ISBN 978-623-125-403-0

PENDEKATAN MODERN DALAM ANALISIS DAN DESAIN TEKNOLOGI INFORMASI

Penulis :
Aisyah Mutia Dawis
Devi Rahmayanti
Taufik Rachman
Ali Impron
Yoseph Pius Kurniawan Kelen

PENDEKATAN MODERN DALAM ANALISIS DAN DESAIN TEKNOLOGI INFORMASI

Aisyah Mutia Dawis
Devi Rahmayanti
Taufik Rachman
Ali Impron
Yoseph Pius Kurniawan Kelen

GET PRESS INDONESIA

PENDEKATAN MODERN DALAM ANALISIS DAN DESAIN TEKNOLOGI INFORMASI

Penulis :
Aisyah Mutia Dawis
Devi Rahmayanti
Taufik Rachman
Ali Impron
Yoseph Pius Kurniawan Kelen

ISBN : 978-623-125-601-0

Editor : Mila Sari., S.ST, M.Si
Penyunting : Rantika Maida Sahara., S.Tr.Kes
Desain Sampul dan Tata Letak : Atyka Trianisa, S.Pd

Penerbit : GET PRESS INDONESIA
Anggota IKAPI No. 033/SBA/2022

Redaksi :
Jln. Palarik Air Pacah No 26 Kel. Air Pacah
Kec. Koto Tangah Kota Padang Sumatera Barat
Website : www. getpress.co.id
Email : adm.getpress@gmail.com

Cetakan pertama, Januari 2025

KATA PENGANTAR

Segala Puji dan syukur atas kehadirat Allah SWT dalam segala kesempatan. Sholawat beriring salam dan doa kita sampaikan kepada Nabi Muhammad SAW. Alhamdulillah atas Rahmat dan Karunia-Nya penulis telah menyelesaikan Buku Pendekatan Modern Dalam Analisis Dan Desain Teknologi Informasi ini.

Buku Ini Membahas Pengantar Analisis Dan Desain Teknologi Informasi, Metodologi Dan Kerangka Kerja Dalam Analisis TI, Desain Sistem Yang Berfokus Pada Pengguna, Teknologi Modern Dalam Desain Sistem TI, Keamanan Dan Privasi Dalam Desain TI.

Proses penulisan buku ini berhasil diselesaikan atas kerjasama tim penulis. Demi kualitas yang lebih baik dan kepuasan para pembaca, saran dan masukan yang membangun dari pembaca sangat kami harapkan.

Penulis ucapkan terima kasih kepada semua pihak yang telah mendukung dalam penyelesaian buku ini. Terutama pihak yang telah membantu terbitnya buku ini dan telah mempercayakan mendorong, dan menginisiasi terbitnya buku ini. Semoga buku ini dapat bermanfaat bagi masyarakat Indonesia.

Padang, Desember 2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB 1
PENGANTAR ANALISIS DAN DESAIN TEKNOLOGI INFORMASI

1.1 Pentingnya Analisis dan Desain Teknologi Informasi dalam Era Digital

Perkembangan teknologi informasi yang pesat membawa sejumlah tantangan, seperti keamanan data, privasi, dan kompleksitas sistem. Serangan siber yang semakin canggih, regulasi data yang ketat, serta tuntutan pengguna akan pengalaman yang lebih personal dan aman semakin menyulitkan pengelolaan sistem informasi. Namun, di balik tantangan tersebut, teknologi juga membuka peluang yang sangat besar (Fitriana, Permanasari and Sanjaya, 2023). Analisis dan desain sistem informasi yang tepat dapat menjadi kunci bagi organisasi untuk mengatasi tantangan dan memanfaatkan peluang yang ada.

Dengan merancang sistem yang aman, andal, dan fleksibel, organisasi dapat tumbuh dan berkembang dalam era digital yang dinamis. Analisis yang mendalam terhadap data yang dihasilkan oleh sistem memungkinkan organisasi untuk mengidentifikasi tren, membuat keputusan yang lebih baik, dan meningkatkan efisiensi operasional (H. and Irbayuni, 2023). Selain itu, desain sistem yang berfokus pada pengguna dapat meningkatkan kepuasan pelanggan, loyalitas merek, dan daya saing organisasi. Salah satu tantangan terbesar yang dihadapi saat ini adalah bagaimana mengintegrasikan berbagai teknologi baru seperti kecerdasan buatan (AI), *Internet of Things* (IoT) (Dawis, Murhadi and Ardhani, 2023), dan big data ke dalam sistem informasi yang sudah ada. Analisis dan desain yang cermat diperlukan untuk memastikan bahwa integrasi ini berjalan lancar dan menghasilkan nilai tambah bagi organisasi. Selain itu, penting juga untuk mempertimbangkan aspek etika dalam pengembangan sistem informasi, seperti penggunaan data secara bertanggung jawab dan menghindari bias algoritma.

Gambar 1.1. Illustrasi desain sistem informasi
(Sumber : www.freepik.com)

Dalam menghadapi persaingan yang semakin ketat, organisasi perlu mampu beradaptasi dengan cepat terhadap perubahan lingkungan bisnis. Analisis dan desain sistem informasi yang agile memungkinkan organisasi untuk mengembangkan sistem yang dapat dengan mudah diubah dan disesuaikan dengan kebutuhan bisnis yang terus berkembang. Dengan demikian, organisasi dapat tetap relevan dan kompetitif dalam jangka panjang.

1.2 Evolusi Metode Pengembangan Sistem Informasi

Perkembangan teknologi informasi yang pesat telah memicu evolusi signifikan dalam metode pengembangan sistem informasi. Dari pendekatan tradisional yang kaku hingga metode modern yang lebih fleksibel dan adaptif, perjalanan ini telah membentuk cara kita merancang dan membangun sistem informasi (Al Hakim *et al.*, 2022). Pada awalnya, metode pengembangan sistem informasi didominasi oleh pendekatan waterfall yang linier dan sekuensial. Setiap tahap pengembangan, mulai dari analisis hingga implementasi, harus diselesaikan secara berurutan sebelum melanjutkan ke tahap berikutnya. Meskipun metode ini menawarkan struktur yang jelas, namun kurang fleksibel dalam menghadapi perubahan kebutuhan pengguna atau teknologi.

Seiring berjalannya waktu, keterbatasan metode waterfall mendorong munculnya pendekatan-pendekatan baru yang lebih adaptif. Salah satunya adalah metode agile yang menekankan pada fleksibilitas, kolaborasi, dan pengiriman produk yang sering. Metode agile memungkinkan tim pengembangan untuk merespons perubahan dengan cepat dan memberikan nilai bisnis secara berkelanjutan. Beberapa framework agile yang populer antara lain Scrum dan Kanban (Wahyudi, Sunardi and Riadi, 2022). Selain agile, muncul juga pendekatan DevOps yang mengintegrasikan pengembangan perangkat lunak (*development*) dan operasi IT (*operations*). DevOps bertujuan untuk mempercepat pengiriman perangkat lunak, meningkatkan kualitas, dan meningkatkan kolaborasi antara tim pengembangan dan operasi (Grotta, 2021).

Perkembangan terbaru dalam metode pengembangan sistem informasi:

1. ***Low-code/no-code development***
   Memungkinkan pengembangan aplikasi dengan sedikit atau tanpa coding, mempercepat waktu pengembangan. Platform *low-code/no-code* menyediakan antarmuka visual yang intuitif dan *drag-and-drop* (Gan *et al.*, 2023), sehingga pengguna dengan berbagai tingkat keahlian dapat membangun aplikasi dengan cepat. Fleksibilitas tinggi memungkinkan penyesuaian dan perluasan aplikasi sesuai dengan kebutuhan bisnis yang terus berkembang.
2. ***Microservices architecture***
   Membagi aplikasi menjadi layanan-layanan kecil yang independen, meningkatkan skalabilitas dan fleksibilitas. Setiap layanan dapat dikembangkan, di-deploy, dan di-*scale* secara independen (Toh *et al.*, 2022), sehingga memungkinkan tim pengembangan untuk bekerja secara paralel dan lebih cepat. Dengan arsitektur ini, perusahaan dapat dengan mudah menyesuaikan kapasitas sistem untuk memenuhi permintaan yang fluktuatif.
3. ***AI-driven development***
   Menggunakan kecerdasan buatan untuk membantu dalam proses pengembangan, seperti pengujian otomatis dan pembuatan kode. AI dapat membantu dalam menemukan

pola dan tren dalam data, sehingga memungkinkan tim pengembangan untuk menciptakan fitur-fitur baru yang inovatif (Jizzakh, 2021). AI juga dapat digunakan untuk memprediksi perilaku pengguna, sehingga aplikasi dapat disesuaikan dengan kebutuhan pengguna secara *real-time*.

1.3 Konsep Dasar Analisis Sistem: Memahami Jiwa Sistem Informasi

Analisis sistem merupakan fondasi kokoh bagi pembangunan sistem informasi yang sukses. Ibarat seorang arsitek merancang sebuah bangunan, seorang analis sistem merancang *blueprint* sebuah sistem informasi. Proses ini melibatkan pemahaman mendalam tentang kebutuhan pengguna, identifikasi masalah, dan perumusan solusi yang tepat. Analisis sistem adalah seni dan ilmu dalam menerjemahkan kebutuhan bisnis menjadi solusi teknologi. Seorang analis sistem tidak hanya sekadar memahami teknologi, tetapi juga harus memiliki pemahaman yang mendalam tentang bisnis dan organisasi. Mereka harus mampu berkomunikasi dengan berbagai pihak yang terlibat, mulai dari manajemen puncak hingga pengguna akhir, untuk menggali informasi yang relevan.

Proses analisis sistem melibatkan serangkaian aktivitas yang sistematis. Dimulai dengan pengumpulan data melalui wawancara, observasi, dan studi dokumen, seorang analis sistem akan mengidentifikasi masalah-masalah yang ada dalam sistem yang sedang berjalan. Selanjutnya, data tersebut akan dianalisis untuk menemukan pola, tren, dan akar penyebab masalah (Harahap *et al.*, 2023). Hasil analisis ini kemudian digunakan untuk merumuskan persyaratan sistem yang baru. Salah satu tantangan terbesar dalam analisis sistem adalah mengelola kompleksitas. Sistem informasi modern seringkali melibatkan banyak komponen yang saling terkait, seperti perangkat keras, perangkat lunak, basis data, dan jaringan. Seorang analis sistem harus mampu mengelola kompleksitas ini dengan menggunakan berbagai teknik pemodelan, seperti

diagram alir data (DFD), diagram entiti-relasi (ERD), dan *Unified Modeling Language* (UML).

Tujuan Analisis Sistem

Tujuan utama analisis sistem adalah untuk menghasilkan gambaran yang jelas dan akurat tentang sistem yang ada atau yang akan dibangun. Dengan melakukan analisis sistem secara menyeluruh, kita dapat menghindari kesalahan desain yang mahal dan memakan waktu. Selain itu, analisis sistem juga membantu memastikan bahwa sistem yang dibangun dapat bertahan dalam jangka panjang dan mudah diadaptasi dengan perubahan kebutuhan bisnis. Dengan demikian, kita dapat:

1. **Mengidentifikasi masalah**
   Menemukan ketidaksesuaian antara sistem yang ada dengan kebutuhan pengguna atau tujuan organisasi. Misalnya, sistem yang terlalu lambat, tidak efisien, atau tidak aman. Selain masalah teknis, analisis sistem juga mengidentifikasi masalah fungsional, seperti kurangnya fitur, ketidakakuratan data, atau kesulitan dalam penggunaan sistem. Dengan mengidentifikasi semua jenis masalah, kita dapat memastikan bahwa solusi yang dihasilkan benar-benar komprehensif.
2. **Menganalisis kebutuhan**
   Menganalisis kebutuhan. Menentukan persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh sistem baru. Persyaratan fungsional berkaitan dengan apa yang harus dilakukan oleh sistem, sedangkan persyaratan non-fungsional berkaitan dengan bagaimana sistem harus dilakukan (misalnya, kinerja, keamanan, dan usabilitas). Proses analisis kebutuhan melibatkan interaksi intensif dengan pengguna akhir, stakeholder, dan pihak terkait lainnya untuk memahami secara mendalam apa yang mereka harapkan dari sistem.
   Persyaratan fungsional dijabarkan dalam bentuk fitur-fitur spesifik yang harus dimiliki oleh sistem, seperti kemampuan untuk melakukan perhitungan tertentu, menghasilkan laporan, atau mengelola data. Persyaratan non-fungsional

lebih bersifat kualitatif dan menentukan kualitas keseluruhan sistem, seperti kecepatan respons sistem, tingkat keamanan data, kemudahan penggunaan, dan kompatibilitas dengan sistem lain. Dalam menentukan persyaratan, penting untuk menggunakan teknik-teknik seperti wawancara, survei, dan observasi. Teknik-teknik ini membantu mengumpulkan informasi yang relevan dan akurat mengenai kebutuhan pengguna. Selain itu, penggunaan model-model analisis seperti *Use Case*, *Data Flow Diagram*, dan *Entity Relationship Diagram* juga sangat bermanfaat untuk memvisualisasikan dan mendokumentasikan persyaratan sistem.

3. **Mengevaluasi kinerja**
   Mengukur sejauh mana sistem yang ada mencapai tujuannya. Evaluasi ini mencakup aspek-aspek seperti efisiensi, efektivitas, dan kepuasan pengguna. Proses evaluasi kinerja ini sangat penting untuk memastikan bahwa sistem yang telah dibangun berfungsi dengan baik dan memberikan nilai tambah bagi organisasi. Efisiensi mengukur seberapa optimal sumber daya (waktu, biaya, dan sumber daya lainnya) yang digunakan oleh sistem untuk mencapai hasil yang diinginkan. **Efektivitas** mengukur sejauh mana sistem berhasil mencapai tujuan yang telah ditetapkan. Sedangkan **kepuasan pengguna** mengukur sejauh mana pengguna merasa puas dengan sistem yang ada, baik dari segi kemudahan penggunaan, fitur yang tersedia, maupun kualitas layanan yang diberikan.
4. **Merancang solusi**
   Merumuskan solusi yang inovatif dan efektif untuk mengatasi masalah yang telah diidentifikasi. Proses ini melibatkan kreativitas dan kemampuan berpikir analitis yang tinggi. Seorang analis sistem harus mampu mengidentifikasi berbagai alternatif solusi, mengevaluasi kelebihan dan kekurangan masing-masing, serta memilih solusi yang paling optimal. Solusi yang baik tidak hanya mampu mengatasi masalah yang ada, tetapi juga harus

mempertimbangkan faktor-faktor lain seperti biaya, waktu implementasi, dan keberlanjutan.

Dalam merumuskan solusi, seorang analis sistem juga harus memperhatikan prinsip-prinsip desain yang baik. Prinsip-prinsip ini mencakup usabilitas, efisiensi, keamanan, fleksibilitas, dan maintainability. Usabilitas memastikan bahwa sistem mudah digunakan oleh pengguna, efisiensi menjamin bahwa sistem dapat beroperasi dengan cepat dan tanpa membuang-buang sumber daya, keamanan melindungi sistem dari ancaman eksternal, fleksibilitas memungkinkan sistem untuk beradaptasi dengan perubahan kebutuhan, dan maintainability memudahkan pemeliharaan dan pengembangan sistem di masa depan.

Selain itu, seorang analis sistem juga harus mempertimbangkan faktor-faktor eksternal seperti regulasi pemerintah, standar industri, dan tren teknologi terbaru. Dengan demikian, solusi yang dihasilkan tidak hanya relevan dengan kebutuhan saat ini, tetapi juga dapat bertahan dalam jangka panjang.

Tahapan Analisis Sistem

Proses analisis sistem umumnya melibatkan beberapa tahapan berikut:

1. **Pengumpulan data**

   Mengumpulkan informasi yang relevan dari berbagai sumber, seperti wawancara dengan pengguna, observasi langsung, dan dokumentasi sistem yang ada. Proses pengumpulan data ini merupakan langkah awal yang krusial dalam pengembangan sistem informasi. Data yang akurat dan komprehensif akan menjadi dasar dalam merancang sistem yang memenuhi kebutuhan pengguna dan tujuan organisasi.

   Wawancara dengan pengguna memungkinkan kita untuk menggali lebih dalam mengenai harapan, kebutuhan, dan kendala yang mereka hadapi dalam pekerjaan sehari-hari. Observasi langsung memberikan pemahaman yang lebih

baik tentang alur kerja, interaksi pengguna dengan sistem yang ada, dan potensi masalah yang mungkin terjadi. Sementara itu, dokumentasi sistem yang ada memberikan gambaran umum tentang struktur, fungsi, dan data yang digunakan oleh sistem saat ini (Wahyudi, Sunardi and Riadi, 2022). Selain ketiga metode di atas, teknik lain yang dapat digunakan adalah survei. Survei memungkinkan kita mengumpulkan data dari sejumlah besar responden dalam waktu yang relatif singkat. Analisis dokumen juga penting untuk memahami kebijakan, prosedur, dan standar yang berlaku dalam organisasi.

2. **Analisis data**
   Menganalisis data yang telah dikumpulkan untuk mengidentifikasi pola, tren, dan hubungan antara berbagai elemen dalam sistem. Tujuan utama dari analisis data adalah untuk mendapatkan pemahaman yang lebih mendalam tentang sistem yang sedang diteliti, sehingga dapat digunakan sebagai dasar untuk membuat keputusan yang lebih baik.
3. **Pemodelan sistem**
   Membuat model sistem yang menggambarkan struktur dan fungsi sistem secara logis. Model ini dapat berupa diagram alir data (DFD), diagram entiti-relasi (ERD), atau model UML (Peranginain, 2022). DFD menggambarkan aliran data dalam sistem, ERD merepresentasikan hubungan antara entitas data, sedangkan UML menyediakan berbagai diagram untuk memodelkan berbagai aspek sistem secara lebih detail.
4. **Verifikasi dan validasi**
   Memastikan bahwa model sistem yang telah dibuat sudah benar dan sesuai dengan kebutuhan pengguna. Proses validasi ini melibatkan verifikasi model dengan merujuk kembali pada data yang telah dikumpulkan sebelumnya dan melibatkan pengguna untuk memberikan umpan balik. Teknik-teknik seperti walkthrough, inspection, dan prototyping dapat digunakan untuk memvalidasi model sistem. Proses validasi seringkali bersifat iteratif, di mana

model akan terus diperbaiki dan disempurnakan berdasarkan umpan balik yang diperoleh. Tujuan akhir dari proses validasi adalah untuk memastikan bahwa model sistem yang dihasilkan dapat secara akurat merepresentasikan sistem yang sebenarnya dan memenuhi kebutuhan pengguna.

Pentingnya Analisis Sistem

Analisis sistem yang dilakukan dengan baik akan menghasilkan desain sistem yang lebih baik, efektif, dan efisien. Selain itu, analisis sistem juga dapat membantu mengurangi risiko kegagalan proyek, meminimalkan biaya pengembangan, dan meningkatkan kualitas sistem secara keseluruhan. Dengan melakukan analisis sistem, kita dapat memperoleh pemahaman yang lebih mendalam tentang proses bisnis, kebutuhan pengguna, dan kendala yang ada. Hal ini memungkinkan kita untuk merancang sistem yang benar-benar mendukung tujuan bisnis. Analisis sistem yang komprehensif akan menghasilkan sistem yang lebih mudah dipelihara dan dikembangkan di masa mendatang, sehingga dapat meningkatkan produktivitas dan daya saing organisasi.

DAFTAR PUSTAKA

Dawis, A.M., Murhadi, M. and Ardhani, R. (2023) 'Virtual Reality Sebagai Media Promosi Wisata Kuliner Halal Di Solo Baru Sukoharjo', INTEK : Jurnal Informatika dan Teknologi Informasi, 6(1), pp. 87–93. Available at: https://doi.org/10.37729/intek.v6i1.3161.

Fitriana, S.M., Permanasari, A.E. and Sanjaya, G.Y. (2023) 'Evaluasi Migrasi Dan Integrasi Sistem Informasi Manajemen Rumah Sakit Menggunakan Metode Hot-Fit Di RS Pratama Kota Yogyakarta', Jurnal Admmirasi [Preprint]. Available at: https://doi.org/10.47638/admmirasi.v8i1.233.

Gan, W. et al. (2023) 'Exposure to ultrafine particles and cognitive decline among older people in the United States', Environmental Research, 227, p. 115768. Available at: https://doi.org/https://doi.org/10.1016/j.envres.2023.115768.

Grotta, A. (2021) 'DevOpsBL: DevOps-based learning on information systems higher education', 27th Annual Americas Conference on Information Systems, AMCIS 2021 [Preprint]. Available at: https://api.elsevier.com/content/abstract/scopus_id/85118663865.

H., D.A.R. and Irbayuni, S. (2023) 'Menuju Karir Masa Depan: Pelatihan Skill untuk Analis Data Sebagai Peluang Baru di Era Digital Melalui Program Studi Independen di Bitlabs Academy', TRIDHARMADIMAS: Jurnal Pengabdian Kepada Masyarakat Jayakarta, 3(1), p. 5. Available at: https://doi.org/10.52362/tridharmadimas.v3i1.1165.

Al Hakim, R.R. et al. (2022) 'Pemanfaatan Teknologi IoT untuk Pertanian Berkelanjutan (IoT Technology for Sustainable Agriculture)', E-Prosiding Seminar Nasional Inovasi Teknologi Pertanian Berkelanjutan (INOPTAN), 1(1), pp. 1–9. Available at: https://jurnal.unikastpaulus.ac.id/index.php/inoptan/article/view/1400.

Harahap, C.P. et al. (2023) 'Perancangan Sistem Jadwal Perkuliahan di Kampus Institut Tekhnologi dan Sains Padang Lawas Utara', Jurnal Pendidikan Tambusai, 7(2), pp. 16127–16136.

Jizzakh, O.A. ugli A. (2021) 'Impact Factor : International Scientific Journal Theoretical & Applied Science THE EFFECT OF DIVERSITY OF THE NATIONALITY , BOARD OF DIRECTOR , INVESTMENT DECISION , FINANCING DECISION , AND Impact Factor ':, International scientific journal [Preprint]. Available at: https://doi.org/10.15863/TAS.

Peranginain, kasiman (2022) 'PERANCANGAN APLIKASI PEMBAYARAN BUKU PAKET DAN BUKU MODUL PADA SEKOLAH DASAR SWASTA HANG TUAH 3 BERBASIS JAVA Ita', Informatika, 03(03), pp. 477–483.

Toh, W.L. et al. (2022) 'Investigating predictors contributing to the expression of schizotypy during the COVID-19 pandemic', Journal of Psychiatric Research, 150, pp. 231–236. Available at: https://doi.org/https://doi.org/10.1016/j.jpsychires.2022.03.060.

Wahyudi, A., Sunardi, S. and Riadi, I. (2022) 'Peran Strategis Scrum Master Pada Pengembangan Perangkat Lunak Perpustakaan Sekolah Berbasis Android', JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 7(3), pp. 711–717. Available at: https://doi.org/10.29100/jipi.v7i3.2994.

BAB 2
METODOLOGI DAN KERANGKA KERJA DALAM ANALISIS TI

2.1 Pendahuluan

Analisis TI merupakan pendekatan menyeluruh yang membantu organisasi memahami, merancang, dan mengoptimalkan sistem TI untuk mendukung tujuan strategis mereka. Dengan pendekatan metodologis yang terstruktur, analisis ini tidak hanya meningkatkan efisiensi dan keamanan, tetapi juga memastikan bahwa solusi teknologi yang dihasilkan dapat mendukung pertumbuhan organisasi secara berkelanjutan. Analisis ini meliputi proses yang komprehensif untuk mempelajari, mengevaluasi, dan merancang sistem teknologi berbasis komputer dan informasi untuk memecahkan masalah, meningkatkan kinerja, atau mendukung operasi dalam suatu organisasi. Ini mencakup berbagai aspek mulai dari pemahaman kebutuhan pengguna, arsitektur sistem, integrasi perangkat keras dan perangkat lunak, hingga optimalisasi dan keamanan infrastruktur TI. Agar tujuan ini dapat terwujud, tentu saja dibutuhkan perecanaan tata kelola TI yang professional, baik menyangkut keamanan maupun sistem audit. Oleh karena itu, diperlukan tolok ukur beserta peralatan yang digunakan untuk memilih, menerapkan serta mengoperasikan kerangka kerja dan metodologi yang relevan, efektif dan efisien.

Proses ini dimulai dengan identifikasi kebutuhan dan tujuan pengguna atau organisasi. Analis TI dapat dimulai dengan mengumpulkan informasi melalui wawancara, survei, atau observasi untuk memahami masalah spesifik yang dihadapi pengguna atau kebutuhan peningkatan yang mereka harapkan, dan memastikan bahwa solusi yang dikembangkan akan memenuhi kebutuhan operasional yang sesungguhnya dan berfungsi optimal dalam konteks bisnis.

Setelah memahami kebutuhan, langkah berikutnya adalah mendesain arsitektur sistem yang mendukung fungsi dan tujuan

yang telah diidentifikasi. Ini mencakup desain perangkat keras, perangkat lunak, jaringan, basis data, dan keamanan sistem. Biasanya, arsitektur sistem terdiri dari lapisan pengguna, aplikasi, data, dan infrastruktur jaringan, dimana masing-masing dirancang agar saling terintegrasi dan efisien.

Beberapa pendekatan yang digunakan dalam perancangan arsitektur ini termasuk UML (*Unified Modeling Language*) untuk pemodelan sistem, TOGAF (*The Open Group Architecture Framework*) untuk arsitektur bisnis, dan ITIL (*Information Technology Infrastructure Library*) untuk manajemen layanan.

Memilih teknologi yang paling sesuai untuk kebutuhan spesifik organisasi adalah bagian inti dari analisis teknologi informatika. Ini mencakup evaluasi berbagai perangkat keras, perangkat lunak, platform cloud, serta solusi keamanan yang dapat memenuhi persyaratan sistem. Sebagai contoh, pemanfaatan *cloud computing*, big data, atau solusi berbasis kecerdasan buatan (AI) untuk meningkatkan efisiensi atau memperluas fungsionalitas sistem.

Tahap analisis TI berikutnya adalah pengembangan dan integrasi sistem. Dimana pada tahap ini, sistem dibangun atau dikembangkan berdasarkan desain yang telah disetujui, dengan memperhatikan integrasi yang sempurna antara berbagai komponen, baik dalam hal *hardware* maupun *software*. Pengembangan dan iterasi sistem TI, bisa dilakukan dengan metode Agile, DevOps, dan metode iteratif, yang sering digunakan untuk memastikan bahwa pengembangan berlangsung secara bertahap, dengan masukan pengguna yang berkelanjutan, sehingga memungkinkan penyesuaian cepat terhadap kebutuhan yang berubah.

Selanjutnya, untuk menjamin bahwa sistem dapat berjalan dengan andal, aman, dan sesuai dengan ekspektasi fungsional, dilakukan pengujian dan validasi sistem. Proses pengujian ini meliputi pengujian unit, pengujian integrasi, pengujian kinerja, dan pengujian keamanan untuk mengidentifikasi dan memperbaiki potensi kelemahan serta risiko lainnya.

Guna mencapai TI yang maksimal, Laurent Renard (2016) menjelaskan pentingnya menyusun organisasi tata kelola TI,

menyusun manajemen portofolio yang kemudian diuraikan dalam bentuk manajemen proyek termasuk analisis bisnis agar dapat menyediakan produk serta layanan yang terbaik. Laurent mendeskripsikan metodologi dan kerangkat kerja ini sebagaimana gambar 2.1.

2.2 Kerangka Kerja dan Metodologi

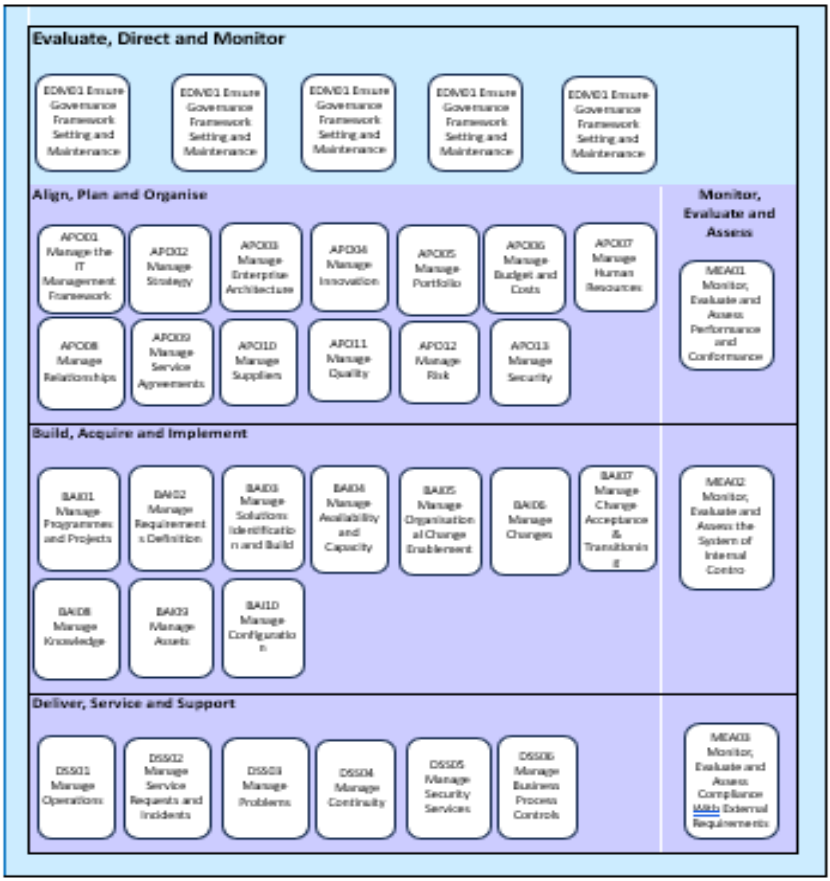
2.2.1 COBIT

Analisis TI dibutuhkan untuk menentukan cara yang yang efektif dan efisien dalam memungkinkan organisasi mencapai tujuannya melalui pemeliharaan risiko pada tingkat yang sesuai dengan pemangku kepentingan. Pada tata kelola TI, COBIT merupakan serangkaian kegiatan, meliputi penyelarasan, perencanaan, dan pengaturan; membangun, memperoleh, dan menerapkan proses kontrol sistem informasi dan teknologi dengan tujuan menyelaraskan TI dengan bisnis. COBIT beroperasi pada tingkat strategis yang telah diselaraskan dengan berbagai standar TI serta praktik terbaik yang lebih spesifik, seperti *Information Technology Infrastructure Library* (ITIL), standar *dari International Organization for Standardization/International Electrotechnical Commission* (ISO/IEC) 27000, *Capability Maturity Model Integration* (CMMI), *Open Group Architecture Framework* (TOGAF), *Projects in Controlled Environment* versi 2 (PRINCE2), dan *Project Management Professional* (PMP). Dalam hal ini, COBIT bertindak sebagai integrator yang bertujuan menghubungkan tata kelola dan TI dan kebutuhan bisnis. Adapun proses COBIT selengkapnya tersaji pada gambar 2.2.

Life Cycle TI	Developments				Operations
COBIT (IT governance)	Align, Plan and Organize; Build, Acquire and Implement				Deliver, Service and Support
	Monitor, Evaluate and Assess				
TOGAF (IT/enterprise architecture)	implementation governance, architecture change management				
PfMP (Portfolio Management Professional)	Programs/projects (identification and categorization, evaluation, selection, prioritization)		Portfolio (reporting and review, monitoring/control, risk management)		
MoP (Management of Portofolios)	Understand, categorize, prioritize, balance, plan		Management control, benefits management, financial management, risk management, stakeholder engagement, organizational governance, resource management		
PgMP (Program Management Professional)	Pre-program preparation	Program initiation	Program setup		Delivery of program benefits, program closure
MSP (Managing Successful Program)	Identifying a program	Defining a program	Delivering the capability, managing the tranches		Realizing benefits, closing program
PMP (Predictive Project Management)	Initiating	Planning	Executing, monitoring/ controlling	Closing	
PRINCE2 (Iterative project management)	Starting project	Initiating project	Directing project, controlling a stage, managing stage boundaries, managing product delivery	Closing project	
Scrum (Adaptative project management)	Product vision	Road map	Product backlog, sprint planning, development, daily meeting, sprint review, sprint retrospective		
PMI-PBA (Business analysis)	Needs assessement	Business analysis planning	Requirements elicitation and analysis, traceability and monitoring	Solution evaluation	
CBAP (Business analysis)	Strategy analysis	Business analysis planning and monitoring	Elicitation and collaboration, requirement life cycle Management, requirement analysis and design	Solution evaluation	

ITIL (Classical IT life cycle)	Strategy	Design	Transition	Operations
			Improvement	
DevOps (Agile IT life cycle)	Flow->			<-Feedback
	Continuous experimentation and learning			
Lean Six Sigma (Processes optimization)				Define, measure, analyze, improve, control

Gambar 2.1. Metodologi dan Kerangka Kerja Analisis TI
(Sumber : Laurent Renard, 2016)

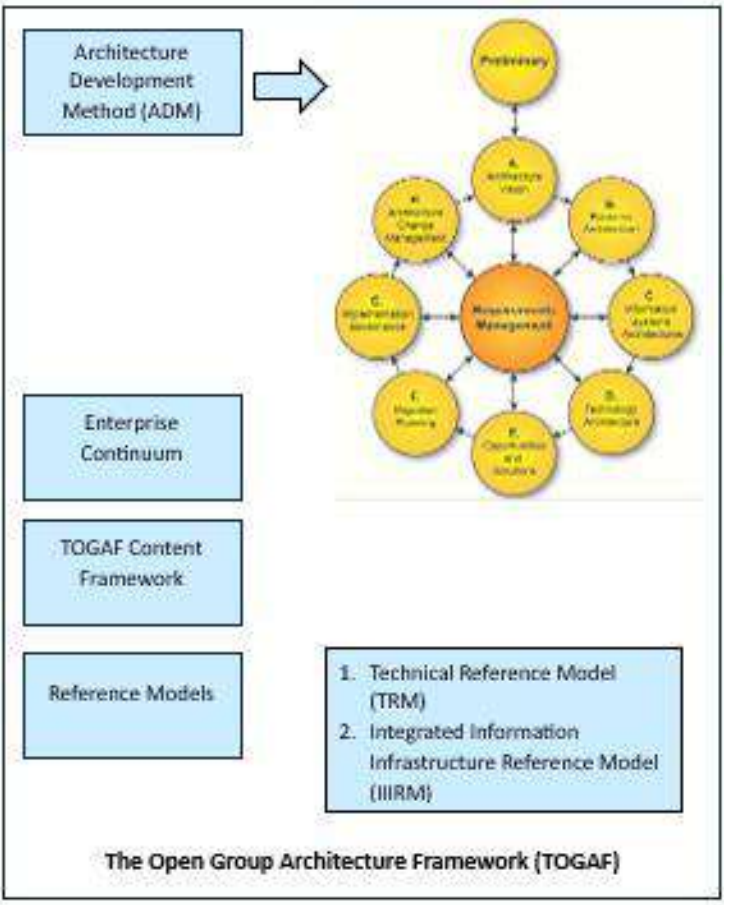


Gambar 2.2. Proses COBIT Manajemen Tata Kelola TI
(Sumber : Laurent Renard, 2016)

2.2.2 *The Open Group Architecture Framework*

TOGAF merupakan *framework* standar yang digunakan untuk melakukan perancangan, perencanaan, implementasi, dan mengelola arsitektur TI organisasi melalui pendekatan sistematis untuk mengelola arsitektur organisasi agar mendukung strategi bisnis, teknologi, dan operasional secara efektif.

TOGAF terdiri dari beberapa bagian inti yang saling mendukung, sebagaimana dideskripsikan pada gambar 2.3. Sebagai catatan, TOGAF memberikan nilai dengan memastikan konsistensi dan evolusi yang efisien dari berbagai komponen arsitektur.



Gambar 2.3. Komponen Inti TOGAF
(Sumber : Wikata, 2018)

2.2.3 *Portfolio Management Professional Certification*

Manajemen portofolio adalah pengelolaan terpusat atas berbagai proses, metode, dan teknologi yang digunakan untuk menganalisis serta mengelola program atau proyek yang sedang berjalan maupun yang direncanakan, berdasarkan karakteristik utamanya. Tugas utama manajer portofolio adalah mengoptimalkan alokasi sumber daya dan penjadwalan aktivitas guna mencapai tujuan operasional dan keuangan organisasi secara maksimal, sambil mematuhi berbagai batasan yang ditentukan oleh pelanggan, sasaran strategis, atau faktor eksternal lainnya.

Laurent menjelaskan, bahwa seorang Profesional Manajemen Portofolio (PfMP) harus mempu menyusun strategi serta pengimplementasian proyek, menggambarkan secara detil alur kerja dan hubungan antara proyek, program, manajemen proyek organisasi; serta mampu menjelaskan proses manajemen portofolio beserta rencana komunikasi, kinerja, risiko, dan manajemen perubahan yang diperlukan. Untuk manajemen portofolio, program, dan proyek yang terintegrasi sepenuhnya, disarankan untuk menggunakan PfMP, Profesional Manajemen Program (PgMP), dan PMP secara kolaboratif karena kompatibilitas total dari ketiga metodologi ini, yang semuanya telah ditulis oleh Project Management Institute (PMI), asosiasi keanggotaan nirlaba terbesar di dunia untuk profesi manajemen proyek.

2.2.4 *Management of Portfolios*

Sebagai manajemer proyek, *Manajemen Portofolio* (MoP) mengelola program perubahan ditinjau dari sudut pandang strategis. Metode ini memberikan deskripsi semua aktivitas perubahan, termasuk apa yang ada dalam portofolio, biaya yang timbul, risiko yang mungkin timbul, kemajuan yang dihasilkan, serta dampak yang ditimbulkan sesuai tujuan strategis organisasi.

2.2.5 *Program Management Professional Certification*

Manajemen program merupakan proses pengelolaan beberapa proyek yang saling berkaitan guna memperoleh manfaat optimal. Manajemen program menekankan keterkaitan sumber daya di semua proyek, mengatur proyek dan biayanya, juga risiko

program. (program dapat dibedakan dari proyek melalui contoh program iPhone, yang terdiri dari proyek iPhone (perangkat keras), proyek iOS, proyek Portal iTunes, dan berbagai proyek iApps.

Sertifikasi Program Management Professional (PgMP) memiliki tujuan strategis, menentukan manfaat, serta target hasil proyek dan menyediakan kemugkinan penyeleisaian permasalahan yang mungkin timbul. PgMP menyediakan perspektif holistik untuk menangani seluruh siklus hidup penciptaan nilai, dari konsepsi hingga realisasi manfaat, dan membuat hubungan antara manajer bisnis dan manajer proyek.

2.2.6 *Managing Successful Programs*

Pengelolaan Program yang Berhasil (MSP) dapat ditinjau berdasakan:

1. Alur transformasional (menyediakan rute melalui siklus hidup program dari konseps hingga penyampaian kemampuan, hasil, dan manfaat)
2. Tata kelola (menetapkan pemimpin proyek, tim pelaksana, struktur organisasi, dan metode kontrol yang sesuai, guna mendapatkan hasil sesuai yang diinginkan)
3. Prinsip (berasal dari pelajaran positif dan negatif dari pengalaman dalam mengerjakan program sebelumnya dan merupakan faktor umum yang mendukung keberhasilan pada setiap perubahan transformasional)

2.2.7 *Project Managements*

Manajemen proyek menjalankan tindakan pengorganisasian, memberikan motivasi, serta melaksanakan kontrol teradap sumber daya secara tepat untuk mencapai tujuan tertentu dan keberhasilan tertentu. Proyek adalah kegiatan yang direncanakan dengan jadwal tertentu dan dirancang untuk menghasilkan produk, layanan, atau hasil yang bermanfaat serta memberikan nilai tambah. Kriteria keberhasilan utama sebuah proyek meliputi ruang lingkup, waktu, dan biaya, dimana kualitas, risiko, dan sumber daya merupakan tantangan tersendiri. Pada sudut pandang sponsor, proyek ditentukan oleh profitabilitasnya. Dimana:
Profitabilitas: P = Manfaat/Biaya dengan P>1

Biaya Modal (CC) (CC terdiri dari Bunga Dividen (DI) pemegang saham dan Bunga Utang Jangka Panjang (LTDI) bankir) dan digabungkan dalam rumus berikut:

CC = xDI + yLTDI

x dan y : mewakili bagian masing-masing dari sumber daya jangka panjang perusahaan.

Tipe proyek dikelompokkan ke dalam: prediktif, iteratif, dan adaptif, dimana masing-masing memiliki metodologi yang berbeda yaitu PMP, PRINCE2, dan Scrum.

Predictive Type: Project Management Professional

PMP adalah sertifikasi manajemen proyek yang dijalankan berdasarkan panduan standar yang telah disepakati, yaitu PMBOK Guide (Panduan untuk Kerangka Pengetahuan Manajemen Proyek). Panduan ini menyediakan landasan dalam metodologi manajemen proyek klasik dan pedoman untuk pengelolaan proyek individual; mendefinisikan konsep terkait manajemen proyek; dan menjelaskan proses dan siklus hidup manajemen proyek.

Iterative Type: Projects in Controlled Environments, Version 2

PRINCE2 adalah metode manajemen proyek berbasis proses yang menggunakan pendekatan iteratif untuk memastikan pelaksanaan proyek yang efektif. Metode ini didasarkan pada tujuh prinsip, tujuh tema, dan tujuh proses inti. Keunggulan utama PRINCE2 meliputi: fokus pada justifikasi bisnis yang jelas, struktur organisasi yang terdefinisi untuk tim manajemen proyek, pendekatan perencanaan yang berorientasi pada produk, pembagian proyek ke dalam tahap-tahap yang dapat dikelola dan dikendalikan, serta fleksibilitas dalam penerapannya sesuai dengan kebutuhan proyek.

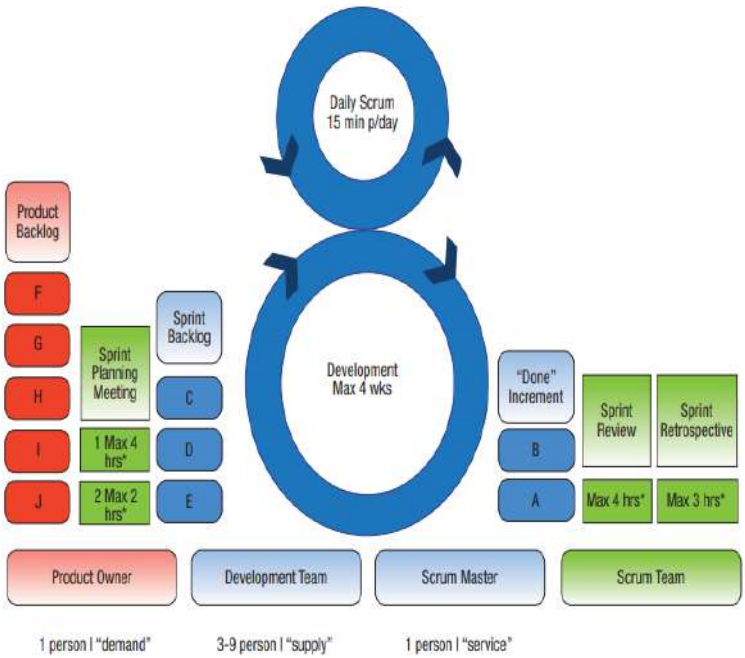
Proyek dianggap iteratif jika cakupan awal dapat ditentukan, namun detailnya disempurnakan sebelum memulai setiap siklus, dengan nilai yang dihasilkan setelah setiap tahap (contoh: implementasi modul manajemen hubungan pelanggan baru/CRM). Metode ini sangat cocok untuk proyek TI dengan sifat berulang.

Adaptive Type: Scrum

Scrum adalah kerangka kerja Agile yang digunakan untuk pengembangan perangkat lunak dengan pendekatan adaptif dan bertahap (inkremental). Dalam proses pengembangan produk, Scrum mengadopsi metode fleksibel dan menyeluruh, di mana tim berkolaborasi sebagai satu kesatuan untuk mencapai tujuan bersama. Kerangka kerja ini menekankan pentingnya kemampuan tim untuk mengatur diri sendiri, dengan menyoroti kolokasi fisik anggota tim serta komunikasi langsung yang dilakukan secara rutin setiap hari, baik di antara anggota tim maupun lintas disiplin dalam proyek.

Prinsip utama Scrum adalah pengakuan bahwa kebutuhan dan keinginan pelanggan dapat berubah selama proses produksi, terutama akibat perubahan dalam lingkungan atau kondisi. Karena itu, Scrum mengadopsi pendekatan empiris yang mengakui bahwa masalah tidak selalu dapat didefinisikan sepenuhnya di awal. Sebaliknya, kerangka kerja ini, seperti yang digambarkan dalam Gambar 2.4, berfokus pada memaksimalkan kemampuan tim untuk merespons kebutuhan yang muncul secara efektif dan menghasilkan output dengan cepat.

Proyek dikategorikan sebagai adaptif ketika cakupan keseluruhannya tidak dapat didefinisikan dengan jelas sejak awal dan perlu disempurnakan sebelum setiap sprint, dengan hasil yang bernilai signifikan dihasilkan setelah setiap sprint (contoh proyek adaptif: Uber). Scrum sangat sesuai untuk proyek adaptif, khususnya dalam bisnis inovatif berbasis web yang membutuhkan peluncuran cepat (time-to-market). Dalam situasi ini, meluncurkan inovasi lebih awal dapat memberikan keuntungan kompetitif yang besar, seperti menetapkan standar pasar atau bahkan menciptakan monopoli.



Gambar 2.4. Krangka Kerja Scrum
(Sumber : Laurent Renard, 2016)

2.2.8 *Business Analysis*

Analisis bisnis adalah bidang keahlian yang bertujuan untuk mengidentifikasi kebutuhan bisnis dan merancang solusi untuk menyelesaikan masalah yang dihadapi. Sementara manajer proyek berfokus pada cakupan proyek, analis bisnis memusatkan perhatian pada cakupan produk, khususnya melakukan analisis terhadap persyaratan dan memastikan bahwa perubahan yang diterapkan dalam organisasi selaras dengan tujuan strategisnya. Perubahan ini dapat mencakup aspek strategi, struktur organisasi, kebijakan, aturan bisnis, proses operasional, maupun sistem informasi.

Individu yang memiliki sertifikasi manajemen proyek, seperti PMP, atau bekerja di organisasi yang menerapkan praktik manajemen proyek, dapat memanfaatkan metodologi Professional Business Analysis (PBA). Di sisi lain, analis bisnis yang bekerja dalam organisasi yang berfokus sepenuhnya pada analisis bisnis

dapat memilih untuk mendapatkan sertifikasi (*Certified Business Analysis Professional* atau CBAP).

Project Management Institute-Professional Business Analysis

Pemegang sertifikasi PMI-PBA adalah profesional yang ahli dalam berkolaborasi dengan pemangku kepentingan untuk mengidentifikasi kebutuhan organisasi, merumuskan persyaratan proyek, dan memastikan proyek menghasilkan manfaat bisnis sesuai harapan. Sertifikasi PMI-PBA dirancang untuk analis bisnis yang memiliki pengalaman, atau ingin mengembangkan keahlian, dalam manajemen proyek atau program, terutama jika mereka bekerja di perusahaan yang menerapkan metodologi manajemen proyek berbasis panduan PMBOK (kerangka kerja yang digunakan dalam sertifikasi PMP).

Certified Business Analysis Professional

Certified Business Analysis Professionals yang biasa disingkat sebagai CBAP memiliki keahlian dalam memfasilitasi perubahan dalam konteks organisasi dengan mengidentifikasi kebutuhan dan menginisiasi solusi yang memberikan nilai lebih bagi pemangku kepentingan. Sertifikasi CBAP ditujukan untuk analis bisnis yang ingin melanjutkan karier mereka di bidang yang sama.

2.3 IT *Life Cycle Model*

Model siklus hidup TI adalah istilah yang digunakan dalam bidang rekayasa sistem, sistem informasi, dan rekayasa perangkat lunak untuk mendeskripsikan rangkaian tahapan, mulai dari perencanaan, pengembangan, pengujian, penerapan, hingga pengoperasian dan perbaikan sistem informasi secara berkelanjutan. Siklus hidup TI berfokus pada pengelolaan berbagai fase, mulai dari pengumpulan kebutuhan pelanggan hingga penyampaian nilai yang diinginkan secara efektif, serta perbaikan terus-menerus untuk mempertahankan daya saing.

2.3.1 *Information Technology Infrastructure Library*

ITIL merupakan serangkaian praktik dalam manajemen layanan TI (ITSM) yang dirancang untuk menyelaraskan layanan TI

dengan kebutuhan bisnis. ITIL mendefinisikan proses, prosedur, dan tugas yang bersifat umum dan tidak bergantung pada organisasi tertentu, tetapi dapat diadaptasi oleh berbagai organisasi untuk mendukung strategi mereka, memberikan nilai, dan menjaga standar kompetensi. ITIL sangat cocok digunakan bersama metode manajemen proyek tradisional maupun iteratif, seperti PMP atau PRINCE2.

2.3.2 *Development and Operations*

Development and Operations atau DevOps merupakan sekumpulan praktik dalam ITSM yang berorientasi pada pendekatan Agile, dengan tujuan menyelaraskan layanan TI dengan kebutuhan bisnis. Pendekatan ini menekankan pentingnya komunikasi, integrasi, otomatisasi, dan pengukuran kolaborasi antara pengembang perangkat lunak, jaminan kualitas (QA), dan operasi TI. Tujuan DevOps adalah membantu organisasi dalam menghasilkan produk dan layanan perangkat lunak dengan cepat serta meningkatkan kinerja operasional. Pendekatan DevOps mencakup seluruh jalur pengiriman dan mencakup peningkatan frekuensi penerapan, yang dapat mempercepat waktu pemasaran dan mengurangi tingkat kegagalan rilis baru. DevOps sangat cocok untuk kerangka manajemen proyek adaptif seperti Agile, terutama Scrum.

2.4 *Process Optimization*

Optimalisasi proses adalah bidang yang fokus pada pengelolaan proses untuk mengoptimalkan berbagai parameter tertentu tanpa melanggar batasan-batasan yang ada. Tujuan utama dari optimalisasi proses biasanya adalah untuk mengurangi biaya dan meningkatkan hasil serta/atau efisiensi. Optimalisasi proses merupakan salah satu alat kuantitatif penting dalam pengambilan keputusan di industri.

Lean Six Sigma

Lean Six Sigma merupakan metodologi optimasi proses yang mengandalkan kolaborasi tim untuk meningkatkan kinerja dengan cara sistematis mengurangi delapan jenis pemborosan: cacat,

produksi berlebih, waktu tunggu, potensi yang tidak dimanfaatkan, transportasi, inventaris, gerakan, dan pemrosesan tambahan. Lean Six Sigma mengintegrasikan prinsip-prinsip Lean dalam organisasi dengan pendekatan Six Sigma, menjadikannya metodologi yang secara unik didorong oleh pemahaman mendalam terhadap kebutuhan pelanggan. Metode ini menggunakan fakta, data, dan analisis statistik secara disiplin, serta menekankan perhatian berkelanjutan pada pengelolaan, peningkatan, dan perancangan ulang proses bisnis. Fokus utama Lean Six Sigma adalah mengoptimalkan nilai yang dihasilkan melalui berbagai proses.

2.5 Menentukan Alat, Metode dan Kerangka Kerja yang Tepat dan Relevan

Kemampuan mendefinisikan proyek dan memprediksi masalah yang mungkin timbul, sangat diperlukan untuk menentukan alat yang sesuai, relevan dan bermanfaat dalam menghasilkan solusi yang tepat. Sebagai contoh, saat melihat paku dan sekrup, seseorang harus mengetahui keberadaan palu dan obeng untuk memilih alat yang tepat. Namun, untuk mendapatkan nilai dan manfaat terbaik, semua tergantung pada kondisi, dimana seseorang harus menguasai kedua alat tersebut. Jika seseorang hanya memiliki palu, semuanya tampak seperti paku, yang merupakan gambaran dunia yang reduktif. Semakin seseorang menguasai alat yang tepat, dunia akan menjadi semakin besar dan kaya.

Dalam mengelola proyek IT yang semakin kompleks, ditambah perkembangan teknologi dan kebutuhan yang semakin beragam, penting diterapkan strategi efektif dalam mengelola proyek agar lebih terstruktur dan terorganisir.

Dengan penggunaan metodologi dan kerangka kerja yang jelas, pendelegasian tugas menjadi lebih jelas, sehingga lebih efisien, efektif, relevan dan bermanfaat dalam penyelesaian proyek. Penentuan strategi untuk setiap sprint plannya, dapat meminimalisir dan menyelesaikan permasalahan terhadap risiko yang mungkin timbul dalam penyelesaian proyek.

Selajutnya, dengan penggunaan alat, metodologi dan kerangka kerja yang tepat dan relevan, dapat memudahkan pelaksanaan proyek IT sehingga dapat diselesaikan tepat waktu.

DAFTAR PUSTAKA

Adamides, E., & Karacapilidis, N. (2020). Information technology for supporting the development and maintenance of open innovation capabilities. *Journal of Innovation & Knowledge*, *5*(1), 29-38.

Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Computers in industry*, *101*, 1-12.

Enric Senabre Hidalgo. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. Heliyon 5 (2019) e01447. doi: http://doi.org/10.1016/j.heliyon.2019. e01447

Grover, V., Chiang, R. H., Liang, T. P., & Zhang, D. (2018). Creating strategic business value from big data analytics: A research framework. *Journal of management information systems*, *35*(2), 388-423.

Hemon-Hildgen, A., & Rowe, F. (2022). Conceptualising and defining DevOps: a review for understanding, not a framework for practitioners. *European Journal of Information Systems*, *31*(5), 568-574.

Hidalgo, Enric Senabre. 2019. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. Heliyon. Vol. 5, Issue 3. https://doi.org/10.1016/j.heliyon.2019. e01447

Laurent Renard, 2016. Essential Frameworks and Methodologies to Maximize the Value of IT. ISACA Journal. Vol. 2. (1:7). isaca.org

Project Management Institute, General Information About Business Analysis, PMI Professional in Business Analysis (PMI-PBA) FAQs, 2015, www.pmi.org/~/media/PDF/ Certifications/PMI-PBA_FAQs_v2.ashx

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, *94*, 3563-3576.

Wikata, E.R., N.Y. Setiawan, and Y.T. Mursityo, Perencanaan Sistem Penjualan Menggunakan Togaf Architecture Development Method (TOGAF-ADM) Studi Pada PT. Millennium Pharmacon International Tbk Cabang Malang. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2018. 2: p. 10.

Hemon-Hildgen, A., & Rowe, F. (2022). Conceptualising and defining DevOps: a review for understanding, not a framework for practitioners. *European Journal of Information Systems*, *31*(5), 568-574.

BAB 3
DESAIN SISTEM YANG BERFOKUS PADA PENGGUNA

3.1 Pendahuluan

Pendekatan modern dalam analisis dan desain teknologi informasi (TI) berfokus pada pengembangan yang tangkas dan adaptif, memastikan bahwa sistem yang dibangun mampu beradaptasi dengan perubahan cepat dalam kebutuhan bisnis dan teknologi.

Beberapa pendekatan modern yang diterapkan adalah pengembangan berbasis agilitas, DevOps, serta pendekatan berbasis arsitektur yang berpusat pada layanan (*Service-Oriented Architecture*) dan cloud. Salah satu konsep yang juga semakin menonjol adalah desain sistem yang berfokus pada pengguna (User-Centered Design/UCD), yang bertujuan untuk meningkatkan efektivitas, efisiensi, dan kepuasan pengguna.

3.2 Karakteristik Kunci Pengembangan Tangkas dan Adaptif

Pengembangan yang tangkas dan adaptif merupakan pendekatan dalam pengembangan perangkat lunak atau sistem yang berfokus pada fleksibilitas, kolaborasi, dan respon cepat terhadap perubahan. Tujuan utama dari pendekatan ini adalah untuk memastikan bahwa tim pengembangan dapat dengan cepat menyesuaikan diri dengan perubahan kebutuhan bisnis atau teknologi, sehingga hasil akhirnya benar-benar relevan dan sesuai dengan harapan pengguna. Karakteristik kunci dari pengembangan tangkas dan adaptif, antara lain: iterasi dan incremental, kolaborasi tim yang kuat, fokus pada kebutuhan pengguna, adaptabilitas tinggi terhadap perubahan, pengurangan risiko dan pengoptimalan kualitas. Pendekatan tangkas dan adaptif ini semakin populer

karena mampu mengatasi kompleksitas dan ketidakpastian yang sering terjadi dalam proyek-proyek TI modern.

3.2.1 Iterasi dan Incremental

Pengembangan dilakukan dalam siklus-siklus pendek yang disebut *sprint* (dalam metodologi Scrum, misalnya), di mana setiap iterasi menghasilkan versi perangkat lunak yang dapat diujicoba. Hal ini memungkinkan pengguna untuk memberikan masukan sejak awal, yang kemudian digunakan untuk penyempurnaan di iterasi berikutnya. Pendekatan iteratif ini membantu dalam mendeteksi masalah lebih awal dan melakukan perbaikan secara berkelanjutan.

3.2.2 Kolaborasi Tim yang Kuat

Metodologi tangkas, seperti Agile, menekankan pentingnya komunikasi yang erat dan kolaborasi antar anggota tim pengembangan, serta antara tim pengembangan dan pemangku kepentingan (*stakeholder*). Hal ini membantu memastikan bahwa semua orang memiliki pemahaman yang sama tentang apa yang perlu dicapai dan siap merespon masukan yang muncul.

3.2.3 Fokus pada Kebutuhan Pengguna

Pendekatan ini mengutamakan kepuasan pengguna dengan mengakomodasi kebutuhan mereka, baik yang disampaikan secara langsung maupun yang muncul selama proses pengembangan. Dalam setiap iterasi, sistem diperiksa dari sudut pandang pengguna, sehingga tim dapat segera menyesuaikan fitur atau desain sesuai dengan umpan balik yang diterima

3.2.4 Adaptabilitas Tinggi terhadap Perubahan

Dalam pengembangan tangkas, perubahan dianggap sebagai hal yang wajar dan bahkan diharapkan. Jika ada perubahan dalam kebutuhan bisnis atau teknologi, tim dapat menyesuaikan rencana dan prioritas mereka tanpa harus mengulang proses dari awal. Hal ini mengurangi risiko proyek berakhir dengan sistem yang sudah ketinggalan zaman atau tidak relevan.

3.2.5 Pengurangan Risiko dan Pengoptimalan Kualitas

Dengan merilis sistem dalam potongan-potongan kecil yang terus disempurnakan, pendekatan tangkas membantu mengidentifikasi dan mengatasi risiko atau masalah kualitas sejak awal. Ini berbeda dari pendekatan tradisional di mana masalah sering kali baru ditemukan di akhir proyek, yang bisa menyebabkan biaya perbaikan menjadi jauh lebih tinggi.

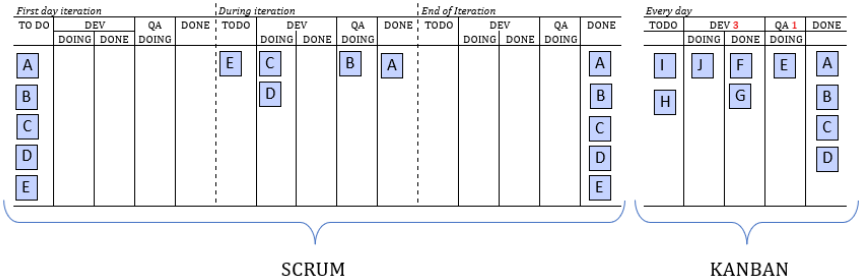
Metodologi yang sering diterapkan dalam pendekatan tangkas dan adaptif, meliputi:

1. Scrum. Dalam Scrum, pengembangan dilakukan dalam *sprint* (iterasi dalam durasi tetap) dengan tim yang berkolaborasi dalam peran tertentu seperti *Product Owner*, *Scrum Master*, dan pengembang. Dengan kata lain, dalam Scrum, tim bekerja dalam sprint dengan peran-peran khusus (peran tertentu), menggunakan rapat-rapat harian (*daily stand-up*) untuk melacak kemajuan, mengatasi hambatan, dan menyesuaikan prioritas sesuai kebutuhan (Wonoharjito et al, 2019).

   Scrum juga didefinisikan sebagai kerangka kerja tangkas yang dalam praktiknya juga memudahkan tim dalam mengimplementasikan produk baru, cepat beradaptasi serta memiliki kemampuan dalam melakukan perbaikan berkelanjutan. (Brezočnik, L., et al, 2016).
2. Kanban. Kanban adalah metodologi yang lebih fleksibel dan berfokus pada visualisasi proses kerja melalui papan tugas (task board), seperti misalnya, "To Do," "In Progress," "Done". Pendekatan ini membantu tim memahami kapasitas kerja mereka dan mencegah overload.yang membantu tim melacak kemajuan dan mengidentifikasi potensi hambatan. Brechner, E., (2015) dan Klipp P. (2014) menyebutkan bahwa prinsip kerja kanban dalam *task scheduling*, berdasarkan teknik *Justin-Time*, dimana untuk menghilangkan pengulangan yang tidak diperlukan dalam workflow, sistem ini diimplementasikan dalam waktu yang sedikit lambat. Klipp P. (2014) menjelaskan bahwa memvisualisasikan alur kerja merupakan hal penting pada metode Kanban, yang dilakukan dengan membuat papan

(board) Kanban yang mendeskripsikan semua langkah yang diperlukan pada proses pengembangan. Brechner, E., (2015) menambahkan perlunya membatasi pekerjaan yang dilakukan (Work in Progress/WiP) pada setiap langkah, agar optimal baik dalam hal tenaga kerja yang terlibat, kompleksitas sistem, serta parameter lainnya. Pembatasan ini memaksa kita fokus pada satu tugas dalam satu waktu. Sebagaimana prinsip "*achieve more by doing less* "

Penjelasan tentang *board* dalam implementasi metode Scrum dan metode Kanban, dijelaskan dalam gambar 3.1.



Gambar 3.1. Perbandingan *Board* pada Metode Scrum dan Kanban (Sumber: Brezočnik, L., & Majer, Č. 2016)

3.3 Pendekatan Modern Berbasis Agilitas

Pendekatan modern berbasis agilitas berbeda dari metode tradisional (seperti model Waterfall) yang biasanya mengikuti langkah-langkah yang kaku dan linier. Dalam metodologi berbasis agilitas, pengembangan dilakukan melalui siklus-siklus pendek yang memungkinkan tim untuk terus beradaptasi dengan perubahan dan menghasilkan perangkat lunak yang relevan dan berkualitas tinggi. Srivastava, A. et al (2017) menjelaskan bahwa sifat adaptif dan siklus hidup yang fleksibel, merupakan kelebihan pengembangan agile. Hal ini sangat berguna untuk pengembangan produk yang disesuaikan.

Metode pendekatan berbasis agilitas mengutamakan komunikasi yang erat antara tim pengembang dan pemangku kepentingan (*stakeholders*), seperti pengguna akhir, manajer proyek, dan klien. Kolaborasi ini memastikan bahwa setiap pihak

memiliki pemahaman yang sama tentang tujuan proyek dan perubahan kebutuhan dapat segera diakomodasi dan diimplementasikan. Pendekatan ini memungkinkan perubahan untuk diintegrasikan dengan cepat, bahkan di tahap-tahap akhir proyek. Karena fokusnya adalah memenuhi kebutuhan pengguna dan adaptif terhadap situasi yang dinamis, tim dapat mengubah prioritas atau menyesuaikan fitur sesuai umpan balik yang diterima. Sebagai konsekuensi atas visi pada pendekatan agilitas, otomatis pengujian dilakukan terus-menerus sepanjang siklus pengembangan, guna mendeteksi sejak dini terjadinya bug atau masalah kualitas, sehingga bisa segera diperbaiki. Metode seperti *Continuous Integration* dan *Continuous Delivery* mendukung praktik ini, guna membantu tim merilis pembaruan atau fitur baru secara berkelanjutan. Oleh karena itu, tim dalam metodologi agilitas biasanya bersifat mandiri dan memiliki otonomi dalam menentukan cara terbaik menyelesaikan tugas. Tim ini juga bertanggung jawab dalam merencanakan dan mengatur prioritas mereka sendiri. Menurut prinsip Agile, tim yang *self-organizing* cenderung lebih produktif dan memiliki rasa tanggung jawab yang lebih besar.

Pendekatan berbasis agilitas didasari oleh *Manifesto Agile*, yang dikembangkan pada tahun 2001 dan mencakup 12 prinsip yang menjadi panduan utama. Beberapa prinsip inti manifesto tersebut adalah:

1. Mengutamakan Kepuasan Pelanggan melalui Pengiriman Cepat dan Berkesinambungan
2. Mengakomodasi Perubahan Kebutuhan, Bahkan di Akhir Proses Pengembangan
3. Pengiriman Perangkat Lunak Berfungsi Secara Berkelanjutan
4. Kolaborasi Erat antara Pemilik Produk dan Tim Pengembang
5. Dukungan bagi Tim yang Termotivasi dan Self-Organizing

Pendekatan berbasis agilitas sangat efektif dalam menghadapi ketidakpastian dan perubahan cepat di dunia teknologi modern. Fleksibilitas, kolaborasi, dan adaptabilitas membuat pendekatan ini cocok untuk proyek-proyek inovatif yang menuntut respons cepat dan kualitas yang terjaga. Selain itu, pendekatan berbasis agilitas ini memiliki beberapa keunggulan, antara lain:

1. Kecepatan dan Fleksibilitas – Tim dapat merespons perubahan dengan cepat dan menyesuaikan arah pengembangan sesuai kebutuhan.
2. Kualitas Produk yang Lebih Baik – Pengujian yang konsisten memastikan perangkat lunak berkualitas tinggi dan memenuhi harapan pengguna.
3. Kepuasan Pengguna yang Tinggi – Umpan balik dari pengguna dapat diperoleh sejak awal dan digunakan untuk meningkatkan produk secara berkelanjutan.

Implementasi metode Agile, yang terbukti dapat diprediksi dan paling mudah adalah *scrumb*, sebab kerangka kerja yang dilakukan memiliki kemampuan untuk mengatasi kompleksitas masalah, serta kecepatannya dalam beradaptasi dengan perubahan (Fagarasan, C., et al, 2021).

Contoh Implementasi Pendekatan Agilitas, banyak dilakukan oleh perusahaan teknologi besar seperti Google, Microsoft, dan Amazon menggunakan metodologi berbasis agilitas untuk mempercepat siklus pengembangan produk mereka. Produk seperti *Google Chrome* dan *AWS* terus mengalami pembaruan dengan versi-versi baru yang dirilis secara reguler, memungkinkan pengguna untuk segera merasakan peningkatan kualitas dan fitur terbaru. Berdasarkan survey yang dilakukan oleh Hayat, F., et al (2019) didapatkan bahwa

Sebagian besar perusahaan perangkat lunak memiliki dampak positif saat menggunakan pengembangan Agile (Scrum) pada proyek perangkat lunak.

3.4 Desain sistem berbasis DevOps

Desain sistem berbasis **DevOps** merupakan pendekatan dengan mengintegrasikan proses pengembangan perangkat lunak *(Development*) dengan operasi teknologi informasi (*Operations*) untuk meningkatkan kolaborasi, efisiensi, dan kecepatan dalam siklus pengembangan perangkat lunak. DevOps bertujuan untuk menciptakan alur kerja yang lebih responsif, memungkinkan pengembangan dan peluncuran fitur baru dilakukan dengan lebih cepat, stabil, dan otomatis.

Pendekatan DevOps didasarkan pada beberapa prinsip utama yang bertujuan untuk menyelaraskan pengembangan dan operasi dalam sebuah tim. DevOps mendorong kolaborasi antara tim pengembang dan tim operasi, mengintegrasikan kedua tim ini, sehingga setiap anggota tim memiliki pemahaman yang sama tentang sistem dan tujuan proyek. Hal ini membantu mencegah konflik antara pengembangan dan penerapan.

DevOps memanfaatkan praktik *Continuous Integration* (CI) dan *Continuous Delivery* (CD) untuk mempercepat proses pengembangan dan pengujian. CI/CD memungkinkan pengembang mengintegrasikan kode mereka secara berkala ke dalam repositori utama dan melakukan pengujian otomatis untuk mendeteksi dan memperbaiki bug secara dini. Dengan CD, perubahan yang sudah diuji secara otomatis dapat diluncurkan langsung ke produksi. DevOps juga sangat menekankan otomatisasi proses, mulai dari integrasi, pengujian, hingga deployment. Otomatisasi ini berfungsi membantu mengurangi kesalahan manusia, mempercepat peluncuran fitur baru, dan memastikan konsistensi dalam proses. Alat-alat otomatisasi seperti Jenkins, GitLab CI, dan Docker sering digunakan untuk membuat pipeline CI/CD.

Setelah perangkat lunak diluncurkan, DevOps tetap memantau kinerja dan kestabilan sistem secara terus-menerus. Pemantauan ini memungkinkan tim untuk segera mendeteksi masalah atau performa yang menurun dan memberikan perbaikan dengan cepat. Feedback dari pengguna juga digunakan untuk terus meningkatkan produk.

DevOps berfokus pada respons cepat terhadap masalah yang muncul serta menjaga keandalan sistem, terutama dalam fase deployment. Sistem deployment otomatis mengurangi risiko downtime saat menerapkan pembaruan, memastikan pengguna mendapatkan pengalaman yang lebih stabil dan konsisten.

Desain sistem berbasis DevOps memungkinkan perusahaan merespons kebutuhan pengguna dan bisnis dengan lebih cepat dan efisien. Dengan integrasi yang erat antara pengembangan dan operasi, DevOps mendorong pengembangan perangkat lunak yang berkelanjutan, berkualitas tinggi, dan sesuai dengan standar modern.

Desain sistem berbasis DevOps melibatkan berbagai komponen dan alat untuk mendukung kinerjanya, yaitu: Pipeline CI/CD, *Containerization dan Orchestration, Infrastructure as Code* (IaC), Monitoring dan Logging.

3.4.1 Pipeline CI/CD

Pipeline CI/CD berfungsi sebagai alur yang mengintegrasikan setiap perubahan kode ke dalam sistem utama dan menguji perubahan tersebut secara otomatis sebelum di-deploy ke lingkungan produksi. Pipeline ini terdiri dari berbagai tahap seperti kompilasi, pengujian, deployment, dan verifikasi. Alat seperti Jenkins, GitLab CI/CD, CircleCI, dan Bamboo sering digunakan untuk mengotomatisasi pipeline CI/CD.

Konsep Dasar CI/CD mengacu pada proses integrasi kode secara berkelanjutan ke dalam repositori utama, biasanya dilakukan beberapa kali sehari. Setiap perubahan kode yang dikirimkan (*commit*) akan melalui proses pengujian otomatis. CI bertujuan untuk mendeteksi bug lebih awal dengan cara sering mengintegrasikan dan menguji kode. Setelah melalui pengujian di tahap CI, CD memastikan kode yang sudah diuji bisa di-deploy ke lingkungan produksi atau staging secara otomatis. Dengan CD, aplikasi yang sudah diuji dapat dirilis kapan saja ke pengguna akhir. Perubahan yang lolos dari pengujian otomatis langsung dideploy ke produksi tanpa campur tangan manual. Hal ini memastikan pengguna langsung mendapatkan perubahan begitu kode baru berhasil diuji.

Setiap tahap pada Pipeline CI/CD bertujuan memastikan kualitas dan kesiapan kode sebelum diterapkan di lingkungan produksi. Berikut adalah tahapan Pipeline CI/CD:

1. *Source/Version Control*. Semua kode disimpan dan dikelola di *repositori version control*, seperti Git. Setiap perubahan kode dikirim ke repositori, yang akan memicu pipeline CI/CD untuk memulai proses.
2. Build. Setelah perubahan terdeteksi, kode akan dikompilasi dan dibangun menjadi bentuk yang bisa dijalankan. Pada tahap ini, semua dependensi akan disertakan, dan binary akan dibuat. Kegagalan pada tahap ini mengindikasikan

masalah pada kode yang perlu diperbaiki sebelum melanjutkan.

3. Pengujian Otomatis. Tahap ini menjalankan pengujian otomatis untuk memastikan kualitas dan stabilitas kode. Ada beberapa jenis pengujian dalam CI/CD, yaitu:
   a. Unit Testing: Menguji fungsi atau komponen individu untuk memastikan mereka bekerja sebagaimana mestinya.
   b. Integration Testing: Menguji interaksi antar modul atau komponen untuk memastikan mereka berfungsi dengan baik bersama-sama.
   c. UI/End-to-End Testing: Menguji alur aplikasi dari perspektif pengguna akhir.
   d. Security Testing: Mengidentifikasi potensi kerentanan dalam kode.
4. Deployment ke Lingkungan Staging Setelah melalui pengujian, kode akan dideploy ke lingkungan staging untuk simulasi. Tahap ini memungkinkan tim melihat kode berjalan dalam lingkungan yang mirip dengan produksi, sehingga dapat mengidentifikasi masalah sebelum kode mencapai pengguna akhir.
5. Deployment ke Produksi (Production) Jika kode berhasil melalui semua tahap pengujian dan verifikasi, ia siap untuk di-deploy ke lingkungan produksi. Dalam CD, deployment bisa dilakukan secara manual (dengan persetujuan) atau otomatis (untuk Continuous Deployment). Deployment ke produksi sering kali menggunakan strategi seperti *blue-green deployment* atau *canary release* untuk mengurangi risiko downtime.
6. Pemantauan dan Feedback Setelah aplikasi dideploy ke produksi, pipeline CI/CD biasanya juga melibatkan pemantauan (monitoring) untuk mengidentifikasi masalah atau gangguan. Feedback dari pengguna atau sistem monitoring akan diambil untuk perbaikan lebih lanjut.

Pipeline CI/CD bermanfaat dalam mendeteksi terjadinya bug sejak dini. Pengujian yang dilakukan secara otomatis pada tahap-tahap awal memastikan bahwa masalah dapat ditemukan dan

diperbaiki lebih cepat, mengurangi risiko dan masalah besar terjadi pada tahap produksi. CI/CD mengotomatiskan proses deployment, memungkinkan perubahan fitur atau perbaikan bug dirilis dengan cepat dan konsisten tanpa hambatan manual. Positifnya, tim dapat bekerja dalam alur yang lebih kecil dan terus-menerus mengintegrasikan dan menguji kode mereka, sehingga dapat mendukung proyek besar dengan banyak pengembang, sebab semua pengembang dapat menggabungkan kode mereka secara berkala, meningkatkan kolaborasi dan konsistensi dalam pengembangan.

Namun demikian, dalam pengembangan Pipeline CI/CD terdapat beberapa tantangan antara lain kompleksitas pengaturan, dimana pengaturan Pipeline CI/CD bisa menjadi rumit, terutama untuk aplikasi besar atau aplikasi yang terdistribusi, yang pasti membutuhkan pengetahuan dan waktu untuk menyiapkan pengujian otomatis dan proses deployment. Tantangan berikutnya adalah kompleksitas pengelolaan nfrastruktur, dimana CI/CD biasanya membutuhkan lingkungan terpisah untuk pengujian dan staging. Infrastruktur ini harus dikelola dengan baik agar tidak terjadi kesalahan dalam proses deployment. Tantangan berikutnya yaitu dalam hal pengujian. Pipeline yang terlalu banyak mengalami proses pengujian bisa memperlambat proses. Namun, kurangnya pengujian berisiko menyebabkan masalah di produksi.

3.4.2 *Containerization* dan *Orchestration*

*Containerizatio*n dan *Orchestration* merupakan dua konsep kunci dalam infrastruktur modern yang memungkinkan pengembangan, pengujian, dan penerapan aplikasi yang lebih cepat, efisien, dan konsisten. Kedua konsep ini mendukung pendekatan DevOps dengan mempermudah pengelolaan dan penskalaan aplikasi dalam lingkungan yang dinamis.

1. ***Containerization***

   Containerization adalah proses mengisolasi aplikasi dan semua dependensinya (seperti pustaka, konfigurasi, dan runtime) ke dalam unit yang disebut *container*. Container berfungsi sebagai lingkungan yang terisolasi di mana aplikasi dapat berjalan konsisten, terlepas dari lingkungan tempat

container tersebut dijalankan (baik di komputer pengembang, server, atau cloud). Cara Kerja Containerization menggunakan kernel sistem operasi yang sama, tetapi mereka tetap terisolasi satu sama lain. Berbeda dengan virtual machine (VM) yang membutuhkan hypervisor dan sistem operasi penuh di setiap instance, container menggunakan kernel OS host, sehingga lebih ringan dan cepat.

Containerization bermanfaat dalam portabilitas, dimana Container yang telah dibuat dapat dijalankan di berbagai lingkungan tanpa harus melakukan konfigurasi ulang. Ini memungkinkan aplikasi yang dikembangkan di satu mesin bisa berjalan konsisten pada mesin yang lainnya. Dalam hal efisiensi sumber daya, Container lebih ringan dari VM karena berbagi kernel OS yang sama, sehingga membutuhkan lebih sedikit sumber daya, memungkinkan server untuk menjalankan lebih banyak container dibanding VM. Container mendukung *Continuous Integration* dan *Continuous Delivery* (CI/CD) karena memungkinkan pengembang untuk melakukan build, test, dan deploy secara otomatis dan cepat. Selain itu, Containerization dapat mengurangi risiko konflik atau ketergantungan antar aplikasi yang dijalankan di server yang sama, karena setiap container berjalan dalam lingkungan yang terisolasi.

2. *Orchestration*

Orchestration adalah proses pengelolaan container secara otomatis dalam skala besar. Ketika aplikasi terdiri dari puluhan atau bahkan ratusan container yang tersebar di beberapa server, pengaturan manual menjadi sangat kompleks. Orchestration memudahkan pengaturan container-container ini sehingga mereka bekerja bersama sebagai satu kesatuan.

Fungsi orchestration sebagai pengaturan dan distribusi, mengelola dan mendistribusikan container ke server yang tersedia berdasarkan sumber daya yang dibutuhkan, serta menjamin penggunaannya secara optimal. Sebagai fungsi Auto-Scaling,
orchestration dapat menambah atau mengurangi jumlah container secara otomatis berdasarkan beban kerja. Hal inilah

yang memungkinkan aplikasi untuk menangani lonjakan lalu lintas tanpa perlu campur tangan manual. Sebagai pemulihan otomatis, jika salah satu container mengalami gangguan atau gagal, orchestration dapat secara otomatis menggantinya dengan container baru untuk menjaga ketersediaan aplikasi. Orchestration juga memungkinkan pembaruan aplikasi dilakukan secara bertahap (misalnya, *rolling update*) untuk meminimalkan downtime. Pembaruan dapat dilakukan pada sebagian container saja sehingga pengguna tidak terpengaruh saat perubahan dilakukan.

Penggunakan containerization dan orchestration secara bersama-sama juga bermanfaat dalam:

a. **Skalabilitas dan Fleksibilitas yang Tinggi.** Dengan containerization dan orchestration, aplikasi dapat ditingkatkan kapasitasnya sesuai dengan permintaan pengguna tanpa harus menambah server fisik.
b. **Pengurangan Risiko Downtime.** Orchestration mengelola container secara otomatis, memastikan pemulihan dari kegagalan dan penggantian container yang bermasalah, sehingga menjaga kelangsungan operasional aplikasi.
c. **Peningkatan Efisiensi Operasional.** Orchestration mengurangi kebutuhan pengelolaan manual untuk container-container yang jumlahnya besar, memungkinkan tim untuk fokus pada pengembangan daripada operasi.
d. **Keamanan dan Isolasi yang Lebih Baik.** Dengan setiap aplikasi atau komponen terpisah dalam container, risiko konflik dependensi dan akses tak diinginkan antar aplikasi bisa diminimalkan.

Secara garis besar, penggunaan *Containerization* dan *Orchestration* merupakan solusi efektif guna pengembangan aplikasi dalam skala besar. Containerization menjadikan aplikasi lebih portabel dan efisien, sedangkan *orchestration* mengelola deployment dan penskalaan container secara otomatis. Kedua pendekatan ini bekerja bersama untuk mendukung infrastruktur

yang fleksibel, scalable, dan handal, membuatnya sangat cocok untuk perusahaan yang menginginkan inovasi cepat dan operasi yang berkelanjutan.

3.4.3 *Infrastructure as Code* (IaC)

Infrastructure as Code (IaC) merupakan pendekatan dalam mengelola dan mengatur infrastruktur teknologi informasi dengan menggunakan file konfigurasi atau kode yang bisa dieksekusi, bukan melalui pengaturan manual. Dengan IaC, tim dapat mengotomatisasi penyediaan, konfigurasi, dan pengelolaan infrastruktur seperti server, jaringan, basis data, dan lingkungan aplikasi menggunakan skrip dan alat otomatisasi.

IaC adalah bagian penting dari DevOps dan *Continuous Delivery* (CD), yang mendukung tim IT dalam membuat infrastruktur yang lebih konsisten, skalabel, dan dapat diulangi. Konsep utama dalam IaC, adalah:

1. Deklaratif dan Imperatif. Dimana deklaratif (*Declarative*), merupakan pendekatan yang memungkinkan Anda mendefinisikan "apa" yang diinginkan tanpa mendetailkan "bagaimana" langkah-langkah untuk mencapainya. Contoh: menulis konfigurasi yang menyatakan bahwa aplikasi membutuhkan tiga server, tanpa menjelaskan proses pembuatan server tersebut. Sedangkan Imperatif (Imperative) merupakan pendekatan yang menjelaskan "bagaimana" cara mencapainya, dengan langkah-langkah yang lebih terperinci. Contoh: mengonfigurasi setiap langkah pembuatan server secara eksplisit.
2. Idempotensi, yang merupakan kemampuan untuk menjalankan kode atau konfigurasi yang sama berulang kali tanpa mengubah hasil akhir. Dalam IaC, idempotensi memastikan bahwa menjalankan skrip berkali-kali akan selalu menghasilkan infrastruktur yang sama, terlepas dari kondisi awalnya.
3. Versi dan Kontrol. Dengan IaC, konfigurasi infrastruktur dikelola sebagai kode dan disimpan dalam sistem kontrol versi seperti Git. Hal ini memungkinkan tim melacak

perubahan, membandingkan versi, dan memulihkan konfigurasi sebelumnya dengan mudah.

IaC memungkinkan pengelolaan infrastruktur secara otomatis, konsisten, dan terukur, menjadikannya bagian penting dari pendekatan DevOps modern. Dengan alat seperti Terraform, Ansible, dan *CloudFormation*, tim dapat mendefinisikan, menyebarkan, dan mengelola infrastruktur dengan pendekatan yang mirip dengan pengembangan perangkat lunak, sehingga meningkatkan kecepatan, ketepatan, dan stabilitas dalam manajemen infrastruktur. Dalam implementasinya, Infrastructure as Code (IaC) biasanya melibatkan beberapa langkah, yatu:

1. Definisi infrastruktur sebagai kode, dengan menggunakan file konfigurasi atau skrip kode, termasuk rincian tentang server, jaringan, penyimpanan, dan konfigurasi keamanan yang diperlukan untuk aplikasi atau lingkungan tertentu.
2. Pengujian Konfigurasi. Sebelum digunakan, konfigurasi dapat diuji dalam lingkungan staging atau simulasi untuk memastikan infrastruktur yang akan disiapkan sesuai harapan dan tidak menyebabkan konflik.
3. Deployment Otomatis. Setelah diuji, kode konfigurasi digunakan untuk membuat infrastruktur secara otomatis, menghilangkan konfigurasi manual yang rentan terhadap kesalahan. Deployment ini dilakukan melalui pipeline CI/CD atau alat manajemen infrastruktur yang mendukung IaC.
4. Pemantauan dan Perbaikan Infrastruktur yang dikelola dengan IaC dipantau untuk memastikan kestabilan dan keandalannya. Jika ada perubahan atau perbaikan yang diperlukan, kode konfigurasi dapat diperbarui dan dijalankan kembali untuk memperbarui infrastruktur.

Berbagai manfaat Infrastructure as Code dapat dijelaskan sebagai berikut :

1. IaC mengotomatiskan penyediaan dan konfigurasi infrastruktur, mengurangi waktu yang dibutuhkan untuk mengatur server, jaringan, atau basis data secara manual.

Infrastruktur yang biasanya membutuhkan waktu berhari-hari bisa disiapkan dalam hitungan menit.
2. Karena infrastruktur didefinisikan dalam bentuk kode, konfigurasi yang sama dapat diterapkan berulang kali dengan hasil yang konsisten, sehingga dapat menghilangkan kesalahan konfigurasi manual.
3. Versi dan Pembaruan yang Terkelola dengan Baik dengan adanya kontrol versi. IaC memungkinkan tim melacak perubahan konfigurasi, memulihkan ke konfigurasi sebelumnya, dan mengelola infrastruktur seperti proyek perangkat lunak.
4. Karena infrastruktur otomatis, risiko kesalahan manusia dalam pengaturan dan pengelolaan berkurang secara signifikan. Otomatisasi juga mengurangi gangguan operasional yang bisa terjadi akibat kesalahan manual.
5. Skalabilitas dan Fleksibilitas. IaC memungkinkan infrastruktur dengan cepat diskalakan sesuai permintaan, mendukung aplikasi yang membutuhkan sumber daya tambahan dalam waktu singkat.

3.4.4 Monitoring dan Logging

Monitoring dan Logging merupakan dua praktik penting dalam manajemen infrastruktur dan aplikasi yang membantu tim TI dan DevOps memastikan performa, stabilitas, dan keamanan sistem. Kedua metode ini bekerja sama untuk mendeteksi masalah, mengidentifikasi penyebab, dan mengoptimalkan kinerja sistem secara proaktif.

Monitoring (pemantauan) adalah proses pengumpulan dan analisis data dari sistem atau aplikasi secara real-time untuk menilai kinerjanya. Monitoring bertujuan mendeteksi potensi masalah dan memberikan peringatan (alert) ketika terjadi perubahan yang tidak diinginkan, seperti lonjakan pemakaian CPU, memori yang berlebihan, atau waktu respons aplikasi yang lambat. Jenis monitoring dibedakan atas:

1. *Application Performance Monitoring* (APM), berfungsi memantau kinerja aplikasi, seperti waktu respons, throughput, dan error rate. Alat APM membantu

pengembang melihat metrik di setiap lapisan aplikasi, seperti aplikasi web, API, atau database, sehingga masalah dapat segera ditemukan dan diperbaiki.
2. *Infrastructure Monitoring*, berfungsi memantau server, database, jaringan, dan komponen infrastruktur lainnya. Metrik yang diperoleh meliputi penggunaan CPU, memori, penyimpanan, dan bandwidth.
3. *Network Monitoring*, berfungsi memantau aktivitas mencurigakan dalam sistem, seperti percobaan akses tak sah atau anomali pada aliran data. Monitoring ini penting untuk mencegah ancaman keamanan dan serangan terhadap sistem.
4. *User Experience Monitoring* (UEM), yang mengukur pengalaman pengguna akhir dengan melacak metrik seperti waktu loading halaman, interaksi, dan kesalahan antarmuka pengguna. UEM membantu tim memahami bagaimana aplikasi berfungsi dari perspektif pengguna.

Monitoring berguna dalam pendeteksian dan pencegahan timbulnya kendala sejak dini, memungkinkan tim mengetahui potensi masalah lebih awal dan bertindak sebelum berdampak ke pengguna. Monitoring jug dapat memberikan data real-time tentang kinerja sistem membantu tim memahami beban dan pemakaian, memungkinkan optimalisasi dan perencanaan kapasitas yang lebih baik. Selain itu, monitoring yang fokus pada UX (*User Experience*) memungkinkan tim menemukan area aplikasi yang memerlukan peningkatan agar pengguna mendapatkan pengalaman yang lebih baik. Dengan alert otomatis pada sistem monitoring, tim dapat segera merespons jika terjadi masalah, mengurangi waktu downtime dan meningkatkan keandalan sistem.

Logging adalah proses pencatatan data atau kejadian (event) yang terjadi dalam sistem atau aplikasi. Log mencatat informasi rinci, seperti waktu kejadian, jenis kesalahan, dan status sistem, yang dapat digunakan untuk melacak aktivitas atau melakukan analisis masalah. Jenis logging, dibedakan menjadi:

1. Application Log, merupakan catatan yang berisi informasi tentang aktivitas internal aplikasi, seperti proses yang

dijalankan, error, atau data transaksi. Application log sangat membantu dalam memecahkan masalah spesifik aplikasi.
2. System Log, yang mencatat kejadian pada sistem operasi atau perangkat keras, seperti aktivitas login, akses jaringan, atau perubahan konfigurasi. System log penting untuk memantau keamanan dan performa sistem secara keseluruhan.
3. Security Log, yang berisi catatan aktivitas keamanan, seperti percobaan akses yang tidak sah, login gagal, atau modifikasi data sensitif. Security log digunakan untuk mendeteksi potensi serangan dan menjaga kepatuhan dengan standar keamanan.
4. Audit Log, yang merupakan Log yang melacak aktivitas pengguna dan perubahan pada sistem atau data. Audit log berguna untuk analisis forensik atau penilaian kepatuhan.

Logging bermanfaat dalam menganalisis dan menyelesaikan timbulnya masalah (kendala) secara cepat. Log menyediakan data rinci tentang kesalahan atau kejadian tertentu, membantu tim dalam menganalisis akar penyebab dan memperbaiki masalah lebih cepat. Log bisa menjadi bukti yang menunjukkan bahwa sistem telah beroperasi sesuai standar kepatuhan dan membantu perusahaan menjaga rekam jejak keamanan. Logging juga dapat meningkatkan keamanan dengan mencatat percobaan akses yang tidak sah dan aktivitas mencurigakan, sehingga memungkinkan tim keamanan merespons dengan cepat. Logging dapat melacak perubahan sistem dengan menyediakan sejarah perubahan, membantu tim TI untuk melacak modifikasi yang dapat mempengaruhi performa atau stabilitas sistem.

Monitoring dan Logging bekerja sama, saling melengkapi dalam memberikan pemahaman menyeluruh tentang performa, stabilitas, dan keamanan sistem. Kinerja keduanya ditunjukkan dalam:

1. Deteksi dan Respon Cepat (Monitoring). Monitoring memberikan gambaran umum tentang kondisi sistem secara real-time dan memberi peringatan jika terjadi

masalah. Misalnya, jika CPU mendadak naik tajam, monitoring akan memberi alert.
2. Analisis dan Pemecahan Masalah (Logging). Setelah menerima alert, log dapat diperiksa untuk menemukan rincian yang menyebabkan kenaikan CPU. Log memberikan data spesifik yang membantu tim menganalisis dan menyelesaikan masalah dengan cepat.
3. Pengelolaan Berkelanjutan. Monitoring dan logging memberikan umpan balik yang berkelanjutan, membantu tim TI mengoptimalkan dan memelihara sistem.

Sebagai contoh, implementasi monitoring dan logging dapat dilihat pada sebuah aplikasi e-commerce yang menerima ribuan permintaan setiap menit menggunakan kombinasi monitoring dan logging untuk memastikan kelancaran operasi, melalui:

1. Monitoring mengawasi metrik seperti waktu respons server, jumlah permintaan per detik, dan error rate. Jika waktu respons melebihi batas, monitoring akan mengirimkan alert kepada tim.
2. Logging merekam detail setiap permintaan dan error, mencatat kejadian tertentu yang dapat dianalisis oleh tim ketika mereka melihat alert dari monitoring. Ini memungkinkan mereka menemukan masalah di level aplikasi atau basis data dengan lebih cepat.

3.5 *Service-Oriented Architecture* (SOA) dan *Cloud Computing*

Pendekatan berbasis arsitektur yang berpusat pada layanan, atau biasa disebut sebagai SOA (*Service-Oriented Architecture)* dan cloud computing merupakan dua pendekatan yang populer untuk membangun sistem informasi modern yang fleksibel, dapat diskalakan, dan mudah diintegrasikan. Keduanya memainkan peran penting dalam meningkatkan efisiensi, mengurangi biaya, dan memungkinkan perusahaan untuk beradaptasi dengan perubahan teknologi dengan lebih cepat.

Service-Oriented Architecture (SOA) adalah pendekatan desain arsitektur yang berfokus pada penyusunan aplikasi sebagai kumpulan layanan yang saling berkomunikasi dan bekerja sama. SOA memungkinkan komponen aplikasi atau modul yang berbeda berfungsi secara independen tetapi tetap terintegrasi, karena setiap modul atau layanan menyediakan fungsionalitas spesifik dan memiliki antarmuka yang jelas. Dalam SOA, layanan dibuat agar tidak bergantung erat satu sama lain, sehingga satu layanan dapat diganti atau dimodifikasi tanpa memengaruhi layanan lain. Ini mendukung fleksibilitas dan kemudahan pemeliharaan. SOA menggunakan antarmuka terstandar seperti API atau protokol khusus (misalnya, SOAP atau REST) untuk memungkinkan komunikasi antar layanan, terlepas dari bahasa pemrograman atau platform yang digunakan. Selain itu, layanan-layanan dalam SOA dapat digunakan kembali oleh aplikasi atau modul lain tanpa memerlukan penulisan ulang kode, yang membantu menghemat waktu dan sumber daya. Hal ini memungkinkan layanan-layanan disusun menjadi aplikasi kompleks dengan mengatur alur proses bisnis secara fleksibel, sehingga berbagai layanan dapat dikombinasikan dan dikelola secara otomatis. Layanan-layanan SOA biasanya dirancang berdasarkan fungsi bisnis tertentu, seperti pembayaran, autentikasi, atau pengelolaan inventaris, sehingga layanan tersebut mudah diakses dan diimplementasikan di berbagai aplikasi.

Cloud Computing adalah model komputasi yang menyediakan akses fleksibel dan skalabel ke sumber daya komputasi melalui internet. Cloud memungkinkan perusahaan untuk menyewa sumber daya TI sesuai kebutuhan tanpa harus memiliki dan mengelola infrastruktur sendiri.

Model layanan dalam cloud computing dapat dijelaskan sebagai berikut:

1. *Infrastructure as a Service* (IaaS) IaaS menyediakan infrastruktur dasar seperti server, penyimpanan, dan jaringan yang dapat digunakan sesuai kebutuhan. Contoh: Amazon Web Services (AWS) EC2, Google Cloud Compute Engine.

2. *Platform as a Service* (PaaS) PaaS menyediakan lingkungan untuk pengembangan dan penyebaran aplikasi yang dilengkapi dengan layanan seperti basis data, middleware, dan alat pengembangan. Contoh: Google App Engine, Microsoft Azure.
3. *Software as a Service* (SaaS) SaaS memungkinkan pengguna mengakses aplikasi yang disediakan secara online oleh penyedia layanan. Contoh: Google Workspace, Salesforce, Microsoft Office 365.

SOA dan cloud computing bisa diintegrasikan untuk memberikan solusi yang lebih kuat dan efisien. Layanan yang dibangun dengan pendekatan SOA dapat disimpan dan dijalankan di cloud. Cloud menyediakan sumber daya yang fleksibel dan andal untuk layanan-layanan ini, yang mendukung skalabilitas dan ketersediaan yang lebih tinggi. API memungkinkan komunikasi antara layanan SOA di cloud dan aplikasi lainnya, membuat integrasi antar sistem lebih mudah dan meningkatkan interoperabilitas. SOA dapat memanfaatkan container dan orkestrasi seperti Kubernetes di cloud untuk pengelolaan dan skalabilitas layanan yang lebih mudah. Containerisasi memungkinkan layanan SOA disebarkan dengan lebih cepat, sementara cloud memberikan infrastruktur elastis untuk mendukung aplikasi container.

3.6 Desain Sistem Berfokus pada Pengguna (*User-Centered Design* - UCD)

Desain sistem berfokus pada pengguna adalah metode yang menekankan pentingnya keterlibatan pengguna dalam setiap tahap pengembangan sistem, memprioritaskan kebutuhan, preferensi, dan keterlibatan pengguna akhir pada setiap tahap pengembangan. Konsep ini berpandangan bahwa desain sistem yang efektif hanya dapat dicapai melalui pemahaman mendalam tentang kebutuhan, tujuan, termasuk perilaku, tujuan, dan hambatan mereka, untuk menciptakan sistem yang mudah digunakan dan memberikan pengalaman yang memuaskan bagi pengguna yang akan memanfaatkan sistem tersebut. Menurut *Norman* (2018), dalam pendekatan ini, pemahaman tentang perilaku pengguna, kebutuhan,

dan pola interaksi menjadi fokus utama, yang kemudian diterjemahkan ke dalam elemen-elemen desain yang intuitif dan mudah digunakan.

Dalam beberapa tahun terakhir, pendekatan UCD terus diperkuat oleh temuan penelitian yang menunjukkan peningkatan kepuasan dan kinerja pengguna ketika mereka dilibatkan dalam proses desain. Penelitian oleh *Brhel et al.* (2021) menunjukkan bahwa organisasi yang menerapkan prinsip UCD cenderung mendapatkan tingkat adopsi yang lebih tinggi serta umpan balik positif dari pengguna.

Selain itu, studi lain oleh *Gulliksen dan Göransson* (2022) menyebutkan bahwa UCD bukan hanya berdampak pada antarmuka pengguna tetapi juga membantu dalam menyelaraskan sistem dengan konteks penggunaan spesifik, seperti budaya kerja atau kebutuhan industri. Desain yang menyesuaikan diri dengan konteks pengguna memungkinkan organisasi untuk mencapai tujuan bisnis yang lebih optimal.

Pendekatan UCD umumnya mencakup beberapa tahap, seperti berikut:

1. Analisis Pengguna
   Pada tahap ini, pengembang melakukan riset mendalam untuk memahami siapa pengguna sistem dan apa yang mereka butuhkan dari sistem tersebut. Teknik seperti wawancara, observasi, dan survei sering digunakan. Misalnya, menyoroti pentingnya melibatkan pengguna dalam fase analisis awal untuk memastikan kebutuhan mereka terakomodasi dalam rancangan sistem.
2. Pembuatan Persona dan Scenarios
   Persona adalah profil fiktif yang mewakili tipe pengguna sistem tertentu, sedangkan skenario adalah narasi pendek tentang bagaimana persona tersebut akan menggunakan sistem. Hal ini membantu tim pengembangan untuk fokus pada pengalaman pengguna yang spesifik. Persona dapat meningkatkan efektivitas komunikasi dalam tim pengembangan, serta membuat desain lebih relevan dengan kebutuhan pengguna.

3. Prototyping dan Uji Coba Pengguna (*Usability Testing*)
   Pada tahap ini, desainer membuat prototipe sistem untuk diuji langsung oleh pengguna. Prototipe dapat berupa *wireframe*, *mockup*, atau bahkan versi fungsional sistem. Uji coba ini penting untuk mengidentifikasi masalah-masalah antarmuka atau interaksi yang mungkin dialami pengguna. Pengujian usability memungkinkan deteksi dini terhadap hambatan pengguna, sehingga perbaikan dapat dilakukan sebelum sistem benar-benar diimplementasikan.
4. Iterasi dan Penyempurnaan
   Berdasarkan masukan dari uji coba pengguna, tim melakukan perubahan dan penyempurnaan terhadap desain. Proses iterasi ini memungkinkan sistem menjadi lebih matang dan selaras dengan kebutuhan pengguna yang terus berkembang. Beberapa studi menggarisbawahi bahwa iterasi berulang dalam UCD menghasilkan antarmuka yang lebih intuitif dan meningkatkan engagement pengguna

DAFTAR PUSTAKA

Srivastava, A., Bhardwaj, S., & Saraswat, S. 2017. *SCRUM model for agile methodology.* In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864-869). IEEE.

Hayat, F., Rehman, A. U., Arif, K. S., Wahab, K., & Abbas, M. 2019. *The influence of agile methodology (Scrum) on software project management*. In 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 145-149). IEEE.

Wonohardjo, E. P., Sunaryo, R. F., & Sudiyono, Y. (2019). A systematic review of SCRUM in software development. *JOIV: International Journal on Informatics Visualization*, *3*(2), 108-112.

Fagarasan, C., Popa, O., Pisla, A., & Cristea, C. (2021, August). Agile, waterfall and iterative approach in information technology projects. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1169, No. 1, p. 012025). IOP Publishing.

Pichler, R. 2010. Agile Product Management with Scrum: Creating Products That Customers Love. Addison Wesley.

Swisher, W. P. 2014. Implementing Scrumban. Retrieved September 10, 2016, from https://switchingtoscrum.files.wordpress.com/2013/12/im ple menting-scrumban_v1-32.pdf.

VersionOne. 2016. VersionOne 10th Annual State of Agile Report.

Brezočnik, L., & Majer, Č. 2016. *Comparison of agile methods: Scrum, Kanban, and Scrumban*. In *Proc. 19th Int. Multiconference Inf. Soc* (Vol. 100, No. July, pp. 1-5).

Brechner, E. 2015. *Agile project management with Kanban*. Pearson Education.

Klipp P. 2014. *Getting Started with Kanban.* Amazon Digital Services LLC.

Brezočnik, L., & Majer, Č. 2016. *Comparison of agile methods: Scrum, Kanban, and Scrumban.* In Proc. 19th Int. Multiconference Inf. Soc (Vol. 100, No. July, pp. 1-5).

BAB 4
TEKNOLOGI MODERN DALAM DESAIN SISTEM TI

4.1 Pendahuluan

Teknologi informasi (TI) telah berkembang pesat dalam beberapa dekade terakhir, mengubah cara kita bekerja, berkomunikasi, dan bahkan berinteraksi dengan dunia. Salah satu aspek paling menarik dari perkembangan teknologi ini adalah bagaimana kemajuan tersebut memengaruhi desain sistem TI. Desain sistem TI, yang pada dasarnya adalah proses merancang dan membangun struktur perangkat lunak dan perangkat keras yang mendukung suatu organisasi atau aplikasi (Álvarez D., Buonaffina F. and Zabala A., 2014; Dzhamaldinova, Kurdyukova and Kunanbayeva, 2021), kini semakin didorong oleh teknologi-teknologi modern yang mampu memberikan solusi yang lebih efisien, fleksibel, dan scalable .

Dalam konteks ini, "teknologi modern" merujuk pada inovasi-inovasi terbaru yang memengaruhi desain dan pengembangan sistem TI. Beberapa teknologi utama yang memainkan peran penting dalam desain sistem TI saat ini termasuk *cloud computing* (Kampichler and Eier, 2014), big data, kecerdasan buatan (AI), *Internet of Things* (IoT), dan blockchain. Setiap teknologi ini tidak hanya menawarkan cara baru untuk mengelola dan memproses data, tetapi juga membuka peluang baru dalam hal efisiensi operasional, pengalaman pengguna, dan pengambilan keputusan berbasis data.

Peran teknologi modern dalam desain sistem TI sangatlah besar, karena mereka memberikan alat dan framework yang diperlukan untuk membangun sistem yang lebih cepat, lebih kuat, dan lebih aman. Seringkali, penerapan teknologi-teknologi ini dalam desain sistem TI akan memberikan fleksibilitas tambahan yang sangat dibutuhkan oleh perusahaan untuk bersaing di pasar global yang semakin dinamis (Lovrekovic and Sukic, 2011; Juwono *et al.*,

2014). Misalnya, dengan memanfaatkan cloud computing, perusahaan dapat memperluas infrastruktur TI mereka tanpa perlu berinvestasi besar dalam perangkat keras (Kampichler and Eier, 2014). Big data dan analitik memungkinkan perusahaan untuk membuat keputusan yang lebih tepat dengan mengolah data dalam jumlah besar dan menggunakan wawasan yang diperoleh untuk merancang sistem yang lebih responsif dan adaptif.

Sebagai tambahan, AI dan machine learning membawa kemampuan untuk otomatisasi dan pengambilan keputusan cerdas dalam desain sistem TI (Chamberlain *et al.*, 2020). Teknologi-teknologi ini memungkinkan sistem untuk belajar dari data yang dikumpulkan dan secara mandiri meningkatkan kinerjanya, menciptakan pengalaman pengguna yang lebih baik dan mengurangi beban kerja manusia. Demikian pula, IoT menghubungkan perangkat yang sebelumnya terpisah dalam satu ekosistem terintegrasi, memungkinkan sistem untuk berfungsi lebih efisien melalui pengumpulan dan analisis data real-time dari berbagai sensor dan perangkat.

Meskipun teknologi-teknologi ini menawarkan banyak manfaat, namun juga membawa tantangan baru dalam desain sistem TI. Isu seperti keamanan, privasi, dan keterbatasan infrastruktur sering kali menjadi hambatan utama dalam penerapan teknologi modern ini dalam desain sistem TI. Oleh karena itu, sangat penting bagi para profesional TI, terutama mahasiswa Informatika, untuk memahami tidak hanya manfaat, tetapi juga tantangan yang terkait dengan teknologi-teknologi ini.

Bab ini akan membahas secara mendalam tentang berbagai teknologi modern yang membentuk desain sistem TI, mulai dari cloud computing hingga IoT, serta bagaimana teknologi-teknologi tersebut dapat diintegrasikan untuk menciptakan sistem TI yang lebih efektif, efisien, dan aman. Dengan memahami peran dan aplikasi dari teknologi-teknologi ini, kita dapat merancang sistem TI yang lebih unggul dan siap menghadapi tantangan masa depan.

4.2 *Cloud Computing* dalam Desain Sistem TI

Salah satu teknologi yang paling mengubah cara kita mendesain dan mengelola sistem TI adalah cloud computing. Cloud computing memungkinkan organisasi untuk mengakses, menyimpan, dan memproses data melalui internet, tanpa memerlukan infrastruktur fisik yang besar di lokasi pengguna. Dengan kemudahan akses, fleksibilitas, dan biaya yang lebih rendah, cloud computing menjadi teknologi utama dalam desain sistem TI modern.

Keuntungan Cloud Computing dalam Desain Sistem TI

Cloud computing menawarkan sejumlah keuntungan signifikan dalam desain sistem TI, terutama dalam hal skalabilitas, fleksibilitas, dan biaya. Beberapa keuntungan utama yang dapat dimanfaatkan oleh desainer sistem TI adalah:

1. **Skalabilitas yang Mudah**
   Salah satu fitur utama dari cloud computing adalah skabilitas. Dalam desain sistem TI, kemampuan untuk menambah kapasitas atau mengurangi sumber daya sesuai kebutuhan sangat penting. Dengan cloud computing, organisasi dapat dengan mudah meningkatkan atau menurunkan kapasitas penyimpanan, pengolahan data, atau kapasitas jaringan tanpa perlu mengkhawatirkan infrastruktur fisik yang rumit. Desain sistem TI yang dibangun di cloud memberikan fleksibilitas yang sangat besar untuk menyesuaikan dengan permintaan yang berubah-ubah.
2. **Efisiensi Biaya**
   Cloud computing mengurangi kebutuhan akan infrastruktur fisik yang mahal, seperti server dan perangkat keras lainnya. Sebagai gantinya, pengguna hanya membayar untuk sumber daya yang mereka gunakan, yang menjadikannya sebagai model biaya yang lebih efisien. Dalam desain sistem TI, cloud dapat mengurangi biaya awal yang signifikan dan biaya pemeliharaan yang berkelanjutan.
3. **Akses Data Secara Real-Time**
   Dengan *cloud computing*, data dapat diakses dari mana saja dan kapan saja, asalkan pengguna memiliki koneksi internet. Dalam

desain sistem TI yang memanfaatkan cloud, hal ini memungkinkan kolaborasi yang lebih baik antara tim yang berada di lokasi berbeda dan memungkinkan pengambilan keputusan yang lebih cepat.

4. **Keamanan yang Terintegrasi**
   Penyedia layanan cloud, seperti *Amazon Web Services* (AWS), Microsoft Azure, dan Google Cloud, menawarkan infrastruktur keamanan yang kuat, termasuk enkripsi data, otentikasi multi-faktor, dan pemantauan 24/7. Hal ini memungkinkan organisasi untuk memanfaatkan infrastruktur keamanan yang lebih baik daripada yang dapat mereka bangun secara internal.

Jenis-jenis *Cloud Computing* dalam Desain Sistem TI

Ada tiga model utama *cloud computing* yang dapat dipilih oleh desainer sistem TI, yaitu *Infrastructure-as-a-Service* (IaaS), *Platform-as-a-Service* (PaaS), dan *Software-as-a-Service* (SaaS). Masing-masing model ini menawarkan tingkat kontrol dan fleksibilitas yang berbeda.

1. ***Infrastructure-as-a-Service* (IaaS)**
   IaaS adalah model cloud yang menyediakan infrastruktur TI dasar, seperti server, penyimpanan, dan jaringan. Pengguna dapat mengelola dan mengonfigurasi perangkat keras virtual yang disediakan oleh penyedia cloud. Dalam desain sistem TI, IaaS memungkinkan organisasi untuk membangun dan mengelola lingkungan IT mereka sendiri dengan sumber daya yang sangat dapat disesuaikan.
2. ***Platform-as-a-Service* (PaaS)**
   PaaS menyediakan platform dan alat yang diperlukan untuk mengembangkan, menguji, dan meluncurkan aplikasi. PaaS mengurangi beban pengelolaan infrastruktur dengan menyediakan platform yang terintegrasi untuk pengembangan aplikasi. Dalam desain sistem TI, PaaS sangat berguna bagi pengembang yang ingin fokus pada pembuatan aplikasi tanpa khawatir tentang manajemen server atau perangkat keras lainnya.

3. ***Software-as-a-Service* (SaaS)**
   SaaS adalah model di mana perangkat lunak dan aplikasi disediakan sebagai layanan yang dapat diakses melalui internet. Pengguna dapat mengakses aplikasi ini melalui web browser tanpa perlu menginstalnya di perangkat mereka. Dalam desain sistem TI, SaaS memberikan solusi yang sangat hemat biaya untuk aplikasi perangkat lunak yang sudah terintegrasi dengan baik dan dapat diakses kapan saja. Contoh SaaS termasuk Google Workspace, Microsoft 365, dan Salesforce.

Penggunaan *Cloud Computing* dalam Desain Sistem TI Modern

Cloud computing memainkan peran penting dalam memungkinkan desain sistem TI yang lebih fleksibel dan dapat disesuaikan. Beberapa area penerapan cloud computing dalam desain sistem TI antara lain:

1. **Data Analytics**
   Cloud computing memungkinkan pengolahan dan analisis data besar secara lebih efisien. Sistem yang dibangun di cloud dapat menangani volume data yang sangat besar dan melakukan analisis secara real-time, memberi perusahaan wawasan yang lebih cepat dan lebih baik. Misalnya, dalam sistem manajemen pelanggan (CRM), data yang terkumpul dari berbagai titik interaksi dengan pelanggan dapat dianalisis untuk menghasilkan analitik yang mendalam dan personalisasi layanan.
2. **Penyimpanan Data dan Backup** Cloud storage menawarkan solusi penyimpanan data yang lebih terjangkau dan dapat diakses secara global. Desain sistem TI dapat memanfaatkan cloud untuk penyimpanan cadangan dan pemulihan data, memastikan bahwa data penting tetap aman dan dapat dipulihkan dalam situasi darurat. Model *Cloud Disaster Recovery* (CDR) sangat berguna untuk menjaga kontinuitas bisnis, dengan data yang selalu tersedia di cloud dan dapat dipulihkan dengan cepat.
3. **Kolaborasi dan Pengelolaan Proyek** Cloud computing memungkinkan kolaborasi lebih efektif antara tim yang terpisah oleh ruang fisik. Alat seperti Google Docs, Trello, atau

Slack memungkinkan tim untuk bekerja bersama secara real-time, berbagi file, berkomunikasi, dan mengelola proyek secara efisien. Dalam desain sistem TI, penggunaan platform cloud ini dapat meningkatkan produktivitas tim pengembang dan mengurangi hambatan komunikasi.

4. **Keamanan dan Kepatuhan Data** Keamanan data merupakan masalah penting dalam desain sistem TI, terutama di sektor yang sangat teratur seperti keuangan dan kesehatan. Penyedia layanan cloud menyediakan lapisan keamanan yang terintegrasi dan sering kali memenuhi standar kepatuhan yang ketat, seperti General Data Protection Regulation (GDPR) atau Health Insurance Portability and Accountability Act (HIPAA). Sistem TI yang dibangun di cloud dapat memanfaatkan pengamanan ini, mengurangi risiko pelanggaran data dan meningkatkan kepatuhan terhadap regulasi yang berlaku.

Tantangan dalam Mengimplementasikan *Cloud Computing*

Meskipun *cloud computing* menawarkan berbagai keuntungan, implementasinya tetap membawa beberapa tantangan, antara lain:

1. Ketergantungan pada Penyedia Layanan Cloud
   Penggunaan cloud computing mengharuskan organisasi untuk bergantung pada penyedia layanan cloud untuk infrastruktur dan layanan. Hal ini bisa menjadi masalah jika penyedia mengalami gangguan layanan atau bahkan kebangkrutan. Oleh karena itu, desain sistem TI perlu memasukkan pertimbangan terkait pemulihan dan diversifikasi penyedia layanan untuk mengurangi risiko.
2. Keamanan dan Privasi Data
   Meskipun penyedia layanan cloud menawarkan keamanan tingkat tinggi, menyimpan data di cloud selalu menimbulkan kekhawatiran mengenai potensi pelanggaran privasi dan akses yang tidak sah. Sistem TI yang dibangun di cloud harus memperhatikan kontrol akses, enkripsi data, dan kebijakan privasi yang ketat untuk melindungi data sensitif.

3. Konektivitas Internet yang Stabil
   Karena cloud computing bergantung pada internet, akses yang buruk atau tidak stabil dapat mengganggu kinerja sistem. Dalam desain sistem TI, sangat penting untuk memastikan bahwa konektivitas internet yang digunakan cukup stabil untuk mendukung aplikasi dan sistem berbasis cloud.

Kesimpulan

Cloud computing telah merevolusi cara sistem TI dibangun dan dikelola, memungkinkan fleksibilitas, skalabilitas, dan efisiensi biaya. Dalam desain sistem TI modern, cloud computing memberikan kemampuan untuk mengakses sumber daya TI dengan cara yang lebih efisien dan lebih aman. Meskipun ada tantangan dalam implementasi dan pengelolaan sistem cloud, keuntungan yang diberikan menjadikan cloud computing sebagai teknologi yang sangat penting dalam desain dan pengelolaan sistem TI masa depan.

4.3 Big Data dan Analitik dalam Desain Sistem TI

Pada era digital saat ini, data telah menjadi salah satu aset paling berharga bagi organisasi di berbagai sektor. Seiring dengan semakin berkembangnya teknologi dan semakin banyaknya perangkat yang terhubung ke internet, volume, kecepatan, dan variasi data yang dihasilkan terus meningkat secara eksponensial. Dalam konteks desain sistem TI, munculnya konsep Big Data dan analitik data telah memberikan dampak besar terhadap cara sistem dirancang dan dijalankan. Big Data merujuk pada kumpulan data yang begitu besar dan kompleks sehingga tidak bisa diolah dengan metode tradisional. Namun, dengan penerapan teknologi yang tepat, data besar ini dapat digunakan untuk memperoleh wawasan yang sangat berharga, yang dapat meningkatkan pengambilan keputusan dan optimasi operasional dalam suatu organisasi.

Big Data terdiri dari tiga karakteristik utama yang dikenal dengan istilah 3V:

1. **Volume** – Merujuk pada jumlah data yang sangat besar yang perlu dikelola dan dianalisis. Data ini berasal dari berbagai sumber seperti transaksi bisnis, sensor IoT, media sosial, dan lainnya.

2. **Velocity** – Kecepatan data yang terus berubah dan berkembang dengan cepat. Data dihasilkan dalam waktu nyata, seperti transaksi online atau data sensor yang dikirim dalam interval waktu yang sangat pendek.
3. **Variety** – Variasi tipe data, yang dapat meliputi data terstruktur (seperti database relasional), data tidak terstruktur (seperti teks atau gambar), dan semi-terstruktur (seperti log file atau JSON).

Dengan adanya Big Data, desainer sistem TI kini harus memikirkan cara mengelola data yang sangat besar ini dengan efisien. Data yang datang dari berbagai sumber memerlukan sistem penyimpanan dan pengolahan yang dapat menangani beban besar, sambil memastikan kecepatan dan akurasi dalam pemrosesan data. Untuk itu, teknologi seperti Hadoop, Spark, dan NoSQL databases menjadi sangat penting dalam desain sistem TI yang berfokus pada Big Data.

Hadoop adalah framework open-source yang memungkinkan penyimpanan dan pemrosesan data dalam jumlah besar secara terdistribusi. Dengan memanfaatkan banyak komputer dalam satu jaringan, Hadoop dapat membagi data besar menjadi potongan-potongan yang lebih kecil dan memprosesnya secara paralel. Hal ini sangat berguna dalam mendesain sistem TI yang menangani volume data yang luar biasa besar, seperti yang terjadi dalam e-commerce atau sistem perbankan.

Sementara itu, Apache Spark adalah framework komputasi terdistribusi yang lebih cepat daripada Hadoop dalam memproses data real-time dan batch. Spark memungkinkan analisis data yang lebih cepat dan efisien, serta mendukung analisis berbasis memori, yang penting untuk aplikasi yang membutuhkan respon cepat. Dalam hal ini, Spark berperan penting dalam mendesain sistem TI yang mengandalkan kecepatan dalam pengambilan keputusan, seperti dalam sistem peringatan dini atau rekomendasi berbasis data.

NoSQL databases seperti MongoDB dan Cassandra menawarkan cara baru untuk menyimpan dan mengakses data dalam format yang tidak terstruktur, yang sangat cocok untuk Big

Data. Dengan kemampuannya untuk menangani data dalam skala besar, serta fleksibilitas dalam jenis data yang dapat disimpan, database NoSQL sering digunakan dalam desain sistem TI yang memerlukan skalabilitas tinggi dan responsivitas cepat.

4.4 Analitik Data dalam Desain Sistem TI

Selain pengelolaan data itu sendiri, analitik data memainkan peran yang sangat penting dalam desain sistem TI. Analitik data merujuk pada penggunaan berbagai teknik dan alat untuk mengeksplorasi, menganalisis, dan memperoleh wawasan dari data besar. Melalui analitik, organisasi dapat menemukan pola tersembunyi, tren yang muncul, dan hubungan yang sebelumnya tidak terlihat yang dapat mendukung keputusan bisnis yang lebih baik.

Tiga jenis analitik yang sering digunakan dalam sistem TI modern adalah:

1. ***Descriptive Analytics*** **(Analitik Deskriptif):** Analitik deskriptif berfokus pada pemahaman apa yang telah terjadi dalam data. Ini termasuk statistik dasar, seperti rata-rata, distribusi frekuensi, atau grafik yang menggambarkan tren masa lalu. Dalam desain sistem TI, analitik deskriptif dapat digunakan untuk menghasilkan laporan operasional atau untuk menganalisis hasil dari suatu aplikasi atau sistem.
2. ***Predictive Analytics*** **(Analitik Prediktif):** Analitik prediktif bertujuan untuk memprediksi apa yang akan terjadi di masa depan berdasarkan data masa lalu. Teknik-teknik seperti regresi, analisis klaster, dan machine learning digunakan untuk membuat prediksi tentang tren masa depan. Dalam desain sistem TI, analitik prediktif dapat digunakan untuk merancang aplikasi yang dapat memberikan rekomendasi atau peringatan dini, seperti dalam sistem rekomendasi produk atau sistem deteksi penipuan.
3. ***Prescriptive Analytics*** **(Analitik Preskriptif):** Analitik preskriptif memberikan rekomendasi untuk tindakan yang harus diambil berdasarkan analisis data. Teknik ini menggunakan model simulasi, algoritma optimisasi, dan AI untuk memberikan solusi terbaik dalam berbagai skenario.

Dalam desain sistem TI, analitik preskriptif dapat digunakan untuk merancang sistem yang tidak hanya menganalisis data, tetapi juga dapat memberikan rekomendasi atau bahkan mengambil keputusan otomatis, seperti dalam sistem pengelolaan inventaris atau sistem pengendalian lalu lintas.

Dengan meningkatnya volume data yang tersedia, kebutuhan untuk menganalisis dan menginterpretasikan data menjadi semakin penting. Oleh karena itu, desain sistem TI modern harus memperhitungkan kapasitas untuk menyimpan, mengelola, dan menganalisis Big Data secara efisien. Teknologi seperti machine learning, deep learning, dan AI kini digunakan untuk memperkuat kemampuan analitik dalam sistem TI, mengubah data besar menjadi wawasan yang dapat ditindaklanjuti (Pathak and Choudhary, 2023).

Studi Kasus: Big Data dalam E-Commerce dan Ritel

Di sektor e-commerce, penggunaan Big Data sangat meluas. Misalnya, Amazon dan Netflix menggunakan analitik prediktif untuk memberikan rekomendasi produk atau film berdasarkan preferensi pengguna. Dengan menggunakan algoritma yang memproses data besar dalam waktu nyata, perusahaan-perusahaan ini dapat memberikan pengalaman pengguna yang sangat dipersonalisasi, yang pada gilirannya meningkatkan kepuasan dan loyalitas pelanggan.

Dalam ritel, perusahaan seperti Walmart menggunakan analitik data untuk menganalisis tren pembelian pelanggan, memprediksi kebutuhan produk di lokasi tertentu, dan mengoptimalkan stok. Melalui penerapan analitik data, mereka mampu membuat keputusan berbasis data yang lebih baik dan lebih cepat, yang meningkatkan efisiensi operasional dan mengurangi pemborosan.

4.5 Kecerdasan Buatan dan Pembelajaran Mesin dalam Desain Sistem TI

Kecerdasan Buatan (AI) dan Pembelajaran Mesin (*Machine Learning*) merupakan dua komponen utama dalam revolusi teknologi saat ini, dan keduanya memiliki dampak yang besar

terhadap desain sistem TI modern. Keduanya memungkinkan sistem untuk belajar dari data, membuat prediksi, dan mengambil keputusan otomatis tanpa campur tangan manusia. Dalam konteks desain sistem TI, teknologi ini menawarkan kemampuan untuk menciptakan sistem yang lebih cerdas, adaptif, dan efisien.

AI mengacu pada simulasi proses kecerdasan manusia dalam mesin yang diprogram untuk berpikir, belajar, dan memecahkan masalah. Dalam desain sistem TI, AI digunakan untuk menciptakan aplikasi yang lebih pintar, seperti asisten virtual, pengenalan suara, dan sistem rekomendasi. Sistem berbasis AI dapat memproses data dalam jumlah besar untuk mengambil keputusan yang lebih baik dan lebih cepat daripada manusia, serta mendeteksi pola atau anomali yang sulit ditemukan dengan cara manual.

Salah satu contoh penerapan AI dalam desain sistem TI adalah sistem rekomendasi yang digunakan oleh perusahaan-perusahaan e-commerce dan platform media sosial. Amazon, misalnya, menggunakan AI untuk menganalisis riwayat belanja pengguna dan kemudian memberikan rekomendasi produk berdasarkan preferensi individu tersebut. AI juga digunakan dalam sistem pencarian untuk memberikan hasil yang relevan sesuai dengan kata kunci atau perilaku pengguna sebelumnya.

Dalam desain sistem TI, AI juga digunakan untuk menciptakan aplikasi pengenalan wajah dan pengenalan suara. Teknologi ini sangat berguna dalam berbagai bidang, termasuk keamanan (misalnya, untuk otentikasi pengguna) dan layanan pelanggan (misalnya, chatbot yang dapat memahami dan merespons pertanyaan pengguna). Sistem berbasis AI dapat mempelajari pola perilaku pengguna seiring waktu dan memberikan solusi atau jawaban yang lebih tepat berdasarkan pengalaman sebelumnya.

Meskipun AI memberikan banyak manfaat dalam desain sistem TI, ada beberapa tantangan yang perlu diatasi. Salah satunya adalah masalah bias dalam data. Jika data yang digunakan untuk melatih sistem AI memiliki bias atau ketidakakuratan, maka hasil yang dihasilkan juga akan terpengaruh. Oleh karena itu, penting bagi desainer sistem TI untuk memastikan bahwa data yang digunakan

dalam pelatihan AI adalah representatif dan bebas dari bias yang dapat mempengaruhi keputusan yang diambil oleh sistem.

4.6 *Machine Learning* dalam Desain Sistem TI

Machine Learning adalah bagian dari AI yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya tanpa diprogram secara eksplisit. Dalam pembelajaran mesin, algoritma digunakan untuk menganalisis pola dalam data dan membuat prediksi atau keputusan berdasarkan pola tersebut. Ada beberapa pendekatan dalam pembelajaran mesin, di antaranya adalah *supervised learning, unsupervised learning* dan *reinforcement learning*.

1. **Supervised Learning**
   Pada pendekatan ini, sistem diberi data yang telah diberi label atau kategori. Tujuan dari pembelajaran terawasi adalah untuk mengajarkan sistem untuk mengenali pola dalam data dan dapat membuat prediksi berdasarkan data baru yang belum dilabeli. Contoh aplikasi dari pembelajaran terawasi adalah sistem deteksi penipuan dalam transaksi kartu kredit, di mana sistem dilatih untuk mengenali pola-pola yang mengindikasikan adanya penipuan.
2. **Unsupervised Learning**
   Pada pembelajaran UL, sistem diberikan data yang tidak dilabeli dan harus menemukan pola atau struktur sendiri. Salah satu contoh penggunaannya adalah klasterisasi data pelanggan untuk menemukan segmen pasar yang berbeda berdasarkan kebiasaan atau preferensi mereka. Pembelajaran tidak terawasi juga digunakan dalam analisis sentimen di media sosial untuk mengidentifikasi tema atau topik yang sedang tren.
3. **Reinforcement Learning**
   Reinforcement Learning adalah jenis pembelajaran mesin di mana sistem belajar dengan mencoba tindakan dan menerima umpan balik positif atau negatif berdasarkan hasil dari tindakan tersebut. Teknologi ini digunakan dalam pengembangan robotika dan sistem permainan komputer seperti AlphaGo, yang dapat mengalahkan pemain manusia dalam permainan Go. Dalam desain sistem TI, pembelajaran penguatan dapat

digunakan untuk menciptakan aplikasi yang dapat belajar dan beradaptasi dengan situasi yang berubah seiring waktu.

Penerapan Pembelajaran Mesin dalam Sistem TI

Pembelajaran mesin dapat diterapkan dalam berbagai jenis sistem TI untuk meningkatkan kinerja dan memberikan wawasan yang lebih baik. Beberapa contoh penerapan ML dalam desain sistem TI meliputi:

1. **Sistem Pengenalan Gambar dan Video**
   Pembelajaran mesin digunakan dalam aplikasi seperti pengenalan wajah, deteksi objek dalam video, dan sistem pemantauan video cerdas. Teknologi ini dapat digunakan dalam bidang keamanan untuk mendeteksi perilaku mencurigakan di tempat umum atau di fasilitas sensitif.
2. **Sistem Prediksi**
   Di industri keuangan, sistem berbasis pembelajaran mesin digunakan untuk menganalisis tren pasar dan memprediksi harga saham atau mata uang digital. Teknologi ini memungkinkan pengambilan keputusan yang lebih cepat dan lebih tepat berdasarkan data historis dan analisis prediktif.
3. **Asisten Virtual dan Chatbots**
   Pembelajaran mesin juga digunakan dalam pembuatan asisten virtual dan chatbots yang dapat berinteraksi dengan pengguna dan memberikan bantuan otomatis. Sistem ini dapat mempelajari cara berbicara dan memahami pertanyaan pengguna untuk memberikan jawaban yang lebih relevan dan personal.

Tantangan dalam Implementasi AI dan Pembelajaran Mesin dalam Desain Sistem TI

Meskipun teknologi AI dan pembelajaran mesin menawarkan banyak manfaat, ada beberapa tantangan yang harus dihadapi oleh desainer sistem TI dalam mengimplementasikannya:

1. **Keterbatasan Data**: Salah satu tantangan utama dalam pembelajaran mesin adalah kurangnya data berkualitas yang dapat digunakan untuk melatih model. Sistem AI dan ML

membutuhkan data dalam jumlah besar dan berkualitas tinggi untuk dapat memberikan hasil yang akurat dan andal.

2. **Keamanan dan Privasi**: Penggunaan AI dan ML dalam desain sistem TI membawa tantangan terkait dengan keamanan dan privasi data. Data yang digunakan untuk melatih model sering kali bersifat sensitif, dan penting untuk memastikan bahwa data tersebut tidak disalahgunakan atau jatuh ke tangan yang salah.
3. **Etika dan Bias**: Seperti yang disebutkan sebelumnya, sistem AI dan ML dapat terpengaruh oleh bias yang ada dalam data. Oleh karena itu, penting untuk memastikan bahwa model yang dikembangkan adil, transparan, dan tidak memperburuk ketidaksetaraan yang ada.

4.7 *Internet of Things* (IoT) dalam Desain Sistem TI

Internet of Things (IoT) merupakan salah satu inovasi teknologi terbesar di abad ke-21, yang memungkinkan perangkat fisik untuk saling terhubung dan bertukar data melalui jaringan internet. Teknologi ini mengubah cara kita berinteraksi dengan dunia di sekitar kita, dan dalam konteks desain sistem TI, IoT membawa dampak yang sangat besar terhadap cara kita membangun dan mengelola sistem.

IoT berfungsi dengan menghubungkan berbagai perangkat yang dapat mengumpulkan dan berbagi data, seperti perangkat rumah pintar, kendaraan otonom, perangkat medis, dan berbagai sistem industri. Dengan adanya jaringan perangkat yang saling terhubung, data yang terkumpul dapat dianalisis untuk menghasilkan wawasan yang lebih baik, memperbaiki efisiensi operasional, dan menciptakan pengalaman pengguna yang lebih cerdas dan responsif.

Penerapan IoT dalam Desain Sistem TI

Penerapan IoT dalam desain sistem TI dapat ditemukan dalam berbagai bidang, mulai dari rumah pintar hingga industri manufaktur dan kesehatan. Beberapa contoh penerapan IoT dalam desain sistem TI meliputi:

1. **Sistem Rumah Pintar**
   Salah satu aplikasi IoT yang paling populer adalah dalam rumah pintar. Perangkat seperti lampu pintar, termostat pintar, kamera pengawasan, dan asisten virtual memungkinkan pengguna untuk mengontrol perangkat di rumah mereka dari jarak jauh melalui smartphone atau perangkat lainnya. Dalam desain sistem TI, IoT digunakan untuk mengintegrasikan berbagai perangkat ini dalam satu sistem yang dapat dikendalikan secara terpusat. Selain itu, sensor IoT dapat mengumpulkan data terkait kebiasaan pengguna untuk meningkatkan efisiensi energi dan kenyamanan rumah.
2. **Sistem Kesehatan dan Medis**
   Dalam dunia medis, IoT dapat digunakan untuk memantau kondisi pasien secara real-time. Perangkat medis yang terhubung, seperti monitor detak jantung, alat ukur tekanan darah, dan alat pengukur gula darah, dapat mengirimkan data langsung ke profesional medis atau pusat data untuk dianalisis. Ini memungkinkan dokter untuk memantau kondisi pasien secara lebih efektif dan merespons perubahan dalam kesehatan pasien dengan cepat. Dalam desain sistem TI, integrasi perangkat medis ini memerlukan arsitektur yang mampu menangani volume data besar secara aman dan efisien.
3. **Industri Manufaktur (Industry 4.0)**
   IoT juga memainkan peran besar dalam revolusi industri yang dikenal dengan istilah Industry 4.0. Di sektor manufaktur, IoT digunakan untuk menciptakan pabrik pintar yang dapat beroperasi secara otomatis dan efisien. Sensor IoT dipasang pada mesin untuk memantau kinerja, mendeteksi masalah lebih awal, dan mengoptimalkan proses produksi. Dalam desain sistem TI, IoT memungkinkan koneksi antara perangkat dan mesin yang berbeda, memungkinkan kontrol dan pemantauan proses produksi dalam waktu nyata.
4. **Kendaraan Otonom dan Transportasi Cerdas**: Kendaraan otonom, atau mobil yang dapat mengemudi sendiri, adalah contoh aplikasi IoT yang sedang berkembang pesat. Teknologi ini mengandalkan berbagai sensor, kamera, dan perangkat IoT lainnya untuk mengumpulkan data dari lingkungan sekitar

kendaraan dan mengambil keputusan berdasarkan data tersebut. Desain sistem TI untuk kendaraan otonom memerlukan infrastruktur yang mendukung komunikasi antara kendaraan, infrastruktur jalan, dan sistem kontrol lalu lintas secara real-time.

5. **Pertanian Cerdas (*Smart Agriculture*)**: IoT juga digunakan dalam **pertanian cerdas** untuk meningkatkan efisiensi dan hasil pertanian. Dengan menggunakan sensor IoT untuk memantau kelembaban tanah, suhu, dan kondisi cuaca, petani dapat membuat keputusan yang lebih cerdas mengenai waktu tanam, irigasi, dan pemupukan. Dalam desain sistem TI, integrasi data IoT ini memungkinkan pengelolaan sumber daya yang lebih baik, mengurangi pemborosan, dan meningkatkan hasil pertanian.

Tantangan dan Pertimbangan dalam Mengimplementasikan IoT dalam Desain Sistem TI

Meskipun IoT menawarkan banyak manfaat, ada beberapa tantangan yang perlu diperhatikan saat mengimplementasikannya dalam desain sistem TI:

1. **Skalabilitas**
   Salah satu tantangan terbesar dalam desain sistem IoT adalah skala jaringan. Dengan semakin banyaknya perangkat yang terhubung, sistem harus mampu menangani volume data yang sangat besar. Hal ini membutuhkan arsitektur yang sangat skalabel dan kemampuan untuk memproses data secara efisien.
2. **Keamanan dan Privasi**
   Keamanan adalah tantangan utama dalam sistem IoT. Setiap perangkat yang terhubung ke internet dapat menjadi titik kelemahan dalam jaringan. Oleh karena itu, penting untuk mengimplementasikan protokol keamanan yang kuat, seperti enkripsi data, otentikasi perangkat, dan pemantauan berkelanjutan untuk melindungi perangkat IoT dan data yang dikirimkan. Selain itu, mengingat data pribadi yang terkumpul dari perangkat IoT, perlindungan privasi juga menjadi masalah yang perlu diperhatikan.

3. **Kompleksitas Integrasi**
   IoT melibatkan berbagai perangkat dari berbagai produsen yang harus bekerja sama dalam satu sistem. Desain sistem TI harus memperhatikan kompatibilitas antar perangkat yang mungkin menggunakan protokol yang berbeda. Proses integrasi ini dapat menjadi sangat kompleks, terutama dalam industri besar atau aplikasi yang melibatkan berbagai sistem yang berbeda.
4. **Pemeliharaan dan Manajemen Jaringan**
   Mengelola jaringan IoT yang terdiri dari ribuan atau bahkan jutaan perangkat memerlukan sistem manajemen yang efisien. Sistem TI harus mampu memantau status perangkat secara real-time, mengidentifikasi perangkat yang bermasalah, dan melakukan pemeliharaan preventif untuk memastikan bahwa perangkat IoT tetap berfungsi dengan baik.

4.8 Kesimpulan

IoT memiliki potensi besar dalam desain sistem TI, dan teknologi ini akan terus berkembang seiring dengan kemajuan teknologi sensor, komunikasi, dan analisis data. Dalam beberapa tahun ke depan, kita dapat mengharapkan lebih banyak perangkat yang terhubung, lebih banyak data yang dikumpulkan, dan lebih banyak aplikasi yang memanfaatkan kekuatan data IoT untuk memberikan solusi yang lebih efisien, cerdas, dan otomatis.

Untuk mencapai potensi penuh dari IoT, tantangan-tantangan teknis seperti skalabilitas, keamanan, dan integrasi harus diatasi dengan baik. Desainer sistem TI harus beradaptasi dengan perkembangan teknologi ini dan merancang sistem yang dapat mengelola perangkat IoT secara efektif, memastikan keamanan dan privasi data, serta memberikan manfaat maksimal bagi pengguna.

DAFTAR PUSTAKA

Álvarez D., D.A., Buonaffina F., R.E. and Zabala A., S.A. (2014) 'Diagnosis of a operational, technology and communicative situation of nueva esparta state police, Venezuela | Diagnóstico de la situación operacional, tecnológica y comunicacional de la policía del estado nueva esparta, venezuela', *Espacios*, 35(2), p. 19. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84897901402&partnerID=40&md5=c5541d827cafc5977f76352c3201d883.

Chamberlain, L.B. *et al.* (2020) 'Automated decision systems for cybersecurity and infrastructure security', in *Proceedings - 2020 IEEE Symposium on Security and Privacy Workshops, SPW 2020*, pp. 196–201. Available at: https://doi.org/10.1109/SPW50608.2020.00048.

Dzhamaldinova, M.D., Kurdyukova, N.O. and Kunanbayeva, K.B. (2021) 'Integrated Management Systems for Digital Technology Enterprises', in *Proceedings of the 2021 IEEE International Conference 'Quality Management, Transport and Information Security, Information Technologies', T and QM and IS 2021*, pp. 27–30. Available at: https://doi.org/10.1109/ITQMIS53292.2021.9642769.

Juwono, E. *et al.* (2014) 'Aligning it with business model to perform organizational capabilities in achieving business performance', *Journal of Computer Science*, 10(12), pp. 2636–2639. Available at: https://doi.org/10.3844/jcssp.2014.2636.2639.

Kampichler, W. and Eier, D. (2014) 'A D-MILS console subsystem for advanced ATM communication services', in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pp. 6D21-6D28. Available at: https://doi.org/10.1109/DASC.2014.6979505.

Lovrekovic, Z. and Sukic, E. (2011) 'Information technologies do not solve problems in business - they just provide solutions if used in the right way', *Technics Technologies Education Management*, 6(3), pp. 579–587. Available at:

https://www.scopus.com/inward/record.uri?eid=2-s2.0-80054026718&partnerID=40&md5=d23c72112693dc7932527f1385284a57.

Pathak, P. and Choudhary, P. (2023) 'A Comprehensive Review of Various Machine Learning Techniques', in *Explainable Machine Learning Models and Architectures*, pp. 1–10. Available at: https://doi.org/10.1002/9781394186570.ch1.

BAB 5
KEAMANAN DAN PRIVASI DALAM DESAIN TEKNOLOGI INFORMASI

5.1 Pengertian Keamanan Dan Privasi Dalam Desain Teknologi Informasi

Dalam bidang teknologi informasi, keamanan dan privasi memiliki hubungan yang saling berkaitan, meskipun memiliki tujuan dan sasaran yang berbeda. Keamanan dalam desain teknologi informasi mengacu pada metode, teknologi, dan praktik yang dimaksudkan untuk menganalisis sistem, jaringan, data, dan infrastruktur dari sumber yang tidak sah. Tujuannya adalah untuk memantau ketersediaan, integritas, dan kerahasiaan data dan sistem. Adapun privasi dalam teknologi informasi diartikan sebagai data pribadi individu yang digabung, disimpan, digunakan, dibagian dan memastikan bahwa individu memiliki control atas informasi pribadinya. Tujuannya untuk memastikan data pribadi pengguna tetap aman dan dapat diakses dengan izin yang diberikan.

Keamanan dalam desain teknologi informasi merupakan perlindungan terhadap data dari berbagai ancaman seperti kebocoran, kerusakan, atau akses yang tidak sah. Keamanan data melibatkan serangkaian tindakan teknis dan prosedural untuk melindungi informasi dari ancaman keamanan peretasan sistem oleh orang yang tidak berwenang (Sarce Joel, Abdussalaam and Yunengsih, 2023). Setiap sistem teknologi informasi harus dirancang dengan mempertimbangkan kontrol akses yang ketat, enkripsi data, dan deteksi ancaman untuk meminimalisir risiko kebocoran atau peretasan. Keamanan juga mencakup pemantauan jaringan secara *real-time* untuk mengidentifikasi aktivitas mencurigakan dan merespons dengan cepat. Desain yang aman tidak hanya melindungi data pengguna tetapi juga memastikan ketersediaan layanan agar tidak terganggu oleh serangan siber. Implementasi lapisan keamanan berlapis-lapis menjadi standar yang harus diterapkan di semua aspek sistem.

Privasi dalam desain TI berfokus pada perlindungan informasi pribadi pengguna, memastikan bahwa data dikumpulkan, diproses, dan disimpan sesuai dengan aturan yang berlaku. Privasi merupakan hal yang sangat krusial apalagi di era Teknologi Informasi saat ini. Data pribadi adalah data yang berupa identitas dan penanda personal seseorang yang bersifat pribadi (Imam Teguh Islamy *et al.*, 2018). Dalam merancang sistem, penting untuk meminimalkan pengumpulan data yang tidak perlu dan memastikan mekanisme enkripsi diterapkan pada setiap tahap. Kontrol privasi, seperti pengaturan izin pengguna dan anonimisasi data, harus diimplementasikan agar data individu tidak disalahgunakan. Desain yang baik juga memungkinkan pengguna memiliki kendali penuh atas informasi mereka, termasuk pilihan untuk menghapus data yang telah disimpan. Dengan begitu, privasi pengguna dapat tetap terjaga, bahkan di tengah perkembangan teknologi yang pesat.

Keamanan dan privasi merupakan dua aspek yang sangat penting dan saling melengkapi dalam desain teknologi informasi. Keamanan berfokus pada perlindungan sistem dan data dari ancaman eksternal, sedangkan privasi berkonsentrasi pada pengelolaan dan perlindungan data pribadi pengguna. Desain TI yang efektif harus mengintegrasikan kedua elemen ini secara holistik, menerapkan kontrol keamanan yang ketat seperti enkripsi, pemantauan real-time, dan deteksi ancaman, sambil juga memastikan privasi pengguna melalui minimalisasi pengumpulan data, kontrol akses yang tepat, dan memberikan kendali kepada pengguna atas informasi mereka. Dengan memadukan keamanan yang kuat dan penghormatan terhadap privasi, sistem TI dapat memberikan perlindungan menyeluruh terhadap data dan kepercayaan pengguna, yang sangat penting di era digital saat ini di mana ancaman siber dan kekhawatiran privasi terus meningkat. Pendekatan ini tidak hanya melindungi organisasi dan individu dari risiko keamanan, tetapi juga membangun kepercayaan pengguna yang sangat penting untuk adopsi dan penggunaan teknologi yang berkelanjutan.

5.2 Peran Keamanan dan Privasi Dalam Desain Teknologi Informasi

Keamanan dan privasi dalam desain teknologi informasi memiliki beberapa peran yang krusial dalam era modern saat ini, berikut akan dijabarkan apa saja peran keamanan dan privasi teknologi informasi :

1. **Bidang Ekonomi**
   Dalam bidang ekonomi, keamanan dan privasi data merupakan faktor penting yang mempengaruhi kepercayaan daring (*online trust*). Dengan adanya keamanan dan privasi data pribadi membuat pengguna tidak merasa terancam apabila melakukan transaksi digital (Dewi Rosadi and Gumelar Pratama, 2018).
2. **Bidang Big Data**
   Keamanan dan privasi data dalam era *big data* berfungsi sebagai perlindungan terhadapa data yang rentan dari serangan siber. Keamanan dan privasi dalam big data harus dilakukan dengan benar untuk menghindari pelanggaran keamanan dan privasi seperti kebocoran informasi pribadi atau data sensitif kepada pihak yang tidak diinginkan (Munawar, Kom and Putri, 2020).
3. **Bidang Kesehatan**
   Keamanan dan privasi memainkan peran krusial dalam implementasi Rekam Medis Elektronik (RME) karena informasi yang terkandung di dalamnya bersifat sensitif dan harus dilindungi dari akses yang tidak sah. Dalam bidang kesehatan enekankan pentingnya penerapan teknik-teknik keamanan data, seperti enkripsi dan kontrol akses, untuk menjaga integritas dan kerahasiaan data pasien. Selain itu, diperlukan dasar hukum yang jelas untuk mengatur regulasi privasi agar pengembangan RME dapat dilakukan secara aman dan efektif (Ningtyas and Lubis, 2018).
4. **Bidang Pendidikan**
   Keamanan dan privasi data sangat penting dalam pembelajaran online, karena pelanggaran dapat membocorkan informasi pribadi dan nilai siswa. Teknologi blockchain membantu melindungi data ini dengan cara yang

aman, sehingga pemberi kerja dapat memverifikasi sertifikat siswa tanpa risiko penipuan (Iswanto *et al.*, 2022).

5. **Bidang Hukum**
   Keamanan dan privasi data merupakan aspek yang sangat penting dalam konteks hukum, terutama dalam perlindungan data pribadi dan pencegahan kejahatan siber. Dengan adanya regulasi yang ketat, seperti Undang-Undang Informasi dan Transaksi Elektronik (UU ITE) di Indonesia, diharapkan individu dan organisasi dapat melindungi informasi sensitif mereka dari akses yang tidak sah serta penyalahgunaan, sehingga menciptakan kepercayaan di masyarakat dan menjaga integritas hukum(Ramayanti and Lubis, 2023).

5.3 Elemen Keamanan dan Privasi Dalam Teknologi Informasi

5.3.1 Enkripsi Data

Enkripsi data adalah elemen krusial dalam desain keamanan teknologi informasi. Teknik ini mengubah data asli (*plaintext*) menjadi menjadi bentuk terenkripsi (*ciphertext*) menggunakan algoritma enkripsi dan kunci enkripsi yang sesuai, sehingga melindungi kerahasiaan dan integritas data dari akses tidak sah (Jaya and Yuliastuti, 2024). Dalam enkripsi data memiliki unsur-unsur penting yang bekerja bersama untuk memastikan keamanan dan kerahasiaan informasi. Berikut adalah unsur-unsur utama dalam enkripsi data:

1. ***Plaintext***
   Plaintext adalah data atau informasi yang belum dienkripsi dan masih dalam bentuk yang bisa dibaca atau dimengerti. Plaintext adalah input yang akan diubah menjadi ciphertext dalam proses enkripsi.
2. ***Chipertext***
   Hasil dari proses enkripsi, *ciphertext* adalah bentuk terenkripsi dari data asli yang tidak dapat dibaca tanpa kunci enkripsi yang benar. Hanya pihak yang berwenang dengan kunci yang tepat yang bisa mengubah *ciphertext* kembali menjadi *plaintext*.

3. **Kunci Enkripsi**
   Kunci adalah serangkaian data yang digunakan untuk mengenkripsi dan mendekripsi informasi. Kunci ini bisa berupa kunci simetris (untuk enkripsi simetris) atau kunci publik dan privat (untuk enkripsi asimetris). Panjang dan kompleksitas kunci memengaruhi tingkat keamanan enkripsi.
4. **Algoritma Enkripsi**
   Algoritma adalah metode atau prosedur matematis yang digunakan untuk mengubah *plaintext* menjadi *ciphertext*. Algoritma ini mengikuti serangkaian aturan yang menentukan bagaimana proses enkripsi dan dekripsi berlangsung. Contoh algoritma enkripsi populer adalah AES (*Advanced Encryption Standard*), RSA (*Rivest-Shamir-Adleman*), dan *Blowfish*.
5. **Kunci Dekripsi**
   Kunci yang digunakan untuk mengubah *ciphertext* kembali menjadi *plaintext*. Pada enkripsi simetris, kunci yang sama digunakan untuk enkripsi dan dekripsi. Sementara pada enkripsi asimetris, kunci publik digunakan untuk enkripsi dan kunci privat digunakan untuk dekripsi.
6. **Inisialisasi Vektor**
   Ini adalah metode yang digunakan untuk mengembalikan *ciphertext* ke bentuk *plaintext*. Algoritma dekripsi sering kali merupakan kebalikan dari algoritma enkripsi dan menggunakan kunci yang benar untuk memecahkan *ciphertext.*
7. **Inisialisasi Vektor**
   Elemen tambahan yang sering digunakan dalam algoritma enkripsi blok untuk meningkatkan keamanan. IV digunakan untuk memastikan bahwa enkripsi terhadap *plaintext* yang sama menghasilkan *ciphertext* yang berbeda setiap kali, sehingga sulit bagi penyerang untuk menebak pola enkripsi.
8. **Mode Operasi**
   Mode operasi adalah teknik yang digunakan untuk menerapkan algoritma enkripsi pada blok data. Beberapa mode operasi populer dalam enkripsi blok adalah ECB

(*Electronic Codebook*), CBC (*Cipher Block Chaining*), dan GCM (*Galois/Counter Mode*). Mode ini menentukan bagaimana blok-blok data dienkripsi dan dihubungkan satu sama lain.

9. ***Hashing***
   Meskipun bukan elemen langsung dari enkripsi, *hashing* adalah proses menghasilkan "sidik jari" digital dari data yang tidak dapat dikembalikan ke bentuk aslinya. Hashing digunakan untuk memverifikasi integritas data, bukan untuk enkripsi, tetapi sering dipadukan dengan enkripsi untuk memastikan data tidak berubah selama transmisi.

Dengan memperhatikan unsur-unsur dalam enkripsi data tersebut, dapat dipastikan bahwa data yang disimpan hanya akan diakses oleh pihak yang berwenang. Adapun jenis-jenis enkripsi data, sebagai berikut :

1. **Enkripsi Simetris (*Symmetric Encryption*)**
   Enkripsi simetris menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi. Kunci ini harus dijaga kerahasiaannya dan dibagikan antara pengirim dan penerima dengan cara yang aman. Kelebihan enkripsi simetris adalah kecepatannya, karena proses enkripsi dan dekripsinya relatif lebih cepat dibandingkan enkripsi asimetris (Basri, 2016). Namun, kelemahannya adalah tantangan dalam distribusi kunci dengan aman. Beberapa algoritma enkripsi simetris yang sering digunakan antara lain :
   a. AES (*Advanced Encryption Standard*) – Standar enkripsi yang banyak digunakan karena keamanan dan efisiensinya.
   b. DES (*Data Encryption Standard*) – Algoritma lama yang kini sudah dianggap kurang aman karena panjang kunci yang pendek.
   c. 3DES (*Triple DES*) – Versi yang lebih aman dari DES dengan tiga kali proses enkripsi.
   d. *Blowfish* – Algoritma yang cepat dan sering digunakan dalam perangkat lunak dan aplikasi kriptografi.

2. **Enkripsi Asimetris (*Asymmetric Encryption*)**
   Enkripsi asimetris menggunakan dua kunci yang berbeda tetapi berpasangan, yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk mengenkripsi data, sedangkan kunci privat digunakan untuk mendekripsinya. Dengan demikian, kunci publik dapat didistribusikan secara bebas, tetapi kunci privat harus dirahasiakan. Enkripsi asimetris lebih lambat daripada enkripsi simetris, tetapi lebih aman dalam hal pengiriman kunci (Saputro, Hidayati and Ujianto, 2020). Ini sering digunakan untuk pertukaran kunci dan autentikasi. Berikut adalah enkripsi asimetris yang umum :
   a. RSA (*Rivest-Shamir-Adleman*) – Algoritma asimetris paling banyak digunakan untuk enkripsi data dan tanda tangan digital.
   b. ECC (*Elliptic Curve Cryptography*) – Algoritma yang lebih modern dan lebih efisien dibandingkan RSA, khususnya dalam hal ukuran kunci yang lebih kecil namun tetap aman.
3. **Enkripsi *Hybrid***
   Enkripsi *hybrid* menggabungkan kelebihan enkripsi simetris dan asimetris. Proses ini sering digunakan dalam sistem enkripsi modern, seperti dalam protokol HTTPS. Dalam skema ini, enkripsi asimetris digunakan untuk pertukaran kunci simetris dengan aman, kemudian enkripsi simetris digunakan untuk mengenkripsi data dengan cepat setelah kunci dibagikan.
4. **Enkripsi End-to-End (*End-to-End Encryption* - E2EE)**
   Enkripsi end-to-end memastikan bahwa hanya pengirim dan penerima yang dapat membaca pesan, dengan mengenkripsi data pada sisi pengirim dan mendekripsinya pada sisi penerima. Bahkan server perantara yang mengirimkan data tidak dapat mendekripsinya. Ini sering digunakan dalam layanan pesan instan seperti *WhatsApp* dan *Signal* untuk menjamin privasi pengguna.

3.3.2 Autentikasi dan Otorisasi

Autentikasi dan otorisasi adalah dua komponen fundamental dalam keamanan sistem informasi yang bekerja bersama untuk mengontrol akses pengguna. Beriukut adalah jenis-jenis autentikasi dan otorisasi yang paling umun digunakan:

1. **Autentikasi**
   a. **Autentikasi Berbasis Kata Sandi (*Password-based Authentication*):** Pengguna memasukkan username dan password untuk login. Metode ini umum tetapi rentan terhadap serangan.
   b. **Autentikasi Dua Faktor (2FA) dan Multi-Faktor (MFA):** Menambahkan lapisan keamanan tambahan dengan memerlukan lebih dari satu bentuk autentikasi, seperti kode SMS atau aplikasi autentikator selain kata sandi. MFA bahkan melibatkan biometrik seperti sidik jari atau pemindai wajah untuk meningkatkan keamanan.
   c. **Autentikasi Biometrik (*Biometric Authentication*):** Menggunakan sidik jari, pemindaian wajah, atau pemindaian iris untuk autentikasi. Biasanya digunakan di perangkat mobile atau akses fisik ke sistem.
   d. **Autentikasi Berbasis Sertifikat (*Certificate-based Authentication*):** Memanfaatkan sertifikat digital untuk memverifikasi identitas pengguna atau perangkat. Umum digunakan dalam aplikasi perusahaan.
   e. **Autentikasi Berbasis Token (*Token-based Authentication*):** Pengguna mendapatkan token setelah login yang kemudian digunakan untuk mengakses sumber daya tanpa perlu login berulang. Contoh populer adalah OAuth yang digunakan dalam layanan pihak ketiga seperti Google atau Facebook.
2. **Otorisasi**
   a. ***Access Control List* (ACL):** Menentukan izin akses berdasarkan daftar pengguna atau grup pengguna terkait dengan sumber daya tertentu, seperti izin baca, tulis, atau eksekusi pada file.

b. ***Role-Based Access Control* (RBAC):** Akses diberikan berdasarkan peran pengguna dalam organisasi, seperti *admin*, *user*, atau *guest*. Setiap peran memiliki hak akses yang berbeda.
c. ***Attribute-Based Access Control* (ABAC):** Otorisasi ditentukan berdasarkan atribut seperti peran pengguna, lokasi, waktu, atau status. Lebih fleksibel daripada RBAC.
d. **OAuth:** Protokol yang memungkinkan aplikasi pihak ketiga untuk mendapatkan akses terbatas ke sumber daya pengguna tanpa meminta kredensial. Contohnya, login dengan Google atau Facebook di aplikasi lain.
e. ***Single Sign-On* (SSO):** SSO memungkinkan pengguna untuk mengakses beberapa aplikasi dengan satu kali login. Setelah login ke satu layanan, pengguna dapat mengakses layanan lainnya tanpa harus login ulang. SAML dan OAuth adalah protokol umum untuk SSO.
f. ***Least Privilege*:** Prinsip ini memastikan bahwa pengguna hanya diberikan akses minimum yang diperlukan untuk menjalankan tugas mereka, sehingga mengurangi risiko jika akun dikompromikan.

5.3.3 Keamanan Jaringan

Keamanan jaringan merupakan aspek kritis dalam melindungi infrastruktur teknologi informasi dari berbagai ancaman siber. Ini mencakup serangkaian tindakan, perangkat, dan protokol yang dirancang untuk mengamankan integritas, kerahasiaan, dan aksesibilitas data serta sumber daya jaringan. Berikut jenis-jenis keamanan yang digunakan dalam kemanan jaringan:

1. ***Firewall***

   Firewall adalah sistem yang memonitor dan mengontrol lalu lintas jaringan masuk dan keluar berdasarkan aturan keamanan yang telah ditetapkan. Ini bertindak sebagai penghalang antara jaringan internal yang aman dan jaringan eksternal yang tidak aman seperti internet. Contoh

Penggunaan firewall adalah mencegah serangan seperti malware, ransomware, dan serangan *denial-of-service* (DoS).

2. **IDS (*Intrusion Detection System*)**
   IDS (*Intrusion Detection System*) adalah keamanan untuk memantau jaringan pada aktivitas mencurigakan atau berbahaya dan memberikan peringatan kepada administrator jika ada ancaman yang terdeteksi.
3. **IPS (*Intrusion Prevention System*)**
   Instrusion Prevention System tidak hanya mendeteksi, tetapi juga mencegah ancaman dengan memblokir atau memitigasi serangan yang terdeteksi secara otomatis.
4. ***Virtual Private Network* (VPN)**
   Virtual Private Network adalah sebuah keamanan yang menciptakan saluran terenkripsi antara perangkat pengguna dan jaringan yang mereka akses. Ini melindungi data selama transmisi dari pencurian atau penyadapan, serta memungkinkan akses aman ke jaringan internal dari lokasi jauh. Contoh penerapan VPN adalah Digunakan oleh karyawan untuk mengakses jaringan perusahaan dari rumah atau tempat lain dengan aman, atau oleh individu untuk menjaga privasi saat menjelajah internet.

5.4 Model Keamanan dan Privasi Dalam Teknologi Informasi

Dalam dunia teknologi informasi, terdapat beberapa model keamanan dan privasi yang sangat dikenal dan sering digunakan oleh profesional IT, organisasi, dan perusahaan untuk memastikan perlindungan data dan sistem mereka. Model ini tidak hanya memberikan kerangka kerja konseptual untuk memahami dan mengimplementasikan keamanan, tetapi juga memandu dalam pengelolaan risiko serta kepatuhan terhadap standar dan regulasi. Berikut adalah beberapa model keamanan dan privasi yang paling dikenal dan digunakan:

5.4.1 CIA Triad (*Confidentiality, Integrity, Availability*)

Model CIA Triad adalah model dasar dan fundamental dalam keamanan informasi (Harahap *et al.*, 2023). Model ini terdiri dari tiga komponen utama yang harus dipenuhi untuk melindungi informasi:

1. ***Confidentiality*** **(Kerahasiaan):** Memastikan bahwa informasi hanya dapat diakses oleh pihak yang berwenang.
   Contoh penerapannya: Enkripsi data, kontrol akses, dan autentikasi pengguna.
2. ***Integrity*** **(Integritas):** Memastikan bahwa data tidak dapat diubah, dirusak, atau dihapus oleh pihak yang tidak sah.
   Contoh penerapannya: Tanda tangan digital, checksum, dan mekanisme deteksi perubahan data.
3. ***Availability*** **(Ketersediaan):** Memastikan bahwa sistem dan data dapat diakses oleh pengguna yang sah saat dibutuhkan.
   Contoh penerapannya: Redundansi jaringan, backup, disaster recovery, dan pemeliharaan sistem.

5.4.2 NIST *Cybersecurity Framework*

NIST Cybersecurity Framework adalah kerangka kerja yang dikembangkan oleh *National Institute of Standards and Technology* (NIST) di Amerika Serikat untuk membantu organisasi dalam mengelola risiko keamanan siber. Kerangka kerja ini dirancang untuk meningkatkan kemampuan perlindungan, deteksi, dan respons terhadap insiden keamanan siber. Model ini terdiri dari lima fungsi utama:

1. ***Identify*** **(Identifikasi):** Menentukan aset, risiko, dan proses kritis yang harus dilindungi.
2. ***Protect*** **(Perlindungan):** Menerapkan langkah-langkah untuk melindungi sistem dari serangan.
3. ***Detect*** **(Deteksi):** Memantau dan mendeteksi kejadian yang dapat mengancam keamanan.
4. ***Respond*** **(Respon):** Merespons insiden yang terjadi untuk meminimalkan dampaknya.
5. ***Recover*** **(Pemulihan):** Memulihkan kemampuan sistem yang terpengaruh oleh insiden

5.4.3 ISO/IEC 27001 dan ISO/IEC 27002

ISO/IEC 27001 adalah standar internasional untuk manajemen keamanan informasi (Information Security Management System - ISMS) yang memberikan panduan untuk merancang, menerapkan, memantau, dan meningkatkan keamanan informasi di dalam organisasi.

1. **ISO/IEC 27001:** Fokus pada persyaratan untuk menerapkan dan menjalankan ISMS. Standar ini mencakup kebijakan keamanan, manajemen risiko, kontrol akses, dan pemantauan.
2. **ISO/IEC 27002:** Menyediakan pedoman yang lebih rinci untuk pengendalian keamanan informasi berdasarkan kebijakan yang ada di ISO/IEC 27001.

5.4.4 Zero Trust Model

Zero Trust adalah pendekatan keamanan yang mengasumsikan bahwa ancaman dapat berasal dari luar dan dalam jaringan organisasi. Oleh karena itu, tidak ada entitas, baik di dalam maupun di luar jaringan, yang secara otomatis dipercaya. Berikut komponen yang terapat dalam zero trust model:

1. **Verifikasi Terus Menerus:** Setiap permintaan akses harus diverifikasi ulang, tanpa terkecuali.
2. **Prinsip *Least Privilege* (Hak Akses Minimal):** Pengguna diberikan hak akses paling minimal yang diperlukan untuk menyelesaikan pekerjaannya.
3. **Segregasi Jaringan:** Pemisahan jaringan untuk membatasi akses dan mengurangi risiko jika satu segmen terkena serangan.

Model ini semakin populer terutama dalam konteks cloud dan remote work, di mana akses ke data tidak lagi dibatasi oleh perimeter jaringan yang kaku.

5.4.5 *Privacy by Design*

Privacy by Design (PbD) adalah konsep yang menekankan pentingnya perlindungan privasi pengguna sejak tahap awal perancangan sistem, layanan, atau produk. Prinsip ini

dikembangkan oleh Ann Cavoukian, mantan *Information and Privacy Commissioner of Ontario,* Kanada, pada tahun 1990-an. PbD bertujuan untuk memastikan bahwa privasi bukan hanya sekadar fitur tambahan, tetapi menjadi bagian integral dari seluruh proses pengembangan produk. Prinsip-prinsip yang diterapkan dalam *privacy bye design* antara lain sebagai berikut:

1. **Proaktif, Bukan Reaktif:** Mengidentifikasi dan menangani risiko privasi sejak awal.
2. **Privasi sebagai Pengaturan Default:** Mengatur sistem untuk menjaga privasi pengguna secara default, tanpa perlu konfigurasi tambahan.
3. **Fokus pada Kepentingan Pengguna:** Memastikan pengguna memiliki kendali atas data pribadi mereka dan tahu bagaimana data tersebut digunakan.

5.4.6 *Defense in* Depth (Pertahanan Berlapis)

Defense in Depth adalah pendekatan yang menggunakan beberapa lapisan kontrol keamanan untuk melindungi aset informasi. Setiap lapisan memiliki kontrol dan strategi untuk menghadapi ancaman tertentu. Komponen dalam pendekatan ini adalah:

1. **Lapisan Fisik:** Keamanan fisik seperti akses gedung, CCTV, dan penjaga keamanan.
2. **Lapisan Jaringan:** Firewall, IDS/IPS, segmentasi jaringan, dan VPN.
3. **Lapisan Aplikasi:** WAF (*Web Application Firewall*), pemindaian kerentanan, dan pengetesan aplikasi.
4. **Lapisan Data:** Enkripsi, manajemen hak akses, dan kontrol identitas.

Dengan model ini, jika satu lapisan keamanan gagal, lapisan lainnya masih bisa memberikan perlindungan.

5.4.7 COBIT (*Control Objectives for Information and Related Technologies*)

COBIT adalah kerangka kerja yang sangat penting bagi organisasi yang ingin memastikan tata kelola TI yang baik dan

selaras dengan strategi bisnis. Penerapan COBIT dapat membantu meningkatkan efisiensi, kepatuhan, dan keamanan informasi, serta mendukung pencapaian tujuan strategis organisasi. Berikut prinsip yang digunakan dalam model COBIT:

1. *Plan and Organize*: Merencanakan dan mengatur strategi keamanan.
2. *Acquire and Implement*: Mengakuisisi dan mengimplementasikan kontrol dan teknologi keamanan.
3. *Deliver and Support*: Memberikan dukungan keamanan secara berkelanjutan.
4. *Monitor and Evaluate*: Memantau dan mengevaluasi kinerja sistem keamanan.

5.4.8 *Bell-LaPadula Model*

Model Bell-LaPadula adalah model keamanan yang digunakan dalam lingkungan dengan fokus pada kerahasiaan informasi, seperti organisasi militer dan pemerintah. Model ini menggunakan aturan untuk membatasi akses data berdasarkan tingkat otorisasi pengguna. Dua pinsip utama dalam Bell-LaPadula Model adalah:

1. No Read Up (*Simple Security Property*): Pengguna tidak dapat membaca data dari tingkat keamanan yang lebih tinggi.
2. No Write Down (*Star Property*): Pengguna tidak dapat menulis data ke tingkat keamanan yang lebih rendah.

5.5 Manajemen Risiko Keamanan Siber (*Cybersecurity Risk Management*)

Manajemen Risiko Keamanan Siber (*Cybersecurity Risk Management*) adalah proses identifikasi, penilaian, dan mitigasi risiko yang terkait dengan keamanan informasi dan sistem siber suatu organisasi. Tujuannya adalah untuk memastikan bahwa risiko terhadap sistem, data, dan proses bisnis dapat dikendalikan secara efektif dan sejalan dengan toleransi risiko yang telah ditetapkan oleh organisasi.

5.5.1 Konsep Dasar Manajemen Risiko Keamanan Siber

Manajemen risiko keamanan siber adalah pendekatan holistik yang menggabungkan penilaian risiko teknis, operasional, dan bisnis. Dengan manajemen risiko yang baik, organisasi dapat melindungi aset informasi dari ancaman internal maupun eksternal, memastikan kelangsungan operasional, dan meminimalkan dampak finansial serta reputasi akibat insiden keamanan. Komponen utama dalam manajemen risiko keamanan siber meliputi:

1. **Identifikasi Risiko:** Mengidentifikasi ancaman dan kerentanan yang ada pada sistem serta potensi dampak yang dapat terjadi jika risiko tersebut dieksploitasi.
2. **Analisis dan Penilaian Risiko:** Mengukur seberapa besar kemungkinan ancaman tersebut terjadi (likelihood) dan dampaknya terhadap bisnis.
3. **Mitigasi Risiko:** Menerapkan kontrol keamanan untuk mengurangi risiko ke tingkat yang dapat diterima.
4. **Pemantauan dan Review Risiko:** Menilai kembali risiko secara berkala dan mengadaptasi strategi mitigasi berdasarkan perubahan dalam lingkungan bisnis dan teknologi.

5.5.2 Tahapan Manajemen Risiko Keamanan Siber

Proses manajemen risiko keamanan siber biasanya terdiri dari beberapa tahapan berikut:

1. **Identifikasi Aset dan Risiko:**
   a. Mengidentifikasi aset informasi yang kritis, seperti data sensitif, sistem aplikasi, dan infrastruktur jaringan.
   b. Menentukan nilai atau tingkat pentingnya aset tersebut bagi organisasi, serta mengidentifikasi ancaman yang mungkin muncul.
2. **Penilaian Risiko (*Risk Assessment*):**
   a. Menggunakan metode kuantitatif atau kualitatif untuk menilai risiko berdasarkan kemungkinan kejadian dan dampak yang dapat ditimbulkan.
   b. Misalnya, menggunakan matriks risiko untuk memvisualisasikan risiko yang tinggi, sedang, dan rendah.

3. **Analisis Risiko (*Risk Analysis*):**
    a. Melakukan analisis mendalam untuk menentukan penyebab utama dari setiap risiko serta potensi kerentanannya.
    b. Contoh: Melakukan threat modeling untuk mengidentifikasi kemungkinan serangan yang dapat dieksploitasi oleh penyerang.
4. **Evaluasi Risiko (*Risk Evaluation*):**
    a. Mengevaluasi risiko yang telah diidentifikasi untuk menentukan prioritas mitigasi berdasarkan tingkat keparahannya terhadap operasional bisnis.
5. **Mitigasi Risiko (*Risk Mitigation*):**
    a. Merencanakan dan menerapkan kontrol keamanan untuk mengurangi risiko, seperti kebijakan keamanan, firewall, enkripsi, dan pelatihan karyawan.
    b. Misalnya, risiko pencurian data sensitif dapat diatasi dengan menerapkan enkripsi end-to-end pada data.
6. **Pemantauan dan Review:**
    a. Pemantauan risiko dilakukan secara berkala untuk mengidentifikasi perubahan yang dapat memengaruhi profil risiko.
    b. Review dan audit risiko secara berkala untuk memastikan efektivitas kontrol keamanan yang diterapkan.

5.5.3 Tantangan dalam Manajemen Risiko Keamanan Siber

Beberapa tantangan utama yang dihadapi dalam manajemen risiko keamanan siber adalah:

1. **Perubahan Lanskap Ancaman:** Ancaman siber terus berkembang dengan cepat, sehingga strategi manajemen risiko harus fleksibel dan adaptif.
2. **Keterbatasan Sumber Daya:** Sumber daya keamanan yang terbatas, baik dari segi personel maupun finansial, dapat membatasi kemampuan organisasi dalam menerapkan kontrol mitigasi yang memadai.
3. **Kompleksitas Infrastruktur TI:** Infrastruktur TI yang kompleks, termasuk penggunaan cloud, IoT, dan sistem

terdistribusi, dapat menyulitkan pengelolaan risiko secara efektif.

4. **Kepatuhan Regulasi:** Mengintegrasikan manajemen risiko dengan persyaratan kepatuhan regulasi, seperti GDPR dan CCPA, memerlukan pemahaman mendalam tentang berbagai undang-undang yang berlaku.

5.5.4 Manfaat Manajemen Risiko Keamanan Siber

Manfaat utama dari penerapan manajemen risiko keamanan siber yang baik meliputi:

1. **Mengurangi risiko terjadinya insiden keamanan dan dampak negatifnya.**
   Dengan melakukan identifikasi dan analisis risiko, organisasi dapat mengetahui potensi ancaman yang mungkin dihadapi serta kerentanannya. Ini memungkinkan organisasi untuk mengambil langkah mitigasi proaktif untuk mencegah insiden terjadi, seperti serangan siber, kebocoran data, atau ancaman dari pihak internal. Contoh: Mengimplementasikan enkripsi untuk data sensitif atau firewall yang dikonfigurasi dengan baik dapat mencegah serangan berbasis jaringan.
2. **Memastikan kelangsungan operasional bisnis dan meminimalkan downtime.**
   Manajemen risiko yang efektif membantu organisasi dalam mengembangkan rencana pemulihan bencana dan keberlangsungan bisnis (*Business Continuity Plan*), sehingga jika terjadi gangguan atau insiden, organisasi dapat pulih lebih cepat dan mengurangi waktu henti (*downtime*). Hal ini penting untuk memastikan bahwa layanan kritis tetap berjalan dan dampak operasional dapat diminimalkan.
3. **Mematuhi regulasi dan standar keamanan yang berlaku, sehingga mengurangi potensi denda dan penalti.**
   Manajemen risiko yang baik membantu organisasi untuk mematuhi berbagai regulasi keamanan data dan privasi, seperti GDPR (*General Data Protection Regulation*), CCPA (*California Consumer Privacy Act*), dan standar industri

lainnya seperti ISO 27001 atau NIST. Mematuhi regulasi ini tidak hanya menghindarkan organisasi dari denda atau penalti, tetapi juga membantu membangun kepercayaan dengan pelanggan dan mitra bisnis.

4. **Melindungi reputasi organisasi dengan menunjukkan komitmen terhadap keamanan informasi.**
   Melindungi reputasi organisasi dengan menunjukkan komitmen terhadap keamanan informasi sangat penting di era digital saat ini, di mana pelanggaran data dan insiden keamanan dapat merusak kepercayaan publik dan merugikan bisnis secara signifikan. Ketika organisasi menunjukkan bahwa mereka memiliki langkah-langkah keamanan yang kuat dan memprioritaskan perlindungan informasi, hal ini mencerminkan komitmen mereka terhadap kepercayaan, transparansi, dan tanggung jawab terhadap para pemangku kepentingan (*stakeholders*).

5.6 Etika Keamanan Dan Privasi Teknologi Informasi

Etika Keamanan dan Privasi dalam Teknologi Informasi berkaitan dengan prinsip-prinsip moral dan panduan yang harus dipatuhi oleh individu, organisasi, dan komunitas dalam mengelola informasi, melindungi privasi, serta menggunakan teknologi informasi secara bertanggung jawab. Etika ini mencakup aspek seperti penggunaan data pribadi, hak privasi, transparansi, keamanan informasi, dan pertimbangan moral dalam pengembangan teknologi.

5.6.1 Prinsip-Prinsip Etika Keamanan dan Privasi

Prinsip-prinsip etika keamanan dan privasi memberikan pedoman bagi individu dan organisasi tentang bagaimana data dan informasi harus dikelola dengan cara yang menghormati hak-hak individu, menjaga kerahasiaan, dan mempromosikan keadilan serta transparansi. Berikut adalah prinsip-prinsip utama yang sering digunakan sebagai acuan dalam etika keamanan dan privasi:

1. **Prinsip Privasi (*Privacy Principle*)**
   Prinsip ini menyatakan bahwa setiap individu memiliki hak untuk privasi informasi pribadinya. Organisasi harus memastikan bahwa data pribadi hanya dikumpulkan dan digunakan dengan persetujuan yang jelas dari pemilik data. Informasi ini tidak boleh digunakan, disebarkan, atau dijual tanpa persetujuan eksplisit dari individu tersebut.
   Contoh penerapan: Organisasi harus meminta persetujuan pengguna untuk memproses data mereka dan memberikan pilihan kepada pengguna untuk menarik persetujuan kapan saja.
2. **Prinsip Transparansi (*Transparency Principle*)**
   Transparansi adalah prinsip di mana organisasi harus bersikap terbuka mengenai cara mereka mengumpulkan, menyimpan, dan menggunakan informasi pribadi. Organisasi harus menyediakan kebijakan privasi yang mudah dimengerti dan memberikan informasi lengkap tentang apa yang dilakukan dengan data pengguna.
   Contoh penerapan: Memberikan pemberitahuan yang jelas kepada pengguna tentang pengumpulan data, jenis data yang dikumpulkan, serta tujuan penggunaan data tersebut.
3. **Prinsip Akuntabilitas (*Accountability Principle*)**
   Organisasi yang mengumpulkan dan mengelola data harus bertanggung jawab atas perlindungan data tersebut. Mereka harus mengambil langkah-langkah yang tepat untuk mencegah akses yang tidak sah, kebocoran data, atau penyalahgunaan informasi, serta bersedia menanggung konsekuensi jika terjadi pelanggaran.
   Contoh penerapan: Memiliki petugas perlindungan data (*Data Protection Officer*) yang bertanggung jawab atas implementasi kebijakan keamanan informasi dan kepatuhan terhadap regulasi privasi.
4. **Prinsip Keadilan (*Fairness Principle*)**
   Pengumpulan dan penggunaan data harus dilakukan dengan cara yang adil, tanpa menipu atau menyesatkan pemilik data. Organisasi harus memastikan bahwa data hanya

digunakan untuk tujuan yang sah dan sesuai dengan persetujuan yang diberikan oleh pemilik data.
Contoh penerapan: Tidak menggunakan informasi pribadi untuk melakukan diskriminasi atau profiling yang dapat merugikan individu, seperti mengurangi peluang kerja berdasarkan informasi pribadi yang tidak relevan.

5. **Prinsip Perlindungan Data (*Data Protection Principle*)**
   Data pribadi harus dilindungi dengan langkah-langkah keamanan yang memadai untuk mencegah akses tidak sah, pengubahan, pengungkapan, atau penghapusan data yang tidak sah. Organisasi wajib menerapkan teknologi enkripsi, pengendalian akses, dan audit keamanan secara berkala untuk melindungi data yang disimpan.
   Contoh penerapan: Menerapkan enkripsi end-to-end untuk data sensitif dan membatasi akses hanya kepada personel yang berwenang.

5.6.2 Etika dalam Penggunaan Data dan Teknologi

Berikut adalah 5 etika dalam penggunaan data dan teknologi yang wajib dan paten diterapkan untuk memastikan bahwa teknologi digunakan dengan cara yang bertanggung jawab dan tidak merugikan pemilik data:

1. **Prinsip Pengumpulan Data yang Sah dan Berizin (*Legitimate Data Collection and Consent*)**
   Data pribadi hanya boleh dikumpulkan jika memiliki izin dari pemilik data dan untuk tujuan yang jelas serta sah. Organisasi harus memastikan bahwa pengguna memberikan persetujuan eksplisit (*informed consent*) dan mengetahui dengan pasti bagaimana data mereka akan digunakan.
   Contoh penerapan: Mengumpulkan data pengguna melalui formulir yang jelas, dengan memberikan informasi rinci tentang tujuan pengumpulan data serta menyediakan pilihan bagi pengguna untuk menolak pengumpulan data.
2. **Prinsip Minimasi Penggunaan Data (*Data Minimization*)**
   Hanya data yang benar-benar diperlukan untuk keperluan yang sah yang boleh dikumpulkan, disimpan, dan digunakan. Organisasi sebaiknya menghindari

pengumpulan data yang berlebihan, yang berisiko meningkatkan kerentanan keamanan serta melanggar privasi.

Contoh penerapan: Menghindari pengumpulan informasi sensitif yang tidak diperlukan, seperti data etnis, agama, atau preferensi politik jika tidak terkait dengan layanan yang diberikan.

3. **Prinsip Keamanan dan Kerahasiaan Data (*Data Security and Confidentiality*)**

   Data yang dikumpulkan harus dilindungi dengan langkah-langkah keamanan yang memadai untuk mencegah akses, pengungkapan, atau modifikasi yang tidak sah. Organisasi wajib menerapkan praktik keamanan siber yang ketat seperti enkripsi, pengendalian akses, dan audit keamanan secara berkala.

   Contoh penerapan: Menggunakan protokol enkripsi untuk melindungi data sensitif, memantau aktivitas pengguna yang mencurigakan, dan menerapkan autentikasi ganda (two-factor authentication) untuk akses data.

4. **Prinsip Penggunaan yang Adil dan Tidak Diskriminatif (*Fair and Non-Discriminatory Use*)**

   Data yang dikumpulkan tidak boleh digunakan untuk tujuan yang dapat merugikan pemilik data, seperti diskriminasi, manipulasi, atau eksploitasi. Penggunaan data harus adil dan sesuai dengan persetujuan yang telah diberikan oleh pengguna.

   Contoh penerapan: Tidak menggunakan informasi pribadi untuk menolak akses terhadap layanan tertentu atau membuat profil yang dapat mengakibatkan diskriminasi atau ketidakadilan.

5. **Prinsip Transparansi dan Akuntabilitas (*Transparency and Accountability*)**

   Organisasi harus transparan tentang bagaimana mereka mengumpulkan, menggunakan, menyimpan, dan membagikan data pengguna. Mereka harus memberikan akses kepada pemilik data untuk mengetahui data apa saja yang telah dikumpulkan dan bagaimana data tersebut

digunakan. Selain itu, organisasi harus bertanggung jawab jika terjadi pelanggaran keamanan atau penyalahgunaan data.

Contoh penerapan: Memberikan laporan transparansi tahunan kepada publik tentang kebijakan dan insiden keamanan informasi serta mengembangkan kebijakan privasi yang mudah diakses dan dipahami.

5.7 Awareness Dan Pelatihan Keamanan Serta Privasi Teknologi Informasi

Awareness dan pelatihan keamanan serta privasi teknologi informasi adalah komponen penting dalam upaya organisasi untuk melindungi aset informasi dan data pribadi dari ancaman keamanan. Meningkatkan kesadaran (*awareness*) dan memberikan pelatihan yang efektif kepada karyawan, mitra, dan pihak terkait dapat membantu mencegah insiden keamanan yang disebabkan oleh kesalahan manusia, seperti phishing, penggunaan kata sandi yang lemah, atau kebocoran data.

5.7.1 Pentingnya *Awareness* dan Pelatihan Keamanan serta Privasi Teknologi Informasi

Awareness dan pelatihan ini penting karena beberapa alasan:

1. **Mengurangi Risiko Insiden Keamanan:** Sebagian besar insiden keamanan terjadi karena kesalahan manusia, seperti mengklik tautan phishing atau membagikan informasi sensitif tanpa disengaja. Dengan pelatihan yang baik, karyawan akan lebih waspada dan dapat mengenali ancaman seperti phishing, social engineering, atau malware.
2. **Mematuhi Regulasi dan Standar Keamanan:** Banyak regulasi seperti GDPR, HIPAA, dan ISO 27001 mewajibkan organisasi untuk memberikan pelatihan kepada karyawan mengenai keamanan informasi dan privasi. Ini penting untuk memastikan organisasi mematuhi hukum dan mengurangi risiko denda atau sanksi.
3. **Membentuk Budaya Keamanan yang Kuat:** Meningkatkan kesadaran keamanan di tempat kerja akan membentuk budaya keamanan yang lebih kuat, di mana setiap karyawan

merasa bertanggung jawab atas keamanan informasi, baik di tempat kerja maupun saat bekerja dari jarak jauh.

4. **Mengurangi Potensi Kebocoran Data:** Awareness dan pelatihan yang tepat dapat mengurangi risiko terjadinya kebocoran data akibat perilaku karyawan yang tidak aman, seperti penggunaan perangkat penyimpanan eksternal yang tidak terenkripsi atau berbagi dokumen melalui saluran komunikasi yang tidak aman.

5.7.2 Komponen Penting dalam Program Awareness dan Pelatihan

Program awareness dan pelatihan yang efektif harus mencakup beberapa komponen utama:

1. **Materi Pelatihan yang Relevan dan Up-to-Date:** Materi pelatihan harus mencakup ancaman terkini seperti ransomware, phishing, serta perkembangan teknologi seperti cloud security dan IoT security. Program ini juga perlu mencakup kebijakan dan prosedur organisasi terkait keamanan informasi.
2. **Simulasi dan Pengujian:** Melakukan simulasi serangan seperti uji coba phishing (phishing simulation) dan pengujian keamanan internal dapat membantu karyawan merasakan ancaman nyata dan mempelajari cara merespons dengan tepat.
3. **Keterlibatan Manajemen:** Dukungan dari manajemen atas program ini sangat penting untuk memastikan pelatihan diprioritaskan dan sumber daya yang diperlukan tersedia.
4. **Pelatihan yang Interaktif:** Menggunakan pelatihan interaktif, seperti kuis, skenario kasus, atau diskusi kelompok, dapat membuat pelatihan lebih menarik dan efektif dibandingkan hanya ceramah satu arah.
5. **Frekuensi Pelatihan:** Pelatihan harus dilakukan secara berkala, bukan hanya satu kali di awal masa kerja. Misalnya, pelatihan bisa dilakukan setiap kuartal atau setidaknya sekali setahun untuk memastikan karyawan terus terinformasi tentang ancaman dan kebijakan terbaru.

5.7.3 Jenis-jenis Pelatihan Keamanan dan Privasi yang Umum Diberikan

Beberapa jenis pelatihan yang umum diterapkan dalam organisasi meliputi:

1. **Pelatihan Dasar Keamanan Informasi:** Meliputi pemahaman tentang dasar-dasar keamanan informasi, seperti cara melindungi data pribadi, penggunaan kata sandi yang kuat, dan mengenali email phishing.
2. **Pelatihan Spesifik Privasi Data:** Menyoroti pentingnya melindungi informasi pribadi pelanggan atau informasi sensitif lainnya. Ini mencakup penanganan data sesuai dengan standar atau regulasi tertentu, seperti GDPR dan CCPA.
3. **Pelatihan Social Engineering:** Mengajarkan karyawan tentang metode rekayasa sosial (*social engineering*) yang sering digunakan untuk mengeksploitasi kelemahan manusia, seperti menyamar sebagai petugas IT untuk mendapatkan akses ke sistem.
4. **Pelatihan Respons Insiden (*Incident Response Training*):** Mengajarkan cara merespons insiden keamanan seperti kebocoran data atau serangan malware. Ini membantu karyawan bertindak dengan cepat untuk meminimalkan dampak serangan.

5.7.4 Langkah-Langkah untuk Membangun Program *Awareness* dan Pelatihan yang Efektif

Untuk membangun program awareness dan pelatihan yang efektif, organisasi dapat melakukan langkah-langkah berikut:

1. **Melakukan Penilaian Awal (*Initial Assessment*):** Mengidentifikasi tingkat kesadaran keamanan saat ini di organisasi serta area yang perlu diperbaiki.
2. **Menentukan Tujuan Pelatihan:** Menetapkan tujuan yang jelas untuk program pelatihan, seperti meningkatkan kemampuan mengenali serangan phishing atau pemahaman tentang kebijakan privasi data perusahaan.
3. **Membuat Rencana Pelatihan:** Merancang program pelatihan dengan materi yang relevan, jadwal pelatihan, dan

metode penyampaian yang sesuai dengan kebutuhan organisasi.

4. **Menggunakan Teknologi dan Alat Bantu:** Memanfaatkan platform pembelajaran daring (online learning platform) yang interaktif serta alat bantu lainnya seperti video tutorial, modul e-learning, dan materi pendukung lainnya.
5. **Evaluasi dan Umpan Balik:** Mengukur efektivitas pelatihan melalui kuis, survei umpan balik, dan tes simulasi. Hasil evaluasi ini dapat digunakan untuk menyempurnakan program pelatihan di masa mendatang.

DAFTAR PUSTAKA

Basri (2016) 'KRIPTOGRAFI SIMETRIS DAN ASIMETRIS DALAM PERSPEKTIF KEAMANAN DATA DAN KOMPLEKSITAS KOMPUTASI'. Available at: http://ejournal.fikom-unasman.ac.id.

Dewi Rosadi, S. and Gumelar Pratama, G. (2018) 'URGENSI PERLINDUNGANDATA PRIVASIDALAM ERA EKONOMI DIGITAL DI INDONESIA', *Veritas et Justitia*, 4(1), pp. 88–110. Available at: https://doi.org/10.25123/vej.2916.

Harahap, A.H. *et al.* (2023) 'Pentingnya Peranan CIA Triad Dalam Keamanan Informasi dan Data Untuk Pemangku Kepentingan atau Stakholder', 1(2).

Imam Teguh Islamy *et al.* (2018) 'PENTINGNYA MEMAHAMI PENERAPAN PRIVASI DI ERA TEKNOLOGI INFORMASI', *Jurnal Teknologi Informasi dan Pendidikan*, 11(2).

Iswanto *et al.* (2022) 'Pemanfaatan Teknologi Blockchain di Bidang Pendidikan', *TEMATIK*, 9(2), pp. 171–181. Available at: https://doi.org/10.38204/tematik.v9i2.1082.

Jaya, A. and Yuliastuti, G.E. (2024) 'SNESTIK Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika Implementasi Multi Enkripsi Algoritma Stream Cipher dan Affine Cipher untuk Pengamanan Data Customer'. Available at: https://doi.org/10.31284/p.snestik.2024.5604.

Munawar, Z., Kom, M. and Putri, N.I. (2020) *KEAMANAN JARINGAN KOMPUTER PADA ERA BIG DATA*, *Jurnal Sistem Informasi-J-SIKA*.

Ningtyas, A.M. and Lubis, I.K. (2018) *LITERATUR REVIEW PERMASALAHAN PRIVASI PADA REKAM MEDIS ELEKTRONIK*, *Jurnal Pseudocode*. Available at: www.ejournal.unib.ac.id/index.php/pseudocode.

Ramayanti, H. and Lubis, A.F. (2023) *Peran Hukum dalam Mengatasi Serangan Cyber yang Mengancam Keamanan Nasional*, *Jurnal Hukum dan HAM Wara Sains*.

Saputro, T.H., Hidayati, N. and Ujianto, E.I.H. (2020) 'JIP (Jurnal Informatika Polinema) SURVEI TENTANG ALGORITMA KRIPTOGRAFI ASIMETRIS'.

Sarce Joel, A., Abdussalaam, F. and Yunengsih, Y. (2023) 'TATA KELOLA REKAM MEDIS BERBASIS TEKNOLOGI INFORMASI DALAM PENANGANAN KERAHASIAAN DAN KEAMANAN DATA PASIEN DENGAN METODE KRIPTOGRAFI', *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, 4(3), pp. 837–848. Available at: https://doi.org/10.35870/jimik.v4i3.287.

BIODATA PENULIS

Aisyah Mutia Dawis, S.Kom., M.Kom.
Dosen Program Studi Sarjana Sistem dan Teknologi Informasi
Fakultas Sains dan Teknologi Universitas 'Aisyiyah Surakarta

Penulis lahir di Kota Surakarta. Ia Lulus S1 pada tahun 2013 hingga mendapat gelar Sarjana Komputer di Universitas Muhammadiyah Surakarta serta mendapatkan penghargaan sebagai lulusan terbaik Se Fakultas Komunikasi dan Informatika, kemudian ia melanjutkan studi S2, lulus pada tahun 2019 hingga mendapat gelar Magister Komputer di Universitas Amikom Yogyakarta. Saat ini ia tercatat sebagai dosen tetap di Program Studi Sarjana Sistem dan Teknologi Informasi di Universitas 'Aisyiyah Surakarta. Selain mengajar ia aktif dalam kegiatan tridharma lainnya diantaranya ialah penelitian dan pengabdian kepada masyarakat. Kegiatan penelitian internal dan eksternal pernah dilakukannya. Beberapa penelitian berjudul : *" Virtual Reality Tour Sebagai Media Informasi Pengenalan Gedung Kampus 2 Universitas' Aisyiyah Surakarta"* dan *"Utilization of Virtual Reality Technology in Knowing the Symptoms of Acrophobia and Nyctophobia"*. Ia sering menjadi *invited speaker* diantaranya menjadi tenaga pengajar BPPTIK Kementerian Kominfo. Serta sebagai bentuk pengabdian kepada masyarakat, ia pun pernah terlibat aktif sebagai trainer bimbingan teknis bantuan TIK SMP, SMA, Mahasiswa, Tenaga Pendidik, Guru dan Dosen dari tahun 2010 hingga saat ini.

BIODATA PENULIS

Devi Rahmayantim, S.T., M.T.
Dosen Program Studi Teknik Elektro
Fakultas Teknik dan Sistem Informasi Universitas Gajayana

Penulis lahir di Prbolinggo tanggal 10 Juni 1975. Penulis adalah dosen tetap pada Program Studi Teknik Elektro Fakultas Teknik dan Sistem Informasi, Universitas Gajayana Malang. Menyelesaikan pendidikan S1 pada Jurusan Teknik Elektro Universitas Brawijaya dan melanjutkan S2 pada Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya. Penulis memiliki peminatan pada bidang keahlian Telekomunikasi (termasuk perkembangan Teknologi Informasi dan Komunikasi Digital) serta Engineering Management dalam bidang IT. Selain sebagai dosen, penulis juga rajin melakukan penelitian dan kegiatan pengabdian kepada masyarakat sesuai bidang keahlian.

BIODATA PENULIS

Taufik Rachman, S.Kom., M.T.
Dosen Program Studi Teknik Informatika

Penulis lahir di kota Kendari tanggal 11 Mei 1971. Penulis adalah dosen tetap pada Program Studi Teknik Informatika Sekolah Tinggi Teknologi STIKMA Internasional Malang. Menyelesaikan pendidikan Strata-1 pada Jurusan Teknik Informatika di Institut Sains & Teknolgo Palapa Malang dan menyelesaikan Studi Magister Jurusan Teknik Industri di Institut Teknologi Nasional Malang. Penulis konsern pada Matakuliah Analisa Perancangan Sistem Informasi, Rekayasa Perangkat Lunak, dan Basis Data sesuai bidang keahliannya

BIODATA PENULIS

Ir. Ali Impron, S.Kom., M.Kom.
Dosen Program Studi Informatika
Fakultas Teknik dan Pertanian
Universitas Muhammadiyah Sampit

Penulis lahir di Grobogan, Jawa Tengah, pada tahun 1983. Penulis adalah seorang dosen dan profesional IT dengan pengalaman lebih dari 17 tahun dalam mengelola layanan IT, internet service provider, dan industri pertambangan. Penulis memiliki keahlian dalam infrastruktur IT, IoT, Data Center, dan Bussiness Intelligence. Saat ini, penulis menjabat sebagai Head of Information Technology di PT. Darma Henwa Tbk, di mana penulis bertanggung jawab untuk mengelola tim IT dan memastikan layanan IT berkualitas tinggi.

Penulis menyelesaikan gelar Sarjana Teknik Informatika dari STMIK AKAKOM Yogyakarta dan gelar Magister Teknologi Informasi dari Universitas Teknologi Digital Indonesia, Yogyakarta. Saat ini, penulis sedang menempuh Program Doktor Ilmu Teknik di Universitas Negeri Yogyakarta. Selain perannya di dunia industri, penulis juga tercatat sebagai dosen tetap di Universitas Muhammadiyah Sampit.

Selama karirnya, penulis telah memperoleh berbagai sertifikasi internasional yang mengakui keahliannya di bidang teknologi informasi, termasuk Microsoft Certified IT Professional (MCITP), Cisco Certified Network Associate (CCNA), dan Juniper

Networks Certified Associate (JNCIA). Penulis memiliki minat besar dalam penelitian dan integrasi produk IT terbaru dengan proses bisnis, serta memiliki pemahaman mendalam tentang administrasi sistem, keamanan IT, dan jaringan.

BIODATA PENULIS

Yoseph Pius Kurniawan Kelen
Dosen Prodi Teknologi Informasi
Fakultas Pertania, Sains dan Kesehatan Universitas Timor

Penulis, lahir di Lahir di Kota Larantuka Nusa Tenggara Timur pada tahun 1980. Menempuh pendidikan S1 pada Program Studi Teknik Informatika dan Statistika Universitas Bina Nusantara Jakarta, lulus pada tahun 2005 dan S2 di Universitas Diponegoro Semarang pada Program Studi Sistem Informasi dengan peminatan Decision Support System, lulus pada tahun 2014. Saat ini penulis mengajar di Univeritas Timor Kefamenanu, semenjak tahun 2007 menjadi Dosen Tetap pada Program Studi pendidikan Matematika FKIP Univeritas Timor,selanjutnya pada tahun 2017 pindah homebase ke program studi Teknologi Informasi Fakultas Pertanian,Sains dan Kesehatan Universitas Timor. Disela-sela kesibukannya sebagai dosen tetap di prodi Teknologi Informasi, penulisa juga menjadi dosen titak tetap di STIPAS Kefamenanu dan Tutor pada Universitas Terbuka (UT) UPBJJ Kupang.