

$$E_1(x, y) = \frac{1}{n} \int_{-a}^n f \left(\frac{1+dx}{(z-x)^2} \right)$$

$$f(x) = \frac{1}{\pi} \sum_{i=1}^{\infty} \frac{\sqrt{(x-u)e^{+i}}}{2\pi}$$

$$= \sum_{i=0}^{\infty} \exp(Xp^{(k)}) - \frac{b}{(\alpha -$$

$$V^2 = \frac{1}{2} \frac{m \delta l r}{v}$$

$$e_3 \ell^2 = \frac{1}{4} a e_2 - b^2 + a \omega t_2$$

$$\int_{i=0}^{\infty} \frac{\partial (zv - dx)}{x^2} = \dots \int_{i=0}^{-\infty} \frac{n(X(n+e)) d}{(a(l+b)-e)^2}$$

KOMPUTASI KOMPREHENSIF

Teori Formal Hingga Akselerasi Digital Abad 21

Tim Penulis:

Nur Azizah | Dedy Iskandar | Wiranti Sri Utami

Novan Ardianto | Fandi Ali Mustika

Budi Chehabudin | Martono

KOMPUTASI KOMPREHENSIF

Teori Formal Hingga Akselerasi Digital Abad 21

**Nur Azizah
Dedy Iskandar
Wiranti Sri Utami
Novan Ardianto
Fandi Ali Mustika
Budi Chehabudin
Martono**



KOMPUTASI KOMPREHENSIF

Teori Formal Hingga Akselerasi Digital Abad 21

Penulis:

Nur Azizah
Dedy Iskandar
Wiranti Sri Utami
Novan Ardianto
Fandi Ali Mustika
Budi Chehabudin
Martono

Tata Letak : Lilis Khalistul Karimah
Desain Cover : Asep Nugraha
Ukuran : UNESCO 15,5 x 23 cm
Halaman : v, 110
ISBN : 978-634-7522-91-7
Terbit Pada : Agustus 2026
Anggota IKAPI : No. 073/BANTEN/2023

Hak Cipta 2026 @ Sada Kurnia Pustaka dan Penulis

Hak cipta dilindungi undang-undang dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa izin tertulis dari penerbit dan penulis.

PENERBIT PT SADA KURNIA PUSTAKA

Jl. Kramat, Panenjoan Kec. Carenang, Kab. Serang – Banten, 42195
Email : sadapenerbit@gmail.com
Website : sadapenerbit.com & repository.sadapenerbit.com
Telpon/WA : +62 838 1281 8431

KATA PENGANTAR

Perkembangan ilmu komputer dan teknologi informasi berlangsung sangat cepat, didorong oleh kemajuan komputasi, kecerdasan buatan (*Artificial Intelligence*), komputasi awan (*Cloud Computing*), *Internet of Things* (IoT), komputasi berkinerja tinggi (*High-Performance Computing*), analisis data besar (*Big Data Analytics*), hingga komputasi kuantum (*Quantum Computing*). Berbagai inovasi tersebut telah mengubah cara manusia bekerja, belajar, berkomunikasi, berbisnis, serta mengambil keputusan berbasis data. Oleh karena itu, pemahaman terhadap konsep-konsep fundamental komputasi sekaligus perkembangan teknologi mutakhir menjadi kebutuhan yang sangat penting bagi mahasiswa, dosen, peneliti, praktisi, maupun masyarakat luas.

Buku “Komputasi Komprehensif: Teori Formal Hingga Akselesrasi Digital Abad 21” bertujuan untuk memberikan wawasan yang komprehensif mengenai konsep, strategi, dan implementasi *CRM* dalam dunia bisnis modern.

Harapan kami, buku ini dapat menjadi referensi yang bermanfaat dalam kegiatan pembelajaran, penelitian, maupun pengembangan inovasi di bidang ilmu komputer dan teknologi informasi. Selain itu, buku ini diharapkan mampu mendorong pembaca untuk berpikir kritis, analitis, dan inovatif dalam menghadapi berbagai tantangan komputasi modern yang semakin kompleks serta mampu memanfaatkan teknologi secara efektif, etis, dan bertanggung jawab.

Akhir kata, penulis menyampaikan terima kasih kepada semua pihak yang telah memberikan dukungan, bantuan, dan inspirasi selama proses penyusunan buku ini. Semoga segala kontribusi yang diberikan memperoleh balasan yang terbaik.

Tim Penulis

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	iv
BAB 1 EVOLUSI PARADIGMA DAN TAKSONOMI ILMU KOMPUTER	1
A. Pendahuluan	1
B. Sejarah Perkembangan Ilmu Komputer	4
C. Evolusi Paradigma Ilmu Komputer	7
D. Taksonomi Ilmu Komputer	11
BAB 2 ANALISIS ALGORITMA DAN STRUKTUR DATA.....	16
A. Pendahuluan	16
B. Tujuan dan Pentingnya Analisis Algoritma.....	17
C. Jenis-Jenis Analisis Algoritma	18
D. Efisiensi Algoritma.....	18
E. Pengertian Teknik Analisis Kompleksitas	20
F. Pengertian dan Konsep Dasar Rekursi.....	21
G. Pengertian dan Karakteristik Array	23
H. Teknik Perancangan Algoritma	24
BAB 3 TINGKAT LANJUT ARSITEKTUR SISTEM TERDISTRIBUSI DAN KOMPUTASI PARALEL	27
A. Pendahuluan	27
B. Arsitektur Sistem Terdistribusi Modern	29
C. <i>Client-Server Architecture</i>	30
D. <i>Three-Tier Architecture</i>	37
E. <i>N-Tier Architecture</i>	40
BAB 4 ARSITEKTUR JARINGAN KOMPUTER DAN KOMUNIKASI DATA LAPIS JAMAK.....	45
A. Pendahuluan	45
B. Konsep Dasar Komunikasi Data	47
C. Arsitektur Jaringan Komputer.....	54
D. Model OSI (<i>Open System Interconnection</i>)	57
BAB 5 KRIPTOGRAFI MODERN DAN KETAHANAN SIBER (<i>CYBER RESILIENCE</i>).....	60
A. Pendahuluan	60
B. Konsep Dasar Kriptografi Modern	63
C. Algoritma Kriptografi Modern	67
D. <i>Cyber Resilience</i> sebagai Evolusi <i>Cyber Security</i>	73

BAB 6 REKAYASA PERANGKAT LUNAK MODERN DAN MANAJEMEN SIKLUS HIDUP (SDLC)	78
A. Konsep Dasar Rekayasa Perangkat Lunak Modern.....	79
B. Siklus Hidup Pengembangan Perangkat Lunak (SDLC)	80
C. Tahapan Manajemen Siklus Hidup Perangkat Lunak Secara Rinci	82
D. Metodologi Pengembangan Perangkat Lunak Modern	85
E. Tantangan dan Tren Masa Depan Rekayasa Perangkat Lunak..	87
BAB 7 CLOUD COMPUTING DAN INFRASTRUKTUR EDGE TO CLOUD	89
A. Pendahuluan	89
B. Perbedaan Edge dan <i>Cloud Computing</i>	91
C. Infrastruktur <i>Edge</i>	94
D. Penutup	99
DAFTAR PUSTAKA.....	100
BIOGRAFI PENULIS.....	105

BAB 1

EVOLUSI PARADIGMA DAN TAKSONOMI ILMU KOMPUTER

Dr. Nur Azizah, S.Kom., M.Akt., M.Kom.
Universitas Raharja

A. Pendahuluan

Ilmu komputer merupakan salah satu disiplin ilmu yang mengalami perkembangan paling pesat dalam beberapa dekade terakhir. Perkembangannya tidak hanya dipengaruhi oleh kemajuan teknologi perangkat keras (*hardware*) dan perangkat lunak (*software*), tetapi juga oleh perubahan paradigma berpikir dalam merancang, mengembangkan, dan memanfaatkan sistem komputasi. Pada awal kemunculannya, ilmu komputer lebih berfokus pada proses perhitungan matematis dan pengolahan data secara otomatis menggunakan mesin komputasi. Namun, seiring meningkatnya kompleksitas kebutuhan masyarakat dan industri, ruang lingkup ilmu komputer berkembang menjadi disiplin multidisipliner yang mencakup kecerdasan buatan, komputasi awan, keamanan siber, ilmu data, *Internet of Things* (IoT), robotika, hingga komputasi kuantum.

Menurut Denning (2017), ilmu komputer tidak hanya mempelajari komputer sebagai perangkat, tetapi juga mempelajari proses komputasi (*computation*), representasi informasi, algoritma, serta prinsip-prinsip yang mendasari pengolahan informasi. Dengan demikian, ilmu komputer merupakan disiplin yang terus berevolusi mengikuti perkembangan teknologi dan kebutuhan manusia. Evolusi tersebut melahirkan berbagai

paradigma baru yang mengubah cara para ilmuwan, peneliti, dan praktisi memandang proses komputasi. Paradigma dalam ilmu komputer dapat dipahami sebagai seperangkat konsep, pendekatan, asumsi, serta metode yang menjadi dasar dalam menyelesaikan suatu permasalahan komputasi.

Pergeseran paradigma terjadi ketika pendekatan lama tidak lagi mampu menjawab tantangan baru sehingga diperlukan pendekatan yang lebih efektif, efisien, dan adaptif. Kuhn (2012) menjelaskan bahwa paradigma merupakan kerangka berpikir yang menjadi acuan komunitas ilmiah dalam memecahkan masalah pada suatu periode tertentu. Konsep ini relevan dalam ilmu komputer karena perkembangan teknologi selalu diikuti oleh perubahan paradigma, mulai dari komputasi terpusat (*centralized computing*), komputasi terdistribusi (*distributed computing*), komputasi awan (*cloud computing*), hingga komputasi cerdas berbasis kecerdasan buatan (*artificial intelligence*).

Perubahan paradigma tersebut juga tercermin dalam perkembangan paradigma pemrograman. Pemrograman yang awalnya didominasi oleh pendekatan prosedural kemudian berkembang menjadi pemrograman berorientasi objek (*object-oriented programming*), pemrograman fungsional (*functional programming*), pemrograman deklaratif (*declarative programming*), serta paradigma pemrograman reaktif (*reactive programming*). Setiap paradigma menawarkan pendekatan yang berbeda dalam membangun perangkat lunak agar lebih mudah dikembangkan, dipelihara, dan disesuaikan dengan kebutuhan sistem yang semakin kompleks.

Selain perubahan paradigma, ilmu komputer juga berkembang melalui perluasan bidang kajian yang melahirkan berbagai cabang ilmu. Oleh karena itu, diperlukan suatu klasifikasi atau taksonomi ilmu komputer agar hubungan antardisiplin dapat dipahami secara sistematis. Taksonomi ilmu komputer mengelompokkan bidang-bidang seperti algoritma dan struktur data, rekayasa perangkat lunak, sistem operasi, jaringan komputer, basis data, kecerdasan buatan, keamanan siber, ilmu data, grafika komputer, interaksi manusia dan komputer, serta berbagai bidang lainnya. Pengelompokan ini membantu akademisi, peneliti, maupun praktisi dalam memahami ruang lingkup, batasan, serta keterkaitan antarbidang.

Menurut *Computing Curricula 2020* yang diterbitkan oleh ACM dan IEEE *Computer Society* (2020), disiplin komputasi saat ini meliputi *Computer Science, Computer Engineering, Software Engineering, Information Systems, Information Technology, dan Cybersecurity* sebagai bidang utama yang saling melengkapi. Masing-masing disiplin memiliki karakteristik, kompetensi inti, serta fokus pengembangan yang berbeda, namun seluruhnya berkontribusi terhadap perkembangan teknologi *digital* secara menyeluruh.

Perkembangan teknologi digital pada era Revolusi Industri 4.0 dan *Society 5.0* semakin mempercepat evolusi ilmu komputer. Integrasi antara kecerdasan buatan, komputasi awan, analitik data besar (*big data analytics*), *blockchain, internet of things, edge computing*, serta komputasi kuantum telah mengubah berbagai sektor, termasuk pendidikan, kesehatan, industri manufaktur, pemerintahan, keuangan, hingga transportasi. Kondisi tersebut menuntut lulusan dan profesional di bidang komputer untuk memiliki pemahaman yang komprehensif mengenai evolusi paradigma dan struktur keilmuan agar mampu beradaptasi terhadap perubahan teknologi yang berlangsung sangat cepat.

Peter J. Denning (2017) menyatakan bahwa *computer science is the study of computation and information processes, whether they are implemented in machines, nature, or society*. Pernyataan ini menegaskan bahwa fokus utama ilmu komputer adalah komputasi dan proses informasi, bukan sekadar perangkat komputer. Thomas S. Kuhn (2012) menjelaskan bahwa paradigma merupakan kerangka konseptual yang dianut oleh komunitas ilmiah dan akan berubah ketika muncul pendekatan baru yang lebih mampu menjelaskan fenomena atau menyelesaikan masalah. ACM & IEEE *Computer Society* (2020) menegaskan bahwa disiplin komputasi modern terdiri atas beberapa bidang utama yang saling terintegrasi, yaitu *Computer Science, Computer Engineering, Software Engineering, Information Systems, Information Technology, dan Cybersecurity*.

Pemahaman mengenai evolusi paradigma dan taksonomi ilmu komputer memiliki nilai strategis dalam pendidikan tinggi. Mahasiswa tidak hanya dituntut menguasai keterampilan teknis, tetapi juga memahami landasan konseptual yang melatarbelakangi perkembangan disiplin ilmu komputer. Dengan memahami sejarah perkembangan,

BAB 2

ANALISIS ALGORITMA DAN STRUKTUR DATA

Dedy Iskandar, S.Kom., M.T.I.
Universitas Raharja Tangerang

A. Pendahuluan

Analisis algoritma adalah proses mengevaluasi kinerja suatu algoritma berdasarkan efisiensi waktu (*time complexity*) dan penggunaan memori (*space complexity*). Tujuan analisis algoritma adalah menentukan seberapa efektif suatu algoritma dalam menyelesaikan masalah ketika ukuran data masukan (*input*) bertambah besar. Analisis ini membantu pengembang perangkat lunak memilih algoritma yang paling efisien sehingga aplikasi dapat berjalan lebih cepat dan menggunakan sumber daya secara optimal.

Menurut Thomas H. Cormen dkk. (2022), "*Algorithm analysis is concerned primarily with the amount of computational resources that an algorithm requires.*" Artinya, analisis algoritma berfokus pada jumlah sumber daya komputasi yang dibutuhkan suatu algoritma, terutama waktu dan memori. Sementara itu, Donald E. Knuth (1997) menyatakan bahwa analisis algoritma merupakan ilmu yang mempelajari efisiensi prosedur komputasi sehingga dapat dibandingkan satu sama lain berdasarkan kebutuhan sumber daya yang digunakan. Struktur data adalah cara mengorganisasi, menyimpan, mengelola, dan mengakses data di dalam komputer agar dapat digunakan secara efisien. Pemilihan struktur data yang tepat akan memengaruhi kinerja algoritma, baik dari

segi waktu eksekusi maupun penggunaan memori. Oleh karena itu, struktur data merupakan salah satu konsep fundamental dalam ilmu komputer dan pengembangan perangkat lunak.

Menurut Thomas H. Cormen dkk. (2022), struktur data adalah metode untuk menyimpan dan mengatur data sehingga operasi seperti pencarian, penyisipan, penghapusan, dan pembaruan dapat dilakukan secara efisien. Struktur data yang dipilih harus sesuai dengan karakteristik masalah yang akan diselesaikan. Sementara itu, Michael T. Goodrich, Roberto Tamassia, dan Michael H. Goldwasser (2014) menyatakan bahwa struktur data merupakan representasi logis dan fisik dari data yang dirancang untuk mendukung operasi tertentu secara efektif dan efisien.

B. Tujuan dan Pentingnya Analisis Algoritma

Analisis algoritma memiliki beberapa tujuan, yaitu:

1. Mengukur efisiensi suatu algoritma.
2. Membandingkan dua atau lebih algoritma yang menyelesaikan masalah yang sama.
3. Menentukan algoritma terbaik berdasarkan kebutuhan sistem.
4. Mengoptimalkan penggunaan memori komputer.
5. Meminimalkan waktu pemrosesan data.
6. Mendukung pengembangan perangkat lunak yang berkinerja tinggi.

Menurut Steven S. Skiena (2020), analisis algoritma membantu pengembang memahami biaya komputasi sehingga mampu memilih solusi yang paling efisien.

Dalam pengembangan perangkat lunak modern, jumlah data yang diproses semakin besar. Algoritma yang baik akan memberikan waktu proses yang lebih singkat dan penggunaan memori yang lebih hemat. Oleh karena itu, analisis algoritma menjadi bagian penting dalam pengembangan sistem informasi, kecerdasan buatan, data mining, jaringan komputer, maupun komputasi awan.

Manfaat analisis algoritma antara lain:

1. Meningkatkan performa aplikasi;
2. Mengurangi biaya komputasi;
3. Meningkatkan skalabilitas sistem;

4. Mempermudah pemeliharaan program;
5. Menghasilkan perangkat lunak yang lebih andal.

C. Jenis-Jenis Analisis Algoritma

Secara umum, ada tiga kategori analisis algoritma.

1. *Best Case*

Best Case adalah kondisi ketika algoritma bekerja pada keadaan yang paling menguntungkan sehingga membutuhkan waktu paling sedikit. Contoh: Pencarian linear menemukan data pada elemen pertama.

2. *Average Case*

Average case menggambarkan rata-rata waktu yang diperlukan algoritma berdasarkan seluruh kemungkinan *input*. Analisis ini paling sering digunakan karena mencerminkan kondisi penggunaan nyata.

3. *Worst Case*

Worst case merupakan kondisi ketika algoritma membutuhkan waktu paling lama. Contoh: Pencarian linear menemukan data pada elemen terakhir atau data tidak ditemukan.

Menurut Robert Sedgewick dan Kevin Wayne (2011), analisis *worst case* merupakan pendekatan yang paling umum digunakan karena memberikan batas atas (*upper bound*) terhadap waktu eksekusi algoritma.

D. Efisiensi Algoritma

Efisiensi algoritma terdiri atas dua aspek utama.

1. *Time Complexity*

Time Complexity adalah ukuran banyaknya operasi yang dilakukan algoritma terhadap ukuran *input*.

Misalnya:

Algoritma	Kompleksitas
<i>Linear Search</i>	$O(n)$
<i>Binary Search</i>	$O(\log n)$
<i>Bubble Sort</i>	$O(n^2)$
<i>Merge Sort</i>	$O(n \log n)$

BAB 3

TINGKAT LANJUT ARSITEKTUR SISTEM TERDISTRIBUSI DAN KOMPUTASI PARALEL

Wiranti Sri Utami, S.Kom., M.T.I.
Universitas Cendekia Abditama

A. Pendahuluan

Perkembangan teknologi informasi telah membawa perubahan yang sangat signifikan terhadap cara organisasi membangun dan mengelola sistem komputer. Pada awalnya, sebagian besar aplikasi dikembangkan menggunakan arsitektur terpusat (*centralized system*), yaitu seluruh proses komputasi dijalankan pada satu komputer atau satu *server*. Model ini cukup efektif untuk melayani jumlah pengguna yang relatif sedikit. Namun, seiring meningkatnya kebutuhan akan layanan digital yang mampu melayani jutaan pengguna secara bersamaan, sistem terpusat mulai menghadapi berbagai keterbatasan, seperti rendahnya skalabilitas, tingginya risiko *single point of failure*, serta keterbatasan dalam meningkatkan kapasitas komputasi. Oleh karena itu, organisasi mulai beralih menuju arsitektur sistem terdistribusi yang lebih fleksibel, andal, dan mudah dikembangkan (Mukherjee et al., 2024). Perlu adanya sistem terdistribusi yang mampu menghubungkan berbagai komputasi dengan layanan jaringan.

Sistem terdistribusi (*Distributed System*) merupakan sekumpulan komputer yang saling terhubung melalui jaringan dan bekerja sama untuk menyediakan suatu layanan sehingga bagi pengguna sistem

tersebut tampak sebagai satu kesatuan. Meskipun proses komputasi dilakukan pada banyak komputer yang berbeda, pengguna tidak perlu mengetahui lokasi fisik data maupun proses yang sedang dijalankan. Karakteristik inilah yang menjadikan sistem terdistribusi sebagai fondasi utama dalam pengembangan layanan digital modern seperti *cloud computing*, *e-commerce*, media sosial, layanan perbankan digital, *Internet of Things* (IoT), hingga sistem berbasis *Artificial Intelligence* (Avasthi et al., 2024).

Selain sistem terdistribusi, kebutuhan akan pemrosesan data yang semakin besar juga mendorong berkembangnya konsep komputasi paralel (*Parallel Computing*). Komputasi paralel merupakan teknik pemrosesan yang membagi suatu pekerjaan menjadi beberapa bagian sehingga dapat diproses secara bersamaan menggunakan banyak inti prosesor (*multi-core processor*), GPU (*Graphics Processing Unit*), maupun beberapa komputer dalam suatu klaster. Pendekatan ini mampu meningkatkan kecepatan komputasi secara signifikan dibandingkan dengan pemrosesan sekuensial, sehingga banyak dimanfaatkan pada analisis *Big Data*, simulasi ilmiah, *Machine Learning*, *Deep Learning*, dan *High-Performance Computing/HPC* atau komputasi berkinerja tinggi (Silva, R., et al., 2024).

Sistem terdistribusi dan komputasi paralel memiliki fokus yang berbeda. Sistem terdistribusi berorientasi pada pembagian layanan dan sumber daya ke banyak komputer yang saling berkomunikasi melalui jaringan, sedangkan komputasi paralel berfokus pada pelaksanaan beberapa proses secara serentak untuk mempercepat waktu komputasi. Integrasi kedua konsep tersebut memungkinkan organisasi membangun sistem yang memiliki tingkat skalabilitas tinggi, ketersediaan layanan yang lebih baik, serta efisiensi pemanfaatan sumber daya komputasi. Pendekatan ini telah menjadi standar dalam pengembangan aplikasi modern berbasis *cloud-native*, *microservices*, *containerization*, dan *edge computing* (Mukherjee et al., 2024).

Saat pengguna mengakses aplikasi *e-commerce*, permintaan yang dikirimkan tidak diproses oleh satu *server* saja. Permintaan tersebut didistribusikan ke berbagai layanan, seperti autentikasi pengguna, katalog produk, pembayaran, manajemen inventori, serta sistem rekomendasi. Setiap layanan dapat berjalan secara independen pada *server* yang berbeda dan memproses data secara paralel. Pendekatan ini

memungkinkan aplikasi tetap memberikan respons yang cepat meskipun melayani jutaan pengguna secara bersamaan, sekaligus meningkatkan keandalan sistem apabila salah satu layanan mengalami gangguan (Laigner et al., 2024).

Perkembangan teknologi seperti *cloud computing*, *edge computing*, *container orchestration*, *microservices*, serta komputasi berbasis GPU menunjukkan bahwa sistem terdistribusi dan komputasi paralel merupakan dua teknologi yang saling melengkapi dalam membangun infrastruktur digital modern. Oleh karena itu, pemahaman terhadap konsep, arsitektur, serta implementasi kedua teknologi tersebut menjadi kompetensi penting bagi mahasiswa Teknik Informatika maupun praktisi teknologi informasi agar mampu merancang sistem yang efisien, andal, aman, dan mampu beradaptasi terhadap kebutuhan komputasi masa depan (Silva, R., et al., 2024).

B. Arsitektur Sistem Terdistribusi Modern

Arsitektur sistem terdistribusi merupakan kerangka kerja yang menjelaskan bagaimana berbagai komponen perangkat lunak, perangkat keras, serta jaringan komputer saling berinteraksi untuk menyediakan layanan kepada pengguna. Seiring berkembangnya kebutuhan aplikasi modern yang harus mampu melayani jutaan pengguna secara bersamaan, model arsitektur sistem juga mengalami perkembangan yang sangat pesat. Perubahan tersebut ditandai dengan pergeseran dari arsitektur monolitik menuju arsitektur yang lebih modular, fleksibel, dan mudah dikembangkan seperti *Service-Oriented Architecture* (SOA) dan *Microservices* (Mukherjee et al., 2024).

Pemilihan arsitektur sistem menjadi salah satu faktor penting dalam menentukan performa, skalabilitas, keamanan, kemudahan pemeliharaan, serta kemampuan sistem untuk beradaptasi terhadap perubahan kebutuhan bisnis. Oleh karena itu, seorang pengembang perangkat lunak perlu memahami karakteristik setiap model arsitektur agar dapat menentukan solusi yang paling sesuai dengan kebutuhan organisasi (Avasthi et al., 2024).

Pada subbab ini akan dibahas lima model arsitektur sistem terdistribusi yang banyak diterapkan pada aplikasi modern, yaitu *Client-*

BAB 4

ARSITEKTUR JARINGAN KOMPUTER DAN KOMUNIKASI DATA LAPIS JAMAK

Novan Ardianto, S.Kom., M.Eng.
Universitas Mercu Buana

A. Pendahuluan

1. Latar Belakang Kebutuhan Komunikasi Data

Pada awal perkembangannya, komputer dirancang sebagai mesin yang berdiri sendiri (*stand-alone*), yang hanya berfungsi untuk memproses data secara terisolasi. Namun, seiring dengan ledakan teknologi informasi dan tingginya kebutuhan efisiensi, komputer tidak lagi bekerja secara soliter. Kebutuhan untuk berbagi sumber daya perangkat keras (seperti *printer* dan ruang penyimpanan), pertukaran informasi secara cepat, serta kemampuan mengakses sistem secara terpusat dari lokasi yang berjauhan telah mendorong lahirnya konsep penghubungan antarkomputer (Tanenbaum & Wetherall, 2023).

Kompleksitas muncul ketika perangkat dari berbagai vendor dengan sistem operasi dan perangkat keras yang berbeda harus saling berinteraksi. Tanpa adanya standar, komunikasi antarperangkat yang berbeda ini menjadi mustahil dilakukan. Oleh karena itu, dibutuhkan suatu kerangka kerja atau arsitektur standar yang dapat mengatur bagaimana data dikemas, dikirimkan, diterjemahkan, dan diterima

agar pertukaran informasi dapat berjalan tanpa kendala. Inilah yang melatarbelakangi pengembangan arsitektur jaringan komputer dan komunikasi data berlapis (lapis jamak).

2. Definisi Jaringan Komputer dan Komunikasi Data

Untuk memahami arsitektur jaringan, penting bagi kita untuk membedakan dua konsep fundamental yang saling berkaitan namun berbeda fokusnya:

- a. Jaringan komputer merupakan kumpulan dari dua atau lebih perangkat komputer (serta perangkat pendukungnya) yang saling terhubung melalui media transmisi tertentu. Tujuannya adalah untuk saling berbagi sumber daya (*resource sharing*), berkomunikasi, dan meningkatkan reliabilitas sistem.
- b. Komunikasi data merupakan proses pertukaran data (berupa sinyal digital atau analog) antara dua perangkat atau lebih melalui media transmisi. Fokus utama dari komunikasi data adalah bagaimana informasi direpresentasikan, ditransmisikan, dan dijaga keutuhannya dari pengirim (*sender*) hingga penerima (*receiver*).

Secara sederhana, jaringan komputer adalah infrastruktur atau jalurnya, sedangkan komunikasi data adalah proses atau isinya. Arsitektur jaringan berlapis hadir untuk mengatur interaksi antara infrastruktur dan proses tersebut secara terstruktur.

3. Tujuan dan Ruang Lingkup

Bab ini bertujuan untuk memberikan pemahaman komprehensif mengenai bagaimana arsitektur jaringan komputer dan komunikasi data dirancang menggunakan pendekatan berlapis (*multilayered architecture*). Pembahasan akan difokuskan untuk membedah kompleksitas komunikasi data menjadi bagian-bagian yang lebih kecil dan terkelola (lapisan).

Ruang lingkup materi yang akan dibahas dalam bab ini meliputi:

- a. Konsep dasar sistem komunikasi data, mulai dari komponen, arah transmisi, hingga jenis sinyal.
- b. Pengenalan arsitektur jaringan komputer dan komponen penyusunnya.
- c. Konsep mendasar arsitektur lapis jamak, termasuk mekanisme enkapsulasi dan interaksi antar lapisan.

- d. Kajian mendalam terhadap dua model referensi arsitektur jaringan utama di dunia: Model Referensi OSI (7 lapisan) dan Model TCP/IP (4 lapisan).
- e. Studi kasus alur perjalanan data (*data flow*) dari satu komputer ke komputer lain melalui media jaringan.

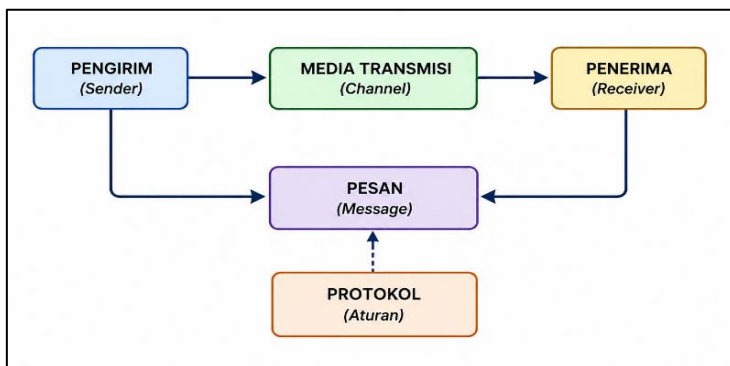
Melalui pemahaman atas ruang lingkup ini, diharapkan pembaca dapat memahami filosofi mengapa komunikasi data modern dibangun secara berlapis dan bagaimana setiap lapisan berkontribusi dalam menjamin keberhasilan transmisi data.

B. Konsep Dasar Komunikasi Data

Sebelum memahami bagaimana arsitektur jaringan berlapis bekerja, kita perlu memahami terlebih dahulu elemen-elemen dasar yang memungkinkan terjadinya pertukaran informasi. Komunikasi data adalah proses pengiriman data dari satu perangkat ke perangkat lainnya melalui media transmisi. Proses ini melibatkan berbagai komponen, arah, mode, hingga jenis sinyal yang digunakan.

1. Komponen Sistem Komunikasi Data

Setiap sistem komunikasi data, seberapa pun sederhana atau kompleksnya, terdiri dari lima komponen utama yang saling bekerja sama.



Gambar 4.1: Komponen Sistem Komunikasi Data

Sumber: Diolah Penulis

a. Pengirim (*Sender*)

Pengirim adalah perangkat atau entitas yang menginisiasi komunikasi dengan mengirimkan data. Pengirim bertanggung

BAB 5

KRIPTOGRAFI MODERN DAN KETAHANAN SIBER (*CYBER RESILIENCE*)

Fandi Ali Mustika, S.Kom., M.T.
Universitas Mercu Buana

A. Pendahuluan

Transformasi digital telah menjadi salah satu pendorong utama perubahan dalam berbagai aspek kehidupan masyarakat, termasuk sektor pemerintahan, pendidikan, kesehatan, keuangan, manufaktur, dan industri kreatif. Perkembangan teknologi seperti *Cloud Computing*, *Internet of Things* (IoT), *Artificial Intelligence* (AI), Big Data, serta jaringan komunikasi berkecepatan tinggi telah meningkatkan efisiensi operasional sekaligus mempercepat proses pengambilan keputusan berbasis data. Di sisi lain, digitalisasi juga menyebabkan meningkatnya jumlah data yang dipertukarkan melalui jaringan publik maupun privat, sehingga aspek keamanan informasi menjadi salah satu prioritas utama dalam pembangunan ekosistem digital.

Bersamaan dengan pesatnya transformasi digital tersebut, ancaman keamanan siber (*cyber threats*) berkembang semakin kompleks, baik dari sisi teknik, skala, maupun dampaknya. Serangan

siber tidak lagi hanya dilakukan oleh individu, tetapi juga melibatkan kelompok kriminal terorganisasi, pelaku spionase siber, hingga aktor yang didukung negara (*state-sponsored actors*). Bentuk serangan seperti *ransomware*, *phishing*, *Distributed Denial of Service* (DDoS), *Advanced Persistent Threat* (APT), pencurian identitas digital, eksploitasi kerentanan perangkat lunak, dan kebocoran data menjadi ancaman nyata bagi organisasi maupun individu. Dampaknya tidak hanya berupa kerugian finansial, tetapi juga menurunnya kepercayaan publik, terganggunya layanan digital, serta hilangnya aset informasi yang bernilai strategis.

Data merupakan aset yang sangat berharga pada era ekonomi digital. Oleh karena itu, perlindungan terhadap data menjadi kebutuhan mendasar dalam setiap sistem informasi. Salah satu teknologi yang memiliki peran penting dalam menjaga keamanan data adalah kriptografi. Kriptografi merupakan ilmu dan teknik yang digunakan untuk mengamankan informasi melalui proses transformasi data sehingga hanya pihak yang memiliki hak akses yang dapat membaca maupun memodifikasi informasi tersebut. Dengan penerapan algoritma kriptografi yang tepat, aspek kerahasiaan (*confidentiality*), integritas (*integrity*), otentikasi (*authentication*), dan non-repudiation dapat diwujudkan sehingga risiko penyalahgunaan informasi dapat diminimalkan.

Perkembangan kriptografi telah mengalami evolusi yang sangat signifikan. Pada awalnya, teknik kriptografi hanya menggunakan metode sederhana berbasis substitusi dan transposisi karakter. Seiring meningkatnya kemampuan komputasi, algoritma kriptografi modern berkembang menjadi lebih kompleks dengan memanfaatkan pendekatan matematika dan teori bilangan. Saat ini berbagai algoritma seperti *Advanced Encryption Standard* (AES), *Rivest-Shamir-Adleman* (RSA), *Elliptic Curve Cryptography* (ECC), serta fungsi hash SHA-2 dan SHA-3 telah menjadi standar internasional dalam melindungi komunikasi digital, transaksi elektronik, layanan perbankan, *e-commerce*, hingga infrastruktur kritis.

Namun demikian, penerapan kriptografi saja tidak cukup untuk menghadapi dinamika ancaman siber modern. Organisasi tidak hanya dituntut untuk mampu mencegah serangan, tetapi juga harus memiliki kemampuan untuk mendeteksi, merespons, mempertahankan layanan,

serta memulihkan sistem secara cepat ketika insiden keamanan terjadi. Konsep inilah yang dikenal sebagai *Cyber Resilience* atau ketahanan siber. *Cyber Resilience* menggabungkan aspek keamanan informasi, manajemen risiko, keberlangsungan bisnis (*Business Continuity*), pemulihan bencana (*Disaster Recovery*), serta tata kelola teknologi informasi menjadi suatu pendekatan yang menyeluruh.

Berbeda dengan pendekatan keamanan tradisional yang berorientasi pada pencegahan serangan, *Cyber Resilience* menekankan kemampuan organisasi untuk tetap menjalankan fungsi bisnis meskipun mengalami gangguan atau serangan siber. Pendekatan ini semakin relevan mengingat tidak ada sistem yang sepenuhnya kebal terhadap serangan. Oleh sebab itu, organisasi harus mampu membangun sistem yang adaptif, tangguh, dan memiliki mekanisme pemulihan yang efektif.

Dalam konteks komputasi modern, integrasi antara kriptografi dan *Cyber Resilience* menjadi fondasi penting dalam menjaga kepercayaan terhadap layanan digital. Teknologi enkripsi melindungi kerahasiaan informasi, sedangkan strategi *Cyber Resilience* memastikan keberlangsungan operasional organisasi ketika menghadapi ancaman maupun insiden keamanan. Kombinasi keduanya menjadi salah satu pilar utama dalam pembangunan ekosistem digital yang aman, andal, dan berkelanjutan.

Bab ini membahas konsep dasar kriptografi modern, perkembangan algoritma yang digunakan dalam pengamanan informasi, serta keterkaitannya dengan strategi *Cyber Resilience*. Selain itu, dibahas pula implementasi kriptografi pada berbagai layanan digital, tantangan yang dihadapi akibat perkembangan teknologi seperti komputasi kuantum dan kecerdasan buatan, serta pentingnya membangun ketahanan siber sebagai bagian dari transformasi digital di abad ke-21.

BAB 6

REKAYASA PERANGKAT LUNAK MODERN DAN MANAJEMEN SIKLUS HIDUP (SDLC)

Budi Chehabudin, M.Kom.
Universitas Garut

Dewasa ini, hampir tidak ada ranah kehidupan manusia, pendidikan, layanan kesehatan, birokrasi, transportasi, hingga interaksi sosial yang lepas dari sistem berbasis komputer. Ketergantungan ini menuntut pembangunan perangkat lunak dilakukan secara terencana, sebab kegagalan sistem sekecil apa pun dapat menimbulkan kerugian finansial, rusaknya kepercayaan pengguna, hingga terhentinya layanan publik.

Rekayasa perangkat lunak tumbuh sebagai bidang keilmuan yang mengatur proses pembangunan sistem agar produk yang dihasilkan efisien, aman, dan mudah dirawat, dengan menyerap prinsip manajemen proyek dan psikologi kognitif agar relevan dengan kebutuhan pengguna (Sommerville, 2021). Bab ini menguraikan konsep dasar, siklus hidup pengembangan, metodologi, serta tantangan dan arah perkembangan rekayasa perangkat lunak modern.

A. Konsep Dasar Rekayasa Perangkat Lunak Modern

1. Pengertian Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah penerapan cara kerja sistematis dan terukur dalam membangun perangkat lunak, mencakup penggalan kebutuhan, perancangan, penulisan kode, pengujian, penerapan, hingga pemeliharaan agar sistem tetap optimal, guna menekan risiko kegagalan proyek, mengendalikan biaya, dan mempersingkat waktu pengerjaan (Pressman & Maxim, 2020). *Coding* sendiri hanya sebagian kecil dari proses ini; pemrogram yang cakap belum tentu menghasilkan sistem bermutu tanpa pemahaman kebutuhan bisnis, arsitektur yang matang, serta strategi pengujian dan pemeliharaan yang memadai (Sommerville, 2021), sejalan dengan SWEBOK yang mendefinisikan rekayasa perangkat lunak sebagai pendekatan sistematis, disiplin, dan terukur dalam pengembangan, pengoperasian, dan pemeliharaan perangkat lunak (IEEE Computer Society, 2014).

Sebagai ilustrasi, kegagalan sistem informasi rumah sakit sering kali bukan disebabkan oleh lemahnya penguasaan bahasa pemrograman, melainkan oleh analisis kebutuhan alur kerja tenaga medis yang tidak digali secara memadai sejak awal, sebagaimana ditegaskan pula oleh ISO/IEC/IEEE 12207 (2017) mengenai proses siklus hidup perangkat lunak.

2. Perjalanan dan Perkembangan Rekayasa Perangkat Lunak

Pada masa awal komputasi, perangkat lunak dibangun secara sederhana tanpa metode baku karena skala sistem masih kecil. Seiring meningkatnya kompleksitas, banyak proyek mengalami keterlambatan, pembengkakan anggaran, bahkan kegagalan total, sehingga muncul model terstruktur seperti *Waterfall*, *Spiral*, *Prototype*, dan *Incremental* (Pressman & Maxim, 2020).

Perkembangan internet, perangkat bergerak, dan komputasi awan mendorong lahirnya pendekatan *Agile* yang lebih luwes merespons perubahan kebutuhan, ditandai dengan kelahiran Manifesto Agile pada 2001 sebagai reaksi terhadap kekakuan model tradisional (Beck et al., 2001).

3. Ciri Khas Perangkat Lunak Masa Kini

Perangkat lunak masa kini menampilkan karakteristik yang saling berkaitan: orientasi pengguna (*user experience*) sejak perancangan awal, kemudahan integrasi antarsistem, keamanan yang melindungi data dari kebocoran maupun serangan siber, termasuk kepatuhan terhadap regulasi data pribadi, serta kemudahan pemeliharaan (*maintainability*) berkat struktur sistem yang rapi.

Karakteristik tersebut tidak dapat dipandang terpisah; sistem skalabel namun lemah keamanannya tetap berpotensi merugikan besar, seperti terlihat pada sejumlah insiden kebocoran data *platform e-commerce* belakangan ini.

4. Peran Rekayasa Perangkat Lunak dalam Ekosistem Digital

Di luar aspek teknis, rekayasa perangkat lunak modern berperan strategis dalam transformasi digital organisasi; insinyur perangkat lunak kini menjadi mitra strategis yang memberi masukan terkait kelayakan teknologi, biaya, dan risiko pengembangan sistem. Penguasaan konsep dasarnya menjadi fondasi penting sebelum mempelajari topik lanjutan seperti arsitektur perangkat lunak dan manajemen proyek TI (Pressman & Maxim, 2020).

B. Siklus Hidup Pengembangan Perangkat Lunak (SDLC)

1. Pengertian dan Tujuannya

Software Development Life Cycle (SDLC) merupakan rangkaian proses logis dan sistematis yang digunakan industri perangkat lunak untuk merancang, membangun, menguji, hingga merawat sistem, berfungsi sebagai cetak biru yang memandu tim bekerja secara runtut hingga produk siap dipakai (ISO/IEC/IEEE 12207, 2017). Penerapannya bertujuan menghasilkan produk bermutu, menekan pemborosan biaya dan waktu, mengelola risiko sedini mungkin, serta menjaga transparansi bagi seluruh pemangku kepentingan (Pressman & Maxim, 2020).

2. Alasan Pentingnya SDLC

Penerapan SDLC memberi manfaat nyata: *mencegah scope creep*, memudahkan alih tugas berkat dokumentasi tertata, memungkinkan

BAB 7

CLOUD COMPUTING DAN INFRASTRUKTUR EDGE TO CLOUD

Martono, S.Pd.Kim., M.TI.
Universitas Raharja

A. Pendahuluan

Selama beberapa dekade terakhir, semakin banyak industri yang memperoleh manfaat dengan memindahkan aplikasi TI mereka ke *cloud*, kemudian memanfaatkan skalabilitas dan kemampuannya dalam pemrosesan *big data*. *Cloud Computing* mengandalkan pemrosesan data terpusat di pusat data yang besar dan berkinerja tinggi. Pendekatan ini ideal untuk aplikasi yang membutuhkan daya komputasi besar, penyimpanan berskala luas, dan pengelolaan data secara terpusat. Sifat terpusat dari *Cloud Computing* memungkinkan skalabilitas yang sangat besar dengan kemampuan sumber daya yang dapat dialokasikan secara dinamis sesuai permintaan atau kebutuhan. Aplikasi yang berjalan di *cloud* sering kali memperoleh banyak manfaat berupa ketersediaan tinggi. Hal ini karena pusat data dirancang dengan kemampuan *redundansi* dan *failover*.

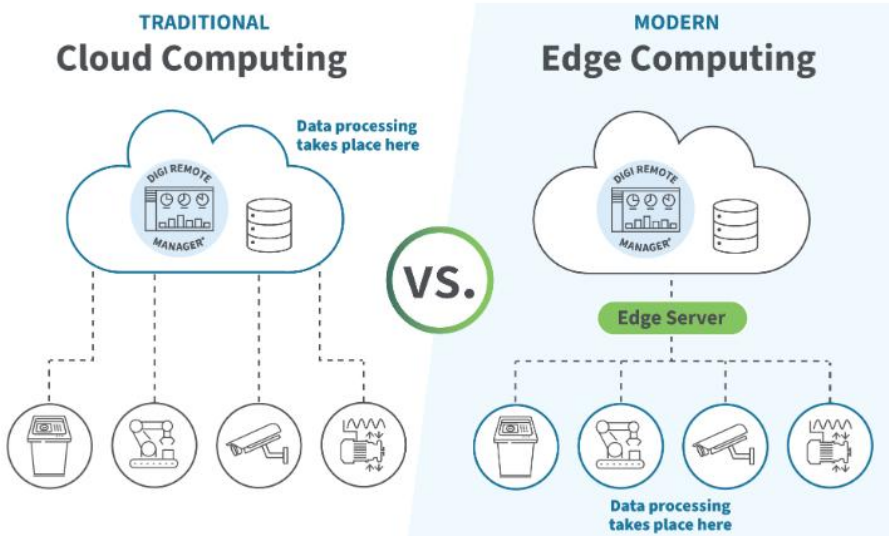
Sebaliknya, *edge computing* melakukan proses desentralisasi dengan menempatkan proses lebih dekat ke sumber data. Misalnya: perangkat IoT, sensor, atau server lokal. Kedekatan ini akan memangkas waktu pemrosesan data sehingga memungkinkan pengambilan keputusan dan respons secara *real-time*. *Edge computing* sangat bermanfaat dalam situasi yang menuntut latensi rendah dan pemrosesan segera seperti pada kendaraan otonom, *smart city* dan otomatisasi industri.

Edge computing juga mengurangi ketergantungan pada *bandwidth* dan konektivitas *cloud* dengan memproses data secara lokal. Desentralisasi ini dapat menghasilkan penggunaan sumber daya yang lebih efisien daripada sebelumnya karena data yang tidak perlu dikirim ke *cloud* tidak akan dikirim. Hal ini tentu akan menghemat *bandwidth* sekaligus energi. *Edge computing* merupakan filosofi jaringan yang berfokus untuk menempatkan pemrosesan data sedekat mungkin dengan sumber data guna mengurangi latensi dan penggunaan *bandwidth*. Secara lebih sederhana, *penerapan edge computing* berarti mengurangi jumlah proses yang dijalankan di *cloud* dan memindahkan proses-proses tersebut ke lokasi lokal, seperti komputer pengguna, perangkat IoT atau *server edge*. Membawa komputasi ke *network edge* akan meminimalkan komunikasi jarak jauh yang harus terjadi antara klien dan server.

Bagi perangkat internet, *network edge* merupakan titik di mana perangkat atau jaringan lokal yang memuat perangkat tersebut berkomunikasi dengan internet. Istilah ‘tepi’ atau ‘*edge*’ ini kadang agak samar definisinya. Misalnya, komputer pengguna atau prosesor di dalam kamera IoT dapat dianggap sebagai *network edge*, namun router pengguna, ISP, atau *server edge* lokal juga dapat dikategorikan sebagai bagian dari *network edge* tersebut. Poin pentingnya adalah bahwa *network edge* terletak secara geografis dekat dengan perangkat, berbeda halnya dengan server asal (*origin server*) dan server *cloud* yang bisa jadi berada sangat jauh dari perangkat yang berkomunikasi dengannya.

B. Perbedaan Edge dan Cloud Computing

Gambar berikut ini akan memberikan ilustrasi perbedaan antara *cloud* tradisional dan *Edge Cloud*:



Gambar 7.1: Perbedaan *Cloud* tradisional dan *Edge Cloud*

Sumber: <https://www.digi.com/blog/post/edge-computing-vs-cloud-computing>

Dalam *edge computing*, tugas pemrosesan data dijalankan di dekat lokasi asal data tersebut dihasilkan guna memungkinkan waktu respons yang cepat. Di sisi *edge*, data dapat diproses lebih awal, disaring, atau diagregasikan sehingga jumlah data yang perlu dikirim ke *cloud* akan menjadi lebih sedikit.

Saat membandingkan manfaat komputasi *edge* dan *cloud*, ada banyak aspek yang perlu dipertimbangkan, seperti:

1. Performa: Dengan *edge computing*, lebih mudah untuk mencapai latensi rendah dan *bandwidth* tinggi karena kedekatan fisik antara *node* komputasi dan sumber data. Hal ini juga berarti bahwa volume data yang lebih besar dapat diproses dengan waktu respons yang lebih cepat.
2. Skalabilitas: Skalabilitas dan elastisitas merupakan keunggulan utama yang dimiliki oleh *cloud computing*. Hal ini harus menjadi perhatian.

DAFTAR PUSTAKA

- Al-Zewairi, M., Al-Zewairi, M., Otoom, A., & Almaaitah, N. (2020). Experimental Evaluation of TCP and UDP Performance in IoT Networks. *IEEE Access*, 8, 222178–222189. <https://doi.org/10.1109/ACCESS.2020.3043491>
- Association For Computing Machinery (ACM), & IEEE Computer Society. (2020). *Computing Curricula 2020: Paradigms For Global Computing Education*. ACM & IEEE Computer Society.
- Avasthi, S., Tripathi, S. L., Dhanda, N., & Verma, S. B. (Eds.). (2024). *Decentralized Systems and Distributed Computing*. Wiley.
- Ayu Sekar Pertiwi, N., Umardiyah, F., Naufal Mansyur, M., Munir, M., Sapi, I., Sholichah, A., Nailatu Fudlah, T., & A Wahab Hasbullah, U. K. (2024). *Sosialisasi Kesadaran Keamanan Digital di Era Revolusi Industri 4.0* (Vol. 5, Issue 1).
- Bass, L., Clements, P., & Kazman, R. (2025). *Software Architecture in Practice* (5th ed.). Addison-Wesley Professional.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for agile software development. Agile Alliance. <https://agilemanifesto.org/>
- Bhatnagar, S., & Mahant, R. (2025). *The Art of Decoding Microservices: An In-Depth Exploration of Modern Software Architecture*. Apress. <https://doi.org/10.1007/979-8-8688-1267-5>
- Brookshear, J. G., & Brylow, D. (2019). *Computer Science: An Overview (13th ed.)*. Pearson.
- Campbell-Kelly, M., Aspray, W., Ensmenger, N., & Yost, J. R. (2014). *Computer: A History of the Information Machine (3rd ed.)*. Westview Press.
- Ceruzzi, P. E. (2012). *Computing: A Concise History*. MIT Press.
- Christudas, B., & Telang, T. (2025). *Practical Microservices Architectural Patterns: Build Highly Scalable Distributed Applications with Spring*

- Boot 3 and Spring Cloud* (2nd ed.). Apress.
<https://doi.org/10.1007/979-8-8688-1606-2>
- Computer History Museum. (2023). *Timeline of Computer History*.
- Daniel, J., Mota, G., Wang, X., & Guerra, E. (2025). *Architecture Refactoring Towards Service Reusability in the Context of Microservices*. Dalam *Agile Processes in Software Engineering and Extreme Programming (XP 2025)*. Springer.
- Denning, P. J. (1989). *Computer Science: The Discipline*. In *Encyclopedia of Computer Science*. Macmillan.
- Denning, P. J. (2017). Remaining Trouble Spots With Computational Thinking. *Communications of The ACM*, 60(6), 33–39.
- Denning, P. J., & Martell, C. H. (2015). *Great Principles of Computing*. MIT Press.
- Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. MIT Press.
- Donald E. Knuth. (1997). *The Art of Computer Programming, Volume 1: Fundamental Algorithms* (3rd ed.). Addison-Wesley.
- Hohpe, G., & Woolf, B. (2024). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional.
- IEEE Computer Society. (2014). Guide to the software engineering body of knowledge (SWEBOK) (Version 3.0). IEEE Computer Society Press.
- International Organization for Standardization. (2006). Software engineering — Software life cycle processes — Maintenance (ISO/IEC 14764:2006).
- International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers. (2017). Systems and software engineering — Software life cycle processes (ISO/IEC/IEEE 12207:2017).
- ISO/IEC 27001:2022 Information Security, Cybersecurity and Privacy Protection---Information Security Management Systems---Requirements* (Issue ISO/IEC 27001:2022). (2022). ISO.
- Jon Kleinberg, & Éva Tardos. (2006). *Algorithm Design*. Addison-Wesley.

- Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations* (2nd ed.). IT Revolution Press.
- Knuth, D. E. (1997). *The Art of Computer Programming* (3rd ed.). Addison-Wesley.
- Kuhn, T. S. (2012). *The Structure of Scientific Revolutions* (4th ed.). University of Chicago Press.
- Kurose, J. F., & Ross, K. W. (2021). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.
- Laigner, Rodrigo., and Zhou, Yongluan. (2024). *Benchmarking Data Management Systems for Microservices*. airXiv. <https://doi.org/10.48550/arXiv.2405.11529>
- Majumdar, S. (2024). *Introduction*. In *Resource Management on Distributed Systems*. Wiley. <https://doi.org/10.1002/9781119912965.ch1>
- Michael T. Goodrich, Roberto Tamassia, & Michael H. Goldwasser. (2014). *Data Structures and Algorithms in Python*. Wiley.
- Mukherjee, A., De, D., & Buyya, R. (Eds.). (2024). *Resource Management in Distributed Systems*. Springer.
- National Institute of Standards and Technology. (2024). *The NIST Cybersecurity Framework (CSF) 2.0*. U.S. Department of Commerce. <https://www.nist.gov/cyberframework>
- Newman, S. (2021). *Building microservices: Designing fine-grained systems* (2nd ed.). O'Reilly Media.
- Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv. <https://doi.org/10.48550/arXiv.2302.06590>
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and Software Technology*, 141, Article 106700. <https://doi.org/10.1016/j.infsof.2021.106700>

- Richard Bellman. (1957). *Dynamic Programming*. Princeton University Press.
- Richards, M. (2024). *Software Architecture: The Hard Parts* (2nd ed.). O'Reilly Media.
- Robert Sedgewick, & Kevin Wayne. (2011). *Algorithms* (4th ed.). Addison-Wesley.
- Ross, R., Pillitteri, V., Dempsey, K., Riddle, M., & Guissanie, G. (2020). *Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations* (Issue SP 800-171 Revision 2). <https://doi.org/10.6028/NIST.SP.800-171r2>
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Sahoo, K. S., Tiwari, P. K., Sahoo, B., & Obaidat, M. S. (2021). SDLBA: A Software-Defined Layered Based Architecture for IoT. *IEEE Internet of Things Journal*, 8(24), 17504–17517. <https://doi.org/10.1109/JIOT.2021.3094026>
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). John Wiley & Sons.
- Silva, R., et al. (2025). State of the Art in Parallel and Distributed Systems: Emerging Trends and Challenges. *Electronics*, 14(4), 677.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- Sommerville, I. (2021). *Software engineering* (10th ed.). Pearson Education.
- Stallings, W. (2019). *Data and Computer Communications* (12th ed.). Pearson.
- Stallings, W. (2020). *Computer Organization And Architecture* (11th ed.). Pearson.
- Stallings, W. (2023). *Cryptography and Network Security: Principles and Practice* (9th ed.). Pearson.
- Steven S. Skiena. (2020). *The Algorithm Design Manual* (3rd ed.). Springer.
- Steven S. Skiena. (2020). *The Algorithm Design Manual* (3rd ed.). Springer.
- Sulistyo, F.; Mustika, F. (2020). LOAD BALANCING PROTOCOL LEACH IN WIRELESS SENSOR NETWORK. *Conference on Electrical*

Engineering, Telematics, Industrial Technology, and Creative Media (CENTIVE 2019), 25–31. <http://conferences.ittelkom-pwt.ac.id/index.php/centive/article/view/83>

- Tanenbaum, A. S., & Austin, T. (2013). *Structured Computer Organization (6th ed.)*. Pearson.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems (4th ed.)*. Pearson.
- Tanenbaum, A. S., & Wetherall, D. J. (2023). *Computer Networks (6th ed.)*. Pearson Education.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, & Clifford Stein. (2022). *Introduction to Algorithms (4th ed.)*. MIT Press.
- Turing, A. M. (1936). On Computable Numbers, With An Application to The Entscheidungs Problem. *Proceedings of The London Mathematical Society*, 42(2), 230–265.
- Vernon, V. (2024). *Domain-Driven Design Distilled: Applying Domain-Driven Design to Modern Enterprise Applications*. Addison-Wesley Professional.
- Vogel, O., Arnold, I., Chughtai, A., & Kehrner, T. (2025). *Software Architecture: A Comprehensive Framework and Guide*. Springer.
- Wolff, E. (2024). *Microservices: Flexible Software Architecture*. Addison-Wesley Professional.
- Yang, J. (2025). *Cloud Computing and MicroServices*. Springer. <https://doi.org/10.1007/978-3-031-93478-0>

BIOGRAFI PENULIS



Dr. Nur Azizah, S.Kom., M.Akt., M.Kom.

Penulis lulus S1 Program Studi Sistem Informasi di Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Budi Luhur Jakarta dan menyelesaikan Program Studi Magister Akuntansi S2 di Fakultas Ekonomi dan Bisnis Universitas Budi Luhur Jakarta dan Program Studi Magister Komputer S2 di Fakultas Teknologi Universitas Budi Luhur Jakarta. Lulus Program Doktor S3 pada Fakultas Teknologi Informasi Universitas Gunadarma Jakarta. Saat ini, sebagai Dekan Fakultas Ekonomi dan Bisnis Universitas Raharja merangkap sebagai Ketua Program Studi Teknik Informatika Program Magister.

Penulis mengampu mata kuliah Rekayasa Perangkat Lunak, Sistem Informasi Akuntansi, Sistem Informasi Manajemen, Sistem Basis Data, *Business Intelligence Management*, *IT Risk Management*, *Expert System*, *Machine Learning* dan Pengantar Kecerdasan Buatan. Aktif sebagai Pengurus Pusat Asosiasi Perguruan Tinggi Ilmu Komputer (APTIKOM), sebagai pengurus Asosiasi Perguruan Tinggi Swasta (APTISI) Wilayah IVB-Banten dan sebagai Pengurus Asosiasi Dosen Indonesia (ADI) Banten. Aktif menulis di beberapa jurnal nasional dan internasional dan buku-buku yang sudah terbit. Aktif sebagai pendakwah di majelis-majelis Ta'lim Muslimah.

Email Penulis: nur.azizah@raharja.info.



Dedy Iskandar, S.Kom., M.T.I.

Penulis lulus S1 Program Studi Sistem Informasi di Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Bina Darma Palembang dan menyelesaikan Program Studi Magister Teknik Informatika di Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Raharja Tangerang. Saat ini, saya adalah dosen Fakultas Sains dan

Teknologi Universitas Raharja. Saya juga masih mengajar di SMK Pustek Serpong, dan juga sebagai tutor online pada Universitas Terbuka. Penulis mengampu mata kuliah Rekayasa Sistem, Logika Algoritma dan Pemrograman, Sistem Basis Data, Organisasi Komputer, Jaringan Komputer, Struktur Data. Aktif sebagai penulis di beberapa jurnal nasional yang terindeks Sinta 3 dan Sinta 4. Sekarang saya juga aktif di beberapa organisasi seperti Partai Golkar sebagai Ketua Ranting dan Kosgoro sebagai Ketua Pelatihan dan Pengembangan.

Email Penulis: iskandar@raharja.info.



Wiranti Sri Utami, S.Kom., M.T.I.

Berbekal ketertarikan yang tinggi terhadap perkembangan teknologi informasi sejak masa sekolah, penulis memilih untuk menekuni bidang informatika sebagai jalur pendidikan dan karier profesional. Ketertarikan tersebut menjadi motivasi untuk terus memperdalam kompetensi dalam pengembangan perangkat lunak, analisis data, serta penerapan teknologi informasi dalam mendukung penyelesaian

berbagai permasalahan di berbagai sektor. Pendidikan tinggi ditempuh pada Program Studi Teknik Informatika hingga jenjang sarjana, kemudian dilanjutkan pada Program Magister Teknik Informatika dengan konsentrasi *Business Intelligence*. Melalui proses akademik tersebut, penulis memperoleh pemahaman yang komprehensif mengenai pengembangan sistem informasi, pengelolaan dan analisis data, serta pemanfaatan teknologi sebagai dasar dalam pengambilan keputusan yang efektif.

Sebagai akademisi, penulis mengabdikan diri pada dunia pendidikan tinggi dengan menjalankan tugas sebagai dosen di bidang informatika. Berbagai mata kuliah yang diampu meliputi Algoritma dan Pemrograman, Pemrograman Visual, Sistem Berbasis Pengetahuan, *Wireless/Mobile Computing*, serta Rekayasa Perangkat Lunak. Selain melaksanakan kegiatan pembelajaran, penulis juga aktif melakukan penelitian, menyusun buku ajar dan buku referensi, serta

memublikasikan hasil penelitian pada jurnal ilmiah nasional sebagai bagian dari komitmen terhadap pengembangan ilmu pengetahuan. Fokus keilmuan penulis mencakup Rekayasa Perangkat Lunak, Kecerdasan Buatan (*Artificial Intelligence*), *Machine Learning*, *Business Intelligence*, Sistem Pendukung Keputusan, dan Analisis Data. Melalui pelaksanaan Tridharma Perguruan Tinggi, penulis terus berupaya menghasilkan karya akademik yang inovatif dan aplikatif serta memberikan kontribusi bagi perkembangan ilmu pengetahuan, dunia pendidikan, serta kemajuan teknologi informasi di Indonesia.

Email Penulis: wirantisriutami1@gmail.com



Novan Ardianto, S.Kom., M.Eng.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2000 silam. Hal tersebut membuat penulis memilih untuk masuk ke Universitas Bina Nusantara di Kota Jakarta dengan memilih jurusan Komputerisasi Akuntansi dan berhasil lulus pada tahun 2004. Penulis kemudian melanjutkan pendidikan S2 di

Universitas Gadjah Mada dengan mengambil jurusan Magister Teknik Informatika (MTI) dengan program beasiswa dari Pemprov DKI Jakarta dan berhasil lulus pada tahun 2012. Penulis memiliki kepakaran di bidang *server*, *cybersecurity* dan *web*. Dan untuk mewujudkan karier sebagai dosen profesional, penulis pun aktif sebagai peneliti di bidang kepakarannya tersebut. Selain peneliti, penulis juga aktif sebagai praktisi di salah satu instansi pemerintah sebagai kepala unit.

Email Penulis: novan.ardianto@mercubuana.ac.id



Fandi Ali Mustika, S.Kom., MT

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2008 silam. Hal tersebut membuat penulis memilih untuk masuk ke Pondok Pesantren Daar El-Qolam dengan memilih Jurusan Ilmu Pengetahuan Alam (IPA) dan berhasil lulus pada tahun 2009. Penulis kemudian melanjutkan pendidikan ke perguruan tinggi dan berhasil menyelesaikan studi S1 di Prodi Teknik Informatika Universitas

Mercu Buana pada tahun 2013. Pada tahun 2014, penulis baru masuk S2 di Universitas Mercu Buana Program Pascasarjana. Dua tahun kemudian, penulis menyelesaikan studi S2 di Prodi Teknik Elektro Program Pascasarjana Universitas Mercu Buana. Penulis memiliki kepakaran di bidang *Wireless Sensor*, *Cloud Computing* dan *Cyber Security*. Dan untuk mewujudkan karier sebagai dosen profesional, penulis pun aktif sebagai peneliti di bidang kepakarannya tersebut. Selain peneliti, penulis juga aktif sebagai praktisi di salah satu perusahaan swasta sebagai Operation Manager.

Email Penulis: fandi.ali@mercubuana.ac.id



Budi Chehabudin, M.Kom.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2004 silam. Hal tersebut membuat penulis memilih untuk masuk ke STMIK LPKIA Bandung dengan memilih jurusan Manajemen Informatika dan berhasil lulus diploma pada tahun 2008. Penulis kemudian melanjutkan pendidikan ke program sarjana di perguruan tinggi yang sama dan berhasil

menyelesaikan studi S1 di jurusan Sistem Informasi STMIK LPKIA Bandung pada tahun 2009. Enam tahun kemudian, penulis menyelesaikan studi S2 di jurusan Sistem Informasi Program Pascasarjana STMIK LIKMI.

Penulis memiliki kepakaran di bidang Web Technology dan Database. Pernah mengajar di beberapa perguruan tinggi. Dan untuk mewujudkan karier sebagai dosen profesional, penulis pun aktif sebagai peneliti di bidang kepakarannya tersebut. Beberapa penelitian telah dilakukan di perguruan tinggi. Selain peneliti, penulis juga menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Atas dedikasi dan kerja keras dalam menulis buku, mudah-mudahan bisa diterima oleh praktisi maupun pemula.

Email Penulis: chehacool@gmail.com



Martono, S.Pd.Kim., M.Ti.

Keterarikan penulis pada bidang komputer awalnya hanya karena hobi dan iseng-iseng. Sambil menyelesaikan pendidikan kimia jenjang Diploma 3 di IKIP Jakarta, pada saat itu penulis juga mengikuti pelatihan elektronika komputer. Belajar tentang jaringan komputer dimulai dari era *Novell Netware* dan *Microsoft NT 4.0* pada saat jaringan LAN yang masih populer menggunakan kabel koaksial hingga kini era jaringan menggunakan Wi-Fi dan Fiber Optics. Hal tersebut lebih karena tuntutan pekerjaan.

Pendidikan D3 diselesaikan di IKIP Jakarta dan S1 diselesaikan di Universitas Terbuka pada tahun 2006, masih pada jurusan yang sama, yakni Pendidikan Kimia. Penulis melanjutkan Pendidikan S2 di STMIK Raharja pada program studi Teknik Informatika dan selesai pada tahun 2017. Awalnya penulis bekerja di perusahaan yang merupakan vendor di bidang teknologi informasi yang melayani perusahaan untuk menangani *hardware*, sistem teknologi informasi, jaringan dan keamanan komputer. Kemudian penulis bergabung dengan perusahaan teknologi informasi yang memberikan pelatihan-pelatihan ke sekolah-sekolah dan juga mengembangkan *software-software* pendidikan dan pelatihan robotika.

Dari sini penulis akhirnya banyak melakukan eksplorasi *software*, network, mikrokontroler dan robotika serta terlibat baik secara langsung maupun tidak langsung dalam pengembangan *software*.

Beberapa tulisan ringan ditulis dalam bentuk *blog* dan sering menggunakan *nickname* martonokita. Adapun tulisan ilmiahnya telah diterbitkan dalam beberapa buku dan dalam beberapa Jurnal Ilmiah. Saat ini penulis memberikan pelatihan di lembaga pendidikan, mengajar di Universitas Raharja dan beberapa perguruan tinggi.

Email Penulis: martono@raharja.info

KOMPUTASI KOMPREHENSIF

Teori Formal Hingga Akselerasi Digital Abad 21

Perkembangan ilmu komputer dan teknologi informasi berlangsung sangat cepat, didorong oleh kemajuan komputasi, kecerdasan buatan (*Artificial Intelligence*), komputasi awan (*Cloud Computing*), *Internet of Things* (IoT), komputasi berkinerja tinggi (*High Performance Computing*), analisis data besar (*Big Data Analytics*), hingga komputasi kuantum (*Quantum Computing*). Berbagai inovasi tersebut telah mengubah cara manusia bekerja, belajar, berkomunikasi, berbisnis, serta mengambil keputusan berbasis data. Oleh karena itu, pemahaman terhadap konsep-konsep fundamental komputasi sekaligus perkembangan teknologi mutakhir menjadi kebutuhan yang sangat penting bagi mahasiswa, dosen, peneliti, praktisi, maupun masyarakat luas. Buku "Komputasi Komprehensif: Teori Formal Hingga Akselesrasi Digital Abad 21" bertujuan untuk memberikan wawasan yang komprehensif mengenai konsep, strategi, dan implementasi CRM dalam dunia bisnis modern, tersusun dalam tujuh bab sebagai berikut: (1) Evolusi Paradigma dan Taksonomi Ilmu Komputer, (2) Analisis Algoritma dan Struktur Data, (3) Tingkat Lanjut Arsitektur Sistem Terdistribusi dan Komputasi Paralel, (4) Arsitektur Jaringan Komputer dan Komunikasi Data Lapis Jamak, (5) Kriptografi Modern dan Ketahanan Siber (*Cyber Resilience*), (6) Rekayasa Perangkat Lunak Modern dan Manajemen Siklus Hidup (SDLC), (7) *Cloud Computing* dan Infrastruktur *Edge-to-Cloud*.