

“BACA DAN TULIS”  
DIEKSEKUSI  
“SALIN DAN TEMPEL”  
ATAU  
“TERIMA DAN KIRIM”

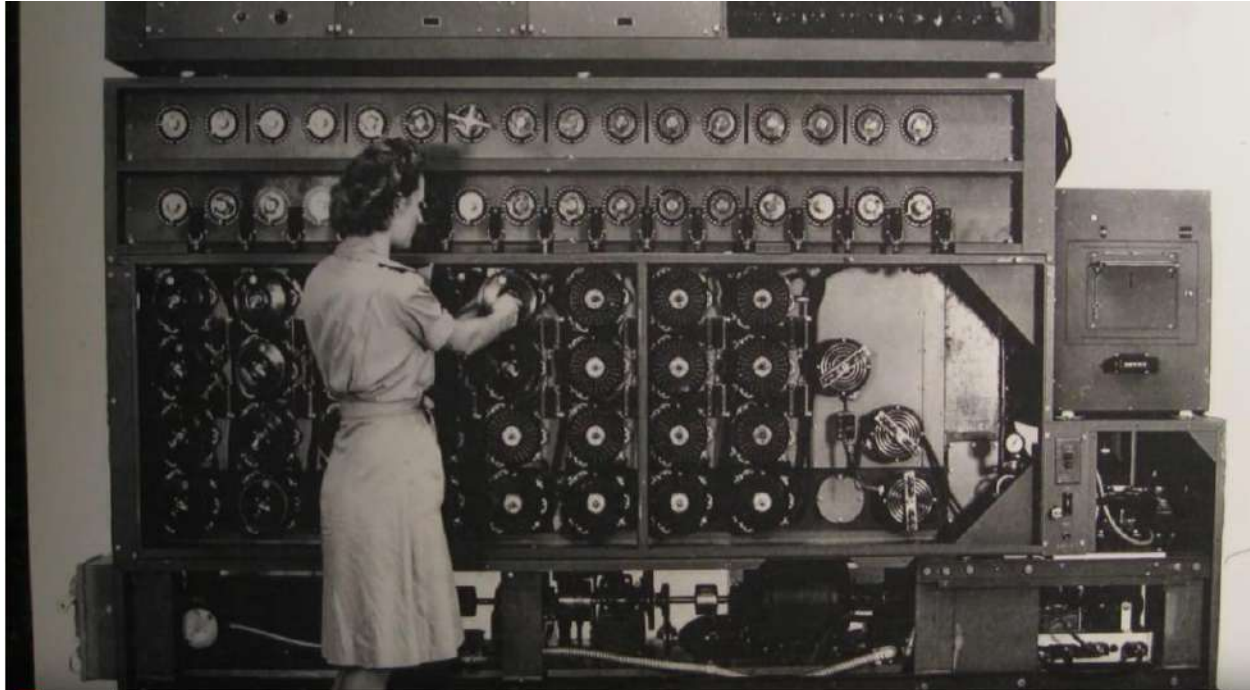
Buku Pembelajaran MKU  
Pengantar Teknologi Informasi  
(PTI)

Oleh  
Tim MKU PTI  
Edward Baringin O. Sihite, M.Sc  
Rutman Lumbantoruan, SE., M.M  
Risma Uly Manalu, S.Kom, MMSI  
Dikky A. Hutauruk, ST., M.Sc  
Dr. Dirk Roy Kolibu, M.Th  
Ir. Surjo Abadi, M.Sc  
Judo Ignatius Nempung, ST., M.T  
Dina Robiana Sitompul, S.Kom., MTI

Universitas Kristen Indonesia  
2020

## Bagian Pertama “CARA KERJA PERANGKAT KERAS DAN PERANGKAT LUNAK

**Tujuan : CPMK I Mampu mendefinisikan, menganalisis prinsip dan cara kerja perangkat keras dan perangkat lunak komputer. (S8, S9, S11, KU1, KU2, KK1, P1, P2, P3)**



Gambar 1. Komputer mekanik yang digunakan oleh Alan Turing pada akhir Perang Dunia Kedua. [5]

Deskripsi Bahasa Pemrograman. [6]

Fortran (FORMula TRANslator) dikembangkan oleh IBM Corporation pada pertengahan 1950-an untuk digunakan untuk ilmiah dan teknik aplikasi yang membutuhkan perhitungan matematika yang kompleks. Ini masih banyak digunakan dan versi terbarunya mendukung berorientasi objek pemrograman.

COBOL (COmmon Business Oriented Language) dikembangkan pada akhir 1950-an oleh produsen komputer, pemerintah AS, dan pengguna komputer industri berdasarkan bahasa dikembangkan oleh Grace Hopper, seorang perwira karir Angkatan Laut AS dan ilmuwan komputer. COBOL masih banyak digunakan untuk aplikasi komersial yang membutuhkan manipulasi besar dan tepat jumlah data. Versi terbarunya mendukung pemrograman berorientasi objek.

Pascal, penelitian pada 1960-an menghasilkan pemrograman terstruktur — pendekatan disiplin untuk menulis program yang lebih jelas, lebih mudah uji dan debug dan lebih mudah untuk dimodifikasi daripada program besar yang diproduksi dengan teknik sebelumnya. Salah satu hasil yang lebih nyata dari penelitian ini adalah pengembangan Pascal oleh Profesor Niklaus Wirth pada tahun 1971. Ini dirancang untuk pengajaran yang terstruktur pemrograman dan populer di kuliah untuk beberapa dekade.

Ada, berdasarkan Pascal, dikembangkan di bawah sponsor dari Departemen Pertahanan AS (DOD) selama tahun 1970 dan awal 1980-an. DOD menginginkan satu bahasa yang akan mengisi sebagian besar kebutuhannya. Bahasa berbasis Pascal dinamai Lady Ada Lovelace, putri penyair Lord Byron. Dia dikreditkan dengan menulis program komputer pertama di dunia pada awal 1800-an (untuk perangkat komputasi mekanis Analytical Engine dirancang oleh Charles Babbage). Ada juga mendukung berorientasi objek pemrograman.

Basic dikembangkan pada 1960-an di Dartmouth College untuk membiasakan pemula dengan teknik pemrograman. Banyak versi yang terbaru berorientasi objek.

C diimplementasikan pada tahun 1972 oleh Dennis Ritchie di Bell Laboratories. Ini awalnya dikenal secara luas sebagai bahasa pengembangan sistem operasi UNIX. Saat ini, sebagian besar kode untuk sistem operasi tujuan umum ditulis dalam C atau C ++.

Objective-C adalah bahasa berorientasi objek berdasarkan C. Itu dikembangkan pada awal 1980-an dan kemudian diakuisisi oleh NeXT, yang pada gilirannya diakuisisi oleh Apple. Ini telah menjadi bahasa pemrograman utama untuk sistem operasi OS X dan semua perangkat yang didukung iOS (seperti iPod, iPhone dan iPad).

Java Sun Microsystems pada 1991 mendanai penelitian internal perusahaan proyek yang dipimpin oleh James Gosling, yang menghasilkan C ++ berdasarkan bahasa pemrograman berorientasi objek disebut Java. Tujuan utama dari Java adalah untuk dapat menulis program yang akan berjalan di berbagai sistem komputer dan perangkat kontrol komputer. Ini adalah kadang-kadang disebut "tuliskan sekali, jalankan di mana saja." Java digunakan untuk mengembangkan aplikasi perusahaan skala besar, untuk meningkatkan fungsionalitas server web (komputer yang menyediakan konten kita lihat di browser web), untuk menyediakan aplikasi untuk perangkat konsumen (mis., ponsel cerdas, tablet, set-top televisi kotak, peralatan, mobil dan lainnya) dan banyak lainnya tujuan. Java juga merupakan bahasa utama untuk mengembangkan Android aplikasi smartphone dan tablet.

Visual Basic Bahasa Visual Basic Microsoft diperkenalkan di awal 1990-an untuk menyederhanakan pengembangan aplikasi Microsoft Windows. Versi terbarunya mendukung pemrograman berorientasi objek.

C #, tiga bahasa pemrograman utama berorientasi objek Microsoft adalah Visual Basic (berdasarkan Basic asli), Visual C ++ (berdasarkan C ++ dan C # (berdasarkan C ++ dan Java, dan dikembangkan untuk mengintegrasikan Internet dan web ke dalam aplikasi komputer).

PHP adalah berorientasi objek, "open-source" Bahasa "scripting" yang didukung oleh komunitas pengguna dan pengembang dan digunakan oleh banyak situs web termasuk Wikipedia dan Facebook. PHP adalah platform independen — implementasi ada untuk semua UNIX utama, Linux, Mac dan Windows sistem operasi. PHP juga mendukung banyak basis data, termasuk MySQL.

Perl (Ekstraksi Praktis dan Bahasa Laporan), salah satu yang paling banyak digunakan bahasa scripting berorientasi objek untuk pemrograman web, dikembangkan pada tahun 1987 oleh Larry Wall. Ini fitur kaya kemampuan dan fleksibilitas pemrosesan teks.

Python, bahasa scripting berorientasi objek lain, dirilis secara terbuka pada tahun 1991. Dikembangkan oleh Guido van Rossum dari Lembaga Penelitian Nasional untuk Matematika dan Ilmu Komputer di Amsterdam (CWI), Python banyak sekali belajar dari Modula3 — bahasa pemrograman sistem. Python "extensible" —itu dapat diperluas melalui kelas dan antarmuka pemrograman.

JavaScript adalah bahasa scripting yang paling banyak digunakan. Ini terutama digunakan untuk menambah kemampuan program ke halaman web — misalnya, animasi dan interaktivitas dengan pengguna. Itu disediakan dengan semua browser web utama.

Ruby on Rails Ruby — dibuat pada pertengahan 1990-an oleh Yukihiro Matsumoto — adalah sebuah

open-source, bahasa pemrograman berorientasi objek dengan sintaksis sederhana yang mirip dengan Perl dan Python. Ruby on Rails menggabungkan bahasa scripting Ruby dengan aplikasi web Rails kerangka kerja yang dikembangkan oleh 37Signals. Buku mereka, *Getting Real* (tersedia gratis di [getreal.37signals.com/toc.php](http://getreal.37signals.com/toc.php)), adalah harus dibaca untuk pengembang web. Banyak pengembang Ruby on Rails telah melaporkan peningkatan produktivitas dibandingkan bahasa lain saat mengembangkan aplikasi web intensif basis data. Ruby on Rails digunakan untuk membangun antarmuka pengguna Twitter.

Scala ([www.scala-lang.org/node/273](http://www.scala-lang.org/node/273))— singkat untuk “bahasa yang dapat diskalakan” - dirancang oleh Martin Odersky, seorang profesor di École Polytechnique Fédérale de Lausanne (EPFL) di Swiss. Dirilis pada tahun 2003, Scala menggunakan pemrograman berorientasi objek dan paradigma pemrograman fungsional dan dirancang untuk itu terintegrasi dengan Java. Pemrograman dalam Scala dapat mengurangi jumlah kode dalam aplikasi Anda secara signifikan. Twitter dan Foursquare menggunakan Scala.

## Bab 1 GARIS BESAR PENGANTAR TEKNOLOGI INFORMASI

Tujuan: Sub-CP-MK 1 Memahami secara garis besar mata kuliah Pengantar Teknologi Informasi dengan pemahaman RPS.

Waktu: Satu tatap muka.

### 1.1. Definisi Teknologi Informasi Dan Ilmu Komputer Berdasarkan Wikipedia

- **Teknologi Informasi**

Teknologi Informasi (TI), atau dalam bahasa Inggris dikenal dengan istilah Information technology (IT) adalah istilah umum untuk teknologi apa pun yang membantu manusia dalam membuat, mengubah, menyimpan, mengomunikasikan dan/atau menyebarkan informasi. TI menyatukan komputasi dan komunikasi berkecepatan tinggi untuk data, suara, dan video. Contoh dari Teknologi Informasi bukan hanya berupa komputer pribadi, tetapi juga telepon, TV, peralatan rumah tangga elektronik, dan peranti genggam modern (misalnya ponsel). [1]

Dalam konteks bisnis, Information Technology Association of America menjelaskan Pengolahan, penyimpanan dan penyebaran vokal, informasi bergambar, teks dan numerik oleh mikroelektronika berbasis kombinasi komputasi dan telekomunikasi. [2] Istilah dalam pengertian modern pertama kali muncul dalam sebuah artikel 1958 yang diterbitkan dalam Harvard Business Review, di mana penulis Leavitt dan Whisler berkomentar bahwa "teknologi baru belum memiliki nama tunggal yang didirikan. Kita akan menyebutnya teknologi informasi (TI)". [3] Beberapa bidang modern yang muncul dari teknologi informasi adalah generasi berikutnya teknologi web, bioinformatika, "Cloud Computing", sistem informasi global, Skala besar basis pengetahuan dan lain-lain.

- **Ilmu Komputer**

Ilmu komputer (bahasa Inggris: Computer Science), Secara umum diartikan sebagai ilmu yang mempelajari baik tentang komputasi, perangkat keras (hardware) maupun perangkat lunak (software). Ilmu komputer mencakup beragam topik yang berkaitan dengan komputer, mulai dari analisis abstrak algoritma sampai subyek yang lebih konkret seperti bahasa pemrograman, perangkat lunak, termasuk perangkat keras. Sebagai suatu disiplin ilmu, Ilmu Komputer lebih menekankan pada pemrograman komputer, dan rekayasa perangkat lunak (software), sementara teknik komputer lebih cenderung berkaitan dengan hal-hal seperti perangkat keras komputer (hardware). Namun, kedua istilah tersebut sering disalah artikan oleh banyak orang.

Tesis Church-Turing menyatakan bahwa semua alat komputasi yang telah umum diketahui sebenarnya sama dalam hal apa yang bisa mereka lakukan, sekalipun dengan efisiensi yang berbeda. Tesis ini kadang-kadang dianggap sebagai prinsip dasar dari ilmu komputer. Para ahli ilmu komputer biasanya menekankan komputer von Neumann atau mesin Turing (komputer yang mengerjakan tugas yang kecil dan deterministik pada suatu waktu tertentu), karena hal seperti itulah kebanyakan komputer digunakan sekarang ini. Para ahli ilmu komputer juga mempelajari jenis mesin yang lain, beberapa diantaranya belum bisa dipakai secara praktikal (seperti komputer neural, komputer DNA, dan komputer kuantum) serta beberapa diantaranya masih cukup teoretis (seperti komputer random and komputer oracle).

Ilmu Komputer mempelajari apa yang bisa dilakukan oleh beberapa program, dan apa yang tidak (komputabilitas dan intelegensia buatan), bagaimana program itu harus mengevaluasi suatu hasil (algoritma), bagaimana program harus menyimpan dan mengambil bit tertentu dari suatu informasi (struktur data), dan bagaimana program dan pengguna berkomunikasi (antarmuka pengguna dan bahasa pemrograman).

Ilmu komputer berakar dari elektronika, matematika dan linguistik. Dalam tiga dekade terakhir dari abad 20, ilmu komputer telah menjadi suatu disiplin ilmu baru dan telah mengembangkan metode dan istilah sendiri.

Departemen ilmu komputer pertama didirikan di Universitas Purdue pada tahun 1962. Hampir semua universitas sekarang mempunyai departemen ilmu komputer.

Penghargaan tertinggi dalam ilmu komputer adalah Turing Award, pemenang penghargaan ini adalah semua pionir di bidangnya.

Edsger Dijkstra mengatakan:

Ilmu komputer bukan tentang komputer sebagaimana astronomi bukan tentang teleskop

Fisikawan Richard Feynman mengatakan:

Ilmu komputer umurnya tidak setua fisika; lebih muda beberapa ratus tahun. Walaupun begitu, ini tidak berarti bahwa "hidangan" ilmuwan komputer jauh lebih sedikit dibanding fisikawan. Memang lebih muda, tapi dibesarkan secara jauh lebih intensif!

- Catatan tentang istilah 'Informatika' dan 'Ilmu komputer'

Dalam bahasa Indonesia, istilah Informatika diturunkan dari bahasa Prancis *informatique*, yang dalam bahasa Jerman disebut *Informatik*. Sebenarnya, kata ini identik dengan istilah *computer science* di Amerika Serikat dan *computing science* di Inggris. Namun, istilah *informatics* dalam bahasa Inggris memiliki makna yang sedikit berbeda, yaitu lebih menekankan pada aspek pengolahan informasi secara sistematis dan rasional.

## 1.2. RPS Pengantar Teknologi Informasi (PTI)

RPS PTI terlampir pada lampiran menarasikan materi pembelajaran perkuliahan selama 16 minggu yang mengacu pada CP (capaian pembelajaran). Dalam CP, Capaian Pembelajaran Lulusan Program Studi (CPL-Prodi) Sarjana dan Diploma Tiga yang dibebankan pada mata kuliah telah dirumuskan. Lewat CPL-Prodi diturunkan Capaian Pembelajaran Mata Kuliah (CPMK). CPMK PTI sebagai berikut:

1. CPMK I, Mampu mendefinisikan, menganalisis prinsip dan cara kerja perangkat keras dan perangkat lunak komputer. (S8, S9, S11, KU1, KU2, KK1, P1, P2, P3)
2. CPMK II, Mampu memahami fungsi dan proses kerja jaringan komputer serta cara kerja aplikasi perangkat lunak desktop dan web based. (S8, S9, S11, KU1, KU2, KK1, P1, P2, P3)
3. CPMK III, Mampu mencodingkan kalkulasi matematika kedalam pemrograman bahasa tingkat tinggi seperti bahasa C lalu dikompilasi ke bahasa tingkat rendah atau bahasa mesin atau bahasa biner sehingga bisa dieksekusi yang dapat dilihat dimonitor. (S8, S9, S11, KU1, KU2, KU5, KK1, KK2, P1, P2, P3)
4. CPMK IV, Mampu menyelesaikan studi kasus aplikasi teknologi informasi. (Proyek bersama kelompok) (S8, S9, S11, KU1, KU2, KU5, KK1, KK2, P1, P2, P3)

## 1.3. Proyek Bersama Kelompok

Dalam rangka memahami mata kuliah PTI diwajibkan untuk melakukan proyek bersama kelompok yang bertujuan untuk mengerti pemrograman bahasa tingkat tinggi (high level language) seperti bahasa C. Contoh program C yang sederhana yaitu menjumlahkan dua variabel yang ditulis sebanyak 10 line code sama prinsip cara kerjanya dengan program C yang sangat kompleks seperti program sistem operasi linux yang ditulis dalam bahasa C sebanyak 50.000 line code. Jadi dengan proyek instalasi sistem operasi linux di flashdisk, program C sistem operasi linux sebagai perangkat lunak dapat menggerakkan semua sumberdaya perangkat keras dari komputer sehingga komputer yang tidak memiliki perangkat lunak dapat digunakan oleh user untuk mengeksekusi aplikasi perangkat lunak website youtube.com atau google.com atau kompas.com.

#### 1.4. Daftar Pustaka

- [1] Williams / Sawyer, (2007), Using Information Technology terjemahan Indonesia, Penerbit ANDI, ISBN 979-763-817-0.
- [2] Longley, Dennis; Shain, Michael (1985), Dictionary of Information Technology (edisi ke-2), Macmillan Press, hlm. 164, ISBN 0-333-37260-3.
- [3] "information technology", Oxford English Dictionary (edisi ke-2), Oxford University Press, 1989 <http://dictionary.oed.com/>, diakses tanggal 20 November 2010 Tidak memiliki atau tanpa |title= (bantuan)
- [4] [Http://en.wikipedia.org/](http://en.wikipedia.org/); akses 7/12/2020.
- [5] [Https://www.youtube.com/](https://www.youtube.com/) TEDx Talks, Can we make quantum technology work? | Leo Kouwenhoven | TEDxAmsterdam; akses 7/12/2020.
- [6] Deitel, Paul; Deitel, Harvey (2014), C++ How to Program, Ninth Edition, Pearson, ISBN-13: 978-0-13-337871-9

#### 1.5. Pertanyaan Quis

1. Jelaskan yang anda ketahui tentang perangkat keras ?
2. Jelaskan yang anda ketahui tentang perangkat lunak ?
3. Ajukan pertanyaan kepada dosen pengampu ?

## Bab 2 Perangkat Keras

Tujuan: Sub-CP-MK 2 Mampu mendefinisikan, menganalisis prinsip dan cara kerja perangkat keras komputer.

Waktu: Dua tatap muka.

### 2.1. Eksplorasi Dunia Digital.

- Mega opportunity mengubah dunia ke teknologi .
- Dampak teknologi digital dalam kehidupan pengguna teknologi.
- Perkembangan evolusi komputer dan teknologi digital.
- Kegunaan dari bermacam-macam komputer.
- Teknologi internet mengubah peradaban.
- Perbedaan era informasi dari era-era sebelumnya.
- Pengaruh sosial dan etis akibat penggunaan teknologi informasi.

### 2.2. Komputer ada dimana-mana.

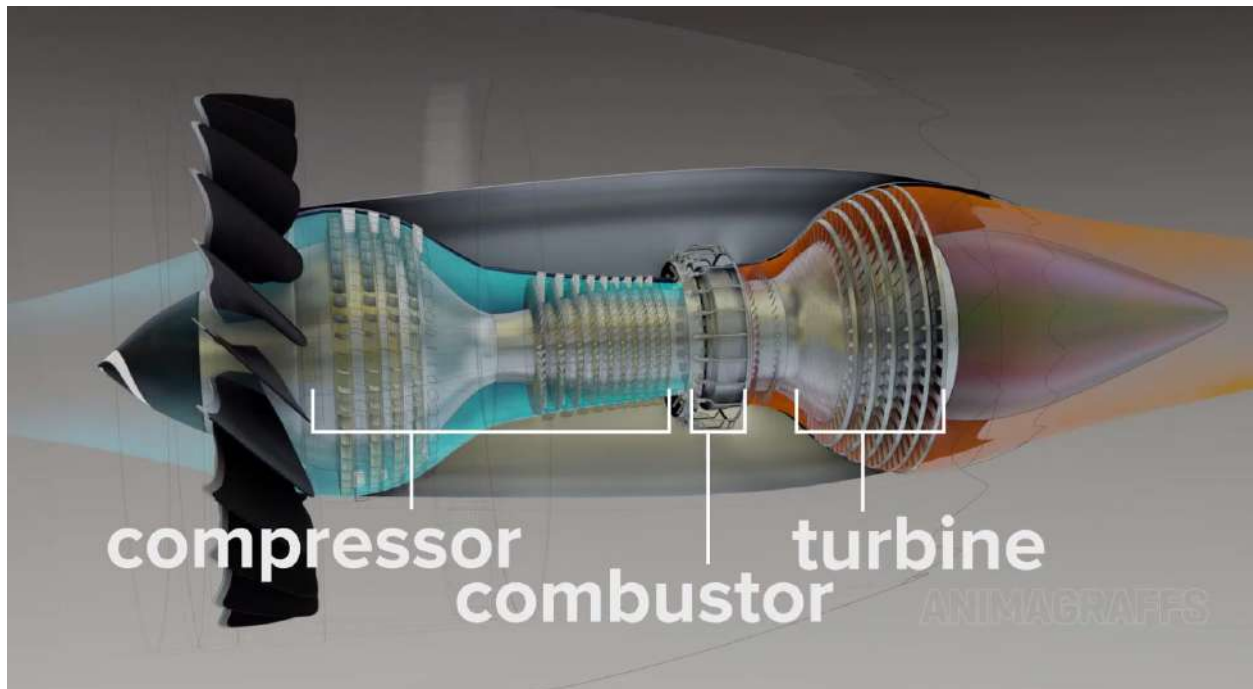
Apakah produk jet engine diproduksi tanpa alat bantu komputer sebagai perangkat kerasnya?



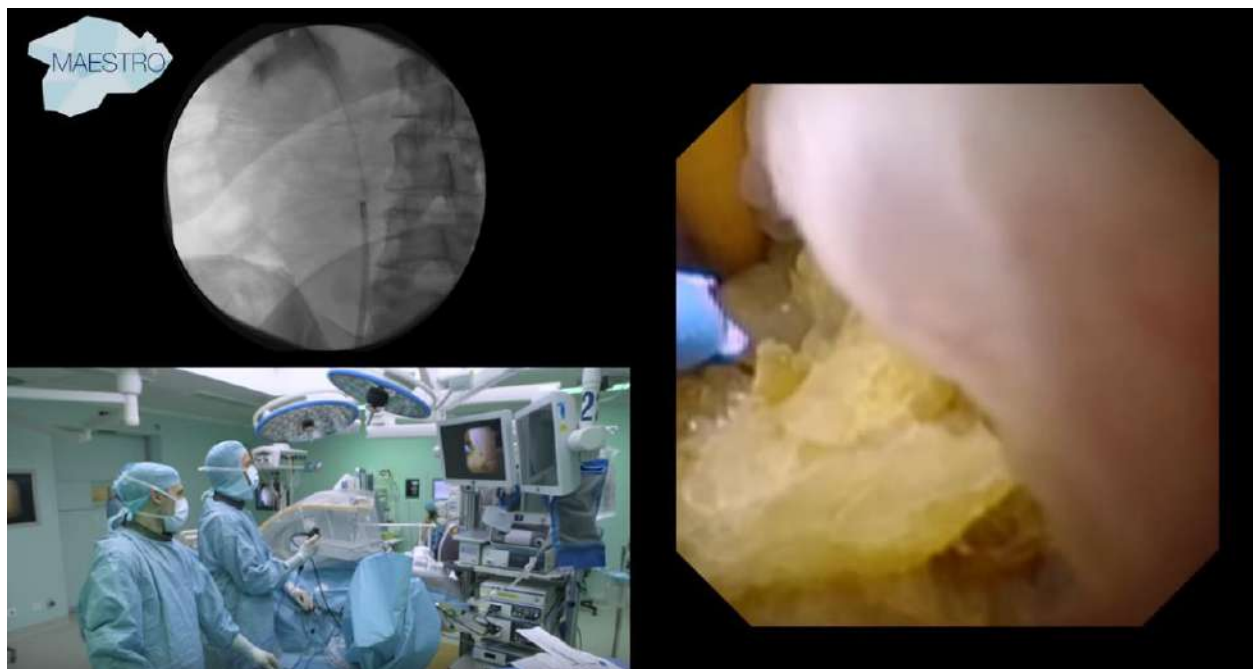
Gambar 2.1. Pabrik Rolls Royce memproduksi jet engine. [1]

Apakah membuat simulasi cara kerja jet engine diperlukan alat bantu komputer sebagai perangkat kerasnya pada gambar 2.2 ? Apakah dalam pembelajaran dunia pendidikan diperlukan animasi sehingga anak didik dapat memahami mata pelajaran matematika, biologi, fisika, kimia, IPA dan lain-lain untuk mengerti lebih cepat? Apakah <https://ppdb.jakarta.go.id> dapat diakses tanpa komputer atau smartphone untuk mendapatkan bangku di SD, SMP, SMA dan SMK ? Apa dokter dapat memecahkan batu ginjal tanpa bantuan komputer dan aplikasi perangkat lunak yang menuntun dokter pada gambar 2.3 ? Bagaimana dengan smartTV, smartphone, apakah smartTV dan smartphone adalah perangkat keras komputer? Bahkan belajar bahasa asingpun sudah dapat dipelajari lewat interaksi dengan aplikasi web. Juga layanan bank dapat diakses lewat daring dan segala macam transaksi pembayaran dapat dilakukan di internet. Jadi memang benar, komputer ada dimana-mana.





Gambar 2.2. Aplikasi perangkat lunak menganimasi cara kerja jet engine. [2]



Gambar 2.3. Operasi penghancuran batu ginjal dapat dilakukan dengan bantuan komputer sebagai alat bantu elektronika medis. [3]

### 2.3. Komputer dalam Perspektif

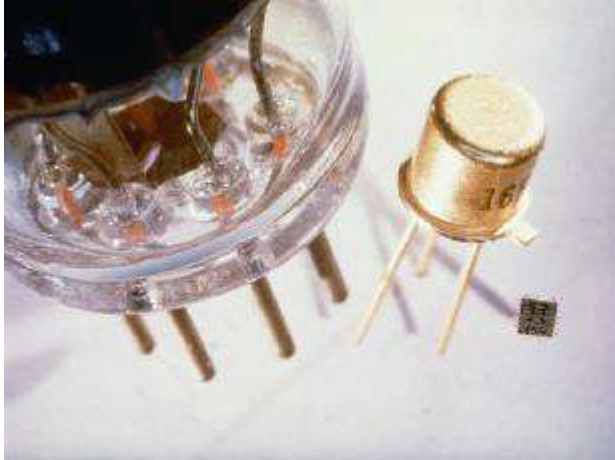
Sejarah singkat komputer:

- 1938—Konrad Zuse menyelesaikan Z1.
- 1939—Atanasoff-Berry Computer tercipta.
- 1943—Alan Turing mengembangkan Colossus.
- 1944—Mark I selesai, digunakan untuk menghitung ballistics tables.

- 1945—ENIAC tercipta.
- 1951—UNIVAC I (komputer komersial general-purpose pertama) diserahkan ke U.S. Census Bureau.

Hardware komputer:

- Komputer awal menggunakan vacuum tube (tabung hampa udara).
- Tabung hampa udara digantikan oleh transistor.
- Pertengahan 1960an, komputer mulai menggunakan integrated circuit yaitu silicon chip yang mengandung ratusan transistor.



Gambar 2.4. Vacuum tube, transistor, dan integrated circuit.

Kelebihan integrated circuit:

- Handal: Lebih tidak mudah gagal.
- Ukuran: Chip yang mungil menggantikan papan yang besar.
- Kecepatan: Listrik berpindah dalam jarak yang lebih kecil.
- Efisiensi: Chip menggunakan listrik yang lebih sedikit dan menghasilkan panas yang lebih sedikit.
- Biaya: Produksi massal membuat harga chip tidak mahal.

Penggunaan Komputer di Indonesia: [5]

- 1956 Bank Indonesia mulai menggunakan UNIVAC 1004.
- 1964 TNI angkatan darat di Bandung menggunakan IBM System 1401.
- 1967 ITB sebagai perguruan tinggi pertama yang menggunakan komputer, yaitu IBM System 1401.
- 1972 PUSILKOM UI mulai melakukan kegiatan operasional komputasi di lingkungan kampus UI.
- 1973 dan sampai awal tahun 1980-an terdapat beberapa lembaga pemerintah menggunakan lebih dari 1 unit komputer a.l. Pertamina, Bank Rakyat Indonesia (IBM S/370), BNI 1946 (IBM S/370, S/32), BAPINDO (Q1), Pupuk Sriwijaya (IBM S/370), PT Tambang Timah (NCR Century), PT Semen Cibinong (IBM S/3), Asuransi Jiwasraya (IBM S/370), PLN (IBM S/370, S/3, UNIVAC 90/30), Perumtel (NCR), Garuda Indonesian Airways (IBM S/370), DKI Jaya (IBM S/370), Pusat Reaktor Atom Bandung (PDP 11/34), Pupuk Kujang (IBM 4331).
- Sejak mulai maraknya penggunaan PC pada awal tahun 80-an, selanjutnya semakin banyak perusahaan dan personal di Indonesia yang menggunakan komputer dan jumlahnya menjadi tidak terhitung.

2.4. Sekilas Organisasi Komputer.

Empat operasi dasar :

- Menerima input dari dunia luar.
- Memproses informasi: melakukan operasi aritmatik dan logik terhadap informasi.
- Menghasilkan output: memberikan informasi ke dunia luar.
- Menyimpan dan mengambil informasi dari memori dan tempat penyimpanan (storage).

### Komponen Perangkat Keras Komputer:

- Input devices (piranti masukan).
- Output devices (piranti Keluaran).
- Microprocessor (CPU).
- Memory dan storage device (piranti penyimpanan).
  - Primary storage.
  - Secondary storage.
- Peripherals.

### 2.5. Taksonomi Komputer.

#### Taksonomi Ringkas Komputer:

- Embedded System



Gambar 2.5. Embedded Systems.

- Personal Computer



Gambar 2.6. Personal Computer, Workstation dan Portable computer.

- Workstation

- Portable Computer

- Handheld Device



Gambar 2.7. Smartphone.

- Server



Gambar 2.8. Server.

## - Mainframe dan Supercomputer



Gambar 2.9. Mainframe dan Supercomputer.

### 2.6. Jaringan Komputer dan Internet.

### 2.7. Implikasi Teknologi Digital.

### 2.8. Daftar Pustaka.

- [1] <https://www.youtube.com/> Rolls-Royce, Rolls-Royce | How we assemble the Trent XWB; the world's most efficient aero engine; akses 7/12/2020.
- [2] <https://www.youtube.com/> Animagraffs, How Jet Engines Work; akses 7/12/2020.
- [3] <https://www.youtube.com/> Maestro, Flexible ureteroscopy and laser lithotripsy in ureter of an obstructing stone; akses 7/12/2020
- [4] George Beekman & Ben Beekman, (2012), "Digital Planet: Tomorrow's Technology and You", 10th edition, Pearson.
- [5] <http://www.teknologiinformasidankomunikasi.com/ict/sejarahpengguna-ict-indonesia/>; akses 24 Agustus 2012.
- [6] Stalling, William, (2003), "Organisasi dan Arsitektur Komputer: Edisi Bahasa Indonesia Jilid 1", Penerbit Indeks.

### 2.9. Pertanyaan Quis.

1. Jelaskan jaringan komputer merupakan satu komputer yang virtual.
2. Jelaskan yang anda ketahui tentang organisasi komputer.
3. Bagaimana pendapat anda kalau pemerintah Indonesia tidak mengizinkan lewat provider internet [www.youtube.com](http://www.youtube.com), [www.google.com](http://www.google.com), WA, dan Twitter diakses oleh user diwilayah Indonesia? Berikan penjelasan argumen anda?
4. Apakah anda setuju dan jelaskan bila pemerintah mewajibkan menggunakan perangkat lunak buatan Indonesia. Contohnya [www.google.com](http://www.google.com) versi perangkat lunak buatan Indonesia yang mirip google? Atau [www.youtube.com](http://www.youtube.com) versi perangkat lunak buatan Indonesia yang mirip youtube?
5. Uraikan jaringan komputer dan internet ?
6. Uraikan implikasi teknologi digital ?

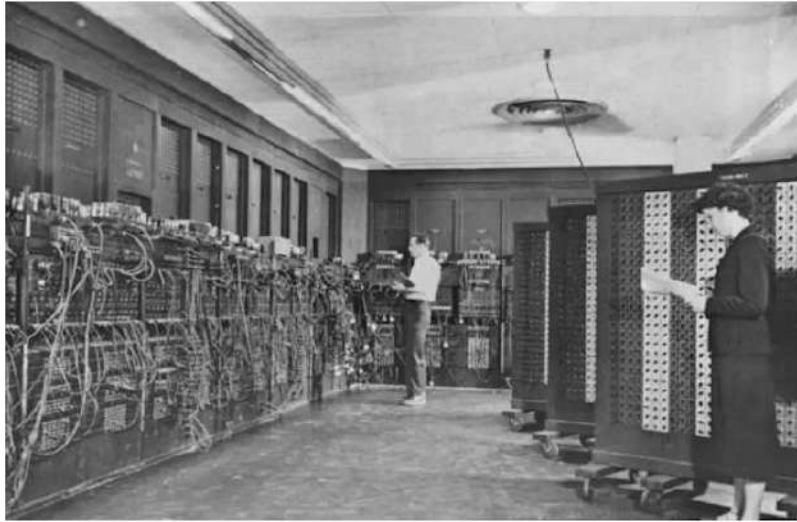
### Bab 3

## Perangkat Lunak Komputer dalam Bahasa Tingkat Tinggi (High Level Language)

Tujuan: Sub-CP-MK 3 Mampu mendefinisikan, menganalisis prinsip dan cara kerja perangkat lunak komputer.

Waktu: Dua tatap muka.

### 3.1. Pengantar Berpikir Komputasi dan Pemrograman Prosedural.



Bits → Assembly → HLL → OOP → Framework

Gambar 3.1. Perkembangan Perangkat Lunak Mulai Bits Sampai Pemrograman Dengan Framework.

Pada perkembangan perangkat lunak, Sistem operasi belum ditemukan sehingga operator menjadi bagian dari perangkat lunak yang memindahkan barisan bits ke pita input lalu dieksekusi oleh CPU menghasilkan pita output dalam bentuk barisan bits dimana CPU (Central Processor Unit) dijalankan mulai dari program dijalankan (start) sampai diberhentikan (stop). Setelah pertengahan 1960an, komputer mulai menggunakan integrated circuit yaitu silicon chip yang mengandung ratusan transistor. Perangkat lunak menggunakan pita input barisan bits tetap dibantu operator karena sudah difilekan ke pita yang berisi barisan bits untuk dieksekusi CPU komputer hanya mengerjakan satu pekerjaan yaitu pita output yang berisi barisan bits untuk dibaca di unit output untuk satu pemrograman.

Pemrograman bahasa assembly dibuat saat era konsep store-program pada tahun 1946 yang dikenal dengan komputer Neumann yaitu memproses data input barisan bits kedalam register input diproses dengan aritmatik logik unit berdasarkan program kontrol unit sebagai register yang mencatat program dimulai line code awal (start) sampai line code akhir (stop). Output barisan bits dari hasil proses ALU disimpan kedalam register output. Setiap waktu eksekusi nilai barisan bits disimpan di memori utama untuk diproses oleh output unit seperti monitor atau printer. Sampai sekarang konsep store-program atau komputer Neumann digunakan dengan kecepatan CPU dalam GHz dan main memori dalam Gbyte. Jadi hampir 75 tahun, prinsip kerja komputer Neumann digunakan.

Pada sekitar 1972 di Bell laboratorium, Dua programmer membuat operating sistem menggunakan HLL (High Level Language) yaitu bahasa C. Mereka berdualah yang membuat bahasa C agar lebih gampang untuk membuat sistem operasi. Gunanya sistem operasi yang bernama UNIX adalah komputer bisa

mengerjakan pekerjaan lebih dari satu atau multiprogramming. Bahasa Tingkat Tinggi atau High Level Language (HLL) C diterjemahkan ke bahasa assembly sebagai pemroses register-register di CPU. Jadi kalau sistem operasi UNIX ditulis dalam bahasa assembly akan terlalu kompleks dan tidak bisa dimengerti dalam pemahaman bahasa manusia karena pemrograman bahasa assembly merupakan bahasa mesin atau bahasa biner atau barisan bits. Bahasa assembly dibuat untuk mengolah barisan bits sebagai input dieksekusi oleh ALU menghasilkan barisan bits output.

Karena kecepatan komputer bertambah cepat dan memori sampai gigabyte, pemrograman perangkat lunak berkembang ke fase OOP yaitu Objek Orientasi Pemrograman lalu ke fase framework pemrograman. Agar dengan framework pemrograman atau dengan OOP dapat diselesaikan dalam ukuran dibawah satu bulan untuk pekerjaan pembuatan aplikasi perangkat lunak suatu perusahaan.

Saat ini karena untuk pembelajaran, Fase bahasa tingkat tinggi (High Level Language) dipelajari. Pemrograman bahasa C/C++ cocok untuk menjadi dasar dan start awal ke fase yang lebih kompleks yaitu OOP dan Framework pemrograman.

Pseudocode adalah pendefinisi algoritma dari program komputer kedalam bahasa manusia yang lebih mudah dipahami.

Pseudocode biasa digunakan untuk perencanaan sebelum melakukan pemrograman.

Pseudocode memang bertujuan untuk membuat para pembaca (manusia termasuk non-programmer, bukan untuk mesin) mudah mengerti inti dari algoritma yang dituliskan.

Oleh karena itu, pseudocode menggunakan bahasa alami manusia (High Level Language –HLL).

Walaupun begitu, pseudocode untuk pemrograman ini memiliki ketentuannya masing-masing. Contohnya: klausa kondisional (if, switch, dll) harus menggunakan huruf kapital diawalnya.

Contoh Pseudocode Program Kecil

Program OPER

{ Contoh program operasi pengalian 2 berdasarkan input pengguna }

Kamus

I : integer  
X : real  
flag : boolean

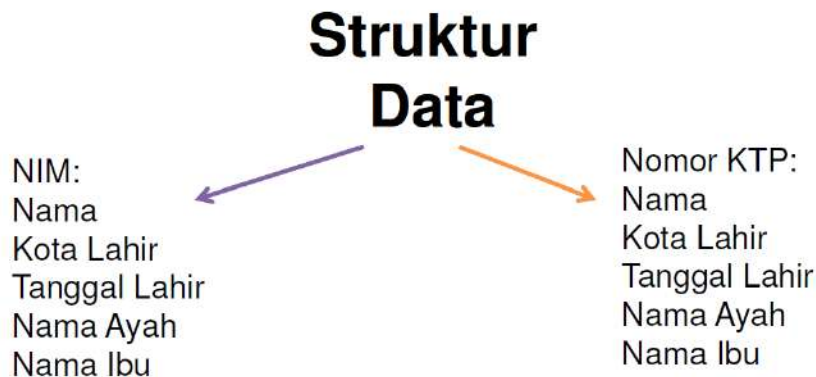
Algoritma

Input (I)  
X ← 1.0  
flag ← true  
output (X)  
output (I\*2, flag)

Karakteristik Berpikir Komputasi.

- Mampu memberikan pemecahan masalah menggunakan komputer atau perangkat lain.
- Mampu mengorganisasi dan menganalisa data.
- Mampu melakukan representasi data melalui abstraksi dengan suatu model atau simulasi.
- Mampu melakukan otomatisasi solusi melalui cara berpikir algoritma (sekumpulan langkah terurut).
- Mampu melakukan identifikasi, analisa dan implementasi solusi dengan berbagai kombinasi langkah/cara dan sumberdaya yang efisien dan efektif.

- Mampu melakukan generalisasi solusi untuk berbagai masalah berbeda. Data Mahasiswa dengan Data Penduduk.



Gambar 3.2. Maksud dari Struktur Data.

Abstraksi data dengan Model dan Simulasi.

- Mampu melakukan otomatisasi solusi melalui cara berpikir algoritma (sekumpulan langkah terurut).
- Mampu melakukan identifikasi, analisa dan implementasi solusi dengan berbagai kombinasi langkah/cara dan sumberdaya yang efisien dan efektif.
- Mampu melakukan generalisasi solusi untuk berbagai masalah berbeda.

Kemampuan minimum yang diharapkan.

- Kemampuan melakukan dekomposisi masalah.
  - Menganalisa resep suatu masakan.
- Kemampuan mengenali pola.
  - Misalnya pola harga barang yang naik kalau setiap lebaran.
- Kemampuan menggeneralisir pola dan mengabstraksi pola.
  - Abstraksi kalender kerja yang merepresentasikan apa yang dikerjakan dan kapan.
- Kemampuan perancangan program.
  - Instruksi untuk memasak dengan suatu resep.
  - Instruksi membuka paket lemari knock down.

Keuntungan berpikir komputasi.

- Percaya diri dalam berhadapan dengan kompleksitas masalah yang lebih besar dan lebih sulit.
- Toleransi terhadap ambiguitas.
- Mampu berhubungan dengan masalah yang openended.
- Mampu berkomunikasi dan bekerjasama untuk mencapai tujuan yang sama.

Berpikir Prosedural.

- Sejumlah aksi dijalankan secara berurutan (sekuensial).
- Setiap aksi akan memberikan efek eksekusi tertentu.
- Jika diikuti terus menerus, aksi-aksi ini harus selesai.
  - Tidak bisa terus menerus.

### 3.2. Bagaimana Program itu dibuat ? [3]

Program = Algoritma + Data Struktur.

Program Menghitung Tabungan.

- Masalah:
  - Tabungan di bank selalu bertambah setiap tahun.





## Kode Program Bahasa C++

```
input (NilaiRp)  
NilaiRp ← NilaiRp + NilaiRp * 10%  
output(NilaiRp)
```

*cin: Console Input* (diketikkan lewat keyboard) → `cin >> NilaiRp;`  
`NilaiRp = NilaiRp + NilaiRp * 0.1;`  
`cout << NilaiRp;`

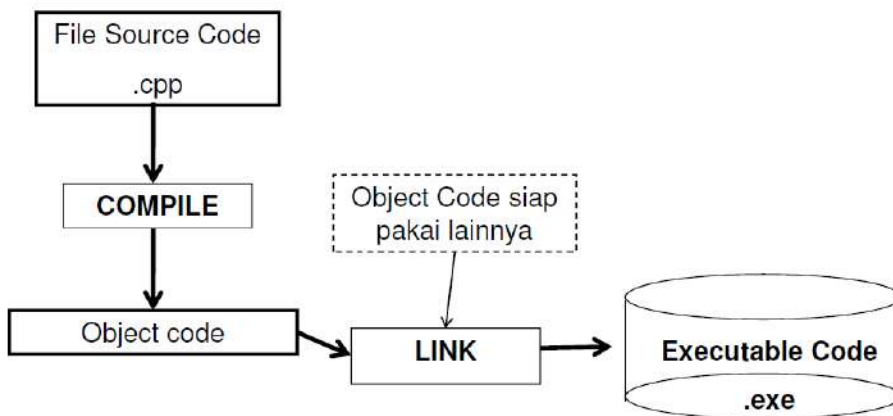
*cout: Console Output* →  
Gambar 3.4. Dituangkan ke kode program bahasa C++.

## Kode C++

```
int main()  
{  
    int NilaiRp; Pendefinisian Data  
  
    cin >> NilaiRp;  
    NilaiRp = NilaiRp + NilaiRp * 0.1;  
    cout << NilaiRp;  
    return 0;  
}
```

Gambar 3.5. Disimpan dalam file.cpp.

Edit, Kompilasi (Compile) dan Eksekusi.



Gambar 3.6. File.cpp dikompilasi menjadi File.obj lalu dikonversikan ke barisan bits menjadi File.exe.

Karakteristik berpikir komputasi, pemrograman prosedural, dan bagaimana program itu dibuat. Pengetahuan ini diambil dari daftar pustaka [3].

### 3.3. Daftar Pustaka.

[1] Tanenbaum, Andre, (2009), "Modern Operating System", Pearson, ISBN Q-1B-fi1BMST-L

[2] Stalling, William, (2003), "Organisasi dan Arsitektur Komputer: Edisi Bahasa Indonesia Jilid 1", Penerbit Indeks.

[3] Munir, Rinaldi, (2016), "ALGORITMA DAN PEMROGRAMAN DALAM BAHASA PASCAL, C, DAN C++" EDISI KEENAM, Informatika.

### 3.4. Pertanyaan Quis.

1. Coba kamu tulis program gambar 3.5 di aplikasi program C++.
2. Apakah File.cpp bila dikompilasi menjadi File.obj dan dikonversikan ke barisan bits menjadi File.exe ?

Bagian Kedua “Jaringan Komputer Dianggap Satu Komputer Secara Virtual”

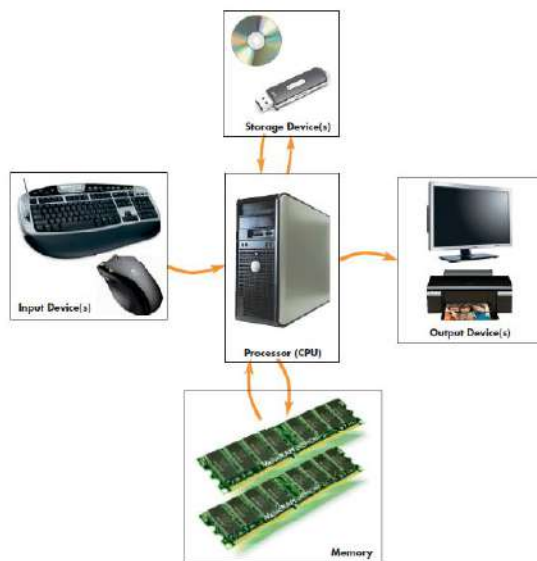
*CPMK II Mampu memahami fungsi dan proses kerja jaringan komputer serta cara kerja aplikasi perangkat lunak desktop dan web based. (S8, S9, S11, KU1, KU2, KK1, P1, P2, P3)*

## Bab 4 Organisasi Dan Arsitektur Komputer

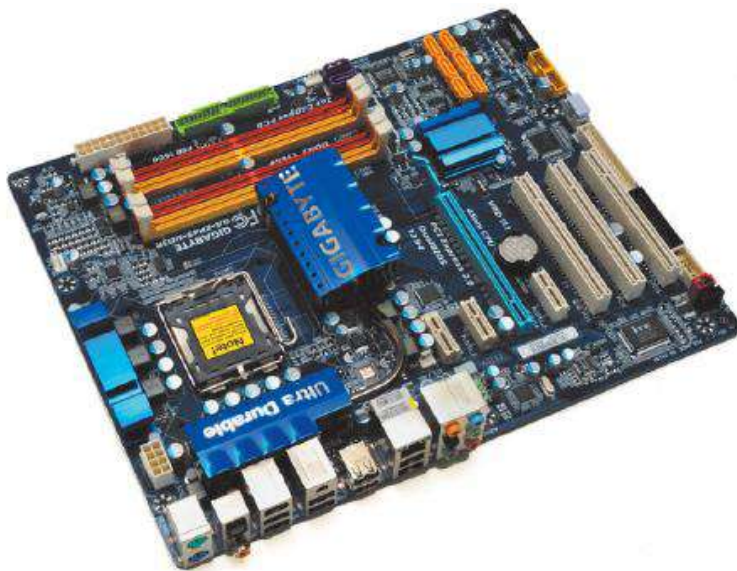
Tujuan: Sub-CP-MK 4 Mampu memahami fungsi dan proses kerja organisasi dan arsitektur komputer.

Waktu: Satu tatap muka.

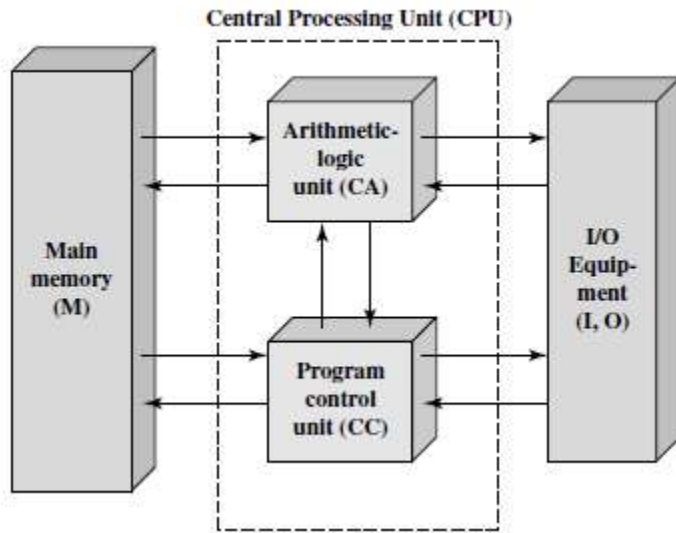
### 4.1. Sistem Komputer.



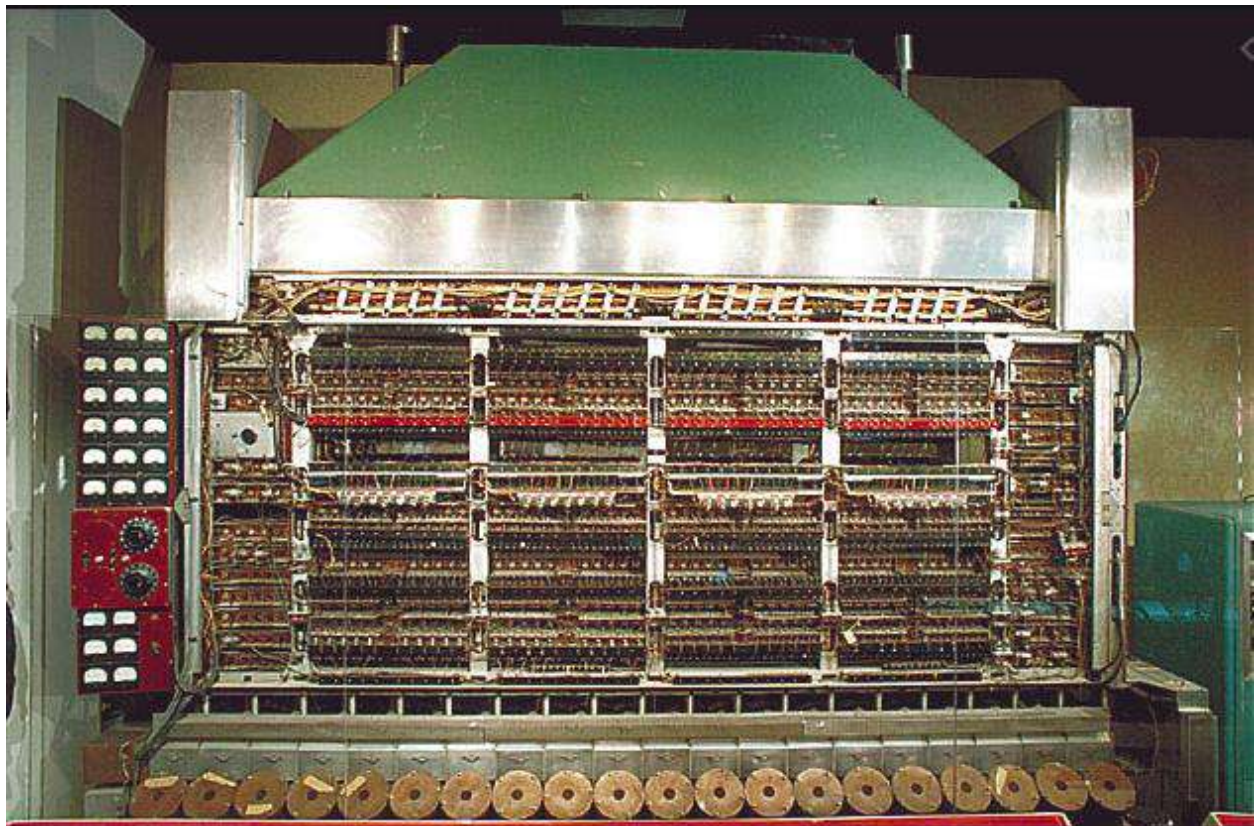
Gambar 4.1. Empat komponen sistem komputer.



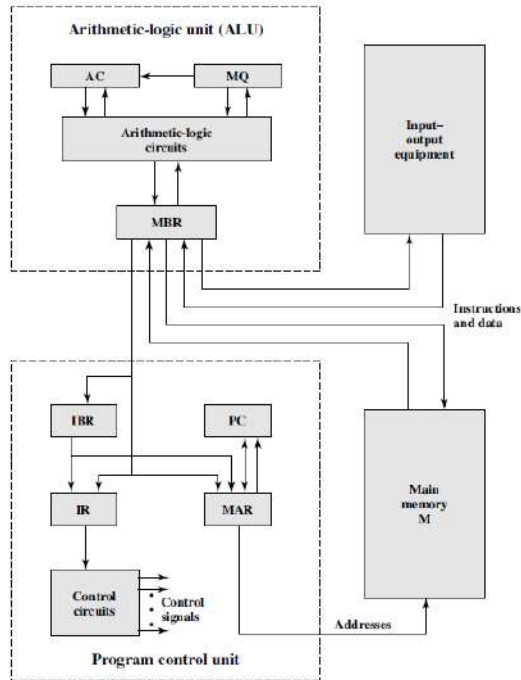
Gambar 4.2. Motherboard komputer yang menghubungkan CPU, Memori dan I/O sehingga menjadi empat komponen dari sistem komputer.



Gambar 4.3. Struktur komputer konsep store-program atau komputer IAS (Institute for Advanced Study).



Gambar 4.4. Komputer IAS atau The Von Neumann Machine. [3]



Gambar 4.5. Pengembangan Struktur Komputer IAS.

Instruction Type	Opcode	Symbolic Representation	Description
Data transfer	00001010	LOAD MQ	Transfer contents of register MQ to the accumulator AC
	00001001	LOAD MQ,M(X)	Transfer contents of memory location X to MQ
	00100001	STOR M(X)	Transfer contents of accumulator to memory location X
	00000001	LOAD M(X)	Transfer M(X) to the accumulator
	00000010	LOAD -M(X)	Transfer -M(X) to the accumulator
	00000011	LOAD  M(X)	Transfer absolute value of M(X) to the accumulator
	00000100	LOAD - M(X)	Transfer - M(X)  to the accumulator
Unconditional branch	00001101	JUMP M(X,0:19)	Take next instruction from left half of M(X)
	00001110	JUMP M(X,20:39)	Take next instruction from right half of M(X)
Conditional branch	00001111	JUMP+ M(X,0:19)	If number in the accumulator is nonnegative, take next instruction from left half of M(X)
	00010000	JUMP+ M(X,20:39)	If number in the accumulator is nonnegative, take next instruction from right half of M(X)
Arithmetic	00000101	ADD M(X)	Add M(X) to AC; put the result in AC
	00000111	ADD  M(X)	Add  M(X)  to AC; put the result in AC
	00000110	SUB M(X)	Subtract M(X) from AC; put the result in AC
	00001000	SUB  M(X)	Subtract  M(X)  from AC; put the remainder in AC
	00001011	MUL M(X)	Multiply M(X) by MQ; put most significant bits of result in AC, put least significant bits in MQ
	00001100	DIV M(X)	Divide AC by M(X); put the quotient in MQ and the remainder in AC
	00010100	LSH	Multiply accumulator by 2; i.e., shift left one bit position
00010101	RSH	Divide accumulator by 2; i.e., shift right one position	
Address modify	00010010	STOR M(X,8:19)	Replace left address field at M(X) by 12 rightmost bits of AC
	00010011	STOR M(X,28:39)	Replace right address field at M(X) by 12 rightmost bits of AC

Gambar 4.6. Set Instruksi IAS.



Berdasarkan gambar 4.8. (a) yaitu format instruksi (instruction format), program bahasa assembly dikodekan kedalam barisan bits dalam program hexadecimal. Berdasarkan gambar 4.9., Ada tiga line code yang mempunyai nomer barisan hexadecimal dimulai dengan nomer 300 sebagai start dan diakhiri dengan nomer 302 sebagai stop. Program hexadecimal terdiri dari tiga line yaitu sebagai berikut:

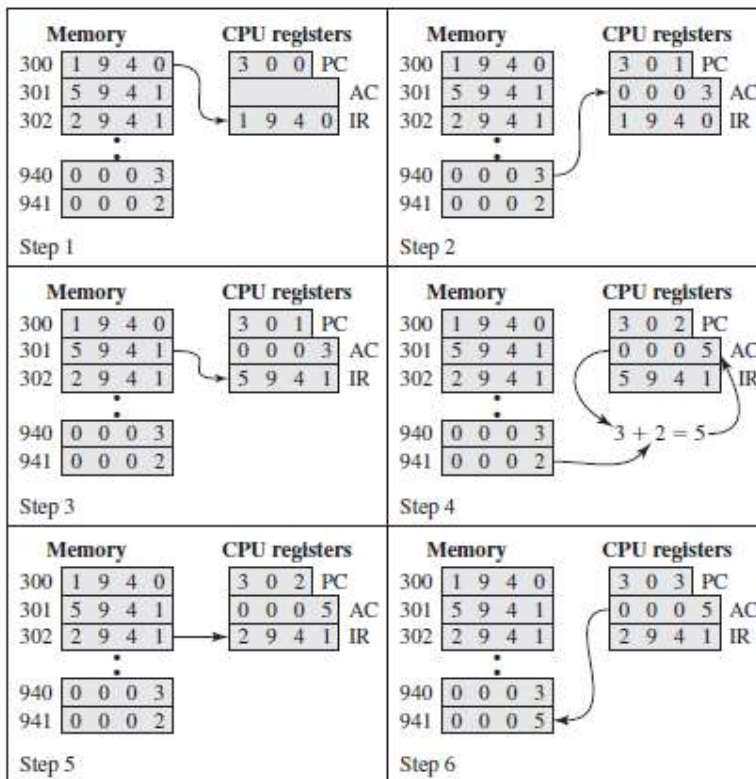
1940 yang berarti dalam bahasa assembly adalah load AC dari data di alamat memori 940.

5941 yang berarti dalam bahasa assembly adalah add ke AC dari data di alamat memori 941.

2941 yang berarti dalam bahasa assembly adalah store AC ke memori di alamat memori 941.

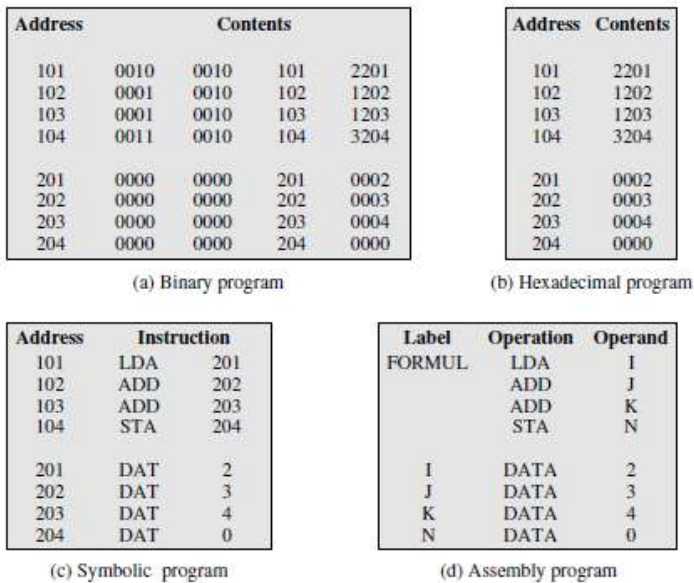
Catatan konversi hexadecimal ke biner (barisan bits berderet 4 bit) yaitu:

Hexadecimal 0	biner 0000
Hexadecimal 1	biner 0001
Hexadecimal 2	biner 0010
Hexadecimal 3	biner 0011
Hexadecimal 4	biner 0100
Hexadecimal 5	biner 0101
Hexadecimal 6	biner 0110
Hexadecimal 7	biner 0111
Hexadecimal 8	biner 1000
Hexadecimal 9	biner 1001
Hexadecimal A	biner 1010
Hexadecimal B	biner 1011
Hexadecimal C	biner 1100
Hexadecimal D	biner 1101
Hexadecimal E	biner 1110
Hexadecimal F	biner 1111



Gambar 4.9. Contoh Program Eksekusi Hipotesis Machine (konten memori dan register dalam hexadecimal).





Gambar 4.10. Komputasi Dari Formula  $N = I + J + K$ .

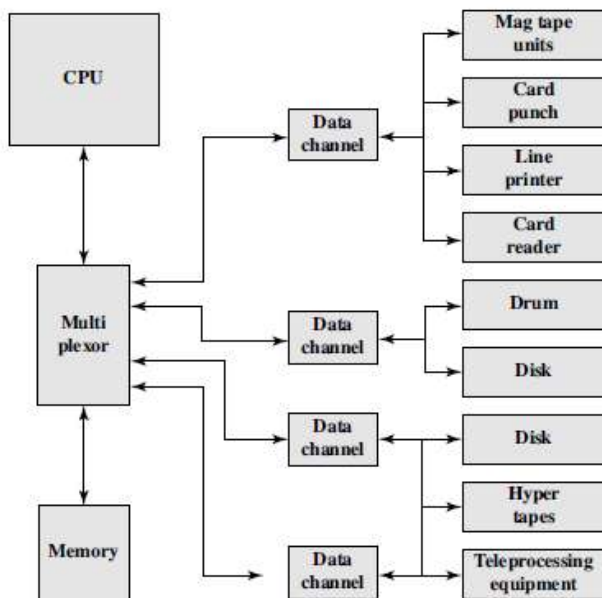
Program C++ dikodekan dari program bahasa assembly untuk  $N = I + J + K$  yang ditulis sebagai berikut.

```

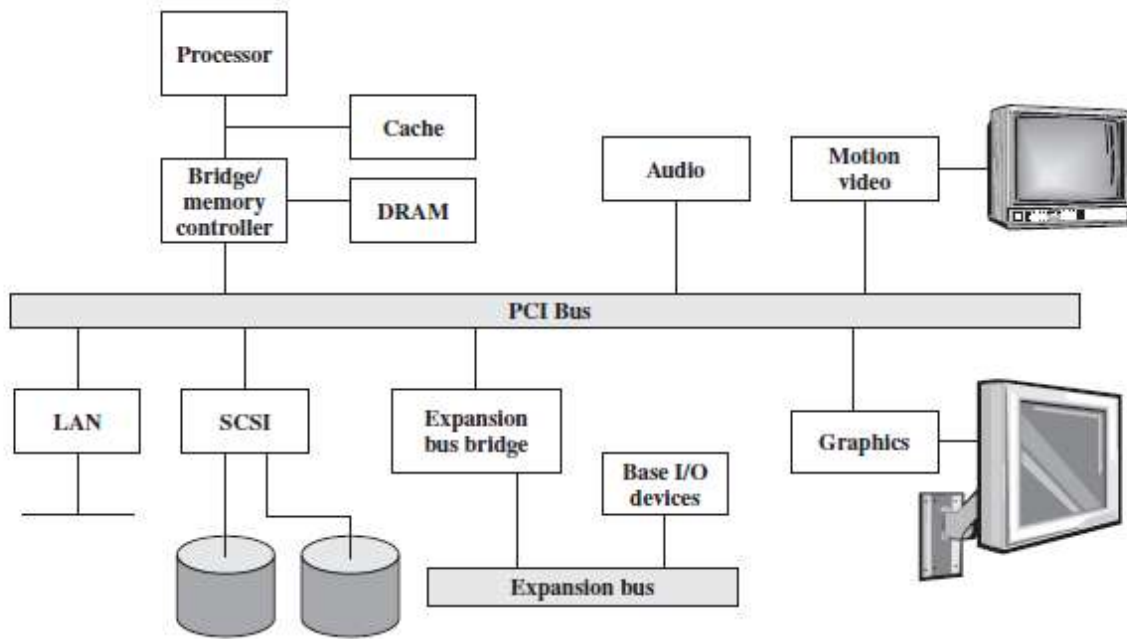
Void main ()
{
    int I = 2;
    int J = 3;
    int K = 4;
    int N = 0;
    N = I + J + K;
}

```

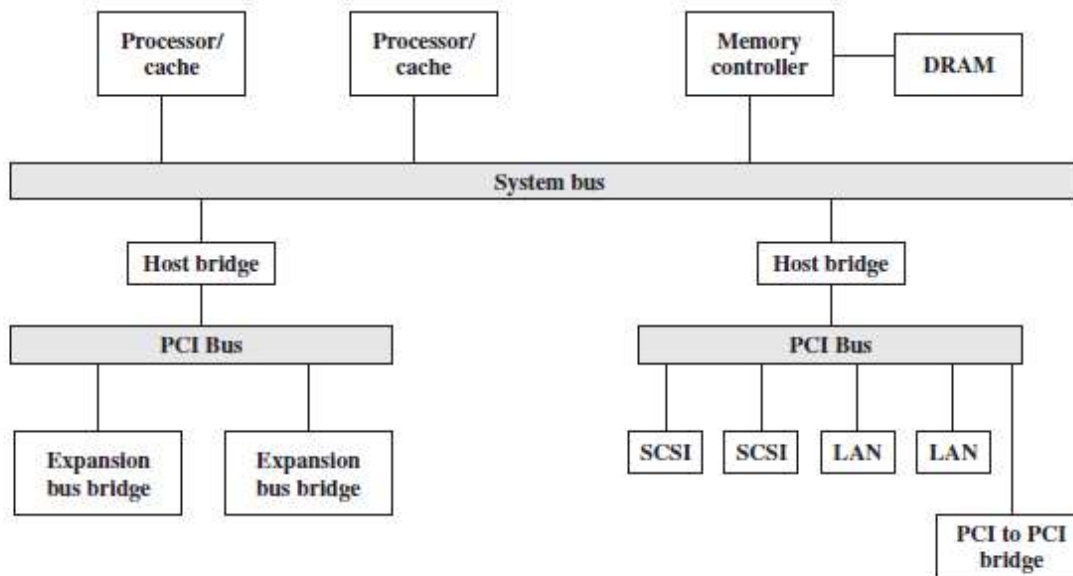
#### 4.2. Perangkat Keras Komputer.



Gambar 4.11. Konfigurasi IBM 7094.



(a) Typical desktop system



(b) Typical server system

Gambar 4.12. Arsitektur Komputer Dekstop System Dan Server System

#### 4.3. Daftar Pustaka.

- [1] George Beekman & Ben Beekman, (2012), "Digital Planet: Tomorrow's Technology and You", 10th edition, Pearson.
- [2] Stalling, William, (2003), "Organisasi dan Arsitektur Komputer: Edisi Bahasa Indonesia Jilid 1", Penerbit Indeks.
- [3] [https://en.wikipedia.org/wiki/IAS\\_machine](https://en.wikipedia.org/wiki/IAS_machine); akses 7/12/2020.

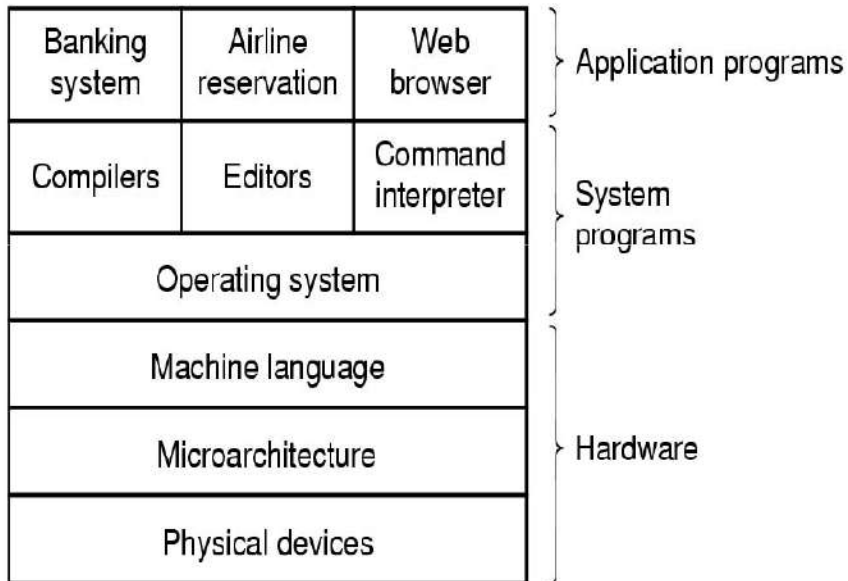
#### 4.4. Pertanyaan Quis

1. Berikan kesimpulan dari bab 4.
2. Mengapa arsitektur komputer IAS (komputer klasik) selama 75 tahun masih dipakai ? Bagaimana dengan arsitek komputer quantum (komputer masa depan)?

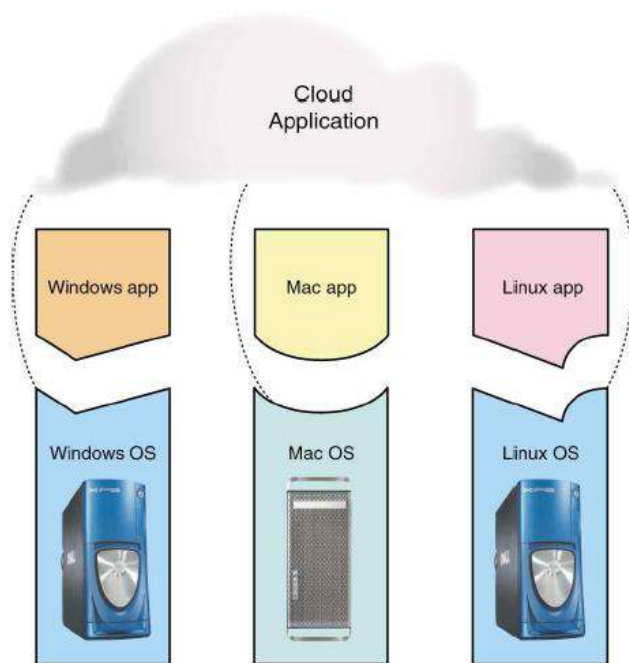
## Bab 5 Jaringan Komputer

Tujuan: Sub-CP-MK 5 Mampu memahami fungsi dan proses kerja jaringan komputer serta cara kerja aplikasi perangkat lunak desktop dan web based  
Waktu: Satu tatap muka.

### 5.1. Perangkat Lunak Komputer.



Gambar 5.1. Kategori Perangkat Lunak. [3]

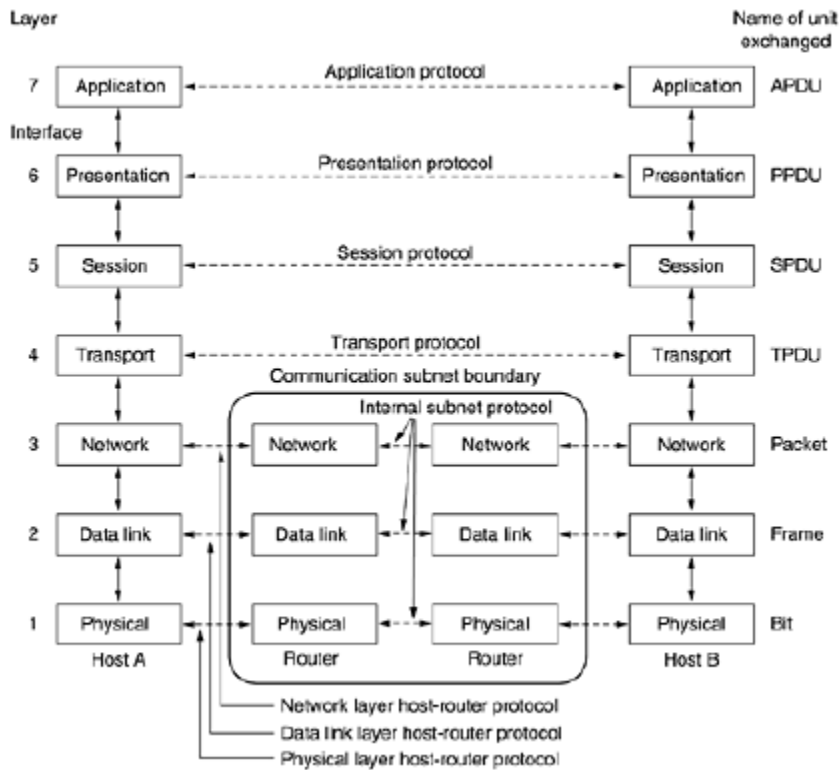


Gambar 5.2. Kompatibilitas Aplikasi Cloud Didesain Untuk Bisa Dijalankan Di Platform Berbeda-beda. [2]

Dalam perangkat lunak sistem, Sistem operasi didesain untuk beroperasi pada perangkat keras tertentu. Contoh untuk perangkat keras arsitektur CPU Mac, perangkat lunak sistem yang dibutuhkan adalah sistem operasi Mac. Contoh lainnya untuk arsitektur CPU Intel Core i7-7700 CPU @ 3.60 GHz, 4 Corw, 8 Logical Processor, perangkat lunak sistem yang dibutuhkan adalah sistem operasi Microsoft Windows 10 Enterprise 2016 LTSB dengan tipe sistem x64-based PC. Atau dengan arsitektur CPU Intel Core i7-7700 CPU @ 3.60 GHz, 4 Corw, 8 Logical Processor, perangkat lunak sistem yang dibutuhkan adalah sistem operasi linux dengan tipe sistem x64-based PC. Hal ini seperti yang digambarkan pada gambar 5.2.

Aplikasi didesain untuk berjalan di atas sistem operasi tertentu. Contohnya untuk aplikasi windows tidak dapat diinstalasi di sistem operasi linux atau sistem operasi Mac. Aplikasi windows hanya bisa diinstalasi di sistem operasi windows. Begitu juga aplikasi linux hanya bisa diinstalasi di sistem operasi linux dengan tipe sistem x64-based PC. Tipe sistem arsitektur CPU dipasaran saat ini adalah tipe sistem 32 bit atau 64 bit. Sistem operasi linux juga saat ini mempunya dua tipe yaitu sistem operasi 32 bit atau 64 bit. Sama seperti sistem operasi windows juga saat ini mempunya dua tipe yaitu sistem operasi 32 bit atau 64 bit. Sistem operasi 32 bit hanya bisa diinstalasi di CPU tipe sistem 32 bit atau sistem operasi 64 bit hanya bisa diinstalasi pada CPU tipe sistem 64 bit.

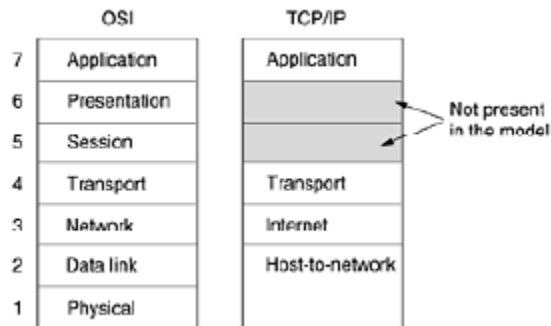
## 5.2. Jaringan Komputer.



Gambar 5.3. Model OSI (Open Systems Interconnection) Untuk Referensi Jaringan Komputer. [1]

Ada dua model untuk jaringan komputer dipakai sehingga jaringan komputer yang ada didunia saling menyambung dari node komputer sumber ke node komputer tujuan. Model jaringan komputer berdasarkan ISO (Organisasi Internasional untuk Standardisasi) adalah model OSI. Sedangkan, model jaringan komputer berdasarkan komersial atau produk eksis di pasaran adalah model TCP/IP. Memungkinkan aplikasi cloud didesain untuk bisa dijalankan di platform berbeda-beda dikarenakan jaringan komputer. Akibatnya jaringan komputer yang berada didunia dikoneksi oleh internet seolah-olah

menjadi satu komputer raksasa yang bisa diakses melalui akses internet oleh milyaran user komputer dan smarphone.



Gambar 5.4. Model TCP/IP Untuk Referensi Jaringan Komputer. [1]

### 5.3. Daftar Pustaka.

[1] Tannenbaum, Andrew, (1996), "Jaringan Komputer: Edisi Bahasa Indonesia Jilid 1", Prentice Hall.

[2] George Beekman & Ben Beekman, (2012), "Digital Planet: Tomorrow's Technology and You", 10th edition, Pearson.

[3] Munir, Rinaldi, (2016), "ALGORITMA DAN PEMROGRAMAN DALAM BAHASA PASCAL, C, DAN C++" EDISI KEENAM, Informatika.

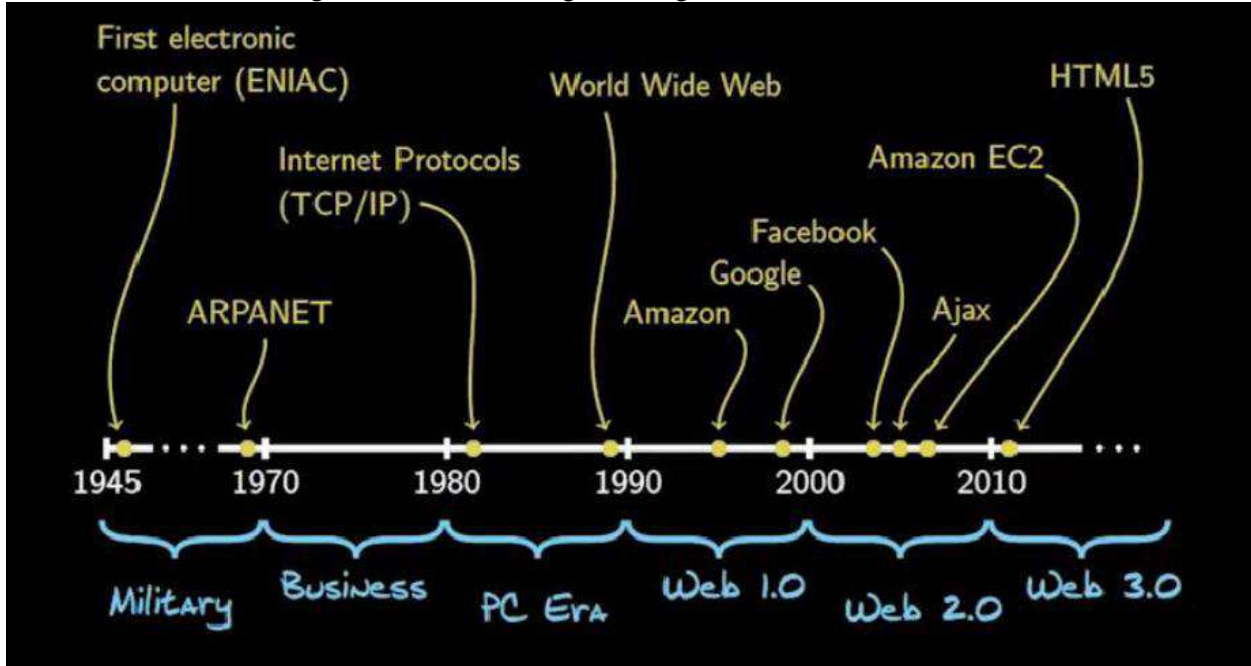
### 5.4. Pertanyaan Quis.

1. Beri kesimpulan dari bab 5.
2. Apa bedanya perangkat lunak sistem dengan aplikasi perangkat lunak ?

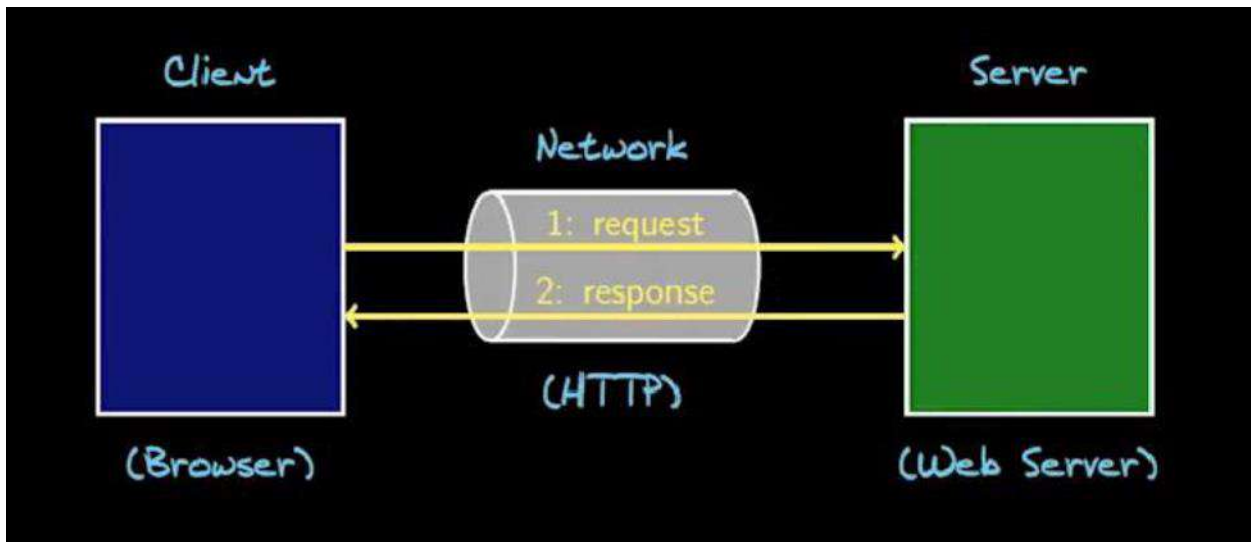
Bab 6  
Pemanfaatan Teknologi Informasi Di Berbagai Bidang

Tujuan: Sub-CP-MK 6 Mampu memahami cara kerja aplikasi perangkat lunak desktop dan web based  
Waktu: Satu tatap muka.

6.1. Pemanfaatan Teknologi Informasi di Berbagai Bidang.

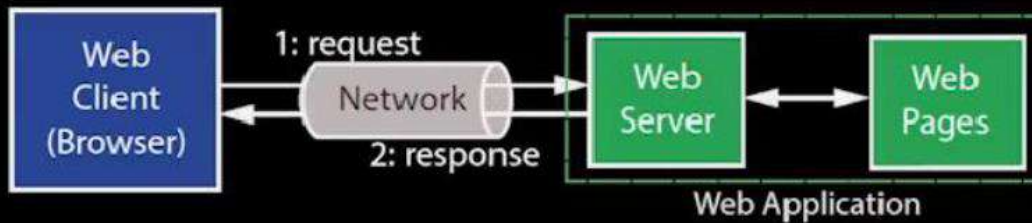


Gambar 6.1. Sejarah perkembangan pemrograman internet berdasarkan garis waktu.



Gambar 6.2. Model Client-Server.

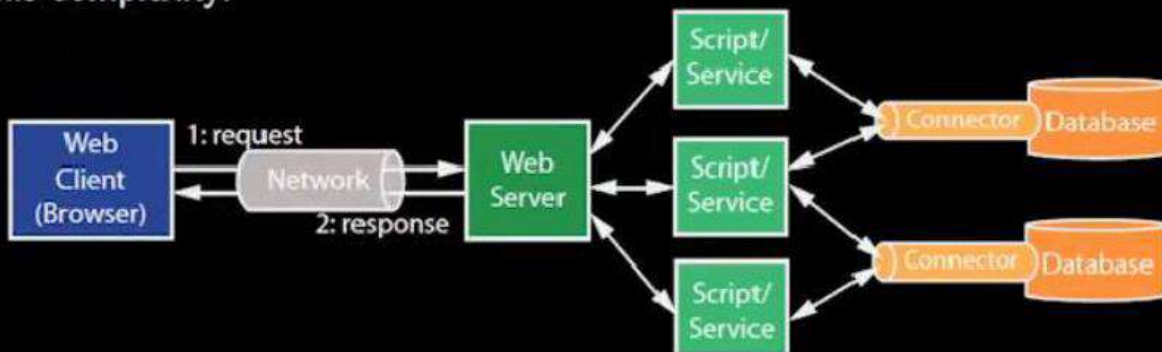
A Web 1.0 application architecture is not much more complicated than the client-server model we previously showed:



- The web server is primarily fetching static web pages – not much interactivity.
- No separation of data from its presentation.
- The browser is very simple – it only needs to render HTML.

Gambar 6.3. Web 1.0.

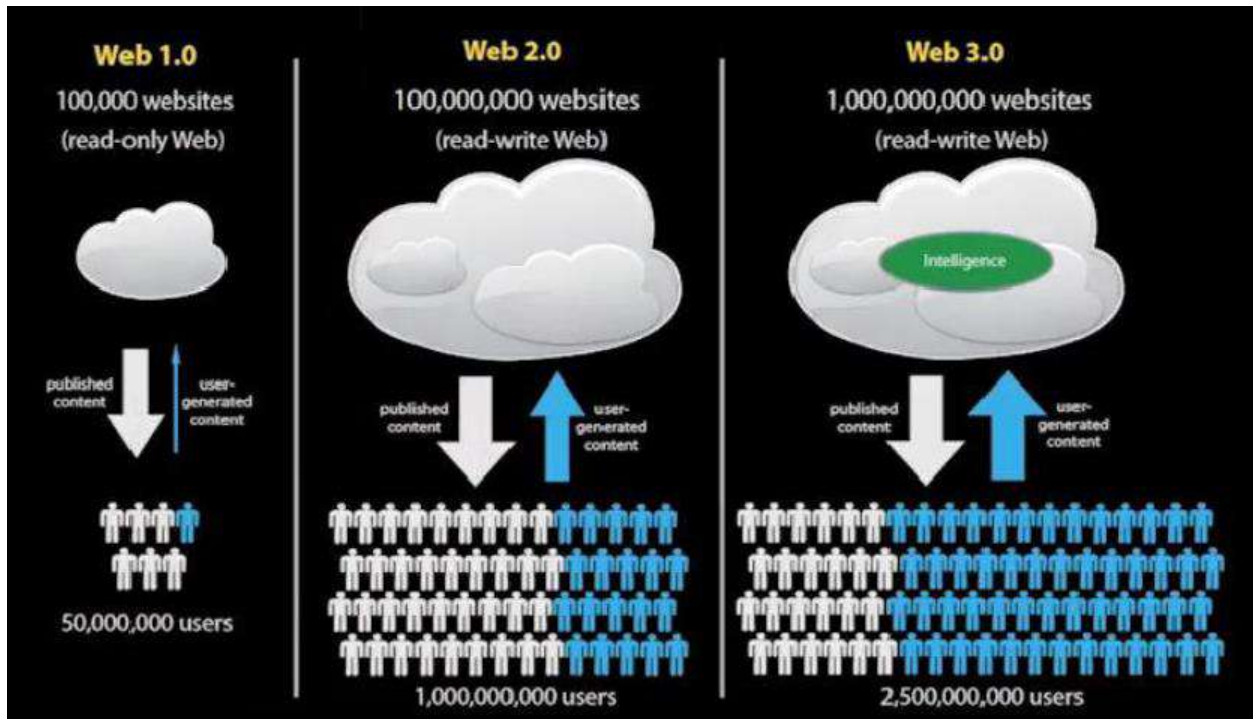
Web 2.0 and 3.0 application architectures are better organized to deal with this complexity:



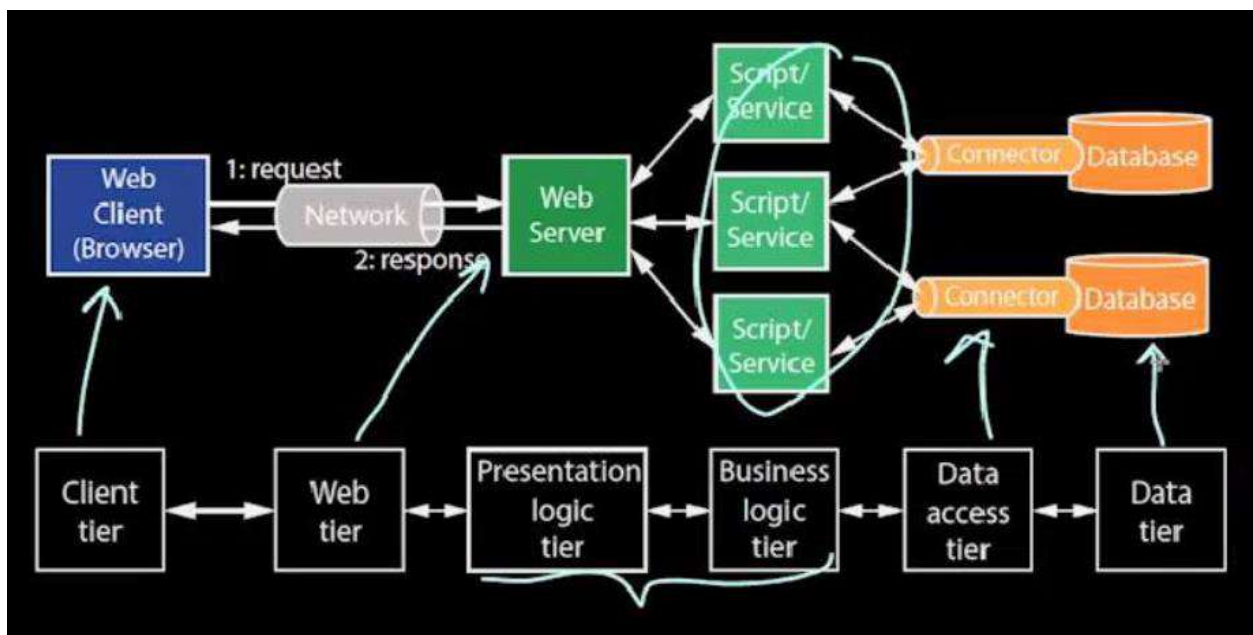
- Server-side functionality is partitioned more intelligently – we'll spend quite a bit of time studying this.
- The browser is more capable, with better standards support.

Gambar 6.4. Aplikasi Web Model Sekarang.





Gambar 6.5. Kemampuan Aplikasi Web Model dari Web 1.0, Web 2.0, Dan Web 3.0.



Gambar 6.6. Arsitektur Aplikasi Web.

Aplikasi web pada gambar 6.4. digunakan dalam E-commerce dan E-business. Disain database dibangun berdasarkan proses bisnis perusahaan atau institusi yang akan dibuat Aplikasi webnya.

6.2. Komputer di Tempat Kerja, di Sekolah dan di Rumah.

6.3. Sistem Informasi, Database, E-commerce dan E-business. Pengembangan dari 6.1. dari sisi perancangan database.

#### 6.4. Aplikasi Produktivitas

Animasi, words Processing, and CAD/CAM.

#### 6.5. Grafik, Media Digital, dan Multimedia.

[www.youtube.com](http://www.youtube.com) dan [www.google.com](http://www.google.com).

Seni melukis dan seni suara dan seni drama.

#### 6.6. Daftar Pustaka.

[1] George Beekman & Ben Beekman, (2012), “Digital Planet: Tomorrow’s Technology and You”, 10th edition, Pearson.

#### 6.7. Pertanyaan Quis.

1. Jelaskan perbandingan dan perbedaan aplikasi perangkat lunak desktop dan web based.

## Bagian Ketiga Pemrograman C++

**CPMK III Mampu mencodingkan kalkulasi matematika kedalam pemrograman bahasa tingkat tinggi seperti bahasa C lalu dikompilasi ke bahasa tingkat rendah atau bahasa mesin atau bahasa biner sehingga bisa dieksekusi yang dapat dilihat dimonitor. (S8, S9, S11, KU1, KU2, KU5, KK1, KK2, P1, P2, P3)**

### Bab 7 Dasar Pemrograman C++

Sub-CP-MK 7 Mampu mencodingkan kalkulasi matematika kedalam pemrograman bahasa tingkat tinggi seperti bahasa C lalu dikompilasi ke bahasa tingkat rendah atau bahasa mesin atau bahasa biner sehingga bisa dieksekusi yang dapat dilihat dimonitor.

Waktu: Dua tatap muka.

#### 7.1. Struktur Dasar Program Prosedural(dalam Bahasa C++). [1]

Apa hasil eksekusinya?

```
#include <iostream>
using namespace std;
int main () {
//KAMUS
int A;
int B;
//ALGORITMA
cin >> A;
cin >> B;
A = A + B;
cout << A << endl;
cout << B << endl;
return 0;
}
```

Apa hasil eksekusinya?

```
#include <iostream>
using namespace std;
int main () {
//KAMUS
string nama;
//ALGORITMA
cout << "Tuliskan namamu: " << endl;
cin >> nama ;
cout << "Namamu adalah : " << nama << endl;
return 0;
}
```

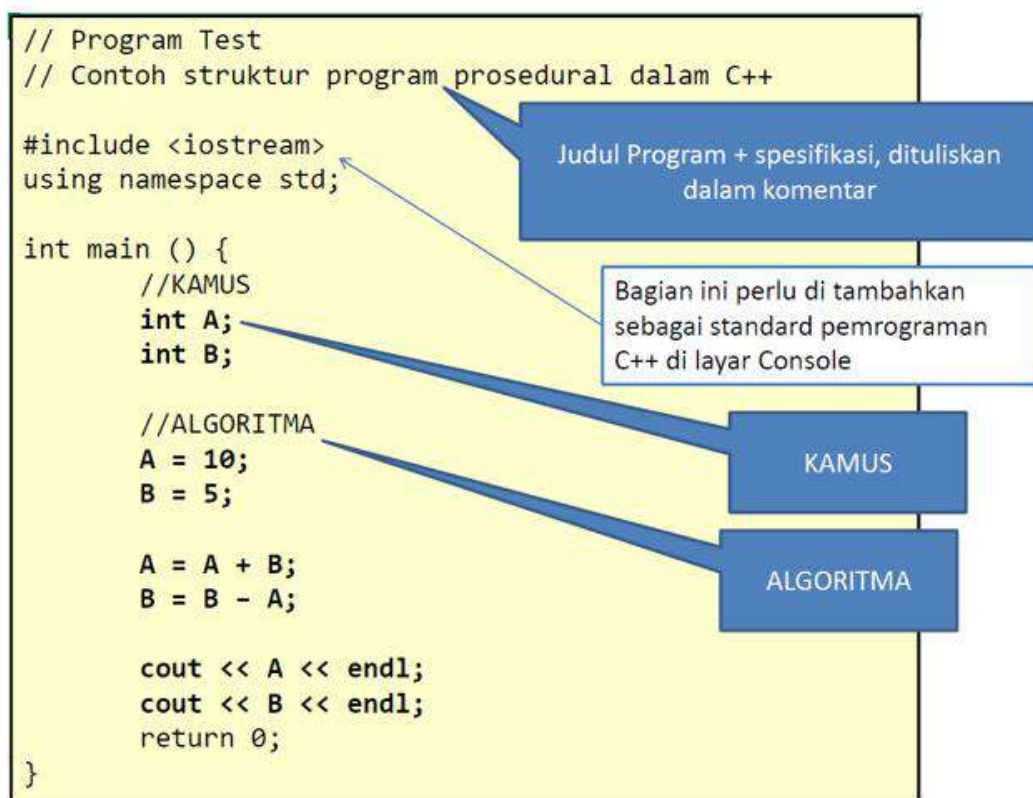
Apa hasil eksekusinya?

```
#include <iostream>
using namespace std;
int main () {
```

```

//KAMUS
int A;
int B;
//ALGORITMA
A = 10;
B = 5;
A = A + B;
B = B - A;
cout << A << endl;
cout << B << endl;
return 0;
}

```



Gambar 7.1. Struktur Dasar Program.

Pada gambar 7.1., `iostream` adalah salah satu header file yang ada di C++. Header ini digunakan untuk fungsi input dan output yang ada di C++. Contoh fungsi input/output: `cin` dan `cout`. `using namespace std` adalah perintah yang digunakan untuk mendeklarasikan/ memberitahukan kepada compiler C++ bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace `std`.

## 7.2. Kamus: Tipe Data, Variabel, Konstanta, Ekspresi. [1]

Kamus dipakai untuk mendeklarasi nama-nama yang digunakan dalam program. Deklarasi nama yang didefinisikan pemrogram yaitu:

- type
- variabel
- konstanta

- Tipe data **primitif** atau tipe **dasar** (dalam C++)
  - Boolean (bool)
  - Integer (int)
  - Real (float)
  - Character (char)
  - String (string)
- Tipe data **turunan** atau **bentukan**
  - Dibentuk dari gabungan tipe dasar
  - Contoh
    - Tipe DataMahasiswa
      - Dibentuk dari
        - » NIM: string
        - » Nama: string
        - » Umur: integer
        - » Kota: string
    - Tipe Array
      - Dibentuk dari kumpulan integer, misalnya 10 data tentang umur

Gambar 7.2. Tipe-Tipe Jenis Data.

#### Contoh Tipe Data

Umur → Integer contoh: 25, 44, 35.  
 Kota → String, contoh: “Jakarta”, “Bandung”  
 Nama → String, contoh: “Budi”, “Ali”  
 Suhu → Integer atau float, contoh: 37.5, 100  
 Luas → Integer atau float, contoh: 400, 43.5  
 BeratBadan → Integer atau float, contoh: 60.5, 75  
 NIM → Integer atau string?, contoh: 15812001

```
// Kamus
typedef struct {
    int x;
    int y;
} Point;
typedef struct {
    string NIM;
    string Nama;
    int Umur;
    string Kota;
} DataMahasiswa;
```

Gambar 7.3. Contoh Deklarasi Tipe Bentuk/Komposit/Struct

Variabel menyimpan nilai ber-”tipe data” sesuai dengan deklarasi.

Variabel :

- deklarasi (supaya nama dikenal),
- inisialisasi nilai (siap dimanipulasi)

Contoh

- Deklarasi variabel

```
int i;  
float A;
```

- Inisialisasi variabel

```
i = 100;
```

- Artinya variabel i di isi dengan nilai 100

```
A = 8.25;
```

- Artinya variabel A diisi dengan nilai real 8.25

Operasi terhadap variabel sangat tergantung dari tipe datanya.

Operasi tipe dasar

- int : \* / + - % < > <= >= == !=
- bool : && || ! !=
- float : \* / + - < > <= >= !=
- char : == !=

Membuat nama variabel yang benar dan “baik” adalah nama variabel harus dimulai dengan huruf dan dapat

diikuti dengan huruf lagi dan angka dan tidak boleh ada tanda baca. Dalam nama variabel tidak boleh dipisahkan oleh spasi. Cari nama variabel yang bisa dimengerti, agar tidak membingungkan. C++ adalah bahasa yang case sensitive. Kesalahan penulisan huruf besar dan kecil menyebabkan error.

Konstanta berbeda dengan Variable, suatu konstanta tidak boleh diubah nilainya.

- Contoh

```
const float PI = 3.1415
```

```
const int nilai = 1000
```

### 7.3. Algoritma. [1]

Buatlah program untuk menghitung jumlah dari dua buah pecahan.

- Spesifikasi program

- Program menerima masukan pecahan pertama berupa pembilang dan penyebut.

- Kemudian program menerima pecahan kedua.

- Lalu program akan melakukan penjumlahan.

- Kemudian menampilkan hasilnya berupa pembilang dan penyebut hasil penjumlahan.

Langkah solusi dilakukan untuk menyelesaikan problem.

- Program menerima masukan pecahan pertama berupa pembilang dan penyebut.

```
cin >> pembilang1;
```

```
cin >> penyebut1;
```

- Kemudian program menerima pecahan kedua.

```
cin >> pembilang2;
```

```
cin >> penyebut2;
```

- Lalu program akan melakukan penjumlahan

```
pembilang3 = pembilang1 * penyebut2 + pembilang2 * penyebut1;
```

```
penyebut3 = penyebut1 * penyebut2;
```

- Kemudian menampilkan hasilnya berupa pembilang dan penyebut hasil penjumlahan

```
cout << pembilang3;
```

```
cout << penyebut3;
```

```

// Program JumlahPecahan
// Menghitung pembilang dan penyebut pecahan dari penjumlahan
// dua buah pecahan yang diketahui pembilang dan penyebutnya
#include <iostream>
using namespace std;
int main()
{ // KAMUS
  int pembilang1, pembilang2, pembilang3;
  int penyebut1, penyebut2, penyebut3;
  // ALGORITMA
  cin >> pembilang1;
  cin >> penyebut1;

  cin >> pembilang2;
  cin >> penyebut2;

  pembilang3 = pembilang1 * penyebut2 + pembilang2 * penyebut1;
  penyebut3 = penyebut2 * penyebut1;

  cout << pembilang3 << endl;
  cout << penyebut3 << endl;
  return 0;
}

```

Gambar 7.4. Coding Dalam C++ Program untuk menghitung jumlah dari dua buah pecahan.

Buatlah program untuk menghitung perkalian dari dua buah pecahan.

• Spesifikasi program

- Program menerima masukan pecahan pertama berupa pembilang dan penyebut.
- Kemudian program menerima pecahan kedua.
- Lalu program akan melakukan perkalian.
- Kemudian menampilkan hasilnya berupa pembilang dan penyebut hasil penjumlahan.

```

// Program KaliPecahan
// Menghitung pembilang dan penyebut pecahan dari perkalian
// dua buah pecahan yang diketahui pembilang dan penyebutnya
#include <iostream>
using namespace std;
int main()
{ // KAMUS
  int pembilang1, pembilang2, pembilang3;
  int penyebut1, penyebut2, penyebut3;

  // ALGORITMA

  cin >> pembilang1;
  cin >> penyebut1;

  cin >> pembilang2;
  cin >> penyebut2;

  pembilang3 = pembilang1 * pembilang2;
  penyebut3 = penyebut1 * penyebut2;

  cout << pembilang3;
  cout << penyebut3;
  return 0;
}

```

Gambar 7.5. Coding dalam C++ Program untuk menghitung perkalian dari dua buah pecahan.

Buatlah program untuk jarak dari kecepatan dan waktu suatu kendaraan.

- Spesifikasi program
  - Program menerima masukan kecepatan.
  - Kemudian program menerima masukan waktu.
  - Lalu program akan melakukan perhitungan.
  - Kemudian menampilkan hasilnya berupa perhitungan jarak.

Analisa Masalah (dekomposisi Masalah)

- Program menerima masukan kecepatan.
- Kemudian program menerima masukan waktu.
- Lalu program akan melakukan perhitungan.
- Kemudian menampilkan hasilnya berupa perhitungan jarak.

Solusi

- Program menerima masukan kecepatan.  
cin >> v;
- Kemudian program menerima masukan waktu.  
cin >> t;
- Lalu program akan melakukan perhitungan.  
Jarak = v \* t;
- Kemudian menampilkan hasilnya berupa perhitungan jarak.  
cout << Jarak << endl;

```
// Program HitungJarak
// Menghitung jarak berdasarkan masukan
// kecepatan (v) dan waktu (t)
#include <iostream>
using namespace std;

int main()
{ // KAMUS
  int jarak, v, t;

  // ALGORITMA

  cin >> v;
  cin >> t;

  jarak = v * t;

  cout << jarak << endl;
  return 0;
}
```

Gambar 7.6. Coding dalam C++ program untuk jarak dari kecepatan dan waktu suatu kendaraan. (Solusi Pertama)



```

// Program HitungJarak
// Menghitung jarak berdasarkan masukan
// kecepatan (v) dan waktu (t)
#include <iostream>
using namespace std;

int main()
{ // KAMUS
  int jarak, v, t;

  // ALGORITMA
  cout << "kecepatan = ";
  cin >> v;
  cout << "waktu = ";
  cin >> t;

  jarak = v * t;

  cout << "Jarak = " << jarak << endl;
  return 0;
}

```

Gambar 7.7. Coding dalam C++ program untuk jarak dari kecepatan dan waktu suatu kendaraan. (Solusi Kedua)

Buatlah program yang memeriksa perbedaan dua buah jam.

- Spesifikasi program
  - Program menerima masukan jam, menit, dan detik yang pertama.
  - Kemudian menerima masukan jam, menit, dan detik yang kedua.
  - Kemudian program menghitung selisih waktu.
  - Selanjutnya menampilkan hasilnya berupa perhitungan selisihnya.

Program yang memeriksa perbedaan dua buah jam.

- Program menerima masukan jam, menit dan detik yang pertama.
 

```
cin >> jam1;
cin >> menit1;
cin >> detik1;
```
- Kemudian menerima masukan jam menit dan detik yang kemudian.
 

```
cin >> jam2;
cin >> menit2;
cin >> detik2;
```
- Kemudian program menghitung selisih waktu.
- Konversi dulu ke detik.
 

```
totaldetik1 = jam1 * 3600 + menit1 * 60 + detik1;
totaldetik2 = jam2 * 3600 + menit2 * 60 + detik2;
```
- Hitung selisihnya.
 

```
detikSelisih = totaldetik2 - totaldetik1;
```
- Hitung untuk jam, menit dan detik
 

```
jamHasil = detikSelisih / 3600;
```

```

menitHasil = (detikSelisih % 3600) / 60;
detikHasil = detikSelisih % 60;

```

- Kemudian Cetak hasil.

– Contoh hasil:

Selisih = 5 jam 3 menit 4 detik.

– Jadi:

```

cout << "Selisih = " << jamHasil << " jam" << menitHasil << " menit" << detikHasil << " detik"
<< endl;

```

<pre> // Program HitungJam // Menghitung selisih jam #include &lt;iostream&gt; using namespace std;  int main() { // KAMUS   int jam1, menit1, detik1;   int jam2, menit2, detik2;   int totaldetik1, totaldetik2;   int detikSelisih;   int jamHasil, menitHasil, detikHasil;    // ALGORITMA   // Masukan jam pertama   cin &gt;&gt; jam1;   cin &gt;&gt; menit1;   cin &gt;&gt; detik1;   // Masukan jam kedua   cin &gt;&gt; jam2;   cin &gt;&gt; menit2;   cin &gt;&gt; detik2; </pre>	<pre> // Perhitungan selisih jam totaldetik1 = jam1 * 3600 + menit1 * 60 + detik1; totaldetik2 = jam2 * 3600 + menit2 * 60 + detik2;  detikSelisih = totaldetik2 - totaldetik1;  jamHasil = detikSelisih / 3600; menitHasil = (detikSelisih % 60) / 60; detikHasil = detikSelisih % 60;  // Penulisan selisih jam cout &lt;&lt; "Selisih = " &lt;&lt; jamHasil &lt;&lt; " jam "   &lt;&lt; menitHasil &lt;&lt; " menit " &lt;&lt; detikHasil   &lt;&lt; " detik" &lt;&lt; endl;  return 0; } </pre>
---	---

Gambar 7.8. Coding dalam C++ program yang memeriksa perbedaan dua buah jam.

Solusi menggunakan tipe bentukan.

- Deklarasi type Jam:

```

typedef struct {
  int JJ; // Bagian jam
  int MM; // Bagian menit
  int DD; // Bagian detik
} Jam;

```

Program yang memeriksa perbedaan dua buah jam.

- Program menerima masukan jam yang pertama (jam1) :

```

cin >> jam1.JJ;
cin >> jam1.MM;
cin >> jam1.DD;

```

- Kemudian menerima masukan jam yang kedua (jam2):

```

cin >> jam2.JJ;
cin >> jam2.MM;
cin >> jam2.DD;

```

- Kemudian program menghitung selisih waktu

- Konversi dulu ke detik

```

totaldetik1 = jam1.JJ * 3600 + jam1.MM * 60 + jam1.DD;
totaldetik2 = jam2.JJ * 3600 + jam2.MM * 60 + jam2.DD;

```

- Hitung selisihnya

```

detikSelisih = totaldetik2 - totaldetik1;

```

- Hitung selisih dalam bentuk jam (jamHasil):  
 $\text{jamHasil.JJ} = \text{detikSelisih} / 3600;$   
 $\text{jamHasil.MM} = (\text{detikSelisih} \% 3600) / 60;$   
 $\text{jamHasil.DD} = \text{detikSelisih} \% 60;$

- Kemudian Cetak hasil

– Contoh hasil:

Selisih = 5 jam 3 menit 4 detik

– Jadi:

```
cout << "Selisih = " << jamHasil.JJ << " jam" << jamHasil.MM << " menit" << jamHasil.DD <<
"detik" << endl;
```

<pre>// Program HitungJam // Menghitung selisih jam #include &lt;iostream&gt; using namespace std;  typedef struct {     int JJ, MM, DD; } Jam;  int main() { // KAMUS     Jam jam1, jam2;     int totaldetik1, totaldetik2;     int detikSelisih;     Jam jamHasil;      // ALGORITMA     // Masukan jam pertama     cin &gt;&gt; jam1.JJ;     cin &gt;&gt; jam1.MM;     cin &gt;&gt; jam1.DD;</pre>	<pre>// Masukan jam kedua     cin &gt;&gt; jam2.JJ;     cin &gt;&gt; jam2.MM;     cin &gt;&gt; jam2.DD;      // Perhitungan selisih jam     totaldetik1 = jam1.JJ * 3600 + jam1.MM * 60 + jam1.DD;     totaldetik2 = jam2.JJ * 3600 + jam2.MM * 60 + jam2.DD;      detikSelisih = totaldetik2 - totaldetik1;      jamHasil.JJ = detikSelisih / 3600;     jamHasil.MM = (detikSelisih % 60) / 60;     jamHasil.DD = detikSelisih % 60;      // Penulisan selisih jam     cout &lt;&lt; "Selisih = " &lt;&lt; jamHasil.JJ &lt;&lt; " jam "         &lt;&lt; jamHasil.MM &lt;&lt; " menit " &lt;&lt; jamHasil.DD         &lt;&lt; " detik" &lt;&lt; endl;      return 0; }</pre>
---	---

Gambar 7.9. Coding dalam C++ program yang memeriksa perbedaan dua buah jam dengan solusi menggunakan tipe bentukan.

#### 7.4. Daftar Pustaka.

[1] Munir, Rinaldi, (2016), “ALGORITMA DAN PEMROGRAMAN DALAM BAHASA PASCAL, C, DAN C++” EDISI KEENAM, Informatika.

#### 7.5. Pertanyaan Quis.

1. Narasikan yang dapat dipelajari dari bab 7.
2. Eksekusi kan coding dalam C++ mulai dari gambar 7.4. sampai gambar 7.9. pada compiler C++.

## Bab 8 Percabangan, Looping Dan Subprogram

Sub-CP-MK 8 Mampu mencodingkan kedalam pemrograman bahasa tingkat tinggi seperti bahasa C  
Waktu: Dua tatap muka.

### 8.1. Analisis Kasus/Percabangan (dalam Bahasa C++). [1]

Satu Kasus.

```
if ( kondisi ) {  
    aksi;  
}
```

- Kondisi benar → aksi dilakukan.
- Kondisi tidak benar → tidak terjadi apa-apa.

```
#include <iostream>  
using namespace std;  
  
int main () {  
    // KAMUS  
    int i;  
  
    // ALGORITMA  
    cin >> i;  
    if (i >= 0) {  
        cout << "positif" << endl;  
    }  
    return 0;  
}
```

Gambar 8.1. Contoh Satu Kasus.

Dua Kasus Komplementer.

```
if (kondisi) {  
    aksi-1;  
}  
else { // not kondisi  
    aksi-2;  
}
```

- Perhatikan kondisi eksplisitnya!

```
#include <iostream>  
using namespace std;  
int main () {  
    // KAMUS  
    int i;  
    // ALGORITMA  
    cin >> i;  
    if (i >= 0) {  
        cout << "positif";  
    }  
    else { // i < 0  
        cout << "negatif";  
    }  
    return 0;  
}
```

Gambar 8.2. Dua Kasus Komplementer.

Banyak Kasus.

```
if (kondisi-1) {
    aksi-1;
}
else if (kondisi-2) {
    aksi-2;
}
else {
    ...
}
```

- (kondisi-x) domain harga: [true, false].
- Property yang perlu diperhatikan untuk (kondisi-x): disjoint dan complete.

```
#include <iostream>
using namespace std;

int main () {
// KAMUS
    int i;
// ALGORITMA
    cin >> i;
    if (i > 0) {
        cout << "positif" << endl;
    }
    else if (i == 0) {
        cout << "nol" << endl;
    }
    else {
        cout << "negatif" << endl;
    }
}
return 0;
}
```

Gambar 8.3. Contoh Banyak Kasus.

Switch.

```
switch (var) {
    case val-1: aksi-1; break;
    case val-2: aksi-2; break;
    ...
    default: aksi-n; break;
}
```

- var bertipe ordinal, val-1 , val-2, dst adalah constant.
- Break untuk keluar dari switch.
- Default jika tidak ada yang cocok.

```

#include <iostream>
using namespace std;
int main () {
// KAMUS
    int i;
// ALGORITMA
    cin >> i;
    switch (i) {
        case 1 : cout << "januari" << endl;
                break;
        case 2 : cout << "februari" << endl;
                break;
        default: cout << "bulan lain" << endl;
                break;
    }
    return 0;
}

```

Gambar 8.4. Contoh Switch.

Analisis Kasus sebagai sarana validasi.

```

if ( not kondisi-benar ) {
    pesan-kesalahan ;
}
else { // kondisi-benar
    aksi;
}

```

- Digunakan untuk memvalidasi masukan program sehingga masukan yang salah tidak diproses.
- aksi hanya dilakukan jika kondisi-benar terpenuhi, jika tidak maka diberikan pesan-kesalahan.

Contoh analisis kasus untuk validasi

- Buatlah program yang digunakan untuk menuliskan ke layar nama-nama bulan dari nomor bulan, yaitu 1 s.d. 12 yang dibaca dari keyboard.
- Program memvalidasi terlebih dahulu apakah masukan dari keyboard sudah benar atau belum, yaitu apakah nomor bulan sudah tepat 1 s.d. 12. Jika tidak tepat, maka dikeluarkan pesan kesalahan yaitu "Masukan nomor bulan tidak tepat".

```

#include <iostream>
using namespace std;
int main () {
// KAMUS
    int nobulan;
// ALGORITMA
    cin >> nobulan;
    if ((nobulan < 1) || (nobulan > 12)) {
        cout << "Masukan nomor bulan tidak tepat" << endl;
        // pesan kesalahan
    } else { // nobulan di antara 1 s.d. 12
        switch (i) {
            case 1 : cout << "januari" << endl; break;
            case 2 : cout << "februari" << endl; break;
            ... // lanjutkan sendiri
        }
    }
    return 0;
}

```

Gambar 8.5. Contoh analisis kasus untuk validasi.

### Soal-1

- Buatlah program yang menerima masukan 2 buah integer dan menghasilkan manakah dari kedua integer tersebut yang paling besar.
- Misal integernya adalah A dan B:
  - jika  $A \geq B$  maka A,
  - atau  $A < B$  maka B.

```
// Program Max2Integer
// Menuliskan nilai terbesar dari masukan 2 integer
#include <iostream>
using namespace std;
int main () {
// KAMUS
    int A, B;

// ALGORITMA
    cin >> A;
    cin >> B;
    if (A >= B) {
        cout << A << endl;
    } else { // A < B
        cout << B << endl;
    }
    return 0;
}
```

Gambar 8.6. Alternatif Solusi Soal-1.

### Soal-2

- Buatlah sebuah program yang menerima suhu air (dalam derajat celcius) dan menuliskan wujud air ke layar sebagai berikut:
  - Jika suhu air  $\leq 0$  derajat, maka tuliskan “beku”.
  - Jika suhu air  $> 0$  dan kurang dari 100 derajat, maka tuliskan “cair”.
  - Jika suhu air  $\geq 100$ , maka tuliskan “uap”.

```
// Program SuhuAir
// Menuliskan wujud air tergantung masukan suhu
#include <iostream>
using namespace std;
int main () {
// KAMUS
    float suhu;

// ALGORITMA
    cin >> suhu;
    if (suhu <= 0) {
        cout << "Beku" << endl;
    } else if ((suhu > 0) && (suhu < 100)) {
        cout << "Cair" << endl;
    } else { // suhu >= 100
        cout << "Uap" << endl;
    }
    return 0;
}
```

Gambar 8.7. Alternatif Solusi Soal-2.

Soal-3

- Buatlah program yang menerima 3 buah tahanan dan menghasilkan total tahanan jika dirangkai seri.
- Ketiga masukan tahanan tidak boleh bernilai negatif. Jika salah satu masukan tahanan bernilai negatif, maka diberikan pesan kesalahan “Masukan tahanan tidak boleh negatif”.

```
// Program Tahanan Seri
// Menuliskan total tahanan seri dari 3 buah masukan tahanan
#include <iostream>
using namespace std;
int main () {
// KAMUS
    float R1, R2, R3, Rtotal;

// ALGORITMA
    cin >> R1; cin >> R2; cin >> R3;

    if ((R1 >= 0) && (R2 >= 0) && (R3 >= 0)) {
        Rtotal = R1 + R2 + R3;
        cout << Rtotal << endl;
    } else { // Masukan R1 < 0 atau R2 < 0 atau R3 < 0
        cout << "Masukan tahanan tidak boleh negatif" << endl;
    }
    return 0;
}
```

Gambar 8.8. Alternatif Solusi Soal-3.

Soal-4

- Sebuah kalkulator super sederhana Calc-SS, digunakan untuk menerima masukan 2 buah bilangan bulat, misalnya A dan B, dan menuliskan beberapa hasil operasi aritmatika terhadap kedua bilangan tersebut, tergantung pilihan operasi melalui character sebagai berikut:
  - Pilihan = a : menuliskan hasil A + B.
  - Pilihan = b : menuliskan hasil A – B.
  - Pilihan = c : menuliskan hasil A \* B.
  - Pilihan = d : menuliskan hasil A/B→hasilnya real/float.
  - Pilihan = e : menuliskan hasil A div B→hasilnya integer.
  - Pilihan = f : menuliskan hasil A mod B.
- Jika dimasukkan pilihan menu character lain, maka dituliskan “Bukan pilihan menu yang benar”.

Masukan			Keluaran
A	B	pilihan operasi	
3	2	a	A + B = 5
3	2	b	A – B = 1
3	2	c	A * B = 6
3	2	d	A / B = 1.5
3	2	e	A div B = 1
3	2	f	A mod B = 1
3	2	h	Bukan pilihan menu yang benar

Gambar 8.9. Contoh masukan dan keluaran.



```

//Program Calc-SS
//Menuliskan hasil operasi aritmatika antara 2 integer, tergantung
//pilihan menu
#include <iostream>
using namespace std;
int main () {
// KAMUS
    int A, B;
    char pilihan; //pilihan operasi
// ALGORITMA
    cout << "Masukkan bil pertama (A) = "; cin >> A;
    cout << "Masukkan bil kedua (B) = "; cin >> B;
    cout << "Masukkan pilihan operasi = "; cin >> pilihan;
    switch (pilihan) {
        case 'a' : cout << "A + B = " << A + B << endl; break;
        case 'b' : cout << "A - B = " << A - B << endl; break;
        case 'c' : cout << "A * B = " << A * B << endl; break;
        case 'd' : cout << "A / B = " << (float)A / (float)B << endl;
                    break;
        case 'e' : cout << "A div B = " << A / B << endl; break;
        case 'f' : cout << "A mod B = " << A % B << endl; break;
        default : cout << "Bukan pilihan menu yang benar" << endl;
                    break;
    }
    return 0;
}

```

### Alternatif Solusi Soal-4

Type casting dari int menjadi float

Gambar 8.10. Alternatif Solusi Soal-4.

## 8.2. Pengulangan/Looping (dalam Bahasa C++). [1]

Pengulangan / Looping:

- Elemen:
  - Kondisi pengulangan: ekspresi logik.
  - Badan pengulangan: aksi yang diulang.
- Jenis-jenis notasi pengulangan:
  1. Berdasarkan kondisi pengulangan di akhir : while.
  2. Berdasarkan kondisi pengulangan di awal : do-while.
  3. Berdasarkan pencacah : for.

Pengulangan Berdasarkan Kondisi Pengulangan di Awal (while).

*Inisialisasi-aksi*  
*First-Element*

```

while (kondisi-pengulangan)
{
    Aksi
    Next-Element
}

```

//Kondisi-pengulangan=false

*Terminasi*

Gambar 8.11. Pengulangan Berdasarkan Kondisi Pengulangan di Awal (while).

While

- Pengulangan dikendalikan oleh elemen pengulangan yang diinisialisasi sebagai First-Element dan diubah nilainya dalam badan pengulangan menjadi Next- Elem.
- Aksi akan dilakukan selama kondisipengulangan masih dipenuhi pengulangan (berharga true).
- Tes terhadap kondisi-pengulangan dilakukan setiap kali sebelum aksi dilaksanakan.
- Pengulangan ini berpotensi untuk menimbulkan Aksi “kosong” (tidak pernah melakukan apa-apa) karena pada test yang pertama, kondisi pengulangan tidak dipenuhi (berharga false) sehingga langsung ke luar loop.

```
// Program JumlahAngka
// Menghitung 1+2+3+...+N; N > 0
#include <iostream>
using namespace std;
int main () {
// KAMUS
    int N, i, sum;
// ALGORITMA
    cin >> N;
    sum = 0; //Inisialisasi
    i = 1; //First-Element
    while (i <= N) { //Kondisi-pengulangan
        cout << i << endl; //Aksi
        sum = sum + i; //Aksi
        i = i + 1; //Next-Element
    } // i > N
    cout << sum << endl; //Terminasi
    return 0;
}

// Alternatif ekspresi
while (i <= N) {
    cout << i << endl;
    sum+=i;
    i++;
} // i > N
```

Gambar 8.12 Contoh While.

Pengulangan Berdasarkan Kondisi Pengulangan di Akhir (do-while).

```
Inisialisasi-Aksi
First-Element

do
{
    Aksi
    Next-Element

} while (kondisi-pengulangan);

Terminasi
```

Gambar 8.13. Pengulangan Berdasarkan Kondisi Pengulangan di Akhir (do-while).

Do-While

- Pengulangan dikendalikan oleh elemen pengulangan yang diinisialisasi sebagai First-Element dan diubah nilainya dalam badan pengulangan menjadi Next-Element.

- Aksi minimal akan dilakukan satu kali karena pada waktu eksekusi pengulangan yang pertama tidak dilakukan test terhadap kondisi-pengulangan.
- Aksi akan dihentikan jika kondisi-pengulangan tidak dipenuhi (berharga false), akan diulang jika kondisipengulangan tercapai.
- Test terhadap kondisi pengulangan dilakukan setelah Aksi dilaksanakan.
- Pengulangan berpotensi mengalami “kebocoran”, jika ada kemungkinan bahwa seharusnya Aksi tidak Pernah boleh dilakukan untuk kasus tertentu.

```
// Program JumlahAngka
// Menghitung 1+2+3+...+N; N > 0
#include <iostream>
using namespace std;
int main () {
// KAMUS
    int N, i, sum;
// ALGORITMA
    cin >> N;
    sum = 0; //Inisialisasi-aksi
    i = 1; //First-element
    do {
        cout << i << endl; //Aksi
        sum = sum + i; //Aksi
        i = i + 1; //Next-Element
    } while (i <= N); //Kondisi Pengulangan
    cout << sum << endl; //Terminasi
    return 0;
}

// Alternatif ekspresi
do {
    cout << i << endl;
    sum+=i;
    i++;
} while ( i <= N );
```

Gambar 8.14. Contoh Do-While.

Pengulangan Berdasarkan Pencacah (for).

```
Inisialisasi-aksi

for (pencacah = hmin;
    pencacah <= hmaks;
    pencacah++)
{
    Aksi
}

Terminasi
```

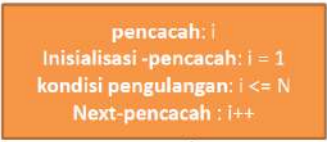
pencacah++ →  
pencacah = pencacah + 1

Gambar 8.15. Pengulangan Berdasarkan Pencacah (for).

For

- pencacah harus suatu variable dengan type yang terdefinisi suksesor dan predesesornya, misalnya integer.
- Nilai pencacah adalah dari hmin s.d. hmaks.
- Aksi akan dilakukan dengan memperhitungkan harga-harga dari pencacah yang di-”jelajahi”.
- Harga pencacah di-increment melalui operasi pencacah++ (alias pencacah=pencacah+1), setiap kali Aksi selesai dilaksanakan.
- Jika pencacah=hmaks, maka pengulangan berhenti.

```
// Program JumlahAngka
// Menghitung 1+2+3+...+N; N > 0
#include <iostream>
using namespace std;
int main () {
// KAMUS
    int N, i, sum;
// ALGORITMA
    cin >> N;
    sum = 0; //Inisialisasi-aksi
    for (i = 1; i <= N; i++) {
        cout << i << endl; //Aksi
        sum = sum + i;      //Aksi, alternatif: sum+=i;
    }
    cout << sum << endl; //Terminasi
    return 0;
}
```



Gambar 8.16. Contoh For.

### 8.3. Subprogram (dalam Bahasa C++). [1]

#### ATM-SS.

- Sebuah mesin ATM Super Sederhana (ATM-SS) milik BSSJ (Bank Super Sederhana Juga) memiliki cash dispenser yang digunakan untuk menampung uang.
- ATM-SS hanya dapat digunakan untuk memasukkan uang ke dalam cash dispenser, menarik uang dari cash dispenser tersebut, dan menampilkan sisa uang yang ada di cash dispenser.
- Untuk mengakses ATM-SS, pengguna harus memasukkan pin. Pin yang valid adalah nilai integer di antara 101 dan 1000. Jika pin tidak valid, maka program langsung selesai.

#### ATM-SS.

- Jika pin valid, pengguna dapat memilih beberapa menu sbb:
  - Jika memasukkan menu huruf ‘A’ atau ‘a’ : memasukkan sejumlah uang ke cash dispenser.
  - Jika memasukkan menu huruf ‘B’ atau ‘b’ : menarik sejumlah uang dari cash dispenser.
  - Jika memasukkan menu huruf ‘C’ atau ‘c’ : menampilkan jumlah uang yang ada di cash dispenser.
  - Jika memasukkan menu huruf ‘X’ atau ‘x’ : keluar dari program.
  - Jika memasukkan huruf lain, program akan menampilkan pesan “Bukan pilihan menu yang benar”.

#### Kode yang berulang.

- Semakin besar program, akan semakin banyak bagian kode yang berulang.
- Sangat tidak efisien jika bagian kode yang sama/serupa diketik berulang-ulang atau bahkan termasuk kalau dicopy paste.
- Di samping itu, dalam banyak persoalan, ada berbagai rumus/formula yang berulang-ulang dipakai dalam satu program.
- Bagaimana jika ada cara supaya bagian kode tersebut tidak perlu diketik berulang-ulang, tapi tetap dapat

digunakan berkali-kali dalam program yang sama.

### Subprogram

- A set of instructions designed to perform a frequently used operation within a program.
- 2 (dua) jenis subprogram:
  - Fungsi.
  - Prosedur.

### Definisi Fungsi.

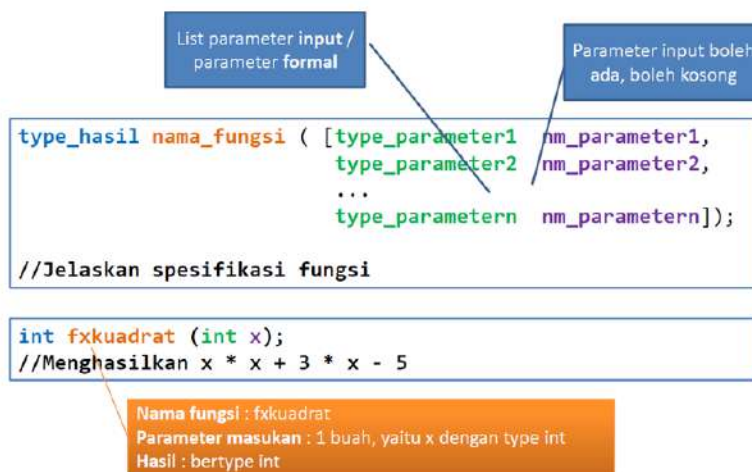
- Fungsi adalah sebuah transformasi akibat pemetaan suatu nilai (dari domain) ke nilai lain (dalam range) → sama seperti di matematika.
- Fungsi mempunyai nama dan sekelompok parameter formal (harga masukan yang diberi nama dan dijelaskan type-nya) serta memiliki hasil (dalam suatu type tertentu pula).
- Fungsi harus didefinisikan terlebih dahulu supaya dapat digunakan dalam bagian ALGORITMA program.

### Tahapan Memanfaatkan Fungsi.

1. Mendefinisikan fungsi.
  - Memberikan nama.
  - Mendefinisikan parameter formal (parameter input).
  - Mendefinisikan type hasil.
2. Merealisasikan fungsi.
  - Membuat algoritma fungsi: memroses input → hasil.
3. Menggunakan fungsi dalam program utama.
  - Memanggil fungsi dengan menggunakan parameter actual.

### Contoh Fungsi

- Fungsi bernama  $f(x)$  memiliki satu parameter  $x$  didefinisikan sebagai  $f(x) = x^2 + 3x - 5$ .
  - jika diberi harga  $x = 4$  maka  $f(x)$  akan menghasilkan 23.
  - jika diberi harga  $x = 1$  maka  $f(x)$  akan menghasilkan -1.
- Fungsi  $f(x,y)$  memiliki dua parameter  $x$  dan  $y$ , didefinisikan sebagai  $f(x,y) = x^2 + 3xy - 5y - 1$ .
  - jika diberi harga  $x = 0$  dan  $y = 0$  maka  $f(x,y)$  akan menghasilkan -1.
  - jika diberi harga  $x = 1$  dan  $y = 0$  maka  $f(x,y)$  akan menghasilkan 0.



Gambar 8.17. Mendefinisikan Fungsi.

Mendefinisikan Fungsi.

- Parameter input boleh tidak ada (kosong).
  - Fungsi tidak membutuhkan apa-apa dari pemakainya untuk menghasilkan harga.
- Jika list parameter input (parameter FORMAL) ada (tidak kosong, minimal satu nama), maka merupakan satu atau beberapa nama beserta type-nya.
- Fungsi harus menghasilkan suatu harga.
  - Harga yang dihasilkan oleh fungsi harus memiliki suatu type tertentu.

```
type-hasil nama_fungsi ( [type-parameter1 nm-parameter1,
                          type-parameter2 nm-parameter2,
                          ...
                          type-parametern nm-parametern] );
// Jelaskan spesifikasi fungsi
{
    // KAMUS LOKAL
    // Deklarasikan semua NAMA yang dipakai dalam algoritma
    // fungsi

    // ALGORITMA
    // Deretan teks algoritma :
    // pemberian harga, analisa kasus, pengurangan, dll.

    // Pengiriman harga di akhir fungsi, harus sesuai dengan
    // type hasil, caranya adalah:
    return (hasil);
}
```

Gambar 8.18. Merealisasikan Fungsi.

```
int fxkuadrat (int x)
//Menghasilkan  $x * x + 3 * x - 5$ 
{
    //KAMUS LOKAL
    //tidak ada nama lokal yang perlu dideklarasikan

    //ALGORITMA
    return (x * x + 3 * x - 5);
}
```

Gambar 8.19. Contoh Realisasi Fungsi.

```

//Judul dan spesifikasi program
#include <iostream>
using namespace std;

DEKLARASI FUNGSI

// PROGRAM UTAMA
int main () {
    PEMAKAIAN FUNGSI
    return 0;
}

REALISASI FUNGSI

```

Dalam REALISASI FUNGSI bisa terdapat pemakaian fungsi lain

Gambar 8.19. Kode Fungsi Dalam Program.

```

//Judul dan spesifikasi program
#include <iostream>
using namespace std;
// DEKLARASI FUNGSI
int fxkuadrat (int x);
//Menghasilkan x * x + 3 * x - 5

// PROGRAM UTAMA
int main () {
    // KAMUS
    int x, p, hasil;
    // ALGORITMA
    x = 3;
    hasil = fxkuadrat(x);
    p = 10 + fxkuadrat(5);
    cout << hasil << " " << p << " " << fxkuadrat(10) << endl;
    return 0;
}

//REALISASI FUNGSI
int fxkuadrat (int x)
//Menghasilkan x * x + 3 * x - 5
{
    //KAMUS LOKAL
    //ALGORITMA
    return (x * x + 3 * x - 5);
}

```

CONTOH

OUTPUT:  
13 45 125

Gambar 8.20. Contoh Kode Fungsi Dalam Program.

### Pemanggilan Fungsi

- Fungsi hanya dapat dipakai sebagai bagian ekspresi—bukan merupakan suatu instruksi yang dipanggil independen.
- Dalam ekspresi, fungsi hanya dapat diletakkan di ruas kanan.
- Saat pemanggilan terjadi korespondensi antara parameter input (formal) dengan parameter aktual sesuai dengan urutan penulisan dalam list-nama parameter input.
- List parameter aktual harus sama jumlah, urutan, dan typenya dengan list parameter input pada pendefinisian fungsinya.
- Harga yang dihasilkan oleh fungsi dapat didefinisikan domainnya dengan lebih rinci
- Pada akhir dari eksekusi fungsi, harga yang dihasilkan oleh fungsi dikirimkan ke pemakainya.
- Fungsi boleh dipakai oleh program utama, prosedur, atau fungsi lain.

#### 8.4. File: atmss.cpp. [1]

```
// File: atmss.cpp
// Dibuat oleh : Fazat Nur Azizah
// Tanggal : 14 September 2013
// Program ATMSS
// Program ATM Super Sederhana (ATM-SS) milik BSSJ (Bank Super Sederhana Juga)
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    char menu;
    int pin;
    long int cashdispenser;
    long int jmluang;

    // ALGORITMA
    cout << "Selamat datang di Mesin ATM Bank Super Sederhana Juga" << endl;
    cout << "Masukkan pin Anda : "; cin >> pin;
    if ((pin >= 101) && (pin <= 1000)) { // Memeriksa akses
        cashdispenser = 0;
        // Tampilan Menu dan Input Menu
        cout << "A atau a : Memasukkan uang ke cash dispenser" << endl;
        cout << "B atau b : Menarik uang" << endl;
        cout << "C atau c : Memeriksa sisa uang di cash dispenser" << endl;
        cout << "X atau x : Selesai" << endl;
        cout << "Silakan masukkan pilihan jenis transaksi = "; cin >> menu;
        while ((menu != 'X') && (menu != 'x')) {
            switch (menu) {
                case 'A' : cout << "Jumlah uang yang dimasukkan = "; cin >> jmluang;
                           cashdispenser = cashdispenser + jmluang;
                           cout << "Jumlah uang dalam cash dispenser = " << cashdispenser << endl;
                           break;
                case 'a' : cout << "Jumlah uang yang dimasukkan = "; cin >> jmluang;
                           cashdispenser = cashdispenser + jmluang;
                           cout << "Jumlah uang dalam cash dispenser = " << cashdispenser << endl;
                           break;
                case 'B' : cout << "Jumlah uang yang ditarik = "; cin >> jmluang;
                           if (cashdispenser - jmluang < 0) {
                               cout << "Jumlah penarikan terlalu besar" << endl;
                               cout << "Tidak bisa menarik lebih besar dari " << cashdispenser << endl;
                           } else { // cashdispenser-jmluang >= 0; sisa uang cukup
                               cashdispenser = cashdispenser - jmluang;
                               cout << "Silakan ambil uang sejumlah = " << jmluang << endl;
                               cout << "Jumlah uang dalam cash dispenser = " << cashdispenser << endl;
                           }
                           break;
                case 'b' : cout << "Jumlah uang yang ditarik = "; cin >> jmluang;
                           if (cashdispenser - jmluang < 0) {
                               cout << "Jumlah penarikan terlalu besar" << endl;
                               cout << "Tidak bisa menarik lebih besar dari " << cashdispenser << endl;
                           } else { // cashdispenser-jmluang >= 0; sisa uang cukup
                               cashdispenser = cashdispenser - jmluang;
                               cout << "Silakan ambil uang sejumlah = " << jmluang << endl;
                               cout << "Jumlah uang dalam cash dispenser = " << cashdispenser << endl;
                           }
                           break;
                case 'C' : cout << "Jumlah uang dalam cash dispenser = " << cashdispenser << endl;
                           break;
                case 'c' : cout << "Jumlah uang dalam cash dispenser = " << cashdispenser << endl;
            }
        }
    }
}
```



```

        break;
    default : cout << "Bukan pilihan menu yang benar" << endl;
        break;
    }
    // Tampilan Menu dan Input Menu
    cout << "A atau a : Memasukkan uang ke cash dispenser" << endl;
    cout << "B atau b : Menarik uang" << endl;
    cout << "C atau c : Memeriksa sisa uang di cash dispenser" << endl;
    cout << "X atau x : Selesai" << endl;
    cout << "Silakan masukkan pilihan jenis transaksi = "; cin >> menu;
    }
    } else {
        cout << "Unauthorized access!" << endl;
    }
    }
    return 0;
}
// Keterangan :
// Bagian kode dengan warna yang sama adalah bagian kode yang sama teksnya

```

#### 8.5. Daftar Pustaka.

[1] Munir, Rinaldi, (2016), "ALGORITMA DAN PEMROGRAMAN DALAM BAHASA PASCAL, C, DAN C++" EDISI KEENAM, Informatika.

#### 8.6. Pertanyaan Quis.

1. Tuliskan pseudocode file : atmss.cpp
2. Buatlah program looping.

## Bagian Keempat Proyek Bersama kelompok

***CPMK IV Mampu menyelesaikan studi kasus aplikasi teknologi informasi. (Proyek bersama kelompok) (S8, S9, S11, KU1, KU2, KU5, KK1, KK2, P1, P2, P3)***

### Bab 9

#### Instalasi System Operasi Linux Di Flashdisk

Sub-CP-MK 9 Mampu menyelesaikan studi kasus aplikasi teknologi informasi. (Proyek bersama kelompok).

Waktu: Dua tatap muka.

##### 9.1. Instalasi system operasi linux di flashdisk.

Setelah mempelajari materi pembelajaran MKU Pengantar Teknologi Informasi, pengetahuan dan skill dasar pemrograman Bahasa C++ diperoleh. Kemampuan dasar ini ditingkatkan dengan mempelajari system operasi linux dari sisi instalasi system operasi di flashdisk. Dengan pengalaman instalasi system operasi linux, program C system operasi system linux yang sangat kompleks memiliki kemiripan dengan program C++ yang dipelajari pada bab sebelumnya yaitu kamus dan algoritma. Apabila program C system operasi linux dilihat source code kernelnya yaitu sebanyak 50.000 line code, kesempatan ini meningkatkan nalar mahasiswa-mahasiswa yang mengambil PTI. Ternyata pengetahuan pemrograman C dan C++ bisa menghasilkan produk yang berupa perangkat lunak yang dimodifikasi untuk dipakai didunia usaha sebagai bukti bahwa pengetahuan tersebut berguna dipasarkan atau diperdagangkan bukan diatas kertas jawaban UAS atau di jurnal tempat hasil penelitian. Kemampuan menjual atau memperdagangkan pengetahuan yang ada pada diri kita merupakan bukti riil daripada jawaban UAS atau Jurnal yaitu **proyek kemanusiaan sehingga kehidupan menjadi lebih baik bagi umat manusia**.

Dengan kesempatan belajar instalasi program C system operasi linux di flasdisk, mahasiswa-masiswa yang mengambil mata kuliah PTI dapat mempelajarinya sendiri melalui [www.youtube.com](http://www.youtube.com) lewat pencarian video dengan mengetik kata kunci “belajar instalasi system operasi linux di flashdisk”. Diberikan kesempatan ini secara kelompok, agar terjadi proses saling membantu dan kebersamaan belajar sehingga setiap mahasiswa memiliki sikap mental peneliti dan pengusaha untuk menyelesaikan Proyek instalasi system operasi linux di flashdisk. Sengaja dalam bab ini, pengetahuan instalasi system operasi linux tidak dinarasikan. Agar mahasiswa mencari sendiri pengetahuan tersebut untuk dipresentasikan dan didemonstrasikan didepan kelas agar dosen pengampu dan mahasiswa mendapatkan pengalaman dari proyek ini.

##### 9.2. Daftar Pustaka.

[1] <https://www.youtube.com>

##### 9.3. Pertanyaan Quis.

1. Presentasikan Proyek instalasi system operasi linux di flashdisk dari setiap kelompok untuk kita.