

# PERANCANGAN

# SISTEM INFORMASI DAN APLIKASINYA



**Sri Tria Siska, S.Kom., M.Kom, Yera Wahda Wahdi, S.Kom., M.Kom,  
Evi Lestari Pratiwi, M.Kom, Abrar Hadi, S.Kom., M.Kom,  
Firdaus, S.Kom., M.Kom, Kusuma Hati, M.M., M.Kom,  
Andi Jamiati Paramita, S.T., M.T, Putri Ariatna Alia**



# Perancangan Sistem Informasi dan Aplikasinya

## Penulis:

Sri Tria Siska, S.Kom., M.Kom  
Yera Wahda Wahdi, S.Kom., M.Kom  
Evi Lestari Pratiwi, M.Kom  
Abrar Hadi, S.Kom., M.Kom  
Firdaus, S.Kom., M.Kom  
Kusuma Hati, M.M., M.Kom  
Andi Jamiati Paramita, S.T., M.T  
Putri Ariatna Alia



# Perancangan Sistem Informasi dan Aplikasinya

Copyright © PT Penamuda Media, 2025

## Penulis:

Sri Tria Siska, S.Kom., M.Kom  
Yera Wahda Wahdi, S.Kom., M.Kom  
Evi Lestari Pratiwi, M.Kom  
Abrar Hadi, S.Kom., M.Kom  
Firdaus, S.Kom., M.Kom  
Kusuma Hati, M.M., M.Kom  
Andi Jamiati Paramita, S.T., M.T  
Putri Ariatna Alia

## Penyunting dan Penata Letak:

Tim PT Penamuda Media

## Desain Sampul:

Tim PT Penamuda Media

## Penerbit:

PT Penamuda Media

## Redaksi:

Casa Sidoarum RT03 Ngentak, Sidoarum Godean Sleman Yogyakarta

Web: [www.penamudamedia.com](http://www.penamudamedia.com)

E-mail: [penamudamedia@gmail.com](mailto:penamudamedia@gmail.com)

Instagram: [@penamudamedia](https://www.instagram.com/penamudamedia)

WhatsApp: +6285700592256

Cetakan Pertama, Juli 2025

X + 174 halaman; 15 x 23 cm

ISBN: 9786347269423

Hak cipta dilindungi undang-undang

Dilarang memperbanyak maupun mengedarkan buku dalam bentuk dan dengan cara apapun tanpa izin tertulis dari penerbit maupun penulis

# Kata Pengantar

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga buku Perancangan Sistem Informasi dan Aplikasinya ini dapat diselesaikan dengan baik. Buku ini disusun untuk memberikan pemahaman menyeluruh mengenai proses perancangan sistem informasi yang meliputi tahap-tahap penting mulai dari analisis kebutuhan, desain sistem dan arsitektur, pengembangan aplikasi, hingga pengujian, implementasi, dan pemeliharaan sistem.

Selain membahas aspek teknis perancangan, buku ini juga memuat pembahasan tentang manajemen proyek sistem informasi, pendekatan yang berorientasi pada pengguna, desain antarmuka (UI/UX), serta integrasi aplikasi dalam konteks nyata. Disertai dengan studi kasus dan pemaparan teknologi serta tools terkini, buku ini diharapkan mampu menjembatani teori dan praktik dalam dunia pengembangan sistem informasi modern.

Materi dalam buku ini dirancang secara sistematis agar dapat digunakan sebagai referensi pembelajaran, baik di lingkungan akademik maupun profesional. Dengan pendekatan yang aplikatif dan relevan terhadap kebutuhan industri, buku ini diharapkan dapat menjadi panduan yang

bermanfaat bagi mahasiswa, dosen, dan para praktisi di bidang teknologi informasi.

Akhir kata, kami menyadari bahwa buku ini masih memiliki kekurangan. Oleh karena itu, kami sangat terbuka terhadap masukan dan saran dari pembaca guna penyempurnaan di masa yang akan datang. Semoga buku ini dapat memberikan kontribusi nyata dalam pengembangan ilmu dan praktik sistem informasi di Indonesia.

Tim Penyusun

# Daftar Isi

Kata Pengantar .....	v
Daftar Isi.....	vii
Bab 1 Metodologi Perancangan Sistem .....	1
A.    Pengertian dan Tujuan Perancangan Sistem .....	2
B.    Pendekatan Perancangan Sistem.....	3
C.    Model-Model Pengembangan Sistem.....	5
D.    Pemilihan Metodologi yang Tepat.....	7
Bab 2 Analisis Kebutuhan Sistem .....	10
A.    Pengumpulan Kebutuhan Pengguna.....	11
B.    Dokumentasi Kebutuhan Sistem.....	13
C.    Analisis Proses Bisnis.....	14
D.    Validasi dan Verifikasi Kebutuhan .....	16
Bab 3 Desain Sistem dan Arsitektur .....	19
A.    Desain Arsitektur Sistem .....	20
B.    Perancangan Komponen dan Modul .....	21
C.    Dokumentasi Desain Sistem .....	23
Bab 4 Pengembangan Aplikasi Sistem Informasi.....	26
A.    Pemilihan Teknologi Pengembangan.....	27
B.    Pengkodean dan Implementasi Modul.....	28
C.    Desain Database dan Struktur Data .....	30

D.	Manajemen Versi dan Kolaborasi.....	32
E.	Quality Assurance dalam Pengembangan.....	33
Bab 5	Pengujian Sistem Informasi.....	36
A.	Tujuan dan Pentingnya Pengujian Sistem.....	37
B.	Jenis-jenis Pengujian Sistem.....	38
C.	Metode dan Alat Pengujian .....	40
D.	Dokumentasi dan Evaluasi Hasil Pengujian .....	42
Bab 6	Implementasi Sistem Informasi.....	44
A.	Strategi Implementasi Sistem .....	45
B.	Persiapan Data dan Migrasi Sistem .....	47
C.	Pelatihan Pengguna dan Dokumentasi.....	48
D.	Evaluasi Awal Pasca Implementasi.....	50
Bab 7	Pemeliharaan dan Pembaruan Sistem .....	52
A.	Jenis-jenis Pemeliharaan Sistem.....	53
B.	Manajemen Perubahan dan Kontrol Versi .....	54
C.	Monitoring dan Audit Sistem Berkala.....	56
D.	Strategi Pembaruan Sistem.....	58
Bab 8	Sistem Informasi dan Manajemen Proyek.....	60
A.	Peran Manajemen Proyek dalam Pengembangan Sistem .....	61
B.	Perencanaan dan Penjadwalan Proyek Sistem .....	62
C.	Pengelolaan Risiko Proyek Sistem .....	64

D.	Pengukuran Kinerja Proyek.....	66
Bab 9	Perancangan Sistem untuk Pengguna .....	68
A.	Analisis Kebutuhan Pengguna .....	69
B.	Keterlibatan Pengguna dalam Perancangan .....	70
C.	Aksesibilitas dan Kemudahan Penggunaan .....	72
D.	Evaluasi Pengalaman Pengguna.....	74
Bab 10	Desain Antarmuka Pengguna (UI/UX) .....	76
A.	Prinsip-prinsip Desain UI/UX.....	77
B.	Proses Desain UI/UX.....	83
C.	Tools Desain UI/UX .....	85
D.	Perancangan UX.....	88
E.	Evaluasi dan Iterasi Desain .....	95
Bab 11	Integrasi Aplikasi Sistem Informasi.....	98
A.	Konsep dan Tujuan Integrasi Sistem.....	99
B.	Teknologi dan Standar Integrasi.....	102
C.	Arsitektur Integrasi.....	105
Bab 12	Studi Kasus: Perancangan Sistem di Bisnis .....	108
A.	Identifikasi Kebutuhan Bisnis.....	109
B.	Analisis dan Perancangan Sistem.....	111
C.	Hasil Implementasi Sistem.....	112
D.	Pembelajaran dan Rekomendasi .....	114

Bab 13 Tools dan Teknologi dalam Perancangan Sistem ....	116
A. Tools Perancangan Visual .....	117
B. Software Pengembangan dan Prototyping .....	120
C. Teknologi Back-end dan Front-end Terkini.....	123
D. Integrasi Tools dalam Workflow Tim.....	125
Bab 14 Perancangan Sistem Berbasis Web .....	129
A. Karakteristik Sistem Web .....	130
B. Teknologi Pengembangan Web .....	132
C. Keamanan dan Skalabilitas Sistem Web .....	135
D. Optimasi Performa dan UX Web .....	138
Bab 15 Tren Masa Depan dalam Perancangan Sistem	
Informasi.....	144
A. Kecerdasan Buatan dan Otomatisasi .....	145
B. Perancangan Sistem Berbasis Cloud dan Edge .....	149
C. Desain Berkelanjutan dan Green IT.....	154
D. Tantangan dan Peluang Masa Depan .....	158
Referensi .....	163
Tentang Penulis .....	166
PROFIL PENULIS.....	<b>Error! Bookmark not defined.</b>

**BAB**

**1**

# **Metodologi Perancangan Sistem**

## A. Pengertian dan Tujuan Perancangan Sistem

**P**erancangan sistem merupakan proses merancang suatu sistem informasi yang mampu memenuhi kebutuhan pengguna dan mendukung proses bisnis organisasi secara efisien dan efektif. Proses ini mencakup penggambaran struktur sistem, alur data, proses-proses yang terjadi di dalamnya, serta spesifikasi teknis dari komponen yang dibutuhkan (Nuh, 2021). Perancangan sistem dilakukan setelah proses analisis sistem selesai, sehingga rancangan yang dibuat didasarkan pada hasil identifikasi kebutuhan, permasalahan, dan tujuan dari sistem yang akan dibangun.

Tujuan utama dari perancangan sistem adalah menciptakan sistem yang dapat memberikan solusi atas permasalahan yang dihadapi oleh organisasi atau pengguna. Sistem yang dirancang diharapkan mampu meningkatkan kinerja, efisiensi, dan kualitas pelayanan. Dengan rancangan yang tepat, sistem dapat memberikan informasi yang akurat, tepat waktu, dan relevan untuk mendukung pengambilan keputusan. Selain itu, sistem juga diharapkan mampu beradaptasi terhadap perubahan kebutuhan dan perkembangan teknologi.

Dalam proses perancangan sistem, terdapat beberapa aspek penting yang harus diperhatikan, seperti

kemudahan penggunaan (user-friendly), keamanan data, keandalan sistem, dan skalabilitas. Perancang sistem juga harus memperhatikan integrasi dengan sistem lain yang sudah ada, serta mempertimbangkan biaya dan waktu implementasi. Oleh karena itu, perancangan sistem bukan hanya aspek teknis, melainkan juga memerlukan pemahaman terhadap kebutuhan bisnis dan lingkungan organisasi.

Perancangan sistem berfungsi sebagai peta atau cetak biru dalam pembangunan sistem informasi. Rancangan yang baik menjadi fondasi penting dalam menghasilkan sistem yang fungsional, terstruktur, dan sesuai dengan ekspektasi pengguna. Tanpa proses perancangan yang matang, pembangunan sistem dapat mengalami banyak kendala, seperti kesalahan fungsi, pemborosan sumber daya, atau ketidaksesuaian dengan tujuan awal.

## **B. Pendekatan Perancangan Sistem**

Pendekatan perancangan sistem adalah metode atau strategi yang digunakan untuk mengembangkan dan merancang suatu sistem informasi sesuai dengan kebutuhan pengguna dan tujuan organisasi (Nursari & Immanuel, 2017).. Pendekatan ini sangat penting karena menentukan bagaimana sistem akan dibangun,

bagaimana komponen-komponen sistem diorganisasi, serta bagaimana interaksi antara komponen tersebut terjadi. Pemilihan pendekatan yang tepat akan mempengaruhi keberhasilan implementasi sistem dan kemampuannya dalam memenuhi kebutuhan pengguna.

Secara umum, terdapat dua pendekatan utama dalam perancangan sistem, yaitu pendekatan terstruktur dan pendekatan berorientasi objek. Pendekatan terstruktur fokus pada proses dan aliran data, serta menggunakan alat bantu seperti Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), dan struktur modular. Pendekatan ini lebih cocok untuk sistem yang berorientasi pada alur kerja yang jelas dan sistematis. Sementara itu, pendekatan berorientasi objek menekankan pada objek-objek dalam sistem, beserta atribut dan metode yang dimilikinya. Pendekatan ini lebih fleksibel, mudah untuk dikembangkan, dan lebih sesuai untuk sistem yang kompleks dan dinamis.

Dalam penerapannya, pendekatan perancangan sistem juga dapat dibagi menjadi dua kategori lain, yaitu top-down dan bottom-up. Pendekatan top-down dimulai dari gambaran umum sistem secara keseluruhan, lalu dipecah menjadi bagian-bagian yang lebih kecil. Sedangkan pendekatan bottom-up dimulai dari bagian-bagian kecil sistem, yang kemudian digabung menjadi

sistem utuh. Pendekatan top-down sering digunakan dalam proyek yang memiliki visi sistem menyeluruh sejak awal, sedangkan pendekatan bottom-up cocok untuk sistem yang dibangun dari subsistem atau modul yang telah ada sebelumnya.

Pemilihan pendekatan perancangan sistem harus mempertimbangkan beberapa faktor, seperti kompleksitas sistem, sumber daya yang tersedia, kebutuhan pengguna, serta metode pengembangan sistem yang digunakan (misalnya waterfall, agile, atau prototyping). Pendekatan yang tepat akan memudahkan proses implementasi, pemeliharaan, dan pengembangan lebih lanjut. Oleh karena itu, perancang sistem harus memiliki pemahaman yang mendalam tentang karakteristik pendekatan-pendekatan tersebut agar dapat membuat keputusan perancangan yang strategis dan efektif.

### **C. Model-Model Pengembangan Sistem**

Model pengembangan sistem adalah kerangka kerja atau metodologi yang digunakan dalam proses pembangunan sistem informasi, mulai dari tahap perencanaan hingga implementasi dan pemeliharaan. Model ini membantu tim pengembang untuk mengatur alur kerja, membagi tugas, serta mengontrol waktu dan

sumber daya secara lebih terstruktur. Dengan menggunakan model yang tepat, pengembangan sistem dapat dilakukan secara sistematis, efisien, dan sesuai dengan kebutuhan pengguna (Nursari & Immanuel, 2017).

Salah satu model pengembangan sistem yang paling klasik adalah Waterfall Model. Model ini bersifat linear dan berurutan, dimulai dari tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan. Kelebihan model ini adalah kesederhanaannya dan dokumentasi yang lengkap di setiap tahap. Namun, kelemahannya adalah kurang fleksibel dalam menghadapi perubahan kebutuhan karena setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya.

Selain Waterfall, terdapat Prototyping Model yang mengutamakan pembuatan versi awal sistem atau prototipe yang kemudian diuji dan disempurnakan berdasarkan umpan balik pengguna. Model ini cocok untuk proyek yang memiliki kebutuhan yang belum sepenuhnya jelas sejak awal. Ada juga Iterative dan Incremental Model yang membagi sistem menjadi bagian-bagian kecil (increment), yang dikembangkan secara bertahap melalui siklus berulang (iterasi), sehingga

memungkinkan pengujian dan perbaikan terus-menerus selama proses pengembangan berlangsung.

Model yang kini semakin populer adalah Agile Model, yang menekankan kolaborasi antar tim, keterlibatan pengguna secara aktif, dan respons cepat terhadap perubahan. Agile sangat efektif dalam lingkungan proyek yang dinamis dan membutuhkan fleksibilitas tinggi. Model ini membagi proyek menjadi iterasi singkat atau sprint, di mana tiap iterasi menghasilkan produk yang berfungsi sebagian dan dapat langsung diuji. Pemilihan model pengembangan sistem harus disesuaikan dengan kompleksitas proyek, kebutuhan pengguna, ketersediaan sumber daya, serta tingkat fleksibilitas yang dibutuhkan dalam proses pengembangan.

#### **D. Pemilihan Metodologi yang Tepat**

Pemilihan metodologi yang tepat dalam pengembangan sistem informasi merupakan langkah krusial yang menentukan keberhasilan proyek. Metodologi pengembangan sistem adalah kerangka kerja atau pedoman yang digunakan untuk mengatur alur kerja dan aktivitas dalam membangun sistem, mulai dari analisis kebutuhan hingga implementasi dan pemeliharaan. Setiap metodologi memiliki karakteristik, keunggulan, dan keterbatasan masing-masing, sehingga

pemilihannya harus disesuaikan dengan kondisi proyek dan kebutuhan pengguna.

Faktor utama yang perlu dipertimbangkan dalam memilih metodologi adalah kompleksitas sistem, ukuran tim pengembang, waktu yang tersedia, dan tingkat kejelasan kebutuhan pengguna. Misalnya, jika kebutuhan sudah sangat jelas dan sistem tidak terlalu kompleks, metodologi Waterfall mungkin cukup efektif karena alurnya yang terstruktur dan dokumentatif. Namun, jika kebutuhan masih berubah-ubah dan komunikasi intens dengan pengguna sangat diperlukan, metodologi Agile atau Prototyping lebih cocok karena keduanya mendukung fleksibilitas dan iterasi yang cepat.

Selain itu, ketersediaan sumber daya seperti waktu, anggaran, dan teknologi juga berpengaruh terhadap pemilihan metodologi. Proyek dengan sumber daya terbatas mungkin memerlukan pendekatan yang lebih efisien dan cepat, seperti Rapid Application Development (RAD). Di sisi lain, proyek yang memerlukan tingkat keamanan tinggi dan dokumentasi formal, seperti di sektor pemerintahan atau kesehatan, cenderung menggunakan metodologi yang lebih ketat seperti V-Model atau Spiral Model.

Dengan mempertimbangkan semua aspek tersebut, pemilihan metodologi sebaiknya dilakukan secara

kolaboratif antara manajer proyek, analis sistem, tim pengembang, dan pengguna akhir. Tujuannya adalah agar sistem yang dikembangkan tidak hanya memenuhi kebutuhan teknis, tetapi juga dapat digunakan dengan optimal dan memberikan nilai tambah bagi organisasi. Metodologi yang tepat akan meminimalkan risiko kegagalan, menghemat waktu dan biaya, serta meningkatkan kualitas hasil akhir sistem.

**BAB**

**2**

# **Analisis Kebutuhan Sistem**

## A. Pengumpulan Kebutuhan Pengguna

**P**engumpulan kebutuhan pengguna adalah proses awal dan sangat penting dalam pengembangan sistem informasi, yang bertujuan untuk memahami secara menyeluruh apa yang dibutuhkan oleh pengguna dalam sebuah sistem. Proses ini melibatkan identifikasi, dokumentasi, dan analisis atas kebutuhan fungsional (apa yang harus dilakukan sistem) dan kebutuhan non-fungsional (seperti keamanan, kecepatan, dan keandalan). Keberhasilan suatu sistem sangat bergantung pada seberapa akurat kebutuhan ini dipahami sejak awal, karena kesalahan pada tahap ini dapat menyebabkan sistem tidak sesuai harapan saat diimplementasikan (Rohman & Bhakti, 2023).

Metode yang umum digunakan dalam pengumpulan kebutuhan pengguna antara lain adalah wawancara, observasi, kuesioner, studi dokumen, dan workshop. Wawancara memungkinkan analisis sistem untuk menggali informasi secara mendalam melalui tanya jawab langsung dengan pengguna. Observasi membantu melihat secara langsung proses kerja yang berlangsung, sehingga dapat ditemukan kebutuhan yang mungkin tidak diungkapkan secara verbal. Sementara itu, kuesioner efektif digunakan jika jumlah responden banyak, dan studi dokumen

memberi pemahaman konteks kerja melalui catatan atau prosedur yang sudah ada sebelumnya.

Selama proses pengumpulan kebutuhan, penting untuk melibatkan berbagai pihak yang terkait, seperti pengguna akhir (end-user), manajer, dan pemangku kepentingan lainnya. Hal ini untuk memastikan bahwa semua perspektif dan kepentingan dapat tertampung dalam sistem yang akan dikembangkan. Di samping itu, pengumpulan kebutuhan sebaiknya dilakukan secara iteratif, agar perubahan atau penyesuaian bisa segera diakomodasi sebelum sistem masuk ke tahap perancangan.

Dokumentasi hasil pengumpulan kebutuhan biasanya dituangkan dalam bentuk System Requirement Specification (SRS), yang akan menjadi acuan utama dalam proses desain dan pembangunan sistem. Dokumen ini harus ditinjau dan disetujui bersama oleh tim pengembang dan pihak pengguna, guna memastikan bahwa tidak ada kesalahpahaman. Dengan proses pengumpulan kebutuhan yang sistematis dan menyeluruh, risiko kegagalan sistem dapat ditekan, dan peluang keberhasilan implementasi sistem akan meningkat secara signifikan.

## **B. Dokumentasi Kebutuhan Sistem**

Dokumentasi kebutuhan sistem adalah proses mencatat dan menyusun secara sistematis seluruh kebutuhan pengguna yang telah dikumpulkan selama tahap analisis. Dokumen ini berfungsi sebagai pedoman utama dalam tahap-tahap selanjutnya dari pengembangan sistem, seperti perancangan, pengkodean, dan pengujian. Tanpa dokumentasi yang jelas dan terstruktur, pengembang sistem bisa mengalami kebingungan, kesalahan interpretasi, atau bahkan membangun sistem yang tidak sesuai dengan harapan pengguna.

Isi dari dokumentasi kebutuhan sistem umumnya mencakup kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional menjelaskan apa saja yang harus dilakukan oleh sistem, seperti fitur-fitur utama dan alur kerja. Sedangkan kebutuhan non-fungsional mencakup aspek kualitas sistem, seperti performa, keamanan, skalabilitas, dan ketersediaan. Selain itu, dokumentasi juga biasanya menyertakan batasan sistem, antarmuka dengan sistem lain, serta asumsi dan ketergantungan eksternal yang relevan.

Dokumen kebutuhan sistem biasanya dituangkan dalam bentuk System Requirements Specification (SRS). SRS ini harus dibuat secara rinci, namun tetap mudah

dipahami oleh semua pihak yang terlibat, baik teknis maupun non-teknis. Oleh karena itu, penting untuk menggunakan bahasa yang jelas, serta menyertakan diagram, tabel, atau ilustrasi lain untuk memperjelas maksud dari setiap kebutuhan. Selain itu, dokumen ini perlu ditinjau dan disetujui oleh pengguna sebelum pengembangan sistem dimulai.

Dokumentasi yang baik memberikan banyak manfaat, antara lain sebagai acuan bagi tim pengembang, dasar untuk pengujian sistem, serta sebagai alat komunikasi yang efektif antara pengguna dan pengembang. Jika di kemudian hari terjadi perubahan kebutuhan atau pengembangan lanjutan, dokumentasi ini juga menjadi referensi penting untuk memastikan kesinambungan proyek. Dengan demikian, dokumentasi kebutuhan sistem bukan hanya formalitas, tetapi bagian vital dari proses pengembangan sistem yang profesional dan terstruktur.

### **C. Analisis Proses Bisnis**

Analisis proses bisnis adalah tahapan penting dalam pengembangan sistem informasi yang bertujuan untuk memahami, mengevaluasi, dan memperbaiki alur kerja atau aktivitas dalam suatu organisasi. Proses ini membantu mengidentifikasi bagaimana suatu pekerjaan

dilakukan, siapa saja yang terlibat, informasi apa yang dibutuhkan, serta bagaimana data mengalir dari satu bagian ke bagian lain. Dengan melakukan analisis proses bisnis secara menyeluruh, tim pengembang sistem dapat memastikan bahwa sistem yang dirancang benar-benar selaras dengan kebutuhan dan tujuan organisasi.

Langkah awal dalam analisis proses bisnis biasanya dimulai dengan pemetaan proses yang sudah ada (*as-is process*), yaitu menggambarkan bagaimana proses berjalan saat ini, termasuk hambatan, inefisiensi, atau aktivitas yang tidak memberikan nilai tambah. Teknik yang umum digunakan dalam tahap ini antara lain diagram alur kerja (*flowchart*), *Business Process Model and Notation (BPMN)*, atau *Data Flow Diagram (DFD)*. Dengan alat bantu tersebut, analis dapat lebih mudah mengkomunikasikan proses kepada pihak-pihak yang terlibat.

Setelah proses bisnis saat ini dipahami, tahap berikutnya adalah merancang proses bisnis yang diinginkan atau ideal (*to-be process*). Proses ini dilakukan dengan mempertimbangkan bagaimana teknologi informasi dapat digunakan untuk meningkatkan efisiensi, mengurangi biaya, mempercepat waktu pelayanan, atau meningkatkan akurasi data. Rancangan proses bisnis baru

ini nantinya menjadi dasar dalam mendesain sistem yang mampu mendukung transformasi tersebut.

Hasil dari analisis proses bisnis akan menjadi bahan penting dalam menentukan kebutuhan sistem dan fitur-fitur yang harus dimiliki. Selain itu, proses ini juga membantu organisasi mengenali kebutuhan perubahan dalam struktur kerja, tanggung jawab karyawan, atau bahkan kebijakan operasional. Dengan demikian, analisis proses bisnis tidak hanya berfungsi sebagai dasar teknis pengembangan sistem, tetapi juga sebagai alat strategis untuk mendorong peningkatan kinerja organisasi secara keseluruhan.

#### **D. Validasi dan Verifikasi Kebutuhan**

Validasi dan verifikasi kebutuhan merupakan tahap penting dalam pengembangan sistem informasi yang bertujuan untuk memastikan bahwa semua kebutuhan yang dikumpulkan benar, lengkap, dan sesuai dengan harapan pengguna. Verifikasi dilakukan untuk mengecek apakah kebutuhan telah ditulis dengan benar, jelas, dan konsisten secara teknis, sedangkan validasi bertujuan memastikan bahwa kebutuhan tersebut benar-benar mencerminkan keinginan dan kebutuhan nyata dari pengguna atau pemangku kepentingan. Tanpa proses ini,

risiko kesalahan dan ketidaksesuaian sistem akan meningkat di tahap implementasi.

Verifikasi kebutuhan biasanya dilakukan oleh tim analis atau pengembang sistem dengan meninjau dokumen kebutuhan secara sistematis. Mereka akan mencari adanya kontradiksi, ambiguitas, duplikasi, atau kelalaian dalam penulisan. Kebutuhan yang tidak dapat diuji atau diukur juga harus diperbaiki agar bisa diimplementasikan secara akurat. Beberapa teknik yang umum digunakan dalam verifikasi adalah walkthrough, peer review, dan inspeksi formal terhadap dokumen kebutuhan sistem.

Sementara itu, validasi kebutuhan dilakukan dengan melibatkan langsung pengguna akhir atau pemilik proses bisnis. Tujuannya adalah untuk memastikan bahwa semua kebutuhan yang tertulis memang mencerminkan keinginan pengguna dan mendukung proses bisnis yang dijalankan. Validasi dapat dilakukan melalui wawancara lanjutan, presentasi prototipe awal, simulasi, atau bahkan pengujian terhadap model proses yang disederhanakan. Komunikasi yang efektif antara pengguna dan pengembang sangat penting dalam proses ini.

Proses validasi dan verifikasi yang baik akan menghasilkan dokumen kebutuhan yang solid dan dapat diandalkan sebagai dasar pengembangan sistem. Ini juga

membantu mengurangi risiko revisi besar di tahap akhir, yang biasanya memakan biaya dan waktu lebih banyak. Dengan kata lain, validasi dan verifikasi bukan hanya langkah teknis, tetapi merupakan bentuk jaminan mutu agar sistem yang dibangun benar-benar relevan, berguna, dan sukses diimplementasikan.

**BAB**

**3**

# **Desain Sistem dan Arsitektur**

## A. Desain Arsitektur Sistem

**D**esain arsitektur sistem adalah proses perancangan struktur utama dari suatu sistem informasi, yang mencakup komponen-komponen utama sistem, hubungan antar komponen, serta teknologi yang digunakan untuk menjalankan sistem tersebut. Tujuannya adalah untuk memastikan bahwa sistem dapat berfungsi secara efisien, aman, dan mudah dikembangkan di masa depan. Desain arsitektur menjadi fondasi teknis yang penting, karena menentukan bagaimana sistem akan diimplementasikan dan diintegrasikan dalam lingkungan organisasi (Dengen & Hatta, 2009).

Dalam menyusun desain arsitektur, beberapa aspek utama yang harus dipertimbangkan antara lain adalah arsitektur perangkat keras, perangkat lunak, jaringan, dan basis data. Salah satu pendekatan yang umum digunakan adalah arsitektur tiga lapis (three-tier architecture), yang membagi sistem menjadi tiga bagian: lapisan presentasi (antarmuka pengguna), lapisan logika aplikasi (pengolahan data), dan lapisan data (penyimpanan dan pengambilan data). Pendekatan ini memberikan kejelasan pemisahan tanggung jawab dan memudahkan dalam pemeliharaan serta pengembangan sistem secara modular.

Selain struktur sistem, desain arsitektur juga mencakup pemilihan teknologi yang tepat seperti bahasa pemrograman, framework, server, dan platform penyimpanan data. Faktor-faktor seperti skalabilitas, keandalan, keamanan, dan interoperabilitas sangat memengaruhi keputusan dalam tahap ini. Misalnya, sistem yang diproyeksikan untuk menangani jutaan pengguna harus dirancang dengan arsitektur yang mendukung load balancing dan replikasi data, sedangkan sistem dengan data sensitif harus mengutamakan lapisan keamanan tambahan.

Dokumen desain arsitektur sistem biasanya disusun oleh arsitek sistem dan tim teknis, kemudian ditinjau oleh pihak manajemen dan pengembang untuk memastikan kesesuaian antara desain teknis dan kebutuhan bisnis. Desain ini juga menjadi pedoman bagi pengembang saat mulai melakukan implementasi sistem. Oleh karena itu, desain arsitektur sistem tidak hanya harus kuat dari sisi teknis, tetapi juga harus fleksibel untuk menghadapi perubahan dan perkembangan teknologi di masa mendatang.

## **B. Perancangan Komponen dan Modul**

Perancangan komponen dan modul merupakan bagian penting dari proses desain sistem yang bertujuan

untuk membagi sistem menjadi bagian-bagian kecil yang lebih mudah dikelola, dikembangkan, dan diuji. Setiap komponen atau modul dirancang untuk menjalankan fungsi tertentu dalam sistem dan bekerja secara mandiri namun tetap saling terintegrasi. Pembagian sistem ke dalam modul-modul ini memudahkan pengelolaan kompleksitas sistem serta mendukung pengembangan secara bertahap (*incremental*).

Dalam proses ini, setiap modul harus dirancang dengan memperhatikan prinsip modularitas, yaitu agar modul memiliki fungsi yang spesifik, independen, dan mudah dipahami. Selain itu, dua konsep penting yang digunakan dalam perancangan adalah *cohesion* dan *coupling*. *Cohesion* mengacu pada sejauh mana elemen dalam suatu modul saling terkait dan bekerja untuk tujuan yang sama — semakin tinggi *cohesion*, semakin baik. Sedangkan *coupling* adalah tingkat ketergantungan antar modul — semakin rendah *coupling*, semakin baik karena modul menjadi lebih fleksibel dan mudah dimodifikasi tanpa memengaruhi bagian lain.

Desain komponen juga harus mempertimbangkan bagaimana antar modul akan berinteraksi, termasuk format data yang digunakan, protokol komunikasi, dan antarmuka (*interface*) antar modul. Untuk sistem berbasis web, misalnya, modul dapat dibagi menjadi modul

antarmuka pengguna (frontend), modul logika aplikasi (backend), dan modul layanan data (API dan database). Setiap modul ini kemudian dirancang dan diuji secara terpisah sebelum digabungkan dalam sistem secara keseluruhan.

Perancangan modul dan komponen yang baik akan memberikan banyak manfaat, seperti kemudahan dalam pengujian unit, pemeliharaan yang lebih cepat, serta peningkatan efisiensi pengembangan karena beberapa tim dapat bekerja secara paralel. Di masa depan, modularitas juga memungkinkan sistem untuk ditingkatkan atau dimodifikasi tanpa harus membangun ulang seluruh sistem. Oleh karena itu, tahap ini bukan hanya aspek teknis, tetapi juga merupakan strategi jangka panjang dalam menjaga kualitas dan fleksibilitas sistem informasi.

### **C. Dokumentasi Desain Sistem**

Dokumentasi desain sistem adalah proses pencatatan secara lengkap dan sistematis seluruh aspek yang telah dirancang selama tahap perancangan sistem. Dokumen ini mencakup gambaran menyeluruh mengenai arsitektur sistem, komponen dan modul, antarmuka pengguna, desain database, serta spesifikasi teknis lainnya. Dokumentasi ini berfungsi sebagai panduan

utama bagi tim pengembang dalam proses implementasi, pengujian, dan pemeliharaan sistem sehingga semua pihak yang terlibat memiliki pemahaman yang sama tentang sistem yang dibangun.

Dokumentasi desain sistem harus dibuat secara jelas, terstruktur, dan mudah dipahami oleh berbagai pihak, baik yang berlatar belakang teknis maupun non-teknis. Dokumen ini biasanya meliputi diagram seperti diagram alir data (DFD), diagram arsitektur, diagram hubungan entitas (ERD), serta deskripsi rinci tentang setiap modul dan fungsinya. Selain itu, dokumentasi juga memuat standar, kebijakan, dan prosedur yang harus diikuti selama pengembangan sistem.

Selain menjadi pedoman bagi tim pengembang, dokumentasi desain sistem juga berperan penting dalam proses validasi dan verifikasi sistem. Dengan adanya dokumentasi ini, manajemen dan pengguna dapat melakukan review untuk memastikan bahwa rancangan sistem sudah sesuai dengan kebutuhan dan tujuan bisnis. Dokumentasi yang baik juga membantu meminimalkan kesalahan selama pembangunan sistem dan mempercepat proses debugging dan troubleshooting.

Dokumentasi desain sistem tidak hanya penting selama fase pengembangan, tetapi juga sangat berharga untuk pemeliharaan dan pengembangan lanjutan di masa

depan. Ketika sistem perlu diperbarui atau ditingkatkan, dokumentasi ini menjadi referensi utama untuk memahami struktur dan fungsi sistem secara menyeluruh. Oleh karena itu, menjaga dokumentasi tetap up-to-date dan akurat adalah bagian penting dari manajemen siklus hidup sistem informasi.

**BAB**

**4**

# **Pengembangan Aplikasi Sistem Informasi**

## A. Pemilihan Teknologi Pengembangan

Pemilihan teknologi pengembangan adalah proses menentukan perangkat lunak, perangkat keras, platform, dan alat bantu teknis lainnya yang akan digunakan untuk membangun sistem informasi. Keputusan ini sangat krusial karena akan mempengaruhi kinerja, skalabilitas, keamanan, serta biaya pengembangan dan pemeliharaan sistem. Teknologi yang dipilih harus sesuai dengan kebutuhan fungsional dan non-fungsional sistem, serta mampu mendukung tujuan jangka panjang organisasi.

Beberapa faktor utama yang perlu dipertimbangkan dalam pemilihan teknologi antara lain adalah kompatibilitas dengan sistem yang sudah ada, tingkat keahlian tim pengembang, ketersediaan dokumentasi dan komunitas pendukung, serta anggaran yang tersedia. Misalnya, jika organisasi sudah menggunakan infrastruktur berbasis Windows, maka pemilihan teknologi seperti .NET dan SQL Server bisa menjadi pilihan yang logis. Sementara itu, untuk pengembangan aplikasi berbasis web atau mobile, teknologi seperti JavaScript, React, Node.js, atau Flutter bisa lebih relevan (Setiyanto et al., 2019).

Selain itu, pertimbangan terhadap skalabilitas dan keamanan juga sangat penting. Sistem yang diproyeksikan

untuk tumbuh dalam jumlah pengguna atau volume data yang besar harus menggunakan teknologi yang mendukung distribusi beban dan penyimpanan data secara efisien, seperti cloud computing atau microservices. Untuk sistem yang menangani informasi sensitif, pemilihan teknologi harus memperhatikan fitur keamanan seperti enkripsi data, otentikasi berlapis, dan dukungan terhadap standar keamanan industri.

Pemilihan teknologi yang tepat akan mempercepat proses pengembangan, meningkatkan kualitas sistem, dan mempermudah pemeliharaan di masa depan. Sebaliknya, keputusan yang kurang tepat dapat menyebabkan keterlambatan, peningkatan biaya, bahkan kegagalan sistem. Oleh karena itu, proses pemilihan teknologi harus dilakukan secara hati-hati dan berbasis analisis kebutuhan teknis maupun strategis, dengan melibatkan tim pengembang, analis sistem, serta pemangku kepentingan terkait.

## **B. Pengkodean dan Implementasi Modul**

Pengkodean dan implementasi modul merupakan tahap lanjutan dalam pengembangan sistem informasi, di mana rancangan sistem yang telah dibuat sebelumnya diterjemahkan ke dalam bentuk kode program yang dapat dijalankan oleh komputer. Proses ini dilakukan

berdasarkan dokumentasi desain sistem yang mencakup struktur modul, alur logika, antarmuka, serta aturan bisnis. Setiap modul dikembangkan secara terpisah agar mudah diuji dan diperbaiki, namun tetap dirancang agar dapat saling terintegrasi dengan modul lainnya dalam sistem.

Pengkodean dilakukan menggunakan bahasa pemrograman yang telah ditentukan dalam tahap pemilihan teknologi, seperti Java, Python, PHP, C#, atau JavaScript. Dalam tahap ini, penting untuk menerapkan prinsip-prinsip pengkodean yang baik, seperti keterbacaan kode (*readability*), penggunaan struktur yang rapi, serta dokumentasi kode internal (*komentar*) untuk memudahkan proses pemeliharaan. Selain itu, pengembang juga harus memastikan bahwa setiap modul mengimplementasikan fitur sesuai dengan kebutuhan fungsional yang telah ditentukan.

Setelah modul selesai dikodekan, proses implementasi dilakukan dengan mengintegrasikan modul ke dalam lingkungan sistem secara bertahap. Modul-modul tersebut diuji secara individual melalui pengujian unit (*unit testing*) untuk memastikan bahwa setiap bagian berfungsi sebagaimana mestinya. Selanjutnya, modul-modul yang telah lulus pengujian akan digabungkan untuk

diuji dalam skala yang lebih besar melalui integrasi dan pengujian sistem secara menyeluruh.

Keberhasilan pengkodean dan implementasi modul sangat bergantung pada kolaborasi yang baik antara tim pengembang, penguji, dan analis sistem. Proses ini juga harus memperhatikan kontrol versi (version control) menggunakan alat bantu seperti Git, agar setiap perubahan kode dapat dilacak dan dikelola dengan baik. Dengan pendekatan yang sistematis, pengkodean dan implementasi modul dapat menghasilkan sistem yang andal, efisien, dan siap digunakan sesuai dengan tujuan awal pengembangan.

### **C. Desain Database dan Struktur Data**

Desain database dan struktur data adalah proses penting dalam pengembangan sistem informasi yang bertujuan untuk merancang cara penyimpanan, pengorganisasian, dan pengelolaan data agar efisien, konsisten, dan mudah diakses. Database merupakan inti dari sistem informasi karena menyimpan semua data yang diperlukan oleh sistem dan pengguna. Struktur data yang baik memungkinkan sistem menjalankan fungsi-fungsi utamanya, seperti pencarian, penyimpanan, pengolahan, dan pelaporan data, secara cepat dan akurat.

Langkah awal dalam desain database adalah mengidentifikasi entitas utama dalam sistem, atribut-atribut yang dimilikinya, serta hubungan antar entitas tersebut. Proses ini sering menggunakan Entity Relationship Diagram (ERD) untuk memvisualisasikan struktur data dan keterkaitannya. Setelah struktur logis dirancang, langkah berikutnya adalah menerjemahkannya ke dalam struktur fisik, misalnya dalam bentuk tabel-tabel dalam sistem manajemen basis data (DBMS) seperti MySQL, PostgreSQL, atau SQL Server.

Desain struktur data juga harus memperhatikan prinsip normalisasi, yaitu proses pengorganisasian data agar tidak terjadi redundansi (pengulangan data) dan untuk menjaga integritas data. Namun, dalam beberapa kasus tertentu, desainer juga dapat menerapkan denormalisasi secara selektif untuk meningkatkan performa sistem, khususnya dalam sistem dengan beban baca yang tinggi. Selain itu, perlu dirancang juga indeks, relasi, dan aturan validasi untuk menjaga kecepatan akses dan konsistensi data.

Desain database yang baik mendukung fleksibilitas dan skalabilitas sistem, serta memudahkan proses pengambilan keputusan melalui data yang bersih dan terstruktur. Kesalahan dalam tahap ini dapat berdampak besar pada kinerja sistem, menyulitkan pengembangan di

masa depan, dan bahkan dapat menyebabkan kehilangan data penting. Oleh karena itu, desain database dan struktur data harus dilakukan dengan cermat, melibatkan pemahaman mendalam terhadap kebutuhan pengguna dan proses bisnis yang dijalankan.

#### **D. Manajemen Versi dan Kolaborasi**

Manajemen versi dan kolaborasi adalah aspek penting dalam pengembangan sistem informasi, terutama ketika melibatkan banyak pengembang atau tim yang bekerja secara paralel. Manajemen versi bertujuan untuk mengatur perubahan terhadap kode sumber, dokumentasi, dan file proyek lainnya agar dapat dilacak, dikendalikan, dan dipulihkan bila diperlukan. Dengan sistem manajemen versi, pengembang dapat bekerja pada bagian masing-masing tanpa khawatir akan konflik atau kehilangan data penting akibat kesalahan pengeditan (Nugraha, 2014).

Salah satu alat yang paling umum digunakan untuk manajemen versi adalah Git, yang sering diintegrasikan dengan platform kolaborasi seperti GitHub, GitLab, atau Bitbucket. Sistem ini memungkinkan pengembang untuk membuat cabang (branch) terpisah untuk fitur tertentu, melakukan perubahan, dan menggabungkannya kembali ke dalam cabang utama (main/master) setelah diuji dan

ditinjau. Fitur seperti commit, pull request, dan merge membantu menjaga kualitas dan stabilitas kode, serta memudahkan audit terhadap riwayat perubahan.

Kolaborasi dalam tim pengembang juga mencakup komunikasi yang efektif, pembagian tugas yang jelas, serta penggunaan alat bantu manajemen proyek seperti Jira, Trello, atau Asana. Alat-alat ini membantu dalam pengelolaan backlog, penjadwalan sprint, pelacakan bug, dan pelaporan progres. Dengan adanya kolaborasi yang baik dan manajemen versi yang disiplin, tim dapat menghindari tumpang tindih pekerjaan, mempercepat penyelesaian proyek, serta meningkatkan transparansi dalam setiap tahap pengembangan.

Manajemen versi dan kolaborasi bukan hanya aspek teknis, tetapi juga bagian dari strategi organisasi dalam menjaga kualitas dan efisiensi pengembangan sistem. Dengan praktik yang baik dalam kedua hal ini, proses pengembangan menjadi lebih terstruktur, terkontrol, dan adaptif terhadap perubahan kebutuhan. Hal ini pada akhirnya akan meningkatkan keberhasilan implementasi sistem dan kepuasan pengguna akhir.

## **E. Quality Assurance dalam Pengembangan**

Quality Assurance (QA) dalam pengembangan sistem adalah serangkaian proses yang bertujuan untuk

memastikan bahwa sistem yang dibangun memenuhi standar kualitas yang telah ditetapkan, baik dari sisi fungsionalitas, keandalan, keamanan, maupun kenyamanan pengguna. QA tidak hanya fokus pada hasil akhir, tetapi juga mencakup pengawasan terhadap seluruh tahapan pengembangan, mulai dari analisis kebutuhan hingga implementasi dan pemeliharaan. Tujuannya adalah untuk mencegah kesalahan sebelum terjadi, bukan hanya memperbaikinya setelah ditemukan.

Salah satu kegiatan utama dalam QA adalah pengujian (testing), yang terdiri dari berbagai jenis, seperti unit testing untuk menguji bagian-bagian kecil kode, integration testing untuk memastikan antar modul bekerja sama dengan benar, dan system testing untuk mengevaluasi keseluruhan sistem secara menyeluruh. Selain itu, dilakukan pula user acceptance testing (UAT), yaitu pengujian oleh pengguna untuk memastikan sistem sesuai dengan kebutuhan nyata mereka. Semua pengujian ini dilakukan secara sistematis dengan menggunakan skenario dan data uji yang telah disiapkan.

QA juga mencakup proses code review, standarisasi pengkodean, penggunaan alat bantu otomatisasi pengujian (automated testing tools), serta dokumentasi proses dan hasil pengujian. Dengan penerapan praktik QA yang baik, tim pengembang dapat mendeteksi kesalahan

lebih awal, mengurangi risiko kegagalan sistem, dan mempercepat waktu perbaikan. Selain itu, QA juga memastikan bahwa sistem memiliki performa yang optimal dan bebas dari kerentanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab.

Dengan adanya tim QA atau personel yang bertanggung jawab terhadap kualitas, pengembangan sistem menjadi lebih terkontrol dan terarah. QA membantu menjaga konsistensi, meningkatkan kepercayaan pengguna, dan mendukung pencapaian standar mutu organisasi atau industri. Oleh karena itu, QA bukan hanya bagian dari proses teknis, tetapi juga merupakan investasi jangka panjang dalam membangun sistem yang andal, aman, dan dapat diandalkan dalam jangka waktu lama.

**BAB**

**5**

# **Pengujian Sistem Informasi**

## **A. Tujuan dan Pentingnya Pengujian Sistem**

Pengujian sistem merupakan tahap krusial dalam proses pengembangan sistem informasi yang bertujuan untuk memastikan bahwa sistem bekerja sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Tujuan utama dari pengujian sistem adalah untuk menemukan dan memperbaiki kesalahan (bug) sebelum sistem digunakan secara luas oleh pengguna. Pengujian ini mencakup aspek fungsional dan non-fungsional, seperti keandalan, performa, keamanan, dan kompatibilitas sistem terhadap berbagai lingkungan operasional.

Pentingnya pengujian sistem terletak pada kemampuannya untuk menjamin kualitas dan stabilitas sistem sebelum diluncurkan. Tanpa proses pengujian yang memadai, sistem berisiko mengalami kegagalan saat digunakan, yang dapat mengakibatkan kerugian finansial, penurunan produktivitas, atau bahkan kerusakan reputasi organisasi. Melalui pengujian yang menyeluruh, pengembang dapat mengidentifikasi kelemahan sistem lebih dini, sehingga mempercepat perbaikan dan mencegah masalah yang lebih besar di masa depan.

Selain itu, pengujian sistem juga berfungsi sebagai sarana validasi bahwa sistem benar-benar memenuhi kebutuhan dan harapan pengguna. Ini penting agar sistem tidak hanya berjalan dengan baik secara teknis, tetapi juga

memberikan manfaat nyata bagi pengguna akhir. Dengan melibatkan pengguna dalam tahap user acceptance testing (UAT), tim pengembang dapat memastikan bahwa fitur-fitur yang dikembangkan relevan, mudah digunakan, dan sesuai dengan proses bisnis yang ada.

Pengujian sistem adalah investasi waktu dan sumber daya yang sangat bernilai dalam proses pengembangan perangkat lunak. Ia membantu menjaga konsistensi, meningkatkan kepercayaan pengguna, dan mengurangi risiko biaya tambahan akibat kesalahan sistem di kemudian hari. Oleh karena itu, setiap proyek sistem informasi yang profesional harus menyertakan rencana pengujian yang terstruktur dan dilakukan secara menyeluruh sebelum sistem dinyatakan siap digunakan (Nugroho, 2016).

## **B. Jenis-jenis Pengujian Sistem**

Pengujian sistem terdiri dari berbagai jenis yang masing-masing memiliki tujuan dan fokus yang berbeda dalam memastikan kualitas perangkat lunak. Salah satu jenis pengujian yang paling dasar adalah unit testing, yaitu pengujian terhadap bagian terkecil dari sistem, biasanya berupa fungsi atau prosedur dalam kode program. Pengujian ini dilakukan oleh pengembang untuk

memastikan bahwa setiap unit berfungsi sesuai dengan logika yang dirancang.

Selanjutnya, ada integration testing yang menguji bagaimana modul atau komponen dalam sistem saling berinteraksi. Meskipun setiap modul telah diuji secara terpisah, interaksi antar modul bisa menimbulkan masalah baru seperti kesalahan dalam pengiriman data atau logika kerja yang tidak sinkron. Integration testing memastikan bahwa semua bagian sistem dapat bekerja bersama-sama dengan lancar dan saling mendukung satu sama lain.

Jenis lain yang sangat penting adalah system testing, yaitu pengujian terhadap seluruh sistem secara menyeluruh. Pengujian ini dilakukan untuk memastikan bahwa sistem secara keseluruhan berfungsi sesuai spesifikasi yang telah ditentukan, baik dari sisi fungsionalitas, keamanan, performa, hingga kompatibilitas. Selain itu, dilakukan juga user acceptance testing (UAT), yaitu pengujian oleh pengguna akhir untuk memastikan bahwa sistem benar-benar sesuai dengan kebutuhan mereka dan siap digunakan dalam lingkungan kerja nyata (Munawaroh, 2006).

Selain jenis-jenis di atas, terdapat pula pengujian non-fungsional seperti performance testing, security testing, dan compatibility testing. Performance testing

mengukur kecepatan dan kestabilan sistem di bawah beban tertentu. Security testing memastikan sistem tidak memiliki celah keamanan yang bisa dieksploitasi. Sementara compatibility testing mengecek apakah sistem bisa berjalan baik di berbagai perangkat, sistem operasi, atau browser. Dengan melakukan berbagai jenis pengujian ini, pengembang dapat menjamin bahwa sistem benar-benar siap digunakan dengan tingkat keandalan yang tinggi.

### **C. Metode dan Alat Pengujian**

Metode pengujian dalam pengembangan sistem informasi merujuk pada pendekatan yang digunakan untuk merancang dan melaksanakan pengujian terhadap sistem. Dua metode utama yang sering digunakan adalah black-box testing dan white-box testing. Black-box testing berfokus pada pengujian fungsionalitas sistem dari sudut pandang pengguna tanpa memperhatikan struktur internal kode. Sebaliknya, white-box testing dilakukan dengan memahami struktur logika program dan menguji jalur-pengambilan keputusan di dalamnya. Ada juga metode grey-box testing, yang menggabungkan kedua pendekatan tersebut.

Selain metode pengujian, berbagai alat bantu (tools) juga digunakan untuk meningkatkan efektivitas dan

efisiensi proses pengujian. Untuk unit testing, alat seperti JUnit (untuk Java), NUnit (untuk .NET), dan PyTest (untuk Python) sangat umum digunakan. Alat-alat ini memungkinkan pengembang untuk secara otomatis menjalankan pengujian terhadap bagian-bagian kecil dari kode dan segera mengetahui jika terjadi kegagalan. Alat otomatisasi ini sangat membantu dalam mempercepat proses pengujian dan mengurangi kesalahan manusia.

Untuk pengujian antarmuka dan fungsionalitas, alat seperti Selenium sangat populer. Selenium digunakan untuk menguji aplikasi berbasis web secara otomatis melalui browser, mensimulasikan interaksi pengguna seperti klik, input, dan navigasi halaman. Sementara itu, untuk performance testing, digunakan alat seperti JMeter dan LoadRunner, yang dapat menguji kemampuan sistem dalam menangani beban tinggi, mengidentifikasi bottleneck, dan mengukur waktu respons sistem dalam berbagai kondisi.

Alat pengujian lainnya juga mendukung continuous integration dan continuous testing dalam pengembangan perangkat lunak modern, seperti Jenkins, GitLab CI/CD, dan CircleCI. Alat-alat ini membantu menjalankan pengujian secara otomatis setiap kali ada perubahan kode, sehingga kualitas sistem dapat dipantau secara real-time. Dengan penggunaan metode dan alat yang tepat,

pengujian menjadi lebih terukur, sistematis, dan andal dalam menjamin kualitas akhir sistem informasi.

#### **D. Dokumentasi dan Evaluasi Hasil Pengujian**

Dokumentasi hasil pengujian adalah proses pencatatan seluruh aktivitas dan temuan yang terjadi selama proses pengujian sistem. Dokumentasi ini mencakup jenis pengujian yang dilakukan, skenario dan data uji, waktu pelaksanaan, hasil yang diperoleh, serta catatan terhadap bug atau error yang ditemukan. Tujuannya adalah untuk menciptakan jejak yang jelas dan terstruktur agar tim pengembang, penguji, dan manajemen dapat meninjau serta memahami kualitas sistem secara objektif dan terukur.

Setiap temuan yang muncul dalam pengujian harus dicatat dengan rinci, termasuk langkah-langkah untuk mereproduksi kesalahan, tingkat keparahan (severity), dan modul yang terdampak. Dokumentasi ini biasanya disimpan dalam sistem pelacakan bug seperti JIRA, Bugzilla, atau TestRail, yang memungkinkan tim untuk memantau status perbaikan dan retesting secara berkala. Hal ini penting agar tidak ada masalah yang terlewat, serta untuk menjamin bahwa semua perbaikan telah dilakukan sebelum sistem diserahkan ke pengguna.

Evaluasi hasil pengujian dilakukan dengan membandingkan hasil aktual dengan hasil yang

diharapkan. Jika terdapat deviasi, tim penguji dan pengembang akan menilai apakah penyebabnya adalah kesalahan dalam kode, desain, atau bahkan dalam pemahaman terhadap kebutuhan sistem. Evaluasi ini juga mempertimbangkan metrik kualitas seperti jumlah bug per modul, tingkat keberhasilan pengujian, serta waktu respons sistem. Dari sini, tim dapat mengambil keputusan apakah sistem siap untuk diluncurkan atau masih memerlukan perbaikan.

Dokumentasi dan evaluasi hasil pengujian berfungsi sebagai alat pengendali mutu dan pengambilan keputusan. Dengan dokumentasi yang baik, proses pengembangan menjadi lebih transparan dan akuntabel, serta memudahkan pelacakan kesalahan jika di kemudian hari muncul masalah. Evaluasi yang sistematis juga menjadi dasar bagi perbaikan berkelanjutan dalam proses pengembangan perangkat lunak, sehingga menghasilkan sistem yang lebih andal, efisien, dan sesuai kebutuhan pengguna.

**BAB**

**6**

# **Implementasi Sistem Informasi**

## A. Strategi Implementasi Sistem

**S**trategi implementasi sistem adalah rencana dan pendekatan yang digunakan untuk memasang dan mengintegrasikan sistem baru ke dalam lingkungan operasional organisasi. Tujuan utama dari strategi ini adalah memastikan transisi dari sistem lama ke sistem baru berjalan lancar tanpa mengganggu aktivitas bisnis secara signifikan. Strategi implementasi yang baik juga meminimalkan risiko kegagalan, memudahkan adaptasi pengguna, dan memastikan sistem baru dapat berfungsi secara optimal sejak awal digunakan (Ahmad et al., 2022).

Ada beberapa strategi implementasi yang umum diterapkan, antara lain direct cutover, parallel running, phased implementation, dan pilot implementation. Metode direct cutover berarti sistem lama langsung digantikan oleh sistem baru dalam satu waktu, yang cepat tapi berisiko tinggi jika terjadi masalah. Parallel running adalah menjalankan kedua sistem secara bersamaan selama periode tertentu untuk membandingkan hasil dan memastikan sistem baru sudah berjalan stabil sebelum sistem lama dihentikan. Phased implementation dilakukan secara bertahap berdasarkan modul atau fungsi tertentu, sementara pilot implementation hanya menerapkan sistem baru di bagian kecil organisasi terlebih dahulu sebagai uji coba.

Pemilihan strategi implementasi sangat bergantung pada kompleksitas sistem, kesiapan organisasi, risiko bisnis, dan sumber daya yang tersedia. Misalnya, untuk sistem yang sangat kritis dengan risiko tinggi, parallel running atau phased implementation lebih disarankan untuk menghindari gangguan besar. Sedangkan untuk sistem yang sederhana dan lingkungan yang mendukung, direct cutover bisa menjadi pilihan agar implementasi lebih cepat dan biaya lebih rendah. Selain itu, pelatihan pengguna dan dukungan teknis selama masa implementasi juga harus diperhitungkan dalam strategi ini.

Strategi implementasi yang matang tidak hanya melibatkan aspek teknis, tetapi juga manajemen perubahan dan komunikasi yang efektif dengan seluruh pemangku kepentingan. Keterlibatan pengguna sejak awal dan pemberian pelatihan yang memadai akan meningkatkan penerimaan sistem baru dan mempercepat proses adaptasi. Dengan demikian, strategi implementasi sistem yang tepat akan membantu organisasi memaksimalkan manfaat teknologi baru dan meminimalkan gangguan terhadap proses bisnis.

## **B. Persiapan Data dan Migrasi Sistem**

Persiapan data dan migrasi sistem merupakan tahap penting dalam proses implementasi sistem baru yang melibatkan pemindahan data dari sistem lama ke sistem baru. Tahap ini bertujuan memastikan bahwa data yang dipindahkan akurat, lengkap, dan sesuai dengan format yang dibutuhkan oleh sistem baru. Persiapan data yang baik akan mencegah masalah seperti kehilangan data, inkonsistensi, atau korupsi data yang dapat mengganggu operasional sistem setelah implementasi.

Proses persiapan data dimulai dengan melakukan audit dan pembersihan data pada sistem lama. Data yang sudah usang, duplikat, atau tidak relevan harus diidentifikasi dan dihapus untuk meningkatkan kualitas data yang akan dimigrasi. Selain itu, perlu dilakukan validasi dan normalisasi data agar sesuai dengan struktur dan format yang dipersyaratkan oleh sistem baru. Dalam beberapa kasus, data juga perlu diubah atau dikonversi menggunakan skrip khusus agar kompatibel dengan database baru.

Migrasi data dilakukan secara bertahap dengan rencana yang matang agar proses berjalan lancar dan minim risiko. Metode migrasi bisa dilakukan secara batch dengan memindahkan sejumlah data dalam satu waktu, atau secara real-time jika sistem lama dan baru dapat

diintegrasikan selama masa transisi. Penting juga melakukan pengujian migrasi untuk memastikan data sudah benar-benar tertransfer dengan baik dan sistem baru dapat mengakses data tersebut tanpa masalah. Pengujian ini meliputi verifikasi kuantitas, kualitas, serta fungsionalitas data.

Selain aspek teknis, persiapan dan migrasi data juga harus melibatkan koordinasi yang baik antara tim pengembang, administrator database, dan pengguna bisnis. Komunikasi yang efektif membantu mengidentifikasi kebutuhan data yang paling krusial dan menghindari kesalahan yang dapat menyebabkan gangguan operasional. Dengan perencanaan dan pelaksanaan migrasi data yang tepat, organisasi dapat memastikan kelangsungan bisnis tanpa kehilangan informasi penting selama transisi ke sistem baru.

### **C. Pelatihan Pengguna dan Dokumentasi**

Pelatihan pengguna merupakan bagian vital dari proses implementasi sistem yang bertujuan memastikan bahwa pengguna akhir memahami cara menggunakan sistem baru dengan efektif dan efisien. Pelatihan ini tidak hanya mengajarkan fungsi dasar sistem, tetapi juga bagaimana menjalankan tugas sehari-hari, menangani masalah umum, serta memanfaatkan fitur-fitur yang ada

untuk meningkatkan produktivitas. Dengan pelatihan yang baik, resistensi terhadap perubahan dapat diminimalkan dan adopsi sistem baru menjadi lebih cepat.

Proses pelatihan biasanya dilakukan melalui berbagai metode, seperti pelatihan langsung (classroom training), workshop, tutorial online, serta penggunaan modul e-learning. Pemilihan metode tergantung pada jumlah pengguna, kompleksitas sistem, serta ketersediaan sumber daya. Selain itu, pelatihan juga harus disesuaikan dengan profil pengguna, karena tiap kelompok pengguna mungkin memiliki kebutuhan dan tingkat pemahaman teknologi yang berbeda-beda. Pelatihan yang terstruktur dan interaktif dapat meningkatkan motivasi serta kemampuan pengguna dalam mengoperasikan sistem.

Dokumentasi pengguna merupakan pendukung penting dari pelatihan, berupa panduan tertulis, manual, FAQ, dan dokumentasi online yang mudah diakses. Dokumentasi ini berfungsi sebagai referensi yang dapat digunakan kapan saja oleh pengguna untuk menyelesaikan masalah atau memahami fitur tertentu tanpa harus selalu bergantung pada pelatih. Dokumentasi yang lengkap dan jelas juga membantu menjaga konsistensi penggunaan sistem di seluruh organisasi,

serta mempercepat proses pembelajaran bagi pengguna baru.

Dengan adanya pelatihan pengguna yang efektif dan dokumentasi yang memadai, proses transisi ke sistem baru dapat berjalan lebih lancar. Keduanya saling melengkapi dalam meningkatkan kesiapan pengguna dan mengurangi risiko kesalahan operasional yang disebabkan oleh kurangnya pemahaman terhadap sistem. Hal ini pada akhirnya akan mendukung pencapaian tujuan implementasi sistem, yaitu meningkatkan efisiensi, produktivitas, dan kualitas layanan organisasi.

#### **D. Evaluasi Awal Pasca Implementasi**

Evaluasi awal pasca implementasi adalah tahap penting yang dilakukan setelah sistem baru mulai digunakan secara operasional. Tujuan utama evaluasi ini adalah untuk menilai sejauh mana sistem yang diimplementasikan berhasil memenuhi kebutuhan pengguna dan tujuan bisnis yang telah ditetapkan sebelumnya. Evaluasi juga berfungsi untuk mengidentifikasi masalah atau kendala yang muncul selama penggunaan awal, sehingga dapat segera ditindaklanjuti untuk memperbaiki kualitas sistem.

Pada tahap ini, berbagai metode evaluasi dapat digunakan, seperti pengumpulan umpan balik langsung dari pengguna melalui wawancara, kuesioner, atau diskusi

kelompok. Selain itu, analisis data operasional sistem seperti waktu respon, frekuensi kesalahan, dan tingkat penggunaan fitur juga penting untuk memberikan gambaran objektif mengenai performa sistem. Evaluasi awal memberikan dasar untuk memahami bagaimana sistem beradaptasi dengan proses bisnis nyata dan apakah ada kebutuhan penyesuaian lebih lanjut.

Hasil evaluasi ini biasanya dijadikan acuan bagi tim pengembang dan manajemen untuk melakukan perbaikan dan peningkatan sistem secara berkelanjutan. Misalnya, jika ditemukan fitur yang sulit digunakan atau bug yang mengganggu operasional, maka langkah perbaikan segera dirancang dan diimplementasikan. Selain itu, evaluasi awal juga bisa mengungkap kebutuhan pelatihan tambahan bagi pengguna agar pemanfaatan sistem bisa lebih optimal.

Dengan melakukan evaluasi awal pasca implementasi secara terstruktur dan berkelanjutan, organisasi dapat memastikan bahwa investasi teknologi yang telah dilakukan memberikan manfaat maksimal. Proses evaluasi ini juga membantu mengurangi risiko kegagalan jangka panjang dan meningkatkan kepuasan pengguna terhadap sistem baru. Pada akhirnya, evaluasi awal menjadi bagian dari siklus perbaikan berkelanjutan yang mendukung keberhasilan penggunaan sistem informasi dalam mendukung tujuan bisnis.

**BAB**

**7**

# **Pemeliharaan dan Pembaruan Sistem**

## A. Jenis-jenis Pemeliharaan Sistem

**P**emeliharaan sistem merupakan kegiatan penting yang dilakukan setelah sistem informasi diimplementasikan agar tetap berfungsi dengan baik dan sesuai kebutuhan pengguna. Pemeliharaan sistem tidak hanya mencakup perbaikan kesalahan (bug fixing), tetapi juga penyesuaian dan peningkatan sistem agar dapat mengikuti perubahan lingkungan bisnis maupun teknologi. Secara umum, pemeliharaan sistem dibagi menjadi beberapa jenis utama yaitu pemeliharaan korektif, adaptif, preventif, dan evolusioner.

Pemeliharaan korektif adalah jenis pemeliharaan yang fokus pada perbaikan kesalahan atau bug yang ditemukan setelah sistem digunakan. Kesalahan ini bisa berupa gangguan fungsi, kerusakan data, atau masalah keamanan yang harus segera diperbaiki agar sistem dapat berjalan normal kembali. Pemeliharaan korektif biasanya bersifat reaktif karena dilakukan sebagai respons terhadap masalah yang dilaporkan oleh pengguna atau ditemukan selama monitoring sistem.

Pemeliharaan adaptif bertujuan untuk menyesuaikan sistem dengan perubahan lingkungan eksternal, seperti perubahan peraturan, kebijakan perusahaan, atau pembaruan teknologi perangkat keras dan perangkat lunak. Pemeliharaan ini memastikan sistem tetap relevan

dan kompatibel dengan kondisi baru agar tetap dapat mendukung proses bisnis secara efektif. Contohnya adalah penyesuaian sistem agar kompatibel dengan versi terbaru sistem operasi atau browser (Arief & Sugiarti, 2022).

Jenis pemeliharaan lainnya adalah pemeliharaan preventif dan evolusioner. Pemeliharaan preventif dilakukan untuk mencegah terjadinya masalah di masa depan dengan melakukan pemeriksaan rutin, pengoptimalan kinerja, dan pengamanan sistem. Sedangkan pemeliharaan evolusioner bertujuan untuk meningkatkan sistem dengan menambahkan fitur baru atau memperbaiki fungsionalitas yang sudah ada agar sistem semakin memenuhi kebutuhan pengguna dan berkembang seiring waktu. Semua jenis pemeliharaan ini saling melengkapi untuk menjaga sistem tetap handal dan bermanfaat dalam jangka panjang.

## **B. Manajemen Perubahan dan Kontrol Versi**

Manajemen perubahan adalah proses sistematis yang digunakan untuk mengelola setiap perubahan yang terjadi pada sistem informasi, mulai dari permintaan perubahan, evaluasi, persetujuan, hingga pelaksanaan dan dokumentasi. Tujuan utama manajemen perubahan adalah memastikan bahwa setiap modifikasi dilakukan

secara terencana, terkontrol, dan tidak mengganggu kestabilan sistem. Proses ini sangat penting untuk menghindari risiko kesalahan atau konflik yang dapat merusak fungsi sistem serta menjaga konsistensi dan kualitas perangkat lunak.

Salah satu aspek penting dalam manajemen perubahan adalah penggunaan alat kontrol versi (version control system) yang memungkinkan tim pengembang untuk melacak setiap perubahan kode sumber dan dokumen terkait. Kontrol versi membantu mengelola berbagai versi file dengan menyimpan riwayat perubahan, sehingga jika ada kesalahan, pengembang dapat mengembalikan kode ke versi sebelumnya. Selain itu, kontrol versi memudahkan kolaborasi tim karena beberapa pengembang dapat bekerja secara bersamaan tanpa saling menimpa pekerjaan satu sama lain.

Alat kontrol versi populer seperti Git, SVN, dan Mercurial memberikan fitur seperti branching dan merging yang memungkinkan pengembangan fitur baru dilakukan secara paralel tanpa mengganggu versi stabil sistem utama. Branching memisahkan jalur pengembangan, sementara merging menggabungkan perubahan setelah selesai diuji. Proses ini mempercepat pengembangan sekaligus menjaga kualitas kode yang dimasukkan ke dalam sistem produksi. Dengan kontrol

versi yang baik, dokumentasi perubahan juga otomatis tercatat, memudahkan audit dan penelusuran.

Manajemen perubahan dan kontrol versi bersama-sama membentuk kerangka kerja yang esensial untuk pengelolaan sistem informasi yang dinamis dan kompleks. Keduanya membantu organisasi mengatasi tantangan perubahan yang cepat dalam teknologi dan kebutuhan bisnis, serta meningkatkan transparansi dan akuntabilitas dalam proses pengembangan. Dengan penerapan yang baik, risiko gangguan akibat perubahan dapat diminimalisir, sehingga sistem dapat terus beradaptasi dan berkembang sesuai dengan kebutuhan pengguna.

### **C. Monitoring dan Audit Sistem Berkala**

Monitoring sistem berkala adalah proses pengawasan secara terus-menerus terhadap kinerja dan kondisi sistem informasi untuk memastikan sistem berjalan sesuai harapan dan bebas dari gangguan. Aktivitas ini mencakup pemantauan penggunaan sumber daya seperti CPU, memori, dan jaringan, serta pemantauan aktivitas transaksi dan akses pengguna. Monitoring yang rutin memungkinkan tim TI untuk mendeteksi masalah sejak dini, seperti penurunan performa, potensi kegagalan, atau ancaman keamanan,

sehingga tindakan pencegahan atau perbaikan dapat segera dilakukan.

Selain monitoring, audit sistem berkala juga menjadi bagian penting dalam menjaga integritas dan keamanan sistem. Audit sistem adalah evaluasi formal dan mendalam yang dilakukan secara periodik untuk menilai kepatuhan sistem terhadap kebijakan, standar, dan regulasi yang berlaku. Audit ini mencakup pemeriksaan konfigurasi sistem, kontrol akses, keamanan data, serta ketepatan pelaksanaan prosedur operasional. Hasil audit memberikan gambaran objektif mengenai kekuatan dan kelemahan sistem serta rekomendasi perbaikan yang diperlukan.

Monitoring dan audit sistem tidak hanya berfokus pada aspek teknis, tetapi juga mencakup proses bisnis dan kepatuhan terhadap regulasi yang relevan. Dengan rutin melakukan monitoring, organisasi dapat mengidentifikasi tren penggunaan dan kebutuhan kapasitas yang mungkin berubah seiring waktu, sehingga perencanaan pengembangan sistem dan sumber daya dapat lebih tepat sasaran. Sedangkan audit membantu memastikan bahwa sistem tidak hanya berjalan lancar, tetapi juga aman dan sesuai dengan standar industri atau peraturan pemerintah.

Monitoring dan audit sistem berkala merupakan praktik terbaik dalam pengelolaan sistem informasi yang efektif dan berkelanjutan. Keduanya saling melengkapi untuk menjaga performa, keamanan, dan kepatuhan sistem, serta meminimalkan risiko gangguan atau penyalahgunaan. Dengan pendekatan ini, organisasi dapat memastikan bahwa sistem tetap andal, efisien, dan mampu mendukung tujuan bisnis dalam jangka panjang.

#### **D. Strategi Pembaruan Sistem**

Strategi pembaruan sistem adalah pendekatan yang dirancang untuk menjaga sistem informasi tetap up-to-date dan relevan dengan kebutuhan bisnis yang terus berkembang serta perubahan teknologi. Pembaruan sistem meliputi berbagai kegiatan, seperti perbaikan bug, peningkatan fitur, pembaruan keamanan, hingga migrasi ke platform atau teknologi baru. Tujuan utama dari strategi ini adalah memastikan sistem tetap efisien, aman, dan mampu memberikan nilai tambah bagi organisasi.

Ada beberapa pendekatan dalam melakukan pembaruan sistem, salah satunya adalah pembaruan bertahap (*incremental update*), di mana perubahan dilakukan dalam bagian-bagian kecil secara berkala. Pendekatan ini mengurangi risiko kegagalan besar karena setiap perubahan dapat diuji secara menyeluruh sebelum

diterapkan sepenuhnya. Selain itu, pembaruan bertahap memudahkan pengguna untuk beradaptasi dengan fitur baru tanpa gangguan signifikan pada operasional sehari-hari.

Strategi lain adalah pembaruan besar atau big bang, di mana sistem diperbarui secara menyeluruh dalam satu waktu tertentu. Meskipun pendekatan ini dapat memberikan hasil cepat, risikonya lebih tinggi karena perubahan yang besar sekaligus bisa menimbulkan masalah kompatibilitas atau gangguan operasional. Oleh karena itu, strategi big bang biasanya dilakukan dengan persiapan matang, termasuk pengujian intensif dan perencanaan rollback jika terjadi kegagalan.

Dalam strategi pembaruan sistem, komunikasi dan pelibatan pengguna sangat penting untuk keberhasilan implementasi. Memberikan informasi yang jelas mengenai perubahan, pelatihan jika diperlukan, serta dukungan teknis akan membantu pengguna menerima pembaruan dengan baik. Selain itu, organisasi perlu melakukan evaluasi pasca pembaruan untuk menilai dampak perubahan dan menentukan langkah perbaikan jika diperlukan. Dengan strategi pembaruan yang tepat, sistem dapat terus berkembang dan mendukung kebutuhan bisnis secara optimal.

**BAB**

**8**

**Sistem Informasi  
dan Manajemen  
Proyek**

## A. Peran Manajemen Proyek dalam Pengembangan Sistem

**M**anajemen proyek memegang peranan penting dalam pengembangan sistem informasi, karena proses pengembangan sistem melibatkan banyak tahapan yang kompleks dan membutuhkan koordinasi antara berbagai pihak. Peran manajemen proyek adalah mengatur, mengawasi, dan mengendalikan seluruh aktivitas pengembangan agar dapat berjalan sesuai dengan tujuan, waktu, anggaran, dan kualitas yang telah ditetapkan. Tanpa manajemen proyek yang baik, risiko kegagalan seperti keterlambatan, pembengkakan biaya, atau hasil yang tidak sesuai kebutuhan pengguna akan semakin tinggi (Setiawan & Khairuzzaman, 2017).

Salah satu fungsi utama manajemen proyek adalah perencanaan yang mencakup penetapan ruang lingkup proyek, penyusunan jadwal kerja, pembagian sumber daya, dan pengelolaan risiko. Perencanaan ini sangat krusial agar setiap langkah dalam pengembangan sistem dapat dilakukan secara sistematis dan terorganisir. Manajer proyek bertanggung jawab memastikan bahwa setiap anggota tim memahami tugas dan tenggat waktunya sehingga kerja tim dapat berjalan lancar dan terkoordinasi.

Selain perencanaan, manajemen proyek juga bertugas melakukan pengawasan dan pengendalian selama proses pengembangan berlangsung. Hal ini meliputi pemantauan kemajuan pekerjaan, pengelolaan perubahan, serta penyelesaian masalah yang muncul di lapangan. Pengawasan yang efektif memungkinkan proyek untuk tetap berada pada jalurnya dan menyesuaikan strategi apabila terjadi kendala, sehingga tujuan proyek tetap dapat tercapai dengan baik.

Peran manajemen proyek juga mencakup komunikasi yang efektif antar semua pemangku kepentingan, mulai dari tim pengembang, pengguna, hingga manajemen organisasi. Manajer proyek harus mampu menjadi penghubung yang memastikan informasi penting tersampaikan dengan jelas dan tepat waktu. Dengan demikian, manajemen proyek tidak hanya menjaga proses pengembangan tetap terkendali, tetapi juga membangun kerjasama yang baik untuk keberhasilan implementasi sistem informasi.

## **B. Perencanaan dan Penjadwalan Proyek Sistem**

Perencanaan proyek sistem adalah tahap awal yang sangat penting dalam manajemen proyek pengembangan sistem informasi. Pada tahap ini, tujuan proyek, ruang lingkup kerja, serta sumber daya yang dibutuhkan

ditentukan secara jelas dan terperinci. Perencanaan yang matang membantu mengantisipasi berbagai risiko dan tantangan yang mungkin muncul selama proses pengembangan, sehingga proyek dapat berjalan lebih lancar dan terarah. Selain itu, perencanaan juga mencakup penentuan metode kerja, pembagian tugas, serta anggaran yang akan digunakan.

Penjadwalan proyek merupakan bagian integral dari perencanaan yang mengatur waktu pelaksanaan setiap aktivitas dalam proyek. Dengan penjadwalan yang baik, manajer proyek dapat menentukan urutan pekerjaan, durasi tiap tahapan, dan tenggat waktu penyelesaian. Penjadwalan yang efektif memudahkan pengawasan kemajuan proyek serta memungkinkan identifikasi potensi keterlambatan atau hambatan sedini mungkin. Alat bantu seperti diagram Gantt dan metode Critical Path Method (CPM) sering digunakan untuk memvisualisasikan dan mengelola jadwal proyek.

Dalam perencanaan dan penjadwalan, penting juga untuk memasukkan buffer waktu atau cadangan waktu untuk menghadapi ketidakpastian yang tidak terduga. Hal ini penting karena dalam pengembangan sistem, seringkali terjadi perubahan kebutuhan, kendala teknis, atau masalah lain yang memerlukan penyesuaian jadwal.

Dengan adanya buffer, proyek tetap memiliki fleksibilitas tanpa harus mengorbankan kualitas atau hasil akhir.

Perencanaan dan penjadwalan proyek sistem merupakan fondasi yang menentukan kesuksesan proyek. Keduanya membantu memastikan bahwa sumber daya digunakan secara efisien, komunikasi antar tim berjalan baik, dan seluruh proses pengembangan dapat diselesaikan tepat waktu sesuai dengan harapan pengguna dan organisasi. Tanpa perencanaan dan penjadwalan yang baik, proyek berisiko mengalami pembengkakan biaya, keterlambatan, dan hasil yang tidak memuaskan.

### **C. Pengelolaan Risiko Proyek Sistem**

Pengelolaan risiko proyek sistem adalah proses identifikasi, analisis, dan pengendalian risiko yang mungkin terjadi selama pengembangan sistem informasi. Risiko dalam proyek bisa berupa berbagai hal, mulai dari keterlambatan jadwal, kelebihan biaya, kegagalan teknis, hingga perubahan kebutuhan pengguna yang mendadak. Pengelolaan risiko yang efektif sangat penting agar potensi masalah tersebut dapat diminimalkan sehingga proyek tetap berjalan sesuai rencana dan mencapai tujuan yang diinginkan.

Tahap pertama dalam pengelolaan risiko adalah identifikasi risiko, yaitu mengenali semua kemungkinan risiko yang dapat mengganggu kelancaran proyek. Ini dapat dilakukan melalui brainstorming, wawancara, atau analisis pengalaman dari proyek sebelumnya. Setelah risiko teridentifikasi, langkah berikutnya adalah analisis risiko untuk menentukan dampak dan probabilitas terjadinya risiko tersebut. Dengan pemahaman ini, tim proyek dapat memprioritaskan risiko mana yang perlu mendapatkan perhatian lebih besar.

Setelah risiko diprioritaskan, langkah berikutnya adalah merencanakan respon terhadap risiko tersebut. Respon risiko bisa berupa penghindaran (menghilangkan penyebab risiko), mitigasi (mengurangi dampak atau kemungkinan risiko), transfer (mengalihkan risiko ke pihak lain, misalnya asuransi), atau menerima risiko dengan persiapan tindakan kontinjensi. Perencanaan ini harus didokumentasikan dengan jelas dan dikomunikasikan kepada seluruh anggota tim agar siap menghadapi situasi yang mungkin terjadi.

Pengelolaan risiko juga meliputi monitoring dan pengendalian risiko selama proyek berlangsung. Risiko yang sudah diidentifikasi harus terus dipantau untuk memastikan bahwa respon yang direncanakan berjalan efektif, dan jika muncul risiko baru, harus segera

ditangani. Dengan pengelolaan risiko yang proaktif dan terstruktur, proyek pengembangan sistem dapat meminimalkan gangguan, meningkatkan peluang keberhasilan, dan mengoptimalkan penggunaan sumber daya yang ada.

#### **D. Pengukuran Kinerja Proyek**

Pengukuran kinerja proyek merupakan proses penting dalam manajemen proyek yang bertujuan untuk menilai sejauh mana proyek berjalan sesuai dengan rencana yang telah ditetapkan. Pengukuran ini membantu tim proyek dan manajemen untuk memahami efektivitas pelaksanaan proyek, mengevaluasi kemajuan, serta menentukan apakah tujuan proyek dapat dicapai dalam batas waktu dan anggaran yang tersedia. Tanpa pengukuran yang akurat, proyek dapat kehilangan arah dan sulit dikendalikan secara efektif.

Salah satu metode yang umum digunakan dalam pengukuran kinerja proyek adalah Key Performance Indicators (KPI), seperti pencapaian target waktu (jadwal), pencapaian target anggaran (biaya), serta tingkat penyelesaian pekerjaan (scope completion). Selain itu, metode Earned Value Management (EVM) juga sering digunakan karena mampu memberikan gambaran yang lebih komprehensif tentang kinerja proyek dari segi

waktu dan biaya secara bersamaan. Dengan membandingkan nilai pekerjaan yang direncanakan, yang telah diselesaikan, dan biaya aktualnya, manajer proyek dapat menilai efisiensi proyek.

Pengukuran kinerja tidak hanya terbatas pada aspek teknis atau finansial, tetapi juga mencakup kualitas hasil kerja, kepuasan pemangku kepentingan, serta efektivitas komunikasi dan kolaborasi tim. Oleh karena itu, pengumpulan data kualitatif dari pengguna, tim proyek, maupun manajemen juga menjadi bagian penting dalam proses evaluasi. Kombinasi data kuantitatif dan kualitatif akan memberikan gambaran menyeluruh terhadap kondisi proyek secara real-time.

Dengan melakukan pengukuran kinerja proyek secara berkala, organisasi dapat mengambil keputusan yang lebih tepat dan responsif, baik dalam bentuk perbaikan strategi, penyesuaian alokasi sumber daya, atau pengelolaan risiko tambahan. Pengukuran kinerja juga menjadi dasar bagi pelaporan kepada manajemen dan pemangku kepentingan lainnya, serta mendukung proses pembelajaran untuk proyek-proyek mendatang. Pada akhirnya, pengukuran kinerja yang baik memastikan proyek tetap berada di jalur yang benar untuk mencapai tujuannya secara efisien dan efektif.

**BAB**

**9**

# **Perancangan Sistem untuk Pengguna**

## A. Analisis Kebutuhan Pengguna

**A**nalisis kebutuhan pengguna adalah tahap awal dan krusial dalam proses pengembangan sistem informasi, yang bertujuan untuk memahami dan mendefinisikan secara tepat apa yang dibutuhkan oleh pengguna akhir dari sistem yang akan dibangun (Nopriandi, 2018). Pada tahap ini, pengembang sistem bekerja sama dengan pengguna untuk menggali informasi terkait proses bisnis, permasalahan yang dihadapi, serta harapan mereka terhadap sistem yang baru. Hasil dari analisis ini menjadi dasar bagi desain dan pengembangan sistem agar sesuai dengan tujuan organisasi.

Proses analisis kebutuhan biasanya melibatkan berbagai teknik pengumpulan data, seperti wawancara, observasi, kuesioner, dan studi dokumentasi. Wawancara mendalam dengan pengguna kunci sangat membantu dalam memahami kebutuhan eksplisit dan implisit, sedangkan observasi langsung dapat mengungkap proses kerja aktual yang tidak selalu terdokumentasi dengan baik. Dengan menggabungkan berbagai teknik ini, tim analis dapat memperoleh gambaran yang menyeluruh dan akurat mengenai kebutuhan pengguna.

Penting untuk membedakan antara kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional berkaitan dengan apa yang harus dilakukan oleh sistem,

seperti mengelola data pelanggan atau menghasilkan laporan keuangan. Sedangkan kebutuhan non-fungsional mencakup aspek-aspek seperti kecepatan sistem, keamanan data, kemudahan penggunaan, dan skalabilitas. Kedua jenis kebutuhan ini harus dianalisis secara seimbang agar sistem yang dibangun tidak hanya mampu menjalankan fungsinya, tetapi juga memenuhi standar kualitas yang diharapkan pengguna.

Analisis kebutuhan pengguna yang dilakukan secara cermat akan mengurangi risiko kesalahan desain, pengembangan ulang, dan ketidakpuasan pengguna di kemudian hari. Ini juga membantu menjaga keselarasan antara sistem dan proses bisnis organisasi. Oleh karena itu, keterlibatan aktif pengguna dalam proses analisis sangat diperlukan agar sistem yang dibangun benar-benar dapat memberikan manfaat maksimal dan mendukung pencapaian tujuan strategis organisasi.

## **B. Keterlibatan Pengguna dalam Perancangan**

Keterlibatan pengguna dalam tahap perancangan sistem informasi merupakan faktor kunci dalam menciptakan sistem yang efektif, relevan, dan mudah digunakan. Pengguna adalah pihak yang akan menggunakan sistem secara langsung dalam operasional sehari-hari, sehingga pemahaman mereka tentang alur

kerja dan kebutuhan praktis sangat berharga bagi tim pengembang. Dengan melibatkan pengguna sejak awal, sistem yang dirancang dapat lebih sesuai dengan kenyataan di lapangan dan menjawab permasalahan secara konkret.

Salah satu bentuk keterlibatan pengguna adalah melalui partisipasi aktif dalam sesi diskusi, workshop, atau prototyping. Dalam sesi ini, pengguna dapat memberikan masukan langsung terhadap rancangan antarmuka, alur proses, dan fitur-fitur yang direncanakan. Prototyping memungkinkan pengguna untuk melihat dan menguji rancangan awal sistem, lalu memberikan umpan balik yang dapat digunakan untuk perbaikan sebelum sistem dikembangkan lebih lanjut. Proses ini sangat efektif dalam menghindari miskomunikasi antara pengembang dan pengguna.

Keterlibatan pengguna juga meningkatkan rasa kepemilikan (*sense of ownership*) terhadap sistem. Ketika pengguna merasa didengarkan dan dilibatkan dalam proses, mereka cenderung lebih menerima perubahan dan lebih cepat beradaptasi dengan sistem baru. Ini membantu mengurangi resistensi dan meningkatkan keberhasilan implementasi sistem. Selain itu, pengguna yang terlibat biasanya lebih memahami cara kerja sistem sehingga pelatihan pun menjadi lebih efisien.

Namun, agar keterlibatan pengguna berjalan efektif, diperlukan komunikasi yang terbuka, fasilitasi yang baik, dan waktu yang memadai. Tim pengembang perlu mampu menjembatani bahasa teknis dengan bahasa bisnis agar pengguna dapat memahami dan memberi masukan yang tepat. Dengan strategi yang tepat, keterlibatan pengguna dalam perancangan bukan hanya meningkatkan kualitas sistem, tetapi juga mempercepat adopsi dan pencapaian manfaat maksimal dari sistem informasi yang dibangun.

### **C. Aksesibilitas dan Kemudahan Penggunaan**

Aksesibilitas dan kemudahan penggunaan (usability) adalah dua aspek penting dalam perancangan sistem informasi yang berorientasi pada pengguna. Aksesibilitas merujuk pada kemampuan sistem untuk dapat diakses dan digunakan oleh semua orang, termasuk mereka yang memiliki keterbatasan fisik, sensorik, atau kognitif. Sementara itu, kemudahan penggunaan menekankan pada seberapa intuitif dan efisien antarmuka sistem dalam membantu pengguna menyelesaikan tugas-tugas mereka. Kedua aspek ini berkontribusi langsung terhadap tingkat kepuasan dan produktivitas pengguna.

Untuk mendukung aksesibilitas, sistem harus dirancang sesuai dengan standar dan pedoman internasional seperti Web Content Accessibility

Guidelines (WCAG). Contohnya adalah penggunaan teks alternatif untuk gambar, kontras warna yang cukup, kemampuan navigasi dengan keyboard, serta dukungan pembaca layar. Penerapan prinsip-prinsip ini memastikan bahwa sistem dapat digunakan oleh semua kalangan tanpa diskriminasi, termasuk pengguna lanjut usia atau penyandang disabilitas.

Kemudahan penggunaan dapat dicapai melalui antarmuka yang sederhana, konsisten, dan mudah dipahami. Prinsip-prinsip desain seperti penempatan tombol yang logis, penggunaan ikon yang familiar, serta penyusunan menu yang rapi akan sangat membantu pengguna dalam bernavigasi dan menyelesaikan pekerjaannya. Selain itu, penyediaan umpan balik yang jelas seperti notifikasi kesalahan atau konfirmasi tindakan juga penting agar pengguna merasa yakin terhadap interaksi mereka dengan sistem.

Dengan memperhatikan aksesibilitas dan kemudahan penggunaan, sistem tidak hanya menjadi lebih inklusif, tetapi juga lebih efisien dan efektif dalam mendukung aktivitas organisasi. Sistem yang mudah digunakan akan mengurangi kebutuhan pelatihan, mempercepat adopsi, serta mengurangi risiko kesalahan operasional. Oleh karena itu, kedua aspek ini harus menjadi pertimbangan utama dalam setiap tahap

pengembangan sistem yang berfokus pada pengalaman dan keberhasilan pengguna.

#### **D. Evaluasi Pengalaman Pengguna**

Evaluasi pengalaman pengguna (user experience/UX evaluation) adalah proses untuk menilai sejauh mana sistem informasi memenuhi kebutuhan, harapan, dan kenyamanan pengguna dalam menggunakan sistem tersebut. Evaluasi ini mencakup aspek fungsionalitas, antarmuka, interaksi, serta kepuasan secara keseluruhan terhadap penggunaan sistem. Tujuan utamanya adalah mengidentifikasi kekuatan dan kelemahan dari perspektif pengguna, sehingga sistem dapat ditingkatkan agar lebih efektif dan menyenangkan saat digunakan.

Evaluasi dapat dilakukan melalui berbagai metode, seperti survei kepuasan, wawancara pengguna, observasi langsung saat pengguna berinteraksi dengan sistem, serta usability testing. Dalam usability testing, pengguna diminta untuk menyelesaikan tugas-tugas tertentu menggunakan sistem, sementara tim pengembang mengamati kesulitan atau kebingungan yang muncul. Data yang dikumpulkan kemudian dianalisis untuk menentukan aspek mana dari sistem yang perlu diperbaiki, baik dari sisi teknis maupun desain interaksi.

Penting juga untuk menggabungkan data kuantitatif dan kualitatif dalam evaluasi UX. Misalnya, waktu

penyelesaian tugas, jumlah klik, dan tingkat kesalahan pengguna bisa diukur secara objektif. Di sisi lain, persepsi pengguna tentang kenyamanan, kejelasan informasi, atau kepuasan emosional terhadap sistem merupakan data kualitatif yang tak kalah penting. Kombinasi kedua jenis data ini memberikan gambaran menyeluruh tentang pengalaman pengguna yang sebenarnya.

Melalui evaluasi pengalaman pengguna secara rutin, organisasi dapat terus meningkatkan kualitas sistem informasi seiring dengan perubahan kebutuhan dan ekspektasi pengguna. Evaluasi UX tidak hanya penting pada tahap akhir pengembangan, tetapi juga sebaiknya dilakukan secara iteratif sejak tahap prototipe hingga sistem berjalan penuh. Dengan begitu, sistem yang dikembangkan akan benar-benar berorientasi pada pengguna, meningkatkan produktivitas, dan mendukung tujuan strategis organisasi secara optimal.

**BAB**

**10**

# **Desain Antarmuka Pengguna (UI/UX)**

## A. Prinsip-prinsip Desain UI/UX

**D**esain UI/UX (User Interface/User Experience) yang baik berlandaskan pada prinsip-prinsip yang bertujuan menciptakan antarmuka yang tidak hanya menarik secara visual, tetapi juga mudah digunakan, intuitif, dan menyenangkan bagi pengguna (Arifin et al., 2022).

### 1. User-Centered Design

Prinsip user-centered design menekankan bahwa setiap rancangan antarmuka harus dibuat berdasarkan kebutuhan, perilaku, dan karakteristik pengguna akhir. Dalam konteks ini, perancang UI/UX dituntut untuk memahami siapa pengguna aplikasi, bagaimana cara mereka berinteraksi dengan sistem, serta apa tujuan yang ingin mereka capai. Riset pengguna, seperti wawancara, observasi, atau survei, menjadi langkah awal yang penting dalam merancang sistem yang benar-benar relevan dengan penggunanya.

Desain yang berpusat pada pengguna memungkinkan perancang untuk menghindari asumsi yang keliru dalam pengambilan keputusan desain. Sebuah sistem yang dirancang tanpa mempertimbangkan kebutuhan pengguna berisiko besar menyebabkan kebingungan, frustrasi, atau

bahkan kegagalan dalam penggunaannya. Oleh karena itu, pendekatan iteratif, seperti membuat prototipe awal dan mengujinya langsung pada pengguna, sangat dianjurkan untuk memperoleh umpan balik yang objektif.

Penerapan user-centered design tidak hanya membuat antarmuka menjadi lebih intuitif, tetapi juga meningkatkan kepuasan dan loyalitas pengguna. Dengan menempatkan pengguna sebagai fokus utama, pengembang menciptakan pengalaman yang tidak hanya fungsional tetapi juga menyenangkan. Prinsip ini juga selaras dengan tren pengembangan sistem modern yang semakin menekankan pada pengalaman pengguna sebagai faktor utama keberhasilan produk digital.

## 2. Konsistensi

Konsistensi merupakan prinsip fundamental dalam UI/UX yang menjamin pengalaman pengguna yang stabil dan dapat diprediksi. Elemen-elemen antarmuka seperti warna, ikon, jenis huruf, hingga posisi tombol harus memiliki pola penggunaan yang seragam di seluruh aplikasi atau website. Dengan demikian, pengguna tidak perlu belajar kembali setiap kali berpindah halaman atau menggunakan fitur baru

karena mereka telah memahami pola interaksi yang berlaku.

Sebagai contoh, penggunaan tombol berwarna hijau untuk aksi “submit” atau tombol merah untuk “hapus” membantu pengguna mengenali fungsi tombol secara instan, tanpa perlu membaca label secara detail. Jika desain berubah-ubah tanpa pola yang jelas, pengguna akan merasa kebingungan dan ragu-ragu dalam berinteraksi dengan sistem. Ketidakkonsistenan juga dapat menyebabkan penurunan kepercayaan terhadap profesionalisme produk tersebut.

Prinsip ini juga berlaku dalam hal bahasa dan istilah yang digunakan. Jika dalam satu bagian sistem digunakan istilah “beranda”, maka di bagian lain tidak seharusnya digunakan istilah lain seperti “halaman utama” untuk merujuk pada hal yang sama. Konsistensi yang baik menciptakan kejelasan, mengurangi beban kognitif pengguna, dan meningkatkan efisiensi penggunaan sistem secara keseluruhan.

### 3. Simplicity

Kesederhanaan atau simplicity dalam desain UI/UX bukan berarti membuat antarmuka menjadi minimalis secara ekstrem, melainkan menekankan

pada kejelasan dan efisiensi dalam penyampaian fungsi. Desain yang sederhana memungkinkan pengguna untuk fokus pada tugas utama tanpa terdistraksi oleh elemen-elemen yang tidak relevan. Dengan demikian, antarmuka menjadi lebih mudah digunakan, terutama bagi pengguna baru atau non-teknis.

Dalam praktiknya, prinsip kesederhanaan dapat diterapkan dengan mengurangi jumlah langkah untuk menyelesaikan sebuah aksi, menghindari teks atau ikon yang berlebihan, serta menata elemen visual secara rapi dan terstruktur. Elemen-elemen yang tidak penting sebaiknya dihapus atau disederhanakan agar tidak membingungkan pengguna. Tata letak yang bersih, pemilihan warna yang bijak, dan pemanfaatan ruang putih (*white space*) juga menjadi bagian dari desain yang sederhana namun efektif.

Desain yang terlalu kompleks cenderung membuat pengguna bingung, bahkan bisa menyebabkan mereka meninggalkan aplikasi sebelum menyelesaikan tugas. Oleh karena itu, prinsip *simplicity* sangat penting dalam menciptakan antarmuka yang intuitif dan efisien. Sistem yang sederhana tidak hanya menghemat waktu pengguna,

tetapi juga mempercepat proses adopsi teknologi baru.

#### 4. Feedback

Prinsip feedback dalam UI/UX merujuk pada kemampuan antarmuka untuk memberikan respon terhadap aksi yang dilakukan oleh pengguna. Sistem yang memberikan umpan balik secara langsung membantu pengguna memahami bahwa tindakan mereka telah diterima atau sedang diproses. Misalnya, perubahan warna tombol setelah diklik, munculnya notifikasi, atau efek visual saat mouse diarahkan ke ikon, semuanya berperan penting dalam menciptakan pengalaman interaktif yang meyakinkan.

Tanpa adanya feedback, pengguna mungkin merasa kebingungan apakah aksi mereka berhasil atau tidak. Hal ini dapat menyebabkan pengulangan tindakan secara tidak perlu, atau bahkan menyebabkan pengguna merasa frustrasi karena kurangnya kejelasan. Oleh karena itu, setiap interaksi harus disertai dengan tanggapan yang jelas dari sistem, baik berupa suara, gerakan, perubahan tampilan, maupun pesan teks yang informatif.

Feedback yang baik juga menciptakan rasa kontrol dan keterlibatan bagi pengguna. Ketika sistem merespons tindakan pengguna secara real-time,

pengguna merasa bahwa sistem bekerja sesuai harapan. Hal ini tidak hanya meningkatkan kepuasan pengguna, tetapi juga memperkuat kepercayaan terhadap kualitas dan kehandalan aplikasi atau sistem yang digunakan.

## 5. Aksesibilitas

Aksesibilitas adalah prinsip penting dalam UI/UX yang bertujuan memastikan bahwa sistem dapat digunakan oleh semua orang, termasuk mereka yang memiliki keterbatasan fisik atau sensorik. Desain antarmuka harus mempertimbangkan kebutuhan pengguna seperti penyandang disabilitas, termasuk tunanetra, buta warna, atau mereka yang memiliki gangguan motorik. Penerapan aksesibilitas bukan hanya tuntutan etika, tetapi juga bagian dari prinsip inklusivitas dalam teknologi.

Salah satu implementasi aksesibilitas adalah penggunaan warna yang memiliki kontras tinggi untuk memudahkan pengguna buta warna membedakan elemen visual. Penggunaan teks alternatif untuk gambar (alt text), navigasi berbasis keyboard, serta ukuran teks yang dapat diperbesar juga merupakan bagian dari desain yang aksesibel. Dengan memastikan bahwa elemen-elemen ini hadir,

pengguna dengan keterbatasan tetap dapat mengakses dan menggunakan sistem dengan nyaman.

Membangun aksesibilitas juga berarti memahami bahwa pengguna memiliki latar belakang, perangkat, dan kondisi yang berbeda-beda. Oleh karena itu, perancang UI/UX harus menguji antarmuka dalam berbagai skenario dan kondisi, serta mengacu pada standar internasional seperti Web Content Accessibility Guidelines (WCAG). Sistem yang aksesibel tidak hanya meningkatkan jangkauan pengguna, tetapi juga mencerminkan komitmen terhadap keadilan dan keberlanjutan dalam pengembangan teknologi digital.

## **B. Proses Desain UI/UX**

Proses desain UI/UX (User Interface/User Experience) merupakan serangkaian langkah sistematis untuk menciptakan antarmuka yang tidak hanya estetik, tetapi juga fungsional dan mudah digunakan. Tahap pertama dari proses ini adalah riset pengguna (user research), di mana tim desain mengumpulkan data tentang kebutuhan, kebiasaan, dan masalah yang dihadapi oleh target pengguna. Riset ini dilakukan melalui wawancara, survei, observasi, dan studi perilaku, yang

bertujuan untuk memahami konteks penggunaan sistem secara mendalam dan membangun persona pengguna.

Langkah berikutnya adalah perencanaan dan perancangan pengalaman pengguna (UX design). Pada tahap ini, hasil riset diterjemahkan ke dalam struktur dan alur interaksi sistem, seperti pembuatan user journey, site map, dan wireframe. Wireframe adalah sketsa kasar yang menggambarkan tata letak elemen-elemen pada halaman sistem, tanpa desain visual. Tujuannya adalah merancang alur logis dan efisien dalam menyelesaikan tugas pengguna, sebelum memperhatikan aspek estetika. Fokus utama UX adalah kemudahan, kenyamanan, dan kejelasan dalam setiap interaksi pengguna dengan sistem.

Setelah struktur dan alur disetujui, proses berlanjut ke tahap desain antarmuka pengguna (UI design), yang mencakup pemilihan warna, tipografi, ikon, dan elemen visual lainnya. Desain UI bertujuan memperkuat UX melalui tampilan yang menarik dan intuitif. Pada tahap ini, elemen visual harus mencerminkan identitas merek dan mematuhi prinsip estetika serta aksesibilitas. UI dan UX bekerja secara sinergis: UI memperindah dan mendukung UX, sedangkan UX memastikan UI dapat digunakan secara optimal.

Tahap selanjutnya adalah pengujian prototipe (usability testing), di mana rancangan diuji oleh pengguna

sesungguhnya untuk melihat seberapa efektif, efisien, dan memuaskan antarmuka yang telah dibuat. Umpan balik dari pengguna digunakan untuk melakukan iterasi perbaikan terhadap desain. Prototipe bisa bersifat statis (gambar interaktif) atau dinamis (klik-dapat dan fungsional secara dasar). Tahap ini penting untuk memastikan bahwa rancangan benar-benar sesuai dengan kebutuhan pengguna sebelum masuk ke tahap pengembangan teknis.

Proses desain UI/UX bersifat iteratif, artinya dapat dilakukan berulang sesuai hasil evaluasi dan masukan pengguna. Setelah sistem diluncurkan, proses desain tidak berhenti, karena pemantauan dan perbaikan terus-menerus diperlukan berdasarkan data penggunaan nyata. Dengan pendekatan proses yang terstruktur dan berorientasi pengguna, UI/UX tidak hanya meningkatkan kepuasan pengguna, tetapi juga mendukung keberhasilan sistem secara keseluruhan dalam mencapai tujuan bisnis dan operasionalnya.

### **C. Tools Desain UI/UX**

Tools desain UI/UX adalah perangkat lunak yang digunakan untuk membantu para desainer menciptakan antarmuka pengguna yang intuitif dan pengalaman pengguna yang optimal. Alat-alat ini mendukung seluruh

proses desain, mulai dari pembuatan wireframe, prototipe interaktif, hingga kolaborasi tim dan pengujian usability. Dengan tools yang tepat, proses desain dapat dilakukan lebih efisien, fleksibel, dan terstruktur, sehingga menghasilkan output yang lebih profesional dan sesuai kebutuhan pengguna (Santi, 2020).

Salah satu tools paling populer dalam desain UI/UX adalah Figma, sebuah aplikasi berbasis web yang memungkinkan kolaborasi real-time antar tim desain. Figma menyediakan fitur lengkap untuk membuat wireframe, desain antarmuka, dan prototipe interaktif dalam satu platform. Karena berbasis cloud, Figma sangat cocok untuk tim yang bekerja jarak jauh, memungkinkan semua anggota proyek untuk memberikan komentar dan revisi secara langsung tanpa perlu pertukaran file yang rumit.

Selain Figma, Adobe XD juga merupakan tools yang banyak digunakan untuk desain dan prototyping. Adobe XD menawarkan integrasi yang kuat dengan ekosistem Adobe lainnya seperti Photoshop dan Illustrator, serta mendukung animasi dan transisi antar layar yang halus. Tools ini cocok untuk desainer yang menginginkan tampilan visual tinggi dan workflow yang sinkron dengan aplikasi desain grafis lainnya. Untuk pengguna macOS,

Sketch juga menjadi pilihan populer karena kesederhanaannya dan dukungan luas terhadap plugin.

Untuk tahap wireframing awal, tools seperti Balsamiq sangat bermanfaat. Balsamiq memungkinkan pembuatan sketsa kasar dengan tampilan yang menyerupai gambar tangan, memudahkan desainer fokus pada struktur dan alur tanpa terganggu oleh elemen visual. Di sisi lain, untuk keperluan pengujian dan validasi UX, tools seperti Maze dan UsabilityHub membantu mengumpulkan data pengguna melalui simulasi dan survei interaktif. Ini memberikan insight berdasarkan perilaku nyata pengguna sebelum sistem dikembangkan.

Pemilihan tools desain UI/UX harus disesuaikan dengan kebutuhan proyek, kompleksitas sistem, dan kebiasaan kerja tim. Tidak ada satu tools yang sempurna untuk semua kebutuhan, namun kombinasi yang tepat dapat mendukung kolaborasi, efisiensi, dan kualitas hasil desain secara signifikan. Tools yang mendukung iterasi cepat, pengujian user-centric, dan komunikasi visual yang jelas akan sangat membantu dalam menciptakan sistem yang tidak hanya fungsional tetapi juga memberikan pengalaman pengguna yang unggul.

## D. Perancangan UX

Perancangan UX adalah proses perancangan produk (dalam hal ini, produk digital) agar dapat digunakan dengan baik, mudah dan memberikan pengalaman yang menyenangkan ketika digunakan. Salah satu metode pengembangan UX adalah menggunakan Garret's Element.

Garret's element memiliki 5 layer utama yaitu Strategy, Scope, Structure, Skeleton dan Surface. Layer pertama memiliki konsep yang abstract, sedangkan layer terakhir memiliki konsep yang telah konkrit.

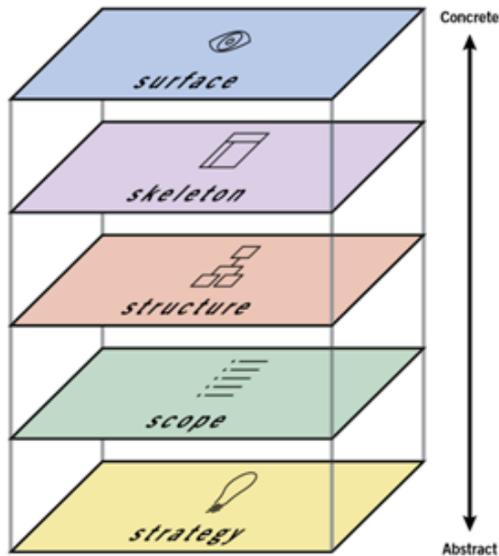


Figure 1 Garret's element layer.

## 1. Strategy:

Pada layer paling dasar, perancangan UX melihat sebuah tampilan antarmuka harus menggambarkan fungsionalitas dan juga informasi. Output dari layer ini adalah abstraksi yang menjadi dasar mengapa antarmuka tersebut dirancang. Layer strategy mencakup apa kebutuhan pengguna (*user needs*) dan apa tujuan produk ini digunakan oleh pengguna (*product objectives*).

- *User needs:*

Kebutuhan pengguna dapat diidentifikasi melalui segmentasi calon pengguna, riset pengguna dan membuat *user persona*.

Segmentasi calon pengguna dilakukan dengan mengelompokkan para pengguna berdasarkan karakteristik demografi, seperti jenis kelamin, usia, jenjang pendidikan, status pernikahan, pendapatan dan lain-lain.

Riset pengguna adalah Upaya untuk mengumpulkan informasi agar perancangan antar muka benar-benar sesuai kebutuhan pengguna. Beberapa metode yang dilakukan antara lain : riset pasar (market research), inkuiri kontekstual

(*contextual inquiry*) , dan pengujian pengguna (user testing).

*User persona* adalah karakter fiksi yang digunakan untuk sebagai model yang semirip mungkin dengan pengguna. Persona dibangun berdasarkan data dan riset yang sudah dilakukan yang menggambarkan calon pengguna.



Figure 2 contoh user persona

- *Product objectives*

Dalam perancangan antarmuka yang sesuai, perlu sentuhan aspek manajemen untuk menentukan tujuan bisnis (*business goals*),

identitas produk (*brand identity*) dan metrics pengukuran (*success metrics*).

## 2. Scope:

Lapisan selanjutnya adalah Scope Plane. Layer ini merupakan lanjutan abstraksi dari layer strategy namun didefinisikan lebih spesifik kedalam 2 bentuk, yaitu functional specification dan content requirement.

- Functional specification berbicara tentang apa saja fitur yang harus ada pada sistem. Kehadiran fitur-fitur yang dibutuhkan akan mempengaruhi bagaimana antarmuka sistem dirancang. Umumnya, functional specification tercantum pada dokumentasi pengembangan sistem sebagai kebutuhan pengguna (user requirement) . Pendefinisian kebutuhan fungsional ini harus dituliskan secara positif, spesifik dan tidak multitafsir. Alih-alih menuliskan “Aplikasi tidak dapat dihentikan otomatis oleh sistem tanpa menyimpan perubahan yang telah dilakukan.” akan lebih sesuai jika ditulis seperti, “Sistem akan menampilkan notifikasi untuk menyimpan perubahan sebelum aplikasi ditutup”.
- Content requirement mencakup elemen yang menjadi konten dalam antarmuka yang mendukung kebutuhan fungsional. Contoh elemen content requirement seperti gambar, video, peta, ilustrasi, icon, tombol dan lain-lain. Penggunaan konten harus sesuai dengan tujuan dan ukuran

yang pas. Contohnya, pada website e-commerce, konten gambar produk harus memiliki ukuran yang cukup besar untuk dilihat oleh pengguna. Sementara tombol “beli”, harus lebih mencolok dibanding tombol lain agar menstimulasi pengguna untuk melakukan pembelian barang.

### 3. Structure:

Layer structure menjadi bentuk konkrit dari 2 layer sebelumnya karena akan mendefinisikan bagaimana pengguna bisa sampai ke tampilan halaman tertentu dan apa yang akan mereka lakukan jika telah sampai pada halaman tertentu. Pada layer ini, terdapat 2 fokus utama yaitu perancangan interaksi (interaction design) dan arsitektur informasi (information architecture).

- *Interaction Design* menggambarkan bagaimana kebiasaan pengguna menggunakan sistem dan merancang bagaimana sistem mengakomodir dan merespon kebiasaan pengguna tersebut. Contohnya, pada aplikasi e-commerce, saat pengguna telah memasukkan barang ke dalam keranjang belanja, secara otomatis mereka biasanya akan mencari tombol bayar dan ingin melihat promo yang tersedia. Umumnya, untuk merancang interaksi pengguna, kita akan menggambarkan task flow/user flow menggunakan diagram seperti activity diagram atau flowchart. Perancangan interaksi ini juga

harus mencakup penanganan kesalahan (*error handling*) yang memungkinkan pengguna ataupun sistem melakukan kesalahan selama beroperasi. Contoh *error handling* yang paling sering digunakan adalah menyediakan laman **404 not found** pada website.

- *Information Architecture* mencakup pembuatan struktur konten sehingga dapat menampilkan skema navigasi yang mungkin akan dilalui oleh pengguna.

#### 4. Skeleton Plane:

Pada layer keempat ini sudah masuk tahap perancangan sistem dalam bentuk low fidelity prototype atau purwarupa sederhana (*wireframe*). Skeleton Plane mencakup bagaimana antarmuka, navigasi dan informasi mulai dirancang kedalam sebuah *wireframe*. *Wireframe* merupakan level paling dasar dalam pengembangan purwarupa. *Wireframe* tidak dapat di klik dan hanya digambarkan dalam 2 dimensi. *Wireframe* hanya menampilkan jalur navigasi untuk pengguna, misalnya pada halaman awal pengguna menekan tombol login, maka pada *wireframe* selanjutnya menggambarkan halaman beranda.

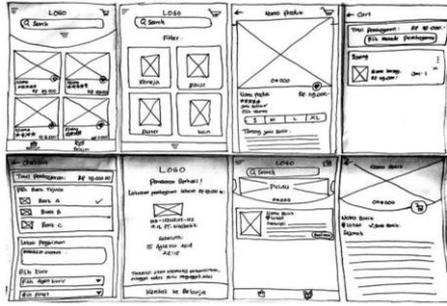


Figure 3 contoh wireframe

## 5. Surface Plane:

Layer paling atas adalah surface plane yang merupakan layer paling konkrit karena mencakup perancangan sensorik pengguna, seperti apa bentuk, warna, bahkan suara yang pengguna lihat. Pada layer ini, permainan warna dan penggunaan teks seperti tipografi sangat mempengaruhi pengguna. Warna-warna standar yang sering digunakan khususnya pada website sudah menjadi ketentuan yang sebaiknya tidak kita ubah, cukup sesuaikan dengan prinsip warna produk kita.

## Button Styles



Figure 4 standar warna pada tombol

Oleh karena itu, output dari tahap ini adalah *high fidelity prototype* yang mana sudah dapat diujicobakan dan berinteraksi dengan pengguna.

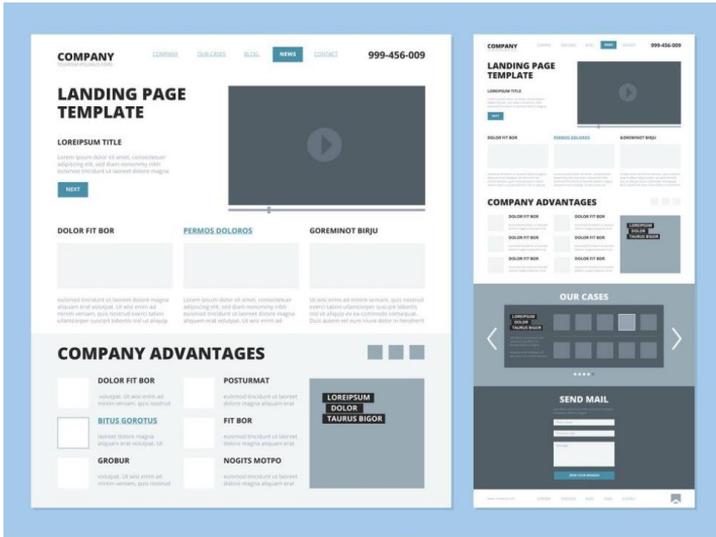


Figure 5 contoh purwarupa

## E. Evaluasi dan Iterasi Desain

Evaluasi dan iterasi desain adalah proses penting dalam pengembangan UI/UX yang bertujuan memastikan bahwa antarmuka dan pengalaman pengguna yang dirancang benar-benar efektif, fungsional, dan sesuai dengan kebutuhan pengguna. Evaluasi dilakukan untuk menilai kualitas desain berdasarkan umpan balik pengguna, pengujian fungsional, serta kesesuaian dengan prinsip desain yang telah ditetapkan. Proses ini memungkinkan tim desain untuk menemukan

kekurangan atau hambatan dalam interaksi pengguna sebelum sistem benar-benar diterapkan secara penuh.

Evaluasi dapat dilakukan melalui beberapa metode, seperti usability testing, heuristic evaluation, atau A/B testing. Dalam usability testing, pengguna diminta menyelesaikan serangkaian tugas menggunakan prototipe desain, sementara pengamat mencatat kesulitan atau kesalahan yang muncul. Heuristic evaluation melibatkan para ahli UX yang menilai antarmuka berdasarkan prinsip-prinsip kegunaan yang umum, sedangkan A/B testing membandingkan dua versi desain untuk melihat mana yang lebih efektif berdasarkan metrik tertentu seperti konversi atau waktu penyelesaian tugas.

Hasil evaluasi digunakan sebagai dasar untuk melakukan iterasi, yaitu proses memperbaiki dan menyempurnakan desain berdasarkan temuan dari evaluasi sebelumnya. Iterasi merupakan inti dari pendekatan design thinking, yang menekankan pentingnya pengembangan desain secara bertahap dan responsif terhadap masukan pengguna. Alih-alih membuat desain final dalam satu langkah besar, iterasi memungkinkan desain berkembang melalui siklus perbaikan yang berkelanjutan.

Setiap siklus iterasi melibatkan pembuatan prototipe baru, pengujian ulang, dan perbaikan berdasarkan hasil

pengujian tersebut. Proses ini berlangsung berulang-ulang hingga sistem mencapai tingkat kegunaan dan kepuasan pengguna yang optimal. Pendekatan iteratif ini terbukti lebih efektif dalam mengakomodasi perubahan kebutuhan, mencegah kesalahan desain yang mahal di tahap akhir, dan menghasilkan solusi yang benar-benar berorientasi pada pengguna.

Dengan menerapkan evaluasi dan iterasi secara konsisten selama proses desain, sistem informasi dapat dikembangkan secara lebih adaptif, responsif, dan berkualitas tinggi. Desain tidak lagi dipandang sebagai hasil akhir, melainkan sebagai proses berkelanjutan yang selalu bisa diperbaiki. Hasil akhirnya adalah produk yang tidak hanya memenuhi spesifikasi teknis, tetapi juga menyenangkan, mudah digunakan, dan relevan dengan kebutuhan nyata penggunanya.

**BAB**

**11**

# **Integrasi Aplikasi Sistem Informasi**

## A. Konsep dan Tujuan Integrasi Sistem

**I**ntegrasi sistem adalah proses menggabungkan berbagai komponen atau subsistem teknologi informasi yang terpisah menjadi satu kesatuan yang saling berinteraksi dan bekerja secara terpadu. Dalam konteks organisasi, integrasi sistem memungkinkan aplikasi, database, perangkat keras, dan proses bisnis yang sebelumnya berjalan secara terpisah, untuk saling bertukar data dan informasi secara otomatis. Dengan integrasi, sistem menjadi lebih efisien karena mengurangi redundansi, mempercepat alur informasi, dan meningkatkan konsistensi data (Rahmawati & Bachtiar, 2018).

Salah satu konsep utama dalam integrasi sistem adalah interoperabilitas, yaitu kemampuan berbagai sistem dan perangkat lunak untuk saling memahami dan menggunakan informasi secara efektif. Ini sering dicapai melalui penggunaan standar komunikasi, seperti API (Application Programming Interface), XML, JSON, atau protokol seperti SOAP dan REST. Interoperabilitas memungkinkan sistem yang dibangun dengan teknologi berbeda tetap dapat berinteraksi dan berbagi data tanpa perlu dilakukan konversi secara manual.

Tujuan utama dari integrasi sistem adalah untuk meningkatkan efisiensi operasional dan pengambilan

keputusan dalam organisasi. Dengan sistem yang terintegrasi, data dapat mengalir secara otomatis antar departemen atau fungsi, seperti keuangan, SDM, logistik, dan pemasaran, tanpa perlu input ulang. Hal ini tidak hanya menghemat waktu dan biaya, tetapi juga mengurangi risiko kesalahan akibat duplikasi atau ketidaksesuaian data.

Integrasi sistem juga bertujuan untuk memberikan pengalaman pengguna yang lebih baik. Misalnya, dalam layanan pelanggan, integrasi memungkinkan staf melihat riwayat pembelian, status pengiriman, dan data layanan dari satu antarmuka terpadu. Pengguna tidak perlu berpindah-pindah aplikasi atau memasukkan data yang sama berulang kali, yang pada akhirnya meningkatkan kepuasan dan kecepatan layanan.

Dalam skala yang lebih besar, integrasi sistem mendukung strategi transformasi digital organisasi. Sistem yang terintegrasi memberikan pondasi kuat untuk otomatisasi proses bisnis, analitik data real-time, serta penerapan teknologi baru seperti AI dan IoT. Organisasi dapat bergerak lebih cepat dalam menanggapi perubahan pasar, melakukan inovasi, dan meningkatkan daya saing secara keseluruhan.

Selain itu, integrasi juga mendukung keamanan dan pengendalian data yang lebih baik. Dengan sistem yang

terpusat dan terintegrasi, organisasi dapat lebih mudah menerapkan kebijakan akses, enkripsi, serta audit terhadap penggunaan data. Ini sangat penting dalam menghadapi regulasi seperti GDPR atau perlindungan data pribadi di berbagai yurisdiksi. Data yang konsisten dan dapat dikendalikan menjadi aset berharga bagi organisasi di era digital.

Namun, penting juga untuk dicatat bahwa proses integrasi sistem bukan tanpa tantangan. Perbedaan platform teknologi, kualitas data yang rendah, serta resistensi dari pengguna atau departemen tertentu dapat menjadi penghambat. Oleh karena itu, integrasi harus dirancang dengan pendekatan strategis, melibatkan semua pemangku kepentingan, dan disertai perencanaan yang matang terkait arsitektur sistem dan pengelolaan perubahan.

Konsep dan tujuan integrasi sistem bukan hanya tentang aspek teknis, tetapi juga menyangkut peningkatan efektivitas organisasi secara menyeluruh. Integrasi membantu menciptakan sistem yang lebih cerdas, adaptif, dan terkoordinasi, yang pada akhirnya memperkuat kemampuan organisasi untuk beroperasi secara lebih efisien, responsif, dan berorientasi pada pengguna serta data.

## **B. Teknologi dan Standar Integrasi**

Dalam dunia pengembangan sistem informasi, teknologi dan standar integrasi memainkan peranan penting agar berbagai sistem yang berbeda dapat saling berkomunikasi dan bertukar data secara efektif. Teknologi integrasi merujuk pada perangkat lunak dan protokol yang memungkinkan penyatuan berbagai aplikasi dan layanan dalam sebuah ekosistem yang terpadu. Standar integrasi adalah aturan atau spesifikasi yang digunakan untuk memastikan interoperabilitas dan konsistensi data antara sistem yang berbeda.

Salah satu teknologi paling umum digunakan dalam integrasi sistem adalah API (Application Programming Interface). API memungkinkan aplikasi yang berbeda untuk saling berinteraksi dengan menyediakan antarmuka yang terstandarisasi untuk pertukaran data. Ada berbagai jenis API, seperti RESTful API yang menggunakan protokol HTTP dan format data JSON atau XML, serta SOAP yang lebih formal dan berbasis XML. API memudahkan pengembang untuk mengakses fungsi sistem lain tanpa perlu mengetahui detail internalnya.

Selain API, teknologi middleware juga sangat penting dalam integrasi sistem. Middleware berfungsi sebagai perantara yang menghubungkan aplikasi dan layanan yang berbeda, mengelola komunikasi dan data secara

otomatis. Contohnya adalah Enterprise Service Bus (ESB), yang menyediakan platform terpusat untuk mengatur pertukaran pesan, orkestrasi layanan, dan transformasi data antara berbagai sistem. Middleware membantu menyederhanakan kompleksitas integrasi terutama pada lingkungan IT yang heterogen.

Standar komunikasi data juga menjadi aspek kunci dalam integrasi. Format seperti XML (Extensible Markup Language) dan JSON (JavaScript Object Notation) digunakan secara luas untuk pertukaran data karena sifatnya yang fleksibel dan mudah dibaca oleh mesin maupun manusia. Selain itu, protokol komunikasi seperti HTTP, HTTPS, FTP, dan MQTT digunakan sebagai saluran pengiriman data antar sistem dengan keamanan dan kecepatan yang diatur sesuai kebutuhan.

Teknologi integrasi juga melibatkan konsep service-oriented architecture (SOA), yaitu pendekatan desain sistem yang berbasis pada penyediaan layanan mandiri yang dapat dipanggil dan digunakan oleh aplikasi lain secara dinamis. SOA memungkinkan fleksibilitas tinggi dalam pengembangan dan integrasi karena layanan dapat diubah, diganti, atau diperbarui tanpa mempengaruhi sistem secara keseluruhan. Pendekatan ini menjadi landasan bagi pengembangan aplikasi berbasis mikroservis yang kini semakin populer.

Selain teknologi, standar integrasi seperti OpenAPI Specification membantu dalam mendokumentasikan dan menyusun API secara konsisten, sehingga memudahkan pengembang dalam mengimplementasikan dan menggunakan API. Standar lain seperti OAuth untuk otorisasi dan keamanan akses API juga penting untuk menjaga integritas dan privasi data yang dikirim antar sistem. Penggunaan standar ini memastikan integrasi yang tidak hanya efektif tapi juga aman.

Dalam dunia cloud computing, teknologi integrasi semakin berkembang dengan hadirnya platform integrasi sebagai layanan (Integration Platform as a Service - iPaaS). iPaaS menyediakan lingkungan berbasis cloud untuk membangun, mengelola, dan mengoperasikan integrasi antara aplikasi on-premise dan cloud secara cepat dan mudah. Contoh iPaaS populer termasuk Microsoft Power Automate, MuleSoft, dan Zapier, yang mempermudah organisasi menghubungkan berbagai sistem tanpa perlu investasi besar dalam infrastruktur.

Teknologi dan standar integrasi adalah fondasi utama dalam menciptakan sistem yang terpadu dan efisien. Pemilihan teknologi dan standar yang tepat harus disesuaikan dengan kebutuhan bisnis, kompleksitas sistem, dan kemampuan organisasi. Dengan implementasi yang benar, integrasi sistem tidak hanya meningkatkan

alur kerja dan kolaborasi antar departemen, tetapi juga membuka peluang untuk inovasi dan transformasi digital yang berkelanjutan.

### **C. Arsitektur Integrasi**

Arsitektur integrasi adalah kerangka kerja atau blueprint yang menggambarkan bagaimana berbagai sistem, aplikasi, dan data akan saling terhubung dan berinteraksi dalam sebuah lingkungan TI. Arsitektur ini sangat penting karena memberikan panduan dalam perancangan solusi integrasi yang efektif, skalabel, dan mudah dikelola. Dengan arsitektur yang baik, organisasi dapat memastikan bahwa berbagai komponen teknologi dapat berfungsi bersama-sama secara harmonis untuk mendukung tujuan bisnis.

Salah satu model arsitektur integrasi yang umum digunakan adalah point-to-point integration, di mana setiap sistem dihubungkan langsung satu sama lain. Meskipun pendekatan ini mudah dan cepat diterapkan pada skala kecil, model ini menjadi sulit diatur ketika jumlah sistem bertambah karena jumlah koneksi akan meningkat secara eksponensial. Model ini rentan terhadap masalah pemeliharaan dan skalabilitas karena perubahan pada satu sistem dapat memengaruhi koneksi yang lain.

Untuk mengatasi kelemahan model point-to-point, Enterprise Service Bus (ESB) diperkenalkan sebagai arsitektur integrasi yang lebih terpusat dan terstruktur. ESB bertindak sebagai mediator yang menghubungkan berbagai sistem melalui bus layanan, mengatur komunikasi, transformasi data, dan orkestrasi layanan. Dengan ESB, pengelolaan integrasi menjadi lebih sederhana dan fleksibel, memungkinkan penambahan atau penggantian sistem tanpa mengganggu keseluruhan arsitektur.

Selain itu, arsitektur integrasi modern juga banyak mengadopsi pendekatan *microservices architecture*. Dalam model ini, aplikasi dibangun sebagai kumpulan layanan kecil yang mandiri dan dapat diintegrasikan melalui API. Setiap layanan bertanggung jawab atas fungsi bisnis tertentu dan dapat dikembangkan, diuji, serta di-deploy secara independen. *Microservices* memungkinkan skala dan fleksibilitas yang tinggi, serta memudahkan integrasi dengan sistem lain melalui protokol standar.

Arsitektur integrasi berbasis API Gateway juga semakin populer, terutama dalam lingkungan cloud dan aplikasi mobile. API Gateway bertindak sebagai pintu gerbang tunggal untuk semua permintaan API, mengelola autentikasi, otorisasi, pemantauan, dan pembatasan lalu lintas. Dengan menggunakan API Gateway, organisasi dapat lebih mudah mengontrol akses dan

mengoptimalkan performa sistem terintegrasi secara keseluruhan.

Selain model dan teknologi tersebut, arsitektur integrasi harus memperhatikan aspek security architecture. Hal ini meliputi pengaturan otentikasi dan otorisasi, enkripsi data saat transit maupun saat penyimpanan, serta audit dan monitoring. Keamanan integrasi sangat penting untuk melindungi data sensitif dan menjaga kepercayaan pengguna terhadap sistem yang terhubung.

Perancangan arsitektur integrasi juga harus memperhitungkan aspek scalability dan performance. Sistem integrasi harus mampu menangani pertumbuhan volume data dan jumlah pengguna tanpa mengalami penurunan performa. Oleh karena itu, arsitektur yang modular dan berbasis layanan seperti microservices dan ESB lebih disukai karena mereka menawarkan skalabilitas horizontal dan distribusi beban kerja.

Kesimpulannya, arsitektur integrasi adalah pondasi penting yang menentukan keberhasilan implementasi integrasi sistem dalam organisasi. Pemilihan model arsitektur harus disesuaikan dengan kebutuhan bisnis, kompleksitas sistem, dan sumber daya yang tersedia. Dengan arsitektur yang tepat, integrasi sistem dapat berjalan lancar, mendukung kolaborasi antar departemen, dan mempercepat proses transformasi digital secara efektif.

**BAB**

**12**

**Studi Kasus:  
Perancangan Sistem  
di Bisnis**

## A. Identifikasi Kebutuhan Bisnis

Sebuah perusahaan ritel bernama Toko Mandiri yang bergerak di bidang penjualan barang kebutuhan rumah tangga mengalami permasalahan dalam pencatatan stok barang, transaksi penjualan, dan pelaporan keuangan. Proses bisnis yang masih manual menyebabkan keterlambatan dalam pemantauan stok, kesalahan pencatatan transaksi, dan ketidaksesuaian laporan keuangan. Manajemen menyadari bahwa mereka membutuhkan sebuah sistem informasi yang dapat mengintegrasikan seluruh proses bisnis secara real-time dan efisien.

Tim manajemen memutuskan untuk melakukan evaluasi kebutuhan sistem informasi berdasarkan proses kerja harian yang berjalan. Mereka mengidentifikasi bahwa sistem harus mampu mengelola inventori secara otomatis, mencatat seluruh transaksi secara digital, dan menghasilkan laporan keuangan yang akurat setiap saat. Selain itu, sistem juga harus memiliki fitur manajemen pelanggan dan kemampuan untuk memberikan laporan penjualan berdasarkan waktu, produk, dan lokasi cabang.

Kebutuhan juga muncul dari sisi operasional, di mana pegawai toko mengeluhkan lamanya waktu yang dibutuhkan untuk mencatat stok manual dan seringnya terjadi kesalahan saat penghitungan barang. Oleh karena

itu, diperlukan sistem yang memiliki antarmuka pengguna yang mudah dipahami dan dapat diakses melalui perangkat komputer maupun mobile. Manajemen juga membutuhkan sistem yang mampu memberikan notifikasi jika terjadi kekurangan stok.

Dalam mengidentifikasi kebutuhan tersebut, pihak manajemen juga melibatkan perwakilan dari bagian gudang, kasir, keuangan, dan IT. Melalui diskusi dan survei internal, mereka mengumpulkan berbagai masukan tentang kebutuhan fitur, alur kerja yang efisien, dan jenis laporan yang dibutuhkan oleh masing-masing divisi. Pendekatan partisipatif ini dilakukan untuk memastikan bahwa sistem yang akan dirancang benar-benar menjawab kebutuhan nyata di lapangan.

Hasil dari tahap identifikasi kebutuhan bisnis ini menghasilkan sebuah dokumen kebutuhan sistem (system requirement specification) yang mencakup kebutuhan fungsional dan non-fungsional. Dokumen ini kemudian menjadi acuan utama dalam tahap selanjutnya yaitu analisis dan perancangan sistem. Kejelasan kebutuhan bisnis menjadi fondasi penting untuk menghindari kesalahan dalam pengembangan sistem dan memastikan implementasi yang tepat sasaran (Azis, 2022).

## **B. Analisis dan Perancangan Sistem**

Setelah kebutuhan bisnis berhasil diidentifikasi secara menyeluruh, tahap selanjutnya adalah melakukan analisis sistem yang ada serta merancang solusi sistem informasi yang tepat. Tim pengembang melakukan analisis SWOT terhadap sistem manual yang sedang berjalan, mengidentifikasi kelemahan dan peluang perbaikan. Dari hasil analisis tersebut, disusun flowchart dan diagram alur kerja yang menggambarkan proses bisnis saat ini dan rancangan sistem baru yang diusulkan.

Perancangan sistem dimulai dengan pembuatan Entity Relationship Diagram (ERD) untuk merancang basis data yang mampu menampung data pelanggan, stok barang, transaksi, dan laporan. Diagram konteks dan Data Flow Diagram (DFD) juga dibuat untuk menggambarkan hubungan antar bagian sistem dan aliran data. Dalam rancangan ini, sistem terdiri dari beberapa modul utama seperti manajemen barang, transaksi penjualan, laporan penjualan, laporan keuangan, dan notifikasi stok.

Selain aspek teknis, desain antarmuka pengguna (UI/UX) juga menjadi perhatian utama. Sistem dirancang menggunakan tampilan yang sederhana dan mudah digunakan oleh karyawan toko yang belum terbiasa dengan teknologi. Beberapa fitur seperti auto-fill, drop-

down menu, dan laporan visual seperti grafik penjualan dimasukkan dalam desain. Rancangan ini dibuat menggunakan perangkat bantu seperti Figma dan dibahas bersama stakeholder untuk mendapatkan umpan balik awal.

Keamanan dan backup data juga dirancang dalam tahap ini. Sistem akan dibangun berbasis cloud untuk memudahkan akses dari berbagai cabang toko serta menjamin keamanan data dari kehilangan atau kerusakan. Sistem juga dilengkapi fitur login user dengan otorisasi berdasarkan jabatan, sehingga tidak semua data dapat diakses oleh semua pengguna.

Dokumen desain sistem yang dihasilkan menjadi acuan bagi tim pengembang dalam melakukan coding dan implementasi sistem. Dengan dokumentasi yang lengkap dan desain yang terstruktur, proses pengembangan dapat dilakukan lebih cepat, terarah, dan minim revisi. Kolaborasi antara tim IT dan pengguna akhir selama proses perancangan sangat membantu menghasilkan sistem yang sesuai kebutuhan nyata di lapangan.

### **C. Hasil Implementasi Sistem**

Setelah proses pengembangan selesai, sistem mulai diimplementasikan secara bertahap di lingkungan kerja Toko Mandiri. Tahap pertama dilakukan pada satu cabang

sebagai pilot project. Sistem diuji coba secara langsung oleh staf toko, termasuk kasir, petugas gudang, dan manajer cabang. Hasil awal menunjukkan peningkatan efisiensi dalam pencatatan transaksi dan pelacakan stok barang.

Dalam waktu dua minggu, transaksi penjualan menjadi lebih cepat karena sistem telah terintegrasi dengan barcode scanner, dan laporan penjualan bisa langsung ditarik dalam bentuk grafik harian. Pencatatan stok juga menjadi lebih akurat karena setiap barang keluar dan masuk otomatis tercatat dalam sistem. Karyawan merasa lebih mudah bekerja karena tidak perlu lagi melakukan pencatatan manual yang berisiko salah tulis.

Laporan keuangan yang sebelumnya dibuat secara manual kini dapat dihasilkan secara otomatis berdasarkan data penjualan dan pengeluaran yang tercatat dalam sistem. Hal ini mempercepat proses pelaporan bulanan ke kantor pusat dan mengurangi beban kerja staf administrasi. Selain itu, sistem memberikan notifikasi otomatis ke gudang jika stok barang mencapai batas minimum, sehingga mempercepat proses pengadaan ulang.

Implementasi tahap pertama juga menunjukkan beberapa tantangan, seperti kebutuhan pelatihan

tambahan untuk staf yang belum terbiasa dengan penggunaan komputer, serta gangguan teknis kecil seperti koneksi internet lambat. Namun, kendala ini segera diatasi dengan pelatihan tambahan dan penyediaan koneksi internet yang lebih stabil di toko.

Setelah evaluasi positif dari cabang pertama, sistem kemudian diimplementasikan di seluruh cabang Toko Mandiri dalam waktu dua bulan. Proses migrasi data dilakukan dengan hati-hati untuk memastikan data dari sistem lama tidak hilang. Hasil akhirnya menunjukkan peningkatan produktivitas, kepuasan pelanggan yang lebih tinggi, serta penghematan waktu dan biaya operasional yang signifikan.

#### **D. Pembelajaran dan Rekomendasi**

Dari studi kasus ini, terdapat beberapa pembelajaran penting yang dapat diambil dalam perancangan dan implementasi sistem informasi bisnis. Pertama, keterlibatan pengguna sejak awal proses sangat krusial untuk menghasilkan sistem yang sesuai dengan kebutuhan nyata. Komunikasi yang baik antara tim pengembang dan stakeholder memastikan tidak ada fitur penting yang terlewat.

Kedua, perancangan sistem yang matang, termasuk aspek UI/UX dan keamanan data, menjadi kunci keberhasilan implementasi. Sistem yang terlalu kompleks

atau tidak ramah pengguna akan menyulitkan adopsi oleh karyawan. Oleh karena itu, penting untuk mengutamakan desain yang intuitif dan sesuai dengan kapabilitas pengguna akhir.

Ketiga, pelatihan dan pendampingan selama proses implementasi harus direncanakan dengan baik. Karyawan yang merasa didukung selama masa transisi lebih cepat beradaptasi dengan sistem baru. Hal ini akan mempercepat pencapaian tujuan efisiensi dan produktivitas yang diharapkan dari sistem yang diterapkan.

Keempat, penerapan sistem berbasis cloud memberikan keunggulan dari sisi fleksibilitas, keamanan, dan skalabilitas. Namun, keberhasilannya sangat tergantung pada infrastruktur pendukung seperti koneksi internet yang stabil dan perangkat keras yang memadai. Oleh karena itu, aspek teknis ini perlu diperhatikan sejak awal.

Sebagai rekomendasi, setiap bisnis yang ingin mengembangkan sistem informasi perlu melakukan analisis kebutuhan yang mendalam dan perancangan sistem yang berorientasi pada pengguna. Selain itu, monitoring dan evaluasi pasca-implementasi harus dilakukan secara berkala untuk memastikan sistem terus berkembang dan menyesuaikan dengan kebutuhan bisnis yang dinamis.

**BAB**

**13**

# **Tools dan Teknologi dalam Perancangan Sistem**

## A. Tools Perancangan Visual

**T**ools perancangan visual adalah perangkat lunak atau aplikasi yang digunakan oleh desainer dan pengembang untuk membuat representasi grafis dari sistem, aplikasi, atau produk digital (Manuhutu & Wattimena, 2019). Tools ini membantu menggambarkan ide, struktur, serta alur interaksi dengan cara yang lebih mudah dipahami dan dikomunikasikan kepada tim maupun pemangku kepentingan. Dalam proses pengembangan sistem, perancangan visual sangat penting untuk memvisualisasikan konsep awal, mengidentifikasi kebutuhan pengguna, dan merancang antarmuka yang efektif.

Salah satu jenis tools perancangan visual yang banyak digunakan adalah perangkat pembuat wireframe dan prototipe. Wireframe berfungsi sebagai kerangka kasar yang menunjukkan tata letak elemen-elemen pada layar tanpa detail visual lengkap. Tools seperti Balsamiq, Figma, dan Adobe XD memungkinkan desainer membuat wireframe dengan cepat dan mudah, sehingga dapat fokus pada struktur dan fungsionalitas daripada estetika. Prototipe interaktif yang dihasilkan dari tools ini juga dapat diuji coba untuk mendapatkan feedback awal (Ria & Budiman, 2021).

Selain itu, tools visual seperti Sketch dan Adobe Illustrator sering digunakan untuk membuat desain antarmuka dengan detail visual yang tinggi. Tools ini mendukung pembuatan elemen grafis, ikon, tipografi, dan warna yang sesuai dengan identitas merek atau tema produk. Dengan fitur layer, grid, dan alignment, desainer dapat menciptakan tampilan yang rapi, proporsional, dan estetik yang akan meningkatkan pengalaman pengguna secara keseluruhan.

Perancangan visual juga memerlukan tools untuk membuat diagram dan model yang menjelaskan proses bisnis atau arsitektur sistem. Contohnya adalah Microsoft Visio, Lucidchart, dan draw.io yang menyediakan fitur untuk membuat flowchart, diagram UML, dan peta proses. Diagram ini membantu tim pengembang dan pemangku kepentingan memahami bagaimana sistem bekerja dari sudut pandang proses dan interaksi antar komponen.

Tools kolaborasi online kini semakin populer karena memungkinkan tim yang tersebar secara geografis bekerja bersama secara real-time. Figma, misalnya, tidak hanya sebagai alat desain visual tapi juga platform kolaborasi yang memungkinkan tim memberikan komentar, revisi, dan masukan langsung pada desain yang sama. Hal ini mempercepat proses iterasi dan meminimalkan miskomunikasi selama pengembangan.

Selain tools khusus desain, beberapa platform pengembangan aplikasi juga menyediakan fitur visual builder yang memudahkan pembuatan antarmuka tanpa harus menulis kode secara manual. Contohnya adalah Webflow untuk web design atau FlutterFlow untuk aplikasi mobile. Tools ini memungkinkan desainer dan developer membuat prototipe dan produk akhir dengan cepat, menggabungkan perancangan visual dan pengkodean secara lebih efisien.

Penggunaan tools perancangan visual tidak hanya meningkatkan produktivitas tapi juga kualitas hasil desain. Dengan representasi visual yang jelas, semua pihak dapat memahami konsep dan fungsi sistem dengan lebih mudah, serta mengidentifikasi potensi masalah lebih awal sebelum pengembangan lebih lanjut. Hal ini tentu mengurangi risiko kegagalan proyek dan mempercepat waktu ke pasar.

Tools perancangan visual menjadi bagian integral dalam proses pengembangan sistem dan produk digital. Mereka menyediakan medium yang efektif untuk mengekspresikan ide, mengkomunikasikan visi, dan menguji konsep sebelum diimplementasikan secara teknis. Pemilihan tools yang sesuai dengan kebutuhan tim dan proyek akan memberikan dampak positif terhadap

kualitas desain dan keberhasilan proyek secara keseluruhan.

## **B. Software Pengembangan dan Prototyping**

Software pengembangan dan prototyping adalah alat penting dalam siklus pengembangan perangkat lunak yang memungkinkan tim untuk membangun, menguji, dan memodifikasi sistem secara cepat dan efisien. Software ini berfungsi untuk membuat model awal (prototipe) yang merepresentasikan fungsi dan tampilan sistem sebelum produk akhir dikembangkan. Prototipe ini membantu dalam mengidentifikasi kebutuhan pengguna, menguji konsep desain, serta meminimalkan risiko kesalahan pada tahap pengembangan selanjutnya.

Salah satu kategori software yang populer adalah tools prototyping interaktif seperti Figma, Adobe XD, dan InVision. Tools ini memungkinkan desainer untuk membuat simulasi antarmuka pengguna (UI) yang dapat diuji dan dievaluasi oleh pengguna maupun stakeholder tanpa perlu membangun sistem yang lengkap. Prototipe interaktif ini bisa menunjukkan bagaimana pengguna akan berinteraksi dengan sistem, termasuk navigasi antar halaman, tombol, dan fitur lain yang diinginkan.

Selain prototyping, ada pula software pengembangan yang mendukung pembuatan aplikasi secara langsung

dengan menggunakan fitur drag-and-drop atau low-code/no-code platform. Contohnya adalah OutSystems, Mendix, dan Microsoft Power Apps yang memudahkan pembuatan aplikasi bisnis tanpa memerlukan coding secara intensif. Platform ini sangat bermanfaat untuk menghemat waktu dan biaya, terutama dalam pengembangan aplikasi yang kompleks namun dengan deadline yang ketat.

Software pengembangan tradisional seperti Visual Studio, Eclipse, dan IntelliJ IDEA juga tetap menjadi tulang punggung dalam pembuatan perangkat lunak. Tools ini menyediakan lingkungan pengembangan terpadu (Integrated Development Environment/IDE) lengkap dengan editor kode, debugger, dan tools pendukung lain yang mempercepat proses coding dan testing. IDE ini sangat membantu programmer dalam menulis kode yang efisien dan mudah di-maintain.

Selain itu, penggunaan version control system (VCS) seperti Git dan platform hosting seperti GitHub dan GitLab sangat penting dalam pengembangan perangkat lunak modern. Tools ini memungkinkan kolaborasi antar pengembang secara terorganisir, melacak perubahan kode, dan mengelola berbagai versi produk secara aman. VCS juga sangat membantu dalam pengembangan

prototipe yang terus diperbarui sesuai feedback pengguna.

Software prototyping juga dapat melibatkan tools untuk membuat mockup statis, seperti Balsamiq dan Mockplus. Mockup ini adalah representasi visual statis yang lebih sederhana dibandingkan prototipe interaktif, berguna untuk mendiskusikan konsep awal desain dan mendapatkan persetujuan dari stakeholder sebelum melanjutkan ke tahap prototyping yang lebih kompleks.

Selain itu, tools simulasi dan testing otomatis juga menjadi bagian penting dari software pengembangan. Contohnya adalah Selenium untuk pengujian aplikasi web, atau Appium untuk aplikasi mobile. Software ini membantu mengotomatisasi pengujian prototipe dan aplikasi sehingga bug dan masalah dapat terdeteksi lebih awal dan diperbaiki sebelum produk final diluncurkan.

Software pengembangan dan prototyping memberikan landasan yang kuat bagi pengembangan sistem yang efektif dan efisien. Dengan menggunakan alat-alat ini, tim pengembang dapat memvisualisasikan dan menguji ide dengan lebih cepat, melakukan iterasi desain yang lebih baik, serta mengurangi risiko kegagalan proyek. Pemilihan software yang tepat sesuai kebutuhan proyek sangat menentukan keberhasilan proses pengembangan perangkat lunak secara keseluruhan.

### C. Teknologi Back-end dan Front-end Terkini

Teknologi back-end dan front-end merupakan dua komponen utama dalam pengembangan aplikasi modern. Back-end berfokus pada sisi server yang mengelola logika bisnis, database, autentikasi, dan komunikasi dengan front-end. Sedangkan front-end adalah bagian antarmuka pengguna yang berinteraksi langsung dengan user. Perkembangan teknologi di kedua bidang ini terus berjalan cepat untuk memenuhi kebutuhan aplikasi yang semakin kompleks, responsif, dan user-friendly.

Pada sisi back-end, teknologi terkini banyak berpusat pada penggunaan framework dan bahasa pemrograman yang mendukung pengembangan aplikasi cepat, skalabel, dan aman. Beberapa bahasa populer adalah JavaScript (Node.js), Python (Django, Flask), Ruby (Ruby on Rails), dan Go. Node.js, misalnya, menggunakan JavaScript untuk back-end sehingga memudahkan pengembang menggunakan satu bahasa untuk front-end dan back-end, meningkatkan efisiensi pengembangan.

Selain bahasa pemrograman, penggunaan arsitektur microservices semakin populer di back-end modern. Dengan microservices, aplikasi dibagi menjadi layanan kecil yang independen dan bisa dikembangkan, diuji, serta di-deploy secara terpisah. Ini meningkatkan skalabilitas

dan fleksibilitas sistem, serta mempermudah pemeliharaan. Contoh platform yang mendukung microservices adalah Kubernetes untuk orkestrasi container dan Docker untuk containerisasi aplikasi.

Teknologi database juga mengalami perkembangan pesat pada back-end. Selain database relasional tradisional seperti MySQL dan PostgreSQL, database NoSQL seperti MongoDB, Cassandra, dan Redis banyak digunakan untuk menangani data dengan skema fleksibel dan volume besar. Penggunaan database berbasis cloud seperti Amazon DynamoDB dan Google Firestore juga meningkat seiring kebutuhan aplikasi yang harus selalu online dan dapat diakses secara global.

Di sisi front-end, framework dan library modern menjadi kunci dalam membangun antarmuka pengguna yang dinamis dan responsif. React.js dari Facebook, Angular dari Google, dan Vue.js adalah tiga teknologi populer yang banyak digunakan untuk membuat aplikasi web interaktif. React misalnya, menawarkan virtual DOM yang meningkatkan performa rendering halaman, serta komponen reusable yang mempercepat pengembangan.

Selain framework, teknologi front-end terkini juga fokus pada pembuatan aplikasi mobile dan progresif. Framework seperti React Native dan Flutter memungkinkan pengembang membuat aplikasi mobile

dengan kode yang hampir sama untuk Android dan iOS. Progressive Web Apps (PWA) juga semakin populer, yang memungkinkan aplikasi web berfungsi layaknya aplikasi native dengan kemampuan offline dan notifikasi push.

Perkembangan tools pendukung juga sangat membantu dalam pengembangan front-end, seperti bundler (Webpack, Rollup), package manager (npm, Yarn), dan tools testing (Jest, Cypress). Tools ini membantu pengembang mengelola dependensi, mengoptimasi performa aplikasi, serta memastikan kualitas kode melalui pengujian otomatis dan manual.

Teknologi back-end dan front-end terkini sangat berfokus pada kecepatan pengembangan, skalabilitas, dan pengalaman pengguna yang baik. Kombinasi teknologi ini memungkinkan pengembangan aplikasi modern yang dapat berjalan lancar di berbagai perangkat dan platform. Pemilihan teknologi yang tepat harus disesuaikan dengan kebutuhan proyek, tim, dan tujuan bisnis agar hasilnya optimal dan berkelanjutan.

#### **D. Integrasi Tools dalam Workflow Tim**

Integrasi tools dalam workflow tim merupakan aspek penting yang menentukan efektivitas dan efisiensi proses kerja dalam pengembangan sistem atau proyek teknologi. Dengan semakin kompleksnya proyek dan banyaknya

anggota tim yang terlibat, penggunaan berbagai tools yang terintegrasi secara mulus memungkinkan koordinasi lebih baik, komunikasi yang lancar, dan pengelolaan tugas yang terstruktur. Integrasi ini membantu meminimalkan kesalahan, mempercepat penyelesaian pekerjaan, serta menjaga konsistensi hasil.

Salah satu contoh penting dari integrasi tools adalah penggunaan platform manajemen proyek seperti Jira, Trello, atau Asana yang terhubung dengan sistem version control seperti GitHub atau GitLab. Integrasi ini memungkinkan tim untuk melacak progress pengembangan berdasarkan tiket atau tugas yang dibuat, sekaligus mengaitkan commit kode secara langsung dengan fitur atau bug tertentu. Dengan demikian, setiap perubahan kode dapat dipantau keterkaitannya dengan kebutuhan atau masalah yang dipecahkan.

Selain manajemen proyek dan version control, integrasi juga dilakukan antara tools komunikasi seperti Slack, Microsoft Teams, atau Discord dengan sistem manajemen tugas dan kode. Misalnya, pemberitahuan otomatis dapat dikirimkan ke channel tertentu ketika ada update status tugas, push kode baru, atau hasil pengujian otomatis. Ini memudahkan seluruh anggota tim untuk tetap update tanpa harus mengecek banyak aplikasi

secara manual, sehingga mempercepat respon terhadap isu yang muncul.

Dalam konteks pengembangan perangkat lunak, integrasi Continuous Integration/Continuous Deployment (CI/CD) juga sangat penting. Tools seperti Jenkins, CircleCI, atau GitLab CI/CD secara otomatis menjalankan build, testing, dan deployment ketika ada perubahan kode yang di-push. Dengan integrasi ini, tim dapat memastikan kualitas kode sebelum dirilis dan mengurangi risiko kesalahan manual dalam proses deployment, sekaligus mempercepat waktu release produk ke pengguna.

Selain itu, integrasi antara tools desain dan pengembangan juga menjadi tren yang penting. Misalnya, platform seperti Figma atau Adobe XD bisa diintegrasikan dengan repository kode atau alat prototyping untuk mempercepat transfer desain menjadi kode yang bisa langsung diuji. Hal ini membantu mengurangi miskomunikasi antara desainer dan developer, serta memastikan implementasi sesuai dengan visi desain awal.

Automasi juga menjadi bagian integral dari workflow yang terintegrasi. Tools seperti Zapier atau Integromat memungkinkan penghubung otomatis antar aplikasi berbeda yang digunakan tim, sehingga tugas-tugas rutin seperti update status, backup data, atau notifikasi dapat

berjalan otomatis tanpa intervensi manual. Ini membantu menghemat waktu dan mengurangi potensi human error dalam workflow sehari-hari.

Kunci keberhasilan integrasi tools dalam workflow tim adalah memilih tools yang kompatibel dan memiliki API terbuka sehingga mudah dikoneksikan satu sama lain. Selain itu, penting juga untuk melakukan pelatihan dan sosialisasi kepada seluruh anggota tim agar mereka memahami bagaimana memanfaatkan integrasi tersebut secara maksimal. Workflow yang terintegrasi dengan baik akan mendorong kolaborasi yang lebih efektif dan menghasilkan produk yang berkualitas tinggi.

Integrasi tools dalam workflow tim merupakan fondasi utama dalam pengelolaan proyek modern yang kompleks. Dengan integrasi yang baik, tim dapat bekerja lebih terkoordinasi, mengurangi hambatan komunikasi, serta meningkatkan kecepatan dan kualitas pengembangan. Investasi dalam membangun ekosistem tools yang terhubung sangat berharga untuk mendukung keberhasilan jangka panjang sebuah proyek atau organisasi.

**BAB**

**14**

# **Perancangan Sistem Berbasis Web**

## A. Karakteristik Sistem Web

**S**istem web adalah aplikasi atau layanan yang berjalan di atas jaringan internet atau intranet dan diakses melalui browser web. Salah satu karakteristik utama sistem web adalah kemampuannya untuk diakses secara global tanpa memerlukan instalasi perangkat lunak khusus pada perangkat pengguna. Cukup menggunakan browser, pengguna dapat berinteraksi dengan sistem dari mana saja dan kapan saja, selama terhubung dengan jaringan. Hal ini membuat sistem web sangat fleksibel dan mudah dijangkau oleh banyak pengguna.

Karakteristik lainnya adalah sifat sistem web yang berbasis client-server. Sistem web terdiri dari dua bagian utama yaitu front-end (client) yang berinteraksi langsung dengan pengguna dan back-end (server) yang mengelola logika bisnis, penyimpanan data, dan pengolahan permintaan. Arsitektur ini memungkinkan pemisahan fungsi dan tanggung jawab sehingga pengembangan dan pemeliharaan sistem dapat dilakukan lebih efisien (Manuhutu & Wattimena, 2019).

Sistem web juga biasanya bersifat dinamis, artinya konten dan fungsi yang ditampilkan dapat berubah sesuai dengan interaksi pengguna, data terbaru, atau aturan bisnis tertentu. Contohnya, pada situs e-commerce, halaman produk dapat berubah berdasarkan stok,

promosi, dan preferensi pengguna. Sifat dinamis ini dimungkinkan melalui penggunaan bahasa pemrograman sisi server seperti PHP, Python, atau Node.js yang menghasilkan halaman web secara real-time.

Keamanan menjadi karakteristik penting lain dalam sistem web. Karena sistem ini dapat diakses secara terbuka melalui internet, maka diperlukan mekanisme keamanan yang kuat seperti autentikasi pengguna, enkripsi data, dan proteksi terhadap serangan siber seperti SQL injection dan cross-site scripting (XSS). Keamanan ini sangat krusial untuk melindungi data sensitif dan menjaga kepercayaan pengguna.

Kemampuan skalabilitas juga menjadi karakteristik sistem web yang penting. Sistem web harus mampu menangani jumlah pengguna dan permintaan yang terus bertambah tanpa mengalami penurunan performa. Dengan teknologi modern seperti cloud computing dan load balancing, sistem web dapat diatur agar secara otomatis menyesuaikan kapasitas server sesuai kebutuhan sehingga tetap responsif dan stabil.

Sistem web biasanya juga mendukung interoperabilitas dan integrasi dengan berbagai layanan dan platform lain melalui API (Application Programming Interface). Hal ini memungkinkan pengembang menghubungkan sistem web dengan aplikasi eksternal,

database, atau layanan pihak ketiga seperti pembayaran online dan media sosial, sehingga memperkaya fungsi dan nilai tambah sistem.

Selain itu, sistem web harus dirancang agar responsif dan kompatibel dengan berbagai perangkat, mulai dari komputer desktop hingga smartphone dan tablet. Desain responsif dan adaptif memastikan tampilan dan fungsi sistem tetap optimal dan mudah digunakan pada layar dengan ukuran dan resolusi yang berbeda-beda, meningkatkan pengalaman pengguna.

Terakhir, pemeliharaan dan pembaruan sistem web relatif lebih mudah dibandingkan aplikasi desktop. Karena kode dan data tersimpan di server, pembaruan dapat dilakukan secara sentral tanpa perlu melibatkan pengguna secara langsung. Hal ini memungkinkan sistem web selalu up-to-date dengan fitur terbaru dan perbaikan keamanan, menjadikannya solusi ideal untuk aplikasi yang terus berkembang.

## **B. Teknologi Pengembangan Web**

Teknologi pengembangan web meliputi berbagai alat, bahasa pemrograman, dan framework yang digunakan untuk membangun aplikasi dan situs web. Teknologi ini terbagi menjadi dua bagian utama, yaitu front-end yang berfokus pada tampilan dan interaksi

pengguna, serta back-end yang menangani logika bisnis, database, dan server. Penggunaan teknologi yang tepat sangat menentukan kualitas, performa, dan skalabilitas sistem web yang dikembangkan.

Di sisi front-end, bahasa dasar yang digunakan adalah HTML, CSS, dan JavaScript. HTML (HyperText Markup Language) berfungsi untuk membuat struktur konten halaman web, CSS (Cascading Style Sheets) mengatur tata letak dan tampilan visual, sedangkan JavaScript memberikan interaktivitas seperti animasi, validasi form, dan manipulasi elemen halaman secara dinamis. Kombinasi ketiga teknologi ini merupakan fondasi bagi semua aplikasi web modern.

Untuk mempermudah pengembangan front-end, banyak framework dan library JavaScript yang populer digunakan, seperti React, Angular, dan Vue.js. React, misalnya, dikenal karena kemampuannya membuat UI berbasis komponen yang mudah digunakan kembali dan performa tinggi berkat virtual DOM. Angular menyediakan framework lengkap dengan fitur seperti routing dan dependency injection, sedangkan Vue.js menawarkan kemudahan belajar dan integrasi yang fleksibel.

Di sisi back-end, berbagai bahasa pemrograman dan framework digunakan sesuai kebutuhan dan preferensi

tim pengembang. Beberapa bahasa populer adalah PHP dengan framework Laravel, Python dengan Django dan Flask, Ruby dengan Ruby on Rails, serta JavaScript menggunakan Node.js. Framework ini menyediakan struktur dan fitur yang memudahkan pengembangan aplikasi yang kompleks seperti autentikasi, routing, dan manajemen database.

Teknologi database juga sangat penting dalam pengembangan web. Database relasional seperti MySQL, PostgreSQL, dan SQL Server masih banyak digunakan karena kemampuannya dalam mengelola data terstruktur secara efisien. Sementara itu, database NoSQL seperti MongoDB dan Cassandra digunakan untuk aplikasi yang membutuhkan skema data lebih fleksibel dan mampu menangani data dalam volume besar atau tipe data yang tidak terstruktur.

Selain bahasa dan database, teknologi server dan hosting juga menjadi bagian dari pengembangan web. Server web populer seperti Apache dan Nginx sering digunakan untuk mengelola permintaan HTTP dan mengirimkan konten web ke pengguna. Cloud platform seperti AWS, Google Cloud, dan Microsoft Azure juga menyediakan layanan hosting yang scalable dan mendukung deployment aplikasi secara global dengan infrastruktur yang andal.

Untuk mempercepat proses pengembangan dan menjaga kualitas, teknologi DevOps juga diterapkan dalam pengembangan web. Tools Continuous Integration/Continuous Deployment (CI/CD) seperti Jenkins, GitLab CI, dan CircleCI memungkinkan otomatisasi build, testing, dan deployment aplikasi. Dengan demikian, pembaruan dapat dilakukan secara cepat dan minim risiko kesalahan.

Terakhir, teknologi pengembangan web kini semakin mengadopsi pendekatan mobile-first dan Progressive Web Apps (PWA) untuk memastikan aplikasi dapat berjalan optimal di berbagai perangkat, termasuk smartphone dan tablet. PWA memungkinkan aplikasi web memiliki fitur seperti offline mode, push notification, dan tampilan seperti aplikasi native, sehingga memberikan pengalaman pengguna yang lebih baik dan luas. Penguasaan teknologi-teknologi ini menjadi kunci sukses dalam pengembangan sistem web modern.

### **C. Keamanan dan Skalabilitas Sistem Web**

Keamanan dan skalabilitas merupakan dua aspek krusial yang harus diperhatikan dalam pengembangan sistem web. Keamanan bertujuan melindungi sistem dan data pengguna dari ancaman seperti peretasan, pencurian data, dan serangan siber lainnya. Sementara skalabilitas

berkaitan dengan kemampuan sistem untuk menangani pertumbuhan jumlah pengguna dan data tanpa mengalami penurunan performa. Keduanya saling berkaitan dan menjadi faktor utama keberhasilan sebuah sistem web dalam jangka panjang.

Keamanan sistem web dimulai dari penerapan autentikasi dan otorisasi yang kuat. Autentikasi memastikan bahwa hanya pengguna yang terdaftar dan valid yang dapat mengakses sistem, sedangkan otorisasi membatasi hak akses pengguna sesuai peran dan kebutuhannya. Penggunaan protokol keamanan seperti OAuth, JWT (JSON Web Token), dan multi-factor authentication (MFA) semakin banyak diterapkan untuk memperkuat lapisan perlindungan ini.

Selain itu, enkripsi data, baik saat transmisi maupun penyimpanan, sangat penting untuk menjaga kerahasiaan informasi. HTTPS dengan SSL/TLS menjadi standar wajib untuk komunikasi yang aman antara browser dan server. Untuk data yang disimpan, enkripsi basis data dan penggunaan teknik hashing pada data sensitif seperti password membantu mencegah pencurian data walaupun terjadi pelanggaran keamanan (Bagir & Putro, 2018).

Sistem web juga harus dilindungi dari serangan umum seperti SQL injection, cross-site scripting (XSS), dan cross-site request forgery (CSRF). Teknik

pengembangan aman (secure coding), penggunaan firewall aplikasi web (WAF), serta penerapan kebijakan Content Security Policy (CSP) dapat meminimalkan risiko serangan tersebut. Penting juga dilakukan audit keamanan secara berkala dan penggunaan tools penetration testing untuk mengidentifikasi celah keamanan.

Untuk aspek skalabilitas, sistem web harus didesain agar mampu menangani peningkatan jumlah pengguna dan data tanpa menimbulkan bottleneck. Pendekatan arsitektur microservices dan penggunaan container seperti Docker memungkinkan pengembangan dan deployment layanan secara terpisah dan mudah diskalakan sesuai kebutuhan. Infrastruktur cloud dengan kemampuan auto-scaling seperti AWS atau Google Cloud juga membantu mengatur kapasitas server secara dinamis.

Load balancing adalah teknologi penting dalam meningkatkan skalabilitas sistem web. Dengan mendistribusikan beban kerja ke beberapa server, load balancer menjaga performa aplikasi tetap optimal saat terjadi lonjakan trafik. Selain itu, caching pada sisi server dan client dapat mengurangi beban permintaan ke server utama dengan menyimpan data sementara yang sering diakses.

Penggunaan database yang scalable juga menjadi perhatian utama. Database relasional dapat menggunakan teknik sharding atau clustering untuk membagi beban kerja, sedangkan database NoSQL secara inheren mendukung distribusi data secara horizontal. Pengelolaan indeks dan query yang efisien juga sangat menentukan performa sistem pada saat beban tinggi.

Akhirnya, pengelolaan keamanan dan skalabilitas harus dilakukan secara berkelanjutan dengan monitoring dan evaluasi yang rutin. Sistem pemantauan performa dan keamanan seperti Grafana, Prometheus, atau Splunk membantu tim IT mendeteksi masalah sejak dini dan mengambil tindakan cepat. Dengan menjaga keamanan dan skalabilitas secara bersamaan, sistem web dapat memberikan layanan yang handal, cepat, dan aman bagi penggunanya.

#### **D. Optimasi Performa dan UX Web**

Optimasi performa dan pengalaman pengguna (User Experience/UX) pada sistem web merupakan dua aspek penting yang sangat menentukan keberhasilan sebuah aplikasi web. Performa yang baik memastikan halaman web cepat dimuat dan responsif, sementara UX yang baik memastikan pengguna merasa nyaman, mudah bernavigasi, dan puas saat menggunakan aplikasi. Keduanya saling berkaitan karena performa yang buruk

dapat merusak pengalaman pengguna, dan UX yang baik tanpa performa optimal juga tidak maksimal.

Salah satu langkah utama dalam optimasi performa web adalah meminimalkan waktu pemuatan halaman (page load time). Waktu muat yang cepat dapat dicapai dengan berbagai teknik, seperti kompresi file, pengurangan ukuran gambar, dan penggunaan format gambar modern seperti WebP. Dengan memperkecil ukuran file yang dikirim dari server ke browser, pengguna dapat mengakses konten dengan lebih cepat, terutama bagi pengguna dengan koneksi internet yang lambat.

Selain itu, teknik lazy loading gambar dan konten juga menjadi metode efektif dalam meningkatkan performa. Lazy loading berarti hanya konten atau gambar yang berada di viewport pengguna yang dimuat terlebih dahulu, sedangkan konten lain akan dimuat saat pengguna menggulir halaman. Teknik ini mengurangi beban awal dan mempercepat waktu muat halaman secara keseluruhan.

Penggunaan caching adalah strategi penting lainnya untuk optimasi performa web. Cache memungkinkan penyimpanan sementara data statis di browser atau di server proxy sehingga ketika pengguna mengunjungi kembali halaman tersebut, data tidak perlu dimuat ulang dari server. Caching dapat dilakukan pada berbagai level,

mulai dari browser cache, server-side cache, hingga Content Delivery Network (CDN) yang menyebarkan konten ke server yang dekat dengan lokasi pengguna.

Content Delivery Network (CDN) juga sangat berperan dalam optimasi performa. CDN adalah jaringan server yang tersebar di berbagai lokasi geografis dan menyimpan salinan konten web. Dengan menggunakan CDN, permintaan pengguna akan dilayani oleh server yang paling dekat, sehingga mengurangi latency dan mempercepat waktu akses. Ini sangat berguna untuk situs dengan pengguna dari berbagai belahan dunia.

Penggunaan teknik minifikasi pada file CSS, JavaScript, dan HTML juga membantu mempercepat loading. Minifikasi menghapus spasi, komentar, dan karakter tidak perlu lainnya dari kode sumber tanpa mengubah fungsi, sehingga ukuran file menjadi lebih kecil dan proses pengunduhan menjadi lebih cepat. Tools seperti UglifyJS dan CSSNano sering digunakan untuk tujuan ini.

Dalam hal UX, desain yang responsif dan adaptif menjadi keharusan untuk memastikan aplikasi web dapat digunakan dengan baik pada berbagai perangkat, mulai dari desktop, tablet, hingga smartphone. Desain responsif menggunakan teknik CSS media queries agar layout dan elemen tampilan menyesuaikan ukuran layar perangkat,

sehingga pengguna mendapatkan pengalaman yang optimal tanpa perlu melakukan zoom atau scroll horizontal.

Navigasi yang mudah dan intuitif juga sangat penting dalam UX. Pengguna harus dapat dengan cepat menemukan informasi atau fitur yang mereka cari tanpa kebingungan. Penggunaan menu yang jelas, breadcrumb, dan pencarian internal yang efektif membantu memudahkan pengguna bernavigasi dalam situs web. Selain itu, konsistensi desain antar halaman juga memberikan kesan profesional dan mempercepat adaptasi pengguna terhadap antarmuka.

Kecepatan respons antar interaksi pengguna juga menjadi bagian dari UX yang perlu dioptimalkan. Animasi dan transisi yang halus tanpa lag akan memberikan pengalaman yang menyenangkan. Namun, penggunaan animasi juga harus bijaksana agar tidak membebani performa sistem dan mengganggu fokus pengguna. Prinsip “less is more” sering diterapkan agar UX tetap efisien dan menarik.

Faktor aksesibilitas juga harus diperhatikan dalam desain UX. Situs web yang dapat diakses oleh semua kalangan, termasuk penyandang disabilitas, menunjukkan kepedulian dan profesionalisme. Penggunaan warna yang kontras, teks alternatif pada gambar, serta navigasi

menggunakan keyboard adalah beberapa aspek penting untuk memastikan aksesibilitas terpenuhi.

Pengujian UX dengan melibatkan pengguna nyata sangat penting untuk mendapatkan masukan yang berharga. Melalui user testing, heatmaps, dan analisis perilaku pengguna, pengembang dapat memahami bagaimana pengguna berinteraksi dengan aplikasi dan area mana yang perlu diperbaiki. Iterasi desain berdasarkan data ini akan meningkatkan kualitas UX secara signifikan.

Selain itu, monitoring performa web secara real-time dengan menggunakan tools seperti Google PageSpeed Insights, Lighthouse, atau GTmetrix membantu pengembang mengidentifikasi bottleneck dan peluang optimasi. Data metrik seperti First Contentful Paint (FCP), Time to Interactive (TTI), dan Largest Contentful Paint (LCP) menjadi indikator utama dalam evaluasi performa.

Penggunaan Progressive Web Apps (PWA) juga merupakan salah satu solusi modern dalam optimasi performa dan UX. PWA memungkinkan aplikasi web berfungsi layaknya aplikasi native dengan fitur offline, push notification, dan loading yang sangat cepat. Dengan PWA, pengalaman pengguna menjadi lebih lancar dan dapat diakses tanpa tergantung penuh pada koneksi internet.

Terakhir, kolaborasi antara tim pengembang, desainer UX, dan pihak pemangku kepentingan harus terus dijaga untuk memastikan optimasi performa dan UX berjalan selaras. Dengan pendekatan yang terintegrasi, sistem web yang dihasilkan tidak hanya cepat dan handal, tetapi juga menyenangkan dan mudah digunakan oleh penggunanya, sehingga mampu meningkatkan kepuasan dan loyalitas pengguna secara signifikan.

**BAB**

**15**

**Tren Masa Depan  
dalam Perancangan  
Sistem Informasi**

## A. Kecerdasan Buatan dan Otomatisasi

**K**ecerdasan Buatan (Artificial Intelligence/AI) merupakan bidang teknologi yang fokus pada pengembangan sistem yang mampu melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia. AI meliputi berbagai teknik seperti machine learning, deep learning, natural language processing, dan computer vision. Teknologi ini memungkinkan komputer untuk belajar dari data, mengenali pola, membuat keputusan, serta berinteraksi dengan lingkungan secara cerdas.

Otomatisasi, di sisi lain, adalah proses menggunakan teknologi untuk menjalankan tugas-tugas secara otomatis tanpa memerlukan campur tangan manusia secara langsung (Zaliluddin & Rohmat, 2018). Otomatisasi dapat diterapkan dalam berbagai sektor mulai dari manufaktur, layanan pelanggan, hingga pengelolaan data. Ketika digabungkan dengan AI, otomatisasi menjadi jauh lebih canggih karena sistem tidak hanya menjalankan perintah, tetapi juga dapat beradaptasi dan mengambil keputusan berdasarkan analisis data.

Salah satu penerapan AI dalam otomatisasi adalah robotic process automation (RPA), yaitu teknologi yang memungkinkan perangkat lunak untuk meniru tindakan manusia dalam proses bisnis rutin seperti pengisian formulir, pemrosesan transaksi, dan manajemen data.

Dengan RPA, perusahaan dapat meningkatkan efisiensi operasional, mengurangi kesalahan manual, dan menghemat biaya.

Dalam dunia bisnis, AI dan otomatisasi membantu meningkatkan produktivitas melalui analisis data yang cepat dan akurat. Sistem AI dapat mengolah data dalam jumlah besar untuk menghasilkan insight yang berguna dalam pengambilan keputusan strategis. Selain itu, otomatisasi memungkinkan proses bisnis berjalan secara konsisten dan tanpa hambatan, sehingga perusahaan bisa lebih fokus pada inovasi dan pengembangan produk.

Di bidang layanan pelanggan, AI digunakan dalam bentuk chatbot dan asisten virtual yang mampu memberikan respon cepat dan personal kepada pengguna. Chatbot yang didukung AI dapat memahami bahasa alami, menjawab pertanyaan, dan menyelesaikan masalah sederhana tanpa perlu melibatkan staf manusia. Hal ini meningkatkan kepuasan pelanggan sekaligus mengurangi beban kerja customer service.

Teknologi AI juga banyak diterapkan dalam otomasi sistem manufaktur melalui robot pintar dan sistem kendali otomatis. Robot ini dapat melakukan pekerjaan repetitif dengan presisi tinggi dan dalam waktu yang lebih singkat dibanding manusia. Selain itu, sensor dan teknologi IoT yang terintegrasi memungkinkan

pemantauan kondisi mesin secara real-time dan prediksi perawatan sebelum kerusakan terjadi.

Otomatisasi berbasis AI juga memainkan peran penting dalam pengelolaan rantai pasok (supply chain management). Dengan analisis prediktif, sistem dapat memperkirakan permintaan pasar, mengoptimalkan stok, dan mempercepat proses pengiriman. Hal ini membantu perusahaan mengurangi biaya penyimpanan dan meningkatkan responsivitas terhadap perubahan pasar.

Sektor kesehatan juga merasakan dampak besar dari AI dan otomatisasi. Sistem AI digunakan untuk mendiagnosa penyakit melalui analisis citra medis, memberikan rekomendasi pengobatan, dan memantau kondisi pasien secara terus-menerus dengan perangkat wearable. Otomatisasi dalam pengelolaan rekam medis dan administrasi rumah sakit mempercepat layanan dan mengurangi kesalahan.

Di bidang keuangan, AI digunakan untuk deteksi penipuan, analisis risiko kredit, dan pengelolaan investasi otomatis (robo-advisors). Dengan otomatisasi, proses verifikasi dan persetujuan pinjaman menjadi lebih cepat, serta keputusan investasi dapat diambil berdasarkan data dan model yang kompleks tanpa bias manusia. Ini meningkatkan keamanan dan efisiensi layanan keuangan.

Implementasi AI dan otomatisasi tidak lepas dari tantangan, seperti kebutuhan akan data yang besar dan berkualitas, serta masalah etika dan privasi. Pengumpulan dan penggunaan data harus mematuhi regulasi yang ketat agar tidak melanggar hak privasi pengguna. Selain itu, keputusan otomatis perlu diawasi untuk menghindari diskriminasi atau kesalahan yang merugikan.

Dampak sosial dari otomatisasi juga menjadi perhatian, terutama terkait potensi pengurangan lapangan kerja. Namun, otomatisasi juga menciptakan peluang pekerjaan baru yang memerlukan keterampilan teknologi tinggi dan pengelolaan sistem AI. Oleh karena itu, peningkatan kapasitas sumber daya manusia melalui pelatihan dan pendidikan menjadi hal penting.

Dalam pengembangan sistem web dan aplikasi modern, AI dan otomatisasi semakin diintegrasikan untuk meningkatkan fungsionalitas dan pengalaman pengguna. Contohnya adalah rekomendasi produk yang dipersonalisasi, sistem pencarian cerdas, dan otomatisasi tugas administratif di belakang layar yang membuat aplikasi lebih responsif dan efisien.

Kecerdasan Buatan dan otomatisasi juga memungkinkan terciptanya sistem yang adaptif dan proaktif. Sistem dapat memprediksi kebutuhan pengguna, menyesuaikan layanan secara real-time, dan memberikan

pengalaman yang lebih personal. Hal ini membuka peluang baru bagi bisnis untuk meningkatkan loyalitas pelanggan dan menciptakan nilai tambah.

AI dan otomatisasi adalah kekuatan pendorong transformasi digital di berbagai sektor. Dengan penerapan yang tepat, kedua teknologi ini mampu meningkatkan efisiensi operasional, kualitas layanan, dan inovasi produk secara signifikan. Namun, implementasi harus disertai dengan pendekatan yang bijak untuk mengatasi tantangan teknis, etis, dan sosial agar manfaatnya dapat dirasakan secara optimal.

## **B. Perancangan Sistem Berbasis Cloud dan Edge**

Perancangan sistem berbasis cloud dan edge merupakan pendekatan modern dalam pengembangan aplikasi dan layanan digital yang menyesuaikan kebutuhan akan skalabilitas, performa, dan keandalan. Cloud computing menyediakan infrastruktur dan sumber daya komputasi yang dapat diakses secara fleksibel melalui internet, sedangkan edge computing membawa proses komputasi lebih dekat ke lokasi pengguna atau sumber data. Kombinasi kedua teknologi ini memungkinkan pengembangan sistem yang efisien dan responsif.

Cloud computing menawarkan keuntungan utama berupa kemampuan untuk mengakses sumber daya komputasi secara elastis sesuai kebutuhan, tanpa harus mengelola infrastruktur fisik secara langsung. Dengan layanan cloud, pengembang dapat memanfaatkan server, penyimpanan, database, hingga layanan AI dan analitik secara on-demand dan dengan biaya yang relatif terukur. Hal ini mendukung pengembangan sistem yang dapat tumbuh dengan cepat dan mudah diadaptasi.

Dalam perancangan sistem berbasis cloud, arsitektur biasanya dirancang dengan prinsip scalable dan fault tolerant. Penggunaan container dan microservices sering dipilih agar setiap komponen dapat berjalan secara independen dan mudah di-scale sesuai kebutuhan trafik. Cloud provider seperti AWS, Azure, dan Google Cloud menyediakan berbagai layanan yang mendukung arsitektur ini, termasuk load balancer, auto-scaling groups, dan managed databases.

Namun, kendala latency atau keterlambatan dalam pengiriman data sering menjadi masalah pada sistem yang hanya bergantung pada cloud, terutama untuk aplikasi yang membutuhkan respon waktu nyata (real-time) atau berada di lokasi dengan koneksi internet yang tidak stabil. Di sinilah edge computing berperan penting

dengan membawa proses komputasi dan penyimpanan lebih dekat ke perangkat pengguna atau sumber data.

Edge computing memungkinkan pengolahan data secara lokal, mengurangi beban komunikasi dengan cloud pusat dan menurunkan latency secara signifikan. Misalnya, dalam sistem IoT (Internet of Things), data dari sensor dapat diproses di edge device untuk mengambil keputusan cepat, seperti mendeteksi kegagalan mesin atau mengatur lalu lintas secara otomatis, tanpa harus mengirimkan data ke cloud terlebih dahulu.

Dalam perancangan sistem hybrid yang menggabungkan cloud dan edge, penting untuk menentukan fungsi mana yang dijalankan di edge dan mana yang tetap berada di cloud. Proses yang membutuhkan latensi rendah dan pengolahan data besar secara lokal biasanya dipindahkan ke edge, sementara fungsi yang membutuhkan kapasitas komputasi tinggi dan penyimpanan besar tetap dijalankan di cloud.

Keamanan menjadi aspek krusial dalam sistem berbasis cloud dan edge. Perancangan harus melibatkan proteksi data baik saat transit maupun saat disimpan di cloud atau edge device. Penggunaan enkripsi, autentikasi yang ketat, serta monitoring keamanan secara terus-menerus sangat penting untuk mencegah kebocoran data dan serangan siber.

Pengelolaan sumber daya juga menuntut perancangan sistem yang mampu memantau penggunaan komputasi, penyimpanan, dan bandwidth di kedua lingkungan, cloud dan edge. Penggunaan platform manajemen hybrid cloud dan edge dapat membantu dalam mengoptimalkan penggunaan sumber daya dan mengotomatisasi scaling sesuai kebutuhan aplikasi.

Dalam hal integrasi, perancangan sistem berbasis cloud dan edge perlu memperhatikan interoperabilitas antar berbagai komponen dan layanan. API yang standar dan protokol komunikasi yang efisien seperti MQTT dan REST API banyak digunakan untuk memastikan data dan perintah dapat ditransfer dengan lancar antara edge device dan cloud.

Pengembangan sistem berbasis cloud dan edge juga mengadopsi model DevOps dan continuous integration/continuous deployment (CI/CD) agar proses update dan pemeliharaan aplikasi dapat dilakukan dengan cepat dan tanpa downtime. Infrastruktur sebagai kode (IaC) menjadi metode populer untuk mengelola konfigurasi lingkungan secara otomatis dan konsisten.

Selain itu, pengujian performa dan keamanan harus dilakukan secara menyeluruh di kedua lingkungan. Pengujian beban (load testing) dan uji keamanan pada edge device serta cloud infrastructure sangat penting

untuk memastikan sistem tahan terhadap lonjakan trafik dan serangan siber.

Perancangan sistem berbasis cloud dan edge juga memperhatikan aspek data analytics dan machine learning. Data yang diolah di edge dapat langsung digunakan untuk analitik real-time, sedangkan data agregat dikirim ke cloud untuk analitik lebih mendalam dan pengembangan model AI. Pendekatan ini memaksimalkan kecepatan respon sekaligus mendapatkan wawasan yang komprehensif.

Penggunaan cloud dan edge computing memungkinkan perusahaan menyesuaikan layanan mereka berdasarkan kebutuhan pengguna dan kondisi lokal, misalnya dengan mengoptimalkan pengalaman pengguna di wilayah dengan koneksi lambat atau menyediakan fitur offline melalui edge device.

Perancangan sistem berbasis cloud dan edge menghadirkan solusi yang fleksibel, cepat, dan handal untuk memenuhi tantangan aplikasi modern yang memerlukan skalabilitas tinggi, latensi rendah, dan keamanan ketat. Pendekatan ini mendorong inovasi dalam berbagai bidang mulai dari smart city, industri 4.0, hingga layanan digital konsumen.

### **C. Desain Berkelanjutan dan Green IT**

Desain berkelanjutan dalam teknologi informasi adalah pendekatan yang menempatkan prinsip ramah lingkungan dan efisiensi sumber daya sebagai fokus utama dalam pengembangan sistem dan infrastruktur TI. Konsep ini bertujuan untuk meminimalkan dampak negatif terhadap lingkungan, seperti emisi karbon, konsumsi energi, dan limbah elektronik, sekaligus memastikan bahwa teknologi yang digunakan dapat terus berkembang tanpa mengorbankan kebutuhan generasi mendatang.

Green IT (Teknologi Informasi Hijau) adalah bagian penting dari desain berkelanjutan yang berfokus pada pengelolaan dan penggunaan teknologi secara efisien dan ramah lingkungan. Green IT mencakup berbagai praktik mulai dari penggunaan perangkat keras hemat energi, optimasi data center, hingga pengelolaan limbah elektronik yang bertanggung jawab. Implementasi Green IT membantu organisasi mengurangi jejak karbon dan meningkatkan citra sosialnya.

Salah satu aspek penting dari desain berkelanjutan adalah efisiensi energi dalam perangkat keras dan perangkat lunak. Desain sistem yang hemat energi mengurangi konsumsi listrik melalui optimasi algoritma, penggunaan prosesor dengan teknologi rendah daya, dan

pemanfaatan server dengan rating efisiensi tinggi. Contohnya, penggunaan teknologi virtualisasi pada data center memungkinkan konsolidasi server sehingga mengurangi jumlah perangkat fisik yang beroperasi.

Selain hardware, pengembangan perangkat lunak juga harus mempertimbangkan efisiensi kode agar tidak membebani sistem dengan proses yang tidak perlu, sehingga mengurangi konsumsi daya. Software yang ringan dan teroptimasi dapat memperpanjang usia baterai perangkat mobile dan mengurangi beban pada infrastruktur cloud, yang secara langsung berdampak pada pengurangan konsumsi energi.

Penggunaan data center hijau merupakan salah satu fokus utama dalam Green IT. Data center hijau dirancang dengan teknologi pendingin efisien, sumber energi terbarukan, dan sistem manajemen energi canggih untuk menekan penggunaan listrik. Selain itu, penempatan data center di lokasi strategis dengan suhu lingkungan yang lebih dingin dapat meminimalisir kebutuhan pendinginan mekanis.

Cloud computing juga dapat berkontribusi pada desain berkelanjutan dengan mengoptimalkan penggunaan sumber daya komputasi secara dinamis. Layanan cloud memungkinkan pengguna berbagi infrastruktur sehingga pemanfaatan server menjadi lebih

maksimal dan mengurangi pemborosan energi dibandingkan dengan penggunaan server pribadi yang seringkali idle.

Desain berkelanjutan juga mencakup pengurangan limbah elektronik melalui praktik daur ulang dan perpanjangan umur perangkat. Perusahaan didorong untuk menerapkan kebijakan penggunaan perangkat yang tahan lama, perbaikan dan upgrade komponen, serta pengelolaan limbah yang sesuai standar lingkungan untuk mengurangi dampak negatif terhadap ekosistem.

Dalam proses pengembangan sistem, prinsip green software engineering mulai diterapkan, di mana seluruh siklus hidup perangkat lunak dirancang untuk efisiensi sumber daya dan minim dampak lingkungan. Hal ini melibatkan pemilihan teknologi, bahasa pemrograman, hingga arsitektur aplikasi yang mendukung konsumsi daya rendah dan penggunaan optimal sumber daya.

Penerapan prinsip green IT juga memperhatikan aspek transportasi dan distribusi teknologi. Misalnya, dengan mengurangi frekuensi pengiriman perangkat keras dan memanfaatkan teknologi virtual meeting untuk mengurangi perjalanan bisnis, organisasi dapat mengurangi emisi karbon yang dihasilkan dari aktivitas operasionalnya.

Desain berkelanjutan mendorong integrasi energi terbarukan dalam infrastruktur TI, seperti penggunaan panel surya atau tenaga angin untuk memasok listrik ke data center atau kantor. Dengan demikian, sumber energi yang digunakan menjadi lebih bersih dan mengurangi ketergantungan pada bahan bakar fosil yang berkontribusi terhadap pemanasan global.

Dalam konteks pengembangan aplikasi, desain berkelanjutan juga mengedepankan penggunaan model pengembangan yang agile dan iterative agar dapat dengan cepat menyesuaikan dan mengoptimalkan penggunaan sumber daya sesuai kebutuhan, menghindari pemborosan selama proses produksi dan pemeliharaan aplikasi.

Kesadaran dan pelatihan tentang pentingnya Green IT bagi pengembang dan manajer proyek juga menjadi kunci sukses implementasi desain berkelanjutan. Dengan pemahaman yang baik, keputusan teknis dan strategis dapat dibuat untuk mendukung tujuan lingkungan tanpa mengorbankan performa dan kualitas sistem.

Audit dan pemantauan secara berkala terhadap konsumsi energi dan dampak lingkungan dari sistem TI juga penting dalam memastikan implementasi desain berkelanjutan berjalan efektif. Penggunaan tools monitoring energi dan carbon footprint membantu

organisasi mengidentifikasi area yang perlu ditingkatkan dan membuat laporan keberlanjutan yang transparan.

Desain berkelanjutan dan Green IT merupakan upaya penting dalam menghadapi tantangan perubahan iklim dan keterbatasan sumber daya alam. Dengan menerapkan prinsip-prinsip ini, teknologi informasi tidak hanya menjadi enabler bisnis dan inovasi, tetapi juga kontributor positif terhadap kelestarian lingkungan dan kesejahteraan masyarakat global.

#### **D. Tantangan dan Peluang Masa Depan**

Masa depan pengembangan sistem informasi dan teknologi menghadirkan berbagai tantangan sekaligus peluang yang sangat dinamis. Salah satu tantangan utama adalah kebutuhan akan inovasi yang cepat untuk menyesuaikan dengan perkembangan teknologi yang terus berubah, seperti kecerdasan buatan, Internet of Things (IoT), cloud computing, dan edge computing. Organisasi harus mampu beradaptasi dengan cepat agar tidak tertinggal dalam persaingan global.

Tantangan lain datang dari segi keamanan siber. Dengan semakin kompleksnya sistem dan volume data yang terus meningkat, risiko serangan siber juga menjadi lebih besar dan beragam. Ancaman seperti ransomware, pencurian data, dan serangan DDoS menuntut

pengembangan sistem yang lebih tangguh dan strategi keamanan yang proaktif serta adaptif untuk melindungi aset digital perusahaan.

Selain itu, privasi data menjadi isu kritis di era digital saat ini. Regulasi perlindungan data seperti GDPR di Eropa dan berbagai peraturan lokal di negara lain memaksa organisasi untuk lebih berhati-hati dalam mengelola data pengguna. Sistem harus dirancang dengan prinsip *privacy by design* dan memiliki mekanisme transparansi serta kontrol atas data yang dikumpulkan dan digunakan.

Ketersediaan sumber daya manusia yang kompeten di bidang teknologi juga menjadi tantangan tersendiri. Perkembangan teknologi yang pesat menuntut tenaga ahli yang terus memperbarui pengetahuan dan keterampilannya. Kekurangan SDM yang mumpuni dapat menghambat inovasi dan implementasi solusi teknologi yang efektif.

Namun, di balik tantangan tersebut, terdapat peluang besar yang dapat dimanfaatkan untuk pertumbuhan dan pengembangan. Teknologi kecerdasan buatan dan otomatisasi membuka jalan bagi peningkatan efisiensi operasional dan pengambilan keputusan yang lebih tepat berbasis data. Hal ini memungkinkan perusahaan untuk mempercepat inovasi dan menghadirkan layanan yang lebih personal serta responsif.

Peluang besar juga muncul dari teknologi cloud dan edge computing yang memungkinkan organisasi untuk mengelola infrastruktur TI secara lebih fleksibel dan efisien. Dengan memanfaatkan cloud, bisnis dapat mempercepat deployment aplikasi, menghemat biaya, dan meningkatkan skalabilitas tanpa perlu investasi besar di awal. Edge computing memungkinkan pemrosesan data lebih cepat dan responsif, terutama untuk aplikasi yang memerlukan real-time processing.

Selain itu, perkembangan teknologi 5G membawa peluang baru dalam hal konektivitas yang lebih cepat dan stabil. Ini membuka jalan bagi aplikasi-aplikasi baru yang memanfaatkan bandwidth tinggi dan latency rendah seperti augmented reality, virtual reality, dan smart city. Dengan koneksi yang lebih baik, layanan digital dapat menjangkau lebih banyak pengguna dan wilayah geografis yang sebelumnya sulit terlayani.

Peran data besar (big data) dan analitik juga semakin vital di masa depan. Dengan kemampuan mengumpulkan dan menganalisis data dalam jumlah besar secara real-time, organisasi dapat mengidentifikasi tren pasar, perilaku konsumen, serta risiko bisnis dengan lebih akurat. Ini membuka peluang untuk menciptakan produk dan layanan yang lebih inovatif serta strategi pemasaran yang lebih efektif.

Keterlibatan pengguna (user engagement) dan pengalaman pengguna (user experience) menjadi fokus utama dalam pengembangan sistem masa depan. Sistem yang dapat beradaptasi dengan preferensi pengguna, mudah diakses, dan responsif terhadap kebutuhan, akan menjadi kunci keberhasilan produk digital. Hal ini membuka peluang untuk desain UI/UX yang lebih canggih dan personalisasi layanan berbasis AI.

Sektor industri 4.0 yang menggabungkan teknologi otomatisasi, IoT, AI, dan data analytics juga menjadi peluang besar untuk meningkatkan produktivitas dan efisiensi manufaktur. Integrasi teknologi ini memungkinkan pemantauan mesin secara real-time, prediksi perawatan, dan optimasi rantai pasok, sehingga mempercepat transformasi digital di sektor industri.

Dalam konteks sosial, tantangan digital divide atau kesenjangan akses teknologi masih menjadi perhatian penting. Peluang masa depan ada pada pengembangan solusi yang inklusif dan mampu menjangkau berbagai lapisan masyarakat, termasuk di daerah terpencil. Dengan teknologi yang lebih murah dan mudah diakses, manfaat digitalisasi dapat dirasakan lebih merata.

Lingkungan hidup juga menjadi fokus penting dalam pengembangan teknologi masa depan. Tantangan perubahan iklim menuntut penerapan desain

berkelanjutan dan Green IT untuk mengurangi jejak karbon dan limbah elektronik. Peluang muncul dalam inovasi teknologi hemat energi dan pengelolaan sumber daya yang ramah lingkungan.

Kolaborasi antar berbagai sektor dan disiplin ilmu menjadi kunci dalam menghadapi tantangan dan meraih peluang masa depan. Pendekatan multidisiplin dan kerja sama antara pemerintahan, industri, akademisi, dan masyarakat dapat mempercepat inovasi dan memastikan teknologi yang dikembangkan memberikan manfaat luas dan berkelanjutan.

Di sisi kebijakan, tantangan regulasi dan etika dalam penggunaan teknologi digital juga semakin kompleks. Peluang masa depan terletak pada pengembangan kerangka kerja yang mendukung inovasi sekaligus melindungi hak pengguna dan menjaga keadilan dalam penggunaan teknologi. Regulasi yang adaptif dan berbasis dialog menjadi kunci keberhasilan.

Tantangan dan peluang masa depan di bidang teknologi informasi dan sistem adalah dua sisi yang saling melengkapi. Dengan pendekatan strategis dan inovatif, organisasi dan masyarakat dapat memanfaatkan peluang yang ada sekaligus mengatasi tantangan dengan efektif, sehingga teknologi menjadi kekuatan positif untuk pembangunan dan kemajuan manusia secara global.

# Referensi

- Ahmad, N., Krisnanik, E., Rupilele, F. G. J., Muliawati, A., Syamsiyah, N., Kraugusteeliana, K., Cahyono, B. D., Sriyeni, Y., Kristanto, T., & Irwanto, I. (2022). *Analisa & Perancangan Sistem Informasi Berorientasi Objek*. Penerbit Widina.
- Arief, S. F., & Sugiarti, Y. (2022). Literature review: analisis metode perancangan sistem informasi akademik berbasis web. *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, 8(2), 87–93.
- Arifin, N. Y., Kom, S., Kom, M., Tyas, S. S., Sulistiani, H., Kom, M., Alim Hardiansyah, S. T., Kom, M., Suri, G. P., & Kom, M. (2022). *Analisa Perancangan Sistem Informasi*. Cendikia Mulia Mandiri.
- Azis, N. (2022). *Analisis Perancangan Sistem Informasi*.
- Bagir, H., & Putro, B. E. (2018). Analisis perancangan sistem informasi pergudangan di CV. Karya Nugraha. *Jurnal Media Teknik Dan Sistem Industri*, 2(1), 30–38.
- Dengen, N., & Hatta, H. R. (2009). Perancangan Sistem Informasi Terpadu Pemerintah Daerah Kabupaten Paser. *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 4(1), 47–54.
- Manuhutu, M., & Wattimena, J. (2019). Perancangan sistem informasi konsultasi akademik berbasis website. *Jurnal*

*Sistem Informasi Bisnis*, 9(2), 149.

- Munawaroh, S. (2006). Perancangan sistem informasi persediaan barang. *Dinamik*, 11(2).
- Nopriandi, H. (2018). Perancangan sistem informasi registrasi mahasiswa. *Jurnal Teknologi Dan Open Source*, 1(1), 73–79.
- Nugraha, F. (2014). Analisa dan perancangan sistem informasi perpustakaan. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 5(1), 27–32.
- Nugroho, F. E. (2016). Perancangan Sistem Informasi Penjualan Online Studi Kasus Tokoku. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 7(2), 717–724.
- Nuh, A. (2021). *Perancangan Sistem Informasi Inventory Barang*.
- Nursari, S. R. C., & Immanuel, Y. (2017). Perancangan Sistem Informasi Penjualan Online. *Jurnal TAM (Technology Acceptance Model)*, 8(2), 107–116.
- Rahmawati, N. A., & Bachtiar, A. C. (2018). Analisis dan perancangan sistem informasi perpustakaan sekolah berdasarkan kebutuhan sistem. *Berkala Ilmu Perpustakaan Dan Informasi*, 14(1), 76–86.
- Ria, M. D., & Budiman, A. (2021). Perancangan sistem informasi tata kelola teknologi informasi perpustakaan. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 2(1), 122–133.

- Rohman, A., & Bhakti, H. D. (2023). Perancangan Sistem Informasi Persediaan Barang Berbasis Web. *Syntax Lit.; J. Ilm. Indones*, 7(9), 15304–15313.
- Santi, I. H. (2020). *Analisa perancangan sistem*. Penerbit NEM.
- Setiawan, H., & Khairuzzaman, M. Q. (2017). Perancangan Sistem Informasi Manajemen Proyek: Sistem Informasi Kontraktor. *Jurnal Khatulistiwa Informatika*, 5(2), 103–111.
- Setiyanto, R., Nurmaesah, N., & Rahayu, N. S. A. (2019). Perancangan Sistem Informasi Persediaan Barang Studi Kasus di Vahncollections. *Jurnal Sisfotek Global*, 9(1).
- Zaliluddin, D., & Rohmat, R. (2018). Perancangan Sistem Informasi Penjualan Berbasis Web (Studi Kasus Pada Newbiestore). *INFOTECH Journal*, 4(1).

# Tentang Penulis



Sri Tria Siska, S.Kom., lahir di Batusangkar, 19 April 1992. Penulis menyelesaikan Pendidikan Sarjana S-1 Sistem Informasi di Universitas Putra Indonesia “YPTK” Padang dan pendidikan Megister S-2 Sistem Informasi di Universitas Putra Indonesia “YPTK” . Penulis adalah Dosen Program Studi D-3 Teknik Komputer di STT Payakumbuh. Penulis juga mengabdikan sebagai Tutor Online di Universitas Terbuka pada matakuliah Pengantar Sistem Informasi dan Algoritma dan Pemrograman. Penulis juga mengampu beberapa matakuliah lainnya di STT Payakumbuh yaitu Sistem Operasi, Sistem Basis Data, Teknologi Informasi dan Komunikasi dan Algoritma dan Pemrograman Dasar. Penulis juga menulis beberapa buku di Penamuda Media yaitu Algoritma dan Pemrograman Python, Pengantar Teknologi Informasi dan Sistem Operasi Komputer menjadi editor buku Dasar – Dasar Pemrograman. Penulis juga aktif melakukan penelitian di Bidang Data Mining, Program Aplikasi dan Information Systems. Penulis juga aktif didalam pengelola beberapa Jurnal yang sudah terakreditasi. Penulis dapat dihubungi melalui email [sritriasiska@gmail.com](mailto:sritriasiska@gmail.com).



**Evi Lestari Pratiwi, M.Kom**  
**Dosen Manajemen Informatika**  
**Politeknik Negeri Banjarmasin**

Evi Lestari Pratiwi, M.Kom, adalah dosen tetap di Politeknik Negeri Banjarmasin, Kalimantan Selatan. Penulis menyelesaikan Pendidikan Sarjana (S.Kom) di Universitas Muhammadiyah Malang, Magister Komputer (M.Kom.) di STIMIK AMIKOM Yogyakarta dengan konsentrasi di bidang sistem informasi. Aktif mengajar mata kuliah terkait sistem informasi, sistem operasi, dan pemrograman, penulis juga terlibat dalam berbagai kegiatan penelitian dan pengabdian masyarakat yang berfokus pada pengembangan sistem informasi cerdas dan data analytic.

Dengan pengalaman akademik dan praktis di bidang teknologi informasi, penulis berkomitmen untuk menghasilkan bahan ajar yang mudah dipahami namun tetap mendalam secara konsep. Penulis ikut serta menulis untuk memperkaya literatur lokal dalam bidang Computer Organization dan memberikan referensi yang sesuai dengan konteks pembelajaran di perguruan tinggi vokasi.



**YERA WAHDA WAHDI, M.Kom.**

merupakan pendidik/dosen pada Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Ibnu Sina. Lahir di Padang tanggal 18 Januari 1991. Memulai pendidikan S1- Universitas Putra Indonesia "YPTK" Padang, Sumatera Barat 2008 dan S2- Universitas Putra Indonesia "YPTK" Padang Sumatera Barat 2014. Bidang ilmu Penulis fokus pada Sistem Informasi. Penulis juga aktif dalam menulis buku referensi, buku ajar dan menulis artikel pada jurnal nasional terkreditasi dan jurnal internasional berputasi.

E-mail: [wahdawahdiyera@gmail.com](mailto:wahdawahdiyera@gmail.com)

## **Abrar Hadi, S.Kom., M.Kom**



Lahir di Inderapura Kab. Pesisir Selatan, Prov. Sumatera Barat. Sekolah Dasar (SD) dan SMP di selesaikan pada tanah kelahiran, SMK di selesaikan di SMK N 2 Kota Padang, dan Kemudian Menyelesaikan Strata 1 bidang Komputer sains pada tahun 2013 dan Megister Komputer sains pada tahun 2015. Setelah menyelsesaikan Pendidikan formal penulis menjadi dosen pertama kali di STMIK-Amik Riau dari tahun 2015 sampai 2017, kemudian melanjutkan karir dosen di AMIK Datuk Parpatiah Nan Sabatang dan sekarang sudah berganti nama menjadi Politeknik Lembaga Pendidikan dan Pengembangan Profesi Indonesia (LP3I) dari tahun 2017 sampai sekarang, dan untuk mewujudkan karir sebagai dosen professional, penulis pun aktif dalam pengembangan tri dharma perguruan tinggi, terus berupaya untuk memberikan manfaat kepada masyarakat luas melalui bidang ilmu komputer. Penulis telah membuat beberapa Buku Referensi dan book chapter yang ber-ISBN. Penulis sekarang menjabat sebagai Ketua Program Studi Manajemen Informatika Kampus Kota Padang. Telah berkontribusi dalam penelitian dan penerbitan jurnal Terakreditasi secara Nasional dan Internasional.

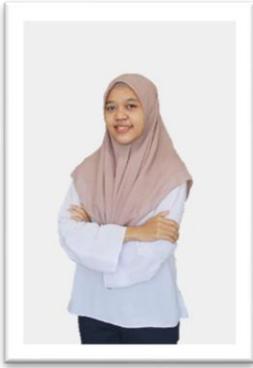


Firdaus, S.Kom., M.Kom lahir di Kota Pariaman pada tanggal 26 Mei 1991. Setelah lulus dari SMK Negeri 3 Pariaman, pada tahun 2009 penulis merantau ke Kota Padang. Pada tahun 2013 dan 2015 penulis menyelesaikan pendidikan Sarjana Komputer (S.Kom) dan Magister Komputer (M.Kom) keduanya di Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK” Padang.

Karir akademik penulis dimulai sebagai dosen STMIK-AMIK Riau pada periode 2015-2017. Kemudian melanjutkan pengabdian di AMIK Bukittinggi dari tahun 2018 hingga 2022, dan sejak tahun 2023 aktif mengajar di Universitas Islam Sumatera Barat. Bidang minat yang ditekuni adalah Decision Support System, Strategic Information System, Applied Database dan Business Intelligence. Sepanjang berkarier di dunia akademik, penulis aktif dalam pelaksanaan tri dharma perguruan tinggi, penulis terus berkarya dan berinovasi demi memberikan kontribusi nyata dalam pengembangan pendidikan di Indonesia, khususnya di bidang ilmu komputer. Penulis juga terlibat dalam penulisan Buku Ajar, Buku Referensi, *Book Chapter*, serta aktif melakukan penelitian dan publikasi di jurnal Nasional terakreditasi dan jurnal Internasional berreputasi. Penulis saat ini menjabat sebagai Ketua Lembaga Penjaminan Mutu di Universitas Islam Sumatera Barat dan sebelumnya pernah menjabat sebagai Direktur AMIK Bukittinggi.



**Kusuma Hati.** Lahir pada bulan Maret 1974, Lulus Program Strata Satu (S1) Jurusan Manajemen Informatika, Universitas Gunadarma Jakarta pada Tahun 1997. Tahun 2006 lulus Program Pasca Sarjana Magister Manajemen, Universitas Budi Luhur. Tahun 2015 Lulus Program Pasca Sarjana Magister Komputer, STMIK Nusa Mandiri Jakarta. Saat ini aktif sebagai Dosen Tetap Program Studi Sistem Informasi di STMIK Antar Bangsa, Tangerang. Aktif juga melakukan penelitian dan mempublikasikan artikel ilmiah pada jurnal nasional dan Internasional serta membuat buku (*book chapter*) bersama rekan-rekan Dosen dari Perguruan Tinggi lain. Artikel ilmiah yang pernah dipublikasikan diantaranya adalah “Sistem Rekomendasi Produk Skin Care, “Digital Ethics Mahasiswa Perguruan Tinggi Swasta di Jakarta Menggunakan Framework Digital Literacy KOMINFO (e-jurnal Bina Insani, 2023), “Sistem Rekomendasi Produk Skin Care Berdasarkan Permasalahan Kulit Wajah dengan Metode Content Based Filtering (e-journal STMIK Antar Bangsa, 2024)”, The Impact of the COVID-19 Pandemic on Land Surface Temperature Change through Remote Sensing Data Processing (International Conference on Computing Engineering and Design / ICCED, 2023)”. Buku yang pernah ditulis bersama rekan-rekan Dosen yang lain adalah *Decision Support System (Simple Multi Attribute Rating Technique/SMART)*, *Indonesia Maju dan Bangkit (Indonesia Maju Dan Bangkit Melalui Pendidikan Berbasis IT)*, *Pemahaman Dasar Ms.Word (Mail Merge dan Penggunaan Surat Massal)*.



Andi Jamiati Paramita lahir di Kendari, pada tanggal 31 Januari 1993. Penulis menempuh pendidikan S1 di Universitas Hasanuddin jurusan Teknik Informatika tahun 2011-2015. Kemudian penulis melanjutkan Pendidikan S2 di jurusan yang sama di Institut Teknologi Bandung pada tahun 2015-2018. Bidang keilmuan yang ditekuni penulis saat ini adalah Rekayasa Perangkat Lunak, dan pengembangan sistem berbasis user-centered design.

Saat ini penulis berkarir sebagai dosen di Program Studi Sistem Informasi, Institut Teknologi dan Bisnis Kalla, di Makassar, Sulawesi Selatan. Selain itu, penulis juga aktif sebagai trainer pelatihan digitalisasi bagi pelaku UMKM yang diselenggarakan oleh Komdigi.

Selama berkarir sebagai akademisi, penulis aktif melakukan penelitian yang dipublikasikan pada beberapa jurnal nasional dan berhasil mendapatkan hibah dari Kementerian. Beberapa penelitian yang pernah dilakukan meliputi pengembangan user experience website rumah sakit yang ditujukan bagi pengguna disabilitas. Selain itu, topik penelitian penulis fokus pada pengembangan aplikasi berbasis website dengan berbagai metode dan framework.



Nama : Putri Ariatna Alia  
Tanggal Lahir : Surabaya, 5 Juli 1994  
Pendidikan Terakhir : S2 Teknik Elektro

**PERANCANGAN**

# **SISTEM INFORMASI DAN APLIKASINYA**

Buku Perancangan Sistem Informasi dan Aplikasinya menyajikan panduan lengkap dan sistematis mengenai proses merancang sistem informasi yang efektif dan aplikatif, mulai dari analisis kebutuhan, desain arsitektur, pengembangan aplikasi, hingga tahap implementasi, pengujian, dan pemeliharaan. Dengan pendekatan teoritis yang diperkuat oleh studi kasus nyata dan penggunaan teknologi terkini, buku ini memberikan pemahaman mendalam tentang bagaimana merancang sistem informasi yang sesuai dengan kebutuhan bisnis dan pengguna. Buku ini sangat relevan bagi mahasiswa, dosen, dan praktisi teknologi informasi yang ingin menguasai proses perancangan sistem secara menyeluruh dan profesional di era digital yang terus berkembang.

ISBN 978-634-7267-42-3



9

786347

269423



**Penamuda.com**

PT Penamuda Media  
Casa Sidharam, Ngentak Godaan  
penamuda\_media