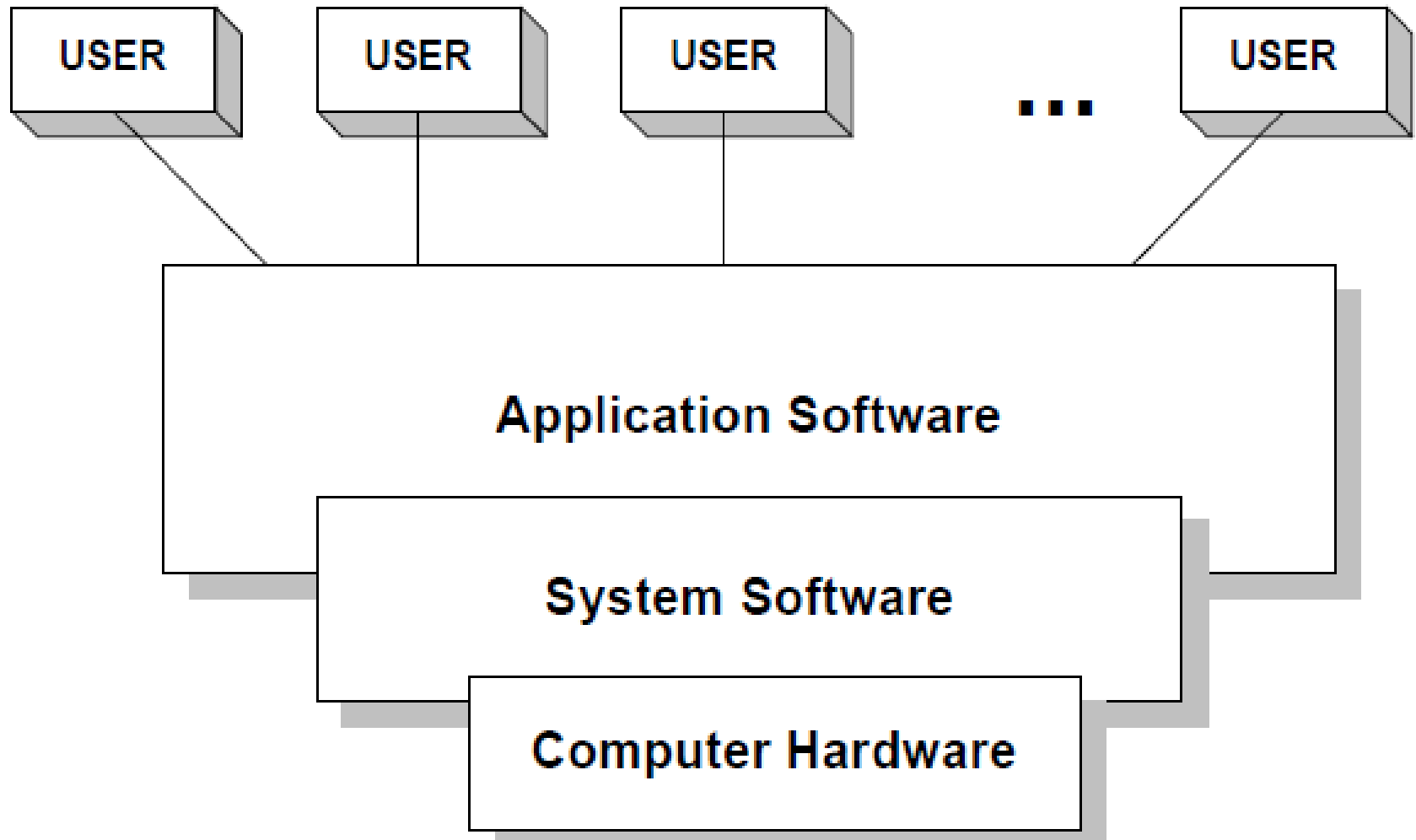


KOMPUTASI

Komputasi

- Pengertian Dasar
- Pemrograman (programming)
- Metode & Teknik
- Struktur Data

Pengertian Dasar



Pengertian Dasar

- **Komputerisasi**
 - Data(masukan), proses, dan simpan informasi dalam komputer atau keluarkan
 - Masuk ke dalam lingkungan masyarakat yang dikelilingi sistem-sistem berbasis komputer (ubiquitous)
 - Melaksanakan, mengendalikan, atau melakukan suatu pekerjaan dengan menggunakan komputer.
 - Contoh ubiquitous systems: hand phone, personal digital assistance (PDA) / personal entertainment organizer (PEO).

Pengertian Dasar

- Komputerisasi, connections:
 - Broad band & connectivity - sistem berbasis komputer dapat disambung ke internet
 - Miniaturization - sistem berbasis komputer akan makin kecil
 - Personalization - perusahaan pembuat akan menciptakan berbagai jenis pilihan, warna, dll. Sesuai selera konsumen
 - Mobility - sistem bersambung ke layanan mobile
 - Portability - sistem bisa dibawa kemanapun
 - Multimedia – smartphome,palm pilots dirancang untuk dapat akses pada media beragam: graphics, text, voice

Komputasi

- Definisi: Kalkulasi
- Synonyms: ciphering, computing, counting, data processing, estimating, estimation, figuring, guess, reckoning, summing, totaling
- Antonyms: conjecture, guessing, guesswork
- Concept: mathematical reasoning

Pemrograman

- • Program ~ instruksi untuk komputer
- • Pemrograman ~ aktifitas runut pembuatan program:
 - Pendefinisian masalah
 - [Deskripsi metoda formal]
 - Penyusunan algoritma [flowchart]
 - Penulisan instruksi (coding) menggunakan sebuah bahasa formal
 - Uji coba & perbaiki kesalahan (debugging)
 - Run program
 - Dokumentasi

Pemrograman

- • Algoritma (algorithm) ~ pseudocode
 - (rules) untuk menyelesaikan suatu masalah
 - Langkah-langkah logis dideskripsi tekstual dengan pola baku untuk menyelesaikan suatu masalah.
- • Program
 - Hasil koversi algoritma ke kode [bahasa program].
 - Spesifikasi rinci sebuah komputasi.
- • Bahasa Program
 - Sebuah notasi untuk mendeskripsi sebuah program.

Pemrograman

- Diagram alir (flowchart)
- Struktur baku flowchart
 - Runtut (sequential / linear)
 - Alternatif/pilihan/seleksi (alternative/selection)
 - Berulang (loop/repetition)

Generasi I(1940-1956)

- Bahasa Mesin Bahasa Generasi Pertama menggunakan kode biner yang terdiri dari string hanya nol (0) dan satu. (1).
- Penggunaan kode biner sulit dipelajari dan digunakan.
- Komputer pertama menggunakan tabung vakum untuk sirkuit dan drum magnetik untuk memori.

Generasi II(1956-1963)

- Bahasa Perakitan
- Bahasa Generasi Kedua menggunakan kode mnemonik yang terdiri dari kata-kata yang sangat singkat untuk perintah.
- Program bahasa assembly perlu diubah menjadi bahasa mesin oleh assembler sebelum bisa dijalankan.
- Tabung hampa digantikan oleh transistor karena lebih unggul, lebih kecil, lebih cepat, lebih murah, hemat energi

Generasi III (1964-1971)

- Bahasa Tingkat Tinggi
- Bahasa Generasi Ketiga memperkenalkan struktur data dan struktur kontrol.
- Bahasa generasi ketiga bersifat portabel atau mesin yang independen dimana program yang ditulis dari satu komputer juga dapat digunakan oleh komputer lain namun kode sumbernya harus dikompilasi ulang oleh kompilator bahasa oleh komputer lain.
- Generasi ini menggunakan Integrated Circuit.
- Fortran, Algol 60 & 68, Pascal C dan ADA adalah BASIC and Java

Generasi IV (1971-Sekarang)

- Bahasa Deklaratif
- Bahasa Generasi Keempat menggunakan perintah seperti bahasa Inggris.
- Menggunakan lingkungan pengembangan graphic(GUI).
- Generasi ini menggunakan Mikroprocessor dimana ribuan sirkuit IC dibangun ke sebuah chip silikon.
- Bahasa yang mencakup dukungan untuk pengelolaan basis data, pembuatan laporan, pengoptimalan matematis, pengembangan GUI, atau pengembangan web
- C#,JavanetBean,Python, Ruby, and Perl,Lisp, SQL dan visual basic,Delphi adalah contohnya

Generasi V

- Bahasa Generasi Kelima pada dasarnya merupakan hasil dari penelitian kecerdasan buatan dari tahun 1980an. Pendekatannya adalah membangun sebuah program yang mengimplementasikan algoritma spesifik untuk memecahkan masalah Kecerdasan buatan.
- Sebagian besar bahasa pemrograman constraint-based dan Logic dan beberapa bahasa deklaratif lainnya merupakan bahasa generasi kelima.
- Contoh: Prolog, OPS5 and Mercury

Pemrograman

- Klasifikasi Paradigma

Imperative/ Algorithmic -----	Declarative -----		Object- Oriented -----
Machine code	Functional/	Logic	Smalltalk
Assembly	Applicative	Programming	Simula
FORTRAN	-----	-----	C++
COBOL	Haskell	PROLOG and	Objective-C
ALGOL	ML	derivatives	Oberon
PL/I	LISP		Modula-3
Pascal	Miranda		Java
Modula-2	Logo		
BASIC	APL		
C			
C++, Java, Modula-3, Oberon			

Pemrograman

- Imperative / algorithmic Paradigm
 - Program dipandang sebagai rentetan instruksi (imperative [Latin] = imperare to command); komputer diperintah untuk mengerjakan sederet step, tiap step ada perubahan dalam isi dan status tiap elemen memori.
 - Secara implisit atau eksplisit program yang dibuat berbasis pada konsep arsitektur komputer von Neumann.
 - Struktur program merupakan kombinasi dari pola-pola:
 - Sequence.
 - Selection.
 - Repetition.
 - Procedural Abstraction
 - Procedural Abstraction with parameter(s).

Pemrograman

- Declarative Paradigm
 - Program merupakan kumpulan deklarasi fungsi-fungsi, dan/atau fakta-fakta
 - PROLOG. Predicate logic.
 - Functional. Functions and application of these functions.
 - Declarative ~ mean that we achieve a program simply by declaring the specification of length.
 - Functional ~ mean a pure mathematical function, [one which takes an argument and returns a result - no side-effects such as change of state (memory contents)].

Pemrograman

- • Object-oriented Paradigm
 - Dalam bahas yang murni OO, objects adalah separate entities yang merepresentasikan state and the means to compute.
 - Filosofi
 - OOA + OOD = OOAD
 - OOL : C++, Java, Modula-2, Newton Script, ADA
 - OOP
 - OODB --> OODBMS
 - C++ dapat dipandang sebagai sebuah bahasa hibrid, imperative/object-oriented language

Metode & Teknik

- Metode Terstruktur
 - ~Metodologi berdasar pada structure & discipline bagaimana membuat sebuah program: bentuk / pola, proses desain, pengkodean, dan uji coba.
 - Objektif
 - readability
 - efficiency
 - reliability
 - discipline (systematic & integrity)
 - reduce cost
 - GOTO-less programming

Metode & Teknik

- Metode Modular

- ~ organizing a program into small, independent units [modules].
- Goals
 - Cacah program menjadi bagian-bagian kecil.
 - Cacah masalah menjadi lebih kecil & sederhana.
 - Verifikasi kebenaran penggunaan tiap modul program independently sebagai bagian dari sistem lebih besar.
- Properties
 - tiap modul: simple, closed, discrete & visible, separately testable
 - tiap modul hanya memiliki single entry & exit point.

Metode & Teknik

- Metode Stepwise Refinement
 - Proses dikerjakan dengan runtutan langkah pasti
 - Tiap langkah jelas kaitan tasks & struktur data
 - Rincian tiap langkah.
- Metoda Top-down / Bottom-up
 - Designing, coding, and testing sebuah program dengan tahapan progresif.
 - Dihasilkan runtut tahapan yang structured & modular
 - Makin bawah modul = subfunctions yang mendefinisikan tasks makin detail.

Metode & teknik

- Teknik Diagram
 - Kategori
 - overview system analysis
 - program architecture
 - program detail
 - data structures
 - Tipe
 - DFD, functional decomposition, structure charts
 - HIPO, Warnier-Orr
 - flowchart, pseudocode, Nassi-Schneidermann
 - action diagram, decision tree, decision table
 - data structure, ERD, HOS charts, IDEF

Metode & teknik

- Diagramming tools
 - AutoSketch; CorelDraw
 - Designer; EasyFlow
 - FlowModel; Illustrator
 - PHOTO-PAINT; Photoshop
 - PowerPoint; netViz
 - QA-Flow; Schematic;
 - SiteMapper; SmartDraw
 - TeamFlow; Visio Technical
 - Visual Thought; WBFlow

Struktur Data

- Data structure
 - Cara penyimpanan data dalam komputer sehingga dapat digunakan secara efisien.
 - Pemilihan struktur data sangat penting dalam desain program [pemrograman].
- Abstract data type (ADT)
 - Struktur data dan operasi-operasi yang dapat dilakukan pada data tersebut.
 - Sebuah class dalam OOD = ADT, classes memiliki sifat-sifat tambahan (inheritance and polymorphism) not normally associated with ADTs.

Struktur Data

- Data Model
 - Hasil proses desain basisdata yang dapat dipakai untuk mengidentifikasi dan mengorganisasi data yang dibutuhkan logically & physically.
 - Data model akan menunjukkan semua informasi yang harus terkandung dalam basisdata, bagaimana cara penggunaannya, dan bagaimana setiap item database akan terkait satu sama lainnya.
- Data warehouse
 - Kumpulan terpadu data perusahaan, dapat diakses oleh business managers, administrators, service providers & researchers yang berhak untuk: dianalisis, menghasilkan ad hoc queries & reports, dan cara pemeliharaan data subsets.

Struktur Data

- Jenis [http://en.wikipedia.org/wiki/Data_structure]
 - Linear data structures:
 - list - array [bitmaps: images, heightfields/Digital Elevation Model]
 - linked list - skip list
 - Hash table
 - Stack [LIFO]
 - Queue [FIFO]
 - Deque
 - – Graph data structures
 - Scene graph
 - Tree: AVL, binary search, B, Parse, Red-black, Splay, Heap String trie, Suffix trie