matematika berikut ke dalam instruksi-instruksi supaya bisa dieksekusi oleh mesin ciptaan Mr. ALGO tersebut.

1. $3 + 8 \times 9$
2. $(1 + 2) \times (8 - 5)$
3. $2 \times 10 - 8 + 3$
4. $3 + 16/2$

Dengan menggunakan *sticky note*, jangan lupa untuk menggambarkan peta dari memori dan CPU pada setiap instruksi yang dieksekusi oleh mesin kalian ke dalam kertas berukuran A0.



Suatu hari, jika akan menjadi ahli perancang komputer, kalian akan berkenalan dengan *chips*, atau komponen elektronik yang membentuk mesin ciptaan. Tentu, harus membuat bahasa yang dipahami oleh komponen-komponen itu dengan memanfaatkan *interface* yang tersedia. Menantang, bukan?

Uji Kompetensi

1. Diberikan dua mesin ciptaan Mr Algo yang lain. Setiap spesifikasinya sebagai berikut ini:

Spesifikasi Mesin-1: operasi aritmatika 2 operan hasilnya disimpan pada register yang menyimpan operan pertama. Misalnya, TAMBAH REG1 REG2 hasilnya disimpan pada REG1.

Spesifikasi Mesin-2: operasi aritmatika 2 operan hasilnya disimpan pada register yang menyimpan operan kedua. Misalnya, TAMBAH REG1 REG2 hasilnya disimpan pada REG2.

Isilah sel yang kosong dengan isi memori AAA1, AAA2, AAA3, dan isi register REG1, REG2, REG3.

| Urutan Instruksi | Mesin 1 | Mesin 2 |
|------------------|---------|---------|
| SIMPAN 8 AAA1 | | |
| SIMPAN 9 AAA2 | | |
| SIMPAN 10 AAA3 | | |
| SALIN AAA1 REG1 | | |
| SALIN AAA2 REG2 | | |
| SALIN AAA3 REG3 | | |
| TAMBAH REG1 REG2 | | |
| TAMBAH REG2 REG3 | | |
| KALI REG2 REG3 | | |
| KALI REG1 REG3 | | |
| SALIN REG1 AAA3 | | |
| SALIN REG2 AAA2 | | |
| SALIN REG3 AAA1 | | |

Dengan mengacu ke mesin yang hanya mempunyai 1 akumulator dan 4 alamat memori AAA1, AAA2, AAA3 dan AAA4, tuliskan urutan instruksi dan gambaran isi memori dan register, untuk melakukan operasi perhitungan:

$$(1 + 4) \times (8 - 5)/5$$

dengan menggunakan memori se-efisien mungkin.



Ayo Kita Renungkan

Kalian telah belajar tentang sistem operasi dan interaksi antara komputer dan perangkat lain. Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Bayangkan seberapa penting sistem operasi dalam interaksi ini.
2. Dari materi ini, apakah kalian telah mengerti alur kerja sistem operasi?
3. Jelaskan dengan kata-kata sendiri, interaksi antara perangkat keras, perangkat lunak, dan pengguna yang kalian ketahui.
4. Menurut kalian, interaksi seperti apa yang paling sering dilakukan ketika kalian menggunakan komputer untuk keperluan pembelajaran?

Ingin Tahu Lebih?

Jika kalian tertarik dengan materi ini dan ingin mendalaminya lebih jauh, berikut buku yang bisa dibaca atau link yang bisa diakses.

1. George Beekman, Digital Planet: Tomorrow's Technology and You Edisi 10, Penerbit Pearson
2. Penjelasan tentang Sistem Operasi: https://en.wikipedia.org/wiki/Operating_system, https://www.tutorialspoint.com/operating_system/os_overview.htm
3. <https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading04.htm>