



Penerbit:
HADIA MEDIA INFORMASI

ORGANISASI DAN ARSITEKTUR KOMPUTER

Alex Copernikus Andaria, S.T., M.Pd
Bahtara Alam Wijayanto, M.Kom
Miwadi, S.Kom., M.Kom
Eka Prasetya Adhy Sugara, S.T., M.Kom
Suwardono, S.Kom., M.Kom
Salman Faricy, S.Kom., M.Kom
S. Narmuslimah, S.T., M.T.
Prajoko, M.Kom
Muhammad Syarif Hartawan
Prayogo, S.Kom., M.Kom
Muhammad Khaerudin, M.Kom
Iman Zaenuddin, S.Kom., M.Kom
Deni Kuswoyo, S.Kom., M.Kom
Hari Wahyudi, M.Ts
Budi Berinton Sitrus, S.T., M.Sc.
Nuhadi, S.Kom., M.Kom
Tuti Handayani, S.Kom., M.Kom

Editor :
Nuhadi, S.Kom., M.Kom.

ORGANISASI DAN ARSITEKTUR KOMPUTER

Penulis

**Alex Copernikus Andaria
Bahtera Alam Wijaksono
Miswadi
Eka Prasetya Adhy Sugara
Suwandono
Salman Farizy
S. Nurmuslimah
Prajoko
Muhammad Syarif Hartawan
Prayogo
Khaerudin
Imam Zaenuddin
Deni Kuswoyo
Hari Wahyudi
Budi Berlinton Sitorus
Nurhadi
Tuti Handayani**

PENERBIT:



Website: www.media.hadlacorp.com

UU No 28 tahun 2014 tentang Hak Cipta
Pasal 113

- 1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
- 2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).
- 3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
- 4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp. 4.000.000.000,00 (empat miliar rupiah).

ORGANISASI DAN ARSITEKTUR KOMPUTER

Tim Penulis:

Alex Copernikus Andaria

Bahtera Alam Wijaksono

Miswadi

Eka Prasetya Adhy Sugara

Suwandono

Salman Farizy

S. Nurmuslimah

Prajoko

Muhammad Syarif Hartawan

Prayogo

Khaerudin

Imam Zaenuddin

Deni Kuswoyo

Hari Wahyudi

Budi Berlinton Sitorus

Nurhadi

Tuti Handayani

Desain Cover:

Sulaiman

Tata Letak:

Sulaiman

Editor

Nurhadi

ISBN:

-

Cetakan Pertama:

Oktober, 2024

Hak Cipta 2024, Pada Penulis

Hak Cipta Dilindungi Oleh Undang-Undang

Copyright © 2024

by HADLA Media Informasi

All Right Reserved

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit

PRAKATA

Puji syukur kepada Allah SWT, karena atas rahmat dan karunia-Nya, buku ajar yang berjudul *Arsitektur dan Organisasi Komputer* ini dapat diselesaikan. Buku ini hadir sebagai referensi bagi mahasiswa dan pembelajar di bidang teknologi informasi, khususnya dalam memahami konsep dasar dan lanjutan terkait arsitektur serta organisasi komputer.

Materi dalam buku ini disusun untuk memberikan pemahaman yang terstruktur mengenai bagaimana sebuah sistem komputer bekerja, mulai dari komponen-komponen dasar seperti prosesor, memori, hingga komunikasi data di dalam sistem tersebut. Pemahaman mengenai arsitektur komputer sangat penting, tidak hanya dalam konteks teori, tetapi juga dalam aplikasinya di dunia teknologi yang terus berkembang pesat.

Buku ini diharapkan dapat menjadi panduan praktis dan komprehensif, disertai contoh-contoh dan soal-soal latihan yang relevan dengan perkembangan terkini. Kami berharap, buku ini dapat memudahkan para pembaca dalam memahami proses desain, pengelolaan, dan optimalisasi sebuah sistem komputer.

Akhir kata, kami berharap buku *Arsitektur dan Organisasi Komputer* ini dapat memberikan manfaat yang sebesar-besarnya bagi semua pihak yang terlibat dalam dunia pendidikan, penelitian, dan pengembangan teknologi komputer.

Terima kasih
Team Penulis

DAFTAR ISI

PRAKATA	iv
DAFTAR ISI	v

KEGIATAN BELAJAR I PENGANTAR ORGANISASI DAN ARSITEKTUR KOMPUTER	1
DESKRIPSI PEMBELAJARAN.....	1
KOMPETENSI PEMBELAJARAN	1
A. DEFINISI ORGANISASI DAN ARSITEKTUR KOMPUTER.....	2
B. PERBEDAAN ARSITEKTUR DAN ORGANISASI KOMPUTER.....	6
C. SEJARAH PERKEMBANGAN KOMPUTER	9
D. RANGKUMAN	11
E. TES OBJEKTIF (PILIHAN GANDA)	13

KEGIATAN BELAJAR II EVOLUSI DAN KINERJA KOMPUTER	15
DESKRIPSI PEMBELAJARAN.....	15
KOMPETENSI PEMBELAJARAN	15
A. SEJARAH PERKEMBANGAN GENERASI KOMPUTER ...	15
B. DESAIN KINERJA.....	24
C. RANGKUMAN.....	25
D. TES FORMATIF.....	25
E. LATIHAN	25

KEGIATAN BELAJAR III SISTEM PERANGKAT LUNAK	27
DESKRIPSI PEMBELAJARAN.....	27
KOMPETENSI PEMBELAJARAN	27
A. PENDAHULUAN	28
B. KLASIFIKASI SISTEM PERANGKAT LUNAK.....	29
C. ARSITEKTUR PERANGKAT LUNAK	32
D. SIKLUS HIDUP PENGEMBANGAN	
E. PERANGKAT LUNAK.....	36
F. KEAMANAN SISTEM PERANGKAT LUNAK.....	39
G. UJICOBA DAN VERIFIKASI PERANGKAT LUNAK.....	43

G. RANGKUMAN	48
H. TES FORMATIF	49
KEGIATAN BELAJAR IV PEMROSESAN DATA.....	53
DESKRIPSI PEMBELAJARAN	53
KOMPETENSI PEMBELAJARAN.....	53
A. DATA DAN PEMROSESAN DATA.....	53
B. SIKLUS PEMROSESAN DATA	54
C. JENIS-JENIS PEMROSESAN DATA	56
D. KOMPONEN UTAMA PEMROSESAN DATA	56
E. TEKNIK PEMROSESAN DATA	59
F. TEKNOLOGI BARU DALAM PEMROSESAN DATA.....	73
G. RANGKUMAN.....	75
H. TES FORMATIF.....	77
I. LATIHAN DAN STUDI KASUS	83
KEGIATAN BELAJAR V SISTEM BILANGAN DAN	
REPRESENTASI DATA	85
DESKRIPSI PEMBELAJARAN	85
KOMPETENSI PEMBELAJARAN.....	85
A. PENGANTAR SISTEM BILANGAN	85
B. SISTEM BILANGAN DESIMAL, BINER, OKTAL, DAN	
HEKSADESIMAL	86
C. KONVERSI ANTAR SISTEM BILANGAN	88
D. OPERASI ARITMETIKA DALAM SISTEM BILANGAN	
BINER	89
E. KOMPLEMEN SATU DAN KOMPLEMEN DUA	89
F. REPRESENTASI DATA.....	90
G. REPRESENTASI BILANGAN FLOATING-POINT	92
H. ENDIANNESS: BIG-ENDIAN VS LITTLE-ENDIAN	92
I. REPRESENTASI KARAKTER DENGAN ASCII DAN	
UNICODE	92
J. REPRESENTASI GAMBAR DAN GRAFIK KOMPUTER ..	93
K. APLIKASI SISTEM BILANGAN	
DALAM PEMROGRAMAN.....	93
L. KOMPRESI DATA DAN EFISIENSI REPRESENTASI.....	94
M. REPRESENTASI GRAFIK DAN GAMBAR KOMPUTER ..	94

N. PENGGUNAAN SISTEM BILANGAN DALAM KRIPTOGRAFI	95
O. APLIKASI SISTEM BILANGAN DAN REPRESENTASI DATA DALAM DUNIA NYATA	96
P. SISTEM BILANGAN DALAM KECERDASAN BUATAN (ARTIFICIAL INTELLIGENCE).....	97
Q. PENGEMBANGAN SISTEM KRIPTOGRAFI DAN BLOCKCHAIN	98
R. SISTEM BILANGAN DAN PENGOLAHAN SINYAL DIGITAL.....	99
S. SISTEM BILANGAN DAN PEMROGRAMAN TINGKAT RENDAH	99
T. APLIKASI SISTEM BILANGAN DALAM GAME DEVELOPMENT	100
U. SISTEM BILANGAN DALAM KOMPUTASI KUANTUM...	100
V. SISTEM BILANGAN DALAM INTERNET OF THINGS (IOT).....	101
W. SISTEM BILANGAN DALAM BIG DATA DAN DATA ANALYTICS	102
X. SISTEM BILANGAN DAN KEAMANAN SIBER	103
Y. RANGKUMAN	104
Z. LATIHAN	104
KEGIATAN BELAJAR VI BAHASA ASSEMBLY	107
DESKRIPSI PEMBELAJARAN	107
KOMPETENSI PEMBELAJARA.....	107
A. PENDAHULUAN	107
B. ARSITEKTUR KOMPUTER DAN BAHASA ASSEMBLY.....	115
C. DASAR – DASAR PEMEROGAMMAN ASSEMBLY	122
D. RANGKUMAN	128
E. TEST FORMATIF	128
F. LATIHAN	129
KEGIATAN BELAJAR VII KOMPUTER DAN PROSESOR.....	131
DESKRIPSI PEMBELAJARAN	131
KOMPETENSI PEMBELAJARAN	131
A. PENGERTIAN KOMPUTER.....	131

B. SEJARAH KOMPUTER.....	133
C. KEMAMPUAN KOMPUTER	135
D. STRUKTUR KOMPUTER.....	136
E. FUNGSI KOMPUTER.....	138
F. PROSESOR	141
G. MAINBOARD	142
H. CARA KERJA PROSESOR.....	149
I. INTERAKSI PROSESOR	149
J. RANGKUMAN	150
K. LATIHAN.....	151

KEGIATAN BELAJAR VIII SISTEM

PENGENDALIAN KOMPUTER.....	153
DESKRIPSI PEMBELAJARAN.....	153
KOMPETENSI PEMBELAJARAN	153
A. PENDAHULUAN.....	153
B. DISKRITISASI DAN IKHTISAR SISTEM DATA SAMPEL.....	157
C. MODEL WAKTU DISKRIT	162
D. RANGKUMAN	177
E. TEST FORMATIF	178
F. LATIHAN.....	179

KEGIATAN BELAJAR IX ARSITEKTUR KOMPUTER MASA DEPAN

DESKRIPSI PEMBELAJARAN.....	181
KOMPETENSI PEMBELAJARAN	181
A. ARSITEKTUR KOMPUTER MASA DEPAN	181
B. INOVASI DALAM DESAIN PERANGKAT KERAS	182
C. KOMPUTASI TERDISTRIBUSI DAN AWAN.....	184
D. EFISIENSI ENERGI.....	185
E. KEAMANAN DAN PRIVASI.....	187
F. INTEGRASI KECERDASAN BUATAN (AI).....	189
G. PENGALAMAN PENGGUNA YANG DITINGKATKAN (UI/UX)	191
H. RANGKUMAN	193
I. LATIHAN.....	194

KEGIATAN BELAJAR X LOGIKA, GERBANG, REGISTER, DAN MEMORI.....	197
DESKRIPSI PEMBELAJARAN.....	197
KOMPETENSI PEMBELAJARAN.....	197
A. PENDAHULUAN.....	197
B. SILICON BASED TRANSISTORS.....	199
C. COMBINATIONAL LOGIC.....	207
D. SEQUENTIAL LOGIC.....	216
E. MEMORIES.....	230
F. RANGKUMAN.....	234
G. TES FORMATIF.....	236
H. LATIHAN.....	237
KEGIATAN BELAJAR XI ARITMATIKA KOMPUTER	226
DESKRIPSI PEMBELAJARAN.....	239
KOMPETENSI PEMBELAJARAN.....	239
A. PENDAHULUAN.....	239
B. ARITMATIKA DAN LOGIKA.....	240
C. STRUKTUR DAN FUNGSI ALU.....	241
D. RANGKAIAN ARITMETIKA.....	242
E. REPRESENTASI INTEGER.....	249
F. TEKNIK REPRESENTASI ANGKA.....	249
G. REPRESENTASI KOMPLEMEN DUA.....	255
H. OPERASI ARITMATIKA BILANGAN BINER TANPA KOMPLEMEN DUA.....	258
I. RANGKUMAN.....	261
J. TEST NORMATIF.....	262
K. LATIHAN SOAL.....	263
KEGIATAN BELAJAR XII PROCESSOR DESIGN	265
DESKRIPSI PEMBELAJARAN.....	265
KOMPETENSI PEMBELAJARAN.....	265
A. DESAIN DASAR PROSESSOR.....	265
B. GAYA DESIGN PROSESOR.....	266
C. PROSES PEMROSESAN INSTRUKSI.....	269
D. UNIT PENGAMBILAN INSTRUKSI.....	271
E. UNIT PENGAMBILAN OPERAND.....	274

F. UNIT AKSES MEMORI	277
G. UNIT EKSEKUSI	280
H. UNIT PENULISAN KEMBALI REGISTER	283
I. DATA PATH DAN CONTROL PATH	287
J. RANCANGAN UNIT EKSEKUSI	290
K. DESAIN ALU (ARITHMETIC LOGIC UNIT)	292
L. IMPLEMENTASI MEMORI	295
M. RANCANGAN KONTROL UNIT	298
N. MIKROPROGRAMMED PROCESSOR	301
O. BUS DALAM MIKROPROGRAMMED PROCESSOR	304
P. RANCANGAN UNIT FETCH MIKROPROGRAMMED	306
Q. RANCANGAN DECODE UNIT	308
R. RANGKUMAN	311
S. TES FORMATIF	312

KEGIATAN BELAJAR XIII PENGOPTIMALAN KINERJA

KOMPUTER	317
DESKRIPSI PEMBELAJARAN	317
KOMPETENSI PEMBELAJARAN	317
A. PENGANTAR OPTIMASI KINERJA	317
B. DEFINISI KINERJA KOMPUTER	318
C. DAMPAK KINERJA KOMPUTER	320
D. PENGALAMAN PENGGUNA	322
E. FAKTOR-FAKTOR KINERJA KOMPUTER	324
F. TEKNIK DAN STRATEGI MENINGKATKAN KINERJA KOMPUTER	325
G. PERANGKAT LUNAK PENDUKUNG	327
H. RANGKUMAN	330
I. LATIHAN	331

KEGIATAN BELAJAR XIV MEMORI KOMPUTER.....333

DESKRIPSI PEMBELAJARAN	333
KOMPETENSI PEMBELAJARAN	333
A. PENTINGNYA MEMAHAMI SISTEM MEMORI	333
B. PENGERTIAN MEMORI	334
C. FUNGSI MEMORI	339
D. KINERJA MEMORI.....	339

E. JENIS MEMORI.....	341
F. HIERARKI MEMORY.....	350
G. KAPASITAS DAN BANDWIDTH MEMORI KOMPUTER	352
H. RANGKUMAN	353
I. SOAL LANJUTAN.....	354
KEGIATAN BELAJAR XV SISTEM MULTIPROSESOR.....	355
DESKRIPSI PEMBELAJARAN.....	355
KOMPETENSI PEMBELAJARAN	355
A. LATAR BELAKANG.....	355
B. KLASIFIKASI MULTIPROSESOR.....	359
C. MULTIPROSESOR MIMD	362
D. KOHEREN DAN KONSISTENSI MEMORI	365
E. LOOSELY-COUPLED ARCHITECTURE	367
F. RANGKUMAN	369
G. LATIHAN.....	369
KEGIATAN BELAJAR XVI ARSITEKTUR	
KOMPUTER MODERN	371
DESKRIPSI PEMBELAJARAN.....	371
KOMPETENSI PEMBELAJARAN	371
A. ARSITEKTUR KOMPUTER BERBASIS CLOUD	371
B. VIRTUALISASI DAN KOMPUTASI AWAN	377
C. SISTEM EMBEDDED DAN IOT (INTERNET OF THINGS).....	380
D. QUANTUM COMPUTING: TANTANGAN DAN MASA DEPAN.....	384
E. RANGKUMAN.....	387
F. TES FORMATIF	388
G. LATIHAN.....	388
KEGIATAN BELAJAR XVII PERANGKAT I/O DAN	
PENYIMPANAN	389
DESKRIPSI PEMBELAJARAN.....	389
KOMPETENSI PEMBELAJARAN	389
A. DEFINISI INPUT / OUTPUT (I/O).....	389
B. SISTEM INPUT / OUTPUT (I/O).....	389
C. PERANGKAT YANG DIBUTUHKAN SISTEM I/O	392

D. DESAIN SISTEM I/O.....	393
E. LAPISAN DALAM SISTEM I/O.....	395
F. I/O TECHNIQUE.....	397
G. PENYIMPANAN ATAU STORAGE.....	398
H. RANGKUMAN.....	408
I. TES FORMATIF.....	408
J. LATIHAN.....	408
DAFTAR PUSTAKA	409
PROFIL PENULIS	427

KEGIATAN BELAJAR I

PENGANTAR ORGANISASI DAN ARSITEKTUR KOMPUTER

Oleh: Alex Copernikus Andaria, S.T., M.Pd.

DESKRIPSI PEMBELAJARAN

Deskripsi pembelajaran ini memberikan gambaran singkat namun komprehensif tentang apa yang diharapkan dapat dicapai oleh mahasiswa setelah mempelajari bab pengantar ini. Deskripsi ini mencakup pemahaman konseptual, kemampuan analitis, dan pengetahuan historis yang relevan dengan topik organisasi dan arsitektur komputer. Deskripsi ini dapat digunakan sebagai panduan bagi mahasiswa dan instruktur untuk memahami tujuan dan hasil belajar yang diharapkan dari bab ini.

KOMPETENSI PEMBELAJARAN

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Mendefinisikan dan membedakan konsep arsitektur komputer dan organisasi komputer.
 2. Mengidentifikasi komponen-komponen utama dalam arsitektur dan organisasi komputer, termasuk ISA, CPU, memori, dan sistem bus.
 3. Menganalisis hubungan antara arsitektur dan organisasi komputer dalam konteks desain sistem.
 4. Menjelaskan evolusi historis teknologi komputer dan dampaknya terhadap arsitektur dan organisasi modern.
 5. Mendeskripsikan tren terkini dalam arsitektur dan organisasi komputer, seperti multicore processing, komputasi kuantum, dan IoT.
 6. Mengaitkan prinsip-prinsip arsitektur dan organisasi komputer dengan aplikasi praktis dalam sistem komputasi modern.
 7. Mengenali hubungan antara arsitektur dan organisasi komputer dengan bidang terkait seperti rekayasa perangkat lunak dan keamanan informasi.
-

A. DEFINISI ORGANISASI DAN ARSITEKTUR KOMPUTER

1. William Stallings

Dalam bukunya "Computer Organization and Architecture: Designing for Performance," (Stallings 2016) memberikan definisi yang komprehensif mengenai organisasi dan arsitektur komputer:

Arsitektur Komputer: Menurut Stallings, arsitektur komputer adalah "atribut-atribut sistem komputer yang terlihat oleh pemrogram, atau dengan kata lain, atribut-atribut yang mempengaruhi eksekusi logis dari sebuah program." Dengan kata lain, arsitektur komputer mengacu pada komponen dan karakteristik sistem yang berhubungan langsung dengan interaksi pemrogram dengan perangkat keras, seperti set instruksi, register, mode pengalamatan, serta struktur memori.

Arsitektur komputer mencakup:

- a. **Instruction Set Architecture (ISA):** Ini adalah bagian dari arsitektur yang mendefinisikan bagaimana perangkat lunak berinteraksi dengan perangkat keras. ISA menentukan instruksi apa yang tersedia untuk programmer, bagaimana data disimpan dalam memori, serta bagaimana data tersebut diakses.
- b. **Struktur Data:** Arsitektur juga mendefinisikan bagaimana data disusun dan diproses di dalam sistem, termasuk bagaimana CPU menangani operasi matematika atau logika.

Contoh nyata dari arsitektur komputer adalah x86, ARM, atau RISC-V, yang merupakan jenis arsitektur yang mendefinisikan cara kerja instruksi pada sistem tertentu.

Organisasi Komputer: Stallings menjelaskan bahwa organisasi komputer berfokus pada "unit-unit operasional dan interkoneksi mereka yang merealisasikan spesifikasi arsitektural." Dalam pengertian ini, organisasi komputer mengacu pada desain fisik dari komponen-komponen dalam sistem komputer, seperti CPU, bus, memori, serta perangkat input/output (I/O).

Aspek penting dari organisasi komputer meliputi:

- a. **Implementasi CPU:** Bagaimana unit pemrosesan pusat diimplementasikan untuk menjalankan instruksi dari ISA. Ini melibatkan pembuatan jalur data (data path), pengelolaan sinyal kontrol, dan eksekusi pipeline.
 - b. **Struktur Memori:** Desain fisik dari sistem memori, termasuk cache, RAM, dan memori sekunder, serta bagaimana memori tersebut diakses oleh CPU.
-

- c. Bus dan I/O: Bagaimana perangkat keras saling berkomunikasi melalui bus, saluran data, dan perangkat input/output.

Stallings menekankan bahwa perbedaan utama antara arsitektur dan organisasi komputer adalah bahwa arsitektur berfokus pada bagaimana sistem terlihat dan digunakan oleh programmer, sementara organisasi berfokus pada bagaimana komponen-komponen fisik tersebut diimplementasikan dan dihubungkan.

2. John L. Hennessy dan David A. Patterson

Dalam "Computer Architecture: A Quantitative Approach," Hennessy dan Patterson memberikan pendekatan yang sangat kuantitatif dalam mendefinisikan arsitektur dan organisasi komputer:

Arsitektur Komputer: Menurut mereka, arsitektur komputer adalah "antarmuka antara perangkat keras dan perangkat lunak tingkat rendah." Ini berarti arsitektur menghubungkan desain perangkat keras dengan perangkat lunak yang menjalankan instruksi pada level terendah. Definisi ini menekankan pentingnya arsitektur dalam menentukan kinerja sistem secara keseluruhan, terutama pada era sistem parallel dan prosesor multicore.

Mereka menekankan beberapa aspek penting dari arsitektur komputer:

- a. **Design Trade-offs:** Arsitektur komputer selalu melibatkan keputusan tentang keseimbangan antara berbagai metrik kinerja seperti kecepatan, konsumsi daya, dan biaya produksi. Sebagai contoh, perancangan arsitektur multicore harus mempertimbangkan bagaimana menyeimbangkan throughput (jumlah pekerjaan yang bisa dilakukan dalam waktu tertentu) dengan konsumsi daya.
- b. **Instruction Set Architecture (ISA):** Hennessy dan Patterson juga berfokus pada pentingnya ISA sebagai komponen inti arsitektur yang mendefinisikan format instruksi, jenis instruksi, serta cara CPU berkomunikasi dengan perangkat keras lainnya.

Organisasi Komputer: Organisasi komputer menurut mereka adalah implementasi fisik dari arsitektur yang telah didefinisikan. Ini mencakup bagaimana elemen-elemen seperti CPU, memori, dan perangkat I/O diatur dan dikendalikan untuk menjalankan instruksi.

Beberapa poin penting dalam organisasi komputer yang diungkapkan Hennessy dan Patterson meliputi:

- a. Pipeline Processing: Organisasi komputer sering melibatkan penggunaan pipeline untuk meningkatkan efisiensi eksekusi instruksi. Setiap tahap pipeline memproses sebagian dari instruksi sehingga beberapa instruksi dapat diproses secara paralel.
- b. Cache dan Memori: Pengaturan hierarki memori, seperti penggunaan cache memori untuk menyimpan data yang sering diakses, sangat penting dalam organisasi komputer untuk mengoptimalkan waktu akses data.

Hennessy dan Patterson juga memperkenalkan konsep quantitative approach dalam mengevaluasi arsitektur dan organisasi komputer, yang memungkinkan insinyur untuk secara akurat mengukur kinerja sistem dan membandingkan berbagai desain berdasarkan metrik seperti MIPS (Millions of Instructions Per Second), FLOPS (Floating Point Operations Per Second), dan throughput.

3. Andrew S. Tanenbaum dan Todd Austin

Dalam buku "Structured Computer Organization," Tanenbaum dan Austin memberikan perspektif yang sedikit berbeda namun sangat informatif mengenai organisasi dan arsitektur komputer (Tanenbaum and Austin 2013):

Arsitektur Komputer: Menurut mereka, arsitektur komputer adalah "set instruksi dari komputer serta sumber daya perangkat keras yang tersedia bagi perangkat lunak, seperti register dan jalur data." Definisi ini menekankan hubungan antara perangkat keras dan perangkat lunak pada level logis, dan bagaimana arsitektur menyediakan alat bagi programmer untuk memanfaatkan perangkat keras secara efektif.

Arsitektur komputer Tanenbaum mencakup:

- a. Registers: Bagian dari CPU yang menyimpan data sementara selama eksekusi instruksi. Register adalah salah satu sumber daya yang paling sering digunakan oleh perangkat lunak.
- b. Data Path: Jalur fisik tempat data dipindahkan di dalam CPU, dari satu register ke register lainnya, atau dari memori ke CPU.

Organisasi Komputer: Organisasi menurut Tanenbaum adalah cara komponen perangkat keras diimplementasikan dan

dihubungkan untuk mewujudkan arsitektur yang dirancang.

Organisasi mencakup:

- a. Microprogramming: Teknik pengendalian perangkat keras menggunakan mikroinstruksi yang diambil dari kontroler mikroprogrammed. Ini adalah salah satu aspek penting dalam organisasi CPU.
- b. Pipelining and Parallelism: Penggunaan pipeline untuk meningkatkan efisiensi instruksi dan penggunaan paralelisme di berbagai level, seperti pada instruksi dan prosesor, adalah bagian integral dari organisasi komputer.
- c. Tanenbaum juga menjelaskan bahwa arsitektur komputer adalah sesuatu yang tetap, sementara organisasi komputer bisa berubah tanpa mengubah arsitektur, tergantung pada evolusi teknologi perangkat keras yang digunakan.

4. M. Morris Mano

Dalam bukunya "Computer System Architecture," M. Morris Mano menguraikan definisi yang sangat teknis mengenai arsitektur dan organisasi komputer (Brey 2009; Manno 2017):

Arsitektur Komputer: Menurut Mano, arsitektur komputer adalah "desain konseptual dan fundamental dari sistem komputer, yang meliputi struktur dan perilaku sistem." Arsitektur komputer mencakup semua aspek logis dari sistem, termasuk set instruksi, ALU, register, dan memori.

Beberapa aspek penting yang dijelaskan Mano dalam arsitektur komputer adalah:

- a. Arithmetic Logic Unit (ALU): Unit yang bertanggung jawab untuk melakukan operasi aritmatika dan logika dalam CPU.
- b. Control Unit (CU): Bagian dari CPU yang mengendalikan eksekusi instruksi berdasarkan instruksi yang disimpan dalam memori.

Organisasi Komputer: Mano berfokus pada bagaimana desain fisik dari komputer diimplementasikan. Ini mencakup distribusi data, manajemen sinyal kontrol, dan penggunaan berbagai unit fungsional seperti ALU dan register untuk mengimplementasikan set instruksi. Mano juga memberikan penekanan pada bagaimana organisasi komputer memungkinkan implementasi arsitektur yang efisien, dengan fokus pada penggunaan minimal daya dan biaya, tetapi tetap mempertahankan kinerja yang optimal.

5. Barry B. Brey

Barry B. Brey dalam bukunya "The Intel Microprocessors: Architecture, Programming, and Interfacing," memberikan definisi arsitektur komputer yang lebih spesifik untuk prosesor Intel, namun konsepnya berlaku untuk sistem komputer secara umum (Andaria 2024; Brey 2009):

Arsitektur Komputer: Menurut Brey, arsitektur komputer adalah "struktur dan perilaku dari sistem komputer sebagaimana dipersepsikan oleh pengguna, termasuk set instruksi, format, tipe data, jumlah register, mode pengalamatan, dan organisasi memori." Brey menekankan pada antarmuka yang dilihat pengguna atau pemrogram saat mereka berinteraksi dengan perangkat keras.

Organisasi Komputer: Brey menjelaskan bahwa organisasi komputer adalah cara komponen fisik seperti CPU, memori, dan perangkat input/output diimplementasikan dan dihubungkan untuk memungkinkan eksekusi instruksi. Brey berfokus pada bagaimana elemen-elemen seperti bus, pengendali memori, dan prosesor bekerja sama untuk mencapai kinerja yang optimal.

B. PERBEDAAN ARSITEKTUR DAN ORGANISASI KOMPUTER

Dalam ilmu komputer, istilah arsitektur komputer dan organisasi komputer sering digunakan untuk menjelaskan dua aspek utama dari desain sistem komputer. Meskipun kedua istilah ini saling terkait, keduanya memiliki makna dan fokus yang berbeda. Pemahaman yang mendalam mengenai perbedaan antara arsitektur dan organisasi komputer penting bagi siapa pun yang ingin memahami bagaimana komputer dirancang dan bekerja.

1. Arsitektur Komputer

Arsitektur komputer merujuk pada konsep-konsep abstrak yang mendefinisikan struktur dan perilaku sistem komputer, terutama yang terlihat oleh pengguna akhir, termasuk programmer dan perangkat lunak. Arsitektur komputer mencakup elemen-elemen penting berikut:

- a. **Set Instruksi:** Kumpulan instruksi dasar yang dapat dieksekusi oleh CPU. Set instruksi ini menentukan jenis operasi yang bisa dilakukan oleh prosesor, misalnya operasi aritmatika, logika, dan kontrol aliran.
 - b. **Modus Pengalamatan:** Cara di mana alamat-alamat memori dan data ditangani selama proses eksekusi instruksi.
-

- c. **Format Data:** Cara pengorganisasian data di dalam memori dan registri, termasuk panjang kata (word length), endianness (big-endian atau little-endian), dan jenis data (integer, floating-point, dll.).
- d. **Arsitektur Memori:** Pengorganisasian dan pengelolaan memori komputer, termasuk cache, RAM, ROM, dan penyimpanan sekunder.
- e. **I/O (Input/Output):** Cara komputer berinteraksi dengan perangkat eksternal, seperti disk, keyboard, monitor, printer, dan jaringan.

Arsitektur komputer berfokus pada bagaimana CPU berinteraksi dengan sistem keseluruhan dan bagaimana desain memengaruhi kinerja. Ini adalah deskripsi logis dari sistem, tanpa mempertimbangkan teknologi fisik yang digunakan.

Contoh Arsitektur Komputer

Salah satu contoh paling terkenal adalah arsitektur Von Neumann, di mana instruksi dan data disimpan dalam memori yang sama dan diproses oleh CPU secara sekuensial (M. Morris Mano and Charles Kime 2014). Arsitektur ini menjadi dasar bagi sebagian besar komputer modern.

2. Organisasi Komputer

Organisasi komputer merujuk pada cara elemen-elemen perangkat keras yang membentuk komputer saling diatur dan diimplementasikan secara fisik. Organisasi komputer berkaitan dengan komponen-komponen aktual, seperti rangkaian listrik, perangkat keras, koneksi, dan teknologi yang digunakan untuk mengimplementasikan arsitektur tersebut. Organisasi komputer berfokus pada aspek-aspek berikut:

- a. **Unit Fungsional:** CPU, ALU (Arithmetic Logic Unit), unit kontrol, register, memori, dan perangkat I/O.
 - b. **Jalur Data:** Desain dan konfigurasi jalur sinyal yang menghubungkan komponen-komponen dalam komputer. Ini termasuk bus sistem, cache, serta kontrol dan pengaturan memori.
 - c. **Mikroarsitektur:** Implementasi fisik dari set instruksi yang telah didefinisikan oleh arsitektur. Misalnya, bagaimana jalur sinyal data dibangun atau bagaimana cache dikelola untuk meningkatkan efisiensi.
-

- d. Kinerja Sistem: Organisasi berpengaruh pada kecepatan pemrosesan data, throughput, dan latency, karena keterkaitannya dengan optimisasi perangkat keras.

Contoh Organisasi Komputer

Contoh umum dari organisasi komputer adalah perbedaan dalam pengelolaan multiprosesor dan multicore. Sistem multiprosesor memiliki beberapa prosesor fisik yang diatur untuk bekerja secara paralel, sedangkan sistem multicore mengimplementasikan beberapa inti pemrosesan dalam satu chip prosesor untuk meningkatkan efisiensi (Hennessy and Patterson 2012).

3. Perbedaan Utama

Perbedaan utama antara arsitektur dan organisasi komputer dapat dirangkum sebagai berikut:

Arsitektur Komputer	Organisasi Komputer
Berfokus pada desain logis dan set instruksi.	Berfokus pada implementasi fisik dari desain.
Berhubungan dengan cara CPU berinteraksi dengan memori, I/O, dan data.	Berhubungan dengan bagaimana elemen-elemen perangkat keras terhubung dan berfungsi.
Lebih berorientasi pada abstraksi dan desain sistem untuk pengguna akhir dan programmer.	Lebih berorientasi pada kinerja, efisiensi, dan teknologi yang digunakan dalam perangkat keras.
Mengacu pada apa yang harus dilakukan sistem.	Mengacu pada bagaimana sistem melakukan tugas tersebut.
Contoh: Set instruksi x86, arsitektur RISC dan CISC.	Contoh: Pengelolaan cache, organisasi bus sistem, penggunaan register.

4. Keterkaitan Arsitektur dan Organisasi Komputer

Meskipun berbeda, arsitektur dan organisasi komputer sangat berkaitan satu sama lain. Perubahan dalam arsitektur dapat mempengaruhi organisasi dan sebaliknya. Sebagai contoh, perubahan set instruksi (yang merupakan bagian dari arsitektur) dapat memaksa perubahan dalam mikroarsitektur dan desain fisik sistem (yang merupakan bagian dari organisasi).

Arsitektur RISC (Reduced Instruction Set Computing) adalah salah satu contoh di mana perubahan pada arsitektur mempengaruhi desain organisasi. Dengan set instruksi yang disederhanakan, arsitektur RISC memungkinkan desain organisasi yang lebih efisien, terutama dalam hal kinerja dan penghematan daya (Patterson and Hennessy 2014).

C. SEJARAH PERKEMBANGAN KOMPUTER

Perkembangan komputer telah mengalami perjalanan yang panjang dan kompleks, dimulai dari alat bantu sederhana hingga sistem canggih yang menggerakkan teknologi modern. Berikut adalah penjelasan lengkap mengenai sejarah perkembangan komputer hingga teknologi terbaru saat ini (Andaria 2024).

1. Era Awal (Abacus hingga Komputer Mekanik)

Perkembangan komputer pertama kali dapat ditelusuri kembali ke alat bantu hitung seperti abacus, yang digunakan sejak ribuan tahun yang lalu. Abacus terdiri dari batang dan manik-manik yang memudahkan perhitungan dasar. Pada abad ke-17, Blaise Pascal menciptakan Pascaline, kalkulator mekanik pertama yang dapat melakukan operasi penjumlahan dan pengurangan (Ceruzzi, 2003).

KEGIATAN BELAJAR II

EVOLUSI DAN KINERJA KOMPUTER

Oleh: Bahtera Alam Wijaksono, M.Kom

DESKRIPSI PEMBELAJARAN

Pada bab ini mahasiswa mempelajari tentang awal mulainya Sejarah dari teknologi komputer, perkembangan per generasi, perancangan dan desain kinerja komputer, diharapkan mahasiswa memiliki wawasan dan pemahaman untuk sebagai modal dasar mempelajari matakuliah organisasi dan arsitektur komputer.

KOMPETENSI PEMBELAJARAN

1. Setelah mengikuti materi matakuliah ini diharapkan mahasiswa dan mahasiswi memiliki wawasan dan pemahaman :
2. Mampu menguraikan definisi sejarah perkembangan dari perkembangan generasi komputer.
3. Mampu menjelaskan kinerja dari sistem komputer.
4. Mampu menjelaskan komponen komputer yang terlibat.

A. SEJARAH PERKEMBANGAN GENERASI KOMPUTER

Pada bab ini, akan membahas bahas sejarah dari awal dari teknologi komputer yang dibagi menjadi beberapa - beberapa generasi, karena beberapa di antaranya merupakan awal dari perkembangan komputer saat ini beredar di masyarakat.

1. Komputer Primitif

Pada generasi belum adanya istilah komputer yang sebenarnya masih belum ditemukan maka dipada judul disebutkan sebagai komputer primitif, namun pada generasi ini merupakan awal mulanya teknologi komputer digital yang kita kenal saat ini. Pada generasi ini, Komputer primitif pada generasi ini masih sepenuhnya menggunakan mesin mekanis.

Pascaline pertama kali diciptakan oleh ilmuwan Perancis yang bernama Blaise Pascal, pascaline diciptakan saat Blaise Pascal berusia masih berusia sekitar 17 tahun pada tahun 1642.

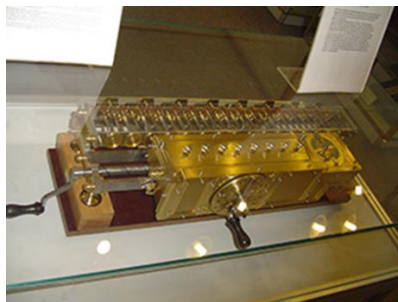
Pascaline adalah mesin hitung pertama yang diberi nama Kalkulator Pascaline hanya dapat melakukan operasi sederhana berupa penambahan dan pengurangan. Ayah dari blaise pascal mempunyai kendala dalam perhitungan aritmetika yang melelahkan diperkerjaan dengan kemampuannya dia berusaha membantu dengan mengembangkan kalkulator pascal.



Gambar 2.1: Kalkulator Pascal pada tahun 1652

Sumber: Website <https://p2k.stekom.ac.id>

Pada tahun 1671, ilmuwan Jerman ahli matematika-filsuf pada tahun 1673 Gottfried Wilhelm von Leibniz merancang mesin penghitung yang disebut Step Reckoner. Step Reckoner mengembakan ide yang dilakukan Pascal dengan menambahkan perkalian dengan penjumlahan dan pergeseran berulang-ulang.



Gambar 2.2 : Kalkulator Leibniz pada tahun 1673

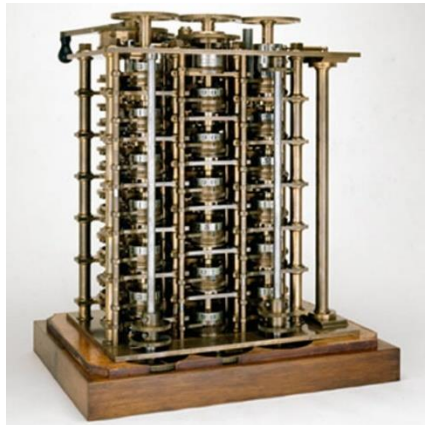
Sumber: Website www.britannica.com

Pada tahun 1791-1871, ilmuwan Cambridge Professor Matematika dari Universitas Cambridge, Charles Babbage, merancang sebuah mesin yang dikenal dengan nama Difference Engine pada tahun 1823. Mesin ini masih menggunakan komponen mekanik sederhana dan dapat menghasilkan tabel matematis.



Gambar 2.3: Charles Babbage (*Brock, 2021*)

Sepuluh tahun setelah rancangan mesin pertama, yaitu tahun 1833, Babbage mengembangkan konsep mesin baru yang dikenal sebagai Analytic Engine. Mesin ini dapat dianggap sebagai konsep awal dari computer. (Abdul Kadir, 2013).



Gambar 2.4: Difference Engines, and Analytical Engines
Sumber: Website <https://www.computerhistory.org>

2. Generasi Pertama: Tabung Vakum (1945-1955)

Pada satu dekade berikutnya, terjadi peralihan menuju awal generasi komputer pertama. Perkembangan komputer pada masa itu tidak hanya digunakan untuk tujuan penelitian, tetapi juga memiliki peran penting dalam berbagai aspek selama Perang Dunia II.

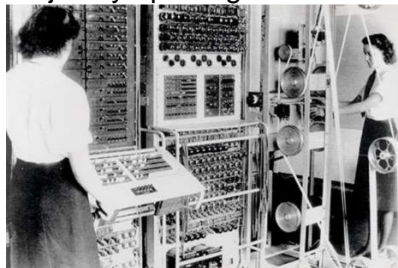
Pada tahun 1930 ada seorang ilmuwan yang merancang mesin Linear Equation Solver yang bernama John Atanasoff dari universitas Iowa. Mesin yang dikembangkan memiliki kemampuan untuk menyelesaikan persamaan linear dan integral diferensial. Pada tahun 1937 sampai dengan 1942 komputer Atanasoff-Berry yang pertama kali dapat mengimplementasikan persamaan linear.

Pada 1944 Jerman Konrad Zuse sebagai pelopor bahasa pemrograman algoritmik pertama kedalam komputer yang disebut dengan Plankalkul (plan Calculus). Konrad Zuse mengembangkan komputasi digital biner pertama Z1 sampai dengan Z3 dengan menggunakan pita kertas berlubang yang dapat berfungsi dengan baik yang digunakan pada perang dunia ke II.



Gambar 2.5: Konrad Zuse (Brock, 2021)

Sedangkan di Inggris, Tommy Flowers adalah seorang insinyur listrik yang merancang sebuah komputer Colossus sebagai perintis yang digunakan untuk menguraikan kode - kode Jerman di Bletchley Park selama terjadinya perang dunia II.



Gambar 2.6: The Colossus Mark II code-breaking computer in 1943 (Heritage, 2023)

Pada saat yang hampir bersamaan, Professor Howard Aiken's bekerjasama dengan Thomas J. Watson Sr sebagai Presiden dari IBM memperkenalkan ke Harvard University Mark I,

yang juga dikenal sebagai IBM ASCC (Automatic Sequence Controlled Calculator). Harvard Mark I merupakan komputer digital otomatis pertama di Amerika (Abdul Kadir, 2013).



Gambar 2.7: Automatic Sequence Controlled Calculator (ASCC)
Sumber: Website <https://www.computerhistory.org>

Pada tahun 1943 Profesor teknik elektro dari University of Pennsylvania John Mauchly beserta mahasiswanya John Eckert mengusulkan untuk membangun satu komputer general - purpose menggunakan vacuum tubes yang menjadikan sebagai komputer generasi pertama ini adalah ENIAC (Electronic Numerical Integrator And Computer). Proyek ENIAC direncanakan sebagai alat pembuatan jarak dan tabel lintasan peluru kendali senjata baru dengan gagasan tersebut diterima oleh angkatan darat pemerintah setempat. Komputer generasi pertama memiliki bobot 30 ton dengan luas 1500 meter persegi dan terdiri atas lebih dari 18.000 vacuum tubes. Proyek ENIAC berhasil diselesaikan pada tahun 1946 sehingga tidak dapat dimanfaatkan pada perang dunia ke dua, namun mesin ini masih digunakan sampai tahun 1955.

Komputer ENIAC mampu berkerja dengan menggunakan bilangan desimal, tidak seperti pendahulunya yang masih

KEGIATAN BELAJAR III

SISTEM PERANGKAT LUNAK

Oleh: Miswadi, S.Kom., M.Kom

DESKRIPSI PEMBELAJARAN

Pada bab ini mahasiswa mempelajari pengertian sistem perangkat lunak, klasifikasi, arsitektur, siklus hidup, dan tantangan dalam sistem perangkat lunak, serta bagaimana teknologi dan konsep terkini berperan dalam masa depan pengembangan perangkat lunak. Diharapkan mahasiswa memiliki wawasan dan pemahaman untuk dapat mengenali dan memahami berbagai tren modern seperti AI, *cloud computing*, dan IoT yang sedang berkembang dalam dunia perangkat lunak.

KOMPETENSI PEMBELAJARAN

Setelah mengikuti perkuliahan ini diharapkan mahasiswa dan mahasiswi memiliki pengetahuan dan kemampuan :

1. Mampu memahami definisi dan peran sistem, sejarah dan evolusi, tren modern dalam pengembangan perangkat lunak.
2. Mampu memahami klasifikasi, karakteristik, dan menerapkan dalam studi kasus perangkat lunak.
3. Mampu memahami konsep, mengidentifikasi dan menerapkan pola dan menganalisis pendekatan pengembangan arsitektur system perangkat lunak.
4. Mampu memahami model pengembangan, mengidentifikasi siklus hidup, mengelola pengembangan perangkat lunak.
5. Mampu memahami prinsip keamanan, menerapkan Teknik keamanan dan mengelola keamanan system perangkat lunak.
6. Mampu memahami konsep ujicoba dan verifikasi, menerapkan teknik ujicoba dan melakukan verifikasi dan validasi pada sistem perangkat lunak,

A. PENDAHULUAN

1. Definisi Sistem Perangkat Lunak

Sistem perangkat lunak merujuk pada kumpulan program, prosedur, dan dokumentasi terkait yang mengelola dan mengendalikan operasi komputer. Perangkat lunak ini menjadi komponen esensial dalam ekosistem komputer karena memungkinkan perangkat keras untuk berfungsi sesuai dengan tujuan yang diinginkan. Dalam konteks yang lebih luas, perangkat lunak adalah jembatan antara manusia dan mesin, yang memungkinkan pengguna untuk berinteraksi dengan perangkat keras melalui antarmuka yang intuitif dan fungsional khusus dan dasar seperti algoritma, logika dan penggunaan bahasa pemrograman itu sendiri.

2. Peran Sistem Perangkat Lunak dalam Arsitektur Komputer

Arsitektur komputer adalah struktur konseptual dan fungsional dari sistem komputer, yang mencakup semua aspek dari perangkat keras hingga perangkat lunak. Dalam arsitektur komputer, perangkat lunak memainkan peran krusial sebagai penghubung antara komponen perangkat keras yang berbeda. Perangkat lunak mengatur operasi dasar seperti manajemen memori, pemrosesan data, komunikasi antar komponen, dan interaksi pengguna. Sebagai contoh, sistem operasi (sebagai perangkat lunak inti) bertindak sebagai perantara antara aplikasi yang dijalankan oleh pengguna dan perangkat keras yang mendasarinya. Sistem operasi mengatur sumber daya perangkat keras, seperti CPU, memori, dan perangkat input/output, serta menyediakan antarmuka pengguna yang memungkinkan interaksi yang efisien dan efektif dengan komputer.

3. Evolusi dan Tren Terkini dalam Sistem Perangkat Lunak

Sejarah perkembangan perangkat lunak mencerminkan evolusi teknologi komputer secara keseluruhan. Dari kode mesin yang sederhana di era komputer generasi pertama hingga perangkat lunak kompleks berbasis jaringan dan cloud pada era modern, evolusi ini dipengaruhi oleh kebutuhan pengguna dan perkembangan teknologi perangkat keras.

Saat ini, tren dalam pengembangan sistem perangkat lunak mencakup berbagai teknologi canggih seperti:

1. **Kecerdasan Buatan (AI) dan Pembelajaran Mesin (Machine Learning)**: Digunakan untuk mengotomatisasi proses, meningkatkan efisiensi, dan menciptakan solusi yang adaptif.
2. **Komputasi Awan (Cloud Computing)**: Memungkinkan distribusi perangkat lunak sebagai layanan (SaaS) yang dapat diakses di mana saja dan kapan saja.
3. **Keamanan Siber (Cybersecurity)**: Menjadi fokus utama mengingat meningkatnya ancaman terhadap data dan infrastruktur digital.
4. **Internet of Things (IoT)**: Mengintegrasikan perangkat fisik dengan sistem perangkat lunak untuk menciptakan ekosistem yang terhubung.

Pada bab ini, akan dibahas lebih lanjut mengenai klasifikasi, arsitektur, siklus hidup, dan tantangan dalam sistem perangkat lunak, serta bagaimana teknologi dan konsep terkini berperan dalam masa depan pengembangan perangkat lunak.

B. KLASIFIKASI SISTEM PERANGKAT LUNAK

1. Perangkat Lunak Sistem

a. Sistem Operasi

- **Definisi dan Fungsi**: Sistem operasi (Operating System atau OS) adalah perangkat lunak yang berfungsi sebagai jembatan antara pengguna dan perangkat keras komputer. Ia mengatur sumber daya komputer seperti CPU, memori, dan perangkat input/output, serta menyediakan lingkungan bagi aplikasi untuk dijalankan.
- **Contoh**: Microsoft Windows, macOS, Linux, dan Android.
- **Komponen Utama**: Kernel (mengelola operasi dasar sistem), User Interface (antarmuka pengguna, baik GUI maupun CLI), dan berbagai layanan sistem (seperti manajemen file dan memori).

b. Driver Perangkat

- **Definisi dan Fungsi**: Driver perangkat adalah program kecil yang memungkinkan sistem operasi untuk

berkomunikasi dengan perangkat keras tertentu. Setiap perangkat keras memerlukan driver yang sesuai agar dapat berfungsi dengan benar dalam sistem komputer.

- **Contoh:** Driver untuk printer, kartu grafis, dan perangkat jaringan.
- **Cara Kerja:** Driver bertindak sebagai penerjemah antara perangkat keras dan sistem operasi, mengonversi perintah sistem operasi menjadi tindakan yang dapat dimengerti oleh perangkat keras.

2. Perangkat Lunak Aplikasi

a. Aplikasi Umum

- **Definisi dan Fungsi:** Aplikasi umum adalah perangkat lunak yang dirancang untuk memenuhi kebutuhan pengguna sehari-hari di berbagai bidang. Ini mencakup perangkat lunak yang sering digunakan untuk tugas-tugas umum seperti pengolahan kata, presentasi, dan pengelolaan data.
- **Contoh:** Microsoft Office (Word, Excel, PowerPoint), Google Docs, dan Adobe Photoshop.
- **Karakteristik:** Mudah digunakan oleh berbagai kalangan, memiliki antarmuka pengguna yang ramah, dan umumnya tersedia di berbagai platform.

c. Aplikasi Khusus

1. **Definisi dan Fungsi:** Aplikasi khusus adalah perangkat lunak yang dirancang untuk tugas-tugas spesifik atau industri tertentu. Ini biasanya digunakan dalam lingkungan profesional yang membutuhkan fitur dan kemampuan khusus.
2. **Contoh:** Software untuk CAD (Computer-Aided Design) seperti AutoCAD, software untuk manajemen basis data seperti Oracle Database, dan software untuk akuntansi seperti QuickBooks.
3. **Penggunaan:** Digunakan oleh profesional dalam bidang tertentu seperti arsitek, insinyur, dokter, dan akuntan untuk memenuhi kebutuhan spesifik mereka.

d. Perangkat Lunak Embedded

Definisi dan Fungsi

- **Definisi:** Perangkat lunak embedded adalah perangkat lunak yang dirancang untuk menjalankan tugas-tugas tertentu di dalam perangkat keras yang biasanya memiliki fungsi khusus. Perangkat lunak ini diintegrasikan ke dalam sistem yang lebih besar dan sering kali tidak terlihat oleh pengguna.
- **Contoh:** Perangkat lunak dalam mesin cuci, perangkat medis, sistem navigasi GPS, dan perangkat IoT (Internet of Things).
- **Fungsi:** Mengontrol dan mengelola operasi perangkat keras tertentu, sering kali dengan sumber daya komputasi yang terbatas dan kebutuhan real-time.

Karakteristik Utama

- **Real-Time Operation:** Banyak sistem embedded memerlukan operasi waktu nyata di mana respons cepat terhadap input sangat penting.
- **Resource Constraints:** Biasanya dirancang untuk berjalan di lingkungan dengan sumber daya terbatas, seperti memori kecil dan prosesor dengan daya rendah.
- **Reliability and Safety:** Harus sangat andal dan sering kali mematuhi standar keselamatan yang ketat, terutama dalam aplikasi kritis seperti perangkat medis atau sistem otomotif.

e. Middleware

Definisi dan Fungsi

- **Definisi:** Middleware adalah perangkat lunak yang menghubungkan aplikasi dan layanan perangkat lunak lainnya, memungkinkan komunikasi dan manajemen data yang mulus di lingkungan yang heterogen. Ini bertindak sebagai "lapisan tengah" yang memfasilitasi interoperabilitas antar aplikasi yang berbeda.
- **Contoh:** Message-oriented middleware (MOM), Object Request Brokers (ORB), dan database middleware.
- **Fungsi:** Menyediakan layanan umum seperti komunikasi antar proses, autentikasi, pengelolaan transaksi, dan konektivitas ke basis data. Middleware memungkinkan aplikasi untuk bekerja sama di berbagai platform dan jaringan.

Peran dalam Arsitektur Perangkat Lunak

- **Interoperabilitas:** Memungkinkan integrasi sistem yang kompleks dengan komponen yang beragam.
- **Abstraksi:** Menyediakan lapisan abstraksi yang menyembunyikan kompleksitas jaringan dan komunikasi antar proses, sehingga memudahkan pengembang untuk fokus pada logika bisnis utama.
- **Manajemen Data dan Transaksi:** Middleware juga dapat mengelola transaksi kompleks dan memastikan integritas data di berbagai sistem.

C. ARSITEKTUR PERANGKAT LUNAK

Definisi dan Komponen Utama Arsitektur Perangkat Lunak

1. Definisi Arsitektur Perangkat Lunak

- **Arsitektur perangkat lunak** adalah struktur tinggi dari suatu sistem perangkat lunak, yang melibatkan organisasi komponen perangkat lunak dan hubungan antar komponen tersebut. Arsitektur perangkat lunak menentukan bagaimana elemen-elemen sistem diatur dan berinteraksi untuk memenuhi kebutuhan fungsional dan non-fungsional dari sistem.
- **Pentingnya Arsitektur:** Arsitektur perangkat lunak sangat penting karena berfungsi sebagai cetak biru untuk desain dan implementasi perangkat lunak. Ia mengatur bagaimana komponen sistem diintegrasikan dan memastikan bahwa sistem dapat diukur, dipelihara, dan diandalkan.

2. Komponen Utama dalam Arsitektur Perangkat Lunak

- **Modul:** Bagian dari perangkat lunak yang dapat diidentifikasi secara independen, biasanya dengan tanggung jawab tertentu seperti manajemen data, logika bisnis, atau interaksi pengguna.
- **Komponen:** Unit perangkat lunak yang dapat dipisahkan dan digunakan kembali, terdiri dari beberapa modul yang bekerja bersama-sama untuk menyediakan fungsi spesifik.

- **Konektor:** Mekanisme yang memungkinkan interaksi antar komponen, termasuk panggilan prosedur, pesan, dan koneksi data.
- **Layer (Lapisan):** Pengorganisasian komponen perangkat lunak dalam lapisan-lapisan dengan tanggung jawab tertentu, seperti lapisan presentasi, logika bisnis, dan data.

3. Pola Arsitektur Perangkat Lunak

a. Layered Architecture (Arsitektur Berlapis)

- **Deskripsi:** Arsitektur ini membagi sistem menjadi beberapa lapisan, di mana masing-masing lapisan hanya berinteraksi dengan lapisan yang berdekatan. Ini menciptakan isolasi yang kuat antara tanggung jawab sistem, membuat sistem lebih mudah dipelihara dan diperluas.
- **Contoh Penggunaan:** Umumnya digunakan dalam aplikasi bisnis, di mana ada pemisahan yang jelas antara antarmuka pengguna, logika bisnis, dan data.
- **Keuntungan:** Mudah untuk memahami dan memelihara. Setiap lapisan dapat diuji dan dikembangkan secara independen.
- **Tantangan:** Mungkin terjadi overhead komunikasi antar lapisan yang dapat mempengaruhi kinerja.

b. Client-Server Architecture

- **Deskripsi:** Dalam pola ini, sistem dibagi menjadi dua komponen utama: client yang mengirim permintaan dan server yang merespons dengan menyediakan layanan atau data. Ini sering digunakan dalam aplikasi jaringan, di mana klien dan server beroperasi di mesin yang berbeda.
- **Contoh Penggunaan:** Aplikasi web di mana browser (client) berkomunikasi dengan server web untuk mendapatkan konten.
- **Keuntungan:** Skalabilitas yang baik, dapat mendukung banyak klien dengan server yang sama.
- **Tantangan:** Keamanan dan pengelolaan koneksi jaringan dapat menjadi rumit.

c. Microservices Architecture

- **Deskripsi:** Arsitektur ini memecah sistem menjadi layanan-layanan kecil dan independen yang

berkomunikasi melalui antarmuka API. Setiap layanan bertanggung jawab atas fungsionalitas bisnis tertentu dan dapat dikembangkan, dikerahkan, dan diukur secara independen.

- **Contoh Penggunaan:** Sistem besar seperti platform e-commerce yang memerlukan pengembangan dan penyebaran fitur yang cepat dan mandiri.
- **Keuntungan:** Fleksibilitas yang tinggi dalam pengembangan dan penyebaran. Setiap layanan dapat menggunakan teknologi yang paling sesuai dengan tugasnya.
- **Tantangan:** Kompleksitas pengelolaan sistem secara keseluruhan meningkat, termasuk pengelolaan komunikasi antar layanan.

4. Pendekatan Pengembangan Arsitektur Perangkat Lunak

a. Model Pengembangan Arsitektur

- **Model Monolitik:** Pendekatan tradisional di mana semua komponen sistem dikembangkan dan dikerahkan sebagai satu unit besar. Meskipun sederhana, model ini kurang fleksibel dan sulit untuk diukur.
- **Model Service-Oriented Architecture (SOA):** Pendekatan yang memanfaatkan layanan-layanan mandiri yang dapat digunakan ulang dan diintegrasikan melalui middleware untuk menciptakan sistem yang lebih fleksibel dan modular.
- **Model Event-Driven Architecture:** Pendekatan ini menekankan pada pemrosesan dan komunikasi berbasis peristiwa (events), yang memungkinkan sistem untuk merespons perubahan atau tindakan secara real-time.

b. Langkah-Langkah Pengembangan Arsitektur

- **Analisis Kebutuhan:** Memahami dan mendefinisikan kebutuhan fungsional dan non-fungsional dari sistem.
- **Desain Tingkat Tinggi:** Mengidentifikasi komponen utama, pola arsitektur yang sesuai, dan hubungan antar komponen.
- **Validasi dan Verifikasi:** Memastikan bahwa arsitektur yang dirancang memenuhi kebutuhan dan kriteria desain, baik melalui simulasi, prototipe, atau review.

- **Implementasi:** Mengembangkan sistem berdasarkan desain arsitektur yang telah divalidasi.
- **Pemeliharaan:** Mengelola dan memelihara sistem seiring dengan perubahan kebutuhan bisnis dan teknologi.

5. Contoh Arsitektur Perangkat Lunak dalam Sistem Nyata

a. Arsitektur Perangkat Lunak di Aplikasi Web

- **Deskripsi:** Aplikasi web modern biasanya menggunakan arsitektur *microservices* yang memisahkan fungsi-fungsi seperti otentikasi, pembayaran, dan manajemen produk ke dalam layanan-layanan independen.
- **Contoh:** Amazon dan Netflix, yang menggunakan *microservices* untuk mendukung skala besar dan fleksibilitas dalam penyebaran fitur baru.
- **Pola yang Digunakan:** *Microservices* dan *client-server*.

b. Arsitektur Perangkat Lunak di Sistem Terdistribusi

- **Deskripsi:** Sistem terdistribusi seperti sistem basis data terdistribusi atau aplikasi IoT menggunakan arsitektur berbasis *event-driven* untuk memastikan data dan operasi yang tersebar di berbagai lokasi dapat tetap sinkron dan konsisten.
- **Contoh:** Sistem manajemen basis data *NoSQL* seperti *Cassandra*, yang menggunakan arsitektur terdistribusi untuk menangani data dalam skala besar dengan ketersediaan tinggi.
- **Pola yang Digunakan:** *Event-driven architecture* dan *SOA*.

D. SIKLUS HIDUP PENGEMBANGAN PERANGKAT LUNAK

1. Model Pengembangan Perangkat Lunak

a. Model Waterfall

- **Deskripsi:** Model *Waterfall* adalah pendekatan linier dalam pengembangan perangkat lunak di mana prosesnya dibagi menjadi tahap-tahap yang jelas: analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Setiap tahap harus diselesaikan sebelum tahap berikutnya dimulai.

- Kelebihan: Sederhana dan mudah dipahami. Cocok untuk proyek dengan persyaratan yang jelas dan tidak berubah.
 - Kekurangan: Kurang fleksibel dalam menangani perubahan kebutuhan selama proses pengembangan. Risiko tinggi jika persyaratan tidak tepat di awal.
 - Contoh Penggunaan: Proyek dengan kebutuhan yang stabil dan jelas, seperti aplikasi perangkat lunak sistem lama.
- b. Model Agile
- Deskripsi: Model Agile adalah pendekatan iteratif dan inkremental yang menekankan kolaborasi tim, adaptasi cepat terhadap perubahan, dan pengiriman perangkat lunak secara bertahap dalam siklus pendek yang disebut sprint. Pendekatan ini menggunakan prinsip-prinsip seperti Scrum atau Kanban untuk mengelola proses pengembangan.
 - Kelebihan: Fleksibel dan responsif terhadap perubahan. Memungkinkan umpan balik terus-menerus dari pengguna dan pemangku kepentingan.
 - Kekurangan: Bisa menjadi sulit untuk mengelola proyek besar dengan tim yang besar jika tidak diatur dengan baik.
 - Contoh Penggunaan: Proyek yang memerlukan fleksibilitas dan adaptasi cepat, seperti pengembangan aplikasi web dan perangkat lunak mobile.
- c. Model DevOps
- Deskripsi: Model DevOps mengintegrasikan pengembangan perangkat lunak (Development) dengan

KEGIATAN BELAJAR IV

PEMROSESAN DATA

Oleh: Eka Prasetya Adhy Sugara, S.T., M.Kom.

DESKRIPSI PEMBELAJARAN

Pada bab ini mahasiswa mempelajari tentang definisi dan konsep dasar pemrosesan data dalam lingkup organisasi dan arsitektur komputer. Pembahasan dalam bab ini meliputi Data dan Pemrosesan Data, Siklus Pemrosesan Data, Jenis-jenis Pemrosesan Data, Komponen Utama Pemrosesan Data, Teknik dalam Pemrosesan Data dan Teknologi Baru dalam Pemrosesan Data.

KOMPETENSI PEMBELAJARAN

Setelah mengikuti perkuliahan ini diharapkan mahasiswa memiliki pengetahuan dan kemampuan antara lain:

1. Mampu memahami dan menjelaskan definisi data dan pemrosesan data
2. Mampu memahami dan menjelaskan siklus pemrosesan data
3. Mampu memahami dan menjelaskan jenis-jenis pemrosesan data
4. Mampu memahami dan menjelaskan komponen utama dalam pemrosesan data
5. Mampu memahami dan menjelaskan berbagai teknologi dalam pemrosesan data
6. Mampu menganalisis dan membuat sebuah artikel ilmiah mengenai implementasi pemrosesan data

A. DATA DAN PEMROSESAN DATA

Data adalah suatu penggambaran fakta, pengertian instruksi yang dapat disampaikan dan diolah oleh manusia atau mesin. Contoh data berupa angka, karakter, alfabet, simbol, gambar, suara dan lain-lain (Sriani, 2020). Pemrosesan data

merupakan salah satu fungsi utama dari sebuah sistem komputer yang merujuk pada serangkaian langkah yang dilakukan untuk mengumpulkan, menyimpan, dan mengolah data menjadi informasi yang berguna (Ranti, 2023). Proses ini melibatkan beberapa fungsi dasar, seperti:

1. Pengambilan Data, proses mengambil data dari sumber input.
2. Penyimpanan Data, proses menyimpan data untuk pemrosesan lebih lanjut.
3. Pemrosesan, melakukan operasi aritmetika dan logika pada data yang disimpan.
4. Penyimpanan hasil, menyimpan hasil sementara dan akhir dari pemrosesan.
5. Output, menampilkan atau mencetak hasil pemrosesan data

Dalam konteks organisasi dan arsitektur komputer, pemrosesan data melibatkan berbagai komponen sistem yang bekerja sama untuk mengelola aliran data. Organisasi komputer mencakup struktur dan cara kerja dari berbagai komponen seperti CPU, memori, dan perangkat input/output (I/O) (Nurchaliza, 2024).

B. SIKLUS PEMROSESAN DATA

Siklus pemrosesan data adalah serangkaian langkah yang dilalui data dari pengumpulan, penggunaan hingga penyimpanan. Dalam konteks organisasi dan arsitektur komputer, siklus ini sangat penting karena mempengaruhi bagaimana data dikelola dan diolah untuk menghasilkan informasi yang berguna.



Gambar 4.1 Siklus Pemrosesan Data

Berikut adalah uraian mengenai tahapan siklus pemrosesan data (Saputra et al., 2018).

1. Pengumpulan Data (*Collection*)

Data dikumpulkan dari berbagai sumber, seperti survei, sensor, atau transaksi. Pengumpulan data yang akurat dan relevan adalah langkah awal yang krusial dalam siklus ini.

2. Persiapan Data (*Preparation*)

Setelah data dikumpulkan, tahap ini melibatkan pembersihan dan pengorganisasian data untuk menghilangkan kesalahan atau duplikasi. Data yang bersih dan terstruktur siap untuk diproses lebih lanjut.

3. Input Data

Data yang telah dipersiapkan dimasukkan ke dalam sistem pemrosesan. Proses input memastikan bahwa data tersedia dalam format yang dapat diproses oleh sistem komputer.

4. Pemrosesan Data (*Data Processing*)

Ini adalah tahap inti di mana data diolah menggunakan metode tertentu untuk menghasilkan informasi. Pemrosesan dapat melibatkan analisis statistik, pemodelan, atau pengolahan data lainnya.

5. Output Data (*Output*)

Hasil dari pemrosesan data disajikan dalam bentuk informasi yang dapat digunakan, seperti laporan, grafik, atau dashboard. Output ini harus relevan dan dapat ditindaklanjuti oleh pengguna.

6. Penyimpanan dan Distribusi (*Storage and Distribution*)

Informasi yang dihasilkan disimpan untuk digunakan di masa mendatang dan didistribusikan kepada pihak-pihak yang membutuhkan. Penyimpanan yang efisien memungkinkan akses cepat ke data saat diperlukan.

C. JENIS-JENIS PEMROSESAN DATA

Pemrosesan data dalam lingkup organisasi dan arsitektur komputer dapat dibedakan menjadi beberapa jenis, yang masing-masing memiliki karakteristik dan tujuan tertentu. Berikut adalah penjelasan mengenai jenis-jenis pemrosesan data yang umum digunakan:

1. Pemrosesan Batch (*Batch Processing*)

Data dikumpulkan dan diproses dalam kelompok pada waktu tertentu. Metode ini efektif untuk memproses volume data yang besar tanpa memerlukan interaksi langsung dari pengguna. Contoh umum termasuk pemrosesan gaji dan laporan bulanan.

2. Pemrosesan Terdistribusi (*Distributed Processing*)

Pemrosesan data dilakukan di beberapa lokasi atau sistem yang berbeda, memungkinkan penggunaan banyak CPU untuk mengelola beban kerja yang besar. Ini meningkatkan efisiensi dan kecepatan pemrosesan data, terutama untuk aplikasi yang memerlukan kapasitas tinggi

3. Pemrosesan Terdistribusi (*Distributed Processing*)

Pemrosesan data dilakukan di beberapa lokasi atau sistem yang berbeda, memungkinkan penggunaan banyak CPU untuk mengelola beban kerja yang besar. Ini meningkatkan efisiensi dan kecepatan pemrosesan data, terutama untuk aplikasi yang memerlukan kapasitas tinggi

4. Parallel Processing

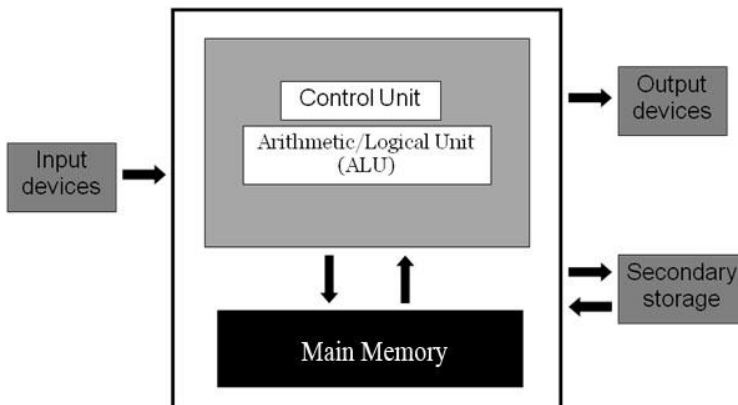
Pemrosesan data dilakukan secara bersamaan dengan memanfaatkan beberapa prosesor untuk meningkatkan efisiensi dan kecepatan.

D. KOMPONEN UTAMA PEMROSESAN DATA

Komponen utama pemrosesan data terdiri dari berbagai elemen yang bekerja sama untuk mengelola, mengolah, dan menyimpan data. Berikut adalah penjelasan mengenai komponen-komponen utama dalam pemrosesan data:

1. *Central Processing Unit (CPU)*

CPU adalah komponen utama dalam arsitektur komputer yang bertanggung jawab untuk menjalankan instruksi dan memproses data. CPU melakukan operasi aritmetika dan logika, serta mengontrol aliran data di dalam sistem. Dalam konteks pemrosesan data, CPU berfungsi untuk mengambil data dari memori, memprosesnya, dan kemudian menyimpan hasilnya kembali ke memori atau mengeluarkannya sebagai output. CPU terdiri dari beberapa bagian penting, yaitu (Mufida et al., 2021):



Gambar 4.2. Bagian-bagian CPU

- a. *Control Unit* (CU): Mengontrol aliran instruksi dan memastikan bahwa semua komponen berfungsi sesuai urutan yang benar.
- b. *Arithmetic Logic Unit* (ALU): Melaksanakan operasi aritmetika dan logika, seperti penjumlahan, pengurangan, dan operasi logika lainnya.
- c. *Memory Unit* (MU): Menyimpan data dan instruksi yang diperlukan untuk diproses oleh CPU

2. *Random Access Memory* (RAM)

Memori utama atau RAM, berfungsi sebagai tempat penyimpanan sementara untuk data yang sedang diproses. RAM memungkinkan CPU untuk mengakses data dengan cepat selama pemrosesan (Farah et al., 2023). Data yang tidak

aktif disimpan di memori sekunder, sementara data yang aktif dan sedang diproses disimpan di RAM, sehingga mempercepat kinerja sistem.

3. Perangkat Penyimpanan (*Storage Devices*)

Perangkat penyimpanan seperti *hard disk drive* (HDD) dan *solid-state drive* (SSD), digunakan untuk menyimpan data secara permanen (Farizy & Harianja, 2020). Data yang tidak sedang diproses disimpan di perangkat ini untuk diakses di masa mendatang. Penyimpanan yang efisien sangat penting untuk menjaga integritas data dan memastikan bahwa informasi dapat diambil kapan saja diperlukan.

4. Sistem Input/Output (I/O)

Sistem I/O mencakup perangkat keras yang digunakan untuk menerima input dari pengguna atau sumber data lainnya dan untuk menampilkan output hasil pemrosesan (Alifah et al., 2023). Contoh perangkat input termasuk keyboard, mouse, dan scanner, sementara perangkat output termasuk monitor, printer, dan speaker. Sistem I/O memainkan peran penting dalam interaksi antara pengguna dan komputer, serta dalam transfer data antara komputer dan perangkat eksternal.

5. Bus Data

Bus data adalah jalur komunikasi yang menghubungkan berbagai komponen dalam sistem komputer (P. A. Maulana et al., 2023). Bus ini memungkinkan transfer data antara CPU, memori, dan perangkat I/O. Kecepatan dan lebar bus data dapat mempengaruhi kinerja keseluruhan sistem.

6. Sistem Operasi (*Operating System*)

Sistem operasi adalah perangkat lunak yang mengelola perangkat keras dan perangkat lunak dalam komputer (M. S. Maulana, n.d.). Ini bertindak sebagai perantara antara pengguna dan perangkat keras, mengatur sumber daya sistem, dan menyediakan antarmuka untuk menjalankan aplikasi.

7. Perangkat Lunak Aplikasi (*Application Software*)

Perangkat lunak aplikasi adalah program yang digunakan untuk melakukan tugas tertentu, seperti pengolahan kata, spreadsheet, dan database (Budi Santoso, 2021). Aplikasi ini bergantung pada sistem operasi dan perangkat keras untuk berfungsi dengan baik.

Komponen utama dalam pemrosesan data saling berinteraksi untuk memastikan bahwa data dapat dikelola dan diproses secara efisien. Memahami fungsi dan hubungan antar komponen ini sangat penting untuk merancang sistem komputer yang efektif dan dapat diandalkan dalam pengolahan data.

E. TEKNIK PEMROSESAN DATA

Teknik pemrosesan data meliputi berbagai metode dan prinsip yang memungkinkan sistem komputer untuk memproses dan mengelola data secara efisien. Berikut adalah beberapa teknik pemrosesan data dalam lingkup organisasi dan arsitektur komputer:

1. *Pipelining*

Pipelining adalah teknik implementasi di mana berbagai instruksi dapat dilaksanakan secara tumpang tindih (*overlapped*) ((Widharma, 2020). Dengan cara ini, setiap instruksi dibagi menjadi beberapa tahap, dan setiap tahap dapat diproses secara bersamaan. Ini mirip dengan jalur perakitan di mana satu bagian dari produk dapat dirakit sementara bagian lain sedang dikerjakan.

Umumnya, *pipelining* dalam pemrosesan instruksi melibatkan beberapa tahap, yang dapat bervariasi tergantung pada arsitektur prosesor. Namun, tahapan umum yang sering digunakan adalah (Abdurohman, 2016):

- a. *Instruction Fetch* (IF), mengambil instruksi dari memori.
 - b. *Instruction Decode* (ID), menginterpretasikan instruksi yang diambil.
 - c. *Execute* (EX), melaksanakan operasi yang ditentukan oleh instruksi.
 - d. *Memory Access* (MEM), mengakses memori jika diperlukan (untuk instruksi yang melibatkan data).
-

e. *Write Back* (WB), menyimpan hasil kembali ke register.

Teknik *Pipeline* memiliki beberapa keuntungan yaitu (1) Peningkatan throughput, *pipelining* memungkinkan lebih banyak instruksi untuk diselesaikan dalam waktu yang lebih singkat, meningkatkan throughput sistem (Zulfikar, 2022). Ini berarti lebih banyak instruksi dapat dieksekusi dalam interval waktu tertentu, yang sangat penting untuk kinerja komputer; (2) Penggunaan sumber daya yang efisien, dengan memanfaatkan waktu idle dari unit pemroses, *pipelining* memastikan bahwa semua bagian dari prosesor digunakan secara optimal. Ini mengurangi waktu yang terbuang dan meningkatkan efisiensi pemrosesan; (3) Kecepatan eksekusi, dalam kondisi ideal, *pipelining* dapat mengurangi waktu eksekusi total untuk program, karena instruksi dapat dieksekusi lebih cepat dibandingkan dengan pemrosesan *non-pipelined*, di mana satu instruksi harus diselesaikan sepenuhnya sebelum instruksi berikutnya dimulai (Widharma, 2020).

Teknik *pipelining* merupakan metode yang sangat efektif untuk meningkatkan kinerja prosesor dengan memungkinkan eksekusi instruksi secara bersamaan dalam tahap yang berbeda. Meskipun memiliki tantangan, seperti hazards dan kompleksitas desain, keuntungan yang ditawarkan dalam hal throughput dan efisiensi penggunaan sumber daya menjadikan *pipelining* sebagai salah satu teknik fundamental dalam arsitektur komputer modern.

2. *Parallel Processing*

Pemrosesan paralel melibatkan penggunaan beberapa prosesor atau unit pemrosesan untuk menangani tugas-tugas yang berbeda atau bagian-bagian berbeda dari satu tugas pada waktu yang sama (Plunkett, 2023). Ini mempercepat pemrosesan data dengan mendistribusikan beban kerja.

Dalam *parallel preprocessing*, proses pemrosesan dibagi menjadi beberapa bagian yang lebih kecil, yang kemudian

dieksekusi secara bersamaan di beberapa node atau prosesor. Setiap node bertanggung jawab untuk memproses bagian tertentu dari data secara independen. Berikut adalah langkah-langkah umum dalam *parallel preprocessing* (Hidayah, 2018):

- a. Pembagian data, data yang besar dibagi menjadi subset yang lebih kecil, yang kemudian didistribusikan ke beberapa node untuk pemrosesan.
- b. Eksekusi paralel, setiap node melakukan pemrosesan pada subset data yang diterimanya. Proses ini dapat mencakup normalisasi, penghapusan noise, atau transformasi data lainnya.
- c. Pengumpulan hasil, setelah semua node menyelesaikan pemrosesan, hasilnya dikumpulkan dan digabungkan oleh node master untuk menghasilkan output akhir.
- d. Output akhir, hasil akhir disimpan atau digunakan untuk analisis lebih lanjut

Parallel preprocessing merupakan teknik yang sangat efektif untuk meningkatkan kecepatan dan efisiensi dalam pemrosesan data, terutama dalam konteks data besar. Dengan membagi tugas pemrosesan dan menjalankannya secara bersamaan, teknik ini memungkinkan penanganan volume data yang besar dengan cara yang lebih cepat dan efisien. Implementasi yang baik dari *parallel preprocessing* dapat memberikan keuntungan signifikan dalam berbagai aplikasi ilmiah dan industri.

3. *Caching*

Caching adalah teknik penyimpanan data sementara di lokasi yang cepat diakses, seperti *cache* memori, untuk mempercepat pemrosesan dengan mengurangi waktu akses ke data yang sering digunakan (Ridhalri, 2022). Tujuan utama *caching* adalah untuk meningkatkan kecepatan akses data dengan menyimpan salinan data di lokasi yang lebih dekat dengan pengguna atau aplikasi yang membutuhkannya. Dengan menyediakan data dari *cache*, waktu yang dibutuhkan untuk mengambil data dari

sumber asli (seperti disk atau server jarak jauh) dapat dikurangi secara signifikan.

Ketika sebuah aplikasi meminta data, *cache* pertama-tama diperiksa untuk melihat apakah data tersebut sudah ada di dalamnya. Jika data ditemukan di *cache* (disebut *cache hit*), maka data langsung dikembalikan dari *cache*. Jika data tidak ditemukan di *cache* (disebut *cache miss*), maka data diambil dari sumber asli, dikembalikan ke aplikasi, dan juga disimpan di *cache* untuk permintaan selanjutnya.

Berbagai jenis *caching* yang digunakan dalam komputasi, antara lain (Saino et al., 2016):

a. *CPU Caching*

Cache yang dibangun dalam CPU untuk mempercepat akses ke instruksi dan data.

b. *Memory Caching*

Menggunakan memori utama (RAM) untuk menyimpan data yang sering diakses dari disk.

c. *Web Caching*

Menyimpan salinan konten web (HTML, gambar, file CSS/JS) di server proxy atau browser pengguna.

d. *Database Caching*

Menyimpan hasil query database yang sering digunakan di memori.

e. *Content Delivery Network (CDN) Caching*

Menyimpan konten statis di server yang tersebar secara geografis untuk mengurangi latensi.

4. *Memory Hierarchy*

Teknik ini mengorganisir memori komputer dalam tingkatan-tingkatan, mulai dari register yang paling cepat dan berkapasitas kecil hingga penyimpanan sekunder yang lebih lambat namun berkapasitas besar (Rafat et al., 2024). Tujuan utama dari *memory hierarchy* adalah untuk mengoptimalkan kinerja sistem dengan memanfaatkan prinsip lokalitas, yang menyatakan bahwa program cenderung mengakses sebagian

kecil data secara berulang (temporal locality) dan data yang berdekatan dengan data yang baru saja diakses (spatial locality).

Memory hierarchy biasanya terdiri dari beberapa level, masing-masing dengan karakteristik yang berbeda. Berikut adalah level-level utama dalam *memory hierarchy*.

a. *Registers*

Registers adalah unit memori berkecepatan tinggi yang terletak di dalam CPU. Mereka digunakan untuk menyimpan data dan instruksi yang paling sering diakses. Memiliki kecepatan yang sangat cepat, tetapi kapasitasnya sangat kecil (biasanya antara 16 hingga 64 bit).

b. *Cache Memory*

Cache adalah memori kecil yang terletak dekat dengan CPU, yang menyimpan salinan data dan instruksi yang baru-baru ini diakses dari memori utama (RAM). Lebih cepat dibandingkan dengan RAM, tetapi lebih lambat dibandingkan dengan registers. *Cache* biasanya dibagi menjadi beberapa level (L1, L2, L3) dengan L1 menjadi yang tercepat dan terkecil.

c. *Main Memory (RAM)*

Merupakan memori utama dari sistem komputer, yang menyimpan data dan instruksi yang sedang digunakan oleh CPU. RAM memiliki kapasitas yang lebih besar dibandingkan dengan cache tetapi lebih lambat. Lebih lambat dibandingkan cache, tetapi lebih besar kapasitasnya.

d. *Secondary Storage*

Merupakan perangkat penyimpanan non-volatile seperti *hard disk drive* (HDD), *solid-state drive* (SSD), dan perangkat penyimpanan lainnya. Data yang tidak sering diakses disimpan di sini. Memiliki akses yang lebih lambat

KEGIATAN BELAJAR V

SISTEM BILANGAN DAN REPRESENTASI DATA

Oleh: Suwandono, S.Kom., M.Kom

DESKRIPSI PEMBELAJARAN

Pada bab ini bertujuan untuk memberikan pemahaman dasar kepada mahasiswa mengenai sistem bilangan dan bagaimana data direpresentasikan dalam komputer. Melalui penjelasan ini, mahasiswa akan mempelajari berbagai jenis sistem bilangan, cara melakukan konversi antar sistem, serta bagaimana data seperti angka, karakter, dan gambar direpresentasikan di memori komputer.

KOMPETENSI PEMBELAJARAN

Setelah mengikuti perkuliahan ini diharapkan mahasiswa dan mahasiswi memiliki pengetahuan dan kemampuan :

1. Mampu memahami berbagai sistem bilangan, seperti **bilangan biner (basis 2), oktal (basis 8), desimal (basis 10), dan heksadesimal (basis 16)**.
2. Mampu **mengkonversi bilangan** dari satu sistem ke sistem bilangan lain, misalnya dari desimal ke biner atau dari biner ke heksadesimal.
3. Mampu memahami bagaimana **data direpresentasikan** di dalam komputer menggunakan bilangan biner.
4. Mampu mengaplikasikan pengetahuan tentang sistem bilangan dan representasi data dalam **pengolahan data digital**, termasuk pemrosesan sinyal, gambar, dan teks.

A. PENGANTAR SISTEM BILANGAN

Sistem bilangan adalah cara untuk merepresentasikan angka dalam format yang dapat digunakan oleh komputer. Sistem bilangan yang paling umum digunakan dalam komputasi adalah sistem bilangan desimal, biner, oktal, dan heksadesimal.

- **Desimal:** Sistem bilangan berbasis 10 yang menggunakan angka 0 hingga 9.

- **Biner:** Sistem bilangan berbasis 2 yang hanya menggunakan dua digit, 0 dan 1.
- **Oktal:** Sistem bilangan berbasis 8, menggunakan digit dari 0 hingga 7.
- **Heksadesimal:** Sistem bilangan berbasis 16 yang menggunakan digit dari 0 hingga 9 serta huruf A hingga F (untuk mewakili nilai 10 hingga 15).

B. SISTEM BILANGAN DESIMAL, BINER, OKTAL, DAN HEKSADESIMAL

1. Sistem Bilangan Desimal

Pada sistem desimal (lat. decum =10), seperti telah kita ketahui bersama bahwa sistem ini berbasis 10 dan mempunyai 10 simbol yaitu dari angka 0 hingga 9. Setiap tempat mempunyai nilai kelipatan dari 10⁰, 10¹, 10², dst . Penulisan bilangan terbagi dalam beberapa tempat dan banyaknya tempat tergantung dari besarnya ilangan.Setiap tempat mempunyai besaran tertentu yang harga masing-masing tempat secara urut dimulai dari kanan disebut Sistem bilangan desimal menggunakan basis 10. Setiap digit dalam bilangan desimal adalah pangkat dari 10.

	ribuan	ratusan	puluhan	satuan
	10 ³	10 ²	10 ¹	10 ⁰

Contoh:

Angka Desimal 10932 (10932₍₁₀₎)

1 0 9 3 2

Pertama	2 . 10 ⁰ = 2 . 1 = 2
Kedua	3 . 10 ¹ = 3 . 10 = 30
Ketiga	9 . 10 ² = 9 . 100 = 900
Keempat	0 . 10 ³ = 0 . 1000 = 0
Kelima	1 . 10 ⁴ = 1 . 10000 = 10000
	<u>10932</u>

Kebiasaan sehari-hari harga suatu bilangan desimal dituliskan dalam bentuk yang mudah sbb :

$$\begin{aligned}
 10932 &= 1. 10000 + 0. 1000 + 9. 100 + 3. 10 + 2. 1 \\
 &= 1. 10^4 + 0. 10^3 + 9. 10^2 + 3. 10^1 + 2. 10^0
 \end{aligned}$$

2. Sistem Bilangan Biner

Sistem Biner (lat. Dual) atau “duo” yang berarti 2, banyak dipakai untuk sinyal elektronik dan pemrosesan data. Kekhususan sistem biner untuk elektronik yaitu bahwa sistem biner hanya mempunyai 2 simbol yang berbeda, sehingga pada sistem ini hanya dikenal angka “ 0 “ dan angka “1 “.

Contoh:

1 0 1 0 1				

$$\begin{aligned}
3174_{(8)} &= 3 \cdot 8^3 + 1 \cdot 8^2 + 7 \cdot 8^1 + 4 \cdot 8^0 \\
&= 3 \cdot 512 + 1 \cdot 64 + 7 \cdot 8 + 4 \cdot 1 \\
3174_{(8)} &= 1660_{(10)}
\end{aligned}$$

4. Sistem Bilangan Heksadesimal

Sistem Heksadesimal yang juga disebut Sedezimalsystem, banyak dipakai pada teknik komputer. Sistem ini berbasis 16 sehingga mempunyai 16 simbol yang terdiri dari 10 angka yang dipakai pada sistem desimal yaitu angka 0 ... 9 dan 6 huruf A, B, C, D, E dan F. Keenam huruf tersebut mempunyai harga desimal sbb : A = 10; B = 11; C = 12; D = 13; E = 14 dan F = 15. Dengan demikian untuk sistem heksadesimal penulisanya dapat menggunakan angka dan huruf.

Contoh:

2 A F 3

			Pertama	$3 \cdot 16^0$	$= 3 \cdot$	$1 =$	3		
			Kedua	$15 \cdot 16^1$	$= 15 \cdot$	$16 =$	240		
			Ketiga	$10 \cdot 16^2$	$= 10 \cdot$	$256 =$	2560		
			Keempat	$2 \cdot 16^3$	$= 2 \cdot$	$4096 =$	8192		
							$2AF3_{(16)} =$	$10995_{(10)}$	

$$\begin{aligned}
2AF3 &= 2 \cdot 16^3 + 10 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 \\
&= 2 \cdot 4096 + 10 \cdot 256 + 15 \cdot 16 + 3 \cdot 1 \\
&= 10955 \text{ (desimal)}
\end{aligned}$$

C. KONVERSI ANTAR SISTEM BILANGAN

1. Konversi dari Desimal ke Biner

Cara konversi dari desimal ke biner adalah dengan membagi bilangan desimal secara berturut-turut dengan 2 dan mencatat sisa pembagian dari bawah ke atas.

Contoh:

Konversikan 13 ke biner: $13 \div 2 = 6$ sisa 1

$6 \div 2 = 3$ sisa 0

$3 \div 2 = 1$ sisa 1

$1 \div 2 = 0$ sisa 1

Jadi, 13 dalam biner adalah 1101.

2. Konversi dari Biner ke Desimal

Cara konversi dari biner ke desimal adalah dengan mengalikan setiap digit biner dengan pangkat 2 sesuai posisinya.

Contoh:

Bilangan biner 1011 ke desimal adalah $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11$

3. Konversi dari Biner ke Heksadesimal dan Sebaliknya

Konversi biner ke heksadesimal dilakukan dengan memecah bilangan biner menjadi kelompok 4 bit.

Contoh:

Biner 11010110 ke heksadesimal: 1101 = D, 0110 = 6, jadi bilangan heksadesimalnya adalah D6

D. OPERASI ARITMETIKA DALAM SISTEM BILANGAN BINER

1. Penjumlahan Biner

Penjumlahan dalam biner mengikuti aturan:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$1 + 1 = 10 \text{ (dengan carry 1)}$$

Contoh:

$$1011 + 1101 = 110001011 + 1101 = 110001011 + 1101 = 11000$$

2. Pengurangan Biner

Pengurangan biner menggunakan konsep peminjaman, mirip dengan pengurangan desimal.

Contoh:

$$1010 - 0101 = 01011010 - 0101 = 01011010 - 0101 = 0101$$

E. KOMPLEMEN SATU DAN KOMPLEMEN DUA

1. Komplemen Satu

Komplemen satu dari bilangan biner diperoleh dengan membalik setiap bit (0 menjadi 1, 1 menjadi 0).

Contoh:

Komplemen satu dari 1010 adalah 0101.

2. Komplemen Dua

Komplemen dua dari bilangan biner diperoleh dengan menambahkan 1 ke hasil komplemen satu.

Contoh:

Komplemen dua dari 1010 adalah $0101 + 1 = 0110$.

F. REPRESENTASI DATA

Data adalah bilangan biner atau informasi berkode biner lain yang dioperasikan untuk mencapai beberapa hasil penghitungan penghitungan aritmatik, pemrosesan data dan operasi logika, representasi data dinyatakan dengan sign, bilangan magnitude dan posisi titik radiks.

Titik radiks memisahkan bilangan bulat dan pecahan. Penggunaan titik radiks berkaitan dengan jajaran bilangan yang dapat ditampung oleh komputer.

Bagaimana komputer merepresentasikan data ?

- Hanya mengenal 2 keadaan → On atau Off
- Menggunakan sistem biner untuk mengenali dua keadaan tersebut
- Sistem dengan dua digit unik → 0 dan 1, yang disebut bits (singkatan dari binary digits)

Sistem pengkodean yang banyak digunakan untuk merepresentasikan data :

1. *ASCII Code - American Standard Code For Information Interchange*

Dalam bidang mikrokomputer ASCII-Code mempunyai arti yang sangat khusus, yaitu untuk mengkodekan karakter (Huruf, Angka dan tanda baca yang lainnya). Code-code ini merupakan code standard yang dipakai oleh sebagian besar sistem mikrokomputer. Selain huruf, angka dan tanda baca yang lain ada 32 (mis ACK, NAK dsb.) merupakan kontrol untuk keperluan transportasi data. Di bawah ini adalah tabel 7 bit ASCII Code beserta beberapa penjelasan yang diperlukan.

2. *EBCDIC - Extended Binary Coded Decimal Interchange Code*

Sistem pengkodean karakter yang digunakan untuk merepresentasikan teks dalam format biner, dikembangkan oleh IBM pada tahun 1963. EBCDIC digunakan terutama pada komputer mainframe IBM dan perangkat keras lainnya yang terkait dengan mainframe.

KEGIATAN BELAJAR VI

BAHASA ASSEMBLY

Oleh: Salman Farizy. S.Kom., M.Kom

DESKRIPSI PEMBELAJARAN

Pada bab VI ini, harapannya adalah agar mahasiswa/i dapat memahami arsitektur mikroprosesor, perintah bahasa mesin, perbedaan antara pemrograman bahasa mesin/pemrograman bahasa assembly dan tentunya dapat membuat pemerogramman Bahasa Rakitan (Bahasa Assembly) sederhana.

KOMPETENSI PEMBELAJARA

Setelah mengikuti perkuliahan ini diharapkan mahasiswa/i punya pengetahuan dan kemampuan :

1. Bagaimana prosesor dan perangkat keras (hardware) komputer bisa bekerja pada tingkat yang sangat rendah yang mencakup tentang pengelolaan register, memori, dan juga operasi dasar aritmatika serta logika.
2. Menulis program yang cukup efisien didalam penggunaan memori dan juga kecepatan eksekusi. Ini sangat penting untuk pengembangan sistem yang butuh performa tinggi, seperti misalnya embedded systems.

A. PENDAHULUAN

1. Apa itu Bahasa Rakitan (Assembly).

Bahasa Assembly, atau dikenal juga sebagai Assembly, adalah bahasa pemrograman tingkat rendah (low level language) yang biasanya digunakan dalam pemerogramman komputer dan juga perangkat (device) yang bisa diprogram.



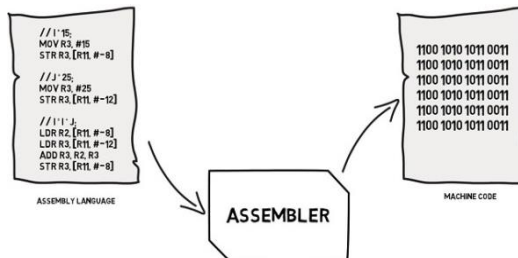
Gambar 6.1. Bahasa

Assembly

Bahasa Assembly ini mengimplementasikan representasi kode mesin dalam bentuk simbol yang lebih gampang dimengerti oleh manusia sebagai pengguna (user), yang menjadi pembeda dengan bahasa pemrograman tingkat tinggi (high level language) adalah, bahwa bahasa Assembly hanya 'support' 1 (satu) atau beberapa jenis arsitektur komputer tertentu saja, sehingga tidak 'seportabel' dengan bahasa tingkat tinggi lainnya. Meskipun demikian, bahasa Assembly memungkinkan para programmer untuk dapat menggunakan kemampuan hardware secara maksimal yang biasanya tidak dapat dilakukan dengan bahasa tingkat tinggi. Bahasa ini tentunya butuh logika kompleks dan juga instruksi yang berbeda dengan bahasa pemrograman lainnya, namun bisa menghasilkan program dengan kecepatan yang cukup baik.

2. Sejarah Bahasa Rakitan (Assembly).

Bahasa rakitan atau biasanya dikenal sebagai bahasa assembly, adalah suatu bahasa pemrograman tingkat rendah (low level language) yang boleh dikatakan hubungannya sangat dekat dengan bahasa mesin atau kode mesin.



Gambar 6.2. Konversi instruksi menjadi kode mesin.

Programmer dalam hal ini menulis instruksi secara langsung dan dikonversi menjadi kode mesin yang dapat dieksekusi oleh CPU (Central Processing Unit).

Fasilitas atau sarana yang terdapat dalam program assembly sangatlah minim, tidak demikian halnya bahasa pemrograman tingkat atas (high level language programming) semuanya sudah 'ready' atau siap pakai.

Seperti kita ketahui bahwa bahasa assembly sendiri tidak diciptakan oleh 1 (satu) individu tertentu saja, melainkan berkembang sebagai suatu kebutuhan yang timbul secara alami didalam sejarah komputer. Tapi, beberapa tokoh dan juga tim yang terlibat dalam pengembangan komputer awal, tentunya berperan penting juga dalam evolusi bahasa assembly.

- **John von Neumann & Tim di IAS (Institute for Advanced Study).**

Bersama dengan timnya tahun 1940-an, mengembangkan suatu arsitektur komputer yang cukup dikenal sebagai arsitektur von Neumann. Walaupun secara tidak langsung menciptakan bahasa assembly, tapi konsep ini akhirnya menjadi dasar untuk komputer modern dan juga memungkinkan pengembangan bahasa assembly sebagai penghubung antara bahasa mesin dan juga pemrogram.

- **Kathleen Booth.**

Merupakan salah 1 (satu) tokoh awal yang terlibat dalam pengembangan bahasa assembly pada tahun 1940-an dan 1950-an berkolaborasi dengan suaminya yang bernama Andrew dalam mengembangkan berbagai komputer. Sebagai salah 1 orang pertama, Kathleen Booth menulis manual untuk bahasa assembly dan mendokumentasikan bahasa assembly yang digunakan pada mesin tersebut.

- **Maurice Wilkes & EDSAC**

Bersama dengan para timnya yang berada di University of Cambridge, membuat suatu project untuk membangun EDSAC (Electronic Delay Storage Automatic Calculator) akhir 1940-an, dengan mengembangkan salah 1 (satu) sistem awal untuk

merangkai/merakit instruksi mesin secara otomatis, yang akhirnya menjadi langkah penting dalam evolusi bahasa assembly.

- **IBM (International Business Machines Corporation) dan Perkembangan Standar Bahasa Assembly.**

IBM Pada 1950-an, memainkan andil dan peran cukup penting dalam pengembangan dan juga standarisasi bahasa assembly untuk komputer mainframe mereka, seperti misalnya IBM 701 serta IBM 704. IBM mengembangkan bahasa assembler yang pertama mengubah kode assembly menjadi suatu kode mesin. Walaupun tidak ada 1 individu yang dapat disebut sebagai penemu dari bahasa assembly, ada beberapa ilmuwan dan juga insinyur komputer yang sudah disebutkan diatas seperti Kathleen Booth, Maurice Wilkes dan juga tim IBM, punya peran kunci dalam pengembangannya. Bahasa assembly lebih merupakan hasil dari perubahan atau evolusi teknologi dan juga kolaborasi antar banyak pihak dalam sejarah awal komputasi.

Dibawah ini akan dijelaskan garis besar dari sejarah perkembangan bahasa rakitan (Assembly) :

- Awal Mula (1940-an sampai dengan 1950-an) ENIAC dan komputer generasi pertama (1st Generation Computers). Memasuki era komputer digital (kira – kira tahun 1940-an), pemrograman dilakukan secara langsung dengan menggunakan kode mesin, yang terdiri dari beberapa deretan angka biner. Disanalah bahasa rakitan muncul sebagai cara

KEGIATAN BELAJAR VII

KOMPUTER DAN PROSESOR

Oleh: S. Nurmuslimah, S.T., M.T.

DESKRIPSI PEMBELAJARAN

Pada bab ini mahasiswa mempelajari tentang Pengenalan Komputer dan Prosesor. Diharapkan mahasiswa memiliki wawasan dan pemahaman untuk mempelajari komputer dan prosesor

khususnya tentang software, hardware dan brainware serta prosesor lebih lanjut.

KOMPETENSI PEMBELAJARAN

Setelah mengikuti perkuliahan ini diharapkan mahasiswa dan mahasiswi memiliki pengetahuan dan kemampuan :

1. Mampu menguraikan definisi komputer dan prosesor
2. Mampu menjelaskan fungsi dan manfaat komputer dan prosesor
3. Mampu menjelaskan tingkatan, struktur, jenis-jenis komputer dan prosesor.

A. PENGERTIAN KOMPUTER

Komputer saat ini dan piranti pendukungnya telah masuk dalam setiap aspek kehidupan dan pekerjaan. Komputer yang ada sekarang memiliki kemampuan yang lebih dari sekedar perhitungan matematik biasa. Diantaranya adalah sistem komputer di kassa supermarket yang mampu membaca kode barang belanjaan, sentral telepon yang menangani jutaan panggilan dan komunikasi, jaringan komputer dan internet yang menghubungkan berbagai tempat di dunia.

Istilah dari komputer berasal dari bahasa latin “computare” yang mengandung arti “menghitung” (to compute atau reckon). (Jogiyanto, 1999).

Komputer adalah mesin penghitung elektronik yang cepat dapat menerima informasi input digital. Memrosesnya dengan suatu program yang tersimpan di memorinya (storage program) dan menghasilkan output informasi (V. C. Hamacher, Z.G.Vranesic, dan S.G.Zaky, 2001).

Menurut buku Computer Annual (Robert H.Blissmer) : “Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas berikut: menerima input, memproses input sesuai dengan programnya, menyimpan perintah-perintah dan hasil dari pengolahan serta menyediakan output dalam bentuk informasi”.

Menurut buku Computer Today (Donald H.Sanders): “Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara

otomatis menerima dan menyimpan data input, memprosesnya dan menghasilkan output dibawah pengawasan serta langkah-langkah dan instruksi-instruksi program yang tersimpan di memori (stored program)".

Menurut buku Introduction to Computer (Gordon B.Davis): "Komputer adalah tipe khusus alat penghitung yang mempunyai sifat tertentu yang pasti".

Arsitektur komputer sangat berkaitan dengan fungsi-fungsi operasi dari masing-masing komponen sistem komputer. Sedangkan organisasi komputer sangat berkaitan dengan unit-unit operasional dan interkoneksinya yang dapat merealisasikan spesifikasi arsitektural. Susunan dari komponen-komponen komputer yang saling terintegrasi dan memiliki tujuan dalam melakukan sebuah operasi tertentu maka terbentuklah sebuah sistem komputer.

Sistem komputer adalah kumpulan dari elemen-elemen komputer (Hardware, software, brainware) yang saling berhubungan (terintegrasi) dan saling berinteraksi untuk melakukan pengolahan data dengan tujuan menghasilkan informasi sesuai dengan yang diharapkan.(Jogiyanto,1999)

Komputer adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasarkan urutan instruksi ataupun program yang diberikan kepadanya. Istilah mengenai sekelompok mesin ataupun istilah mengenai jutaan komponen kemudian dikenal sebagai Hardware Computer atau perangkat keras komputer.

Hardware komputer juga dapat diartikan sebagai peralatan fisik dari komputer itu sendiri. Peralatan yang secara fisik dapat dilihat, dipegang ataupun dipindahkan. Dalam hal ini, komputer tidak mungkin bisa bekerja tanpa adanya program yang telah dimasukkan kedalamnya. Program ini bisa berupa suatu prosedur pengoperasian dari komputer itu sendiri ataupun berbagai prosedur dalam hal pemrosesan data yang telah ditetapkan sebelumnya.

Dan program-program inilah yang kemudian disebut sebagai **Software Computer atau perangkat lunak komputer**. Secara prinsip, komputer hanyalah merupakan sebuah alat yang

bisa digunakan untuk membantu manusia dalam menyelesaikan pekerjaannya.

Pengertian manusia kemudian dikenal dengan istilah brainware atau perangkat manusia.

Konsep hardware-software-brainware adalah merupakan konsep tri-tunggal yang tidak bisa dipisahkan satu dengan lainnya.

B. SEJARAH KOMPUTER

Proses pengolahan data telah dilakukan oleh manusia. Sejak dulu manusia menemukan alat-alat mekanik dan elektronik untuk membantu pekerjaan manusia dalam penghitungan dan pengolahan data supaya bisa mendapatkan hasil lebih cepat. Komputer yang ditemui saat ini adalah suatu evolusi panjang dari penemuan-penemuan manusia sejak dahulu kala berupa alat mekanik maupun elektronik.

1. Komputer Generasi Pertama

Tabung hampa udara (vacuum-tube) sebagai penguat sinyal, merupakan ciri khas komputer generasi pertama dengan bahan bakunya terdiri dari kaca, sehingga memiliki banyak kelemahan seperti mudah pecah dan mudah menyalurkan panas. Untuk menetralkan panas diperlukan komponen lain yang berfungsi sebagai pendingin. Dengan adanya komponen tambahan tersebut maka komputer menjadi besar, berat dan mahal.

Pada tahun 1946, komputer elektronik di dunia yang pertama yakni ENIAC (Electronic Numerical Integrator and Computer yakni perkembangan komputer masa kini yang dibuat oleh pemerintah Amerika Serikat dan University of Pennsylvania. Terdiri dari 18.000 tabung vakum, 70.000 resistor, dan 5 juta titik solder, komputer tersebut merupakan mesin yang sangat besar yang mengkonsumsi daya sebesar 160kW, berbobot 30 ton dengan panjang 30 meter dan tinggi 2,4 meter sehingga memerlukan ruangan kelas tersendiri.

2. Komputer Generasi Kedua

Transistor merupakan ciri khas komputer generasi kedua. Bahan bakunya terdiri atas tiga lapis yaitu : basic, collector dan

emitter. Transistor merupakan singkatan dari Transfer Resistor, yang berarti dengan mempengaruhi daya tahan antara dua dari tiga lapisan, maka daya (resistor) yang ada pada lapisan berikutnya dapat pula dipengaruhi. Dengan demikian fungsi transistor adalah sebagai penguat sinyal, sebagai komponen padat, transistor mempunyai banyak keunggulan seperti misalnya: tidak mudah pecah dan tidak menyalurkan panas, dengan demikian komputer menjadi lebih kecil dan lebih murah.

Beberapa contoh komputer generasi kedua adalah : IBM Serie 1400, MARK IV (diproduksi di Jepang tahun 1957) dan IBM-7090 (buatan Amerika)

3. Komputer Generasi ketiga

Ciri khas komputer generasi ketiga adalah Integrated Circuit atau IC-Chip. IC adalah gabungan dari ribuan transistor dengan bentuk kecil dan ukuran beberapa milimeter. Contoh komputer generasi ini adalah : Apple Computer dan IBM S/360.

4. Komputer Generasi Keempat

Pada generasi ini ditandai dengan munculnya LSI (Large Scale Integration) yang merupakan pemadatan ribuan IC ke dalam sebuah Chip. Istilah chip digunakan untuk menunjukkan suatu lempengan persegi empat yang memuat rangkaian-rangkaian terpadu. LSI kemudian dikembangkan dalam VLSI (Very Large Scale Integration).

Perkembangan berikutnya ditandai dengan munculnya microprocessor dan semiconductor. Perusahaan pembuat microprocessor antara lain : Intel Corporation, Motorola, Zilog, AMD dll. Intel Corp mengeluarkan microprocessor dengan model 4004, 8088, 80286, 80386, 80486, 80586 atau dikenal dengan nama Pentium. Sedangkan pabrik Motorola mengeluarkan model 6502, 6800. Perkembangan teknologi Tablet PC yang merupakan komputer portable masih tergolong generasi keempat.

5. Komputer Generasi Kelima

Mendefinisikan komputer generasi kelima menjadi cukup sulit karena tahap ini masih sangat muda. Contoh imajinatif komputer generasi kelima adalah komputer fiksi HAL9000 dari novel karya Arthur C. Clarke berjudul 2001: Space Odyssey. HAL menampilkan seluruh fungsi yang diinginkan dari sebuah komputer

generasi kelima. Dengan kecerdasan buatan (artificial intelligence atau AI), HAL dapat memiliki nalar untuk melakukan percakapan dengan manusia, menggunakan masukan visual, dan belajar dari pengalamannya sendiri. Kemajuan di bidang desain komputer dan teknologi semakin memungkinkan dalam pembuatan komputer generasi kelima. Dua kemajuan rekayasa terutama kemampuan dalam pemrosesan paralel, yang akan menggantikan model non Neumann. Model non Neumann akan digantikan dengan sistem yang mampu mengkoordinasikan banyak CPU untuk bekerja secara serempak. Kemajuan lain adalah teknologi superkonduktor yang memungkinkan aliran elektrik tanpa ada hambatan apapun, nantinya dapat mempercepat kecepatan informasi.

Perkembangan komputer portabel seperti halnya Tablet PC mulai ada penambahan fitur yang menggunakan Artificial Intelligence mengarah pada generasi komputer ke-5.

Contoh : fitur Voice Recognition. Teknologi terbaru adalah perkembangan drone yang bisa dikendalikan dengan lambaian tangan.

C. KEMAMPUAN KOMPUTER

Suatu komputer memiliki kemampuan sebagai berikut :

1. Komputer mempunyai kecepatan dan ketepatan dalam mengolah data dan menghasilkan informasi.
2. Kemampuan kapasitas penyimpanan data dalam memori atau simpanan luar seperti harddisk saat ini terjadi peningkatan kapasitas yang sangat luar biasa.
3. Kemampuan-kemampuan lain yang dimiliki komputer adalah kecepatan mengirim data dalam format apapun antar komputer dalam jaringan wilayah lokal, regional, maupun global.

Kecepatan akses data dalam sebuah operasi komputer dinyatakan dalam satuan seperti Millisecond (ms), Microsecond (ms), Nanosecond (ns), Picosecond (ps). Satuan Picosecond (ps), merupakan satuan waktu yang paling cepat dalam sebuah akses komputer.

Adapun Satuan Waktu Kecepatan Komputer adalah :

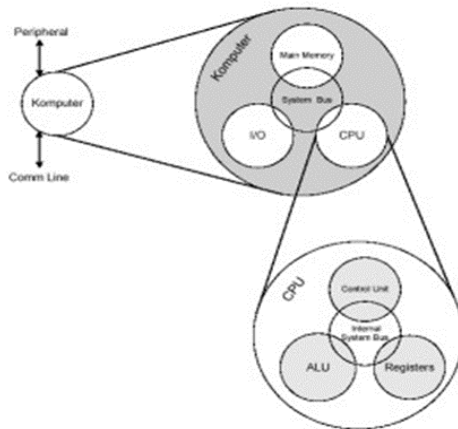
- ✓ Millisecond (ms) dengan kecepatan Ribu operasi per detik (1/1000)
- ✓ Microsecond (μ s) dengan kecepatan Juta operasi per detik (1/1.000.000)
- ✓ Nanosecond (ns) dengan kecepatan Milliar operasi per detik (1/1.000.000.000)
- ✓ Picosecond (ps) dengan kecepatan Tribun operasinya

D. STRUKTUR KOMPUTER

Struktur dalam sistem komputer dapat didefinisikan sebagai cara atau bagaimana masing-masing komponen di dalam sistem komputer tersebut terhubung. sedangkan masing-masing komponen memiliki fungsi dan tugas masing-masing yang menjadi bagian dari tujuan besar dari sistem komputer. Sebagai contoh sebuah komputer memiliki keyboard dan monitor. Keyboard berfungsi untuk memberikan masukan perintah kepada komputer. Monitor berfungsi untuk menampilkan hasil dari pemrosesan yang telah dilakukan oleh komputer. Jika salah satu dari komponen itu tidak ada, maka kita tidak bisa menggunakan komputer tanpa keyboard dan tidak bisa menggunakan komputer tanpa monitor.

Komputer adalah sebuah sistem yang berinteraksi dengan cara tertentu dengan dunia luar. Interaksi dengan dunia luar dilakukan melalui perangkat peripheral dan saluran komunikasi. Dalam struktur internal komputer ada empat struktur utama yaitu :

- ✓ Central Processing Unit (CPU), berfungsi mengendalikan kerja komputer, pemrosesan data dan pusat pengolahan fungsi – fungsi komputer. CPU sering juga disebut sebagai processor (prosesor).
- ✓ Memori Utama, berfungsi sebagai penyimpan data yang akan atau baru saja diproses.
- ✓ I/O (Input/Output), berfungsi memindahkan data antara komputer ke lingkungan atau perangkat lainnya.
- ✓ System Interconnection, yang lebih dikenal dengan istilah bus merupakan mekanisme yang menghubungkan komunikasi internal antara CPU, memori utama dan I/O.



Gambar 7.1 Struktur Dasar Komputer

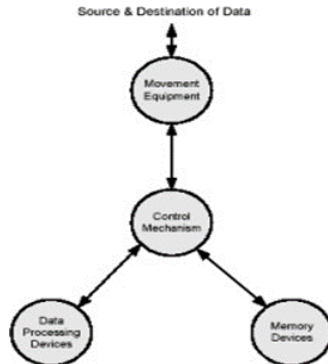
Salah satu komponen yang paling kompleks yang merupakan jantung komputer adalah CPU. Adapun Struktur utama dalam CPU adalah :

- ✓ Control Unit, berfungsi untuk mengontrol operasi CPU dan mengontrol komputer secara keseluruhan.
- ✓ Arithmetic and Logic Unit (ALU), berfungsi untuk membentuk fungsi– fungsi pengolahan data komputer.
- ✓ Register, berfungsi sebagai penyimpan internal bagi CPU. CPU Interconnection, berfungsi menghubungkan seluruh bagian dari CPU.

E. FUNGSI KOMPUTER

Fungsi dasar sistem komputer adalah sangat sederhana. Pada prinsipnya terdapat empat buah fungsi operasi, yaitu :

- ✓ Fungsi Operasi Pengolahan Data, seperti penjumlahan, pengurangan, konversi terhadap data, dsb.
- ✓ Fungsi Operasi Penyimpanan Data, seperti penyimpanan data hasil pemrosesan, pengambilan data yang telah disimpan.
- ✓ Fungsi Operasi Pemindahan Data, seperti pergerakan data internal dan pergerakan data eksternal,
- ✓ Fungsi Operasi Kontrol, berfungsi mengendalikan ketiga fungsi diatas



Gambar 7.2 Fungsi Dasar Komputer

Komputer dapat memproses data dan representasi data, tetapi nantinya data harus disesuaikan dengan mesin pemroses. Dalam pengolahan data, komputer memerlukan unit penyimpanan sehingga diperlukan suatu mekanisme penyimpanan data. Walaupun hasil komputer digunakan pada saat itu, komputer memerlukan media penyimpanan untuk proses data. Dalam interaksi dengan dunia luar sebagai fungsi pemindahan data diperlukan antarmuka (interface), proses ini dilakukan oleh unit Input/Output (I/O) dan perangkatnya disebut peripheral. Saat

KEGIATAN BELAJAR VIII

SISTEM PENGENDALIAN KOMPUTER

Oleh: Prajoko,M.Kom

DESKRIPSI PEMBELAJARAN

Bab ini menyajikan elemen dan konsep dasar sistem yang dikendalikan komputer. Diskritisasi dan pilihan frekuensi sampling akan diperiksa terlebih dahulu, diikuti dengan studi model waktu diskrit dalam domain waktu dan frekuensi, sistem waktu diskrit dalam loop tertutup dan prinsip dasar perancangan pengontrol digital. Diharapkan mahasiswa memiliki wawasan bagaimana cara kerja pengontrolan digital sebagai modal dalam mempelajari matakuliah organisasi arsitektur komputer.

KOMPETENSI PEMBELAJARAN

1. Setelah membaca bab ini, mahasiswa mengetahui perbedaan sistem kontrol digital dan kontrol analog.
2. Mahasiswa mengetahui model transformasi untuk menganalisis sistem domain waktu diskrit
3. Mahasiswa mengetahui model sistem kontrol dalam bentuk fungsi transfer dan persamaan keadaan.

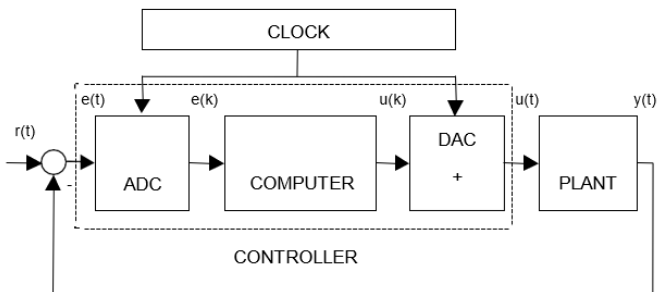
A. PENDAHULUAN

Pendekatan pertama untuk memperkenalkan komputer digital atau mikroprosesor ke dalam loop kontrol ditunjukkan pada Gambar 8.1. Kesalahan terukur antara referensi dan keluaran pabrik diubah ke dalam bentuk digital oleh konverter analog-ke-digital (ADC), pada saat pengambilan sampel k yang ditentukan oleh jam sinkronisasi. Komputer menginterpretasikan sinyal yang diubah $y(k)$ sebagai urutan angka, yang diproses menggunakan algoritma kontrol dan menghasilkan urutan angka baru $\{u(k)\}$ yang mewakili kontrol. Melalui konverter digital-ke-analog (DAC), urutan ini diubah menjadi sinyal analog, yang dipertahankan konstan antara saat pengambilan sampel dengan penahan orde nol (ZOH). Kaskade: ADC-komputer-DAC harus berperilaku dengan cara yang

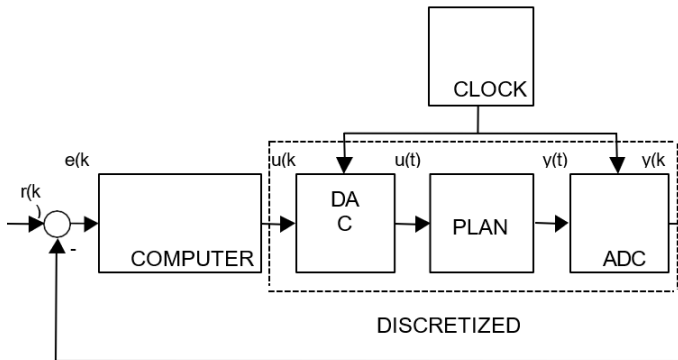
sama seperti pengontrol analog (tipe PID), yang menyiratkan penggunaan frekuensi pengambilan sampel yang tinggi tetapi algoritma yang diterapkan pada komputer sangat sederhana (kami hanya tidak memanfaatkan potensi komputer digital!).

Pendekatan kedua yang jauh lebih menarik untuk pengenalan komputer digital atau mikroprosesor dalam loop kontrol diilustrasikan dalam Gambar 8.2 yang dapat diperoleh dari Gambar 2.1 dengan memindahkan komparator keluaran referensi setelah konverter analog-ke-digital. Referensi sekarang ditetapkan secara digital sebagai urutan yang disediakan oleh komputer.

Pada Gambar 8.2, himpunan DAC - plant - ADC ditafsirkan sebagai sistem terdiskritisasi, yang input kontrolnya adalah urutan $\{u(k)\}$ yang dihasilkan oleh komputer, outputnya adalah urutan $\{y(k)\}$ yang dihasilkan dari konversi A/D dari output sistem $y(t)$. Sistem terdiskritisasi ini dicirikan oleh "model waktu-diskrit", yang menggambarkan hubungan antara deret bilangan $\{u(k)\}$ dan deret bilangan $\{y(k)\}$. Model ini terkait dengan model waktu kontinu dari pabrik. Pendekatan ini menawarkan beberapa keuntungan. Di antara keuntungan-keuntungan ini, kami mengingat hal-hal berikut: 1. Frekuensi pengambilan sampel dipilih sesuai dengan "lebar pita" sistem waktu kontinu (akan jauh lebih rendah daripada pendekatan pertama). 2. Kemungkinan desain langsung algoritma kontrol yang disesuaikan dengan model pabrik terdiskritisasi. 3. Penggunaan komputer yang efisien karena peningkatan periode pengambilan sampel memungkinkan daya komputasi digunakan untuk mengimplementasikan algoritma yang lebih berkinerja tetapi lebih kompleks daripada pengontrol PID, dan yang memerlukan waktu komputasi yang lebih lama.



Gambar 8.1. Realisasi digital dari pengontrol tipe « analog »



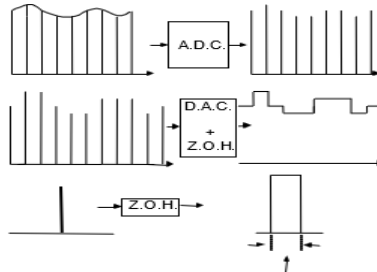
Gambar 8.2. Sistem kontrol digital

Model ini terkait dengan model waktu kontinu dari pabrik. Pendekatan ini menawarkan beberapa keuntungan. Di antara keuntungan-keuntungan ini, kami mengingat hal-hal berikut:

1. Frekuensi pengambilan sampel dipilih sesuai dengan "lebar pita" sistem waktu kontinu (akan jauh lebih rendah daripada pendekatan pertama).
2. Kemungkinan perancangan langsung algoritma kontrol yang disesuaikan dengan model pabrik terdiskritisasi.
3. Penggunaan komputer yang efisien karena peningkatan periode pengambilan sampel memungkinkan daya komputasi digunakan untuk mengimplementasikan algoritma yang lebih berkinerja tetapi lebih kompleks daripada pengontrol PID, dan yang memerlukan waktu komputasi yang lebih lama.

Faktanya, jika seseorang benar-benar ingin memanfaatkan penggunaan komputer digital dalam loop kontrol, "bahasanya" juga harus diubah. Ini dapat dicapai dengan mengganti model sistem waktu kontinu dengan model sistem waktu diskrit, pengontrol waktu kontinu dengan algoritma kontrol digital, dan dengan menggunakan teknik perancangan kontrol khusus. Perubahan ke "bahasa" baru ini (model dinamis waktu diskrit) memungkinkan penggunaan berbagai strategi kontrol berkinerja tinggi yang tidak dapat diimplementasikan oleh pengontrol analog. Rincian pengoperasian ADC (konverter analog-ke-digital), DAC (konverter digital-ke-analog), dan ZOH (zero-order hold) diilustrasikan pada Gambar 8.3. periode

pengambilan sampel Gambar 8.3. Pengoperasian konverter analog-ke-digital (ADC), konverter digital-ke-analog (DAC), dan zero-order hold (ZOH).



Gambar 8.3 Pengoperasian converter analog kedigital (ADC),converter digital ke analog (DAC)

Konverter analog-ke-digital mengimplementasikan dua fungsi:

1. Pengambilan sampel sinyal analog: operasi ini terdiri dari penggantian sinyal kontinu dengan rangkaian nilai yang berjarak sama dalam domain waktu (jarak temporal antara dua nilai adalah periode pengambilan sampel), karena nilai-nilai ini sesuai dengan amplitudo sinyal kontinu di pengambilan sampel instan.
2. Kuantisasi: ini adalah operasi yang dengannya amplitudo suatu sinyal direpresentasikan dengan sekumpulan nilai berbeda yang terpisah (nilai sinyal yang terkuantisasi), umumnya dikodekan dengan urutan biner.

Penggunaan umum konverter A/D resolusi tinggi (di mana sampel dikodekan dengan 12 bit atau lebih) memungkinkan seseorang untuk mempertimbangkan efek kuantifikasi sebagai hal yang dapat diabaikan, dan asumsi ini akan berlaku sebagai berikut.

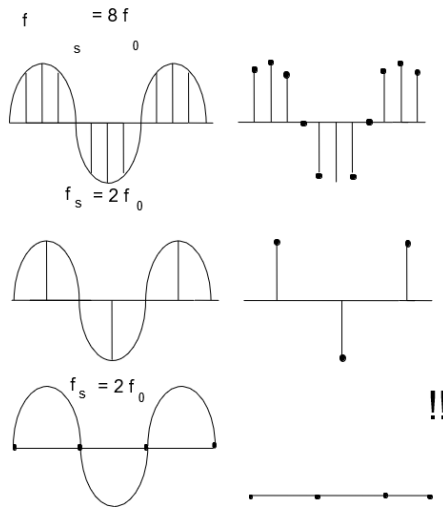
Konverter digital-analog (DAC) mengubah sinyal diskrit, yang dikodekan secara digital, menjadi sinyal kontinu pada saat pengambilan sampel.

Zero-order hold (ZOH) menjaga sinyal kontinu ini tetap konstan antara dua saat pengambilan sampel (periode pengambilan sampel), untuk menghasilkan sinyal waktu kontinu.

B. DISKRITISASI DAN IKHTISAR SISTEM DATA SAMPEL

1. Diskritisasi dan Pilihan Frekuensi Sampling

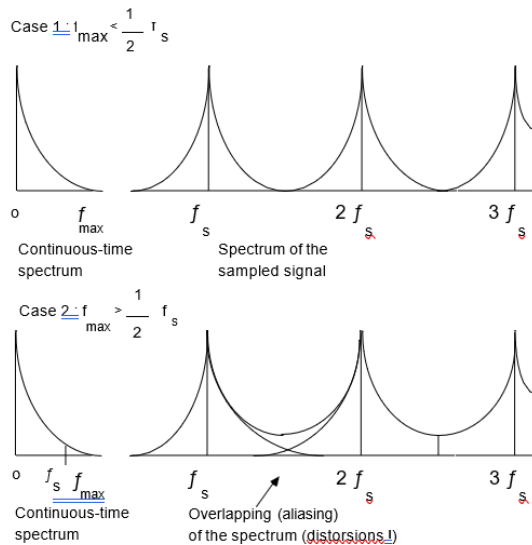
Gambar 8.4 mengilustrasikan diskritisasi sinusoida frekuensi f_0 untuk beberapa frekuensi sampling f_s . Dapat dicatat bahwa, untuk frekuensi sampling $f_s = 8 f_0$, sifat kontinu sinyal analog tidak berubah dalam sinyal sampel. Untuk frekuensi sampling $f_s = 2 f_0$, jika pengambilan sampel dilakukan pada saat-saat $2\pi f_0 t$ selain kelipatan π , sinyal sampel periodik masih diperoleh. Namun, jika pengambilan sampel dilakukan pada saat-saat di mana $2\pi f_0 t = n\pi$, urutan sampel yang sesuai identik dengan nol. Jika frekuensi pengambilan sampel dikurangi di bawah batas $f_s = 2f_0$, sinyal sampel periodik masih muncul, tetapi frekuensinya berbeda dari sinyal kontinu ($f = f_s - f_0$). Untuk merekonstruksi sinyal kontinu dari urutan sampel, frekuensi pengambilan sampel harus memenuhi syarat (teorema Nyquist): $f_s > 2 f_{\max}$ (2.2.1)



Gambar 8.4 Diskritisasi sinyal sinusoidal

di mana **f_{max}** adalah frekuensi maksimum yang akan ditransmisikan. Frekuensi $f_s = 2 f_{\max}$ merupakan batas teoritis; dalam praktiknya, frekuensi pengambilan sampel yang lebih tinggi harus dipilih. Adanya batas maksimum untuk frekuensi yang dapat diubah tanpa distorsi, untuk frekuensi pengambilan sampel tertentu,

juga dapat dipahami ketika diamati bahwa pengambilan sampel sinyal waktu kontinu adalah "modulasi magnitudo" dari frekuensi "pembawa" f_s (analogi dengan modulasi magnitudo dalam pemancar radio). Efek modulasi dapat diamati dalam replikasi spektrum sinyal modulasi (dalam kasus kami sinyal kontinu) di sekitar frekuensi pengambilan sampel dan kelipatannya. Spektrum sinyal yang diambil sampelnya, jika frekuensi maksimum sinyal kontinu (f_{max}) kurang dari $(1/2) f_s$, direpresentasikan di bagian atas Gambar 2.5. Spektrum sinyal yang diambil sampelnya, jika $f_{max} > (1/2)f_s$, direpresentasikan di bagian bawah Gambar 2.5. Fenomena tumpang tindih (aliasing) dapat diamati. Hal ini sesuai dengan munculnya distorsi. Frekuensi $(1/2)f_s$, yang menentukan frekuensi maksimum (f_{max}) yang diizinkan untuk pengambilan sampel tanpa distorsi, dikenal sebagai "frekuensi Nyquist" (atau frekuensi Shannon).

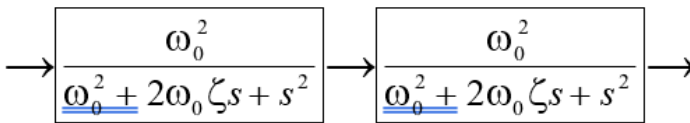


Gambar 8.5. Spektrum sinyal sampel

Untuk frekuensi pengambilan sampel tertentu, guna menghindari pelipatan (aliasing) spektrum dan distorsi, sinyal analog harus disaring sebelum pengambilan sampel guna memastikan bahwa:

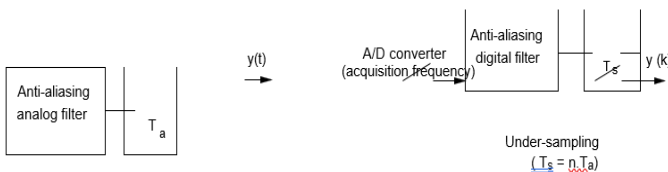
$$f_{\max} < \frac{1}{2} f_s$$

Filter yang digunakan dikenal sebagai "filter anti-aliasing". Filter anti-aliasing yang baik harus memiliki minimal dua sel orde kedua yang disusun bertingkat ($f_{\max} \ll (1/2) f_s$). Contoh filter anti-aliasing jenis ini diberikan pada Gambar 8.6. Filter ini harus menghasilkan redaman besar pada frekuensi yang lebih tinggi dari $(1/2) f_s$, tetapi lebar pita harus lebih tinggi dari lebar pita yang dibutuhkan sistem loop tertutup (umumnya lebih tinggi dari lebar pita sistem loop terbuka). Rangkaian jenis ini (atau yang lebih kompleks) saat ini tersedia.



Gambar 8.6 Anti Aliasing Filter

Dalam kasus pengambilan sampel frekuensi sangat rendah, pertama-tama pengambilan sampel pada frekuensi yang lebih tinggi dilakukan (kelipatan bilangan bulat dari frekuensi yang diinginkan), menggunakan filter anti-aliasing analog yang sesuai. Sinyal sampel yang diperoleh kemudian dilewatkan melalui filter anti-aliasing digital diikuti oleh pembagi frekuensi (desimasi) sehingga menghasilkan sinyal sampel yang memiliki frekuensi yang diperlukan. Prosedur ini ditunjukkan pada Gambar 8.7. Prosedur ini juga digunakan setiap kali frekuensi akuisisi data lebih tinggi daripada frekuensi pengambilan sampel yang dipilih untuk loop yang harus dikontrol (frekuensi pengambilan sampel harus merupakan pembagi bilangan bulat dari frekuensi akuisisi.)



Gambar 8.7. Anti-aliasing filtering with under-sampling

2. Pemilihan Frekuensi Sampling untuk Sistem Kontrol

Frekuensi sampling untuk sistem kontrol digital dipilih sesuai dengan lebar pita yang diinginkan dari sistem loop tertutup. Perhatikan bahwa, tidak peduli bagaimana kinerja yang diinginkan ditentukan, kinerja ini selalu dapat dikaitkan dengan lebar pita sistem loop tertutup. Contoh: Mari kita pertimbangkan kinerja yang dikenakan pada pada respons langkah (overshoot maksimum 5%, waktu naik 2,75 d). Fungsi transfer yang akan ditentukan sesuai dengan fungsi transfer sistem loop tertutup yang diinginkan. Dari diagram yang diberikan pada Gambar 13.11, kami telah menyimpulkan bahwa fungsi transfer loop tertutup harus berupa fungsi transfer orde kedua yang dinormalisasi dengan $\zeta=0,7$ dan $\omega_0=1$ rad/dtk. Dengan segera menggunakan diagram yang , dapat diamati bahwa lebar pita sistem loop tertutup kira-kira sama dengan

$$f_B^{CL} = \frac{1}{2\pi} \text{ Hz}$$

Dimana:

$$f_s: \text{ sampling frequency, } f_z^c$$

Merupakan : lebar pita sistem loop tertutup.

Aturan Persamaan 8.2.3 juga digunakan dalam loop terbuka, ketika diinginkan untuk memilih frekuensi pengambilan sampel guna mengidentifikasi model waktu diskrit suatu pabrik. Dalam kasus ini f_{CL} digantikan oleh estimasi lebar pita pabrik. Untuk tujuan informasi, Tabel 2.1 memberikan periode pengambilan sampel ($T_s = 1/f_s$) yang digunakan untuk kontrol digital berbagai jenis pabrik. Aturan untuk memilih frekuensi pengambilan sampel yang diberikan dalam Persamaan 2.2.3 dapat dihubungkan ke parameter fungsi transfer.

Sistem orde pertama:

$$H(s) = \frac{1}{1 + sT_0}$$

alam kasus ini lebar pita sistem adalah:

$$f_B \approx f_0 = \frac{1}{2\pi T_0}$$

(dengan atenuasi lebih besar dari 3 db diperkenalkan untuk frekuensi lebih tinggi dari $\omega_0 = 1/T_0 = 2\pi f_0$).

Tabel 8.1. Pemilihan periode pengambilan sampel untuk sistem kontrol digital (nilai indikatif)

Type of variable (or plant)	Sampling period (s)
Flow rate	1 – 3
Level	5 – 10
Pressure	1 – 5
Temperature	10 - 180
Distillation	10 - 180
Servo-mechanisms	0.001 - 0.05
Catalytic reactors	10 - 45
Cement plants	20 - 45
Dryers	20 – 45

Dengan menerapkan aturan Persamaan 8.2.3 diperoleh kondisi untuk memilih periode pengambilan sampel ($T_s = 1/f_s$): $T_0 < T_s < 4 T_0$ (2.2.4). Hal ini sesuai dengan keberadaan dua hingga sembilan sampel pada waktu naik respons bertahap. Sistem orde

kedua $H(s) = \frac{2\omega_0^2}{\omega^2 + 2\zeta\omega_0 s + \omega_0^2}$ Lebar pita sistem orde kedua bergantung pada ω_0 dan ζ (lihat Gambar 13.12).

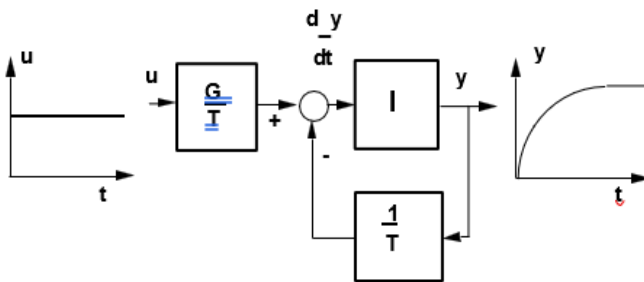
Misalnya: $\zeta = 0,7 \Rightarrow f_B = \omega_0 2\pi \zeta = 1 \Rightarrow f_B = 0,6\omega_0 2\pi$
 Dengan menerapkan aturan Persamaan 2.2.3, diperoleh hubungan berikut antara frekuensi alami ω_0 dan periode sampling T_s : $0,25 \leq \omega_0 T_s \leq 1$; $\zeta = 0,7$ (2.2.5) dan $0,4 \leq \omega_0 T_s \leq 1,75$; $\zeta = 1$ (8.2.6)

Nilai yang lebih rendah sesuai dengan pilihan frekuensi pengambilan sampel yang tinggi dan nilai yang lebih tinggi sesuai dengan pilihan frekuensi pengambilan sampel yang rendah. Untuk mempermudah, mengingat bahwa dalam loop tertutup perilaku yang sering dipilih sebagai perilaku yang diinginkan adalah perilaku orde kedua yang memiliki faktor redaman ζ antara 0,7 dan 1, aturan berikut dapat digunakan (perkiraan Persamaan 8.2.5 dan 8.2.6): $0,25 \leq \omega_0 T_s \leq 1,5$; $0,7 \leq \zeta \leq 1$ (2.2.7)

C. MODEL WAKTU DISKRIT

1. Domain Waktu

Gambar 8.8 mengilustrasikan respons sistem waktu kontinu terhadap input bertahap, respons yang dapat disimulasikan oleh sistem orde pertama (integrator dengan penguatan umpan balik yang ditunjukkan pada gambar).



Gambar 8.8 Continuous-time model

Model yang sesuai dijelaskan oleh persamaan diferensial $\dot{y} = -1/T y(t) + G u(t)$ (8.3.1) atau oleh fungsi transfer $H(s) = G / (1 + sT)$ (8.3.2)

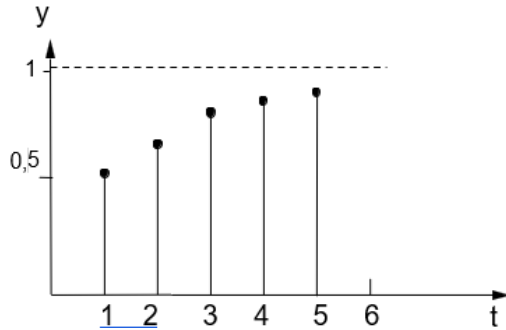
di mana T adalah konstanta waktu sistem dan G adalah gain. Jika input $u(t)$ dan output $y(t)$ disampel dengan periode pengambilan sampel tertentu, representasi $u(t)$ dan $y(t)$ diperoleh sebagai deret angka di mana t (atau k) sekarang adalah waktu diskrit yang dinormalkan (waktu riil dibagi dengan periode pengambilan sampel, $t = t/T_s$). Hubungan antara deret input $\{u(t)\}$ dan deret output $\{y(t)\}$ dapat disimulasikan oleh skema yang diberikan pada Gambar 8.9 dengan menggunakan operator tunda (pergeseran mundur) (dilambangkan dengan q^{-1} : $y(t-1) = q^{-1} y(t)$), alih-alih integrator. Hubungan ini dijelaskan dalam domain waktu oleh algoritma (dikenal sebagai persamaan rekursif atau persamaan perbedaan).

$$y(t) = -a_1 y(t-1) + b_1 u(t-1) \quad (8.3.3)$$

Mari kita telaah lebih rinci model waktu diskrit yang diberikan oleh Persamaan 2.3.3 untuk kondisi awal nol ($y(0) = 0$) dan masukan langkah satuan waktu diskrit: $u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$ Respons dihitung secara langsung dengan menggunakan Persamaan 2.3.3 secara rekursif dari $t = 0$ (dalam kasus model waktu diskrit tidak ada masalah dengan integrasi persamaan diferensial seperti dalam waktu kontinu). Kita akan memeriksa dua kasus. Kasus 1. $a_1 = -0,5$; $b_1 = 0,5$ Nilai keluaran untuk berbagai saat diberikan dalam Tabel 2.2 dan urutan yang sesuai direpresentasikan dalam Gambar 8.10.

Tabel 8.2. Respon langkah model waktu diskrit orde pertama ($a_1 = -0,5$, $b_1 = 0,5$)

T	0	1	2	3	4	5
y(t)	0	0.5	0.75	0.875	0.937	0.969



Gambar 8.10 Step response of a first-order discrete- time model ($a_1 = -0.5, b_1 = 0.5$)

Dapat diamati bahwa respons yang diperoleh menyerupai respons langkah dari sistem orde pertama waktu kontinu yang telah diambil sampelnya. Konstanta waktu ekuivalen untuk sistem waktu kontinu bahkan dapat ditentukan (waktu naik dari 0 hingga 90 %: $t_R = 2,2 T$). Dari Tabel 2.2, di mana dapat diperoleh:

$$\frac{3T_s}{2.2} < T < \frac{4T_s}{2.2}$$

Kasus 2:

$$a_1 = 0.5 ; \quad b_1 = 1.5$$

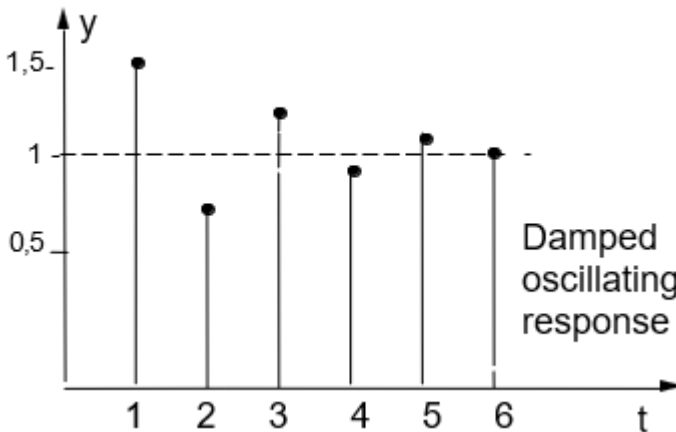
Nilai keluaran untuk berbagai saat diberikan dalam Tabel 8.3 dan urutan yang sesuai direpresentasikan dalam Gambar 8.11.

Tabel 8.3. Respon langkah model waktu diskrit orde pertama ($a_1=0,5; b_1=1,5$)

T	0	1	2	3	4	5
y(t)	0	1.5	0.75	1.125	0.937	1.062

Respons teredam osilasi diamati dengan periode yang sama dengan dua periode pengambilan sampel. Jenis fenomena ini tidak dapat dihasilkan dari diskritisasi sistem orde pertama waktu

kontinu, karena yang terakhir ini selalu a-periodik. Dengan demikian, dapat disimpulkan bahwa model waktu diskrit orde pertama sesuai dengan diskritisasi sistem waktu kontinu orde pertama hanya jika a_1 negatif¹. Respons teredam osilasi diamati dengan periode yang sama dengan dua periode pengambilan sampel. Jenis fenomena ini tidak dapat dihasilkan dari diskritisasi sistem orde pertama waktu kontinu, karena yang terakhir ini selalu a-periodik. Dengan demikian, dapat disimpulkan bahwa model waktu diskrit orde pertama sesuai dengan diskritisasi sistem waktu kontinu orde pertama hanya jika a_1 negatif¹.



Gambar 8.11. Respon langkah model waktu diskrit orde pertama ($a_1=0,5$; $b_1=1,5$)

Sekarang kita kembali ke metode yang digunakan untuk menggambarkan model waktu diskrit. Operator tunda q^{-1} digunakan untuk memperoleh penulisan persamaan rekursif (selisih) yang lebih ringkas yang menggambarkan model waktu diskrit dalam domain waktu (memiliki fungsi yang sama dengan operator $p = d/dt$ untuk sistem waktu kontinu). Hubungan berikut berlaku:

$$q^{-1} y(t) = y(t-1)$$

$$q^{-d} y(t) = y(t-d) \quad (8.3.4)$$

Dengan menggunakan operator q^{-1} , Persamaan 8.3.3 ditulis ulang menjadi:

$$(1 + a_1 q^{-1}) y(t) = b_1 q^{-1} u(t) \quad (8.3.5)$$

Model waktu diskrit juga dapat diperoleh dengan mendiskritkan persamaan diferensial yang menggambarkan model waktu kontinu. Operasi ini digunakan untuk simulasi model waktu kontinu pada komputer digital.

Mari kita perhatikan Persamaan 8.3.1 dan mendekati turunannya dengan :

$$\frac{dy}{dt} = \frac{y(t + T_s) - y(t)}{T_s} \tag{8.3.6}$$

Bisa di tulis dengan :

$$\frac{y(t + T_s) - y(t)}{T_s} + \frac{1}{T} y(t) = \frac{G}{T} u(t) \tag{8.3.7}$$

Dengan mengalikan kedua sisi Persamaan 8.3.7 dengan T_s , dan dengan diperkenalkannya waktu t yang dinormalisasi ($= t/T_s$), maka diperoleh bahwa:

$$y(t+1) + \left(\frac{T_s}{T} - 1 \right) y(t) = \frac{G}{T} T_s u(t) \tag{8.3.8}$$

yang dapat ditulis ulang lebih lanjut sebagai:

$$(1 + a_1 q^{-1}) y(t+1) = b_1 u(t)$$

diamana:

$$a_1 = \frac{T_s}{T} - 1 \quad (< 0); \quad b_1 = \frac{G}{T} T_s$$

Dengan menggeser Persamaan 8.3.9 satu langkah, diperoleh Persamaan 8.3.3. Kami menunjukkan bahwa, untuk merepresentasikan model kontinu orde pertama dengan Persamaan 8.3.9, kondisi $a_1 < 0$ harus diverifikasi. Akibatnya, periode pengambilan sampel T_s harus lebih kecil dari konstanta waktu T ($T_s < T$). Hasil ini sesuai dengan batas atas dalam Persamaan 8.2.4, yang diperkenalkan untuk pemilihan periode pengambilan sampel sistem orde pertama sebagai fungsi lebar pita loop tertutup yang diinginkan.

Jika Persamaan 8.3.6 merupakan perkiraan dari “turunan”, persamaan integrator digital dapat langsung disimpulkan. Jadi, jika waktu yang dinormalisasi digunakan, Persamaan 8.3.6 ditulis sebagai :

$$\frac{d}{dt} y = \underline{py} \approx y(t) - \underline{y}(t-1) = (1 - q^{-1})y(t)$$

8.3.10

di mana $(1 - q^{-1})$ sekarang setara dengan p . Karena integrasi adalah kebalikan dari diferensiasi, maka diperoleh :

$$s(t) = \int y dt = \frac{1}{p} y \approx \frac{1}{1 - q^{-1}} y(t)$$

8.3.11

Dengan mengalikan kedua sisi Persamaan 2.3.11 dengan $(1 - q^{-1})$, maka diperoleh:

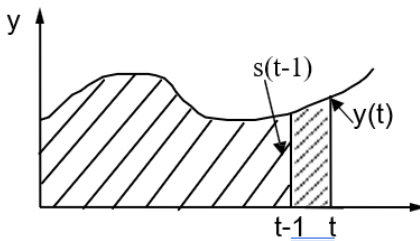
$$s(t) (1 - q^{-1}) = y(t)$$

8.3.12

yang dapat kita tulis ulang menjadi:

$$s(t) = s(t-1) + \underline{1} \cdot y(t) \quad 8.3.13$$

sesuai dengan perkiraan operasi integrasi dengan menggunakan aturan persegi panjang, seperti yang diilustrasikan pada Gambar 2.12 (jika waktu kontinu digunakan, Persamaan 8.3.13 ditulis sebagai $s(t) = s(t-T_s) + T_s \cdot y(t)$)



Gambar 8.12 Integrasi Numerical

2. Domain Frekuensi

Penelitian model waktu kontinu dalam domain frekuensi telah dilakukan dengan mempertimbangkan input periodik tipe eksponensial kompleks.

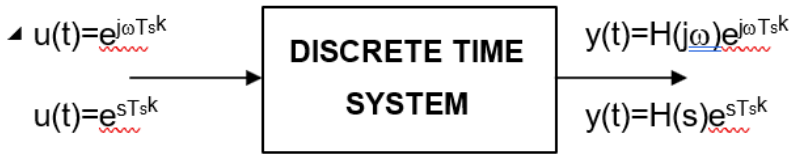
$$e^{j\omega t} = \cos \omega t + j \sin \omega t$$

atau est dengan $s = \sigma + j\omega$. Untuk mempelajari model waktu diskrit dalam domain frekuensi, kita akan mempertimbangkan eksponensial kompleks (sampel), yaitu urutan yang dihasilkan dari eksponensial waktu kontinu kompleks yang dievaluasi pada saat

pengambilan sampel $t = k T_s$. Dengan demikian, urutan ini akan ditulis sebagai:

$$e^{j\omega T_s k}; e^{s T_s k}; k=1,2,3\dots$$

Karena model waktu diskrit yang dipertimbangkan bersifat linier, jika sinyal dengan frekuensi tertentu diterapkan pada input, sinyal dengan frekuensi yang sama, tetapi diperkuat atau dilemahkan sesuai dengan frekuensi, akan ditemukan pada output. Hal ini dirangkum dalam Gambar 2.13. di mana $H(s)$ adalah "fungsi transfer" dari sistem yang menyatakan ketergantungan gain dan deviasi fase pada frekuensi kompleks s ($s = \sigma + j\omega$).



Gambar 8.13. Respon frekuensi sistem waktu diskrit

Jika input sistem dalam bentuk $e^{s T_s k}$, outputnya akan berupa:

$$y(t) = H(s) e^{s T_s k} \tag{8.3.14}$$

dan masing-masing adalah:

$$y(t - 1) = H(s) e^{s T_s (k-1)} = e^{-s T_s} H(s) e^{s T_s k} = e^{-s T_s} y(t) \tag{8.3.15}$$

Dengan demikian dapat diamati bahwa menggeser ke belakang satu langkah sama dengan mengalikan dengan $e^{-s T_s}$.

Mari kita tentukan fungsi transfer yang terkait dengan Persamaan rekursif 8.3.3. Dalam kasus ini $u(t) = e^{sT_s} k$ dan output akan berbentuk Persamaan 8.3.14. Dengan juga menggunakan Persamaan 8.3.15, kita memperoleh:

$$(1 + a_1 e^{-sT_s}) H(s) e^{sT_s k} = b_1 e^{-sT_s} e^{sT_s k} \tag{8.3.16}$$

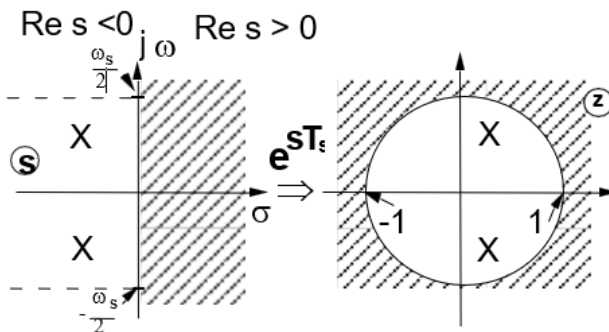
Dimana :

$$H(s) = \frac{b_1 e^{-sT_s}}{1 + a_1 e^{-sT_s}} \tag{8.3.17}$$

mari kita perhatikan perubahan variabel berikut:

$$z = e^{sT_s} \tag{8.3.18}$$

yang berhubungan dengan transformasi setengah bidang kiri bidang s ke bagian dalam lingkaran satuan yang berpusat di titik asal pada bidang z , seperti yang diilustrasikan pada Gambar 8.14.



Gambar 8.14. Pengaruh transformasi $z = e^{sTs}$

Dengan transformasi yang diberikan oleh Persamaan 8.3.18 fungsi transfer yang diberikan pada Persamaan 8.3.17 menjadi :

$$H(z^{-1}) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} \quad 8.3.19$$

Perhatikan bahwa fungsi transfer dalam z^{-1} dapat diperoleh secara langsung dari Persamaan rekursif 2.3.3 dengan menggunakan operator tunda q^{-1} (lihat Persamaan 8.3.5), dan setelah itu dengan menghitung rasio $y(t)/u(t)$ secara formal dan mengganti q^{-1} dengan z^{-1} . Prosedur ini jelas dapat diterapkan pada semua model yang dijelaskan oleh persamaan perbedaan linier dengan koefisien konstan, terlepas dari kompleksitasnya. Hasil yang sama juga dapat diperoleh melalui transformasi z .

Diketahui bahwa fungsi transfer model waktu diskrit sering ditulis dalam bentuk q^{-1} . Tentu saja dipahami bahwa makna q^{-1} bervariasi menurut konteks (operator tunda atau variabel kompleks). Ketika q^{-1} dianggap sebagai operator tunda, ekspresi

KEGIATAN BELAJAR XVII

PERANGKAT I/O DAN PENYIMPANAN

Oleh: Tuti Handayani, S.Kom., M.Kom.

DESKRIPSI PEMBELAJARAN

Pada bab ini mahasiswa mempelajari tentang definisi *input / output*, perangkat yang dibutuhkan sistem I/O, desain sistem I/O, lapisan dalam sistem I/O, Teknik I/O, *storage*, diharapkan mahasiswa memiliki wawasan dan pemahaman untuk sebagai modal dasar mempelajari matakuliah organisasi dan arsitektur komputer.

KOMPETENSI PEMBELAJARAN

Setelah mengikuti materi matakuliah ini diharapkan mahasiswa dan mahasiswi memiliki wawasan dan pemahaman :

1. Mampu mengetahui definisi *input / output*
2. Mampu menjelaskan perangkat yang dibutuhkan sistem I/O
3. Mampu menjelaskan desain sistem I/O
4. Mampu menjelaskan lapisan dalam sistem I/O
5. Mampu menjelaskan teknik I/O
6. Mampu menjelaskan tentang storage / penyimpanan

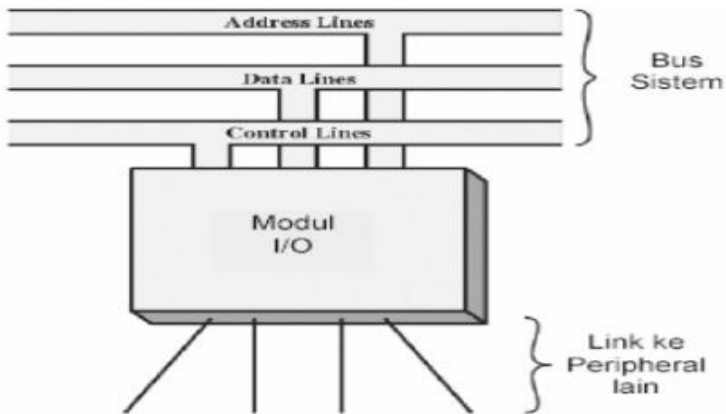
A. DEFINISI INPUT / OUTPUT (I/O).

Untuk berhubungan dengan dunia luar, maka mikroprosesor menggunakan bagian dari sistem mikroprosesor. Unit *input* (masukan) adalah data yang dimasukkan dari luar ke dalam mikroprosesor yaitu unit luar yang digunakan adalah data yang berasal dari *mouse* atau *keyboard*. Sedangkan unit output (*keluaran*), digunakan untuk menampilkan data. Contohnya pada layar monitor atau printer ditampilkan datanya.

B. SISTEM INPUT / OUTPUT (I/O).

Hal terpenting yang perlu kita ketahui adalah modul I/O dapat menjalankan tugasnya yaitu menjembatani antara memori dan *Central Processing Unit (CPU)* dengan dunia luar.

Gambar di bawah ini menyajikan model generik modul I/O.



Gambar 17.1: Model generik modul I/O.

Modul I/O adalah suatu komponen dalam sistem komputer:

1. Antarmuka dengan perangkat eksternalnya untuk menjalankan fungsi-fungsi pengontrolan.
2. Antarmuka internal dengan komputer (memori utama dan *Central Processing Unit / CPU*).
3. Bertanggungjawab dalam pertukaran data antara perangkat luar tersebut dengan memori utama ataupun register-register *Central Processing Unit (CPU)*.
4. Bertanggungjawab atas pengontrolan sebuah perangkat luar.

Dua fungsi utama modul I/O adalah :

1. Sebagai piranti antarmuka dengan peralatan periferil lainnya dengan menggunakan *link* data tertentu.
2. Sebagai piranti antarmuka ke *Central Processing Unit (CPU)* dan memori melalui *bus* sistem.

Untuk lebih jelasnya dapat dilihat pada gambar 17.1 diatas.

Fungsi modul I/O :

1. Pem-*buffer*-an data.

Tujuan utamanya adalah mendapatkan penyesuaian data sehubungan perbedaan laju transfer data dari perangkat periferil dengan kecepatan pengolahan pada *Central Processing Unit (CPU)*. Kecepatan laju transfer data dari

Central Processing Unit (CPU) maupun media penyimpan lebih cepat dibandingkan dengan perangkat periferal.

2. Komunikasi perangkat eksternal.
3. Komunikasi *Central Processing Unit (CPU)*.
4. Kontrol dan pewaktuan.

Ini merupakan hal terpenting untuk mensinkronkan kerja masing-masing komponen penyusun komputer. Berikut adalah langkah-langkah kontrol pemindahan data dari periferal ke *Central Processing Unit (CPU)* melalui sebuah modul I/O:

- a. Permintaan dan pemeriksaan status perangkat dari *Central Processing Unit (CPU)* ke modul I/O.
 - b. Modul I/O memberi jawaban atas permintaan *Central Processing Unit (CPU)*.
 - c. Apabila perangkat eksternal telah siap untuk transfer data, maka *Central Processing Unit (CPU)* akan mengirimkan perintah ke modul I/O.
 - d. Modul I/O akan menerima paket data dengan panjang tertentu dari periferal.
 - e. Selanjutnya data dikirim ke *Central Processing Unit (CPU)* setelah diadakan sinkronisasi panjang data dan kecepatan transfer oleh modul I/O, sehingga paket-paket data dapat diterima *Central Processing Unit (CPU)* dengan baik.
5. Deteksi kesalahan.
Modul I/O akan melaporkan kesalahan apabila perangkat periferal terdapat masalah sehingga proses tidak dapat dijalankan. Misalnya, informasi kesalahan pada periferal printer, yaitu tinta habis, kertas tergulung, kertas habis. Penggunaan bit paritas adalah teknik umum untuk deteksi kesalahan.

C. PERANGKAT YANG DIBUTUHKAN SISTEM I/O

Tabel. 17.1. Perangkat I/O beserta kebutuhan bandwidth (per 2012).

Device	Bus Technology	Bandwidth	Typical Values
Video display via graphics card	PCI Express (version 4)	High	1-10 GB/s
Hard disks	ATA/SCSI/SAS	Medium	150-600 MB/s
Network card (wired/wireless)	PCI Express	Medium	10-100 MB/s
USB devices	USB	Medium	60-625 MB/s
DVD audio/video	PCI	Medium	1-4 MB/s
Speaker/Microphone	AC'97/Intel High. Def. Audio	Low	100 KB/s to 3 MB/s
Keyboard/Mouse	USB/PCI	Very Low	10-100 B/s

Mari kita merancang arsitektur sistem I/O dengan membuat kebutuhan *bandwidth* dan daftar semua perangkat. Pada tabel 17.1, perangkat tampilan (proyektor, TV, monitor) yaitu komponen yang memerlukan jumlah *bandwidth* maksimal. Ketiga perangkat tampilan tersebut terpasang ke kartu grafis. Kartu grafis yaitu data gambar dan video yang diproses oleh grafik prosesor.



Gambar 17.2. Network Interface Card (NIC).

Dalam pembahasan I/O, kita akan sering menggunakan istilah kartu. Untuk memproses gambar dan video, kita dibantu dengan kartu grafis. Untuk memproses audio, kita dibantu dengan kartu suara. Pada gambar 17.2, kita dapat melihat satu set *chip* yang saling berhubungan pada papan sirkuit.

Perangkat eksternal yang menghubungkan ke kartu menggunakan sekumpulan *port*.

Memori utama merupakan perangkat bandwidth yang perlu dihubungkan ke *Central Processing Unit (CPU)*.

DAFTAR PUSTAKA

- Abdelzaher, T., et al. (2015). "The Role of Artificial Intelligence in the Internet of Things." *ACM Transactions on Internet Technology*, 15(2), 1-27. doi:10.1145/2738120.
- Abdul Kadir, T.C.T. (2013) *Pengenalan Teknologi Informasi*. Yogyakarta., Pustaka.Ut.Ac.Id. Edited by A. Riyanto. Yogyakarta. Available at: https://books.google.com/books?hl=en&lr=&id=opb_DwAAQBAJ&oi=fnd&pg=PA71&dq=%22keamanan+jaringan%22&ots=cW_kQe4rwA&sig=evrqQBWhu0IILySGc5L8pqH1xmU.
- Abdurohman, M. (2016). Peningkatan Performansi Prosesor Dlx Dengan Metode Pipeline. *Teknika - Jurnal Penelitian Dan Pengembangan Telekomunikasi, Kendali, Komputer, Elektrik, Dan Elektronika*, 8(2). <https://doi.org/10.25124/tekrika.v8i2.230>
- Abdurohman, M. (2016). Peningkatan Performansi Prosesor Dlx Dengan Metode Pipeline. *Teknika - Jurnal Penelitian Dan Pengembangan Telekomunikasi, Kendali, Komputer, Elektrik, Dan Elektronika*, 8(2). <https://doi.org/10.25124/tekrika.v8i2.230>
- Adri, Muhammad. "P e n g a n t a r A r s i t e k t u r K o m p u t e r – 4 R a n g k a i a n A r i t m a t i k A." : 1–12.
- Agung Ngurah Gunawan, D., Ketut Putra, M.I., Wayan Supardi, Ms.I., Ilham, Ms., Made Satriya Wibawa, M.I., Fuadi, A., 2023. *Dasar Ilmu Komputer dan Jaringan*.
- Agustin, W.D., Maulana, A.D., Wirta, D., Aribowo, D., 2023. Studi Perbandingan Antara Memori DRAM dan Memori SRAM Dalam Sistem Keamanan Komputer. *Jurnal Teknik Mesin, Industri, Elektro dan Informatika* 2, 01–10. <https://doi.org/10.55606/JTMEI.V2I4.2747>
- Aiken, A., Banerjee, U., Kejariwal, A., & Nicolau, A. (2016). Instruction level parallelism. In *Instruction Level Parallelism*. <https://doi.org/10.1007/978-1-4899-7797-7>
- Aiken, A., Banerjee, U., Kejariwal, A., & Nicolau, A. (2016). Instruction level parallelism. In *Instruction Level Parallelism*. <https://doi.org/10.1007/978-1-4899-7797-7>

- Al-Dulaimy, A., Sharma, Y., Khan, M. G., & Taheri, J. (2020). Introduction to edge computing. In *Edge Computing*. https://doi.org/10.1049/PBPC033E_ch1
- Al-Dulaimy, A., Sharma, Y., Khan, M. G., & Taheri, J. (2020). Introduction to edge computing. In *Edge Computing*. https://doi.org/10.1049/PBPC033E_ch1
- Alifah, N., Veranda Deanda, G., Aribowo, D., Vokasional Teknik Elektro, P., & Keguruan dan Ilmu Pendidikan, F. (2023). Peran Teknologi Input dan Output dalam Pengembangan Perangkat Keras dan Perangkat Lunak Komputer. *Jurnal Kendali Teknik Dan Sains*, 1(4), 123–136. <https://doi.org/10.59581/jkts-widyakarya.v1i4>.
- Alifah, N., Veranda Deanda, G., Aribowo, D., Vokasional Teknik Elektro, P., & Keguruan dan Ilmu Pendidikan, F. (2023). Peran Teknologi Input dan Output dalam Pengembangan Perangkat Keras dan Perangkat Lunak Komputer. *Jurnal Kendali Teknik Dan Sains*, 1(4), 123–136. <https://doi.org/10.59581/jkts-widyakarya.v1i4>.
- Almorsy, M., et al. (2016). "Cloud Computing Security Issues and Challenges: A Survey." *Future Generation Computer Systems*, 54, 109-127. doi:10.1016/j.future.2015.07.005.
- Andaria, Alex Copernikus. 2024. 'Arsitektur Komputer'. Pp. 71–100 in *Pengantar Ilmu Komputer*. Agam: Lauk Puyu Press.
- Anderson B.D.O, Moore J.B. (1971) *Linear Optimal Control*, Prentice Hall, Englewood Cliffs, N.J.
- Armbrust, M., et al. (2010). "Above the Clouds: A Berkeley View of Cloud Computing." UC Berkeley Technical Report.
- Åström K.J., Wittenmark B. (1997) *Computer Controlled Systems - Theory and Design*, 3rd edition, Prentice-Hall, Englewood Cliffs, N.J.
- Bao, W., Hong, C., Chunduri, S., Krishnamoorthy, S., Pouchet, L. N., Rastello, F., & Sadayappan, P. (2016). Static and dynamic frequency scaling on multicore CPUs. *ACM Transactions on Architecture and Code Optimization*, 13(4). <https://doi.org/10.1145/3011017>
- Bao, W., Hong, C., Chunduri, S., Krishnamoorthy, S., Pouchet, L. N., Rastello, F., & Sadayappan, P. (2016). Static and dynamic frequency scaling on multicore CPUs. *ACM Transactions on Architecture and Code Optimization*, 13(4). <https://doi.org/10.1145/3011017>

- Barker, S. (2021). "Trends in Hardware Security: Past, Present, and Future." *Journal of Hardware and Systems Security*, 5(3), 213-226. doi:10.1007/s41635-021-00116-5.
- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. 3rd Edition. Addison-Wesley.
- Blissmer, Robert. H. 1985. *Computer Annual, An Introduction to Information System 1985-1986*. Jhon Wiley & Sons. New York.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2020). *The Unified Modeling Language User Guide*. 2nd Edition. Addison-Wesley.
- Bosch, J. (2021). *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. Addison-Wesley.
- Bouchez Tichadou, F., & Rastello, F. (2022). Register Allocation. In *SSA-based Compiler Design*. https://doi.org/10.1007/978-3-030-80515-9_22
- Bouchez Tichadou, F., & Rastello, F. (2022). Register Allocation. In *SSA-based Compiler Design*. https://doi.org/10.1007/978-3-030-80515-9_22
- Bourlès H., Irving E. (1991) La méthode LQG/LTR: une interprétation polynomiale, temps continu / temps discret, *RAIRO-APII*, Vol. 25, pp. 545- 568.
- Bowo, I.T., Kurniawan, F.I., Rachman, A.N., Joosten, Muchlis, I.R., 2024. *Buku Ajar Pengantar Ilmu Komputer* [WWW Document]. URL https://books.google.co.id/books?hl=id&lr=&id=KjAGEQAAQBAJ&oi=fnd&pg=PA31&dq=buku+memori+komputer&ots=SLcjT3LYge&sig=4X1n0rJ7E9dP_ZBESYIPGT6igTQ&redir_esc=y#v=onepage&q=buku%20memori%20komputer&f=false (accessed 9.10.24).
- Brey, Barry B. 2009. *The Intel Microprocessors: Architecture, Programming, and Interfacing*. Eighth Edition. Pearson Education, Inc.
- Brock, D.C. (2021) *Thinking about Machines and Thinking*. paris: premiere section. Available at: <https://computerhistory.org/blog/thinking-about-machines-and-thinking/>.
- BRYANT, R.E.. et al. (2016) *Computer systems: a programmer's perspective*. Pearson.
- Budi Santoso, A. (2021). *Buku Ajar Aplikasi Komputer*. In IAIN Manado Press.

- Budi Santoso, A. (2021). *Buku Ajar Aplikasi Komputer*. In IAIN Manado Press.
- Bustio-Martínez, L., Cumplido, R., Letras, M., Hernández-León, R., Feregrino-Uribe, C., & Hernández-Palancar, J. (2022). FPGA/GPU-based Acceleration for Frequent Itemsets Mining: A Comprehensive Review. In *ACM Computing Surveys* (Vol. 54, Issue 9). <https://doi.org/10.1145/3472289>
- Bustio-Martínez, L., Cumplido, R., Letras, M., Hernández-León, R., Feregrino-Uribe, C., & Hernández-Palancar, J. (2022). FPGA/GPU-based Acceleration for Frequent Itemsets Mining: A Comprehensive Review. In *ACM Computing Surveys* (Vol. 54, Issue 9). <https://doi.org/10.1145/3472289>
- Cacoullos, A., & Iliadis, L. (2019). "Cloud-Based Infrastructure for Internet of Things: A Review." *IEEE Internet of Things Journal*, 6(4), 7465-7476. doi:10.1109/JIOT.2019.2906351.
- Cahyaningrum, Y., Yuspita, Y. E., Diana, Sani, A., Devianto, Y., Trisudarmo, R., Sugianta, I. K. A., Budianto, H., Rahmawati, N., Avizenna, M. H., Fitri, N. A., Aditama, D., Jannah, M., & Effendi, Y. A. (2023). *Arsitektur dan Organisasi Komputer* (A. Arista (ed.); 1st ed.). Penamudamedia.
- Cahyaningrum, Y., Yuspita, Y. E., Diana, Sani, A., Devianto, Y., Trisudarmo, R., Sugianta, I. K. A., Budianto, H., Rahmawati, N., Avizenna, M. H., Fitri, N. A., Aditama, D., Jannah, M., & Effendi, Y. A. (2023). *Arsitektur dan Organisasi Komputer* (A. Arista (ed.); 1st ed.). Penamudamedia.
- Cahyaningrum, Y., Yuspita, Y.E., Diana, Sani, A., Devianto, Y., 2023. *Arsitektur Dan Organisasi Komputer* [WWW Document]. URL <http://repository.ikipgribojonegoro.ac.id/id/eprint/2426> (accessed 9.18.24).
- Cao, Y., et al. (2020). "Towards a User-Centric Approach to Cloud Service Management." *International Journal of Information Management*, 52, 102-116. doi:10.1016/j.ijinfomgt.2020.102116.
- Carminati, A., Starke, R. A., & Oliveira, R. S. de. (2018). On the use of static branch prediction to reduce the worst-case execution time of real-time applications. *Real-Time Systems*, 54(3). <https://doi.org/10.1007/s11241-018-9306-y>
- Carminati, A., Starke, R. A., & Oliveira, R. S. de. (2018). On the use of static branch prediction to reduce the worst-case execution time of real-time applications. *Real-Time Systems*, 54(3). <https://doi.org/10.1007/s11241-018-9306-y>

- Carter, B. and Brown, T.R. (2001) 'Handbook of Operational Amplifier Applications', (October), pp. 1–94.
- Chevance R.J. 2005. *Server Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, and Storage Solutions*. Amerika Serikat. Penerbit. Elsevier
- Chong, E., & Zulkernine, M. (2017). "A Survey of Cloud Security Issues and Challenges." *IEEE Communications Surveys & Tutorials*, 19(2), 1260-1280. doi:10.1109/COMST.2017.2651398.
- Cohen, L. (2017). "The Future of Computer Architecture: Challenges and Opportunities." *IEEE Computer Society*, 50(1), 29-37.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
- Davis, Gordon B. 1981. *Introduction to computers* / Gordon B. Davis. Aukland :: McGraw-Hill.
- Dawson, L. (2021). "User Interface Design for Mobile and Web Applications." *Journal of Usability Studies*, 16(3), 130-145.
- Dean, J., & Ghemawat, S. (2008). *MapReduce: Simplified Data Processing on Large Clusters*.
- Doyle J.C., Francis B.A., Tanenbaum A.R. (1992) *Feedback Control Theory*, Mac Millan, N.Y.
- Elahi , A. 2018. *Computer Systems : Digital Design, Fundamentals of Computer Architecture and Assembly Language*. Amerika Serikat. Penerbit. Springer
- Endraswari, P.M., 2023. *Bahan Ajar Pengantar Teknologi Informasi*.
- Englander I., 2009. *The architecture of computer hardware, system software, and networking: The Information Technology Approach*, Amerika Serikat. Penerbit. John Wiley & Sons
- Evans, E. (2021). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.
- Farah, H., Arjuna, F. T. F., Shafwan, I. R., & Didik, A. (2023). Pengaruh Kapasitas Memori RAM (Random Access Memory) Terhadap Kecepatan Memori Pada Laptop. *Jurnal Elektronika Dan Teknik Informatika Terapan (JENTIK)*, 1(4), 178–186.
- Farah, H., Arjuna, F. T. F., Shafwan, I. R., & Didik, A. (2023). Pengaruh Kapasitas Memori RAM (Random Access Memory) Terhadap Kecepatan Memori Pada Laptop. *Jurnal Elektronika Dan Teknik Informatika Terapan (JENTIK)*, 1(4), 178–186.

- Farizy, S., & Harianja, E. S. (2020). Pengembangan Media Penyimpanan dalam Sistem Berkas. | Jurnal Ilmu Komputer JIK, III(02), 5–9.
- Farizy, S., & Harianja, E. S. (2020). Pengembangan Media Penyimpanan dalam Sistem Berkas. | Jurnal Ilmu Komputer JIK, III(02), 5–9.
- Fauziah, Y. (2014) 'Arsitektur Cloud Computing Pada Sistem Informasi Desa Sebagai Layanan Akses Informasi Desa', in Seminar Nasional Informatika 2014, pp. 224–230.
- Fettes, Q., Clark, M., Bunescu, R., Karanth, A., & Louri, A. (2019). Dynamic Voltage and Frequency Scaling in NoCs with Supervised and Reinforcement Learning Techniques. *IEEE Transactions on Computers*, 68(3). <https://doi.org/10.1109/TC.2018.2875476>
- Fettes, Q., Clark, M., Bunescu, R., Karanth, A., & Louri, A. (2019). Dynamic Voltage and Frequency Scaling in NoCs with Supervised and Reinforcement Learning Techniques. *IEEE Transactions on Computers*, 68(3). <https://doi.org/10.1109/TC.2018.2875476>
- Feynman, R. P. (1998). *Feynman Lectures on Computation*. Addison-Wesley.
- Flynn ,M.J. & Rudd, K.W., 1996. *Parallel Architectures* , ACM Computing Surveys, Vol. 28, No. 1, March 1996 https://www.researchgate.net/publication/220267962_On_Parallel_Architectures
- For a generalization of the concepts of robustness margins and robust stability introduced in this chapter, see Appendix D.
- For an introduction to robust control theory see:
- Fowler, M. (2018). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- Franklin G.F., Powell J.D., Workman M.L. (1998) *Digital Control of Dynamic Systems* , 3rd edition, Addison Wesley, Reading, Mass.
- Freeman, E., & Robson, E. (2020). *Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software*. 2nd Edition. O'Reilly Media.
- Furht, B., & Escalante, A. (2016). *Handbook of Cloud Computing*. Springer.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

- Garlan, D., & Shaw, M. (2021). *An Introduction to Software Architecture*. Addison-Wesley.
- Green, K. P. (2021). "A Comprehensive Study on Future Computing Architectures." *IEEE Access*, 9, 115524-115541. doi:10.1109/ACCESS.2021.3100130.
- Hall.
- HENNESSY, H.L. And PATTERSON, D.A. (2012) *Computer Architecture: A Quantitative Approach Fifth Edition*.
- Hennessy, John L., and David A. Patterson. 2012. *Computer Architecture: A Quantitative Approach*. Fifth Edition. Elsevier, Inc.
- Heritage, E. (2023) FLOWERS, Tommy (1905–1998). Available at: <https://www.english-heritage.org.uk/visit/blue-plaques/tommy-flowers/>.
- Hidayah, A. N. (2018). Implementasi Parallel Computing untuk Mengoptimalkan Komputasi Pada Aplikasi Transliterasi Huruf Latin Ke Aksara Jawa. In Universitas Islam Negeri Maulana Malik Ibrahim (Vol. 1, Issue 1).
- Hidayah, A. N. (2018). Implementasi Parallel Computing untuk Mengoptimalkan Komputasi Pada Aplikasi Transliterasi Huruf Latin Ke Aksara Jawa. In Universitas Islam Negeri Maulana Malik Ibrahim (Vol. 1, Issue 1).
- Hohpe, G., & Woolf, B. (2012). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley.
- Hopcroft et al., 2006 Hopcroft, J. E., Motwani, R., and Ulmann, J. D. 2006. *Introduction to Automata Theory, Languages, and Computation*. Prentice
- Huang, Y., & Wang, H. (2018). "Towards a Sustainable Future: Energy-Efficient Cloud Computing." *Sustainable Computing: Informatics and Systems*, 18, 10-23. doi:10.1016/j.suscom.2017.07.001.
- Huda, N., 2022. *Memori Komputer: Pengertian, Jenis-Jenis, dan Fungsinya* [WWW Document]. URL <https://www.dewaweb.com/blog/pengertian-memori-komputer/> (accessed 8.30.24).
- Husson, 1970 Husson, S. S. 1970. *Microprogramming: Principles and Practices*. Prentice
- Hwang, 200] Hwang, K. 2003. *Advanced computer architecture*. Tata McGraw-Hill Education.
- IEEE. (1985). *IEEE Standard for Floating-Point Arithmetic (IEEE 754)*.

- Ielmini, D., & Wong, H. S. P. (2018). In-memory computing with resistive switching devices. In *Nature Electronics* (Vol. 1, Issue 6). <https://doi.org/10.1038/s41928-018-0092-2>
- Ielmini, D., & Wong, H. S. P. (2018). In-memory computing with resistive switching devices. In *Nature Electronics* (Vol. 1, Issue 6). <https://doi.org/10.1038/s41928-018-0092-2>
- IoT Analytics. (2022). State of the IoT.
- Jacob et al., 2007 Jacob, B., Ng, S., and Wang, D. 2007. *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann.
- Jacob, 2009 Jacob, B. 2009. The memory system: you can't avoid it, you can't ignore it, you can't fake it. *Synthesis Lectures on Computer Architecture*, 4(1):1–77.
- Jansen, A., & Bosch, J. (2020). *Software Architecture as a Set of Architectural Design Decisions*. Springer.
- Janurianto, A., & Suhendra, A. (2012). Analisis Perbandingan Komputasi Sekuensial Dan Komputasi Paralel Gpu Memanfaatkan Teknologi Nvidia Cuda Pada Aplikasi Kompresi Citra Menggunakan Algoritma Dct 8x8. *E-Journal Teknologi Industri*.
- Janurianto, A., & Suhendra, A. (2012). Analisis Perbandingan Komputasi Sekuensial Dan Komputasi Paralel Gpu Memanfaatkan Teknologi Nvidia Cuda Pada Aplikasi Kompresi Citra Menggunakan Algoritma Dct 8x8. *E-Journal Teknologi Industri*.
- Jarti Nanda. CV.Eureka Media Aksara “Organisasi dan Arsitektur komputer”
- Jerger and Peh, 2009] Jerger, N. E. and Peh, L.-S. 2009. On-chip networks. *Synthesis Lectures on Computer Architecture*, 4(1):1–14
- Jogiyanto Hartono, MBA, Ph.D. 1999. *Pengenalan Komputer*. Edisi kedua. Yogyakarta: Penerbit ANDI.
- Kanamori, Y., & Yoo, S.-M. (2020). Quantum Computing: Principles and Applications. *Journal of International Technology and Information Management*, 29(2). <https://doi.org/10.58729/1941-6679.1410>
- Kanamori, Y., & Yoo, S.-M. (2020). Quantum Computing: Principles and Applications. *Journal of International Technology and Information Management*, 29(2). <https://doi.org/10.58729/1941-6679.1410>
- Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography*. CRC Press.

- Katz, R. H., & Patterson, D. A. (2021). *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann.
- Kazman, R., & Bass, L. (2020). *Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler's Children*. 2nd Edition. Pearson.
- Khan, M. A., et al. (2019). "AI-Powered Smart Applications: Future Trends and Challenges." *Future Generation Computer Systems*, 100, 142-154. doi:10.1016/j.future.2019.05.024.
- Klein, M. (2021). *Designing Software Architectures: A Practical Approach*. Addison-Wesley.
- Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., & Yarom, Y. (2020). Spectre attacks: Exploiting Speculative Execution. *Communications of the ACM*, 63(7). <https://doi.org/10.1145/3399742>
- Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., & Yarom, Y. (2020). Spectre attacks: Exploiting Speculative Execution. *Communications of the ACM*, 63(7). <https://doi.org/10.1145/3399742>
- Komalasari, R., Widians, J.A., Meiliani, B.D., Arifin, N.Y., Sepriano, 2023. *Pengantar Ilmu Komputer: Teori Komprehensif Perkembangan Ilmu Komputer Terkini [WWW Document]*. URL https://books.google.co.id/books?hl=id&lr=&id=znHVEAAAQBAJ&oi=fnd&pg=PA8&dq=memori+komputer+adalah&ots=tCFOaM_DeT&sig=rxfxkuMtwJij1dVCP6l8NbP6PM&redir_esc=y#v=onepage&q=memori%20komputer%20adalah&f=false (accessed 9.10.24).
- Komputer, Aritmatika, Eko Saksi Pramukantoro, S Kom, and M Kom. "Modul 9."
- Komputer, Fakultas Ilmu, Universitas Hang, and Tuah Pekanbaru. "Organisasi Dan Arsitektur Komputer."
- Koomey, J. G. (2011). "Growth in Data Center Electricity Use 2005 to 2010." *Analytics Press*.
- Krol, D. (2019). "The Role of AI in Enhancing User Experience." *Journal of Interactive Media*, 7(4), 45-56.
- Kukunas, J. (2015). Power and Performance: Software Analysis and Optimization. In *Power and Performance: Software*

- Analysis and Optimization. <https://doi.org/10.1016/C2013-0-18946-5>
- Kukunas, J. (2015). Power and Performance: Software Analysis and Optimization. In *Power and Performance: Software Analysis and Optimization*. <https://doi.org/10.1016/C2013-0-18946-5>
- Kuo B. (1980) *Digital Control Systems*, Holt Saunders, Tokyo.
- Kurose, J. F., & Ross, K. W. (2017). *Computer Networking: A Top-Down Approach*. Pearson.
- Kwakernaak H. (1993) Robust control and Hinf optimization – a tutorial *Automatica*, vol.29, pp.255-273.
- Landau I.D. (1995) Robust digital control systems with time delay (the Smith predictor revisited), *Int. J. of Control*, Vol. 62, no. 2, pp. 325-347.
- Leffingwell, D. (2020). *SAFe 4.5 Distilled: Applying the Scaled Agile Framework for Lean Software and Systems Engineering*. Addison-Wesley.
- Legito, L., Subekti, R., Ardiada, M.D., Kusuma, A.T.A.P., Efrita, 2023. *Buku Ajar Pengantar Ilmu Komputer* [WWW Document]. URL https://books.google.co.id/books?hl=id&lr=&id=nLbaEAAQBAJ&oi=fnd&pg=PA24&dq=memori+komputer+adalah&ots=N2xVbZNaVz&sig=otweSXOXWeRFdiJolPbUuzyATgw&redir_esc=y#v=onepage&q=memori%20komputer%20adalah&f=false (accessed 9.10.24).
- Liu, X., et al. (2017). "Designing User-Centric Cloud Computing Services: A Review." *Journal of Cloud Computing: Advances, Systems and Applications*, 6(1), 1-12.
- M. Morris Mano, and Charles Kime. 2014. *Logic and Computer Design Fundamentals*. Fourth Edition. United States of America: Pearson Education Limited.
- MAESTRI, S., MERELLI, E. and PETTINI, M. (2022) Agent-based models for detecting the driving forces of biomolecular interactions. *Scientific Reports*, 12(1), [Online] Available from: doi.org/10.1038/s41598-021-04205-8.
- Maharani, N. Z., Luthfi, R. Z., Amri, R., & Ginting, R. E. (2023). Analisis Strategi Peningkatan Daya Kerja Central Processing Unit (CPU) Dalam Proses Pengolahan Data. *Journal Of Informatics And Busisnes*, 1(3), 194–200.
- Maharani, N. Z., Luthfi, R. Z., Amri, R., & Ginting, R. E. (2023). Analisis Strategi Peningkatan Daya Kerja Central Processing

- Unit (CPU) Dalam Proses Pengolahan Data. *Journal Of Informatics And Busines*, 1(3), 194–200.
- Manno, Morris. 2017. *Computer System Architecture*. third-edition.
- Martin, R. C. (2009). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.
- Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
- Maulana, M. S. (n.d.). *SISTEM OPERASI*.
- Maulana, M. S. (n.d.). *SISTEM OPERASI*.
- Maulana, P. A., Habibi, M. Y., Gilang Hermawan, A., & Aribowo, D. (2023). Analisis Sistem Bus USB Dan PCI Pada Organisasi Arsitektur Komputer. *Jurnal Kendali Teknik Dan Sains*, 1(4), 115–122. <https://doi.org/10.59581/jkts-widyakarya.v1i4.1463>
- Maulana, P. A., Habibi, M. Y., Gilang Hermawan, A., & Aribowo, D. (2023). Analisis Sistem Bus USB Dan PCI Pada Organisasi Arsitektur Komputer. *Jurnal Kendali Teknik Dan Sains*, 1(4), 115–122. <https://doi.org/10.59581/jkts-widyakarya.v1i4.1463>
- Mell, P., & Grance, T. (2011). "The NIST Definition of Cloud Computing." National Institute of Standards and Technology.
- Mittal, S. (2019). A survey of techniques for dynamic branch prediction. *Concurrency and Computation: Practice and Experience*, 31(1). <https://doi.org/10.1002/cpe.4666>
- Mittal, S. (2019). A survey of techniques for dynamic branch prediction. *Concurrency and Computation: Practice and Experience*, 31(1). <https://doi.org/10.1002/cpe.4666>
- Mogensen, T. Æ. (2024). *Introduction to Compiler Design*. <https://link.springer.com/book/10.1007/978-3-031-46460-7>
- Mogensen, T. Æ. (2024). *Introduction to Compiler Design*. <https://link.springer.com/book/10.1007/978-3-031-46460-7>
- Mohammad Suryawinata, UMSIDA PRESS." *Arsitektur dan Organisasi Komputer*"
- Morari M., Zafiriou E. (1989) *Robust Process Control*, Prentice Hall International, Englewood Cliffs, N.J.
- Mufida, E., Leidiyana, H., Rachamawati, E., & Hertyana, H. (2021). *Arsitektur Komputer Struktur dan Fungsi*. <https://repository.nusamandiri.ac.id/repo/files/242358/download/Arsitektur-Komputer-Maret-2021.pdf>
- Mufida, E., Leidiyana, H., Rachamawati, E., & Hertyana, H. (2021). *Arsitektur Komputer Struktur dan Fungsi*. <https://repository.nusamandiri.ac.id/repo/files/242358/download/Arsitektur-Komputer-Maret-2021.pdf>

- Narendra K.S., Taylor J.H. (1973) Frequency Domain Criteria for Absolute Stability, Academic Press, New York.
- Nash, J. (2022). "Future Trends in Computer Architecture." IEEE Transactions on Computers, 71(6), 839-853. doi:10.1109/TC.2022.3142612.
- Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information. Cambridge University Press.
- Nurchaliza, R. (2024). Membedah Organisasi dan Arsitektur Komputer: Inti Dunia Digital. <https://dif.telkomuniversity.ac.id/membedah-organisasi-dan-arsitektur-komputer-inti-dunia-digital/>
- Nurchaliza, R. (2024). Membedah Organisasi dan Arsitektur Komputer: Inti Dunia Digital. <https://dif.telkomuniversity.ac.id/membedah-organisasi-dan-arsitektur-komputer-inti-dunia-digital/>
- Nurwasito, Heri. Henryranu P, Barlian. Pramukantoro, Eko Saksi. (2012). Available at https://eling.ub.ac.id/pluginfile.php/110787/mod_resource/content/1/BAB%207.pdf, tahun 2012
- Ogata K. (1987) Discrete-Time Control Systems, Prentice Hall, N.J.
- Olin-Ammentorp, W., & Cady, N. (2019). Biologically-Inspired Neuromorphic Computing. Science Progress, 102(3). <https://doi.org/10.1177/0036850419850394>
- Olin-Ammentorp, W., & Cady, N. (2019). Biologically-Inspired Neuromorphic Computing. Science Progress, 102(3). <https://doi.org/10.1177/0036850419850394>
- Parihar, R. (2017). Branch Prediction Techniques and Optimizations. University of Rochester, USA.
- Parihar, R. (2017). Branch Prediction Techniques and Optimizations. University of Rochester, USA.
- Patterson, D. A., & Hennessy, J. L. (2017). Computer Organization and Design: The Hardware/Software Interface. Morgan Kaufmann.
- Patterson, D. A., & Hennessy, J. L. (2017). Computer Organization and Design RISC-V Edition: The Hardware Software Interface. Morgan Kaufmann.
- Patterson, David A., and John L. Hennessy. 2014. Computer Organization and Design: The Hardware/ Software Interface. Fifth Edition. Elsevier Inc.
- Phillips, C.L., Nagle, H.T. (1995) Digital Control Systems Analysis and Design, 3rd edition, Prentice Hall, N.J.

- Plunkett, K. (2023). Parallel distributed processing. *Psyke & Logos*, 9(2). <https://doi.org/10.7146/pl.v9i2.134261>
- Plunkett, K. (2023). Parallel distributed processing. *Psyke & Logos*, 9(2). <https://doi.org/10.7146/pl.v9i2.134261>
- Poundstone, W. (2024) World War II, *britannica*. Available at: <https://www.britannica.com/biography/John-von-Neumann#ref256592>.
- Pranata, K.B. and Sundaygara, C. (2018) 'Buku Ajar Mata Kuliah Elektronika Dasar 1', *Panduan Pratikum Elektronika Dasar 1*, pp. 1–131.
- Prasetio, B.H., Widasari, E.R., Ichsan, M.H.H., Setiawan, E., Syauqy, D., 2022. *Sistem Digital : Teori dan Aplikasi* [WWW Document]. URL https://www.google.co.id/books/edition/Sistem_Digital/vdzDEAAAQBAJ?hl=id&gbpv=1&dq=tahapan+kerja+memori+komp+uter&pg=PA163&printsec=frontcover (accessed 9.17.24).
- Pressman, R. S. (2020). *Software Engineering: A Practitioner's Approach*. 9th Edition. McGraw-Hill Education.
- Purnamawati, Akil, M. and Nuridayanti (2021) 'Perancangan Embedded System Pada Pembacaan dan Pengendalian Multi Sensor Berbasis Internet Of Things (IoT)', in *Seminar Nasional LP2M UNM*, pp. 752–764.
- Putra, Y.H., 2020. *Analisa Kinerja Sistem* [WWW Document]. URL https://www.google.co.id/books/edition/Analisa_Kinerja_Sistem/u-PgDwAAQBAJ?hl=id&gbpv=1&dq=kinerja+memori&pg=PA20&printsec=frontcover (accessed 9.17.24).
- Rafat, M. Z., Yanti, P., Adawiyah, R., Abror, J. M., & Maulana, A. R. (2024). ANALISIS STRUKTUR MEMORI: HIERARKI MEMORI PADA ORGANISASI ARSITEKTUR KOMPUTER. *Kohesi: Jurnal Multidisiplin Saintek*, 3(5), 100–111.
- Rafat, M. Z., Yanti, P., Adawiyah, R., Abror, J. M., & Maulana, A. R. (2024). ANALISIS STRUKTUR MEMORI: HIERARKI MEMORI PADA ORGANISASI ARSITEKTUR KOMPUTER. *Kohesi: Jurnal Multidisiplin Saintek*, 3(5), 100–111.
- Rafat, M.Z., Yanti, P., Adawiah, R., Abror, J.M., Maulana, A.R., 2024. ANALISIS STRUKTUR MEMORI : HIERARKI MEMORI PADA ORGANISASI ARSITEKTUR KOMPUTER. *Kohesi: Jurnal Sains dan Teknologi* 3, 11–21. <https://doi.org/10.3785/KOHESI.V3I6.3543>
- Ranti, S. (2023). *Pengertian dan Siklus Pemrosesan Data Pada Komputer*.

- <https://tekno.kompas.com/read/2023/02/19/03000017/pengetahuan-dan-siklus-pemrosesan-data-pada-komputer-?page=all>
Ranti, S. (2023). Pengertian dan Siklus Pemrosesan Data Pada Komputer.
- <https://tekno.kompas.com/read/2023/02/19/03000017/pengetahuan-dan-siklus-pemrosesan-data-pada-komputer-?page=all>
Ratna, S., 2023. Sistem Operasi [WWW Document]. URL mozilla-extension://07c6126a-a8c7-4699-ba38-5564008df4b8/enhanced-reader.html?openApp&pdf=https%3A%2F%2Fprints.uniska-bjm.ac.id%2F17188%2F1%2FFullBook%2520Sistem%2520Operasi.pdf (accessed 9.18.24).
- Reddy, P. K., & Kumar, K. (2020). "Performance Analysis of Cloud Computing Systems." *International Journal of Computer Applications*, 975, 20-26. doi:10.5120/21970-9760.
- Ridhalri, R. (2022). Pemanfaatan Caching System Menggunakan Redis Untuk Sistem Pengelolaan Informasi Ambalan Ashabul Kahfi Berbasis Web. *Jurnal Dialogika : Manajemen Dan Administrasi*, 4(1).
<https://doi.org/10.31949/dialogika.v4i1.3750>
- Ridhalri, R. (2022). Pemanfaatan Caching System Menggunakan Redis Untuk Sistem Pengelolaan Informasi Ambalan Ashabul Kahfi Berbasis Web. *Jurnal Dialogika : Manajemen Dan Administrasi*, 4(1).
<https://doi.org/10.31949/dialogika.v4i1.3750>
- Rita Puspita Sari (2024) Mengenal apa itu Quantum Computing? Manfaat dan Penggunaanya, www.cloudcomputing.id/Pengetahuan-Dasar/Mengenal-Quantum-Computing. Available at: <https://www.cloudcomputing.id/pengetahuan-dasar/mengenal-quantum-computing> (Accessed: 10 September 2024).
- Roesetiawan, A. (2018) 'Desain Kinerja Komputer Modern: Analisis dan Pengembangan', *Journal of Informatics and Computing*, 3(1), pp. 1–12.
- Rosidi, M. (2020) 'Desain Kinerja Komputer Modern', *Journal of Computer Science and Technology*, 4(1), pp. 1–10.
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson Education.
- RUSSELL, S.J. and NORVIG, P. *Artificial Intelligence A Modern Approach Third Edition*.

- Sailallah, H.R.P., 2023. Network Attached Storage (NAS): Pengertian dan Manfaatnya [WWW Document]. URL <https://it.telkomuniversity.ac.id/network-attached-storage/#> (accessed 9.17.24).
- Saino, L., Psaras, I., & Pavlou, G. (2016). Understanding sharded caching systems. *Proceedings - IEEE INFOCOM*, 2016-July. <https://doi.org/10.1109/INFOCOM.2016.7524442>
- Saino, L., Psaras, I., & Pavlou, G. (2016). Understanding sharded caching systems. *Proceedings - IEEE INFOCOM*, 2016-July. <https://doi.org/10.1109/INFOCOM.2016.7524442>
- Sanders, Donald H. *Computers today / Donald H. Sanders*. New York: Mc Graw-Hill, 1988.
- Santoso, E. and Informasi, S. (2023) 'Jaringan Komputer Berbasis Cloud: Transformasi Dalam Arsitektur It', *Teknologiterkini.org*, 3(10), pp. 2023–2024.
- Saputra, A., Weriza, J., Hadi, H. S., & Mallisza, D. (2018). E-Administrasi Kependudukan Pada Kelurahan Bungus Barat. *UNES Journal of Information System*, 3(1).
- Sarangi, R. Smruti. 2024. *Basic Computer Architecture*. Attribution-NoDerivatives 4.0 International.
- Sarangi, S.R. (2023) 'Basic Computer Architecture Version 2.2'.
- Sarangi, Smruti R (2024) 'Basic Computer Architecture Version 2.5', (July), pp 641-728.
- Sharma, A., & Sharma, S. (2019). "The Role of AI in Securing Cloud Computing." *Journal of Cyber Security Technology*, 3(4), 245-263. doi:10.1080/23742917.2019.1636617.
- SILBERSCHATS, A., GALVIN, P.B. and GAGNE, G. (2012) *Operating System Concepts Ninth Edition*.
- Silberschatz A. ,Galvine P.B., Gagne G. 2018. *Operating System Concept*. Amerika: Penerbit. Wiley .
- Singh, R., & Rani, S. (2018). "Exploring the Future of User Experience Design: Current Trends and Future Directions." *International Journal of Information Management*, 39, 61-70. doi:10.1016/j.ijinfomgt.2017.11.008.
- Smruti R. Sarangi. 2024. *Basic Computer Architecture Version*
- Soepomo, Prof. 2013. "No Title." 1: 181–90.
- Sommerville, I. (2019). *Software Engineering*. 10th Edition. Pearson.
- Sriani, Diktat : *Arsitektur dan Organisasi Komputer : 2020*
- Sriani. (2020). *Diktat Arsitektur Dan Organisasi Komputer*. Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera

- Utara Medan. [http://repository.uinsu.ac.id/9647/1/Diktat MK ARSIKOM Sriani.pdf](http://repository.uinsu.ac.id/9647/1/Diktat_MK_ARSIKOM_Sriani.pdf)
- Sriani. (2020). Diktat Arsitektur Dan Organisasi Komputer. Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan. [http://repository.uinsu.ac.id/9647/1/Diktat MK ARSIKOM Sriani.pdf](http://repository.uinsu.ac.id/9647/1/Diktat_MK_ARSIKOM_Sriani.pdf)
- Stallings, W. (2010). *Computer Organization and Architecture: Designing for Performance*. Prentice Hall.
- Stallings, W. (2012). *Computer Organization and Architecture: Designing for Performance*. Pearson.
- Stallings, William. 2016. *Computer Organization and Architecture Designing for Performance*. tenth edition. London: Pearson.
- STALLINGS, WILLIAM. and BROWN, LAWRIE. (2014) *Computer security : principles and practice*. Pearson Prentice Hill.
- Suh, M., et al. (2019). "Challenges and Future Directions in Cloud Computing." *Journal of Cloud Computing: Advances, Systems and Applications*, 8(1), 1-15. doi:10.1186/s13677-019-0147-7.
- Suparman, A., Setyawan, A.E., Ma'sum, H., Anyan, Hatta, H.R., 2023. *Pengenalan Dasar Komputer [WWW Document]*. URL https://www.google.co.id/books/edition/PENGENALAN_DASAR_KOMPUTER/xLTSEAAAQBAJ?hl=id&gbpv=1&dq=tahan+kerja+memori+komputer&pg=PA30&printsec=frontcover (accessed 9.17.24).
- Supriyanta. 2015. *Interaksi Manusia & Komputer*. Deepublish: Yogyakarta.
- Sweeney, L. (2002). "k-anonymity: A Model for Protecting Privacy." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5), 557-570.
- Tanenbaum, A. S. (2013). *Computer Organization and Architecture*. Pearson.
- Tanenbaum, A. S., & Austin, T. (2013). *Structured Computer Organization*. Prentice Hall.
- TANENBAUM, A.S. and BOS, H. (2015) *Modern Operating Systems Fourth Edition*.
- Tanenbaum, Andrew S., and Todd Austin. 2013. *Structured Computer Organization*. Sixth Edition. London: Pearson Education, Inc.
- Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2010). *Software Architecture: Foundations, Theory, and Practice*. Wiley.
- Trivusi, 2022. *Pengertian dan Jenis-Jenis Memori Komputer - Trivusi [WWW Document]*. URL

- <https://www.trivusi.web.id/2022/08/memori-komputer.html>
(accessed 8.31.24).
- V. C. Hamacher, Z.G.Vranesic, dan S.G.Zaky. 2001. *Computer Organization*, McGraw-Hill
- Vanderwiel, S. P., & Lilja, D. J. (2000). Data Prefetch Mechanisms. *ACM Computing Surveys*, 32(2), 174–199. <https://doi.org/10.1145/358923.358939>
- Vanderwiel, S. P., & Lilja, D. J. (2000). Data Prefetch Mechanisms. *ACM Computing Surveys*, 32(2), 174–199. <https://doi.org/10.1145/358923.358939>
- Vermesan, O., & Friess, P. (2014). *Internet of Things: From Research and Innovation to Market Deployment*. River Publishers.
- Victor, A., & Qurotul, A. (2013). Arsitektur Komputer Teori dan Perkembangannya. In *Arsitektur Komputer teori dan perkembangannya* (Issue 2).
- Victor, A., & Qurotul, A. (2013). Arsitektur Komputer Teori dan Perkembangannya. In *Arsitektur Komputer teori dan perkembangannya* (Issue 2).
- Voas, J., & Miller, K. (2021). *Software Assessment: Reliability, Safety, Testability*. Addison-Wesley.
- Vogelsang, A., & Eichelberger, H. (2020). *Software Engineering for Modern Web Applications: Methodologies and Technologies*. IGI Global.
- Wahyono, T. 2007. *Building & Maintenance PC Server*. PT Elex Komputindo: Jakarta.
- Wang, Y., & Zheng, L. (2020). "Energy-Efficient Computing: Opportunities and Challenges." *ACM Computing Surveys*, 52(6), 1-34. doi:10.1145/3425481.
- Weinberg, N. (2019) A Brief History of Computers, Computers in the Information Society. Available at: <https://doi.org/10.4324/9780429033124-3>.
- Widharma, I. G. S. (2020). *Arsitektur Reduction Instruction Set Computer Politeknik Negeri Bali* (Issue 01).
- Widharma, I. G. S. (2020). *Arsitektur Reduction Instruction Set Computer Politeknik Negeri Bali* (Issue 01).
- Wilkinson, B. dan Allen, M. 2010. *Parallel Programming - Teknik dan Aplikasi Menggunakan Jaringan Workstation & Komputer Paralel*. Penerbit ANDI: Yogyakarta.
- William Stallings, “ *Computer Organization and Architecture Designing for Performance Eighth Edition*”, Prentice Hall, 2019

- Winarno, W.W. 2021. Sistem Informasi dan Teknologi Informasi: Sebuah Pengantar. Wingit Press: Jakarta.
- Wirk G.S. (1991) Digital Computer Systems, MacMillan, London.
- Zaki, A. 2010. Panduan Hardware Komputer. Penerbit ANDI: Yogyakarta.
- Zames G. (1966) On the input-output stability of time-varying nonlinear feedback systems, IEEE-TAC, vol. AC-11, April, pp. 228-238, July pp. 445-476.
- Zhao, W., et al. (2018). "Distributed Computing: Advances and Challenges." *Future Generation Computer Systems*, 79, 235-242. doi:10.1016/j.future.2017.09.016.
- Zhou, J., et al. (2020). "A Review of Human-Computer Interaction in Smart Environments." *Computers in Human Behavior*, 105, 106189. doi:10.1016/j.chb.2019.106189.
- Zulfikar, A. (2022). Penggunaan Docker Dan Kubernetes Pipeline Dalam Pengembangan Aplikasi Prediksi Cacat Perangkat Lunak Melalui Pendekatan Mlops.
- Zulfikar, A. (2022). Penggunaan Docker Dan Kubernetes Pipeline Dalam Pengembangan Aplikasi Prediksi Cacat Perangkat Lunak Melalui Pendekatan Mlops.

PROFIL PENULIS



ALEX COPERNIKUS ANDARIA, S.T., M.PD.

Penulis bernama lengkap Alex Copernikus Andaria, dengan nama panggilan “Andi”, lahir di Kota Manado pada 18 April 1984. Merupakan anak kedua dari tiga bersaudara, penulis adalah putra dari Dr. Calvin S. Andaria, M.Si, dan Dra. Margaretha A. Bintang. Penulis menikah dengan Thalia P. Wijaya, S.Kom, dan dikaruniai dua orang anak: Charissa V. Andaria sebagai anak pertama dan Guido C. Andaria sebagai anak kedua.

Pendidikan dasar penulis diselesaikan di SDN Malalayang, Sulawesi Utara, pada tahun 1990-1996. Selanjutnya, penulis melanjutkan pendidikan di SMPN 2 Condong Catur, Yogyakarta, dari tahun 1997 hingga 2000, dan kemudian menuntaskan pendidikan di SMUN 2 Sulawesi Utara pada tahun 2000-2003. Penulis menyelesaikan pendidikan Sarjana (S1) di Jurusan Teknik Elektro dan Informatika, Universitas Sam Ratulangi, Sulawesi Utara, dari tahun 2003 hingga 2010, dan melanjutkan pendidikan Magister (S2) pada Jurusan Pendidikan Teknologi dan Kejuruan (cabang Ilmu Komputer) di Universitas Negeri Manado, Sulawesi Utara, dari tahun 2019 hingga 2022. Selain aktif dalam kegiatan penelitian dan pengembangan di bidang sistem komputer dan teknologi pembelajaran, penulis juga menjabat sebagai Dekan Fakultas Ilmu Pengetahuan Teknologi dan Keguruan pada periode 2024-2029. Dengan semangat dan dedikasi yang tinggi, penulis berusaha untuk membagikan ilmu dan pengalamannya kepada semua orang lewat buku ini.

Website; WoS; Orchid; Researchgate; G-scholar; SintalD; E-mail



BAHTERA ALAM WIJAKSONO M.KOM.

Seorang Dosen tetap Prodi Teknik Informatika pada Universitas indraprasta PGRI. Lahir di Jakarta, 22 Oktober 1984. Penulis merupakan anak Kedua dari tiga bersaudara dari pasangan Bapak H. Slamet Widodo dan Ibu Hj. Dewi Purnama.

Pendidikan telah ditempuh program Sarjana (S1) Universitas Budi Luhur Prodi Sistem Informasi dan Program Pasca Sarjana (S2) di Universitas Budi Luhur prodi Teknik Informatika. Penulis memiliki dua buah hati Rui Shanin Anindita dan Raihan Rizalul Haq dari pasangan Rosi Selly Sandiah (bunda kece). Berikut judul buku yang telah ditulis dan terbitkan: Sistem Robotika berbasis Mikrokontroller

MISWADI, S.KOM., M.KOM.



Miswadi “Mas Raden” Lahir di Kota Gombong - Kebumen pada bulan Mei 1975. Tamat dari sekolah STM Wongsorejo Gombong – Kebumen Jurusan Listrik Instalasi tahun 1994 dan menyelesaikan program Studi Diploma III (D3) Teknik Komputer Di STMIK Bani Saleh Bekasi tahun 2000. Dan tahun 2006 melanjutkan Program Studi S1 Jurusan Teknik Informatika (S.Kom) selesai Tahun 2008 di STMIK Bani Saleh Bekasi. Pada tahun 2012 melanjutkan studi Magister Ilmu Komputer (S2) di Universitas Budi Luhur Jakarta dan selesai pada tahun 2015.

Tahun 1994 ~ 2000 bekerja di perusahaan Indomobil Group sebagai Quality Assurance dan terakhir sebagai staf personalia. Saat ini sedang bekerja pada salah satu perusahaan Manufacture di Cikarang, Bekasi tepatnya pada PT Kiyokuni Indonesia sebagai IT Administrator yang bertanggung jawab menangani Sistem Jaringan, Hardware, Software, Aplikasi Database dan Server (Email, Proxy, VPN, Firewall & Proxy). Tahun 2011 mengaplikasikan pengalaman di dunia industri dengan menjadi tenaga pengajar SMK jurusan TKJ dan Dosen di STMIK di Bekasi.

Konsentrasi Inti pada bidang jaringan komputer, infrastruktur dan Network/Database, serta menagani Server dengan Platform Microsoft dan Unix (GNU – Linux). Juga aktif dalam, komunitas IT, organisasi keagamaan dan kegiatan sosial lainnya. Mempunyai Hobby “Jalan-jalan”, membaca, jogging. Senang membantu dan berdiskusi tentang Agama Islam, bidang komputer dan perkembangan teknologi informasi. Informasi lebih lanjut mengenai penulis dapat dilihat pada: E-Mail: miswadi@kiyokuni.co.id, miswadi@gmail.com



**EKA PRASETYA ADHY SUGARA, S. T.,
M. KOM.**

Penulis lahir di Palembang pada tanggal 24 April 1982. Penulis menamatkan pendidikan SD, SMP dan SMA di Palembang. Setelah lulus dari SMA Negeri 5 Palembang, penulis melanjutkan kuliah di Program Studi Teknik Mesin Universitas Gadjah Mada pada tahun 2000. Penulis mulai mengajar pada tahun 2009 sebagai dosen luar biasa dan bergabung menjadi dosen tetap di Program Studi Desain Komunikasi Visual Politeknik Palcomtech (yang kemudian bergabung dengan STMIK menjadi Institut Teknologi dan Bisnis Palcomtech pada 2022). Penulis berkesempatan untuk melanjutkan studi program magister di Program Studi Teknik Informatika Universitas Bina Darma Palembang pada 2013. Pada tahun 2024, penulis dialih tempatkan di Program Studi Sistem Informasi dengan bidang keilmuan di Desain Grafis, Aplikasi Multimedia, Augmented Reality dan Virtual Reality serta Pengembangan Game.

Saat ini, penulis memiliki ketertarikan di bidang Kecerdasan Buatan, Machine Learning, Deep Learning dan Generative AI. Beberapa buku yang pernah ditulis diantaranya berjudul “Penganggaran untuk Taman Pendidikan Al-Qur’an (TPA)” yang merupakan luaran hibah PKM pada tahun 2019, “Cara Praktis Membangun Aplikasi Mobile” pada tahun 2021, dan book chapter “Data Science” pada tahun 2022.



SUWANDONO, S.KOM., M.KOM

Penulis lahir di Cilacap, Jawa Tengah bulan Juni 1976. Penulis menamatkan pendidikan dasar dan menengah di Cilacap, setelah lulus dari SMA Negeri 1 Maos (sekarang SMU Neger 1 Maos) melanjutkan kuliah S1 di STMIK Dian Nuswantoro Semarang (sekarang Universitas Dian Nuswantoro) Jurusan Teknik Informatika, kemudian melanjutkan S2 di STMIK Eresha Jurusan Teknik Informatika lulus di tahun 2016. Memulai karir sebagai Dosen sejak tahun 2001 di solo-Jawa Tengah, kemudian sejak tahun 2004 sampai dengan sekarang memulai karir sebagai dosen di Sekolah Tinggi Perpajakan Indonesia(STPI) di Jakarta, adapun penulis juga menjabat sebagai Kepala Pusat Komputer (Kapuskom) di STPI, saat ini mengampu mata kuliah Pengantar Komputer, Aplikasi Komputer II dan Metodologi Penelitian di STPI. Karir di Programmer menjadi programmer freelance saja. Penulis juga mempunyai hoby tentunya olah raga, hoby penulis berlatih Pencak Silat Merpati Putih sejak bersekolah di SMA N Maos.



SALMAN FARIZY. S.KOM, M.KOM, MCSE, MVP,

Menyelesaikan pendidikan dasar dan menengah di SDN 01 Cipinang Besar, SMPN 25 Cipinang Muara Jakarta Timur dan SMAN 91 Pondok Kelapa Jakarta Timur, sedangkan untuk perguruan tinggi Strata satu (S1) di Universitas Gunadarma dan Pasca Sarjana di STMIK Eresha yang saat ini sudah merger atau bergabung dengan Universitas Pamulang.

Penulis sempat bekerja di beberapa perusahaan asing dan juga lokal seperti Mattel Indonesia, Frigorex Indonesia, Kaltimex Energy, Microsoft Gold Partner, IT Consultant (partime) dan akhirnya memutuskan untuk menjadi Dosen tetap Universitas Pamulang. Mudah-mudahan dengan adanya buku ini akan menambah pengetahuan dan dapat bermanfaat terutama buat adik-adik mahasiswa yang hobi mengexplore dan mencoba hal-hal yang baru.



S. NURMUSLIMAH, S.T., M.T.

Penulis lahir di Bima tanggal 01 Juli 1971. Penulis adalah Dosen Tetap Pada Program Studi Teknik Informatika Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya.

Menyelesaikan Pendidikan S1 pada Jurusan Teknik Elektro dan melanjutkan S2 Pada Jurusan Sistem Jaringan Cerdas Multimedia ITS. Penulis Menekuni Bidang AI dan Jaringan Cerdas Multimedia.



PRAJOKO, M,KOM

Penulis lahir di Kab.Semarang, Jawa Tengah bulan Januari 1971. Penulis menamatkan pendidikan dasar dan menengah di Kab.Semarang, setelah lulus dari SMA Negeri Ambarawa Penulis melanjutkan Pendidikan Profesi Programmer di IMKA semarang. Kemudian Melanjutkan S1 Pendidikan di Jakarta,dan Tamat S2 dari STMIK ERESHA Jakarta,yang sekarang menjadi bagian dari Universitas Pamulang Jakarta.

Penulis memiliki beberapa pengalaman kerja sebagai programmer di berbagai perusahaan Nasional seperti Astra, International, Hyundai Mobil dan lain-lain.

Saat ini penulis berprofesi sebagai dosen di Universitas Muhamadiyah Sukabumi, mengajar untuk mata Kuliah Pemograman Berorientasi Objek, Moobile programming dan Manajemen Proyek Perangkat Lunak.



MUHAMMAD SYARIF HARTAWAN

Penulis lahir di Jakarta, bulan Juli 1972. Penulis menamatkan pendidikan dasar dan menengah di Jakarta, setelah lulus dari SMA Muhammadiyah 5 Tebet Jakarta Selatan melanjutkan kuliah S1 di STMIK Swadharma Jurusan Informatika, kemudian melanjutkan S2 di STMIK Eresha dan S2 di STIE Indonesia School Of Management Jurusan Manajemen informasi. Serta melanjutkan pendidikan Program Doktor di Kampus Asia e University Malaysia Jurusan Information and Communication Technology.

Sebagai praktisi pengajar penulis memiliki beberapa sertifikat internasional seperti IT Security, CISCO dan Grafis Desain. Sertifikasi BNSP juga penulis miliki, seperti IT Auditor, Desain grafis, Network, Programmer, Multimedia, dll.



PRAYOGO, S.KOM., M.KOM.

Penulis lahir di Purworejo, Jawa Tengah bulan Juli 1972. Penulis menamatkan pendidikan dasar dan menengah di Purworejo , setelah lulus dari SMK , hijrah ke Tangerang bekerja sebagai Riset Development dan kemudian sambil melanjutkan kuliah S1 di STMIK Masa Depan Cimone Tangerang Jurusan Teknik Informatika, kemudian melanjutkan S2 di STMIK Eresha Jakarta, yang kemudian merger dengan Universitas Pamulang Jakarta Program studi Magister Teknik Informatika.

Memulai mengajar Tahun 2013 menjadi pengajar Tidak tetap di Amik MAPAN dan STMIK Masa Depan yang hingga saat ini menjadi Dosen Tetap di AMIK MAPAN Kota Tangerang . Penulis juga mempunyai pengalaman mengajar di beberapa kampus yaitu, Institute Global Tangerang, UTPAS di kota Tangerang, Mengajar di Kampus LP3i Cimone Tangerang, selain itu juga mengajar di STBA Teknocrat Cikupa Kabupaten Tangerang. Selain itu Penulis pernah mengajar di Program Prakerja Pemerintah di LPK Geti Serpong sebagai LPK yang ditunjuk Pemerintah. Penulis juga menulis di blog

: <https://e-prayogo.blogspot.com/> . Dan juga mempunyai channel youtube : <https://youtu.be/E7FOhPkbFqw?si=Jaic5gGwO5ZnTmd>
Dan Instagram:

https://www.instagram.com/surya_prayogo.official?igsh=aW40MGY4MWRobDk3



MUHAMMAD KHAERUDIN, M.KOM.

Lahir di Kemayoran Jakarta Pusat pada tanggal 13 Juni 1964, saat ini penulis sebagai Dosen Tetap Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya Riwayat pendidikan formal penulis, dimulai dari Sekolah Dasar Negeri (SDN) 01 Kebon Kosong, Kemayoran lulus tahun 1979, kemudian melanjutkan ke Sekolah Menengah Pertama Negeri (SMPN) 10 Sumur Batu Kemayoran hanya sampai kelas 1, kemudian pindah ke SMPN Pondok Gede Bakasi lulus tahun 1982, Sekolah Menengah Farmasi Ditkesad di Jakarta Pusat lulus tahun 1985, kuliah S1 jurusan Teknik Informatika di STMIK Budi Luhur yang sekarang menjadi Universitas Budi Luhur lulus tahun 1993, Kuliah S2 jurusan Teknik Informatika lulus tahun 2012.



IMAM ZAENUDDIN, S.KOM., M.KOM.

Penulis lahir di Magelang, Jawa Tengah pada tanggal 4 April 1983. Penulis menyelesaikan pendidikan S1 Teknik Informatika di STMIK Pranata Indonesia pada tahun 2006. Kemudian melanjutkan pendidikan S2 Teknik Informatika di STMIK Eresha dan lulus pada tahun 2015 dengan bidang peminatan Rekayasa Bisnis.

Memulai karir sebagai tenaga ahli bidang Informatika serta dosen di STMIK Pranata Indonesia sejak tahun 2009. Saat ini penulis menjabat sebagai Asisten Ahli dengan bidang penugasan Bahasa Pemrograman. Penulis aktif mengajar

mata kuliah seperti Bahasa Pemrograman, Keamanan Jaringan Komputer, dan Algoritma dan Pemrograman.

Selain mengajar, penulis juga aktif melakukan penelitian dan pengabdian masyarakat dan beberapa karya ilmiah yang telah dipublikasikan. Penulis memiliki sertifikasi profesional dari Microsoft yaitu MCSA, MCTS dan Mikrotik yaitu MCTNA, MCTRE. Saat ini penulis terdaftar sebagai anggota Asosiasi Dosen Indonesia (ADI) dan Asosiasi Pendidikan Tinggi Informatika dan Komputer (APTIKOM).



DENI KUSWOYO, S.KOM., M.KOM

Penulis lahir di Palembang, bulan Maret 1988. Penulis menamatkan pendidikan dasar dan menengah di Jakarta, setelah lulus dari SMA Negeri 57 Jakarta melanjutkan kuliah S1 dan S2 di Universitas Budi Luhur Jakarta Program Studi Magister Ilmu Komputer. Memulai karir sebagai IT sejak tahun 2010 sebagai Network Engineer. Saat ini sebagai dosen di Politeknik Negeri Media Kreatif, mengajar pada Program Studi Teknologi Rekayasa Multimedia.



HARI WAHYUDI, M.TH

Penulis lahir di Batu, Malang, Jawa Timur pada tanggal 19 Pebruari 1973. Penulis menyelesaikan pendidikan dasar dan menengah di Kota Batu dan selanjutnya menyelesaikan program Studi S1 di bidang teologi di STT ATI Anjungan – Mempawah – Kalimantan Barat selanjutnya menyelesaikan program S2 di bidang teologi di STT Institut Injil Indonesia Batu – Jawa Timur.

Saat ini penulis mengajar teologi di Bidang Perjanjian Baru di Sekolah Tinggi Teologi ATI Anjungan Pontianak. Selain sebagai seorang dosen, penulis juga seorang tenaga Auditor Mutu Internal di Sekolah Tinggi Teologi ATI Anjungan Pontianak, dan sebagai Editor in Chief di JTI Jurnal Teologi Injili.

BUDI BERLINTON SITORUS S.T, M.SC



Penulis lahir di Jakarta, DKI Jakarta tahun 1975. Penulis menamatkan pendidikan dasar dan menengah di Jakarta. Setelah lulus dari SMA St. Antonius melanjutkan kuliah S1 di STT Telkom Bandung Jurusan Informatika, kemudian melanjutkan S2 di Greenwich University, Inggris, Jurusan Distributed Computer Systems. Pada tahun 2001 lulus S2 dan memulai karir sebagai dosen paruh waktu di Universitas Bina Nusantara tahun 2002. Saat ini sebagai dosen paruh waktu di Universitas Multimedia Nusantara.



NURHADI, S.KOM., M.KOM

Penulis lahir di Bekasi, Jawa Barat bulan Nopember 1978. Penulis menamatkan pendidikan dasar dan menengah di Bekasi, setelah lulus dari STM Negeri 1 Bekasi (sekarang SMK Neger 1 Cikarang Barat) melanjutkan kuliah D3 di STMIK Pranata Indonesia Jurusan Managemen Informatika. kemudian melanjutkan S1 di STMIK Pranata Indonesia Jurusan Sistem informasi. Pada tahun 2015 lulus S2 di universitas Budi Luhur Jakarta Program studi Magister Ilmu Komputer. Memulai karir sebagai IT sejak tahun 2002 sebagai Technical Support, Saat ini penulis aktif sebagai SAP Abap Spesialist.

Saat ini sebagai dosen di STMIK Pranata Indonesia, mengajar untuk mata Kuliah Bahasa Pemrograman 1 dan Bahasa Pemrograman 2. Penulis juga aktif membuat video pembelajaran, salah satunya tentang pemrograman visual basic dan SQL server yang dapat di lihat di channel youtube penulis di <https://bit.ly/PDMVBSQL>.



TUTI HANDAYANI, S.KOM., M.KOM.

Seorang dosen Prodi Teknik Informatika pada Universitas Indraprasta PGRI. Lahir di Jakarta tahun 1980. Penulis pernah bekerja menjadi karyawan swasta 2004-2016 di Tangerang Selatan. Putri ketiga dari empat bersaudara. Pernah mengajar di Universitas Muhammadiyah Tangerang fakultas Teknik serta fakultas Ekonomi dan Bisnis. Pernah mengajar di STT PLN. Pendidikan telah ditempuh program Sarjana (S1) Universitas Budi Luhur Prodi Sistem Informasi dan Program Pasca Sarjana (S2) di Universitas Budi Luhur prodi Sistem Informasi. Memiliki suami bernama Hardi dan seorang putri bernama Fiqa. Hobi kuliner, memasak dan travelling.