MENGENAL

ILMU KOMPUTER:

JALAN MENUJU ERA DIGITAL

Buku ini hadir sebagai pengantar komprehensif untuk memahami ilmu komputer, mulai dari konsep dasar, sejarah perkembangannya, hingga peran pentingnya dalam kehidupan modern. Disusun secara sistematis, buku ini membahas berbagai topik penting, seperti pemrograman, logika komputasi, pengembangan aplikasi, rekayasa perangkat lunak, jaringan, sistem operasi, hingga teknologi terkini seperti kecerdasan buatan, cloud computing, internet of things, keamanan siber, dan transformasi digital.

Dengan bahasa yang mudah dipahami dan penyajian yang runtut, buku ini ditujukan bagi mahasiswa, dosen, peneliti, praktisi teknologi informasi, pendidik, maupun masyarakat umum yang ingin memperluas wawasan dan memahami lebih dalam tentang dunia ilmu komputer serta peluangnya di era digital.

Semoga buku ini dapat menjadi referensi yang bermanfaat dan mendukung pengembangan ilmu komputer serta literasi digital menuju masyarakat yang adaptif dan berdaya saing.

PT Andalas
Global Research Center

Jl. Mulia 2, No. 52A, Sariharjo, Ngaglik,
Sleman, Daerah Istimewa Yogyakarta 55581
+62 811 7488 066
globalpublisherofficial@gmail.com
www.andalaspress.com

MENGENAL İLMU KOMPUTER: JALAN MENUJU ERR DIGITAL

PT Andalas Gloalal
Reseerch Center

PT Andalas
Global Research Center

MENGENAL

ILMU KOMPUTER:

JALAN MENUJU ERA DIGITAL

Edy Wihardjo, S.Pd., M.Pd., MCE., MCF.
Akalily Mardhiyya, S.T., M.T.
Hilda Dwi Yunita, S.Kom., M.T.I.
Baharuddin, S.Kom., M.A.P., M.Kom., M.T.A.
Rahmad Surya Hadi Saputra S.St.Pi, MSc



MENGENAL ILMU KOMPUTER: JALAN MENUJU ERA DIGITAL

Edy Wihardjo, S.Pd., M.Pd., MCE., MCF.
Akalily Mardhiyya, S.T., M.T.
Hilda Dwi Yunita, S.Kom., M.T.I.
Baharuddin, S.Kom., M.A.P., M.Kom., M.T.A.
Rahmad Surya Hadi Saputra S.St.Pi, MSc

Penerbit PT Andalas Global Research Center

Judul	: **Mengenal Ilmu Komputer: Jalan Menuju Era Digital**
Penulis	: Edy W., Akalily M., Hilda D.Y., Baharuddin, Rahmad S.H.S.
Editor	: Fitri Arianti Saputri, S.H.
Design Cover	: Eric Krisna Sandi, S.Kom.
Penerbit	: **PT Andalas Global Research Center** Jl. Mulia 2, No. 52A, Sariharjo, Ngaglik, Sleman, Dareah Istimewa Yogyakarta 55581
Telp	: +62 811 7488 066
Email	: globalpublisherofficial@gmail.com
Website	: www.andalaspress.com
ISBN	: xxxx

Anggota Ikatan Penerbit Indonesia (IKAPI)
vii + 106 halaman; 14,8 cm x 21 cm

Hak Cipta Dilindungi Undang-Undang

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apapun tanpa izin tertulis dari penerbit.

Diterbitkan oleh PT Andalas Global Research Center, Yogyakarta, 2026

KATA PENGANTAR

Kata Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan limpahan kesempatan-Nya sehingga buku yang berjudul "Mengenal Ilmu Komputer: Jalan Menuju Era Digital" ini dapat diselesaikan dengan baik. Penyusunan buku ini merupakan salah satu upaya untuk memberikan pemahaman yang komprehensif mengenai ilmu komputer sebagai fondasi utama dalam perkembangan teknologi digital, inovasi, serta transformasi berbagai sektor kehidupan di era modern.

Ilmu komputer bukan lagi sekadar disiplin ilmu yang mempelajari perangkat keras dan perangkat lunak, melainkan telah berkembang menjadi bidang yang berperan penting dalam mendorong kemajuan teknologi, meningkatkan efisiensi kerja, serta mendukung pengambilan keputusan berbasis data. Materi yang disajikan dalam buku ini mencakup konsep dasar ilmu komputer, sejarah perkembangannya, pemrograman dan logika komputasi, pengembangan aplikasi dan rekayasa perangkat lunak, jaringan dan sistem operasi, hingga berbagai tren teknologi modern seperti kecerdasan buatan, cloud computing,

Internet of Things, keamanan siber, dan transformasi digital. Pembahasan disusun secara sistematis agar dapat menjadi referensi bagi mahasiswa, dosen, peneliti, praktisi teknologi informasi, pendidik, maupun masyarakat umum yang ingin memahami perkembangan ilmu komputer secara menyeluruh.

Penulis menyadari bahwa buku ini masih memiliki berbagai keterbatasan, baik dari segi penyajian maupun kedalaman pembahasan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan demi penyempurnaan karya ini pada edisi berikutnya. Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, masukan, dan bantuan selama proses penyusunan buku ini. Semoga buku ini dapat memberikan manfaat, memperluas wawasan pembaca, serta menjadi salah satu referensi yang bermanfaat dalam mendukung pengembangan ilmu komputer, peningkatan literasi digital, dan pemanfaatan teknologi informasi secara inovatif menuju masyarakat yang adaptif dan berdaya saing di era digital.

DAFTAR ISI

KATA PENGANTAR ..... **iii**

DAFTAR ISI ..... **v**

BAB I PENGENALAN ILMU KOMPUTER DAN PERKEMBANGANNYA **1**

1. Konsep Dasar Ilmu Komputer 1
2. Sejarah dan Perkembangan Ilmu Komputer 4
3. Komponen Sistem Komputer (*Hardware, Software, dan Brainware*) 7
4. Bidang-Bidang dalam Ilmu Komputer 9
5. Peran Ilmu Komputer dalam Kehidupan Modern 12

BAB II PEMROGRAMAN DAN LOGIKA KOMPUTASI **17**

1. Konsep Dasar Pemrograman 17
2. Algoritma dan *Flowchart* 20
3. Logika Komputasi dan Penyelesaian Masalah .. 23
4. Struktur Dasar Pemrograman 26
5. Pengenalan Bahasa Pemrograman Modern..... 29

BAB III PENGEMBANGAN APLIKASI DAN REKAYASA PERANGKAT LUNAK **34**

1. Konsep Rekayasa Perangkat Lunak..... 34

2. Siklus Hidup Pengembangan Perangkat Lunak (SDLC) 37
3. Analisis Kebutuhan dan Perancangan Sistem ... 40
4. Implementasi, Pengujian, dan Pemeliharaan Aplikasi 43
5. Metodologi Pengembangan Perangkat Lunak Modern..... 46
BAB IV JARINGAN DAN SISTEM OPERASI **..... 51**
1. Konsep Dasar Sistem Operasi..... 51
2. Fungsi dan Jenis Sistem Operasi..... 54
3. Konsep Dasar Jaringan Komputer 57
4. Topologi, Perangkat, dan Protokol Jaringan 60
5. Keamanan Dasar Sistem dan Jaringan Komputer 63
BAB V TREN DAN TANTANGAN TEKNOLOGI MODERN **..... 67**
1. Transformasi Digital dan Industri 4.0..... 67
2. Kecerdasan Buatan (*Artificial Intelligence*) dan Machine Learning..... 72
3. Internet of Things (IoT), Cloud Computing, dan Big Data..... 76
4. Keamanan Siber dan Etika Teknologi Informasi 80

5. Prospek Karier dan Tantangan Masa Depan di Bidang Ilmu Komputer 84

DAFTAR PUSTAKA 88

BAB I
PENGENALAN ILMU KOMPUTER DAN PERKEMBANGANNYA

Ilmu komputer merupakan salah satu disiplin ilmu yang paling berpengaruh dalam membentuk peradaban manusia pada abad ini. Bab ini membahas pengenalan ilmu komputer secara menyeluruh, mulai dari konsep dasar, sejarah dan perkembangannya, komponen sistem komputer, keluasan bidang kajian, hingga peranannya dalam kehidupan modern. Secara umum, ilmu komputer dipahami sebagai disiplin yang memadukan teori, eksperimen, dan rekayasa untuk merancang, mengembangkan, dan menganalisis sistem komputasi (Rajkumar dkk. 2024; Kumar.B 2025).

1. Konsep Dasar Ilmu Komputer

Ilmu komputer pada tingkat dasar mencakup algoritma, struktur data, logika, arsitektur komputer, bahasa pemrograman, perangkat lunak, dan teori komputasi sebagai fondasi utama pembelajarannya. Konsep-konsep tersebut saling berkaitan dan membentuk kerangka berpikir sistematis yang digunakan untuk menyelesaikan persoalan komputasi secara efisien dan terstruktur (Rajkumar dkk. 2024; Kumar.B 2025; Kravchuk, Synyuk, dan Kravchuk 2025).

Algoritma dapat dipahami sebagai urutan langkah logis yang disusun untuk menyelesaikan suatu persoalan secara sistematis, sedangkan struktur data merupakan cara pengorganisasian data agar dapat diakses dan diolah secara efisien oleh komputer. Kedua unsur tersebut menjadi dasar bagi hampir seluruh cabang keilmuan

komputer, mulai dari pengembangan perangkat lunak sederhana hingga sistem kecerdasan buatan yang kompleks (Kravchuk, Synyuk, dan Kravchuk 2025).

Logika komputasi mengajarkan cara berpikir yang runtut dan dapat diverifikasi kebenarannya, sehingga setiap solusi yang dirancang dapat diuji dan dipertanggungjawabkan secara ilmiah. Arsitektur komputer, di sisi lain, menjelaskan bagaimana komponen-komponen perangkat keras disusun dan bekerja sama untuk menjalankan instruksi yang diberikan oleh perangkat lunak (Kumar.B 2025; Rajkumar dkk. 2024).

Bahasa pemrograman menjadi jembatan yang menghubungkan gagasan manusia dengan instruksi yang dapat dipahami dan dijalankan oleh mesin. Melalui bahasa pemrograman, algoritma yang telah dirancang dapat diterjemahkan menjadi program yang benar-benar dapat dieksekusi oleh sistem komputer, sehingga pemahaman terhadap konsep ini menjadi keterampilan dasar yang wajib dikuasai oleh setiap pembelajar ilmu komputer (Rajkumar dkk. 2024).

Teori komputasi menyediakan landasan matematis untuk memahami batas kemampuan dan efisiensi suatu algoritma maupun sistem komputasi secara umum. Pemahaman terhadap teori ini membantu mahasiswa dan praktisi ilmu komputer untuk menilai apakah suatu persoalan dapat diselesaikan secara komputasional, serta seberapa efisien penyelesaian tersebut dapat dilakukan (Kravchuk, Synyuk, dan Kravchuk 2025).

Secara keilmuan, ilmu komputer tidak semata berkutat pada aspek teknis pemrograman, tetapi juga melibatkan pendekatan eksperimen dan rekayasa yang menyeluruh. Pendekatan eksperimen digunakan untuk

menguji hipotesis mengenai kinerja suatu sistem, sedangkan pendekatan rekayasa digunakan untuk merancang solusi yang dapat diterapkan secara nyata pada berbagai persoalan di dunia industri maupun masyarakat (A. Kumar dan Raj 2024; Rajkumar dkk. 2024; Kravchuk, Synyuk, dan Kravchuk 2025).

Pemahaman konsep dasar tersebut juga membekali pembelajar dengan kemampuan berpikir komputasional, yaitu kemampuan menguraikan persoalan besar menjadi bagian-bagian kecil yang lebih mudah dipecahkan, mengenali pola yang berulang, serta merancang solusi umum yang dapat diterapkan pada berbagai variasi persoalan serupa. Kemampuan berpikir komputasional tersebut kini dipandang sebagai keterampilan dasar yang relevan tidak hanya bagi pembelajar ilmu komputer, tetapi juga bagi berbagai disiplin ilmu lain yang memanfaatkan teknologi digital (Kravchuk, Synyuk, dan Kravchuk 2025).

Selain unsur teknis, konsep dasar ilmu komputer juga mencakup pemahaman mengenai representasi data dalam bentuk biner, yaitu sistem angka berbasis dua digit yang menjadi dasar seluruh pemrosesan informasi pada komputer modern. Pemahaman terhadap representasi biner ini penting untuk menjelaskan bagaimana teks, gambar, suara, dan berbagai jenis data lain pada akhirnya dapat diolah oleh mesin yang secara fisik hanya mengenali dua kondisi, yaitu hidup dan mati (Rajkumar dkk. 2024).

Pengenalan terhadap sistem operasi juga menjadi bagian penting dari konsep dasar ilmu komputer, mengingat sistem operasi berperan sebagai perantara antara perangkat keras dengan perangkat lunak aplikasi yang digunakan oleh pengguna sehari-hari. Tanpa sistem

operasi, pengguna akan kesulitan berinteraksi secara langsung dengan perangkat keras komputer yang bekerja pada tingkat instruksi mesin yang sangat rumit (A. Kumar dan Raj 2024).

2. Sejarah dan Perkembangan Ilmu Komputer

Akar historis ilmu komputer dapat ditelusuri hingga gagasan mekanis pada abad ke-19, seperti mesin analitis atau Analytical Engine yang dirancang oleh Charles Babbage, serta algoritma yang disusun oleh Ada Lovelace yang kerap disebut sebagai program komputer pertama dalam sejarah. Gagasan-gagasan awal tersebut kemudian dipertegas secara teoretis oleh model komputasi yang dirumuskan oleh Alan Turing pada dekade 1930-an (Kumar.B 2025; Rajkumar dkk. 2024).

Model komputasi Turing memberikan dasar teoretis mengenai apa yang dapat dan tidak dapat dihitung oleh sebuah mesin, sehingga menjadi salah satu tonggak paling penting dalam sejarah perkembangan ilmu komputer sebagai disiplin ilmu yang mandiri. Pemikiran tersebut menjadi fondasi bagi perkembangan komputer elektronik pada masa berikutnya (Taha dkk. 2022).

Tonggak era digital modern umumnya ditautkan pada kemunculan komputer elektronik pertama pada pertengahan abad ke-20, seperti Electronic Numerical Integrator and Computer atau ENIAC pada tahun 1946, yang kemudian diikuti oleh era transistor, sirkuit terpadu, mikroprosesor, komputer pribadi, jaringan internet, hingga komputasi awan (S. Chen 2024; Rajkumar dkk. 2024).

Perkembangan dari tabung hampa menuju transistor menandai lompatan besar dalam hal ukuran, kecepatan, dan efisiensi energi perangkat komputasi.

Penemuan sirkuit terpadu selanjutnya memungkinkan penggabungan ribuan komponen elektronik dalam satu keping kecil, yang pada akhirnya melahirkan mikroprosesor dan membuka jalan bagi lahirnya komputer pribadi yang dapat dijangkau oleh masyarakat luas (Chen, 2024).

Kehadiran jaringan internet pada akhir abad ke-20 mengubah cara komputer digunakan, dari sekadar alat penghitung dan pengolah data menjadi sarana komunikasi dan pertukaran informasi global. Perkembangan ini kemudian berlanjut dengan munculnya komputasi awan yang memungkinkan penyimpanan dan pengolahan data dilakukan secara jarak jauh melalui pusat data yang tersebar di berbagai lokasi (Rajkumar dkk. 2024).

Literatur terbaru menekankan bahwa evolusi ilmu komputer terus bergerak dari sistem tunggal yang tersentralisasi menuju model yang lebih terdistribusi, terspesialisasi, dan saling terhubung, seperti komputasi awan, komputasi tepi atau edge computing, komputasi kabut atau fog computing, internet untuk segala atau internet of things, hingga komputasi kuantum (Gill dkk. 2024; Lindsay dkk. 2021).

Perkembangan mutakhir ilmu komputer tidak lagi hanya berfokus pada peningkatan kecepatan mesin yang sudah ada, melainkan mengarah pada komputasi khusus dan paradigma yang benar-benar baru, seperti komputasi neuromorfik dan komputasi kuantum yang menjanjikan kemampuan pemrosesan yang jauh melampaui arsitektur komputer konvensional (Gill dkk. 2024; Taha dkk. 2022).

Sejarah ilmu komputer secara keseluruhan menunjukkan pergeseran bertahap dari mesin mekanis

sederhana menuju sistem digital yang semakin cerdas dan otonom. Pemahaman terhadap perjalanan sejarah ini penting bagi pembelajar ilmu komputer, karena membantu menempatkan setiap teknologi baru dalam konteks perkembangan yang berkesinambungan, bukan sebagai penemuan yang berdiri sendiri tanpa keterkaitan dengan masa lalu (Kumar.B 2025).

Sistem komputasi terdistribusi yang berkembang pada dekade terakhir menunjukkan bahwa arah masa depan ilmu komputer tidak lagi bertumpu pada satu mesin tunggal yang bekerja sendiri, melainkan pada jejaring perangkat yang saling terhubung dan berbagi beban komputasi secara bersama-sama. Pergeseran paradigma tersebut membawa tantangan baru dalam hal koordinasi, keamanan, dan efisiensi energi yang perlu terus dikaji oleh para peneliti ilmu komputer di masa mendatang (Lindsay dkk. 2021).

Generasi komputer juga sering dikelompokkan berdasarkan teknologi komponen utamanya, mulai dari generasi pertama yang menggunakan tabung hampa udara, generasi kedua yang menggunakan transistor, generasi ketiga yang menggunakan sirkuit terpadu, hingga generasi keempat yang menggunakan mikroprosesor sebagai inti pemrosesan. Pengelompokan tersebut membantu memahami bagaimana setiap lompatan teknologi membawa peningkatan signifikan dalam hal kecepatan, ukuran, dan efisiensi perangkat komputer (S. Chen 2024).

Perkembangan komputer pribadi pada dekade 1970 hingga 1980-an turut mengubah persepsi masyarakat terhadap komputer, dari sekadar mesin raksasa yang hanya dimiliki institusi besar menjadi perangkat yang dapat dimiliki dan digunakan oleh

individu di rumah maupun tempat kerja. Perubahan ini menjadi salah satu pendorong utama meluasnya literasi komputer di kalangan masyarakat umum pada masa berikutnya (Rajkumar dkk. 2024).

3. Komponen Sistem Komputer (*Hardware, Software, dan Brainware*)

Sistem komputer secara konsisten dijelaskan sebagai gabungan tiga komponen utama, yaitu perangkat keras atau hardware, perangkat lunak atau software, dan manusia pengguna atau brainware, dengan beberapa literatur sistem informasi turut menambahkan basis data sebagai komponen pendukung yang penting (Nugroho dan Ali 2022).

Perangkat keras merujuk pada komponen fisik dan infrastruktur perangkat komputer, seperti unit pemroses pusat, memori, media penyimpanan, serta perangkat masukan dan keluaran. Perangkat lunak merujuk pada sistem digital, algoritma, dan aplikasi yang menjalankan instruksi pada perangkat keras, sedangkan manusia pengguna adalah pihak yang mengoperasikan, memelihara, dan menggunakan sistem tersebut secara langsung (Berawi, Sari, dan Miraj 2024).

Bukti paling jelas mengenai pentingnya keterpaduan ketiga komponen tersebut ditemukan pada studi mengenai kota cerdas atau smart city, yang menunjukkan bahwa perangkat keras dan perangkat lunak tidak akan memberikan manfaat optimal tanpa dukungan manusia pengguna yang memiliki literasi teknologi, kemampuan analitis, dan partisipasi aktif dalam memanfaatkan sistem tersebut (Berawi, Sari, dan Miraj 2024).

Oleh karena itu, pengenalan sistem komputer modern tidak dapat dipahami hanya dari aspek teknis

semata, melainkan juga perlu menekankan peran manusia dalam pengambilan keputusan, adaptasi terhadap perubahan teknologi, serta penggunaan teknologi yang inklusif bagi seluruh lapisan masyarakat (Kravchuk, Synyuk, dan Kravchuk 2025).

Kajian mengenai sistem informasi manajemen turut menegaskan bahwa keberhasilan suatu sistem informasi organisasi sangat bergantung pada keterpaduan antara perangkat lunak, basis data, dan manusia pengguna. Basis data berperan penting sebagai tempat penyimpanan data yang terstruktur, sehingga informasi dapat diolah dan disajikan secara akurat kepada pengambil keputusan (Ramadhona, Erliyani, dan Rosdiana 2023).

Interaksi antara ketiga komponen utama sistem komputer tersebut bersifat saling bergantung dan tidak dapat dipisahkan satu sama lain. Perangkat keras yang canggih tanpa perangkat lunak yang sesuai tidak akan dapat berfungsi, perangkat lunak tanpa perangkat keras yang memadai tidak dapat dijalankan, dan keduanya tanpa manusia pengguna yang kompeten tidak akan memberikan manfaat nyata bagi penyelesaian persoalan sehari-hari (Nugroho dan Ali 2022).

Perkembangan teknologi pendidikan turut menunjukkan bahwa keterpaduan ketiga komponen sistem komputer tersebut memengaruhi dinamika ruang kelas modern, di mana perangkat keras berupa komputer atau perangkat mobile, perangkat lunak berupa aplikasi pembelajaran, dan manusia pengguna berupa pengajar maupun peserta didik saling berinteraksi untuk menciptakan pengalaman belajar yang lebih interaktif dan personal (Kusnadi dkk. 2025).

Perangkat keras komputer modern umumnya terdiri atas beberapa unit fungsional utama, yaitu unit pemroses pusat yang bertugas menjalankan instruksi program, unit memori yang menyimpan data dan instruksi sementara selama proses berlangsung, unit penyimpanan yang menyimpan data secara permanen, serta unit masukan dan keluaran yang menjadi jembatan interaksi antara komputer dengan pengguna maupun perangkat lain (Nugroho dan Ali 2022).

Perangkat lunak dapat dikelompokkan menjadi dua kategori besar, yaitu perangkat lunak sistem yang mengatur operasional dasar komputer, seperti sistem operasi, serta perangkat lunak aplikasi yang dirancang untuk membantu pengguna menyelesaikan tugas tertentu, seperti pengolah kata, peramban web, maupun aplikasi perkantoran lainnya. Kedua kategori perangkat lunak tersebut bekerja secara berlapis untuk memastikan pengguna dapat memanfaatkan perangkat keras secara optimal (Ramadhona, Erliyani, dan Rosdiana 2023).

4. Bidang-Bidang dalam Ilmu Komputer

Bidang kajian ilmu komputer kini sangat luas, mencakup rekayasa perangkat lunak, kecerdasan buatan, keamanan siber, komputasi awan, teknologi rantai blok atau blockchain, internet untuk segala, sains data, pengolahan bahasa alami atau natural language processing, visi komputer atau computer vision, hingga komputasi kuantum (Sharjeel dkk. 2025).

Kecerdasan buatan merupakan salah satu bidang yang berkembang paling pesat dalam ilmu komputer, dengan aplikasi yang mencakup pengenalan pola, pengambilan keputusan otomatis, hingga pemrosesan

bahasa alami yang memungkinkan komputer memahami dan menghasilkan bahasa manusia secara lebih alami (Khaleel, Jebrel, dan Shwehdy 2024).

Keamanan siber menjadi bidang yang semakin penting seiring meningkatnya ketergantungan masyarakat terhadap sistem digital. Bidang ini berfokus pada perlindungan data dan sistem dari ancaman peretasan, pencurian data, maupun serangan siber lain yang dapat merugikan individu maupun organisasi secara luas (Sharjeel dkk. 2025).

Komputasi awan memungkinkan penyimpanan dan pemrosesan data dilakukan secara terpusat pada pusat data yang dapat diakses dari mana saja melalui jaringan internet, sehingga mengurangi kebutuhan akan infrastruktur perangkat keras yang mahal bagi individu maupun organisasi kecil. Teknologi rantai blok, di sisi lain, menawarkan mekanisme pencatatan data yang terdesentralisasi dan sulit dimanipulasi, sehingga banyak dimanfaatkan untuk keperluan transaksi digital yang aman (Sharjeel dkk. 2025).

Internet untuk segala menghubungkan berbagai perangkat fisik ke dalam jaringan digital, memungkinkan pertukaran data secara otomatis antarperangkat tanpa memerlukan campur tangan manusia secara langsung. Sains data, di sisi lain, berfokus pada pengolahan dan penafsiran data dalam jumlah besar untuk menghasilkan wawasan yang bermanfaat bagi pengambilan keputusan (Rajkumar dkk. 2024).

Bidang-bidang tersebut kini semakin saling terhubung satu sama lain, misalnya kecerdasan buatan yang dimanfaatkan untuk memperkuat sistem keamanan siber, teknologi rantai blok yang digunakan untuk mengamankan jaringan internet untuk segala, serta

komputasi kuantum yang dipandang sebagai peluang sekaligus ancaman terhadap sistem keamanan kriptografi yang berlaku saat ini (Sharjeel dkk. 2025; Lindsay dkk. 2021).

Keluasan bidang ilmu komputer tersebut menegaskan bahwa disiplin ini tidak lagi dapat dipandang sebagai bidang tunggal yang seragam, melainkan sebagai ekosistem keilmuan yang terus berkembang dan saling memengaruhi. Pemahaman terhadap keterkaitan antarbidang tersebut menjadi bekal penting bagi pembelajar ilmu komputer untuk dapat beradaptasi dengan perkembangan teknologi yang terus berubah dari waktu ke waktu (Kumar.B 2025).

Rekayasa perangkat lunak sebagai salah satu bidang inti ilmu komputer menekankan pentingnya proses pengembangan perangkat lunak yang sistematis, mulai dari analisis kebutuhan, perancangan, pengujian, hingga pemeliharaan sistem setelah digunakan secara luas. Penguasaan terhadap prinsip rekayasa perangkat lunak menjadi bekal penting bagi lulusan ilmu komputer yang ingin berkarier dalam pengembangan aplikasi maupun sistem berskala besar (Dontulwar 2024).

Bidang visi komputer memungkinkan mesin untuk mengenali dan menafsirkan informasi visual, seperti gambar dan video, sehingga dapat dimanfaatkan pada berbagai aplikasi, mulai dari pengenalan wajah, deteksi objek pada kendaraan otonom, hingga analisis citra medis untuk membantu diagnosis penyakit. Bidang komputasi kuantum, di sisi lain, masih berada pada tahap penelitian lanjutan, tetapi menjanjikan kemampuan pemrosesan yang jauh melampaui komputer konvensional untuk persoalan tertentu (Sharjeel dkk. 2025).

5. Peran Ilmu Komputer dalam Kehidupan Modern

Peran ilmu komputer dalam kehidupan modern sangat luas dan menjangkau hampir seluruh aspek kehidupan manusia, mulai dari pendidikan, kesehatan, bisnis, transportasi, hiburan, riset ilmiah, hingga tata kelola perkotaan (Dontulwar 2024; Shon 2024; Lidjali dkk. 2025; Rajkumar dkk. 2024).

Di bidang pendidikan, teknologi informasi dan ilmu komputer memperluas akses terhadap pembelajaran, memungkinkan personalisasi proses belajar sesuai kebutuhan masing-masing peserta didik, serta memfasilitasi verifikasi kredensial pendidikan secara digital yang lebih aman dan efisien (Kusnadi dkk. 2025).

Di bidang kesehatan, pemanfaatan sistem komputer memungkinkan pencatatan rekam medis secara digital, analisis data pasien untuk mendukung diagnosis yang lebih akurat, serta pengembangan teknologi pencitraan medis yang semakin canggih. Kemajuan tersebut turut berkontribusi terhadap peningkatan kualitas layanan kesehatan secara menyeluruh (Shon 2024).

Dalam dunia bisnis, ilmu komputer mendukung otomasi proses kerja, analitik data untuk pengambilan keputusan strategis, serta pengembangan platform perdagangan elektronik yang memperluas jangkauan pasar bagi pelaku usaha. Sektor transportasi juga merasakan manfaat nyata melalui sistem navigasi digital, manajemen lalu lintas berbasis data, serta pengembangan kendaraan otonom yang semakin mendekati kenyataan (Dontulwar 2024).

Di masyarakat modern secara umum, komputer telah menjadi alat esensial yang mengubah cara manusia bekerja, berkomunikasi, dan memecahkan masalah

sehari-hari. Perubahan tersebut tampak nyata pada meluasnya penggunaan komunikasi digital, media sosial, serta berbagai aplikasi yang mempermudah aktivitas rutin masyarakat (Dontulwar 2024; S. Chen 2024).

Meskipun memberikan banyak manfaat, literatur juga secara konsisten mencatat berbagai tantangan yang menyertai pemanfaatan ilmu komputer dalam kehidupan modern, seperti persoalan etika penggunaan teknologi, ancaman terhadap privasi data pribadi, kesenjangan digital antarkelompok masyarakat, kerentanan keamanan siber, serta potensi disrupsi terhadap lapangan kerja akibat otomatisasi (Dontulwar 2024; Lindsay dkk. 2021; Lidjali dkk. 2025; Shon 2024).

Kesenjangan digital menjadi tantangan yang perlu mendapat perhatian khusus, mengingat tidak seluruh lapisan masyarakat memiliki akses yang setara terhadap infrastruktur maupun literasi teknologi. Kondisi ini berpotensi memperlebar jurang ketimpangan sosial apabila tidak diiringi dengan kebijakan yang mendorong pemerataan akses teknologi bagi seluruh kalangan (Lidjali dkk. 2025).

Arah perkembangan ilmu komputer di masa mendatang diperkirakan akan semakin menonjolkan kecerdasan buatan, internet untuk segala, komputasi awan, komputasi kuantum, bahkan nano-komputasi, yang kesemuanya perlu diimbangi dengan kesadaran etika dan upaya memastikan akses yang adil bagi seluruh masyarakat (Dontulwar 2024; Rajkumar dkk. 2024; Taha dkk. 2022; A. Kumar dan Raj 2024).

Kurikulum ilmu komputer tingkat internasional juga terus diperbarui untuk menyesuaikan diri dengan perkembangan teknologi tersebut, dengan menekankan pentingnya kompetensi teknis yang mendalam sekaligus

kesadaran etis dan sosial bagi setiap lulusan ilmu komputer, agar mampu berkontribusi secara bertanggung jawab terhadap perkembangan teknologi di masa depan (A. Kumar dan Raj 2024).

Riset mengenai peran ilmu informasi dalam perkembangan sains dan teknologi turut menegaskan bahwa ilmu komputer tidak berdiri sendiri, melainkan berkelindan dengan berbagai disiplin ilmu lain, sehingga kontribusinya terhadap kemajuan peradaban manusia bersifat lintas sektor dan saling menguatkan (Lidjali dkk. 2025).

Perkembangan nanoteknologi yang berpadu dengan ilmu komputer turut membuka peluang baru bagi terciptanya perangkat komputasi berukuran sangat kecil namun memiliki kemampuan pemrosesan yang signifikan, sehingga berpotensi mengubah cara manusia merancang dan menggunakan perangkat elektronik di masa mendatang (Taha dkk. 2022).

Peran ilmu komputer dalam riset ilmiah juga tidak dapat diabaikan, mengingat hampir seluruh bidang keilmuan saat ini bergantung pada kemampuan komputasi untuk mengolah data penelitian dalam jumlah besar, menjalankan simulasi kompleks, serta mempercepat proses analisis yang sebelumnya membutuhkan waktu jauh lebih lama apabila dilakukan secara manual (S. Chen 2024).

Di bidang hiburan, ilmu komputer mendorong lahirnya industri permainan digital, animasi komputer, serta layanan pemutaran konten hiburan berbasis internet yang dapat diakses kapan saja oleh pengguna. Perkembangan tersebut turut mengubah pola konsumsi hiburan masyarakat modern, dari media konvensional

menuju platform digital yang lebih personal dan interaktif (Gill dkk. 2024).

Pada tata kelola perkotaan, penerapan ilmu komputer memungkinkan pemantauan lalu lintas, pengelolaan sumber daya energi, serta pelayanan publik yang lebih responsif melalui pemanfaatan data secara real-time. Konsep kota cerdas yang memadukan sensor, jaringan komunikasi, dan analisis data tersebut menjadi salah satu wujud nyata bagaimana ilmu komputer dapat berkontribusi terhadap peningkatan kualitas hidup masyarakat perkotaan secara menyeluruh (Kusnadi dkk. 2025).

Pemanfaatan ilmu komputer pada sektor pemerintahan turut memperluas layanan publik berbasis digital, mulai dari perizinan daring, pelaporan pengaduan masyarakat secara elektronik, hingga transparansi anggaran yang dapat diakses publik secara terbuka. Transformasi layanan pemerintahan tersebut menuntut kesiapan infrastruktur teknologi sekaligus kesiapan aparatur dalam mengoperasikan sistem digital secara bertanggung jawab (Dontulwar 2024).

Secara keseluruhan, keluasan peran ilmu komputer pada berbagai sektor kehidupan tersebut menegaskan posisinya sebagai disiplin ilmu yang bersifat lintas sektor, tidak terbatas pada bidang teknologi informasi semata, melainkan menjadi penopang bagi hampir seluruh aktivitas manusia modern, baik dalam skala individu, organisasi, maupun negara (Rajkumar dkk. 2024; Shon 2024).

Pemahaman menyeluruh terhadap konsep dasar, sejarah, komponen sistem, keluasan bidang, serta peran ilmu komputer dalam kehidupan modern yang telah diuraikan pada bab ini menjadi fondasi penting sebelum

mempelajari topik-topik ilmu komputer yang lebih spesifik dan mendalam pada bab-bab selanjutnya.

BAB II
PEMROGRAMAN DAN LOGIKA KOMPUTASI

Pemrograman merupakan keterampilan inti dalam ilmu komputer yang menghubungkan cara berpikir logis manusia dengan kemampuan mesin untuk menjalankan instruksi secara sistematis. Bab ini membahas perjalanan pembelajaran pemrograman secara bertahap, mulai dari konsep dasar, algoritma dan diagram alir atau flowchart, logika komputasi dan penyelesaian masalah, struktur dasar pemrograman, hingga pengenalan bahasa pemrograman modern. Berbagai kajian menunjukkan secara konsisten bahwa pemrograman memadukan logika penyelesaian masalah dengan kemampuan menerjemahkan solusi tersebut ke dalam instruksi formal yang dapat dijalankan oleh komputer (Belmar 2022; Anggreani dan Yahfizham 2024).

1. Konsep Dasar Pemrograman

Program dapat dipahami sebagai kumpulan instruksi terkode yang ditulis dalam bahasa pemrograman formal untuk menyelesaikan suatu persoalan tertentu, sedangkan algoritma menjadi fondasi logika sistematis yang mendasari penyusunan program tersebut. Pemahaman terhadap perbedaan sekaligus keterkaitan antara algoritma dan program menjadi langkah awal yang penting sebelum mempelajari pemrograman secara lebih mendalam (Anggreani dan Yahfizham 2024; Tarsini dan Anggraeni 2024).

Konsep dasar pemrograman tidak dapat dipisahkan dari pemahaman mengenai variabel, yaitu tempat penyimpanan data yang nilainya dapat berubah selama proses eksekusi program berlangsung, serta tipe data, yang menentukan jenis nilai yang dapat disimpan oleh suatu variabel, seperti bilangan bulat, bilangan pecahan, teks, maupun nilai benar atau salah. Kedua konsep tersebut menjadi bekal dasar yang wajib dikuasai sebelum mempelajari struktur pemrograman yang lebih kompleks (Anggreani dan Yahfizham 2024).

Pembelajar pemula juga perlu memahami konsep masukan dan keluaran, yaitu cara program menerima data dari pengguna dan menyajikan hasil pengolahan data tersebut kembali kepada pengguna. Pemahaman terhadap alur masukan dan keluaran ini membantu pembelajar melihat program sebagai suatu proses yang utuh, bukan sekadar kumpulan baris kode yang berdiri sendiri-sendiri (Anggreani dan Yahfizham 2024).

Secara historis, istilah algoritma berasal dari kata algorism yang dapat ditelusuri kembali kepada nama seorang ilmuwan Muslim, yaitu Abu Ja'far Muhammad Ibn Musa Al-Khuwarizmi, yang karyanya menjadi dasar bagi perkembangan metode penyelesaian masalah secara sistematis dalam matematika maupun ilmu komputer. Penelusuran sejarah tersebut menunjukkan bahwa gagasan mengenai langkah penyelesaian masalah yang runtut telah ada jauh sebelum komputer modern diciptakan (Al-Hafiz dkk. 2024).

Pemahaman konsep dasar pemrograman juga mencakup kesadaran bahwa komputer hanya dapat menjalankan instruksi yang diberikan secara tepat dan tidak ambigu. Oleh karena itu, pembelajar pemula perlu dilatih untuk berpikir secara presisi dan detail, karena

kesalahan kecil dalam penulisan instruksi dapat menyebabkan program berjalan tidak sesuai harapan atau bahkan gagal dijalankan sama sekali (Anggreani dan Yahfizham 2024).

Pengenalan konsep dasar algoritma pemrograman dalam konteks kehidupan sehari-hari juga terbukti membantu pembelajar memahami bahwa logika algoritmik sesungguhnya telah diterapkan secara tidak sadar dalam berbagai aktivitas rutin, seperti memasak berdasarkan resep atau menyusun rencana perjalanan. Pendekatan kontekstual semacam ini mempermudah transisi pembelajar dari pemahaman intuitif menuju pemahaman formal mengenai algoritma pemrograman (Al-Hafiz dkk. 2024).

Pembelajar pemula juga perlu memahami perbedaan antara kesalahan sintaks atau syntax error dan kesalahan logika atau logical error yang kerap muncul selama proses penyusunan program. Kesalahan sintaks terjadi ketika penulisan kode tidak sesuai dengan aturan tata bahasa yang berlaku pada bahasa pemrograman tertentu, sedangkan kesalahan logika terjadi ketika kode dapat dijalankan tanpa gangguan teknis, tetapi menghasilkan keluaran yang tidak sesuai dengan yang diharapkan. Pemahaman terhadap kedua jenis kesalahan tersebut membantu pembelajar menelusuri sumber masalah secara lebih terarah ketika program yang disusun tidak berjalan sebagaimana mestinya (Anggreani dan Yahfizham 2024).

Konsep dasar pemrograman juga menekankan pentingnya kebiasaan menuliskan kode secara rapi dan terstruktur sejak tahap awal pembelajaran, termasuk penggunaan penamaan variabel yang jelas dan pemberian komentar pada bagian kode yang penting.

Kebiasaan tersebut tidak hanya memudahkan pembelajar dalam memahami kembali kode yang telah ditulis sebelumnya, tetapi juga menjadi bekal penting ketika kelak bekerja secara kolaboratif dalam tim pengembangan perangkat lunak yang lebih besar (Tarsini dan Anggraeni 2024).

Selain aspek teknis, pembelajaran konsep dasar pemrograman juga sebaiknya menumbuhkan kesadaran mengenai pentingnya pengujian program secara menyeluruh sebelum program tersebut dianggap selesai. Kebiasaan menguji program dengan berbagai kemungkinan masukan, termasuk masukan yang tidak lazim atau berada di luar dugaan, membantu pembelajar menemukan kelemahan program sejak dini, sebelum program tersebut digunakan secara luas oleh pengguna lain (Ngadengon dkk. 2025).

2. Algoritma dan *Flowchart*

Algoritma yang baik harus memenuhi lima karakteristik utama, yaitu keterbatasan atau finiteness yang berarti algoritma harus berhenti setelah sejumlah langkah tertentu, kepastian atau definiteness yang berarti setiap langkah harus terdefinisi secara jelas dan tidak ambigu, adanya masukan atau input, adanya keluaran atau output, serta efektivitas atau effectiveness yang berarti setiap langkah harus dapat dilaksanakan secara nyata (Tarsini dan Anggraeni 2024).

Kelima karakteristik tersebut menjadi tolok ukur untuk menilai apakah suatu rangkaian langkah dapat dikategorikan sebagai algoritma yang valid. Algoritma yang tidak memenuhi salah satu karakteristik tersebut, misalnya algoritma yang tidak pernah berhenti atau tidak memiliki keluaran yang jelas, tidak dapat dianggap

sebagai algoritma yang baik untuk diterapkan dalam pemrograman (Balreira, Silveira, dan Wickboldt 2022).

Notasi algoritma dapat dituliskan melalui tiga metode utama, yaitu kalimat deskriptif yang menjelaskan langkah penyelesaian masalah menggunakan bahasa sehari-hari, kode semu atau pseudocode yang menuliskan langkah algoritma menyerupai struktur bahasa pemrograman tanpa terikat pada sintaks bahasa tertentu, serta diagram alir atau flowchart yang menggunakan simbol visual untuk menggambarkan alur logika program (Tarsini dan Anggraeni 2024; Al-Hafiz dkk. 2024).

Diagram alir menggunakan simbol-simbol standar, seperti bentuk oval untuk menandai titik awal dan akhir program, bentuk persegi panjang untuk menandai proses, bentuk jajaran genjang untuk menandai masukan dan keluaran, serta bentuk belah ketupat untuk menandai pengambilan keputusan. Penggunaan simbol visual tersebut sangat membantu pembelajar pemula untuk memahami alur program secara menyeluruh sebelum dihadapkan pada sintaks bahasa pemrograman yang lebih kompleks (Tarsini dan Anggraeni 2024; Al-Hafiz dkk. 2024; Cook 2025).

Penggunaan diagram alir dalam pengajaran pemrograman pengantar terbukti membantu pembelajar mengembangkan pemahaman konseptual mengenai logika program sebelum mereka dihadapkan pada kerumitan sintaks bahasa pemrograman tertentu. Pendekatan berbasis diagram alir memungkinkan pembelajar berfokus terlebih dahulu pada logika penyelesaian masalah, sebelum kemudian menerjemahkannya ke dalam kode program yang sesungguhnya (Cook 2025).

Kode semu juga memiliki peran penting sebagai jembatan antara bahasa manusia dengan bahasa pemrograman formal. Melalui kode semu, pembelajar dapat menuliskan logika algoritma secara terstruktur tanpa perlu khawatir mengenai kesalahan sintaks, sehingga dapat berfokus sepenuhnya pada ketepatan logika penyelesaian masalah yang sedang dirancang (Tarsini dan Anggraeni 2024).

Ketiga metode notasi algoritma tersebut, yaitu kalimat deskriptif, kode semu, dan diagram alir, pada dasarnya saling melengkapi dan dapat digunakan secara bergantian sesuai dengan kebutuhan serta tingkat pemahaman pembelajar. Penguasaan terhadap ketiga metode ini menjadi bekal penting sebelum pembelajar melangkah ke tahap penulisan kode program yang sesungguhnya (Cook 2025).

Kajian mengenai eksplorasi konsep diagram alir dan kode semu dalam pembelajaran algoritma menunjukkan bahwa kombinasi kedua metode tersebut memberikan hasil belajar yang lebih baik dibandingkan penggunaan salah satu metode saja. Diagram alir memberikan gambaran visual yang mudah dipahami secara sekilas, sedangkan kode semu memberikan gambaran yang lebih dekat dengan struktur bahasa pemrograman yang sesungguhnya, sehingga keduanya saling menguatkan pemahaman pembelajar dari sudut pandang yang berbeda (Tarsini dan Anggraeni 2024).

Penyusunan diagram alir yang baik juga menuntut pembelajar untuk berlatih menyederhanakan persoalan yang kompleks menjadi rangkaian langkah yang jelas dan mudah diikuti. Latihan ini secara tidak langsung melatih kemampuan berpikir sistematis pembelajar, yang kelak akan sangat bermanfaat ketika mereka dihadapkan pada

persoalan pemrograman yang jauh lebih rumit dan melibatkan banyak tahapan proses sekaligus (Al-Hafiz dkk. 2024).

Latihan menerjemahkan diagram alir menjadi kode program yang sesungguhnya juga menjadi tahapan penting dalam proses pembelajaran, karena tahapan ini melatih pembelajar untuk memahami kesesuaian antara representasi visual dengan sintaks bahasa pemrograman yang digunakan. Kemampuan menerjemahkan diagram alir menjadi kode program secara akurat menjadi indikator penting bahwa pembelajar telah benar-benar memahami logika algoritma yang dirancang, bukan sekadar menghafal bentuk visual diagram alir semata (Cook 2025).

3. Logika Komputasi dan Penyelesaian Masalah

Logika komputasi atau computational thinking berfungsi sebagai perancah alami atau natural scaffold untuk mengembangkan keterampilan penyelesaian masalah melalui pemrograman. Pendekatan ini membantu pembelajar mengubah cara pandang terhadap suatu persoalan, dari sekadar mencari jawaban tunggal menjadi merancang proses penyelesaian yang sistematis dan dapat diulang (Belmar 2022).

Tahapan logika komputasi yang paling sering digunakan mencakup empat unsur utama, yaitu dekomposisi atau decomposition yang berarti memecah persoalan besar menjadi bagian-bagian kecil yang lebih mudah dipahami, pengenalan pola atau pattern recognition yang berarti mengenali kesamaan atau keteraturan pada persoalan yang dihadapi, abstraksi atau abstraction yang berarti menyaring informasi penting dan mengabaikan detail yang tidak relevan, serta

penyusunan algoritma sebagai langkah penyelesaian yang sistematis (Wu dkk. 2023).

Dekomposisi memungkinkan pembelajar menghadapi persoalan kompleks dengan cara yang lebih terkendali, karena setiap bagian kecil dari persoalan dapat diselesaikan secara terpisah sebelum digabungkan kembali menjadi solusi yang utuh. Pengenalan pola, di sisi lain, membantu pembelajar mengenali kesamaan antara persoalan baru dengan persoalan yang pernah diselesaikan sebelumnya, sehingga solusi yang sudah ada dapat dimanfaatkan kembali secara efisien (Wu dkk. 2023).

Pemrograman kompetitif, yaitu kegiatan menyelesaikan persoalan pemrograman dalam waktu terbatas dan kerap dilombakan, terbukti dapat meningkatkan keterampilan pemecahan masalah sekaligus mendorong pembelajaran mandiri di kalangan mahasiswa. Aktivitas tersebut memaksa mahasiswa untuk berpikir cepat dan efisien dalam merancang solusi, sekaligus melatih ketahanan mental dalam menghadapi persoalan yang menantang (Yuen, Liu, dan Leong 2023).

Meskipun logika komputasi memberikan manfaat besar, sintaks bahasa pemrograman tingkat tinggi sering kali membebani kemampuan kognitif pembelajar pemula, sehingga pendekatan visual, seperti grafis kura-kura atau turtle graphics maupun alat pembelajaran tanpa komputer atau unplugged, direkomendasikan untuk menjembatani pemahaman konsep abstrak sebelum pembelajar dihadapkan pada sintaks bahasa pemrograman yang sesungguhnya (Insuasti, Insuasty, dan Barón 2025; Gulácsi dan Csernoch 2026).

Pendekatan didaktik berbasis grafis kura-kura memungkinkan pembelajar melihat hasil dari setiap

instruksi yang dituliskan secara langsung dalam bentuk visual, seperti pergerakan garis pada layar, sehingga membantu pembelajar memahami hubungan antara instruksi yang dituliskan dengan hasil yang dihasilkan secara lebih konkret dan mudah dipahami (Insuasti, Insuasty, dan Barón 2025).

Pendekatan tanpa komputer atau unplugged, di sisi lain, memungkinkan pembelajar memahami konsep logika komputasi tanpa harus berhadapan langsung dengan perangkat komputer maupun sintaks bahasa pemrograman. Pendekatan ini terbukti efektif untuk menghilangkan beban sintaks pada tahap awal pembelajaran, sehingga pembelajar dapat berfokus sepenuhnya pada pengembangan keterampilan berpikir komputasional dan algoritmik (Gulácsi dan Csernoch 2026).

Kajian sistematis mengenai teknik penyelesaian masalah dalam studi logika komputasi menunjukkan bahwa keempat unsur logika komputasi, yaitu dekomposisi, pengenalan pola, abstraksi, dan algoritma, jarang diajarkan secara terpisah, melainkan cenderung diterapkan secara bersamaan ketika pembelajar dihadapkan pada persoalan nyata yang kompleks. Temuan ini menegaskan bahwa keempat unsur tersebut sebaiknya dipahami sebagai satu kesatuan proses berpikir, bukan sebagai tahapan yang harus dilalui secara kaku dan berurutan (Wu dkk. 2023).

Penerapan logika komputasi dalam pembelajaran pemrograman juga membantu pembelajar mengembangkan sikap tidak mudah menyerah ketika menghadapi kegagalan, karena proses penyelesaian masalah melalui pemrograman secara alami melibatkan banyak percobaan dan perbaikan sebelum solusi yang

tepat ditemukan. Sikap tersebut menjadi salah satu manfaat tidak langsung dari pembelajaran logika komputasi yang turut membentuk karakter pembelajar secara lebih luas, tidak hanya dalam konteks pemrograman semata (Belmar 2022).

4. Struktur Dasar Pemrograman

Struktur sekuensial merupakan struktur paling dasar dalam pemrograman, yaitu rangkaian instruksi yang dijalankan secara berurutan sesuai dengan urutan penulisannya, tanpa adanya percabangan maupun pengulangan. Struktur ini menjadi fondasi bagi pemahaman struktur pemrograman yang lebih kompleks, karena hampir seluruh program pada dasarnya tersusun dari rangkaian instruksi sekuensial yang kemudian dikombinasikan dengan struktur kontrol lainnya (Al-Hafiz dkk. 2024; Belmar 2022).

Struktur percabangan memungkinkan program mengambil keputusan berdasarkan suatu kondisi tertentu, misalnya dengan menggunakan logika jika-maka-lainnya atau if-then-else, sehingga program dapat menjalankan instruksi yang berbeda-beda bergantung pada nilai atau kondisi yang dievaluasi pada saat program dijalankan (Wang, Wan, dan Dai 2026).

Struktur perulangan memungkinkan suatu blok instruksi dijalankan secara berulang selama kondisi tertentu masih terpenuhi, sehingga sangat berguna untuk menyelesaikan persoalan yang melibatkan pengulangan proses, seperti pengolahan data dalam jumlah besar atau pencarian pola tertentu di antara sekumpulan data (Wang, Wan, dan Dai 2026; Balreira, Silveira, dan Wickboldt 2022).

Struktur kontrol, yang mencakup percabangan dan perulangan, merupakan topik yang paling sering

diidentifikasi sebagai sumber kesulitan utama bagi mahasiswa pemula dalam mempelajari pemrograman. Kompleksitas dalam memahami kapan dan bagaimana suatu kondisi harus dievaluasi kerap menjadi hambatan yang signifikan pada tahap awal pembelajaran pemrograman (Ngadengon dkk. 2025).

Mahasiswa sering kali memahami setiap struktur kode secara terpisah, misalnya memahami struktur percabangan dan struktur perulangan masing-masing secara individual, tetapi mengalami kesulitan ketika harus menggabungkan berbagai struktur tersebut menjadi satu program yang valid dan berjalan sesuai harapan. Kesulitan dalam menggabungkan struktur tersebut menunjukkan bahwa pemahaman konseptual saja tidak cukup tanpa latihan penerapan yang memadai (Balreira, Silveira, dan Wickboldt 2022).

Sebagai solusi atas kesulitan tersebut, model analisis masalah algoritmik atau Problem Analysis Algorithmic Model yang berfokus pada masukan, proses, dan keluaran terbukti dapat menurunkan tingkat kegagalan atau attrition rate mahasiswa secara signifikan dibandingkan dengan metode pengajaran pemrograman yang bersifat tradisional. Model ini membantu mahasiswa memecah persoalan pemrograman menjadi tahapan yang lebih terstruktur dan mudah diikuti (Malik et al., 2021).

Penerapan model tersebut dalam mata kuliah pemrograman pengantar juga menunjukkan bahwa pendekatan yang menekankan pemahaman terhadap hubungan antara masukan, proses, dan keluaran sejak awal pembelajaran membantu mahasiswa membangun kerangka berpikir yang lebih kokoh, sehingga mereka tidak mudah kehilangan arah ketika dihadapkan pada

persoalan pemrograman yang lebih kompleks dan melibatkan banyak variabel sekaligus (Malik dkk. 2021).

Penguasaan terhadap ketiga struktur dasar pemrograman tersebut menjadi prasyarat penting sebelum pembelajar dapat menyusun program yang lebih kompleks, seperti program yang melibatkan fungsi, prosedur, maupun struktur data yang lebih rumit. Oleh karena itu, latihan yang cukup pada tahap penguasaan struktur dasar ini sangat menentukan kelancaran pembelajaran pemrograman pada tahap selanjutnya (Zhang dkk. 2021).

Latihan penggabungan struktur sekuensial, percabangan, dan perulangan sebaiknya dilakukan secara bertahap, dimulai dari persoalan sederhana yang hanya melibatkan satu struktur kontrol, kemudian meningkat menuju persoalan yang melibatkan kombinasi beberapa struktur sekaligus. Pendekatan bertahap tersebut membantu pembelajar membangun kepercayaan diri sebelum dihadapkan pada persoalan pemrograman yang benar-benar kompleks dan menuntut penggabungan berbagai struktur kontrol secara bersamaan (Ngadengon dkk. 2025).

Studi eksperimental mengenai pengajaran logika komputasi melalui pendidikan pengodean pada siswa sekolah dasar menunjukkan bahwa paparan dini terhadap struktur dasar pemrograman, meskipun dalam bentuk yang sangat sederhana, memberikan dampak positif terhadap kemampuan pemecahan masalah siswa di kemudian hari. Temuan ini menegaskan pentingnya memperkenalkan struktur dasar pemrograman sejak usia dini, dengan pendekatan yang disesuaikan dengan tingkat perkembangan kognitif peserta didik (Wang, Wan, dan Dai 2026).

5. Pengenalan Bahasa Pemrograman Modern

Bahasa pemrograman diklasifikasikan ke dalam tiga tingkatan utama, yaitu bahasa tingkat rendah atau low-level yang mendekati bahasa mesin, bahasa tingkat menengah atau intermediate-level, serta bahasa tingkat tinggi atau high-level yang lebih mendekati bahasa manusia dan lebih mudah dipahami oleh pembelajar pemula (Tarsini dan Anggraeni 2024).

Bahasa tingkat rendah memberikan kendali yang sangat rinci terhadap perangkat keras komputer, tetapi menuntut pemahaman teknis yang mendalam dan cenderung sulit dipelajari oleh pemula. Sebaliknya, bahasa tingkat tinggi menyembunyikan kerumitan teknis tersebut melalui sintaks yang lebih mendekati bahasa manusia, sehingga lebih ramah bagi pembelajar yang baru memulai perjalanan belajar pemrograman (Fang dkk. 2022).

Bahasa tingkat menengah menempati posisi di antara kedua kategori tersebut, karena tetap menawarkan tingkat kendali yang cukup rinci terhadap perangkat keras komputer, tetapi dengan sintaks yang relatif lebih mudah dipahami dibandingkan bahasa tingkat rendah. Bahasa tingkat menengah sering digunakan pada pengembangan perangkat lunak sistem, seperti sistem operasi atau perangkat penggerak, yang membutuhkan efisiensi tinggi sekaligus kemudahan pemeliharaan kode dalam jangka panjang (Tarsini dan Anggraeni 2024).

Bahasa pemrograman modern yang berorientasi objek sering kali menciptakan hambatan konseptual tersendiri bagi pemula, karena memaksa pembelajar untuk mendefinisikan kelas dan objek sebelum benar-benar memahami konsep dasar, seperti variabel dan

logika kondisional yang menjadi fondasi pemrograman secara umum (Cook 2025).

Sebagai solusi atas hambatan tersebut, lingkungan visual seperti Flowgorithm memungkinkan pembelajar melakukan transisi secara alami dari logika prosedural yang sederhana menuju lebih dari sepuluh bahasa pemrograman populer, seperti Python, Java, dan C++, tanpa harus langsung dihadapkan pada kerumitan konsep berorientasi objek sejak awal pembelajaran (Cook 2025).

Bahasa Python terbukti unggul dalam membangun motivasi dan kepercayaan diri pembelajar pemula dalam mempelajari pemrograman, berkat sintaksnya yang relatif sederhana dan mudah dibaca. Sementara itu, bahasa C terbukti lebih menguntungkan untuk mempermudah pemahaman mengenai struktur data dasar, karena bahasa tersebut memberikan gambaran yang lebih jelas mengenai bagaimana data dikelola secara langsung dalam memori komputer (Balreira, Silveira, dan Wickboldt 2022).

Perbandingan antara bahasa Python dan bahasa C tersebut menunjukkan bahwa pemilihan bahasa pemrograman pengantar perlu mempertimbangkan tujuan pembelajaran yang ingin dicapai. Apabila tujuan utama adalah membangun motivasi dan kepercayaan diri pembelajar pemula, bahasa Python menjadi pilihan yang lebih tepat, sedangkan apabila tujuan utama adalah pemahaman mendalam mengenai struktur data, bahasa C dapat menjadi pilihan yang lebih sesuai (Balreira, Silveira, dan Wickboldt 2022).

Integrasi antara bahasa pemrograman visual berbasis blok, seperti Scratch, dengan bahasa pemrograman berbasis teks, seperti Python, terbukti

memfasilitasi transisi bertahap pembelajar dari logika visual menuju sintaks berbasis teks yang lebih formal. Pendekatan bertahap tersebut membantu mengurangi kejutan kognitif yang biasa dialami pembelajar ketika berpindah dari lingkungan pemrograman visual menuju lingkungan pemrograman berbasis teks secara langsung (Wang, Wan, dan Dai 2026).

Pemilihan bahasa pemrograman modern sebagai bahasa pengantar pembelajaran perlu mempertimbangkan keseimbangan antara beban kognitif yang ditimbulkan oleh sintaks bahasa tersebut dengan tujuan pembelajaran yang ingin dicapai. Pendekatan yang terlalu dini memperkenalkan konsep lanjutan, seperti pemrograman berorientasi objek, berisiko membebani pembelajar pemula yang bahkan belum sepenuhnya menguasai konsep dasar pemrograman (Cook 2025).

Perkembangan bahasa pemrograman modern juga terus mengarah pada peningkatan keterbacaan kode dan kemudahan penggunaan, tanpa mengorbankan kemampuan bahasa tersebut untuk menyelesaikan persoalan komputasi yang kompleks. Tren tersebut menunjukkan bahwa kemudahan belajar dan kekuatan fungsional suatu bahasa pemrograman tidak selalu saling bertentangan, melainkan dapat dirancang untuk saling melengkapi satu sama lain (Wu dkk. 2024).

Pemilihan bahasa pemrograman pengantar yang tepat juga perlu mempertimbangkan konteks penerapannya, misalnya apakah pembelajar akan melanjutkan ke bidang pengembangan perangkat lunak umum, analisis data, pengembangan permainan, maupun bidang rekayasa lain yang memiliki kebutuhan bahasa pemrograman yang berbeda-beda. Kesesuaian antara

bahasa pengantar dengan tujuan akhir pembelajaran akan memperlancar proses transisi pembelajar menuju bahasa pemrograman lanjutan yang lebih spesifik sesuai bidang yang diminati (Wu dkk. 2023; C.-H. Chen dan Chung 2023; Ngadengon dkk. 2025).

Ketersediaan komunitas dan sumber belajar daring yang luas juga menjadi salah satu pertimbangan penting dalam memilih bahasa pemrograman modern sebagai bahasa pengantar. Bahasa pemrograman yang memiliki komunitas besar dan aktif umumnya menyediakan lebih banyak sumber belajar, contoh kode, serta forum diskusi yang dapat membantu pembelajar pemula mengatasi kesulitan yang dihadapi selama proses belajar mandiri (Balreira, Silveira, dan Wickboldt 2022).

Pada akhirnya, keberhasilan pembelajaran bahasa pemrograman modern tidak semata ditentukan oleh pemilihan bahasa yang tepat, tetapi juga oleh konsistensi latihan yang dilakukan pembelajar secara berkelanjutan. Kombinasi antara pemilihan bahasa yang sesuai dengan tujuan pembelajaran, metode pengajaran yang memperhatikan beban kognitif pembelajar, serta latihan yang konsisten menjadi kunci utama keberhasilan seseorang dalam menguasai keterampilan pemrograman secara mendalam (Berrezueta-Guzman, Metaj, dan Wagner 2026; Yuen, Liu, dan Leong 2023; Munasinghe, Bell, dan Robins 2023; Marín-Marín, García-Tudela, dan Dúo-Terrón 2024; Fang dkk. 2022).

Penguasaan menyeluruh terhadap konsep dasar pemrograman, algoritma dan diagram alir, logika komputasi, struktur dasar pemrograman, serta pengenalan bahasa pemrograman modern yang telah diuraikan pada bab ini menjadi fondasi penting sebelum pembelajar melangkah ke topik pemrograman yang lebih

spesifik dan mendalam pada bab-bab selanjutnya (Yuen, Liu, dan Leong 2023; Munasinghe, Bell, dan Robins 2023).

BAB III
PENGEMBANGAN APLIKASI DAN REKAYASA PERANGKAT LUNAK

Pengembangan aplikasi yang baik tidak dapat dilepaskan dari disiplin rekayasa perangkat lunak sebagai fondasi berpikir yang sistematis. Bab ini disusun mulai dari konsep dasar rekayasa perangkat lunak, kemudian bergerak ke siklus hidup pengembangan perangkat lunak atau Siklus Hidup Pengembangan Perangkat Lunak (SDLC), analisis kebutuhan dan perancangan sistem, implementasi, pengujian, dan pemeliharaan aplikasi, hingga metodologi pengembangan perangkat lunak modern. Kajian lintas sumber menegaskan bahwa rekayasa perangkat lunak merupakan pendekatan sistematis untuk membangun perangkat lunak yang bermutu, tepat waktu, terkelola dengan baik, dan sesuai dengan kebutuhan pengguna (D. S. A. S. Kumar dan Bawankar 2025; Pargaonkar 2023b; Zhao, Wang, dan Jiang 2023; Sisodia dan Pote 2024).

1. Konsep Rekayasa Perangkat Lunak

Rekayasa perangkat lunak dapat dipahami sebagai bidang keilmuan yang berfokus pada prinsip, metode, rancangan, dan penjaminan mutu untuk pengembangan, pemeliharaan, serta pengelolaan sistem perangkat lunak secara menyeluruh (D. S. A. S. Kumar dan Bawankar 2025; Qiu, Shi, dan Qiu 2023). Bidang ini tidak berhenti pada aktivitas menulis kode program, tetapi mencakup keseluruhan rangkaian kegiatan mulai dari penetapan

spesifikasi, perancangan, pembangunan, penerapan, evolusi, hingga pemeliharaan sistem, dengan pembagian peran yang jelas antara penyusun kebutuhan, perancang arsitektur, pengembang, penulis kode, dan penguji (Chazette, Brunotte, dan Speith 2022; Pargaonkar 2023a).

Kehadiran rekayasa perangkat lunak sebagai disiplin tersendiri muncul dari kebutuhan untuk mengatasi kompleksitas sistem yang terus meningkat. Ketika perangkat lunak dibangun tanpa kerangka kerja yang terstruktur, risiko kegagalan proyek, pembengkakan biaya, dan ketidaksesuaian hasil dengan kebutuhan pengguna menjadi jauh lebih besar. Oleh karena itu, tujuan bersama dari berbagai metode rekayasa perangkat lunak adalah menghasilkan perangkat lunak seefisien mungkin tanpa mengorbankan mutu, sekaligus menekan risiko kegagalan proyek (Sisodia dan Pote 2024; Zhao, Wang, dan Jiang 2023).

Mutu perangkat lunak tidak hanya dimaknai sebagai kondisi bebas dari kesalahan program, tetapi juga mencakup sejumlah atribut lain, antara lain keandalan, efisiensi kinerja, keamanan, kompatibilitas, keterpeliharaan, dan portabilitas. Dengan demikian, mutu bukan merupakan tahap terpisah yang hanya diperiksa pada akhir proyek, melainkan aspek yang harus dipadukan ke dalam setiap aktivitas rekayasa perangkat lunak sejak tahap paling awal (Ozkaya 2021).

Penerapan standar mutu yang terukur, misalnya rangkaian karakteristik yang dirumuskan dalam ISO/IEC 25010, memberikan acuan yang dapat digunakan bersama oleh seluruh anggota tim pengembang untuk menilai apakah suatu produk perangkat lunak telah memenuhi harapan pengguna dan pemangku

kepentingan. Pemahaman yang utuh terhadap konsep ini menjadi dasar bagi pembahasan selanjutnya mengenai bagaimana proses pengembangan perangkat lunak dijalankan secara bertahap melalui siklus hidup pengembangan perangkat lunak (Anwar 2026).

Cakupan rekayasa perangkat lunak juga meliputi pengelolaan proyek, yaitu bagaimana sumber daya manusia, waktu, dan anggaran diatur agar tujuan pengembangan dapat tercapai tanpa mengorbankan mutu produk akhir. Pengelolaan proyek yang baik memerlukan pemahaman menyeluruh terhadap setiap peran dalam tim, mulai dari penyusun kebutuhan yang menjembatani keinginan pengguna dengan bahasa teknis, arsitek yang merancang struktur sistem secara keseluruhan, pengembang yang menerjemahkan rancangan menjadi kode program, hingga penguji yang memastikan hasil akhir sesuai dengan spesifikasi (Chazette, Brunotte, dan Speith 2022; Sisodia dan Pote 2024).

Prinsip dasar lain yang melandasi rekayasa perangkat lunak adalah pengulangan dan perbaikan berkelanjutan. Suatu sistem perangkat lunak jarang sekali selesai dalam satu kali proses pembangunan, sebab kebutuhan pengguna, teknologi pendukung, dan lingkungan bisnis terus berubah seiring waktu. Oleh sebab itu, kemampuan untuk mengevaluasi hasil kerja, menerima umpan balik, dan melakukan penyempurnaan secara berkelanjutan menjadi bagian yang tidak terpisahkan dari praktik rekayasa perangkat lunak yang sehat (Qiu, Shi, dan Qiu 2023).

Selain aspek teknis, rekayasa perangkat lunak juga menaruh perhatian besar pada aspek manusia, baik dari sisi pengguna akhir maupun dari sisi tim pengembang itu

sendiri. Komunikasi yang efektif antaranggota tim, dokumentasi yang mudah dipahami, serta budaya kerja yang mendukung pembelajaran berkelanjutan turut menentukan keberhasilan penerapan prinsip-prinsip rekayasa perangkat lunak dalam praktik sehari-hari (R. Kumar dkk. 2023).

Penerapan rekayasa perangkat lunak juga tidak dapat dipisahkan dari penerapan penjaminan mutu sebagai kegiatan yang berkelanjutan sepanjang siklus proyek, bukan sekadar kegiatan pemeriksaan pada tahap akhir. Kajian sistematis terhadap berbagai strategi penjaminan mutu menunjukkan bahwa pelibatan aktivitas penjaminan mutu sejak tahap awal pengembangan mampu menurunkan jumlah kesalahan yang lolos hingga tahap penerapan, sekaligus menekan biaya perbaikan yang harus ditanggung organisasi (Shukla dkk. 2024).

2. Siklus Hidup Pengembangan Perangkat Lunak (SDLC)

Siklus Hidup Pengembangan Perangkat Lunak, atau yang lebih dikenal dengan singkatan SDLC, merupakan kerangka kerja rinci yang digunakan untuk mengembangkan, mengubah, memelihara, memperbarui, dan pada akhirnya menggantikan suatu perangkat lunak (Sisodia dan Pote 2024; Ghumatkar dan Date 2023; Agarwal dkk. 2023). Kerangka kerja ini memberikan struktur langkah demi langkah yang memudahkan tim pengembang dalam melakukan perencanaan, pengendalian mutu, alokasi sumber daya, serta meningkatkan peluang keberhasilan proyek secara keseluruhan (Sisodia dan Pote 2024; Ghumatkar dan Date 2023).

Berbagai kajian menunjukkan kesesuaian pandangan mengenai tahapan yang paling konsisten muncul di dalam SDLC, yaitu analisis kebutuhan, perancangan, implementasi atau penulisan kode program, pengujian atau validasi, penerapan, dan pemeliharaan atau evolusi sistem. Setiap tahap memiliki fokus dan luaran yang berbeda, tetapi saling berkaitan satu sama lain sehingga membentuk satu kesatuan proses yang utuh (R. Kumar dkk. 2023; Agarwal dkk. 2023; Ghumatkar dan Date 2023; Ranjan 2024).

Struktur bertahap semacam ini memberikan sejumlah manfaat penting. Pertama, setiap tahap memiliki luaran yang jelas sehingga kemajuan proyek dapat dipantau secara objektif. Kedua, pembagian tahap memudahkan identifikasi titik-titik yang berpotensi menimbulkan risiko sehingga tindakan pencegahan dapat dilakukan lebih awal. Ketiga, dokumentasi yang dihasilkan pada setiap tahap menjadi rujukan bagi tahap berikutnya sehingga mengurangi kesalahpahaman antaranggota tim. Ketiga manfaat tersebut menegaskan bahwa SDLC bukan sekadar urutan administratif, melainkan kerangka yang membantu tim pengembang mengelola kompleksitas proyek perangkat lunak dari awal hingga akhir (Ghumatkar dan Date 2023).

Meskipun tahapan inti relatif konsisten di berbagai sumber, cara pelaksanaan tahapan tersebut dapat berbeda-beda bergantung pada model atau metodologi yang dipilih, mulai dari model yang bersifat berurutan hingga model yang bersifat iteratif dan adaptif. Pembahasan lebih lanjut mengenai pilihan metodologi tersebut disajikan pada bagian akhir bab ini (Ahmad dkk. 2023).

Tahap analisis kebutuhan menjadi titik awal yang menentukan arah keseluruhan proyek, sebab kesalahan dalam memahami kebutuhan pengguna pada tahap ini akan berdampak pada seluruh tahap berikutnya dan cenderung semakin mahal untuk diperbaiki apabila baru disadari pada tahap akhir. Oleh karena itu, ketelitian dalam menggali, mendokumentasikan, dan memverifikasi kebutuhan pengguna pada tahap awal SDLC memberikan dampak yang besar terhadap efisiensi keseluruhan proyek (Agarwal dkk. 2023).

Tahap desain berperan menerjemahkan kebutuhan yang telah disepakati menjadi arsitektur sistem, struktur data, serta rancangan antarmuka yang akan digunakan oleh pengguna. Keputusan yang diambil pada tahap ini, misalnya pemilihan struktur basis data atau pembagian modul aplikasi, akan memengaruhi kemudahan pengembangan dan pemeliharaan sistem pada tahap-tahap selanjutnya (Ramachandrappa 2024).

Pada tahap implementasi, tim pengembang mulai menuliskan kode program berdasarkan rancangan yang telah disusun sebelumnya. Konsistensi terhadap standar penulisan kode dan pedoman desain menjadi hal yang penting agar hasil kerja setiap anggota tim dapat digabungkan menjadi satu sistem yang utuh tanpa menimbulkan konflik struktur maupun duplikasi fungsi (Agarwal dkk. 2023).

Tahap pengujian dan penerapan memastikan bahwa perangkat lunak yang telah dibangun benar-benar berfungsi sesuai kebutuhan sebelum digunakan secara luas oleh pengguna, sedangkan tahap pemeliharaan menjamin sistem tetap dapat digunakan dan dikembangkan sejalan dengan perubahan kebutuhan di masa mendatang. Keterkaitan yang erat antartahap inilah

yang menjadikan SDLC sebagai kerangka kerja yang bersifat siklis, bukan sekadar urutan linear yang berhenti setelah aplikasi selesai dibangun (Ranjan 2024).

3. Analisis Kebutuhan dan Perancangan Sistem

Rekayasa kebutuhan atau proses analisis kebutuhan memiliki kaitan yang erat dengan keberhasilan proyek, baik dari sisi ketepatan waktu penyelesaian maupun kesesuaian dengan anggaran yang telah ditetapkan (Laplante dan Kassab 2022). Kebutuhan yang dirumuskan secara jelas, lengkap, dan tidak ambigu akan menjadi dasar yang kuat bagi penyusunan peta jalan pengembangan serta strategi pengujian yang lebih efektif pada tahap-tahap berikutnya (White dkk. 2023).

Proses analisis kebutuhan tidak semata-mata bertujuan mengidentifikasi daftar fungsi yang harus dimiliki oleh sistem, tetapi juga membangun kesepahaman bersama di antara para pemangku kepentingan. Kesepahaman ini berperan penting dalam menurunkan risiko miskomunikasi, ketidakpuasan pengguna, dan pekerjaan ulang yang timbul akibat kebutuhan yang tidak terdefinisi dengan baik sejak awal (Chazette, Brunotte, dan Speith 2022).

Setelah kebutuhan dirumuskan, tahap perancangan sistem bertugas menerjemahkan tujuan yang bersifat abstrak dan kebutuhan mutu menjadi keputusan rancangan yang konkret dan dapat dievaluasi (Chazette, Brunotte, dan Speith 2022). Pada tahap ini, prinsip modularitas, enkapsulasi, dan abstraksi digunakan untuk menghasilkan sistem yang lebih efisien, mudah dipelihara, dan mampu berkembang mengikuti pertumbuhan kebutuhan di masa mendatang (R. Kumar dkk. 2023). Modularitas memungkinkan sistem dipecah menjadi bagian-bagian yang dapat dikembangkan dan

diuji secara terpisah, enkapsulasi menyembunyikan detail implementasi yang tidak perlu diketahui oleh bagian lain, sedangkan abstraksi membantu tim berpikir pada tingkat konsep tanpa terjebak pada detail teknis yang berlebihan (R. Kumar dkk. 2023).

Perkembangan kebutuhan pengguna dan teknologi turut memunculkan tuntutan baru dalam proses analisis kebutuhan dan perancangan sistem. Bukti terkini menunjukkan bahwa aspek keterjelasan atau explainability serta pertimbangan kecerdasan buatan yang berpusat pada manusia perlu ditangkap sejak tahap awal, mengingat kebutuhan pada tahap awal proyek dan kebutuhan pada tahap lanjut tidak selalu sama. Kondisi ini menegaskan pentingnya proses analisis kebutuhan yang bersifat berkelanjutan, bukan hanya kegiatan satu kali di awal proyek (Ahmad dkk. 2023; Chazette, Brunotte, dan Speith 2022).

Analisis kebutuhan yang mendalam turut mencakup identifikasi kasus penggunaan atau skenario interaksi pengguna dengan sistem, yang kemudian digunakan sebagai dasar perancangan antarmuka maupun struktur data. Pendekatan analisis kebutuhan dan kasus penggunaan yang matang membantu tim pengembang menyusun spesifikasi yang lebih terperinci sekaligus mengurangi celah antara harapan pengguna dan hasil akhir sistem (D. S. A. S. Kumar dan Bawankar 2025).

Kebutuhan perangkat lunak umumnya dibedakan menjadi dua kelompok besar, yaitu kebutuhan fungsional yang menjelaskan fungsi atau layanan yang harus disediakan oleh sistem, dan kebutuhan nonfungsional yang menjelaskan batasan mutu seperti kinerja, keamanan, dan kemudahan penggunaan. Kedua

kelompok kebutuhan tersebut perlu didokumentasikan secara seimbang, sebab kebutuhan nonfungsional yang terabaikan sering kali menjadi sumber ketidakpuasan pengguna meskipun seluruh kebutuhan fungsional telah terpenuhi (Laplante dan Kassab 2022).

Verifikasi kebutuhan juga menjadi langkah yang tidak boleh dilewatkan sebelum kebutuhan disepakati sebagai acuan pengembangan. Verifikasi ini dilakukan dengan mengembalikan hasil penggalian kebutuhan kepada pemangku kepentingan untuk dikonfirmasi kebenaran dan kelengkapannya, sehingga potensi kesalahpahaman dapat dikoreksi sebelum berlanjut ke tahap perancangan yang membutuhkan biaya perbaikan lebih besar apabila kesalahan baru ditemukan kemudian (Pargaonkar 2023a; Laplante dan Kassab 2022).

Teknik penggalian kebutuhan dapat dilakukan melalui berbagai cara, misalnya wawancara dengan pengguna, pengamatan langsung terhadap proses kerja yang sudah berjalan, penyebaran kuesioner, maupun diskusi kelompok terfokus bersama pemangku kepentingan. Pemilihan teknik yang tepat bergantung pada karakteristik pengguna, kompleksitas domain, serta waktu yang tersedia bagi tim pengembang untuk melakukan penggalian kebutuhan secara menyeluruh (Subahi 2023).

Perancangan sistem tidak hanya berhenti pada perancangan struktur data dan logika program, tetapi juga mencakup perancangan antarmuka pengguna yang memudahkan interaksi antara manusia dan sistem. Antarmuka yang dirancang dengan memperhatikan kemudahan penggunaan akan menurunkan kebutuhan pelatihan bagi pengguna baru sekaligus mengurangi

kesalahan penggunaan pada saat sistem dioperasikan (Chazette, Brunotte, dan Speith 2022).

Dokumentasi hasil analisis kebutuhan dan perancangan sistem, misalnya dalam bentuk spesifikasi kebutuhan perangkat lunak dan diagram rancangan, berperan sebagai acuan bersama bagi seluruh anggota tim sepanjang proyek berlangsung. Dokumentasi yang tersusun rapi juga mempermudah proses pemeliharaan pada masa mendatang, terutama ketika terjadi pergantian anggota tim atau penambahan fungsi baru pada sistem yang telah berjalan (D. S. A. S. Kumar dan Bawankar 2025).

4. Implementasi, Pengujian, dan Pemeliharaan Aplikasi

Implementasi, pengujian, dan pemeliharaan merupakan rangkaian fase eksekusi yang memastikan perangkat lunak benar-benar bekerja sesuai spesifikasi yang telah dirancang dan tetap memberikan nilai setelah dirilis kepada pengguna. Pada tahap implementasi, rancangan yang telah disusun diterjemahkan menjadi kode program yang dapat dijalankan, dengan tetap memperhatikan standar penulisan kode, konsistensi struktur, dan kemudahan pemeliharaan di masa mendatang (Pargaonkar 2023a; D. S. A. S. Kumar dan Bawankar 2025; Ranjan 2024).

Pengujian menjadi aktivitas inti untuk menemukan kesalahan, memeriksa kesenjangan antara perilaku sistem dengan ekspektasi pengguna, serta membangun keyakinan bahwa program benar-benar menjalankan fungsi yang dimaksudkan. Siklus pengujian pada umumnya mencakup enam tahap, yaitu analisis kebutuhan pengujian, perencanaan pengujian, penyusunan kasus uji, penyiapan lingkungan pengujian,

pelaksanaan pengujian, hingga penutupan dan pelaporan hasil (Ranjan 2024).

Keterlacakan atau traceability berperan menghubungkan kebutuhan, kegiatan pengujian, dan pengelolaan perubahan menjadi satu kesatuan yang dapat ditelusuri sepanjang siklus proyek. Dengan keterlacakan yang baik, setiap perubahan kebutuhan dapat segera diketahui dampaknya terhadap rancangan, kode program, maupun kasus uji yang telah disusun, sehingga potensi kesalahan akibat perubahan yang tidak terpantau dapat ditekan (Pargaonkar 2023a).

Standar mutu seperti ISO/IEC 25010 merangkum karakteristik mutu perangkat lunak yang dapat diukur secara objektif, mulai dari fungsionalitas, keandalan, kemudahan penggunaan, efisiensi kinerja, kompatibilitas, keamanan, keterpeliharaan, hingga portabilitas (Anwar, 2026). Penerapan standar semacam ini membantu organisasi menilai kesiapan suatu aplikasi sebelum diterapkan pada lingkungan operasional yang sesungguhnya.

Setelah aplikasi diterapkan, tahap pemeliharaan berperan menjaga sistem agar tetap relevan dengan kebutuhan yang terus berkembang, baik melalui perbaikan kesalahan yang ditemukan setelah rilis, penyesuaian terhadap perubahan lingkungan teknologi, maupun penambahan fungsi baru sesuai permintaan pengguna (Agarwal dkk. 2023; Ghumatkar dan Date 2023). Penerapan strategi penjaminan mutu yang konsisten sepanjang siklus hidup pengembangan, bukan hanya pada tahap pengujian, terbukti membantu organisasi menekan biaya perbaikan dan meningkatkan kepuasan pengguna dalam jangka panjang (Shukla et al., 2024).

Pengujian perangkat lunak dapat dilakukan pada beberapa tingkatan, mulai dari pengujian unit yang memeriksa fungsi terkecil dalam kode program, pengujian integrasi yang memeriksa interaksi antarmodul, pengujian sistem yang menilai kinerja sistem secara keseluruhan, hingga pengujian penerimaan yang melibatkan pengguna akhir untuk memastikan sistem siap digunakan dalam lingkungan operasional. Setiap tingkatan pengujian memiliki tujuan yang berbeda, tetapi seluruhnya diperlukan untuk membangun keyakinan yang menyeluruh terhadap mutu perangkat lunak (Pargaonkar 2023a; D. S. A. S. Kumar dan Bawankar 2025; Ranjan 2024; Subahi 2023).

Kesalahan yang ditemukan pada tahap pengujian perlu dicatat, diprioritaskan berdasarkan tingkat keparahannya, kemudian diperbaiki oleh tim pengembang sebelum dilakukan pengujian ulang. Proses pencatatan dan penelusuran kesalahan yang tertata baik membantu tim memastikan bahwa setiap masalah benar-benar terselesaikan dan tidak muncul kembali pada versi perangkat lunak berikutnya (Marlowe dkk. 2026).

Selain pengujian yang dilakukan secara manual oleh penguji, banyak organisasi kini mengembangkan pengujian otomatis yang dijalankan setiap kali terdapat perubahan pada kode program. Pengujian otomatis memungkinkan tim mendeteksi kesalahan lebih cepat dan mengurangi ketergantungan pada pengujian manual yang memakan waktu, meskipun penyusunan skenario pengujian otomatis tetap memerlukan perencanaan yang matang agar benar-benar mencerminkan kebutuhan pengguna (Zhao, Wang, dan Jiang 2023).

Pemeliharaan perangkat lunak umumnya dibedakan menjadi beberapa jenis, yaitu pemeliharaan

korektif untuk memperbaiki kesalahan yang ditemukan setelah rilis, pemeliharaan adaptif untuk menyesuaikan sistem terhadap perubahan lingkungan seperti pembaruan perangkat keras atau sistem operasi, pemeliharaan perfektif untuk meningkatkan kinerja atau menambah fungsi baru, serta pemeliharaan preventif untuk mencegah timbulnya masalah di masa mendatang. Keempat jenis pemeliharaan tersebut memerlukan perhatian yang berimbang agar sistem tetap dapat diandalkan sepanjang masa pakainya (Ghumatkar dan Date 2023).

Biaya yang dikeluarkan pada tahap pemeliharaan sering kali jauh lebih besar dibandingkan biaya pada tahap pembangunan awal, terutama apabila dokumentasi kode program dan hasil rancangan tidak disusun dengan baik sejak awal proyek. Oleh sebab itu, investasi pada praktik penulisan kode yang rapi, pengujian yang menyeluruh, serta dokumentasi yang lengkap pada tahap awal pengembangan pada akhirnya akan memberikan penghematan biaya pemeliharaan dalam jangka Panjang (Anwar 2026; Shukla dkk. 2024).

5. Metodologi Pengembangan Perangkat Lunak Modern

Pemilihan metodologi pengembangan perangkat lunak sangat memengaruhi cara suatu tim menjalankan tahapan-tahapan yang telah dibahas sebelumnya. Model Waterfall atau model air terjun cocok diterapkan pada proyek dengan kebutuhan yang relatif stabil dan menuntut dokumentasi yang lengkap, tetapi model ini kurang fleksibel ketika terjadi perubahan kebutuhan di tengah proyek (Pargaonkar, 2023).

Sebagai jawaban atas keterbatasan tersebut, pendekatan Agile, Scrum, DevOps, serta pendekatan yang dibantu kecerdasan buatan mulai banyak digunakan

untuk menghadapi perubahan yang cepat, meningkatkan kolaborasi antaranggota tim, dan memperluas otomatisasi dalam proses pengembangan (Pargaonkar 2023b; R. Kumar dkk. 2023; Marlowe dkk. 2026). Pendekatan Agile menekankan pengembangan bertahap dan berulang dengan siklus yang singkat sehingga umpan balik pengguna dapat segera diakomodasi, sementara Scrum menyediakan kerangka kerja yang lebih terstruktur berupa pembagian waktu kerja ke dalam periode singkat yang disebut sprint (Qiu, Shi, dan Qiu 2023).

Pendekatan DevOps memperluas kolaborasi tidak hanya antara perancang dan pengembang, tetapi juga melibatkan tim operasional sejak tahap awal, dengan tujuan mempercepat penerapan perangkat lunak sekaligus menjaga kestabilan sistem yang sedang berjalan. Sementara itu, kehadiran kecerdasan buatan generatif turut membuka peluang otomatisasi pada berbagai aktivitas rekayasa perangkat lunak, mulai dari penyusunan kebutuhan, penulisan kode, hingga pembuatan kasus uji, sehingga kurikulum dan praktik pendidikan rekayasa perangkat lunak juga perlu menyesuaikan diri dengan perkembangan tersebut (Marlowe dkk. 2026).

Pemilihan metodologi pengembangan pada akhirnya bergantung pada karakteristik kebutuhan proyek, tingkat keahlian tim, konteks domain aplikasi yang dibangun, serta pertimbangan pertukaran antara prediktabilitas, fleksibilitas, biaya, dan mutu yang ingin dicapai (Sisodia dan Pote 2024; Shukla dkk. 2024). Tidak ada satu metodologi yang secara mutlak lebih unggul dibandingkan metodologi lain untuk seluruh jenis proyek, sehingga kemampuan tim dalam menyesuaikan

pendekatan dengan konteks proyek menjadi keterampilan yang sangat penting (Agarwal dkk. 2023).

Secara keseluruhan, pengembangan aplikasi yang baik tidak hanya bergantung pada kemampuan menulis kode program, tetapi juga pada kemampuan menyatukan analisis kebutuhan, perancangan, implementasi, pengujian, pemeliharaan, serta pemilihan metodologi yang tepat ke dalam satu proses rekayasa yang terstruktur sekaligus adaptif terhadap perubahan (Ramachandrappa 2024).

Model Waterfall menjalankan setiap tahap SDLC secara berurutan, sehingga suatu tahap baru dapat dimulai setelah tahap sebelumnya benar-benar selesai. Karakteristik ini menjadikan model Waterfall relatif mudah dikelola dan diawasi kemajuannya, tetapi juga membuat model ini kurang mampu mengakomodasi perubahan kebutuhan yang muncul di tengah perjalanan proyek, terutama pada proyek dengan tingkat ketidakpastian kebutuhan yang tinggi .(Pargaonkar 2023a)

Berbeda dengan model Waterfall, pendekatan Agile membagi pengembangan menjadi iterasi-iterasi singkat yang masing-masing menghasilkan bagian fungsi yang dapat digunakan atau dievaluasi oleh pengguna. Pendekatan ini memungkinkan tim menerima umpan balik lebih cepat dan menyesuaikan arah pengembangan tanpa harus menunggu keseluruhan proyek selesai, sehingga lebih sesuai untuk proyek dengan kebutuhan yang masih dapat berubah (Pargaonkar, 2023; Kumar et al., 2023).

Kerangka kerja Scrum menerapkan prinsip Agile melalui pembagian pekerjaan ke dalam periode kerja singkat yang disebut sprint, disertai pertemuan rutin

untuk perencanaan, pemantauan kemajuan, dan evaluasi hasil kerja pada setiap akhir sprint. Peran-peran khusus dalam Scrum, misalnya pemilik produk dan fasilitator tim, membantu menjaga fokus tim terhadap prioritas yang telah disepakati bersama pemangku kepentingan (Anwar 2026).

Di samping model Waterfall, Agile, dan Scrum, terdapat pula pendekatan campuran yang menggabungkan unsur perencanaan menyeluruh di awal proyek dengan fleksibilitas pengembangan bertahap pada pelaksanaannya. Pendekatan campuran semacam ini sering dipilih oleh organisasi yang membutuhkan kepastian ruang lingkup dan anggaran di awal proyek, tetapi tetap ingin memberi ruang penyesuaian terhadap perubahan kebutuhan yang wajar terjadi selama pengembangan berlangsung (Sisodia dan Pote 2024).

Pendekatan DevOps menekankan penyatuan antara aktivitas pengembangan dan operasional melalui otomatisasi proses pembangunan, pengujian, dan penerapan perangkat lunak secara berkesinambungan. Dengan otomatisasi tersebut, perubahan kode program dapat diuji dan diterapkan ke lingkungan operasional dalam waktu yang jauh lebih singkat dibandingkan pendekatan tradisional, sekaligus menurunkan risiko kesalahan akibat proses manual (Ramachandrappa 2024).

Perkembangan terkini menunjukkan bahwa kecerdasan buatan generatif mulai dimanfaatkan untuk membantu berbagai aktivitas rekayasa perangkat lunak, misalnya penyusunan pola permintaan untuk memperbaiki mutu kode program, membantu proses penyusunan ulang kode, penggalian kebutuhan, hingga perancangan sistem (White dkk. 2023). Perkembangan

ini turut mendorong perubahan pada kurikulum pendidikan rekayasa perangkat lunak agar lulusan memiliki kemampuan bekerja sama dengan perangkat berbasis kecerdasan buatan secara bertanggung jawab (Marlowe dkk. 2026).

BAB IV
JARINGAN DAN SISTEM OPERASI

Pembahasan pada bab ini dibagi menjadi tiga inti yang saling berkaitan, yaitu sistem operasi sebagai pengelola sumber daya komputer, jaringan komputer sebagai infrastruktur komunikasi antarperangkat, dan keamanan dasar sebagai lapisan perlindungan bagi keduanya. Berbagai kajian menunjukkan kesesuaian pandangan bahwa sistem operasi bertindak sebagai perantara antara perangkat keras dan perangkat lunak, sedangkan jaringan komputer menghubungkan berbagai perangkat melalui protokol standar untuk keperluan pertukaran data (Jia dkk. 2024; Odun-Ayo dkk. 2021; Umar 2023; Kaur 2023).

1. Konsep Dasar Sistem Operasi

Sistem operasi merupakan lapisan inti yang menyediakan layanan penting bagi jalannya aplikasi serta pengelolaan sumber daya perangkat keras pada suatu komputer. Tanpa sistem operasi, setiap aplikasi harus berinteraksi secara langsung dengan perangkat keras, sehingga proses pengembangan perangkat lunak menjadi jauh lebih rumit dan rawan kesalahan (Jia dkk. 2024; Relan, Desai, dan Sharma 2026; Petrizia, Sujitha, dan Saradha 2025).

Sistem operasi berperan sebagai penghubung antara perangkat keras dan perangkat lunak, dengan menyediakan lapisan abstraksi yang menyembunyikan kerumitan perangkat keras dari pengguna maupun pengembang aplikasi. Dengan adanya lapisan ini, aplikasi dapat dijalankan pada berbagai jenis perangkat keras

tanpa harus ditulis ulang secara khusus untuk setiap perangkat (Jia dkk. 2024; Odun-Ayo dkk. 2021; Umar 2023).

Fungsi dasar sistem operasi mencakup pengelolaan proses, pengelolaan memori, pengelolaan berkas, pengelolaan perangkat, dan pengelolaan keamanan sistem. Kelima fungsi tersebut bekerja secara terpadu untuk memastikan seluruh sumber daya komputer digunakan secara efisien dan aman oleh berbagai aplikasi yang berjalan secara bersamaan (Relan, Desai, dan Sharma 2026; Petrizia, Sujitha, dan Saradha 2025).

Inti dari sistem operasi terletak pada kernel, yaitu bagian yang memiliki akses langsung terhadap perangkat keras dan bertanggung jawab menjalankan fungsi paling mendasar. Rancangan kernel dapat bersifat monolitik, mikrokernel, atau hibrida, dengan pertukaran antara kinerja, keterujian, dan kemampuan beradaptasi terhadap kebutuhan sistem modern (Relan, Desai, dan Sharma 2026).

Kajian perbandingan atribut mutu sistem operasi menunjukkan bahwa setiap rancangan sistem operasi memiliki kelebihan dan keterbatasan tersendiri, misalnya dari sisi keandalan, kinerja, keamanan, maupun kemudahan pemeliharaan, sehingga pemilihan sistem operasi perlu disesuaikan dengan kebutuhan spesifik dari perangkat atau lingkungan yang akan menjalankannya (Odun-Ayo dkk. 2021).

Selain menjalankan fungsi teknis, sistem operasi juga menyediakan antarmuka yang memungkinkan pengguna maupun program lain berinteraksi dengan sumber daya komputer melalui perintah atau tampilan grafis. Antarmuka ini dirancang agar pengguna dapat

memanfaatkan seluruh kemampuan komputer tanpa perlu memahami detail teknis pengelolaan perangkat keras yang berjalan di baliknya (Ogrizović dkk. 2024).

Sistem operasi juga berperan sebagai pengelola sumber daya bersama pada komputer yang menjalankan banyak aplikasi sekaligus. Tanpa pengaturan yang baik, aplikasi yang berjalan bersamaan dapat saling berebut sumber daya seperti pemroses dan memori, sehingga menyebabkan sistem menjadi lambat atau bahkan tidak responsif. Oleh sebab itu, penjadwalan penggunaan sumber daya menjadi salah satu tugas mendasar yang harus dijalankan secara adil dan efisien oleh sistem operasi (Petrizia, Sujitha, dan Saradha 2025).

Keamanan sistem juga menjadi bagian dari konsep dasar sistem operasi, sebab sistem operasi bertanggung jawab mengatur hak akses setiap pengguna maupun program terhadap berkas dan sumber daya lain di dalam komputer. Pengaturan hak akses yang tepat mencegah suatu program mengganggu data atau proses milik program lain tanpa izin, sehingga menjaga kestabilan sistem secara keseluruhan (Relan, Desai, dan Sharma 2026).

Evolusi sistem operasi dari masa ke masa menunjukkan pergeseran fokus rancangan, mulai dari sistem operasi yang hanya mampu menjalankan satu program pada satu waktu, hingga sistem operasi modern yang mampu menjalankan banyak proses secara bersamaan, mendukung banyak pengguna, serta beradaptasi dengan berbagai jenis perangkat keras yang terus berkembang (Jia dkk. 2024).

Konsep dasar sistem operasi juga mencakup pemahaman mengenai mode kerja pemroses, yaitu mode pengguna dan mode kernel. Program aplikasi biasa

dijalankan pada mode pengguna dengan hak akses terbatas, sedangkan fungsi inti sistem operasi dijalankan pada mode kernel dengan hak akses penuh terhadap perangkat keras. Pemisahan mode ini menjadi salah satu mekanisme dasar untuk melindungi sistem dari kesalahan atau tindakan program yang dapat membahayakan kestabilan keseluruhan sistem (Narwal dkk. 2024).

Sistem operasi modern juga dituntut untuk mendukung komputasi tersebar dan komputasi berbasis agen, sejalan dengan meningkatnya kebutuhan aplikasi yang berjalan lintas perangkat dan lintas lokasi. Kajian terkini menunjukkan bahwa rancangan sistem operasi ke depan perlu mempertimbangkan dukungan terhadap agen perangkat lunak otonom yang bekerja mewakili pengguna, sehingga fungsi pengelolaan sumber daya tradisional perlu disesuaikan dengan pola interaksi yang baru (Jia dkk. 2024).

2. Fungsi dan Jenis Sistem Operasi

Pengelolaan proses menjadi fungsi sentral dalam sistem operasi, sebab sistem operasi bertugas membuat, menjalankan, menangguhkan, dan menghentikan proses sambil mengatur penggunaan pemroses dan memori secara bersamaan. Setiap aplikasi yang berjalan pada komputer sesungguhnya merupakan satu atau lebih proses yang dikelola secara ketat oleh sistem operasi (Petrizia, Sujitha, dan Saradha 2025).

Keadaan suatu proses umumnya dilacak melalui beberapa tahap, yaitu baru, siap, berjalan, menunggu, dan berakhir, sedangkan blok kendali proses digunakan oleh kernel untuk menyimpan status proses dan mendukung perpindahan konteks antarproses. Mekanisme ini memungkinkan banyak proses berjalan

secara bergantian dalam waktu yang sangat singkat sehingga tampak berjalan bersamaan dari sudut pandang pengguna (Leidigh 2025).

Pengelolaan memori turut menentukan kinerja sistem secara keseluruhan, sebab pemanfaatan memori yang efisien akan meningkatkan kecepatan respons sistem sekaligus utilisasi sumber daya yang tersedia. Teknik seperti penomoran halaman, segmentasi, dan memori maya menjadi dasar pengelolaan memori pada sebagian besar sistem operasi modern, sedangkan teknik lanjutan seperti penyalinan saat penulisan dan penggembungan memori turut memperkaya evolusi pengelolaan memori pada sistem operasi masa kini (Jalaman dan Teleron 2024).

Sistem operasi tertanam dirancang untuk bekerja pada perangkat dengan sumber daya terbatas namun menuntut respons waktu nyata, sehingga banyak digunakan pada perangkat internet untuk segala, kendaraan bermotor, dan sistem otomasi industri. Sistem operasi desktop lebih menitikberatkan pada kenyamanan penggunaan dan keluasan fungsi umum bagi penggunanya, sementara sistem operasi bergerak dirancang dengan penekanan pada efisiensi daya baterai dan dukungan konektivitas yang berpindah-pindah (Jia dkk. 2024).

Sistem operasi server dirancang untuk melayani banyak permintaan secara bersamaan dengan tingkat keandalan dan keamanan yang tinggi, sehingga banyak digunakan pada pusat data dan lingkungan komputasi awan (Jia dkk. 2024). Kajian perbandingan terhadap sistem operasi populer, seperti Windows, iOS, dan Android, menunjukkan bahwa setiap sistem operasi memiliki pendekatan berbeda dalam pengelolaan

memori dan penerapan mekanisme keamanan, sehingga tingkat kerentanan dan kinerjanya juga bervariasi bergantung pada rancangan arsitektur masing-masing (Umar 2023).

Pengelolaan berkas merupakan fungsi lain yang tidak kalah penting dalam sistem operasi, sebab hampir seluruh aktivitas pengguna melibatkan penyimpanan dan pengambilan data dalam bentuk berkas. Sistem operasi menyediakan struktur direktori, mekanisme penamaan, serta pengaturan hak akses berkas agar data dapat disimpan secara terorganisasi dan hanya dapat diakses oleh pihak yang berwenang (Sankpal dkk. 2022).

Pengelolaan perangkat keras tambahan, seperti pencetak, papan ketik, dan perangkat penyimpanan luar, juga menjadi tanggung jawab sistem operasi melalui program pengendali perangkat atau driver. Program pengendali ini menjembatani perbedaan cara kerja antarperangkat keras sehingga aplikasi dapat menggunakan perangkat tersebut melalui antarmuka yang seragam tanpa perlu mengetahui detail teknis setiap merek perangkat (Roopesh 2024).

Perbandingan antarjenis sistem operasi menegaskan bahwa tidak terdapat satu rancangan sistem operasi yang secara mutlak lebih unggul untuk seluruh kebutuhan. Sistem operasi tertanam mengutamakan efisiensi dan kepastian waktu tanggap, sistem operasi desktop mengutamakan kenyamanan pengguna, sistem operasi bergerak mengutamakan efisiensi daya, sedangkan sistem operasi server mengutamakan keandalan dan keamanan pada skala besar (Jia dkk. 2024).

Pemilihan jenis sistem operasi yang tepat pada akhirnya harus mempertimbangkan karakteristik

perangkat keras yang tersedia, kebutuhan pengguna, serta lingkungan tempat sistem tersebut akan dijalankan. Pemahaman terhadap fungsi dan jenis sistem operasi ini menjadi dasar penting sebelum membahas bagaimana perangkat-perangkat tersebut saling terhubung melalui jaringan komputer pada bagian berikutnya (Al-Hazaimeh dkk. 2025).

3. Konsep Dasar Jaringan Komputer

Jaringan komputer dapat dipahami sebagai kumpulan perangkat yang saling terhubung untuk bertukar data dan berbagi sumber daya melalui media kabel maupun nirkabel. Keberadaan jaringan komputer memungkinkan berbagai perangkat yang terpisah secara fisik untuk tetap dapat berkomunikasi dan berbagi informasi secara efisien (Kaur 2023).

Titik dalam suatu jaringan, yang sering disebut simpul atau node, dapat berupa komputer, telepon pintar, stasiun kerja, sakelar, maupun perute, sedangkan penghubung antarsimpul dapat berupa kabel, gelombang radio, atau media transmisi lain. Kombinasi antara simpul dan penghubung inilah yang membentuk struktur dasar suatu jaringan computer (Kaur 2023).

Protokol jaringan merupakan seperangkat aturan yang mengatur bagaimana data dipertukarkan antara pengirim dan penerima, sehingga perangkat yang berbeda jenis maupun merek tetap dapat saling berkomunikasi dengan lancar. Tanpa adanya protokol yang disepakati bersama, perangkat dari produsen berbeda akan kesulitan untuk saling bertukar data secara benar (Ramachandrappa 2024).

Untuk menjaga kompatibilitas, skalabilitas, dan pemisahan fungsi antarlapisan, komunikasi data pada jaringan komputer umumnya dijelaskan melalui model

berlapis, seperti model OSI dan model TCP/IP. Setiap lapisan pada model tersebut memiliki tanggung jawab khusus, mulai dari pengaturan sinyal fisik pada lapisan paling bawah hingga penyediaan layanan bagi aplikasi pada lapisan paling atas (Narwal dkk. 2024).

Pendekatan berlapis ini memberikan keuntungan besar dalam pengembangan teknologi jaringan, sebab perubahan atau pembaruan pada satu lapisan tidak selalu memengaruhi lapisan lainnya selama antarmuka antarlapisan tetap dijaga konsistensinya. Dengan demikian, produsen perangkat keras dan pengembang perangkat lunak jaringan dapat bekerja secara independen tanpa harus saling bergantung pada rincian teknis pihak lain (Narwal dkk. 2024).

Jaringan komputer dapat dibedakan berdasarkan cakupan wilayahnya, misalnya jaringan wilayah lokal yang menghubungkan perangkat dalam satu gedung atau area terbatas, dan jaringan wilayah luas yang menghubungkan perangkat pada wilayah geografis yang lebih besar. Pemahaman terhadap cakupan wilayah ini penting sebagai dasar untuk memilih topologi, perangkat, dan protokol yang sesuai dengan kebutuhan organisasi (Al-Hazaimeh dkk. 2025).

Manfaat utama jaringan komputer terletak pada kemampuannya memungkinkan berbagi sumber daya, misalnya berbagi pencetak, penyimpanan data, maupun sambungan internet, sehingga organisasi tidak perlu menyediakan sumber daya yang sama secara berulang untuk setiap perangkat. Selain efisiensi biaya, jaringan komputer juga memudahkan komunikasi dan kolaborasi antarpengguna dalam suatu organisasi (Kaur 2023).

Proses komunikasi data pada jaringan komputer melibatkan pengubahan data menjadi bentuk yang dapat

dikirim melalui media transmisi, kemudian diterjemahkan kembali oleh perangkat penerima menjadi bentuk aslinya. Kesalahan pada salah satu tahap pengiriman, misalnya akibat gangguan sinyal atau kerusakan kabel, dapat menyebabkan data yang diterima tidak sesuai dengan data yang dikirim, sehingga diperlukan mekanisme deteksi dan koreksi kesalahan pada level protocol (Rajavarman dkk. 2026).

Kecepatan dan keandalan jaringan komputer juga dipengaruhi oleh media transmisi yang digunakan. Media kabel umumnya menawarkan kecepatan dan keandalan yang lebih tinggi dibandingkan media nirkabel, tetapi media nirkabel menawarkan fleksibilitas pemasangan yang jauh lebih besar, sehingga pemilihan media transmisi perlu disesuaikan dengan kebutuhan mobilitas dan anggaran yang tersedia (Kaur 2023; Narwal dkk. 2024).

Model TCP/IP yang banyak digunakan pada jaringan internet saat ini membagi proses komunikasi data ke dalam beberapa lapisan, yaitu lapisan aplikasi, lapisan transportasi, lapisan internet, dan lapisan akses jaringan. Setiap lapisan bekerja secara berurutan mengolah data dari aplikasi pengirim hingga siap dikirimkan melalui media fisik, kemudian diproses kembali secara berurutan terbalik oleh perangkat penerima hingga data dapat digunakan oleh aplikasi tujuan (Narwal dkk. 2024).

Pemahaman terhadap model berlapis ini penting bagi siapa saja yang ingin mendalami jaringan komputer, sebab hampir seluruh proses pemecahan masalah jaringan dilakukan dengan menelusuri lapisan demi lapisan untuk menemukan sumber gangguan. Kemampuan menelusuri lapisan komunikasi data secara

sistematis akan sangat membantu proses pemeliharaan dan perbaikan jaringan komputer di lingkungan kerja yang sesungguhnya (Jalaman dan Teleron 2024).

4. Topologi, Perangkat, dan Protokol Jaringan

Topologi jaringan menjelaskan susunan fisik maupun logis dari simpul serta jalur aliran data di dalam suatu jaringan komputer. Pemilihan topologi yang tepat memengaruhi keandalan, kemudahan pengelolaan, dan biaya pembangunan jaringan secara keseluruhan (Sankpal dkk. 2022).

Beberapa topologi yang umum dibahas dalam kajian jaringan komputer meliputi topologi bus, ring, star, mesh, tree, dan hybrid, dengan karakteristik serta kelebihan dan keterbatasan masing-masing seperti dirangkum pada tabel berikut (Sankpal dkk. 2022; Satria 2024).

Selain topologi, keberadaan perangkat jaringan menjadi unsur penting yang menentukan bagaimana data mengalir dari satu titik ke titik lain. Perangkat inti jaringan meliputi kartu antarmuka jaringan, perute, hub, sakelar, dan firewall. Kartu antarmuka jaringan memungkinkan suatu perangkat terhubung secara fisik ke jaringan, perute mengarahkan data antarjaringan yang berbeda, sedangkan hub dan sakelar berperan menghubungkan perangkat dalam satu jaringan lokal dengan cara kerja yang berbeda dari sisi efisiensi penyaluran data (Narwal dkk. 2024; Sankpal dkk. 2022).

Firewall berfungsi sebagai penyaring lalu lintas data yang masuk maupun keluar dari suatu jaringan berdasarkan aturan tertentu, sehingga menjadi salah satu perangkat penting bagi pengamanan jaringan sejak lapisan paling dasar. Peran firewall ini akan dibahas lebih

rinci pada bagian keamanan dasar sistem dan jaringan computer (Ogrizović dkk. 2024).

Komunikasi data pada jaringan komputer diatur oleh berbagai protokol dengan fungsi yang berbeda-beda. Protokol yang sering digunakan antara lain TCP/IP sebagai dasar komunikasi data di internet, UDP untuk pengiriman data yang mengutamakan kecepatan dibandingkan keandalan penuh, serta HTTP dan HTTPS untuk komunikasi data pada aplikasi berbasis web (Satria 2024).

Protokol TCP/IP bekerja dengan memastikan setiap paket data yang dikirim benar-benar diterima secara utuh dan berurutan, sehingga cocok digunakan untuk aplikasi yang menuntut keandalan tinggi, misalnya pengiriman berkas atau surat elektronik. Sebaliknya, protokol UDP tidak menjamin urutan maupun kelengkapan penerimaan data, tetapi menawarkan kecepatan yang lebih tinggi sehingga banyak digunakan pada aplikasi yang menuntut respons waktu nyata, seperti panggilan suara maupun permainan daring (Satria 2024).

Protokol HTTPS merupakan pengembangan dari protokol HTTP dengan tambahan lapisan enkripsi, sehingga data yang dipertukarkan antara peramban pengguna dan server tidak mudah disadap oleh pihak yang tidak berwenang. Penggunaan protokol yang tepat sesuai dengan kebutuhan aplikasi menjadi bagian penting dalam perancangan jaringan komputer yang efisien dan aman (Satria 2024).

Pemilihan topologi jaringan pada praktiknya juga mempertimbangkan skala organisasi dan tingkat toleransi terhadap gangguan. Organisasi berskala kecil dengan anggaran terbatas cenderung memilih topologi

yang sederhana seperti star, sedangkan organisasi berskala besar yang menuntut keandalan tinggi lebih memilih topologi mesh atau kombinasi hybrid meskipun biaya pembangunannya lebih besar (Sankpal dkk. 2022).

Selain perangkat inti yang telah disebutkan, jaringan komputer modern juga sering dilengkapi dengan titik akses nirkabel yang memungkinkan perangkat bergerak terhubung ke jaringan tanpa memerlukan kabel. Kehadiran titik akses nirkabel memperluas jangkauan jaringan sekaligus meningkatkan fleksibilitas penggunaan, tetapi juga menambah permukaan yang perlu diperhatikan dari sisi keamanan jaringan (Relan, Desai, dan Sharma 2026).

Pengaturan alamat pada jaringan komputer, misalnya melalui protokol pemberian alamat otomatis, memudahkan setiap perangkat baru yang terhubung ke jaringan untuk memperoleh identitas unik tanpa perlu dikonfigurasi secara manual oleh pengelola jaringan. Kemudahan ini sangat membantu pengelolaan jaringan berskala besar dengan jumlah perangkat yang terus bertambah dari waktu ke waktu (Roopesh 2024).

Perkembangan perangkat jaringan turut mendorong peningkatan kinerja dan keamanan jaringan komputer secara menyeluruh. Sakelar dan perute generasi terbaru umumnya sudah dilengkapi dengan kemampuan pengaturan lalu lintas data, pemantauan kinerja jaringan, dan fitur pengamanan tambahan, sehingga pengelola jaringan dapat memantau dan mengelola jaringan secara lebih menyeluruh dari satu titik kendali (Satria 2024).

Kinerja suatu jaringan komputer dapat diukur melalui beberapa indikator, misalnya kecepatan pengiriman data, keterlambatan pengiriman, serta

tingkat kehilangan data selama proses transmisi. Pemantauan terhadap indikator-indikator tersebut secara berkala membantu pengelola jaringan mengenali potensi masalah sejak dini sebelum berdampak signifikan terhadap layanan yang dirasakan oleh pengguna akhir (Satria 2024; Umar 2023).

5. Keamanan Dasar Sistem dan Jaringan Komputer

Keamanan sistem dan jaringan komputer berpusat pada tiga prinsip utama yang dikenal sebagai kerahasiaan, integritas, dan ketersediaan, yang sering dilengkapi dengan prinsip autentikasi dan nirsangkal. Kerahasiaan memastikan informasi hanya dapat diakses oleh pihak yang berwenang, integritas menjaga data tidak berubah tanpa izin, sedangkan ketersediaan memastikan sistem dan data dapat diakses kapan pun dibutuhkan oleh pengguna yang sah (Leidigh 2025; Rajavarman dkk. 2026).

Ancaman dasar yang paling sering dicatat dalam berbagai kajian keamanan mencakup perangkat lunak perusak, perangkat lunak tebusan, penipuan daring, serangan penolakan layanan, penyadapan data, pemalsuan identitas, serta serangan terhadap layanan jaringan lainnya. Setiap jenis ancaman memiliki cara kerja dan dampak yang berbeda, sehingga memerlukan strategi pencegahan yang disesuaikan dengan karakteristik masing-masing (Leidigh 2025; Al-Hazaimeh dkk. 2025).

Praktik dasar pengamanan menekankan pertahanan berlapis, yang mencakup penggunaan firewall, sistem deteksi dan pencegahan penyusupan, enkripsi data, jaringan privat maya, kontrol akses, segmentasi jaringan, serta audit keamanan secara berkala. Pertahanan berlapis dirancang agar apabila satu

lapisan pertahanan berhasil ditembus, lapisan lain masih dapat mencegah atau meminimalkan dampak serangan (Jia dkk. 2024; Leidigh 2025; Kaur 2023).

Firewall tetap menjadi komponen inti dalam pertahanan berlapis tersebut, dengan beberapa varian seperti penyaring paket, pemeriksa keadaan koneksi, proksi, dan firewall generasi lanjut, tetapi efektivitasnya sangat bergantung pada pembaruan, pemantauan, dan konfigurasi yang dilakukan secara benar oleh pengelola jaringan (Roopesh 2024).

Sistem deteksi dan pencegahan penyusupan berperan memantau lalu lintas jaringan secara berkelanjutan untuk mengenali pola aktivitas yang mencurigakan, kemudian memberikan peringatan atau bahkan menghentikan aktivitas tersebut secara otomatis sebelum menimbulkan kerugian yang lebih besar. Kombinasi antara firewall dan sistem deteksi penyusupan memberikan lapisan pertahanan yang saling melengkapi bagi suatu jaringan (Rajavarman dkk. 2026).

Enkripsi data menjadi salah satu praktik dasar pengamanan yang paling banyak diterapkan, baik untuk data yang sedang dikirim melalui jaringan maupun data yang tersimpan pada perangkat penyimpanan. Dengan enkripsi, data yang berhasil disadap atau dicuri oleh pihak yang tidak berwenang tidak akan dapat dibaca tanpa kunci yang sesuai, sehingga risiko kebocoran informasi dapat ditekan secara signifikan (Leidigh 2025).

Jaringan privat maya memungkinkan pengguna mengakses jaringan internal organisasi secara aman meskipun terhubung melalui jaringan publik yang tidak sepenuhnya dapat dipercaya. Teknologi ini banyak dimanfaatkan oleh organisasi yang memiliki karyawan bekerja dari lokasi yang berbeda-beda, sehingga tetap

dapat mengakses sumber daya internal tanpa mengorbankan aspek kerahasiaan data (Rajavarman et al., 2026).

Segmentasi jaringan membagi satu jaringan besar menjadi beberapa bagian yang lebih kecil dan terpisah, sehingga apabila terjadi gangguan keamanan pada satu segmen, dampaknya dapat dibatasi agar tidak menyebar ke seluruh jaringan. Pendekatan ini banyak diterapkan pada organisasi dengan data sensitif yang memerlukan tingkat perlindungan berbeda-beda antarbagian (Al-Hazaimeh dkk. 2025).

Audit keamanan secara berkala membantu organisasi menemukan celah keamanan yang mungkin belum diketahui sebelumnya, baik melalui pemeriksaan konfigurasi sistem, pengujian penetrasi, maupun evaluasi terhadap prosedur keamanan yang telah diterapkan. Hasil audit ini kemudian dijadikan dasar untuk memperbaiki kelemahan yang ditemukan sebelum dimanfaatkan oleh pihak yang tidak bertanggung jawab (Ogrizović dkk. 2024).

Keamanan tidak hanya berkaitan dengan pemasangan alat atau perangkat lunak pengaman, tetapi juga mencakup manajemen risiko yang berjalan sepanjang siklus hidup sistem, termasuk penilaian ancaman secara berkala, penyusunan prosedur tanggap insiden, dan edukasi berkelanjutan bagi pengguna sistem Edukasi pengguna menjadi unsur yang sering terabaikan, padahal banyak insiden keamanan justru terjadi akibat kelalaian manusia, misalnya mengeklik tautan yang berbahaya atau menggunakan kata sandi yang mudah ditebak (Leidigh 2025; Ogrizović dkk. 2024; Al-Hazaimeh dkk. 2025; Rajavarman dkk. 2026).

Secara keseluruhan, sistem operasi mengelola jalannya komputasi pada suatu perangkat, jaringan komputer memungkinkan terjadinya komunikasi antarperangkat, sedangkan keamanan dasar memastikan bahwa keduanya tetap dapat diandalkan, tersedia, dan terlindungi dari berbagai ancaman yang terus berkembang seiring kemajuan teknologi (Leidigh 2025).

Kesadaran keamanan perlu dibangun sejak tahap perancangan sistem, bukan hanya ditambahkan setelah sistem selesai dibangun. Pendekatan yang menempatkan pertimbangan keamanan sejak tahap awal perancangan terbukti lebih efektif dibandingkan upaya menambal celah keamanan setelah sistem beroperasi, sebab perbaikan pada tahap akhir sering kali memerlukan biaya dan waktu yang jauh lebih besar (Al-Hazaimeh dkk. 2025).

Ancaman keamanan juga terus berkembang mengikuti perkembangan teknologi, sehingga pengelola sistem dan jaringan perlu senantiasa memperbarui pengetahuan mengenai pola serangan terbaru serta memperbarui perangkat lunak dan konfigurasi keamanan secara berkala. Pembelajaran keamanan yang berkelanjutan menjadi kebutuhan mutlak bagi setiap organisasi yang mengandalkan sistem komputer dan jaringan dalam menjalankan aktivitasnya sehari-hari (Ogrizović dkk. 2024; Rajavarman dkk. 2026).

BAB V
TREN DAN TANTANGAN TEKNOLOGI MODERN

Perkembangan teknologi digital dalam dua dekade terakhir telah mengubah hampir seluruh aspek kehidupan manusia, mulai dari cara bekerja, berkomunikasi, hingga cara organisasi mengelola sumber dayanya. Perubahan tersebut tidak berdiri sendiri, melainkan digerakkan oleh konvergensi berbagai teknologi mutakhir, seperti Internet untuk Segala, komputasi awan, data besar, kecerdasan buatan, dan otomasi industri. Secara umum, kajian pustaka menempatkan Industri 4.0 sebagai kerangka integrasi teknologi digital lintas proses bisnis, kecerdasan buatan dan pembelajaran mesin sebagai mesin analitik utamanya, sedangkan keamanan dan etika menjadi batas utama yang menentukan keberlanjutan adopsinya. Bab ini membahas lima pokok bahasan, yaitu transformasi digital dan Industri 4.0, kecerdasan buatan dan pembelajaran mesin, konvergensi Internet untuk Segala, komputasi awan, dan data besar, keamanan siber dan etika teknologi informasi, serta prospek karier dan tantangan masa depan di bidang ilmu computer (Lu 2021; Yaqub dan Alsabban 2023; Sarker 2021; Ababneh dkk. 2025).

1. Transformasi Digital dan Industri 4.0

Transformasi digital merupakan proses perubahan mendasar dalam cara organisasi menciptakan nilai, mengelola operasi, dan berinteraksi dengan pemangku kepentingan melalui pemanfaatan teknologi digital

secara menyeluruh. Pada era Industri 4.0, proses tersebut bertumpu pada integrasi Internet untuk Segala, komputasi awan, data besar, kecerdasan buatan, dan otomasi, yang secara bersama-sama membentuk manufaktur cerdas, rantai nilai baru, serta model bisnis berbasis platform (Javaid dkk. 2024; Yaqub dan Alsabban 2023; Benitez, Ghezzi, dan Frank 2023).

Konsep Industri 4.0 pada mulanya dipahami sebagai revolusi industri keempat yang berpusat pada otomasi dan digitalisasi proses manufaktur, tetapi perkembangan mutakhir menunjukkan bahwa cakupannya telah meluas jauh melampaui lantai produksi. Gagasan ini kini memengaruhi strategi bisnis secara menyeluruh, mengubah pola integrasi rantai pasok, serta menuntut penciptaan nilai pelanggan yang lebih personal dan responsif terhadap data(Yaqub dan Alsabban 2023).

Digitalisasi ekonomi turut mendorong perubahan budaya kerja di lingkungan Industri 4.0, terutama pada aspek keterbukaan terhadap teknologi baru, kolaborasi lintas fungsi, dan pengambilan keputusan berbasis data. Penerapannya tidak lepas dari sejumlah kendala, seperti kesiapan infrastruktur, ketersediaan talenta digital, serta resistensi budaya organisasi yang masih terbiasa dengan proses konvensional (Javaid dkk. 2024).

Teknologi digital dalam kerangka Industri 4.0 juga dipahami sebagai penghubung batas antarorganisasi, bukan sekadar alat operasional internal. Ketika beberapa teknologi digital digabungkan secara terkoordinasi, teknologi tersebut dapat bertransformasi menjadi platform yang menghubungkan berbagai pemangku kepentingan, membentuk ekosistem bisnis baru, dan

membuka peluang model pendapatan yang sebelumnya tidak tersedia (Benitez, Ghezzi, dan Frank 2023).

Manufaktur cerdas menjadi salah satu wujud paling nyata dari transformasi digital, ditandai dengan penggunaan sensor, sistem siber-fisik, dan analitik data untuk memantau proses produksi secara langsung, memprediksi kebutuhan perawatan mesin, serta menyesuaikan kapasitas produksi secara adaptif terhadap permintaan pasar. Pendekatan ini memungkinkan efisiensi biaya sekaligus peningkatan kualitas produk secara bersamaan (Lu 2021).

Selain aspek produksi, transformasi digital turut mengubah pola interaksi antara perusahaan dan pelanggan melalui saluran digital yang lebih personal, seperti aplikasi seluler, platform layanan mandiri, dan sistem rekomendasi berbasis data perilaku pengguna. Perubahan ini menuntut organisasi untuk memiliki kapabilitas pengelolaan data yang matang agar mampu menerjemahkan data mentah menjadi wawasan yang dapat ditindaklanjuti (Yaqub dan Alsabban 2023; Javaid dkk. 2024).

Meskipun potensi manfaatnya besar, adopsi transformasi digital dan Industri 4.0 masih menghadapi tantangan struktural, termasuk kesenjangan kapabilitas teknologi antarorganisasi, tingginya biaya investasi awal, serta kebutuhan tata kelola perubahan yang menyeluruh agar transformasi tidak berhenti pada tahap uji coba semata. Keberhasilan transformasi sangat bergantung pada keselarasan antara strategi bisnis, kesiapan sumber daya manusia, dan arsitektur teknologi yang digunakan (Benitez, Ghezzi, dan Frank 2023; Akbari dkk. 2023).

Perusahaan yang berhasil melewati tahap uji coba menuju penerapan berskala penuh umumnya memiliki

peta jalan digital yang jelas, mencakup penentuan prioritas proses bisnis yang akan didigitalkan terlebih dahulu, penetapan indikator keberhasilan yang terukur, serta mekanisme evaluasi berkala terhadap dampak teknologi terhadap kinerja organisasi. Tanpa peta jalan yang jelas, investasi digital berisiko tersebar pada berbagai inisiatif yang tidak saling terhubung (Javaid dkk. 2024).

Kepemimpinan organisasi memegang peranan penting dalam mendorong keberhasilan transformasi digital, terutama dalam membangun budaya yang terbuka terhadap eksperimen teknologi dan toleran terhadap kegagalan pada tahap awal penerapan. Dukungan pimpinan puncak juga diperlukan untuk memastikan alokasi anggaran dan sumber daya manusia yang memadai bagi inisiatif digital jangka panjang (Yaqub dan Alsabban 2023).

Model bisnis berbasis platform yang tumbuh dari transformasi digital juga mengubah cara organisasi bersaing, karena keunggulan kompetitif tidak lagi semata ditentukan oleh kepemilikan aset fisik, melainkan oleh kemampuan mengelola data, membangun ekosistem mitra, dan menciptakan nilai tambah melalui layanan digital yang terus diperbarui. Pergeseran ini mendorong banyak organisasi tradisional untuk merancang ulang model pendapatannya secara mendasar (Benitez, Ghezzi, dan Frank 2023).

Pada tingkat rantai pasok, integrasi teknologi digital memungkinkan visibilitas yang lebih baik terhadap pergerakan barang dan bahan baku, mulai dari pemasok hingga konsumen akhir, sehingga organisasi dapat merespons gangguan pasokan secara lebih cepat dan akurat. Kemampuan ini menjadi semakin penting

seiring meningkatnya kompleksitas rantai pasok global dan ketidakpastian ekonomi (Lu 2021; Javaid dkk. 2024).

Perkembangan Industri 4.0 juga mendorong lahirnya konsep sistem siber-fisik, yaitu sistem yang menggabungkan komponen fisik mesin produksi dengan lapisan komputasi digital sehingga kedua sisi tersebut dapat saling bertukar informasi secara langsung dan berkelanjutan. Kemampuan ini menjadi dasar bagi terciptanya lini produksi yang dapat menyesuaikan dirinya sendiri terhadap perubahan kondisi operasional tanpa memerlukan intervensi manusia secara terus-menerus (Lu, 2021).

Perbedaan tingkat kesiapan digital antara organisasi berskala besar dan usaha kecil menengah turut menjadi sorotan dalam kajian transformasi digital, karena keterbatasan modal dan akses terhadap talenta digital sering kali menempatkan usaha kecil menengah pada posisi yang kurang menguntungkan dibandingkan organisasi besar dalam mengadopsi teknologi Industri 4.0 secara menyeluruh. Dukungan kebijakan publik dan program pendampingan menjadi salah satu cara untuk mempersempit kesenjangan tersebut (Javaid dkk. 2024; Yaqub dan Alsabban 2023).

Pengukuran keberhasilan transformasi digital idealnya tidak hanya berfokus pada indikator finansial jangka pendek, tetapi juga mempertimbangkan dampak jangka panjang terhadap ketahanan organisasi menghadapi gangguan eksternal, kepuasan pelanggan, serta kemampuan berinovasi secara berkelanjutan. Pendekatan pengukuran yang terlalu sempit berisiko mendorong organisasi mengambil keputusan investasi teknologi yang kurang tepat sasaran (Benitez, Ghezzi, dan Frank 2023; Lu 2021).

Interoperabilitas antarsistem lama dan sistem digital baru menjadi tantangan teknis yang sering muncul pada tahap awal transformasi, terutama bagi organisasi yang telah lama beroperasi dan memiliki infrastruktur teknologi informasi yang dibangun secara bertahap selama bertahun-tahun. Migrasi menuju arsitektur digital yang lebih terintegrasi umumnya memerlukan pendekatan bertahap agar operasional harian tidak terganggu secara signifikan (Keche 2025; Javaid dkk. 2024).

2. Kecerdasan Buatan (*Artificial Intelligence*) dan Machine Learning

Kecerdasan buatan dan pembelajaran mesin merupakan inti dari kemampuan analitik yang mengubah data besar menjadi prediksi, otomasi, serta dukungan pengambilan keputusan di berbagai sektor, termasuk kesehatan, keamanan siber, kota cerdas, perdagangan elektronik, dan manufaktur. Kedua bidang ini memungkinkan sistem komputer mempelajari pola dari data historis tanpa harus diprogram secara eksplisit untuk setiap kondisi yang mungkin terjadi (Mukhamediev dkk. 2022; Asha dkk. 2024; Sarker 2021).

Pembelajaran mesin secara umum dikelompokkan menjadi beberapa pendekatan, yaitu pembelajaran terarah, pembelajaran tidak terarah, pembelajaran semi terarah, pembelajaran penguatan, serta pembelajaran mendalam yang digunakan untuk menganalisis data dalam skala besar dan struktur yang kompleks. Setiap pendekatan memiliki karakteristik dan bidang penerapan yang berbeda, bergantung pada ketersediaan label data serta tujuan analisis yang ingin dicapai (Sarker 2021; Mishra 2024; Janiesch, Zschech, dan Heinrich 2021).

Pembelajaran terarah umumnya digunakan pada persoalan klasifikasi dan regresi, misalnya deteksi penipuan transaksi atau prediksi permintaan produk, karena model dilatih menggunakan data yang telah memiliki label keluaran yang jelas. Sebaliknya, pembelajaran tidak terarah lebih sesuai digunakan untuk menemukan pola tersembunyi dalam data tanpa label, seperti pengelompokan pelanggan berdasarkan perilaku belanja (Sarker 2021).

Pembelajaran mendalam, sebagai cabang khusus dari pembelajaran mesin, memanfaatkan jaringan saraf tiruan berlapis untuk memproses data tidak terstruktur seperti citra, suara, dan teks dengan tingkat akurasi yang terus meningkat. Kemampuan ini menjadikan pembelajaran mendalam sebagai fondasi bagi banyak aplikasi kecerdasan buatan generatif yang berkembang pesat pada saat ini (Janiesch, Zschech, dan Heinrich 2021; Mishra 2024).

Penerapan kecerdasan buatan telah meluas ke berbagai bidang praktis, di antaranya sistem pendukung diagnosis pada layanan kesehatan, sistem deteksi anomali pada keamanan siber, optimalisasi lalu lintas pada kota cerdas, sistem rekomendasi pada perdagangan elektronik, serta perawatan prediktif pada manufaktur. Keberagaman penerapan ini menunjukkan sifat kecerdasan buatan yang dapat digunakan lintas domain selama tersedia data yang memadai (Injadat dkk. 2021; Mishra 2024).

Meskipun manfaatnya luas, tantangan adopsi kecerdasan buatan tetap besar pada level teknologi, organisasi, sumber daya manusia, dan sosial, sehingga penerapannya secara luas masih belum merata di berbagai sektor dan wilayah. Pada level teknologi,

tantangan berkaitan dengan kualitas data, kebutuhan komputasi, serta interoperabilitas sistem lama dengan sistem berbasis kecerdasan buatan (Mukhamediev dkk. 2022; Janiesch, Zschech, dan Heinrich 2021).

Pada level organisasi dan sumber daya manusia, kesenjangan keterampilan analitik data serta kurangnya pemahaman manajerial terhadap potensi dan batasan kecerdasan buatan kerap menjadi penghambat utama implementasi secara menyeluruh. Sementara itu, pada level sosial, kekhawatiran terhadap penggantian tenaga kerja manusia dan ketidaksetaraan akses teknologi turut memengaruhi penerimaan publik terhadap sistem berbasis kecerdasan buatan (Mukhamediev dkk. 2022).

Kualitas data menjadi faktor penentu utama keberhasilan model pembelajaran mesin, karena model yang dilatih menggunakan data yang tidak lengkap, tidak seimbang, atau mengandung bias historis cenderung menghasilkan prediksi yang kurang akurat bahkan berpotensi memperkuat ketimpangan yang sudah ada dalam masyarakat. Oleh karena itu, tahap pembersihan dan validasi data menjadi bagian yang tidak dapat diabaikan dalam proses pengembangan model (Sarker 2021; Injadat dkk. 2021).

Kebutuhan komputasi yang besar, terutama pada pelatihan model pembelajaran mendalam berskala besar, turut menjadi pertimbangan penting bagi organisasi, baik dari sisi biaya infrastruktur maupun dampak konsumsi energi terhadap keberlanjutan lingkungan. Sejumlah organisasi kini mulai mempertimbangkan efisiensi komputasi sebagai kriteria tambahan dalam memilih arsitektur model kecerdasan buatan yang akan diterapkan (Mishra 2024).

Kemampuan menjelaskan hasil keluaran model kecerdasan buatan kepada pengguna nonteknis menjadi semakin penting, khususnya pada sektor yang berdampak langsung terhadap keselamatan dan hak individu, seperti diagnosis medis dan penilaian kelayakan kredit. Kurangnya interpretabilitas model dapat menurunkan kepercayaan pengguna meskipun akurasi model tersebut secara statistik tinggi (Mukhamediev dkk. 2022; Janiesch, Zschech, dan Heinrich 2021).

Integrasi kecerdasan buatan ke dalam proses bisnis yang sudah berjalan juga memerlukan penyesuaian pada alur kerja yang ada, termasuk pelatihan ulang tenaga kerja agar mampu bekerja berdampingan dengan sistem otomatis alih-alih tergantikan sepenuhnya olehnya. Pendekatan kolaboratif antara manusia dan mesin dipandang lebih realistis dibandingkan penggantian tenaga kerja secara total dalam jangka pendek (Injadat dkk. 2021).

Pembelajaran penguatan menonjol sebagai pendekatan yang memungkinkan sistem belajar melalui interaksi langsung dengan lingkungannya, di mana agen menerima umpan balik berupa penghargaan atau hukuman atas setiap tindakan yang diambil, sehingga pendekatan ini banyak diterapkan pada persoalan pengendalian robot, optimalisasi rute logistik, dan permainan strategi yang kompleks (Sarker 2021; Mishra 2024).

Riset di bidang kecerdasan buatan terus bergerak ke arah pengembangan model yang lebih hemat sumber daya, mengingat kebutuhan komputasi model berskala besar tidak selalu sebanding dengan peningkatan manfaat yang diperoleh, sehingga efisiensi algoritma

menjadi arah penelitian yang semakin banyak diminati selain sekadar peningkatan akurasi semata (Mishra 2024; Mukhamediev dkk. 2022).

Penerapan kecerdasan buatan pada sektor kesehatan menunjukkan potensi besar dalam mempercepat proses diagnosis dan personalisasi pengobatan, namun tetap memerlukan pengawasan tenaga medis yang kompeten agar keputusan akhir tidak sepenuhnya diserahkan kepada sistem otomatis tanpa pertimbangan konteks klinis pasien secara menyeluruh (Janiesch, Zschech, dan Heinrich 2021; Injadat dkk. 2021).

Pada sektor keamanan siber, kecerdasan buatan dimanfaatkan untuk mendeteksi pola lalu lintas jaringan yang mencurigakan secara langsung, sehingga organisasi dapat merespons ancaman lebih cepat dibandingkan mengandalkan pemeriksaan manual yang memerlukan waktu lebih lama. Kombinasi antara analitik otomatis dan penilaian pakar keamanan tetap diperlukan untuk mengurangi risiko kesalahan deteksi (Injadat dkk. 2021; Mishra 2024).

3. Internet of Things (IoT), Cloud Computing, dan Big Data

Konvergensi Internet untuk Segala, komputasi awan, dan data besar membentuk infrastruktur utama bagi pengumpulan data secara langsung, penyimpanan yang elastis, dan analitik prediktif pada lingkungan industri maupun layanan modern. Ketiga teknologi ini saling melengkapi, karena perangkat Internet untuk Segala menghasilkan data dalam volume besar yang kemudian memerlukan kapasitas penyimpanan dan pengolahan yang hanya dapat dipenuhi secara efisien

melalui komputasi awan (Alkateeb dan Abdullah 2024; Dritsas dan Trigka 2025; Araujo 2025).

Kajian terbaru menekankan pentingnya arsitektur hibrida, tepi jaringan, dan multi-awan, karena model komputasi awan yang tersentralisasi saja seringkali tidak cukup untuk memenuhi kebutuhan latensi rendah dan skala perangkat pada Internet untuk Segala Industri. Pemrosesan data di tepi jaringan memungkinkan sebagian analitik dilakukan lebih dekat dengan sumber data, sehingga mengurangi keterlambatan respons sistem (Asha dkk. 2024; Dritsas dan Trigka 2025).

Konektivitas jaringan generasi kelima, perangkat lunak berbasis jaringan yang dapat dikonfigurasi, dan pemotongan jaringan dipandang sebagai pendorong penting bagi konektivitas Internet untuk Segala Industri yang andal dan berlatensi rendah. Ketiganya memungkinkan penyediaan sumber daya jaringan yang disesuaikan dengan kebutuhan aplikasi tertentu, misalnya aplikasi kendali jarak jauh yang menuntut respons hampir seketika (Dritsas dan Trigka 2025).

Integrasi Internet untuk Segala, komputasi awan, dan data besar membuka peluang penerapan analitik prediktif, deteksi anomali, pemantauan langsung, serta pengambilan keputusan yang lebih cerdas pada berbagai sektor, termasuk manufaktur, energi, transportasi, dan pengelolaan lingkungan. Pemanfaatan data lingkungan secara langsung, misalnya, memungkinkan pemantauan kualitas udara dan air secara berkelanjutan untuk mendukung pengambilan kebijakan yang lebih responsif (Dritsas dan Trigka 2025; Asha dkk. 2024; Araujo 2025).

Penerapan komputasi awan bersama Internet untuk Segala juga mendorong inovasi dalam pengelolaan data dan penyampaian layanan, karena organisasi dapat

menskalakan sumber daya komputasi sesuai kebutuhan tanpa harus berinvestasi besar pada infrastruktur fisik. Pendekatan ini memberikan fleksibilitas operasional sekaligus efisiensi biaya bagi organisasi dengan beban kerja yang fluktuatif (Asha dkk. 2024).

Meskipun peluangnya besar, hambatan utama dalam penerapan konvergensi ini adalah keamanan data, biaya komputasi awan, interoperabilitas antarperangkat dan platform, keterbatasan analitik secara langsung, serta pengelolaan siklus hidup data mulai dari pengumpulan hingga penghapusan. Ketiadaan standar interoperabilitas yang seragam antarvendor turut memperumit integrasi sistem berskala besar (Dritsas dan Trigka 2025; Alkateeb dan Abdullah 2024).

Keberhasilan implementasi ekosistem Internet untuk Segala, komputasi awan, dan data besar sangat bergantung pada kemampuan organisasi merancang arsitektur yang sesuai dengan karakteristik beban kerja, kebutuhan latensi, serta sensitivitas data yang diproses, alih-alih menerapkan satu model arsitektur secara seragam untuk seluruh kasus penggunaan (Alkateeb dan Abdullah 2024; Araujo 2025).

Volume data yang dihasilkan oleh perangkat Internet untuk Segala tumbuh secara eksponensial seiring bertambahnya jumlah sensor yang terpasang pada lingkungan industri, sehingga organisasi memerlukan strategi pengelolaan data besar yang tidak hanya berfokus pada penyimpanan, tetapi juga pada pemilahan data yang benar-benar bernilai untuk dianalisis lebih lanjut. Tanpa strategi pemilahan yang tepat, biaya penyimpanan dapat meningkat tanpa memberikan nilai tambah yang sepadan (Dritsas dan Trigka 2025; Asha dkk. 2024; Araujo 2025).

Data besar yang dihasilkan dari lingkungan Internet untuk Segala Industri umumnya bersifat heterogen, mencakup data numerik dari sensor, data citra dari kamera pengawas, serta data teks dari log sistem, sehingga diperlukan kerangka analitik yang mampu mengolah berbagai jenis data tersebut secara terpadu. Kompleksitas ini menuntut kolaborasi erat antara ilmuwan data, insinyur jaringan, dan pakar domain industri (Dritsas dan Trigka 2025).

Keamanan pada lapisan perangkat Internet untuk Segala menjadi perhatian tersendiri, mengingat banyak perangkat tepi memiliki kapasitas komputasi terbatas sehingga sulit menjalankan mekanisme enkripsi yang kompleks. Kondisi ini menjadikan perangkat Internet untuk Segala sebagai salah satu titik rawan yang paling sering dieksploitasi dalam serangan siber terhadap infrastruktur industri (Dritsas dan Trigka 2025; Alkateeb dan Abdullah 2024).

Penerapan komputasi awan dan data besar pada sektor pengelolaan lingkungan menunjukkan bahwa integrasi teknologi ini tidak hanya relevan bagi sektor industri manufaktur, tetapi juga bagi upaya pemantauan sumber daya alam secara berkelanjutan, termasuk pengelolaan air, kualitas udara, dan mitigasi bencana berbasis data langsung (Araujo 2025).

Model penyampaian layanan komputasi awan yang beragam, mulai dari infrastruktur sebagai layanan, platform sebagai layanan, hingga perangkat lunak sebagai layanan, memberikan fleksibilitas bagi organisasi untuk memilih tingkat kendali dan tanggung jawab pengelolaan yang sesuai dengan kapasitas internal yang dimiliki. Pemilihan model yang tepat turut memengaruhi

efisiensi biaya serta kecepatan penyampaian layanan kepada pengguna akhir (Asha dkk. 2024).

Interoperabilitas antarplatform komputasi awan menjadi tantangan tersendiri ketika organisasi menerapkan strategi multi-awan untuk menghindari ketergantungan terhadap satu penyedia layanan tunggal, karena setiap penyedia memiliki antarmuka pemrograman dan format data yang berbeda, sehingga diperlukan lapisan integrasi tambahan agar data dapat berpindah secara lancar antarplatform (Dritsas dan Trigka 2025; Alkateeb dan Abdullah 2024).

Pengelolaan siklus hidup data, mulai dari tahap pengumpulan, penyimpanan, pengolahan, hingga penghapusan, perlu dirancang secara sistematis agar organisasi tidak menyimpan data yang sudah tidak relevan dalam jangka waktu yang terlalu lama, mengingat penyimpanan data yang berlebihan tidak hanya meningkatkan biaya tetapi juga memperbesar permukaan risiko keamanan apabila terjadi kebocoran data (Dritsas dan Trigka 2025; Alkateeb dan Abdullah 2024).

Sinergi antara Internet untuk Segala, komputasi awan, dan data besar pada akhirnya menuntut organisasi memandang ketiga teknologi tersebut sebagai satu kesatuan ekosistem yang saling bergantung, bukan sebagai komponen teknologi yang berdiri sendiri, karena kegagalan pada salah satu lapisan dapat memengaruhi kinerja keseluruhan sistem secara signifikan (Alkateeb dan Abdullah 2024; Araujo 2025).

4. Keamanan Siber dan Etika Teknologi Informasi

Keamanan siber dan etika teknologi informasi saat ini bukan lagi isu tambahan, melainkan syarat inti bagi keberlangsungan transformasi digital, karena persoalan

privasi, bias algoritmik, pengawasan, dan tata kelola data secara langsung memengaruhi tingkat kepercayaan publik serta efektivitas sistem yang dibangun (Keche 2025; Ababneh dkk. 2025; Shewale 2025).

Aspek etika keamanan siber mencakup pertimbangan tentang bagaimana data pribadi dikumpulkan, disimpan, dan digunakan oleh organisasi, termasuk batasan yang wajar antara kebutuhan keamanan dan hak privasi individu. Ketegangan antara kedua kepentingan ini kerap muncul ketika organisasi menerapkan pengawasan yang intensif dengan dalih mitigasi risiko keamanan (Ababneh dkk. 2025).

Perbandingan kerangka hukum privasi digital antarnegara menunjukkan bahwa pendekatan regulasi masih beragam, sehingga organisasi yang beroperasi lintas wilayah menghadapi kompleksitas kepatuhan tambahan. Perbedaan tingkat kematangan regulasi ini turut memengaruhi bagaimana hak privasi warga negara dilindungi dalam praktik pengelolaan data digital (Keche 2025).

Perkembangan kecerdasan buatan generatif turut memunculkan dimensi ancaman siber baru, karena teknologi yang sama dapat dimanfaatkan baik untuk memperkuat pertahanan maupun untuk melancarkan serangan yang lebih canggih. Kemunculan ancaman semacam ini menuntut pembaruan strategi keamanan yang sebelumnya berfokus pada ancaman konvensional (Gupta dkk. 2023).

Otomasi serangan pengelabuan, penyisipan perintah berbahaya pada sistem berbasis model bahasa, serta pembuatan konten tiruan yang meyakinkan menjadi beberapa bentuk risiko yang muncul akibat penyalahgunaan kecerdasan buatan generatif dalam

ranah keamanan siber. Situasi ini menuntut tata kelola etis yang lebih ketat terhadap pengembangan dan penggunaan model kecerdasan buatan (Gupta dkk. 2023).

Upaya menyeimbangkan kebutuhan keamanan dengan hak privasi individu memerlukan kerangka kerja yang menempatkan transparansi, akuntabilitas, dan persetujuan pengguna sebagai prinsip utama dalam setiap sistem yang mengumpulkan atau memproses data pribadi. Prinsip ini penting agar penguatan keamanan tidak dilakukan dengan mengorbankan hak dasar individu atas informasi pribadinya (Shewale 2025).

Pada akhirnya, keamanan siber dan etika teknologi informasi perlu dipandang sebagai satu kesatuan yang saling menopang, karena sistem yang aman secara teknis namun mengabaikan aspek etika berpotensi kehilangan kepercayaan publik, sementara sistem yang etis namun lemah secara keamanan tetap rentan terhadap penyalahgunaan (Ababneh dkk. 2025; Shewale 2025).

Tata kelola data yang baik menjadi prasyarat bagi terwujudnya keamanan siber dan etika teknologi informasi secara bersamaan, karena kebijakan yang jelas mengenai siapa yang berwenang mengakses data, untuk tujuan apa data tersebut digunakan, dan berapa lama data disimpan akan mengurangi risiko penyalahgunaan sekaligus memperjelas tanggung jawab organisasi apabila terjadi pelanggaran data (Ababneh dkk. 2025; Keche 2025).

Bias algoritmik menjadi salah satu perhatian etis yang semakin mengemuka seiring meluasnya penggunaan sistem otomatis dalam pengambilan keputusan yang berdampak langsung terhadap individu, misalnya dalam proses seleksi kerja atau persetujuan

pinjaman. Ketidaktransparanan proses pengambilan keputusan algoritmik dapat memperkuat diskriminasi yang sebelumnya sudah ada apabila tidak diawasi secara memadai (Keche 2025; Shewale 2025).

Pengawasan digital yang dilakukan oleh organisasi maupun pemerintah, meskipun bertujuan meningkatkan keamanan, tetap perlu dibatasi oleh prinsip proporsionalitas agar tidak melanggar hak dasar warga negara atas privasi. Kerangka hukum yang berbeda antarnegara turut memengaruhi sejauh mana batasan tersebut diterapkan secara konsisten dalam praktik (Keche 2025).

Kesiapan organisasi menghadapi ancaman siber berbasis kecerdasan buatan generatif juga menuntut investasi berkelanjutan pada pelatihan kesadaran keamanan bagi seluruh pengguna sistem, bukan hanya tim teknis, mengingat banyak insiden keamanan berawal dari kesalahan manusia dalam mengenali upaya rekayasa sosial yang semakin meyakinkan (Lu 2021; Ogunmola 2025).

Standar dan kerangka kerja keamanan siber internasional menjadi acuan penting bagi organisasi dalam menyusun kebijakan internal, meskipun penerapannya perlu disesuaikan dengan konteks operasional dan tingkat risiko masing-masing organisasi agar tidak hanya menjadi kepatuhan formal tanpa dampak nyata terhadap tingkat keamanan yang sesungguhnya (Ababneh dkk. 2025; Akbari dkk. 2023).

Etika dalam pengembangan teknologi informasi juga mencakup tanggung jawab pengembang terhadap dampak jangka panjang dari sistem yang dibangun, termasuk potensi penyalahgunaan di luar tujuan awal perancangan, sehingga pertimbangan etis idealnya sudah

menjadi bagian dari tahap perancangan sistem, bukan sekadar tambahan pada tahap akhir pengembangan (Keche 2025; Shewale 2025).

5. Prospek Karier dan Tantangan Masa Depan di Bidang Ilmu Komputer

Prospek karier di bidang ilmu komputer terus bergeser ke arah peran yang menggabungkan kemampuan teknis dengan kompetensi tata kelola, keamanan, pengelolaan data, dan kecerdasan buatan yang dapat dijelaskan secara transparan kepada pengguna maupun regulator (Mishra 2024; Bongomin dkk. 2025).

Peran rekayasawan kecerdasan buatan dan pembelajaran mesin akan semakin dituntut untuk tidak hanya mampu membangun model dengan akurasi tinggi, tetapi juga menjelaskan proses pengambilan keputusan model tersebut secara memadai kepada pemangku kepentingan yang tidak memiliki latar belakang teknis. Tuntutan interpretabilitas ini muncul seiring meningkatnya penggunaan kecerdasan buatan pada sektor yang berdampak langsung terhadap kehidupan masyarakat, seperti layanan kesehatan dan keuangan (Mishra 2024; Lu 2021).

Peran arsitek komputasi awan dan Internet untuk Segala menuntut kemampuan merancang sistem yang mampu menangani skala perangkat yang terus bertambah, sekaligus menjaga keseimbangan antara biaya operasional, keamanan data, dan kebutuhan pemrosesan di tepi jaringan untuk aplikasi yang sensitif terhadap latensi (Asha dkk. 2024).

Permintaan terhadap peran ilmuwan data juga terus meningkat seiring bertambahnya volume data yang dihasilkan organisasi dari berbagai saluran digital,

sehingga kemampuan mengolah data mentah menjadi wawasan bisnis yang dapat ditindaklanjuti menjadi keterampilan yang semakin bernilai di pasar tenaga kerja teknologi (Mishra 2024; Sarker 2021).

Peran analis keamanan siber menghadapi lanskap ancaman yang semakin dinamis akibat kemunculan kecerdasan buatan generatif, sehingga kompetensi yang dibutuhkan tidak lagi terbatas pada pertahanan teknis, tetapi juga mencakup pemahaman terhadap dimensi etis dari penggunaan kecerdasan buatan dalam operasi keamanan (Gupta dkk. 2023).

Teknologi kembaran digital berkembang sebagai pendukung penting bagi transisi menuju Industri 5.0 yang menekankan sistem industri berpusat pada manusia dan berketahanan tinggi, sehingga membuka peluang karier baru bagi spesialis yang mampu merancang, memvalidasi, dan mengamankan representasi virtual dari sistem fisik secara akurat (Bongomin dkk. 2025).

Selain empat area karier yang dirangkum pada tabel di atas, kebutuhan terhadap tenaga ahli tata kelola data juga tumbuh seiring meningkatnya regulasi perlindungan data di berbagai negara, sehingga organisasi memerlukan peran yang secara khusus memastikan kepatuhan pengelolaan data terhadap ketentuan hukum yang berlaku sekaligus menjembatani kebutuhan bisnis dengan batasan etis (Ababneh dkk. 2025; Keche 2025).

Pendidikan tinggi di bidang ilmu komputer dituntut menyesuaikan kurikulum agar lulusan tidak hanya menguasai keterampilan pemrograman dan analisis data, tetapi juga memiliki pemahaman dasar mengenai etika teknologi, tata kelola data, dan prinsip

keamanan siber sejak tahap awal pembelajaran. Penyesuaian ini penting agar lulusan siap menghadapi kompleksitas persoalan dunia kerja yang jarang dapat diselesaikan hanya dengan kemampuan teknis semata (Janiesch, Zschech, dan Heinrich 2021; Mishra 2024).

Kemampuan beradaptasi terhadap teknologi baru menjadi kompetensi yang tidak kalah penting dibandingkan penguasaan alat tertentu, mengingat siklus pembaruan teknologi di bidang kecerdasan buatan, komputasi awan, dan keamanan siber berlangsung sangat cepat, sehingga keterampilan yang relevan pada satu periode dapat menjadi usang dalam waktu relatif singkat. Pembelajaran berkelanjutan sepanjang karier menjadi keharusan, bukan lagi pilihan tambahan (Mishra 2024).

Kolaborasi lintas disiplin antara tenaga ahli teknis, pengambil kebijakan, dan pakar etika juga semakin diperlukan dalam merancang sistem berbasis kecerdasan buatan dan Internet untuk Segala yang bertanggung jawab, karena persoalan yang dihadapi tidak lagi dapat diselesaikan hanya dari sudut pandang teknis, melainkan memerlukan pertimbangan sosial, hukum, dan ekonomi secara bersamaan (Bongomin dkk. 2025; Ababneh dkk. 2025).

Pengalaman kerja lintas fungsi, misalnya keterlibatan pada proyek yang menggabungkan pengembangan perangkat lunak dengan pemahaman proses bisnis suatu industri tertentu, memberikan nilai tambah tersendiri bagi lulusan ilmu komputer, karena kemampuan menerjemahkan kebutuhan bisnis ke dalam solusi teknis yang tepat guna menjadi salah satu kompetensi yang paling dicari oleh organisasi pada era transformasi digital (Mishra 2024; Asha dkk. 2024).

Secara keseluruhan, tuntutan kompetensi masa depan di bidang ilmu komputer bergeser ke arah kemampuan lintas disiplin yang memadukan penguasaan teknis, pemahaman tata kelola data, kesadaran etika, serta kesiapan beradaptasi terhadap perubahan teknologi yang berlangsung cepat, mengingat setiap area karier yang dibahas menghadapi tantangan yang saling berkaitan dengan aspek keamanan dan keberlanjutan (Mishra 2024; Janiesch, Zschech, dan Heinrich 2021; Bongomin dkk. 2025).

DAFTAR PUSTAKA

Ababneh, Jafar, Majed Mohammed Alnemari, Hani Attar, Hani Alghamdi, Muath Al-Shawaheen, Balqies Saqarat, Laila Abu Romman, dan M Iqtait. 2025. "Cybersecurity Ethical Aspects (CEA)." *IEEE Access* 13:212443–68. https://doi.org/10.1109/access.2025.3641180.

Agarwal, Archi, Avni Agarwal, Deepak Kumar Verma, Deepti Tiwari, dan Rashmi Pandey. 2023. "A Review on Software Development Life Cycle." *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. https://doi.org/10.32628/cseit2390387.

Ahmad, Khlood, Mohamed Abdelrazek, Chetan Arora, Arbind Agrahari Baniya, Muneera Bano, dan John Grundy. 2023. "Requirements engineering framework for human-centered artificial intelligence software systems." *Appl. Soft Comput.* 143:110455. https://doi.org/10.1016/j.asoc.2023.110455.

Akbari, M, S Kok, J Hopkins, G Frederico, H Nguyen, dan A Alonso. 2023. "The changing landscape of digital transformation in supply chains: impacts of industry 4.0 in Vietnam." *The International Journal of Logistics Management*. https://doi.org/10.1108/ijlm-11-2022-0442.

Al-Hafiz, Ahmad, Yahfizham Yahfizham, Jl. Williem, dan V IskandarPsr. 2024. "Analisis Pengenalan Konsep

Algoritma Pemrograman Matematika Pada Kehidupan Sehari Hari." *Morfologi: Jurnal Ilmu Pendidikan, Bahasa, Sastra dan Budaya*. https://doi.org/10.61132/morfologi.v2i1.267.

Al-Hazaimeh, Obaida, Ashraf Abu-Ein, Younes, dan Tariq. 2025. "Cybersecurity Foundations: Understanding Computer, Network, and Internet Security." *Engineering World*. https://doi.org/10.37394/232025.2025.7.12.

Alkateeb, Zeena, dan Dhuha Abdullah. 2024. "Unlocking the Potential: Synergizing IoT, Cloud Computing, and Big Data for a Bright Future." *Iraqi Journal For Computer Science and Mathematics*. https://doi.org/10.52866/ijcsm.2024.05.03.001.

Anggreani, Selviana, dan Yahfizham Yahfizham. 2024. "Pengantar dan Pengenalan Konsep Dasar Algoritma Pemrograman." *Pendekar : Jurnal Pendidikan Berkarakter*. https://doi.org/10.51903/pendekar.v2i1.599.

Anwar, Chairul. 2026. "Implementation of Information System and Software Quality Testing in Company Operational Applications Based on ISO/IEC 25010 (Case Study: PT Snapdev Digital Indonesia)." *Jurnal Teknologi Informatika dan Komputer*. https://doi.org/10.37012/jtik.v12i1.3294.

Araujo, Sufola Das Chagas Silva. 2025. "BIG DATA ANALYTICS BASED ON THE INTERNET OF THINGS AND CLOUD COMPUTING INTEGRATED FOR

EVOLVING ENVIRONMENTAL DATA MANAGEMENT." *International Journal of Applied Mathematics*. https://doi.org/10.12732/ijam.v38i5.1116.

Asha, A, L Rajeshkumar, V Pandi, Shobana D, L J, dan M Dayanidhy. 2024. "Implementing Cloud Computing with Internet of Things (IoT) technologies: Novel approaches to data management and service delivery innovation." In *2024 Global Conference on Communications and Information Technologies (GCCIT)*, 1–6. https://doi.org/10.1109/gccit63234.2024.10862745.

Balreira, D, T Silveira, dan Juliano Araujo Wickboldt. 2022. "Investigating the impact of adopting Python and C languages for introductory engineering programming courses." *Computer Applications in Engineering Education* 31:47–62. https://doi.org/10.1002/cae.22570.

Belmar, Héctor. 2022. "Review on the teaching of programming and computational thinking in the world" 4. https://doi.org/10.3389/fcomp.2022.997222.

Benitez, G, A Ghezzi, dan A Frank. 2023. "When technologies become Industry 4.0 platforms: Defining the role of digital technologies through a boundary-spanning perspective." *International Journal of Production Economics*. https://doi.org/10.1016/j.ijpe.2023.108858.

Berawi, M, M Sari, dan P Miraj. 2024. "Citizen and Technology: The Core in Developing Human-Centric Smart Cities." *CSID Journal of Infrastructure Development*. https://doi.org/10.7454/jid.v7.i3.1185.

Berrezueta-Guzman, Santiago, Vanesa Metaj, dan Stefan Wagner. 2026. "Beyond the Code: A Multi-Modal Assessment Strategy for Fostering Professional Competencies via Introductory Programming Projects." *ArXiv* abs/2603.1. https://doi.org/10.48550/arxiv.2603.18741.

Bongomin, Ocident, M Mwape, N Mpofu, Brendah Kembabazi Bahunde, Richard Kidega, Ibrahim Luqman Mpungu, Godias Tumusiime, dkk. 2025. "Digital Twin technology advancing Industry 4.0 and Industry 5.0 across Sectors." *Results in Engineering*. https://doi.org/10.1016/j.rineng.2025.105583.

Chazette, Larissa, Wasja Brunotte, dan Timo Speith. 2022. "Explainable software systems: from requirements analysis to system evaluation." *Requirements Engineering* 27:457–87. https://doi.org/10.1007/s00766-022-00393-5.

Chen, Chih-Hung, dan Hsiang-Yu Chung. 2023. "Fostering Computational Thinking and Problem-Solving in Programming: Integrating Concept Maps Into Robot Block-Based Programming." *Journal of Educational Computing Research* 62:406–27. https://doi.org/10.1177/07356331231205052.

Chen, Si. 2024. "An Introduction to Computational Science and Computer Development." *Studies in Social Science Research*. https://doi.org/10.22158/sssr.v5n1p135.

Cook, Devin. 2025. "Flowgorithm: Principles for Teaching Introductory Programming Using Flowcharts." *2015 Pacific Southwest Section Meeting Proceedings*. https://doi.org/10.18260/1-2-1153-54443.

Dontulwar, Avni. 2024. "Computers in the Modern Era: Applications and Innovations." *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. https://doi.org/10.32628/cseit24106188.

Dritsas, Elias, dan M Trigka. 2025. "A Survey on the Applications of Cloud Computing in the Industrial Internet of Things." *Big Data Cogn. Comput.* 9:44. https://doi.org/10.3390/bdcc9020044.

Fang, Jianfeng, Danwei Shao, Gwo-Jen Hwang, dan Shao-Chen Chang. 2022. "From Critique to Computational Thinking: A Peer-Assessment-Supported Problem Identification, Flow Definition, Coding, and Testing Approach for Computer Programming Instruction." *Journal of Educational Computing Research* 60:1301–24. https://doi.org/10.1177/07356331211060470.

Ghumatkar, Rutuja, dan Anup Date. 2023. "Software Development Life Cycle (SDLC)." *International Journal for Research in Applied Science and*

Engineering Technology. https://doi.org/10.22214/ijraset.2023.56554.

Gill, S, Huaming Wu, Panos Patros, Carlo Ottaviani, Priyansh Arora, Víctor Casamayor-Pujol, David Haunschild, dkk. 2024. "Modern Computing: Vision and Challenges." *ArXiv* abs/2401.0. https://doi.org/10.1016/j.teler.2024.100116.

Gulácsi, Ádám, dan M Csernoch. 2026. "Removing the burden of syntax: developing computational thinking and algorithmic skills of STEM students." *Teaching Mathematics and Computer Science*. https://doi.org/10.5485/tmcs.2026.15760.

Gupta, Maanak, Charankumar Akiri, Kshitiz Aryal, Elisabeth Parker, dan Lopamudra Praharaj. 2023. "From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy." *IEEE Access* 11:80218–45. https://doi.org/10.1109/access.2023.3300381.

Injadat, Mohammadnoor, Abdallah Moubayed, Ali Bou Nassif, dan A Shami. 2021. "Machine learning towards intelligent systems: applications, challenges, and opportunities." *Artificial Intelligence Review* 54:3299–3348. https://doi.org/10.1007/s10462-020-09948-w.

Insuasti, Jesús, Edwin Insuasty, dan Alexander Barón. 2025. "Didactics of the Fundamentals of Computer Programming with a Computational Thinking Approach Based on Turtle Graphs." *Academia y*

Virtualidad. https://doi.org/10.18359/ravi.7533.

Jalaman, John Ric, dan Jerry Teleron. 2024. "Optimizing Operating System Performance through Advanced Memory Management Techniques: A Comprehensive Study and Implementation." *Engineering and Technology Journal*. https://doi.org/10.47191/etj/v9i05.33.

Janiesch, Christian, Patrick Zschech, dan K Heinrich. 2021. "Machine learning and deep learning." *Electronic Markets* 31:685–95. https://doi.org/10.1007/s12525-021-00475-2.

Javaid, M, Abid Haleem, R Singh, dan A Sinha. 2024. "Digital economy to improve the culture of industry 4.0: A study on features, implementation and challenges." *Green Technologies and Sustainability*. https://doi.org/10.1016/j.grets.2024.100083.

Jia, Shian, Xinbo Wang, Mingli Song, dan Gang Chen. 2024. "Agent Centric Operating System - a Comprehensive Review and Outlook for Operating System." *ArXiv* abs/2411.1. https://doi.org/10.48550/arxiv.2411.17710.

Kaur, Apleen. 2023. "A Research on Computer Networking: Protocol, Challenges & Applications." *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. https://doi.org/10.55041/ijsrem18206.

Keche, Urwi. 2025. "Cybersecurity and Information

Ethics: A Comparative Study of Privacy Rights in Digital India and the Global Framework." *Gurukul International Multidisciplinary Research Journal*. https://doi.org/10.69758/gimrj/2504i5vxiiip0090 .

Khaleel, Mohamed, Abdullatif Jebrel, dan Dunia Shwehdy. 2024. "Artificial Intelligence in Computer Science." *International Journal of Electrical Engineering and Sustainability*. https://doi.org/10.65998/ijees.v2i2.80.

Kravchuk, O, Natalia Synyuk, dan D Kravchuk. 2025. "REVIEW AND ANALYSIS OF DATA PROCESSING INFORMATION TECHNOLOGIES IN THE COURSE OF MODERN COMPUTER SCIENCE." *Herald of Khmelnytskyi National University. Technical sciences*. https://doi.org/10.31891/2307-5732-2025-349-75.

Kumar.B, Sathish. 2025. "Exploring the Scope and Evolution of Computer Science." *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. https://doi.org/10.55041/ijsrem45123.

Kumar, Amruth, dan R Raj. 2024. "Computer Science Curricula 2023 (CS2023): The Final Report." In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. https://doi.org/10.1145/3626253.3633405.

Kumar, Dr.S.Arumugam Satheesh, dan D Bawankar.

2025. “Advanced Requirements Engineering and Use Case Analysis for Software Systems.” *International Academic Journal of Innovative Research*. https://doi.org/10.71086/iajir/v12i4/iajir1232.

Kumar, Ravendra, Ravindra Rawat, V Prajapat, dan Vivek Choudhary. 2023. “SOFTWARE ENGINEERING AND PROGRAMMING.” *Journal of Nonlinear Analysis and Optimization*. https://doi.org/10.36893/jnao.2023.v14i1.102-108.

Kusnadi, Kusnadi, Muhammad Hatta, Goenawan Brotosaputro, Amri Amri, dan Sebastian Harris. 2025. “Information Technology and Its Impact on Modern Classroom Dynamics: A Computer Science Perspective.” *Aptisi Transactions on Technopreneurship (ATT)*. https://doi.org/10.34306/att.v7i1.523.

Laplante, P, dan Mohamad Kassab. 2022. “Requirements Engineering for Software and Systems.” https://doi.org/10.1201/9781003129509.

Leidigh, Christopher. 2025. “Fundamental Principles of Network Security.” *IJARCCE*. https://doi.org/10.17148/ijarcce.2025.144109.

Lidjali, Nazwan, D Pongoh, Aldrin Tongkotou, Nadine Sondak, Sebastian Mewaikere, dan Prisilia Baduang. 2025. “The Role of Information Science in the Development of Science and Technology.” *Jurnal*

Syntax Admiration. https://doi.org/10.46799/jsa.v5i12.1977.

Lindsay, Dominic, S Gill, Daria Smirnova, dan Peter Garraghan. 2021. "The evolution of distributed computing systems: from fundamental to new frontiers." *Computing* 103:1859–78. https://doi.org/10.1007/s00607-020-00900-y.

Lu, Yang. 2021. "The Current Status and Developing Trends of Industry 4.0: a Review." *Information Systems Frontiers* 27:215–34. https://doi.org/10.1007/s10796-021-10221-w.

Malik, S, R Mathew, Abir Al-Sideiri, J Jabbar, Rim Al-Nuaimi, dan R Tawafak. 2021. "Enhancing problem-solving skills of novice programmers in an introductory programming course." *Computer Applications in Engineering Education* 30:174–94. https://doi.org/10.1002/cae.22450.

Marín-Marín, José-Antonio, P García-Tudela, dan Pablo Dúo-Terrón. 2024. "Computational thinking and programming with Arduino in education: A systematic review for secondary education." *Heliyon* 10. https://doi.org/10.1016/j.heliyon.2024.e29177.

Marlowe, Thomas, Cyril Ku, Joseph Laracy, V Kirova, dan Katherine Herbert. 2026. "A Systemic View of a Software Engineering Education Curriculum: Requirements and Guidelines in the Era of Generative AI." *Journal of Integrated Design and*

Process Science 29:108–33. https://doi.org/10.1177/10920617251405471.

Mishra, Akanksha. 2024. "A Comprehensive Review of Artificial Intelligence and Machine Learning : Concepts, Trends, and Applications." *International Journal of Scientific Research in Science and Technology*. https://doi.org/10.32628/ijsrst2411587.

Mukhamediev, R, Yelena Popova, Y Kuchin, E Zaitseva, A Kalimoldayev, A Symagulov, V Levashenko, dkk. 2022. "Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges." *Mathematics*. https://doi.org/10.3390/math10152552.

Munasinghe, Bhagya, Tim Bell, dan Anthony Robins. 2023. "Computational Thinking and Notional Machines: The Missing Link." *ACM Transactions on Computing Education* 23:1–27. https://doi.org/10.1145/3627829.

Narwal, Renu, Nitish Panwar, Yadav, dan Pankaj. 2024. "Computer Networking: Fundamentals, Models, and Security Considerations." *International Journal of Advanced Research in Science, Communication and Technology*. https://doi.org/10.48175/ijarsct-17621.

Ngadengon, Zuraihah, Tamil Selvan Subramaniam, Zurina Yasak, Noor Azmiza Ideris, dan Zuraidah Mohd Ramly. 2025. "Evaluating the Difficulties in

Programming Learning: Insights from Polytechnic Students." *International Journal of Academic Research in Progressive Education and Development*. https://doi.org/10.6007/ijarped/v14-i1/24518.

Nugroho, Fathoni, dan Hapzi Ali. 2022. "DETERMINASI SIMRS: HARDWARE, SOFTWARE DAN BRAINWARE (LITERATURE REVIEW EXECUTIVE SUPPORT SISTEM (ESS) FOR BUSINESS)." *JURNAL MANAJEMEN PENDIDIKAN DAN ILMU SOSIAL*. https://doi.org/10.38035/jmpis.v3i1.871.

Odun-Ayo, Isaac, K Okokpujie, Kingsley Oputa, H Ogbu, Franklin Emmanuel, A Shofadekan, dan Gideon Okuazun. 2021. "Comparative Study of Operating System Quality Attributes." In *IOP Conference Series: Materials Science and Engineering*. Vol. 1107. https://doi.org/10.1088/1757-899x/1107/1/012061.

Ogrizović, Mihajlo, Pavle Vuletic, Stanisavljević, dan Žarko. 2024. "Advanced Network and System Security Teaching." *Electronics*. https://doi.org/10.3390/electronics14010003.

Ogunmola, G. 2025. "Securing the Digital Economy: Innovations in Cybersecurity, Ethical AI and the Future of Digital Payments and Cryptocurrency." *Abhigyan* 43:385–402. https://doi.org/10.1177/09702385251377136.

Ozkaya, Ipek. 2021. "Can We Really Achieve Software Quality?" *IEEE Softw.* 38:3–6.

https://doi.org/10.1109/ms.2021.3060552.

Pargaonkar, Shravan. 2023a. “A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering.” *International Journal of Scientific and Research Publications*. https://doi.org/10.29322/ijsrp.13.08.2023.p14015.

———. 2023b. “Synergizing Requirements Engineering and Quality Assurance: A Comprehensive Exploration in Software Quality Engineering.” *International Journal of Science and Research (IJSR)*. https://doi.org/10.21275/sr23822112511.

Petrizia, Y, R Sujitha, dan Sree Saradha. 2025. “Overview of Process Management in Operating Systems.” *International Journal of Advanced Research in Science, Communication and Technology*. https://doi.org/10.48175/ijarsct-23314.

Qiu, Xiong, Lianzhi Shi, dan Yurui Qiu. 2023. “Application of software engineering methods in computer software development.” *Advances in Computer, Signals and Systems*. https://doi.org/10.23977/acss.2023.070410.

Rajavarman, V, T Kumanan, S Karthick, dan G Senthilvelan. 2026. *Fundamentals of Cyber Security*. https://doi.org/10.47715/978-93-86388-76-6.

Rajkumar, S, D J Sujay, R Sangeetha, G Sudhakar, K Muralikrishna, dan D Kumaraswamy. 2024. "An Overview of Computer Science and Engineering and Its Latest Technologies." *International Research Journal on Advanced Engineering Hub (IRJAEH)*. https://doi.org/10.47392/irjaeh.2024.0099.

Ramachandrappa, Naveen Chikkanayakanahalli. 2024. "SOLID Design Principles in Software Engineering." *International Journal of Computer Trends and Technology*. https://doi.org/10.14445/22312803/ijctt-v72i9p104.

Ramadhona, Nova, Ita Erliyani, dan Rosdiana. 2023. "Exploring the Components of Management Information Systems: Software, Database, and Brainware - A Literature Review." *International Transactions on Education Technology (ITEE)*. https://doi.org/10.33050/itee.v2i1.416.

Ranjan, Raju. 2024. "A Review Paper on Software Testing." *International Research Journal of Modernization in Engineering Technology and Science*. https://doi.org/10.56726/irjmets50146.

Relan, Archita, D Desai, dan Durgesh Sharma. 2026. "A Comprehensive Review of Kernel Management in Operating Systems." In *2026 6th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, 1–6. https://doi.org/10.1109/icnte66387.2026.11437502.

Roopesh, M. 2024. “CYBERSECURITY SOLUTIONS AND PRACTICES: FIREWALLS, INTRUSION DETECTION/PREVENTION, ENCRYPTION, MULTI-FACTOR AUTHENTICATION.” *ACADEMIC JOURNAL ON BUSINESS ADMINISTRATION, INNOVATION & SUSTAINABILITY*. https://doi.org/10.69593/ajbais.v4i3.90.

Sankpal, Mrunali Nagesh, Pratiksha Vijay Chinchwade, Rutuja Rajkumar Kognole, dan K Rode. 2022. “Computer Network Topologies.” *International Journal for Research in Applied Science and Engineering Technology*. https://doi.org/10.22214/ijraset.2022.41553.

Sarker, Iqbal. 2021. “Machine Learning: Algorithms, Real-World Applications and Research Directions.” *Sn Computer Science* 2. https://doi.org/10.1007/s42979-021-00592-x.

Satria, Welnof. 2024. “Analysis of Performance and Security in Modern Computer Networks.” *Dharmawangsa: International Journal of the Social Sciences, Education and Humanitis*. https://doi.org/10.46576/ijsseh.v5i2.4683.

Sharjeel, M, Shehroz Tariq, Mansoor Ahmad, A Aslam, Iltaf Iltaf, Laiba Khawaja, Asma Jameel, Sayed Alamgir Shah, Muneeb Ahmad, dan Waheed Khan. 2025. “An Interdisciplinary Review of Modern Computer Science: Trends and Challenges in AI, Cybersecurity, Cloud, Blockchain, IoT, Data Science, NLP, Vision, Software Engineering, and Quantum

Computing." *Scholars Journal of Engineering and Technology*. https://doi.org/10.36347/sjet.2025.v13i09.005.

Shewale, Vilas. 2025. "The Ethics of Cybersecurity: Balancing Security and Privacy in the Digital Age." *European Journal of Computer Science and Information Technology*. https://doi.org/10.37745/ejcsit.2013/vol13n151120.

Shon, Ayeong. 2024. "Beyond the Screen: Real-World Impacts of Computer Science Innovations." *International Journal of Scientific Research and Management (IJSRM)*. https://doi.org/10.18535/ijsrm/v12i10.ec07.

Shukla, Surendra Kumar, V Gupta, R Rambola, Jaishree Meena, Anupriya, dan Monika Bharti. 2024. "A Systematic Literature Review for Implementing Robust Quality Assurance Strategies Across the Software Development Life Cycle." In *2024 International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit)*, 711–15. https://doi.org/10.1109/globalaisummit62156.2024.10947901.

Sisodia, Jyoti, dan Dr.Suraj Pote. 2024. "Discussing Phases and Models of Software Development Life Cycle." *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. https://doi.org/10.55041/ijsrem28176.

Subahi, Ahmad. 2023. "BERT-Based Approach for Greening Software Requirements Engineering Through Non-Functional Requirements." *IEEE Access* 11:103001–13. https://doi.org/10.1109/access.2023.3317798.

Taha, T, Azeez Abdullah Barzinjy, Faiq Hussain, dan Togzhan Nurtayeva. 2022. "Nanotechnology and computer science: Trends and advances." *Memories - Materials, Devices, Circuits and Systems*. https://doi.org/10.1016/j.memori.2022.100011.

Tarsini, Iin, dan Riska Anggraeni. 2024. "Explore flowchart and pseudocode concepts in algorithms and programming." *Indonesian Journal of Multidisciplinary Science*. https://doi.org/10.55324/ijoms.v3i5.807.

Umar, A. 2023. "A Comparative Study of Modern Operating Systems in terms of Memory and Security: A Case Study of Windows, iOS, and Android." *SLU Journal of Science and Technology*. https://doi.org/10.56471/slujst.v6i.353.

Wang, Xin, Feixue Wan, dan Jiamin Dai. 2026. "Enhancing computational thinking through coding education in primary school students: an experimental study on the impact of early programming exposure on problem-solving skills." *Frontiers in Psychology* 17. https://doi.org/10.3389/fpsyg.2026.1734482.

White, Jules, Sam Hays, Quchen Fu, Jesse Spencer-Smith, dan Douglas Schmidt. 2023. "ChatGPT Prompt

Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design." *ArXiv* abs/2303.0. https://doi.org/10.48550/arxiv.2303.07839.

Wu, Ting-Ting, A Asmara, Yueh-Min Huang, dan Intan Permata Hapsari. 2024. "Identification of Problem-Solving Techniques in Computational Thinking Studies: Systematic Literature Review." *SAGE Open* 14. https://doi.org/10.1177/21582440241249897.

Wu, Ting-Ting, Chia-Ju Lin, Shih-Cheng Wang, dan Yueh-Min Huang. 2023. "Tracking Visual Programming Language-Based Learning Progress for Computational Thinking Education." *Sustainability*. https://doi.org/10.3390/su15031983.

Yaqub, M, dan Abdullah Alsabban. 2023. "Industry-4.0-Enabled Digital Transformation: Prospects, Instruments, Challenges, and Implications for Business Strategies." *Sustainability*. https://doi.org/10.3390/su15118553.

Yuen, K, Dennis Liu, dan H Leong. 2023. "Competitive programming in computational thinking and problem solving education." *Computer Applications in Engineering Education* 31:850–66. https://doi.org/10.1002/cae.22610.

Zhang, Jia-Hua, Bin Meng, Liu-Cong Zou, Yue Zhu, dan Gwo-Jen Hwang. 2021. "Progressive flowchart development scaffolding to improve university

students' computational thinking and programming self-efficacy." *Interactive Learning Environments* 31:3792–3809. https://doi.org/10.1080/10494820.2021.1943687.

Zhao, Chenxiao, Yu Wang, dan Jingjing Jiang. 2023. "Application of Software Engineering Technology in System Software Development." In *2023 Asia-Europe Conference on Electronics, Data Processing and Informatics (ACEDPI)*, 502–6. https://doi.org/10.1109/acedpi58926.2023.00101.