



INSTITUT TEKNOLOGI INDONESIA

**RANCANG BANGUN PLATFORM LINK MANAGEMENT SYSTEM DAN
UTILITAS ONLINE UNTUK INSTITUT TEKNOLOGI INDONESIA**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana komputer

JONATHAN NATANNAEL ZEFANYA

1152200024

TEKNIK INFORMATIKA

TANGERANG SELATAN

2025

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Jonathan Natannael Zefanya

NPM : 1152200024

Tanda Tangan :

Tanggal : 01 Oktober 2025

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Jonathan Natanael Zefanya
NPM : 1152200024
Program Studi : Teknik Informatika
Judul Skripsi : RANCANG BANGUN PLATFORM LINK
MANAGEMENT SYSTEM DAN UTILITAS
ONLINE UNTUK INSTITUT TEKNOLOGI
INDONESIA

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer Pada Program Studi Teknik Informatika Institut Teknologi Indonesia.

DEWAN PENGUJI

Pembimbing : Ir. Muhamad Ramli, M.Kom ()
Penguji 1 : ()
Penguji 2 : ()
Penguji 3 : ()

Ditetapkan di : Kampus Institut Teknologi Indonesia, Tangerang Selatan

Tanggal : 2025

KETUA PROGRAM STUDI TEKNIK INFORMATIKA

(Muhamad Soleh, S,Si, M.Kom)

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul " Rancang Bangun Platform Link Management System dan Utilitas Online untuk Institut Teknologi Indonesia" ini dengan baik. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Institut Teknologi Indonesia.

Penulis menyadari bahwa terselesaikannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Muhamad Ramli, M.Kom, selaku Dosen Pembimbing yang dengan sabar dan penuh perhatian telah menyediakan waktu, tenaga, serta pemikirannya untuk membantu penulis menyusun Tugas Akhir ini. Kritik dan masukan yang beliau berikan menjadi bahan refleksi yang sangat berharga dalam meningkatkan kualitas tulisan ini.
2. Bapak Muhamad Soleh, S.Si, M.Kom, selaku Ketua Program Studi Teknik Informatika yang telah memberikan arahan, dukungan administratif, serta semangat akademik kepada penulis selama menempuh pendidikan, khususnya dalam proses penyusunan Tugas Akhir ini. Bimbingan dan arahnya menjadi fondasi yang sangat berarti bagi penulis dalam menyelesaikan karya ini.
3. Bapak Sunarto, selaku Dosen Penasehat Akademik yang telah membimbing dan mendampingi penulis sejak awal perkuliahan. Nasihat dan arahan beliau dalam menghadapi berbagai dinamika akademik sangat membantu penulis untuk terus berkembang dan menyelesaikan studi dengan lebih terarah.
4. Kedua orang tua penulis, yang selalu menjadi sumber kekuatan dan motivasi utama dalam menjalani kehidupan akademik dan pribadi. Doa yang tidak pernah putus, kasih sayang yang tulus, serta dukungan moril dan materil yang diberikan menjadi pilar yang menguatkan penulis hingga sampai pada titik ini. Penulis sangat bersyukur memiliki orang tua yang luar biasa.

5. Bapak Sandhika Galih, atas kontribusi dan dedikasinya dalam berbagi ilmu melalui platform YouTube, yang telah menjadi sumber pembelajaran berharga bagi penulis selama proses penyusunan skripsi ini.
6. Teman-teman di server *Discord* IMPHEN dan WPU, yang tidak hanya menjadi partner belajar tetapi juga teman berbagi cerita dan tawa di tengah tekanan tugas dan ujian. Kebersamaan dan kekompakan kalian menjadi bagian tak terlupakan dari perjalanan ini.
7. Seluruh teman seangkatan, yang telah melalui banyak hal bersama, dari tugas kelompok, diskusi panjang, hingga berbagai momen perjuangan di ruang kelas. Semangat kebersamaan dan saling mendukung menjadi kenangan berharga yang akan selalu dikenang.

Oleh karena itu, saran dan kritik yang membangun sangat diharapkan guna perbaikan di masa mendatang. Semoga Tugas Akhir ini dapat memberikan manfaat bagi penulis khususnya, serta pembaca pada umumnya. Akhir kata, penulis mengucapkan terima kasih dan semoga Tuhan Yang Maha Esa selalu melimpahkan berkah dan rahmat-Nya kepada semua pihak yang telah membantu.

Tangerang Selatan, 11 Februari 2026

Jonathan Natannael Zefanya

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR /
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Institut Teknologi Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Jonathan Natanael Zefanya

NPM : 1152200014

Program Studi : Teknik Informatika

Jenis karya : Tugas Akhir/Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

**RANCANG BANGUN PLATFORM LINK MANAGEMENT SYSTEM DAN
UTILITAS ONLINE UNTUK INSTITUT TEKNOLOGI INDONESIA**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan Tugas Akhir/Skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Tangerang Selatan
Pada Tanggal 11 Februari 2026
Yang Menyatakan,

Jonathan Natanael Zefanya

ABSTRAK

Nama : Jonathan Natannael Zefanya
Program Studi : Teknik Informatika
Judul : Rancang Bangun Platform Link Management System dan Utilitas Online Untuk Institut Teknologi Indonesia
Dosen Pembimbing : Ir. Muhamad Ramli, M.Kom

Lorem ipsum dolor sit amet consectetur adipiscing elit velit ac ornare pharetra tortor odio, scelerisque ullamcorper sagittis donec lobortis iaculis vestibulum ad fringilla sapien justo. Non lobortis inceptos suspendisse blandit class nascetur torquent suscipit consequat facilisi, efficitur nulla purus scelerisque dictum per nullam massa sollicitudin. Dui quam ad erat magnis mauris ultricies at nec vehicula et taciti dis platea sociosqu, euismod consequat curabitur ante aenean dictumst fermentum leo quis dolor velit nullam proin. Lacus sollicitudin montes maecenas commodo euismod tellus aptent quis pellentesque leo, orci eleifend nec sagittis malesuada feugiat tortor cursus rhoncus. Curabitur ridiculus platea odio conubia euismod senectus etiam venenatis felis, fermentum quis in tempor accumsan cubilia fusce nostra cras neque, malesuada pharetra maximus habitasse magna molestie nulla suscipit. Semper nec donec potenti ipsum nam mattis tristique, in et velit mauris vestibulum integer litora nibh, nostra placerat facilisi tortor proin orci. Aptent elementum faucibus consequat nullam bibendum dolor nunc mattis, fames dignissim fermentum odio vitae sapien class.

Kata kunci: **Kode QR, Pemendek URL, *biolink*, File Transfer**

ABSTRACT

Lorem ipsum dolor sit amet consectetur adipiscing elit velit ac ornare pharetra tortor odio, scelerisque ullamcorper sagittis donec lobortis iaculis vestibulum ad fringilla sapien justo. Non lobortis inceptos suspendisse blandit class nascetur torquent suscipit consequat facilisi, efficitur nulla purus scelerisque dictum per nullam massa sollicitudin. Dui quam ad erat magnis mauris ultricies at nec vehicula et taciti dis platea sociosqu, euismod consequat curabitur ante aenean dictumst fermentum leo quis dolor velit nullam proin. Lacus sollicitudin montes maecenas commodo euismod tellus aptent quis pellentesque leo, orci eleifend nec sagittis malesuada feugiat tortor cursus rhoncus. Curabitur ridiculus platea odio conubia euismod senectus etiam venenatis felis, fermentum quis in tempor accumsan cubilia fusce nostra cras neque, malesuada pharetra maximus habitasse magna molestie nulla suscipit. Semper nec donec potenti ipsum nam mattis tristique, in et velit mauris vestibulum integer litora nibh, nostra placerat facilisi tortor proin orci. Aptent elementum faucibus consequat nullam bibendum dolor nunc mattis, fames dignissim fermentum odio vitae sapien class.

Keywords: *QR Code, Short URL, biolink , File Transfer*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR / SKRIPSI UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK.....	vi
DAFTAR ISI	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. State of the Art.....	4
1.6. Sistematika Penulisan.....	6
BAB 2 TINJAUAN PUSTAKA.....	7
2.1. Link Management System	7
2.1.1. Short URL	7
2.1.2. Microsite.....	8
2.1.3. File Transfer.....	9
2.2. Kode QR (Quick Response Code).....	10
2.3. Utilitas	12
2.4. Keamanan Sistem Informasi (<i>Multi Factor Authentication</i> – MFA).....	13
2.4.1. Klasifikasi Faktor Autentikasi dalam MFA	13
2.4.2. Jenis-Jenis Implementasi MFA.....	14
2.5. Application Programming Interface (API).....	15
2.6. Model-View-Controller (MVC)	18
2.6.1. PHP.....	19
2.6.2. Bootstrap.....	20

2.6.3.	MySQL	21
2.7.	Metode Rapid Application Development	22
2.8.	UML	22
2.9.	ERD	22
2.10.	UI/UX	22
2.11.	BlackBox Testing.....	22
2.12.	User Acceptance Testing (UAT)	22
BAB 3 METODE PENELITIAN		23
3.1.	Tahapan Penelitian.....	23
3.2.	Alur dan Metode Penelitian.....	24
3.3.	Tahap Perencanaan Kebutuhan.....	25
3.3.1.	<i>User Requirements</i>	26
3.3.1.1.	Kebutuhan Fungsional (<i>Functional Requirements</i>):	26
3.3.1.2.	Kebutuhan Non-Fungsional (<i>Non-Functional Requirements</i>):	26
3.3.2.	Analisis Proses.....	27
3.3.3.	Bisnis Identifikasi Aktor dan <i>Stakeholder</i>	29
3.4.	Desain Sistem	30
3.4.1.	Proses Bisnis.....	31
3.4.2.	UI dan UX	33
3.4.3.	<i>Entity Relation Diagram</i> (ERD)	46
3.4.4.	<i>Unified Modeling Language</i> (UML)	48
3.4.4.1.	<i>Use Case Diagram</i> Sistem	48
3.4.4.2.	Class Diagram	55
3.4.4.3.	Sequence Diagram	55
3.5.	Pengembangan Sistem.....	59
3.5.1.	Model Pengembangan: Model-View-Controller (MVC)	60
3.5.2.	Bahasa Pemrograman dan Teknologi Pendukung	60
3.6.	Pengujian dan Penyelesaian.....	61
BAB 4 PEMBAHASAN		62
4.1.	Pengaturan Sistem	62
4.2.	Tampilan Halaman.....	62
4.3.	Pengujian BlackBox	62

4.4.	UAT	62
BAB 5 KESIMPULAN DAN SARAN		63
5.1.	Kesimpulan.....	63
5.2.	Saran	63
DAFTAR REFERENSI (JANGAN LUPA FORMAT APA & CTRL + T)		64
LAMPIRAN		69

DAFTAR GAMBAR

Gambar 2. 1. Short Link	7
Gambar 2. 2. Microsite	8
Gambar 2. 3. File Transfer	9
Gambar 2. 4. Kode QR	10
Gambar 2. 5. Multi Factor Authentication	13
Gambar 2. 6. Application Programming Interface	15
Gambar 2. 7. Model-View-Controller (MVC)	18
Gambar 2. 8. PHP	19
Gambar 2. 9. Bootstrap	20
Gambar 2. 10. MySQL	21
Gambar 3. 1. Tahapan Penelitian	24
Gambar 3. 2. Proses Bisnis	31
Gambar 3. 3. Halaman Login	34
Gambar 3. 4. Halaman Dashboard	35
Gambar 3. 5. Halaman Link	36
Gambar 3. 6. Halaman Statistik	37
Gambar 3. 7. Halaman Buat Biolink	38
Gambar 3. 8. Halaman biolink	38
Gambar 3. 9. Halaman Blok Biolink	39
Gambar 3. 10. Halaman Buat Link Pendek	40
Gambar 3. 11. Halaman Link Pendek	41
Gambar 3. 12. Halaman buat file link	42
Gambar 3. 13. Halaman file link	43
Gambar 3. 14. Halaman List Qr Code	44
Gambar 3. 15. Halaman QR Code	45
Gambar 3. 16. Halaman Tools Online	46
Gambar 3. 17. Entity Relationship Diagram	47
Gambar 3. 18 Use Case Diagram User	49
Gambar 3. 19. Use Case Diagram Admin	52
Gambar 3. 20. Class Diagram	55
Gambar 3. 21. Sequence Diagram User	56
Gambar 3. 22. Sequence Diagram Admin	58

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Di era digital, efektivitas komunikasi dan produktivitas institusi pendidikan sangat bergantung pada kemampuan dalam memanfaatkan *tools online* yang efisien dan terintegrasi. Perguruan tinggi saat ini tidak hanya dituntut untuk memiliki situs web resmi, tetapi juga berbagai layanan digital pendukung seperti sistem manajemen link, pembuatan microsite, konversi dokumen, hingga generator kode QR untuk mendukung kegiatan akademik, promosi, dan penelitian. Beragam aktivitas di lingkungan Institut Teknologi Indonesia (ITI) seperti publikasi penelitian, penyebaran informasi seminar, hingga kegiatan promosi fakultas memerlukan platform digital yang dapat menyederhanakan pengelolaan dan distribusi tautan serta alat bantu daring lainnya secara terpusat.

Sejumlah platform populer seperti Bit.ly, Linktree, dan S.id telah banyak digunakan secara global karena kemampuannya dalam menyediakan layanan pemendek URL, agregator tautan, serta analitik performa pengguna. Bit.ly, misalnya, mencatat lebih dari 600 juta tautan (Bitlink) dibuat setiap bulan dengan 5,7 juta pengguna aktif global (Bitly, 2022). Sementara itu, Linktree digunakan oleh lebih dari 40 juta pengguna di seluruh dunia (Linktree, 2023), dan S.id di Indonesia telah mencapai 1,5 juta pengguna aktif serta menghasilkan lebih dari 15 juta tautan pendek pada 2025 (Langgam.id, 2025). Pertumbuhan industri ini pun signifikan, dengan nilai pasar layanan link-in-bio dan pemendek URL diperkirakan mencapai USD 1,62 miliar pada 2024 dengan CAGR 13,2% (DataIntel, 2024).

Penerapan serupa juga dilakukan oleh lembaga pemerintah seperti s.komdigi.go.id, yang menunjukkan pentingnya manajemen tautan dan sistem digital terpadu bagi institusi publik. Bagi Institut Teknologi Indonesia (ITI), pengembangan platform internal semacam ini misalnya layanan shortlink kampus seperti s.iti.ac.id (atau nama lain yang serupa) dapat menjadi langkah strategis untuk memperkuat digital branding kampus, meningkatkan brand awareness, dan mendukung kemandirian teknologi institusi. Selain itu, pengembangan sistem yang terintegrasi dengan berbagai online tools seperti

generator QR code, converter file, dan microsite builder dapat membantu sivitas akademika ITI menjalankan aktivitas digital dengan lebih efisien dan aman.

Saat ini, penggunaan layanan pihak ketiga yang berbeda-beda di lingkungan kampus sering kali menimbulkan masalah, seperti data analitik yang terpisah, pengelolaan hak akses yang tidak konsisten, serta kesulitan dalam memantau efektivitas komunikasi digital secara menyeluruh. Penelitian menunjukkan bahwa strategi komunikasi universitas yang tidak terorganisir dapat mengurangi efektivitas kehadiran institusional di dunia maya (Capriotti et al., 2023). Selain itu, keterbatasan akses API, ketiadaan autentikasi dua faktor (2FA) pada layanan gratis (Reynolds et al., 2020), serta ancaman keamanan dari layanan pemendek URL berbasis iklan yang berpotensi menimbulkan serangan malware atau phishing (Fukushi et al., 2021; Fukushi et al., 2022) menjadi alasan kuat perlunya solusi internal yang aman dan terpusat.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang menyoroti permasalahan fragmentasi data, keterbatasan fitur, serta inefisiensi pengelolaan pada berbagai platform pihak ketiga di lingkungan Institut Teknologi Indonesia (ITI), maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun platform terpusat yang mengintegrasikan fitur peningkat tautan, halaman biolink, manajemen berkas, generator kode QR serta berbagai *tools online* guna mengatasi penggunaan layanan terpisah di ITI?
2. Bagaimana mengimplementasikan sistem manajemen tim berbasis peran (*role-based access control*) dengan dukungan autentikasi dua faktor (2FA) untuk meningkatkan keamanan dan efektivitas kolaborasi antar unit kerja di ITI?
3. Bagaimana menyajikan data analitik terintegrasi dari seluruh aset digital dalam satu dasbor terpadu sebagai dukungan pengambilan keputusan strategis berbasis data bagi manajemen ITI?
4. Bagaimana mengembangkan *Application Programming Interface* (API) yang memungkinkan integrasi dengan sistem informasi internal ITI serta mendukung skalabilitas dan pengembangan di masa depan?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menghasilkan solusi yang efektif dalam pengelolaan link di lingkungan Institut Teknologi Indonesia (ITI) melalui hal-hal berikut:

1. Merancang dan membangun arsitektur sistem web terintegrasi yang menggabungkan fungsionalitas manajemen tautan, halaman biolink, berbagi berkas, dan kode QR dalam satu platform.
2. Menghadirkan fitur kustomisasi tingkat lanjut bagi pengguna, seperti personalisasi tampilan profil dan pembuatan kode QR dinamis yang dapat diubah arah tautannya.
3. Mengimplementasikan sistem manajemen tim kolaboratif dengan kontrol hak akses berbasis peran serta dasbor aktivitas bersama untuk meningkatkan koordinasi dan keamanan pengguna.
4. Mengembangkan API yang terdokumentasi dengan baik sebagai fondasi bagi integrasi dan pengembangan berkelanjutan dengan sistem maupun layanan eksternal.

1.4. Batasan Masalah

Agar penelitian ini tetap fokus dan terarah, beberapa batasan masalah yang ditetapkan adalah sebagai berikut:

1. Platform berbasis web, Pengembangan sistem hanya difokuskan pada versi web dan tidak mencakup aplikasi mobile (Android/iOS).
2. Batasan kuota penyimpanan, Setiap pengguna akan memiliki batas maksimum penyimpanan berkas (*storage quota*) yang ditentukan untuk menjaga efisiensi sumber daya server.
3. Model tingkatan keanggotaan, Platform akan memiliki kustomise tingkatan keanggotaan sesuai dengan kebutuhan yang ada.
4. Penelitian ini dibatasi pada pengembangan fitur *URL shortener* (pemendek tautan), *file URL*, *biolink URL*, dan *tools online* berbasis *client-side*. Cakupan penelitian ini juga mencakup pengujian fitur-fitur tersebut yang berkaitan dengan kustomisasi tingkatan keanggotaan.
5. Sistem pembayaran di luar cakupan Implementasi sistem gerbang pembayaran (*payment gateway*) untuk mengatur proses langganan tidak termasuk dalam ruang lingkup penelitian ini.

1.5. State of the Art

Tabel 1.1 State Of Art

Judul Jurnal	Pembahasan
<p>Rancang bangun tools URL shortener berbasis website menggunakan framework Laravel (Studi kasus Diskominfo Kalimantan Barat)</p> <p>Peneliti: Adhitama, R. & Jauhari, A.</p> <p>Lokasi: Diskominfo Kalimantan Barat</p> <p>Tahun 2022</p> <p>Nama Jurnal Jurnal Teknologi dan Sistem Informasi</p>	<p>Hasil Penelitian:</p> <p>Mengembangkan aplikasi pemendek URL berbasis web menggunakan framework Laravel dengan fokus pada kebutuhan instansi pemerintahan</p> <p>Alasan Menjadi Tinjauan Penelitian:</p> <p>Memberikan contoh implementasi praktis aplikasi pemendek URL berbasis instansi, yang relevan untuk pengembangan sistem terpusat di ITI.</p>
<p>Rancang bangun aplikasi Lihat.in berbasis web di Fakultas Ilmu Komputer UPN Veteran Jawa Timur dengan algoritma Base64 serta implementasi metode QR-Code</p> <p>Peneliti: Andriani, D. A. & Prasetyo, H.</p> <p>Lokasi: Fakultas Ilmu Komputer UPN Veteran Jawa Timur</p> <p>Tahun 2022</p> <p>Nama Jurnal Jurnal Ilmu Komputer dan Teknologi Informasi</p>	<p>Hasil Penelitian:</p> <p>Mengembangkan aplikasi berbasis web untuk pemendekan tautan dengan tambahan fitur QR-Code.</p> <p>Alasan Menjadi Tinjauan Penelitian:</p> <p>Relevan untuk pengembangan fitur generator QR dinamis pada platform.</p>
<p>Aplikasi sistem informasi pemendek URL (SI SOUP) berbasis web</p>	<p>Hasil Penelitian:</p>

<p>Peneliti: Hidayat, R. & Sari, P.</p> <p>Lokasi: Universitas Bina Sarana Informatika</p> <p>Tahun 2020</p> <p>Nama Jurnal Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)</p>	<p>Membangun sistem pemendek URL dengan fokus pada efisiensi akses tautan.</p> <p>Alasan Menjadi Tinjauan Penelitian: Menjadi referensi teknis untuk membandingkan pendekatan arsitektur aplikasi pemendek URL.</p>
<p>Perancangan prototipe UI/UX pada pemesanan menu di kedai kopi Rumiko berbasis <i>mobile microsite</i></p> <p>Peneliti Permana, I. M. & Lestari, D.</p> <p>Lokasi Tangerang, Indonesia</p> <p>Tahun 2023</p> <p>Nama Jurnal Jurnal Wanarupa: Desain, Komunikasi Visual, dan Media</p>	<p>Hasil Penelitian: Membahas desain UI/UX berbasis microsite untuk meningkatkan pengalaman pengguna.</p> <p>Alasan Menjadi Tinjauan Penelitian: Relevan untuk perancangan antarmuka platform yang user-friendly dan intuitif.</p>
<p>Design and Implementation of Toolbox Based on Component Technology</p> <p>Peneliti: Q. Xingguo dkk.</p> <p>Lokasi: University of Science and Technology, College of Computer Science and Technology, Shanxi, China</p> <p>Tahun 2020</p> <p>Nama Jurnal</p>	<p>Hasil Penelitian: Penelitian ini Menjelaskan arsitektur toolbox berbasis teknologi komponen: bagaimana modul-modul fungsi dipaketkan sebagai komponen yang dapat dipanggil/dipasang secara dinamis sehingga memudahkan perpanjangan dan maintenance.</p> <p>Alasan Menjadi Tinjauan Penelitian:</p>

ICCEA / IEEE Computer Society proceedings entry	Jurnal ini relevan untuk desain platform all-in-one yang memerlukan plugin/komponen terpisah untuk kategori tools (text, generator, developer, dll).
--	---

1.6. Sistematika Penulisan

Laporan Tugas Akhir ini terdiri dari rangkaian sistematika penulisan sebagai berikut:

1) BAB 1 Pendahuluan

Bab ini berisi latar belakang masalah yang menjelaskan alasan perlunya sistem manajemen tautan, penelitian terdahulu, serta teknologi yang digunakan dalam penelitian. Selain itu, bab ini juga memuat rumusan masalah, batasan masalah, dan *State of the Art* yang mendasari pengembangan sistem.

2) BAB 2 Tinjauan Pustaka

Bab ini membahas landasan teori dan konsep-konsep relevan yang mendukung penelitian, guna memperkuat dasar argumentatif dan teoritis dalam pengembangan sistem serta penulisan laporan.

3) BAB 3 Metode

Bab ini menjelaskan metode penelitian yang digunakan, meliputi tahapan, pendekatan, serta teknik yang diterapkan dalam proses perancangan dan pengembangan sistem.

4) BAB 4 Pembahasan

Bab ini menguraikan tahapan pembangunan platform sistem manajemen tautan, mulai dari perancangan, implementasi, hingga uji coba penggunaan, serta analisis hasil pengujian terhadap sistem yang telah dikembangkan.

5) BAB 5 Kesimpulan dan Saran

Bab ini memuat kesimpulan dari hasil penelitian yang telah dilakukan serta saran untuk pengembangan dan penelitian selanjutnya.

BAB 2 TINJAUAN PUSTAKA

2.1. Link Management System

Link Management System (LMS) adalah sistem yang digunakan untuk mengelola, memonitor, dan mengoptimalkan tautan digital secara efisien. Sistem ini membantu pengguna dalam membuat, memperbarui, melacak, serta menganalisis performa tautan tanpa perlu mengubah tautan di berbagai tempat. Menurut (Zhang, Liu, & Cheng, 2024), pengelolaan tautan modern harus memperhatikan keamanan dan integritas data karena banyak tautan pendek yang berisiko disalahgunakan untuk *phishing* atau distribusi *malware*.

2.1.1. Short URL



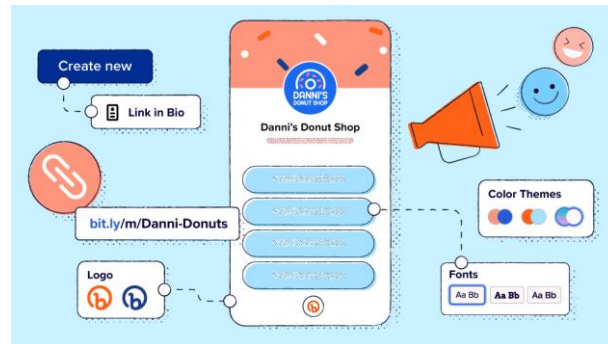
Gambar 2. 1. Short Link

Short URL adalah teknologi yang digunakan untuk mengonversi alamat situs web (*Uniform Resource Locator*) yang panjang menjadi bentuk yang lebih pendek dan mudah dibagikan. Tujuan utama penggunaan *short URL* adalah untuk meningkatkan efisiensi penyebaran tautan, terutama dalam media sosial, kampanye digital, dan sistem manajemen tautan.

Menurut (Prajapati dan Jain ,2025) dalam jurnal “*URL Shortener Using Python Flask Framework*”, sistem *short URL* bekerja dengan cara menghasilkan kode unik yang mewakili URL asli, kemudian menyimpannya dalam basis data yang dapat diakses kembali untuk melakukan pengalihan (*redirect*) ke alamat tujuan. Penelitian ini menjelaskan bahwa penerapan *Python Flask Framework* memungkinkan pengembangan

aplikasi pemendek URL yang ringan, cepat, dan mudah diintegrasikan dengan berbagai sistem web.

2.1.2. Microsite



Gambar 2. 2. Microsite

Biolink atau microsite merupakan halaman web tunggal yang berfungsi sebagai pusat tautan menuju berbagai platform atau sumber informasi lainnya. Konsep ini banyak digunakan dalam konteks *personal branding*, pemasaran digital, dan sistem pelayanan publik berbasis daring karena memungkinkan pengguna mengintegrasikan beberapa tautan dalam satu halaman yang ringkas dan interaktif.

Menurut (Rif'ah, Nurcholis, dan Mulyadi, 2024) dalam jurnal *Jurnal Inovasi Penelitian, Teknologi Informasi dan Komunikasi (JIPTIKA)*, pengembangan microsite dapat membantu pengguna dalam mengelola berbagai tautan dari media sosial dan situs web utama secara efisien. Dalam penelitian berjudul “*Perancangan Aplikasi Microsite Untuk Mengelola Link Pada Media Sosial Menggunakan Framework Laravel*”, dijelaskan bahwa microsite memberikan kemudahan bagi pengguna untuk menghubungkan seluruh tautan penting mereka, seperti portofolio, kontak bisnis, maupun konten promosi dalam satu halaman web responsif.

Sementara itu, menurut (Sulistyo, 2023) dalam bukunya *Integrasi Microsite dalam Sistem Pelayanan Digital Publik*, microsite memiliki peran strategis dalam meningkatkan keterhubungan antar-layanan di sektor publik maupun komersial. Microsite berfungsi sebagai “jembatan digital” yang mengintegrasikan berbagai sistem informasi menjadi

satu titik akses terpadu. Melalui pendekatan ini, organisasi dapat memberikan pengalaman pengguna yang lebih baik dan mempercepat proses akses informasi.

Sulistyo juga menjelaskan bahwa integrasi microsite dalam sistem digital modern mendukung konsep *digital government* dan *smart services*, di mana akses layanan publik dilakukan melalui portal microsite yang menghubungkan berbagai instansi dan layanan terkait. Hal ini menunjukkan bahwa konsep microsite tidak hanya relevan dalam ranah pemasaran digital, tetapi juga penting dalam transformasi layanan publik berbasis teknologi informasi.

Dengan demikian, dapat disimpulkan bahwa biolink atau microsite merupakan komponen penting dalam strategi komunikasi digital modern. Microsite berperan sebagai pusat distribusi tautan yang mengoptimalkan akses informasi, baik dalam konteks personal branding, pemasaran, maupun layanan publik digital.

2.1.3. File Transfer



Gambar 2. 3. File Transfer

Transfer file, secara definitif, adalah proses transmisi file digital dari satu komputer atau sistem ke sistem lain melalui koneksi jaringan atau internet (IBM, 2024). Proses ini diatur oleh seperangkat aturan yang disebut protokol komunikasi, yang memastikan bahwa berbagai sistem dapat "berbicara" dalam bahasa yang sama untuk bertukar data secara andal dan aman (Webasha, 2025). Fungsi esensial dari *transfer file* adalah memfasilitasi pertukaran informasi dan kolaborasi yang efisien, memungkinkan pergerakan data yang aman antar perangkat, lokasi, maupun organisasi (IBM, 2024).

Isu performa dan keandalan juga menjadi fokus penelitian terkini. Untuk file berukuran besar dan jaringan tidak stabil, teknik seperti kompresi, fragmen/segmentasi, *retry/ resume*, serta *reassembly* terbukti meningkatkan *throughput* dan *resilien* transfer.

Selain itu, pada konteks *cloud* dan *edge computing*, pengelolaan kunci enkripsi, kontrol akses berbasis atribut, serta penggunaan enkripsi *end-to-end* adalah rekomendasi untuk mengurangi permukaan serangan dan risiko kebocoran data. Studi terbaru juga menunjukkan tren pergeseran dari hanya mengandalkan protokol tradisional ke penerapan mekanisme *policy-driven* dan arsitektur enkripsi yang terintegrasi dalam ekosistem transfer data (Usman, 2024).

Berdasarkan tinjauan literatur, praktik terbaik (*best practices*) untuk implementasi sistem *transfer file* dalam lingkungan institusi ialah memilih protokol yang sesuai dengan kebutuhan keamanan dan jaringan (mis. SFTP/FTPS untuk keamanan), menerapkan manajemen kunci dan enkripsi *end-to-end*, menggunakan solusi MFT untuk audit dan automasi di lingkungan regulatif, serta menambahkan mekanisme pemulihan (*resume*), kompresi, dan pemeriksaan integritas (*checksums/HMAC*) untuk menjamin keandalan pengiriman. Pendekatan *hybrid* mengombinasikan layanan *cloud storage* yang terenkripsi dengan MFT atau protokol aman sering direkomendasikan untuk menjangkau skala dan kepatuhan (Redwood, 2024).

2.2. Kode QR (Quick Response Code)



Gambar 2. 4. Kode QR

Kode QR (*Quick Response Code*) adalah representasi dua dimensi dari data digital yang dapat dibaca oleh perangkat optik seperti kamera atau *scanner*. Kode ini pertama kali dikembangkan oleh perusahaan Jepang Denso Wave pada tahun 1994 untuk pelacakan suku cadang otomotif. Kini, QR Code digunakan secara luas dalam berbagai bidang, termasuk sistem pembayaran digital, verifikasi dokumen, promosi produk, dan autentikasi data.

Menurut Rachman (2023) dalam bukunya “Implementasi QR (Quick Response) Code pada Sertifikat Tanah Elektronik”, QR Code bekerja dengan menyimpan informasi dalam bentuk pola titik hitam dan putih yang diinterpretasikan menjadi data digital menggunakan perangkat pemindai. Rachman menjelaskan bahwa dalam konteks dokumen elektronik, QR Code digunakan untuk menjamin keaslian dan keamanan sertifikat tanah digital melalui integrasi dengan basis data pemerintah. Dengan demikian, teknologi ini berfungsi sebagai sarana autentikasi yang efisien sekaligus alat pencegah pemalsuan dokumen.

Penelitian Gunawan dan Hidayat (2021) dalam jurnal “Implementasi QR-Code pada Aplikasi E-Market Mandiri untuk Pemberdayaan Ekonomi Kreatif Berbasis Android” menunjukkan bahwa QR Code juga memiliki peran penting dalam pemberdayaan ekonomi digital. Dalam studi tersebut, QR Code digunakan sebagai media transaksi non-tunai dan alat promosi produk dalam aplikasi *e-marketplace* berbasis Android. Teknologi ini memungkinkan pelaku usaha kecil menengah (UKM) menghubungkan informasi produk mereka secara langsung dengan pembeli tanpa harus menggunakan perantara platform besar, sehingga efisien dan ramah pengguna.

Sementara itu, Taufik et al. (2022) dalam jurnal *Asia Pacific Journal of Information Systems* (APJIS) berjudul “Integrated QR Payment System (QRIS): Cashless Payment Solution in Developing Country from Merchant Perspective” menjelaskan bahwa implementasi QR Code dalam sistem pembayaran terintegrasi seperti QRIS (*Quick Response Code Indonesian Standard*) telah menjadi pendorong utama digitalisasi ekonomi di negara berkembang. QRIS memungkinkan berbagai platform pembayaran digital beroperasi dalam satu standar yang sama, mempermudah pedagang menerima pembayaran dari berbagai aplikasi.

Ketiga sumber tersebut menegaskan bahwa teknologi QR Code tidak hanya sekadar alat penyimpan data, tetapi juga menjadi infrastruktur penting dalam sistem digital modern, baik di sektor publik, perdagangan elektronik, maupun keuangan digital. Fungsinya yang cepat, mudah diimplementasikan, dan aman menjadikan QR Code sebagai solusi inovatif dalam mendukung efisiensi serta transparansi layanan digital.

2.3. Utilitas

Dalam sistem informasi dan rekayasa perangkat lunak, utilitas mengacu pada kumpulan fungsi atau layanan pendukung yang berperan dalam memastikan sistem dapat berjalan dengan optimal, efisien, dan mudah digunakan oleh pengguna. Komponen utilitas tidak hanya mencakup aspek teknis seperti pemrosesan data dan penyimpanan, tetapi juga mencakup aspek non-teknis seperti *usability* dan interaksi pengguna dengan sistem.

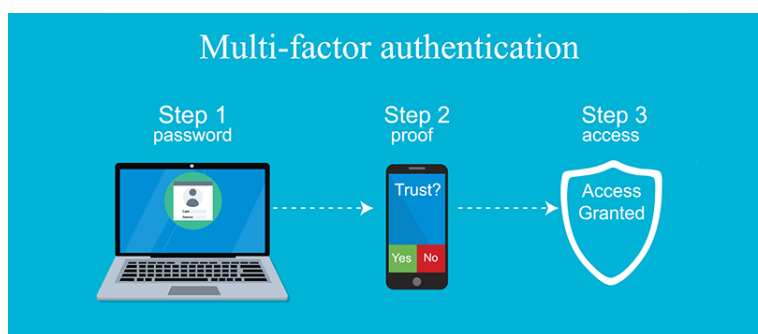
Menurut Suharyadi (2023) dalam bukunya “*Sistem Utilitas*”, utilitas merupakan subsistem penting yang mendukung operasi utama perangkat lunak. Utilitas berfungsi sebagai alat bantu (*tools*) yang menjalankan proses pemeliharaan, *monitoring*, dan pengaturan sistem agar dapat beroperasi sesuai kebutuhan pengguna. Dalam konteks sistem digital modern, utilitas mencakup modul seperti *backup system*, *data recovery*, *network monitoring*, dan *performance optimization*. Suharyadi menekankan bahwa pengelolaan utilitas yang baik dapat meningkatkan keandalan (*reliability*) dan umur sistem secara keseluruhan.

Penelitian Ali (2020) dalam jurnal “*Rekayasa Perangkat Lunak*” menjelaskan bahwa dalam tahap *software engineering lifecycle*, utilitas merupakan bagian dari lapisan pendukung sistem yang berperan dalam menghubungkan komponen fungsional dengan antarmuka pengguna. Utilitas dapat berupa perangkat lunak tambahan seperti *library manager*, *debugging tool*, atau *automation script* yang mempercepat proses pengembangan. Ia juga menyebutkan bahwa aspek utilitas sering kali diabaikan pada tahap awal pengembangan, padahal memiliki peran penting dalam menjamin keberlanjutan sistem pasca implementasi.

Lebih lanjut, MDPI (2022) dalam topik “*Theories and Applications of Human-Computer Interaction*” menyoroti keterkaitan antara utilitas dan kenyamanan pengguna (*user experience*). Dalam teori interaksi manusia dan komputer (HCI), utilitas didefinisikan sebagai kemampuan sistem untuk memberikan manfaat langsung yang sesuai dengan tujuan pengguna. Suatu sistem dikatakan memiliki utilitas tinggi jika pengguna dapat menyelesaikan tugas dengan cepat, akurat, dan tanpa kebingungan. Oleh karena itu, desain antarmuka dan penyediaan fitur pendukung harus mempertimbangkan kebutuhan utilitas ini agar interaksi pengguna dengan sistem lebih efisien dan intuitif.

Berdasarkan berbagai literatur tersebut, dapat disimpulkan bahwa utilitas berperan sebagai komponen penting dalam sistem informasi maupun perangkat lunak. Ia berfungsi tidak hanya sebagai alat bantu teknis, tetapi juga sebagai penentu kualitas interaksi antara manusia dan komputer. Pengembangan sistem dengan utilitas yang baik akan menghasilkan perangkat lunak yang lebih handal, mudah dioperasikan, dan memiliki tingkat kepuasan pengguna yang tinggi.

2.4. Keamanan Sistem Informasi (*Multi Factor Authentication – MFA*)



Gambar 2. 5. Multi Factor Authentication

Multi-Factor Authentication (MFA) adalah mekanisme keamanan berbasis autentikasi yang menggabungkan dua atau lebih metode verifikasi independen untuk memverifikasi identitas pengguna sebelum memberikan akses ke sistem, aplikasi, atau data sensitif. Menurut Kamba dan Dauda (2025), MFA merepresentasikan paradigma baru dalam mekanisme kontrol akses dengan menggabungkan sesuatu yang diketahui pengguna (seperti kata sandi), sesuatu yang dimiliki pengguna (seperti smartphone atau token), dan sesuatu yang merupakan identitas pengguna (data biometrik).

Pendekatan berlapis ini secara signifikan meningkatkan kesulitan bagi individu yang tidak berwenang untuk mendapatkan akses, bahkan jika salah satu faktor telah dikompromikan. Dalam konteks keamanan sistem informasi modern, MFA telah menjadi komponen fundamental dalam strategi pertahanan mendalam (*defense-in-depth*) mengingat meningkatnya sofistikasi serangan siber terhadap organisasi dan individu.

2.4.1. Klasifikasi Faktor Autentikasi dalam MFA

Penelitian yang dilakukan oleh Syahreem dkk. (2024) dalam studi literatur sistematis mereka mengidentifikasi bahwa MFA memanfaatkan tiga kategori faktor utama untuk memverifikasi identitas pengguna:

a. *Something You Know* (Sesuatu yang Anda Ketahui)

Faktor ini mencakup informasi yang hanya diketahui oleh pengguna yang sah, seperti kata sandi (password), nomor identifikasi pribadi (PIN), atau jawaban pertanyaan keamanan. Sebagai faktor paling tradisional, metode berbasis pengetahuan ini mudah diimplementasikan namun rentan terhadap serangan *phishing*, *brute-force*, dan *social engineering*.

b. *Something You Have* (Sesuatu yang Anda Miliki)

Faktor kepemilikan melibatkan perangkat fisik atau virtual yang dimiliki eksklusif oleh pengguna, termasuk smartphone, token hardware, *smart card*, atau aplikasi autentikator. Teknologi ini menambahkan lapisan keamanan tambahan karena penyerang tidak hanya memerlukan informasi pengguna tetapi juga akses fisik ke perangkat tersebut.

c. *Something You Are* (Sesuatu yang Anda Adalah)

Faktor biometrik menggunakan karakteristik fisik atau perilaku unik pengguna sebagai basis autentikasi, meliputi sidik jari (*fingerprint*), pengenalan wajah (*facial recognition*), scan iris, atau pengenalan suara. Jain dan Ross (2020) menekankan bahwa faktor biometrik menawarkan tingkat keamanan yang lebih tinggi karena sifat unik dan permanen dari data biologis manusia.

2.4.2. Jenis-Jenis Implementasi MFA

Dalam literatur keamanan sistem informasi, MFA diimplementasikan melalui berbagai metode sesuai dengan kebutuhan organisasi dan tingkat sensitivitas data yang dilindungi:

1. Autentikasi Dua Faktor (*Two-Factor Authentication* – 2FA)

Merupakan bentuk paling sederhana dari MFA yang menggabungkan dua metode verifikasi. Contoh umum adalah kombinasi kata sandi dengan kode *One-Time Password* (OTP) yang dikirimkan melalui SMS atau aplikasi autentikator. Namun, Garfinkel (2021) mencatat bahwa metode berbasis SMS rentan terhadap serangan *SIM swapping* dan tidak memberikan perlindungan setara dengan faktor biometrik.

2. Autentikasi Multi-Faktor Berbasis *One-Time Password* (OTP)

Metode ini menghasilkan kode sementara yang hanya berlaku untuk waktu terbatas. Syahreem dkk. (2024) mengidentifikasi dua varian utama: HOTP (*HMAC-Based OTP*) dan TOTP (*Time-Based OTP*). TOTP adalah metode yang paling banyak digunakan karena sinkronisasinya berbasis waktu sistem tanpa memerlukan penyimpanan counter.

3. Autentikasi Biometrik Multi-Faktor

Menggabungkan dua atau lebih modalitas biometrik (*multi-biometric*) untuk meningkatkan akurasi dan keamanan. Penelitian menunjukkan bahwa kombinasi sidik jari dan pengenalan wajah menghasilkan tingkat akurasi lebih tinggi dan mengurangi *false acceptance rate* (FAR) serta *false rejection rate* (FRR).

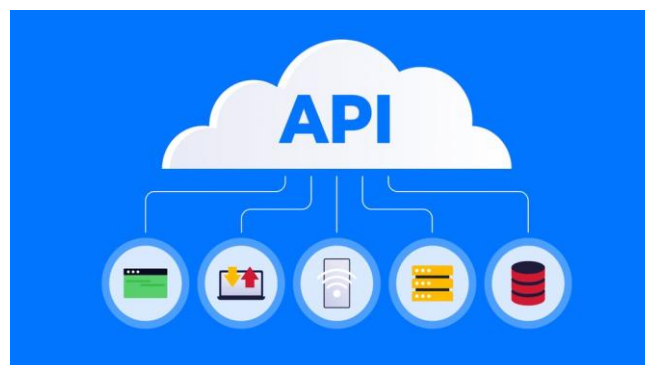
4. Autentikasi Berbasis Token Hardware

Menggunakan perangkat fisik seperti Yubikey atau RSA Token yang menghasilkan kode autentikasi. Metode ini menawarkan keamanan tinggi tetapi memerlukan investasi infrastruktur signifikan dan manajemen siklus hidup perangkat yang kompleks.

5. Autentikasi Berbasis *Smart Card*

Smart card menyimpan sertifikat digital dan kunci kriptografi, memberikan perlindungan berlapis dengan enkripsi. Teknologi ini banyak digunakan dalam lingkungan *enterprise* namun memerlukan pembaca smart card khusus.

2.5. Application Programming Interface (API)



Gambar 2. 6. Application Programming Interface

Application Programming Interface (API) adalah seperangkat protokol, alat, dan definisi untuk membangun perangkat lunak aplikasi yang menentukan bagaimana komponen perangkat lunak harus berinteraksi. API berfungsi sebagai perantara yang memungkinkan komunikasi antara sistem yang berbeda, aplikasi, atau layanan dalam lingkungan yang terdistribusi. Menurut Triawan (2021) dalam penelitiannya tentang penerapan API pada *push notification*, API adalah mekanisme yang menjadi penghubung antara sistem yang sudah ada dengan sistem lain yang akan mengirimkan informasi, memfasilitasi integrasi yang seamless dalam ekosistem teknologi informasi modern.

API memungkinkan pengembang untuk mengakses fungsionalitas atau data tertentu dari aplikasi, sistem operasi, atau layanan mikro tanpa perlu memahami detail implementasi internal. Dengan demikian, API mengurangi kompleksitas pengembangan dan mempercepat proses integrasi sistem. Dalam konteks aplikasi modern, API telah menjadi komponen esensial yang mendorong inovasi, meningkatkan pengalaman pengguna, dan memfasilitasi kolaborasi antar platform.

Penerapan API dalam sistem informasi menciptakan arsitektur yang lebih modular, *scalable*, dan *maintainable*. Organisasi dapat mengembangkan berbagai aplikasi yang berkomunikasi melalui API yang terdefinisi dengan baik, memungkinkan *reusability* kode dan pengurangan duplikasi fungsionalitas. Lebih lanjut, API memungkinkan organisasi untuk mengintegrasikan sistem *legacy* dengan teknologi terbaru tanpa perlu melakukan rekonstruksi menyeluruh terhadap infrastruktur yang sudah ada.

Dalam praktik pengembangan sistem informasi, terdapat berbagai jenis arsitektur API yang dipilih berdasarkan kebutuhan spesifik aplikasi, performa, dan skalabilitas yang diinginkan:

a. REST API (*Representational State Transfer*)

REST API adalah gaya arsitektur API yang paling dominan dan banyak digunakan dalam pengembangan web modern. REST API mengikuti prinsip desain berbasis *resource* yang menekankan interaksi berbasis *resource* melalui HTTP. Menurut Gopal (2025), REST API memanfaatkan HTTP *verbs* (*GET, POST, PUT, DELETE*) untuk memanipulasi data pada server, menawarkan pendekatan yang familiar dan fleksibel. Setiap *resource*

dalam REST API memiliki identifikasi unik melalui URI (*Uniform Resource Identifier*), seperti “base_url/api/users/123”, dan operasi pada *resource* dilakukan melalui HTTP methods yang jelas.

Keunggulan REST API mencakup kesederhanaan implementasi, *widespread adoption*, *caching* yang efisien, dan kompatibilitas yang baik dengan arsitektur microservices. Namun, REST API dapat menyebabkan *over-fetching* (pengambilan data lebih dari yang dibutuhkan) atau *under-fetching* (kurangnya data yang dibutuhkan), mengharuskan *multiple requests* untuk mendapatkan informasi lengkap.

b. GraphQL API

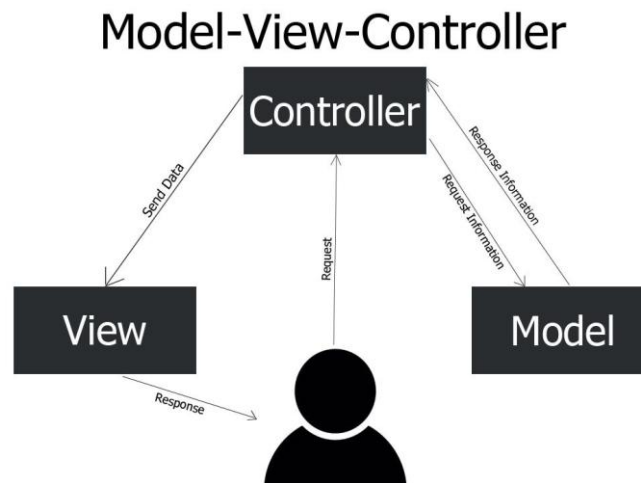
GraphQL adalah query language dan spesifikasi yang memungkinkan client untuk meminta data spesifik dari API dalam satu request, meningkatkan performa dan efisiensi. Berbeda dengan REST API yang menggunakan multiple endpoints, GraphQL beroperasi melalui single endpoint dan memungkinkan client untuk menentukan struktur data response secara eksak.

GraphQL menggunakan tiga konsep utama: queries untuk mengambil data *read-only*, mutations untuk memodifikasi data, dan subscriptions untuk menerima update data secara *real-time* atau *event-based*. Gopal (2025) menjelaskan bahwa GraphQL menggunakan *server-side schema* yang mendefinisikan tipe data dan *services*, memungkinkan sistem untuk secara otomatis mengidentifikasi *error request* dan memberikan pesan *error* yang berguna. Keuntungan GraphQL termasuk flexibility dalam data *retrieval*, *strong typing*, dan *backward compatibility* yang lebih baik melalui *mekanisme deprecation* dibandingkan REST API.

c. SOAP API (*Simple Object Access Protocol*)

SOAP API adalah protokol *message-based* yang kurang umum digunakan dibandingkan REST atau GraphQL. SOAP API menggunakan XML untuk pertukaran data dan mengandalkan *Web Services Description Language* (WSDL) untuk mendefinisikan interface layanan. Keamanan dalam SOAP API melibatkan digital signatures dan message encryption menggunakan protokol seperti SSL atau TLS untuk menjamin integritas data dan kerahasiaan selama pertukaran data. SOAP API lebih cocok untuk lingkungan enterprise dengan kebutuhan formal dan keamanan yang ketat.

2.6. Model-View-Controller (MVC)



Gambar 2. 7. Model-View-Controller (MVC)

Model-View-Controller (MVC) adalah arsitektur desain perangkat lunak yang membagi aplikasi menjadi tiga komponen yang saling terhubung untuk memisahkan tanggung jawab dan meningkatkan modularitas sistem. Menurut Ma'arif (2010) dalam penelitiannya tentang implementasi MVC pada sistem informasi berbasis web, MVC adalah cara untuk menguraikan aplikasi menjadi tiga bagian: *model*, *view*, dan *controller*, yang masing-masing memiliki tugas dan tanggung jawab tersendiri. Arsitektur ini awalnya diterapkan pada model interaksi pengguna berbasis grafis dengan konsep *input*, *processing*, dan *output*.

Penggunaan arsitektur MVC dalam pengembangan sistem informasi diharapkan dapat menghasilkan aplikasi yang lebih *modular*, *reusable*, mudah dimaintain, dan mudah untuk dimigrasikan ke sistem lain. Sihombing (2021) mengungkapkan bahwa penerapan konsep MVC pada sistem informasi berbasis web menghasilkan website yang lebih terstruktur dengan memisahkan data (*model*), tampilan (*view*), dan cara bagaimana memprosesnya (*controller*), menciptakan pemisahan kepedulian yang jelas dan terorganisir. Dengan memisahkan ketiga komponen ini, developer dapat bekerja lebih efisien karena masing-masing bagian dapat dikembangkan, diuji, dan diperbaiki secara independen.

2.6.1. PHP



Gambar 2. 8. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang dirancang khusus untuk pengembangan aplikasi web dinamis. PHP memungkinkan developer untuk membuat konten web yang interaktif dan terhubung dengan database. Sebagai bahasa *scripting* yang dieksekusi di sisi server, PHP menghasilkan HTML yang akan dikirimkan ke browser client, sehingga client tidak perlu melihat kode PHP asli.

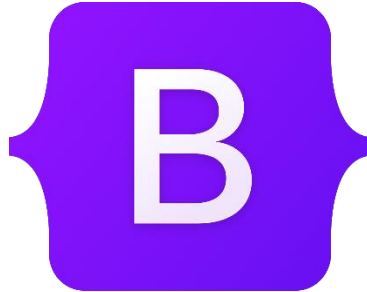
Dalam konteks arsitektur MVC, PHP berfungsi sebagai bahasa pemrograman *backend* yang mengimplementasikan logika kontrol dan model aplikasi. PHP menyediakan akses ke berbagai *library* dan *framework* yang memudahkan implementasi MVC, seperti Laravel, CodeIgniter, dan Symfony. Niarman dkk. (2023) dalam penelitian "*Comparative Analysis of PHP Frameworks for Development of Academic Information System Using Load and Stress Testing*" menunjukkan bahwa PHP tetap menjadi salah satu bahasa pemrograman paling menjanjikan untuk pengembangan layanan web, dengan performa yang handal dalam menangani load dan stress testing.

Keunggulan PHP mencakup kemudahan pembelajaran, kompatibilitas lintas platform, dan dukungan luas dari hosting providers. PHP menyediakan *built-in functions* yang kaya untuk manipulasi *string*, *handling file*, dan operasi database. Dalam pengembangan sistem informasi, PHP memfasilitasi komunikasi antara layer presentasi (*view*) dan layer data (*model*) melalui *controller*, mengorganisir logika bisnis dengan cara yang terstruktur dan mudah dipahami.

Penggunaan PHP dengan prinsip-prinsip OOP (*Object-Oriented Programming*) memungkinkan developer untuk menciptakan kode yang lebih modular dan *maintainable*. Dengan menggunakan class dan object, PHP memfasilitasi implementasi *design patterns*

seperti MVC, *Dependency Injection*, dan *Factory Pattern* yang meningkatkan kualitas dan fleksibilitas kode aplikasi.

2.6.2. Bootstrap



Gambar 2. 9. Bootstrap

Bootstrap adalah *framework front-end* yang paling populer di dunia untuk mengembangkan website dan aplikasi web yang responsif, *mobile-first*, dan dengan desain modern. Bootstrap menyediakan *pre-built components*, *grid system*, dan *stylesheet CSS* yang memudahkan developer untuk membangun *interface* pengguna yang konsisten dan professional tanpa perlu merancang UI dari awal.

Bootstrap menggunakan sistem grid berbasis 12 kolom yang memungkinkan developer untuk membuat layout responsif yang beradaptasi dengan berbagai ukuran layar, dari perangkat mobile dengan lebar kurang dari 768px hingga desktop dengan lebar lebih dari 1200px. Grid system Bootstrap menggunakan media *queries* CSS untuk menerapkan *breakpoints* yang berbeda berdasarkan ukuran *viewport*, memastikan bahwa aplikasi dapat diakses dengan optimal dari berbagai perangkat.

Dalam arsitektur MVC, Bootstrap berfungsi sebagai komponen *view* yang menyediakan *styling* dan *components* untuk *user interface*. Bootstrap menawarkan berbagai *pre-styled components* seperti *navigation bars*, *buttons*, *forms*, *modals*, *cards*, dan *tables* yang dapat diintegrasikan langsung ke dalam template HTML. Dengan menggunakan Bootstrap, developer dapat fokus pada logika bisnis dan fungsionalitas aplikasi daripada menghabiskan waktu untuk styling CSS dari awal.

Bootstrap juga menyediakan *library* JavaScript yang terintegrasi, memungkinkan *interactive components* seperti *dropdown*, *carousel*, *tooltip*, dan *modal* tanpa perlu menulis JavaScript custom. Filosofi Bootstrap yang *mobile-first* memastikan bahwa aplikasi akan berfungsi optimal di perangkat mobile, yang merupakan prioritas utama

dalam pengembangan web modern mengingat mayoritas pengguna internet mengakses layanan melalui smartphone.

2.6.3. MySQL



Gambar 2. 10. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang *open-source* dan paling banyak digunakan dalam pengembangan aplikasi web. MySQL dirancang untuk menyimpan, mengelola, dan mengambil data secara efisien dengan performa yang *reliable* dan skalabilitas yang baik. Sebagai bagian dari *stack* LAMP/LEMP (Linux, Apache/Nginx, MySQL, PHP), MySQL menjadi pilihan utama untuk *backend* database di sejumlah besar aplikasi web di seluruh dunia.

Menurut Siregar (2024) dalam penelitian "Pengembangan database Management system Menggunakan MySQL", pengembangan DBMS menggunakan MySQL melibatkan proses sistematis dari perencanaan awal hingga implementasi dan *maintenance*. Proses ini mencakup identifikasi kebutuhan bisnis, perancangan struktur database dengan mempertimbangkan normalisasi dan hubungan antar entitas, pembuatan table dengan kolom, tipe data, dan *key* yang tepat, penetapan akses *rights* untuk keamanan data, implementasi *functions* dan *stored procedures* untuk memfasilitasi pemrosesan data kompleks, serta penambahan index untuk meningkatkan performa *query*.

MySQL menyediakan berbagai fitur *advanced* seperti *stored procedures*, *triggers*, *views*, dan *transactions* yang meningkatkan integritas data dan performa aplikasi. Dalam implementasi MVC, MySQL berfungsi sebagai layer model yang menyimpan dan mengelola semua data aplikasi. Melalui PHP dengan menggunakan MySQLi atau PDO (PHP Data Objects), aplikasi dapat berkomunikasi dengan MySQL untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) yang fundamental.

Operasi CRUD dalam MySQL melalui PHP diimplementasikan menggunakan SQL *queries* untuk *INSERT, SELECT, UPDATE, dan DELETE*. Normalisasi database dalam MySQL menghilangkan redundansi data dan memastikan konsistensi data, mengikuti

prinsip-prinsip normal forms (1NF hingga 3NF). Penambahan index pada kolom-kolom yang sering diakses secara signifikan meningkatkan kecepatan *query execution*, terutama untuk operasi *SELECT* pada dataset besar.

Fitur *backup* dan *recovery management* dalam MySQL memastikan *business continuity* dan proteksi data terhadap kehilangan atau kerusakan data. MySQL juga menyediakan mekanisme *security* berbasis *user permissions* yang granular, memungkinkan administrator untuk mengontrol akses data pada level database, *table*, dan bahkan *column level*. Dengan menggunakan *prepared statements* dan parameter binding dalam PHP-MySQL *integration*, aplikasi dapat terlindungi dari serangan *SQL injection* yang merupakan salah satu *security vulnerability* paling kritis dalam web applications.

2.7. Metode Rapid Application Development

2.8. UML

2.9. ERD

2.10. UI/UX

2.11. BlackBox Testing

2.12. User Acceptance Testing (UAT)

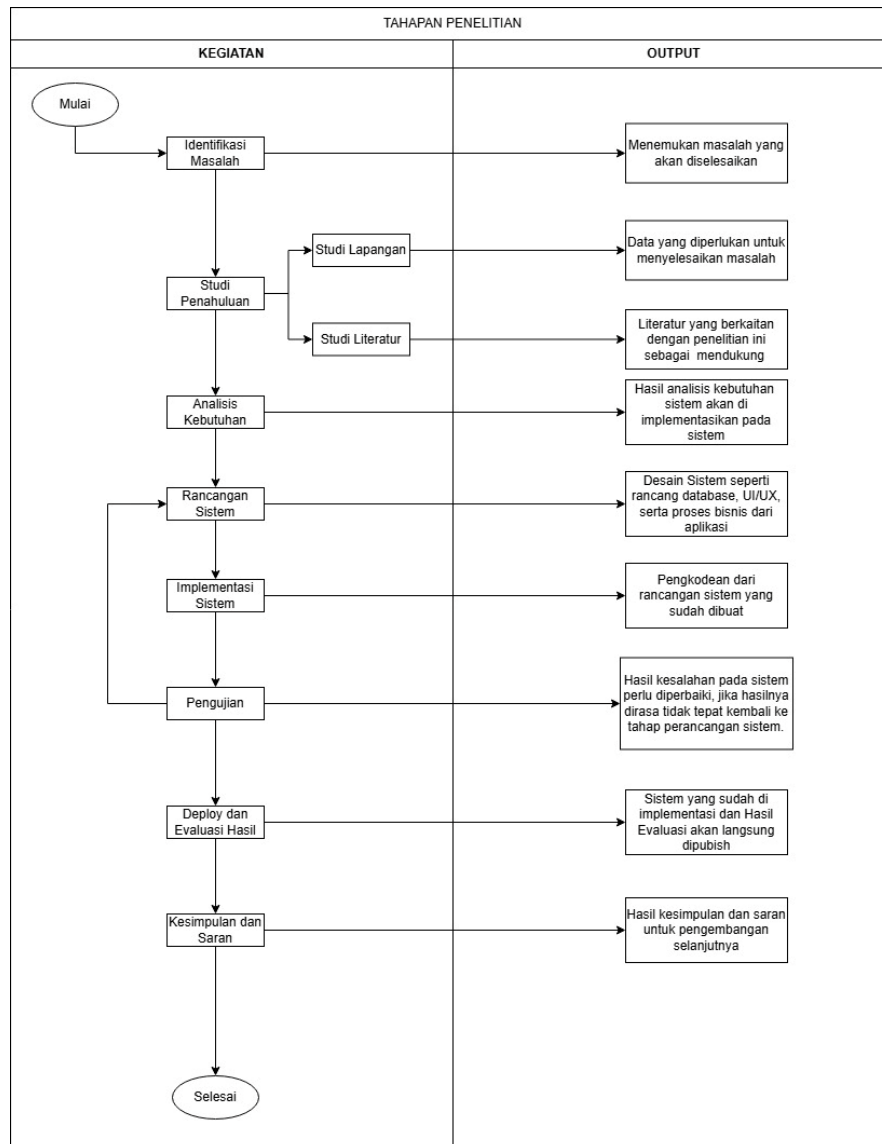
d

BAB 3

METODE PENELITIAN

3.1. Tahapan Penelitian

Dalam proses penelitian, cara pengembangan perangkat lunak yang dipakai adalah model *Rapid Application Development* (RAD). RAD dinilai tepat untuk pengembangan web sebab sistem yang dibuat adalah platform untuk mengelola tautan dan alat daring yang membutuhkan perubahan secara terus-menerus sesuai keinginan pengguna di Institut Teknologi Indonesia. Selain itu karena memungkinkan pengerjaan sebuah sistem dalam kurun waktu yang cukup singkat 60 - 90 hari sesuai dengan perkiraan masa pengerjaan skripsi serta tidak memakan biaya yang mahal. Di bawah ini akan dijelaskan tentang rencana penelitian, termasuk metode yang digunakan, serta jenis dan sifat penelitian tersebut.



Gambar 3. 1. Tahapan Penelitian

3.2. Alur dan Metode Penelitian

Penelitian ini menggunakan pendekatan *Rapid Application Development* (RAD) karena karakteristiknya yang menekankan kecepatan, kolaborasi, dan fleksibilitas dalam proses pembuatan sistem. Model RAD memungkinkan pembuatan prototipe sistem secara cepat melalui tahapan iteratif, di mana pengguna secara aktif terlibat untuk memberikan masukan pada setiap siklus pengembangan.

Setiap komponen utama seperti *shortlink*, *biolink*, kode QR, *transfer file*, dan *Tools Online* dikembangkan secara bertahap melalui fase perencanaan, desain, pembuatan, dan implementasi. Dengan model RAD, pengembang dapat langsung menguji fungsionalitas

inti, menerima masukan dari pengguna, lalu melakukan perbaikan secara cepat tanpa harus menunggu sistem selesai sepenuhnya.

Dengan demikian, metode RAD memberikan keseimbangan antara memenuhi kebutuhan pengguna secara tepat dan mempercepat proses pengembangan, serta memastikan sistem akhir yang dihasilkan benar-benar sesuai dengan kebutuhan sivitas akademika Institut Teknologi Indonesia.

3.3. Tahap Perencanaan Kebutuhan

Tahap pertama yang akan dilaksanakan adalah dengan melakukan *requirement planning* dengan maksud untuk mengetahui kebutuhan sistem manajemen link beserta dengan utilitas online yang diperlukan. Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam proses analisis, perancangan, hingga pengujian sistem. Metode pengumpulan data dalam penelitian ini meliputi:

1. Observasi (Pengamatan Langsung)

Pengembang melakukan pengamatan terhadap kebutuhan pengguna kampus terhadap alat bantu digital seperti tautan pendek, QR Code, dan pengelolaan profil digital (*biolink*).

Observasi dilakukan dengan melihat cara mahasiswa dan dosen menggunakan layanan web serupa serta menganalisis kendala yang mereka hadapi.

2. Wawancara

Wawancara dilakukan dengan beberapa mahasiswa dan dosen di Institut Teknologi Indonesia untuk mengetahui kebutuhan dan harapan mereka terhadap sistem.

Wawancara ini berfokus pada fitur yang paling dibutuhkan, kemudahan penggunaan, serta tampilan antarmuka yang diinginkan.

3. Studi Literatur

Studi literatur dilakukan dengan meninjau jurnal, buku, dan sumber daring terkait sistem manajemen tautan, biolink, dan QR Code generator, serta

implementasi *Rapid Application Development* (RAD) pada pengembangan aplikasi berbasis web.

Hasil dari pengumpulan data ini menjadi dasar untuk menyusun kebutuhan pengguna (*user requirements*) dan rancangan sistem yang sesuai.

3.3.1. User Requirements

Kebutuhan pengguna (*user requirements*) mencakup spesifikasi dan fungsi utama yang harus dimiliki sistem agar dapat digunakan secara optimal oleh sivitas akademika Institut Teknologi Indonesia. Berdasarkan hasil observasi dan wawancara, kebutuhan pengguna dibagi menjadi dua bagian utama:

3.3.1.1. Kebutuhan Fungsional (*Functional Requirements*):

- a) Sistem dapat membuat tautan pendek (*shortlink*) dari URL panjang.
- b) Sistem menyediakan fitur pembuatan halaman profil (*biolink*) dengan beberapa tautan sekaligus.
- c) Sistem dapat menghasilkan kode QR dari tautan yang dimasukkan dapat bersifat *dynamic* (dengan tautan yang sudah dibuat dalam sistem) maupun *static* (dengan tautan yang berasal dari luar sistem).
- d) Sistem menyediakan fitur *transfer file* sederhana melalui tautan unik atau di *custom*.
- e) Sistem memiliki kumpulan alat bantu (*online tools*) seperti *PDF Tools*, *generator tools*, dan *checker tools*.
- f) Sistem menampilkan statistik penggunaan (jumlah klik, tanggal akses terakhir, dan sumber pengunjung).
- g) Pengguna dapat mengelola, mengedit, dan menghapus tautan yang telah dibuat.
- h) Tautan yang dibuat memiliki fungsi *time expired* dan bisa dipasang *password* yang bersifat opsional.

3.3.1.2. Kebutuhan Non-Fungsional (*Non-Functional Requirements*):

- a) Sistem harus mudah digunakan dan memiliki antarmuka yang responsif.
- b) Akses sistem dapat dilakukan melalui berbagai perangkat (*desktop*, *tablet*, dan *smartphone*).
- c) Sistem harus memiliki tingkat keamanan dasar seperti validasi input dan pembatasan akses admin.

- d) Sistem harus dapat berjalan stabil di lingkungan server lokal (localhost) maupun server publik.

3.3.2. Analisis Proses

Analisis proses dilakukan untuk memahami alur kerja sistem yang akan dibangun serta bagaimana setiap komponen saling berinteraksi dalam mendukung fungsi utama aplikasi. Tahap ini bertujuan untuk memetakan proses bisnis dari sistem manajemen tautan dan *tools online*, sehingga dapat diketahui kebutuhan logika, data, dan fitur yang perlu dikembangkan.

Analisis ini melibatkan identifikasi aktor, aktivitas utama, serta hubungan antarproses yang terlibat dalam sistem. Adapun hasil analisis proses dijelaskan sebagai berikut:

1. Identifikasi Aktor

Dalam sistem ini terdapat dua aktor utama, yaitu:

- Admin:

Bertanggung jawab untuk mengelola seluruh data dalam sistem, termasuk verifikasi akun, pemantauan aktivitas pengguna, serta pengaturan konfigurasi global sistem.

- Pengguna (User):

Sivitas akademika Institut Teknologi Indonesia (mahasiswa, dosen, dan staf) yang menggunakan sistem untuk membuat tautan pendek, biolink, QR Code, atau memanfaatkan berbagai tools online yang disediakan.

2. Analisis Proses Utama

Berdasarkan hasil observasi dan kebutuhan pengguna, proses utama dalam sistem dapat dijelaskan sebagai berikut:

- **Pembuatan Shortlink**, Pengguna memasukkan URL panjang ke dalam form, lalu sistem akan mengubahnya menjadi URL pendek menggunakan algoritma pemendek tautan. Data tautan disimpan di database beserta statistik jumlah klik.

- **Pembuatan Biolink**, Pengguna dapat menambahkan beberapa tautan ke dalam satu halaman profil digital. Proses ini melibatkan input data berupa nama, deskripsi, ikon media sosial, dan URL tujuan.
- **Pembuatan QR Code**, Sistem menghasilkan kode QR dari tautan yang dimasukkan oleh pengguna. QR tersebut dapat diunduh dan digunakan untuk keperluan promosi atau berbagi informasi secara cepat.
- **Transfer File Online**, Pengguna dapat mengunggah file ke server, kemudian sistem akan membuat tautan unduhan unik yang dapat dibagikan kepada pengguna lain dalam waktu tertentu.
- **Penggunaan Tools Online**, Sistem menyediakan berbagai alat bantu berbasis web seperti text converter, encoder-decoder, dan generator utilitas sederhana untuk keperluan akademik maupun pribadi.
- **Manajemen Data oleh Admin**, Admin dapat melihat daftar tautan, laporan statistik penggunaan, dan melakukan tindakan seperti menghapus, memblokir, atau memperbaiki data pengguna jika diperlukan.

3. Alur Umum Proses Sistem

Secara garis besar, alur kerja sistem dapat dijelaskan sebagai berikut:

- Pengguna melakukan login (opsional, tergantung fitur publik atau privat).
- Pengguna memilih jenis layanan (*shortlink*, biolink, *QR code*, file, atau *tools*).
- Sistem memproses input pengguna sesuai jenis layanan.
- Hasil (tautan pendek, QR, atau *file link*) ditampilkan dan disimpan dalam database.
- Pengguna dapat melihat statistik penggunaan dan melakukan pengelolaan data.
- Admin memantau aktivitas sistem secara berkala melalui panel admin.

4. Diagram Konseptual Proses (Deskriptif)

Karena penelitian ini berfokus pada pendekatan RAD, analisis proses juga berfungsi sebagai dasar untuk membangun prototipe awal sistem. Secara konseptual, alur sistem dapat digambarkan sebagai berikut:

Input → Proses → Output

- Input: Data pengguna (URL, file, atau teks)
- Proses: Validasi, konversi, dan penyimpanan ke database
- Output: Tautan pendek, halaman biolink, QR Code, atau hasil tools online

3.3.3. Bisnis Identifikasi Aktor dan *Stakeholder*

Tahapan identifikasi aktor dan *stakeholder* dilakukan untuk mengetahui siapa saja pihak yang berperan, terlibat, serta berkepentingan terhadap pengembangan dan penggunaan sistem. Analisis ini penting agar sistem yang dibangun benar-benar sesuai dengan kebutuhan nyata para pengguna dan pihak institusi.

Identifikasi aktor dan *stakeholder* dalam penelitian ini dibagi menjadi dua bagian, yaitu aktor sistem (yang berinteraksi langsung dengan aplikasi) dan *stakeholder* bisnis (yang memiliki kepentingan terhadap keberlangsungan dan manfaat sistem).

2. Aktor Sistem

Aktor merupakan entitas yang secara langsung berinteraksi dengan sistem, baik sebagai pengguna maupun pengelola. Dalam sistem manajemen tautan dan utilitas online ini, terdapat dua jenis aktor utama:

a) Admin

Admin bertanggung jawab dalam mengelola sistem secara keseluruhan. Perannya meliputi:

- Mengelola data pengguna dan tautan yang tersimpan dalam sistem.
- Melakukan pemantauan dan moderasi aktivitas sistem.
- Menjamin kestabilan dan keamanan aplikasi.
- Mengatur konfigurasi sistem serta menambahkan fitur baru jika dibutuhkan.

b) Pengguna (User)

Pengguna merupakan sivitas akademika Institut Teknologi Indonesia yang memanfaatkan sistem untuk kebutuhan digital mereka. Perannya meliputi:

- Membuat tautan pendek, QR code, atau halaman biolink.
- Menggunakan berbagai tools online yang tersedia di sistem.
- Melihat statistik penggunaan tautan.
- Mengelola data miliknya seperti mengedit atau menghapus tautan yang sudah dibuat.

3. *Stakeholder* Bisnis

Stakeholder merupakan pihak-pihak yang memiliki kepentingan terhadap pengembangan dan keberhasilan sistem, baik secara langsung maupun tidak langsung. Identifikasi *stakeholder* dalam penelitian ini meliputi:

a) Sivitas Akademika Institut Teknologi Indonesia

Termasuk mahasiswa, dosen, dan staf administrasi. Mereka merupakan pengguna utama yang mendapatkan manfaat langsung dari sistem, seperti kemudahan membuat tautan digital, QR Code, dan pengelolaan profil biolink untuk kegiatan akademik.

b) Pihak Pengembang Sistem (Developer)

Berperan dalam proses analisis, desain, implementasi, dan pemeliharaan sistem. *Stakeholder* ini bertanggung jawab terhadap kualitas teknis, performa, dan keberlanjutan pengembangan sistem.

c) Lembaga atau Jurusan Terkait

Sebagai pihak yang berpotensi memanfaatkan sistem untuk kegiatan promosi, penelitian, atau pendataan digital kampus. Mereka berkepentingan agar sistem ini dapat mendukung kegiatan institusional secara efisien.

3.4. Desain Sistem

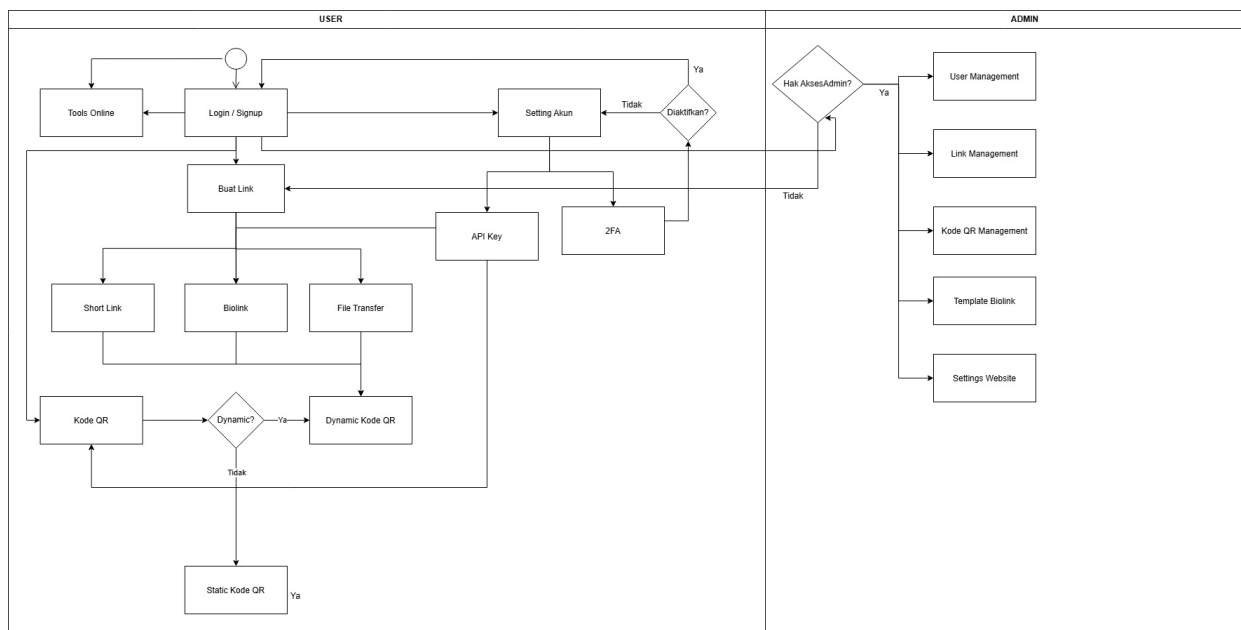
Tahap desain sistem merupakan fase di mana pengembang bersama pengguna secara aktif merancang atau menyempurnakan sistem yang sudah ada. Dalam fase ini, pengembang dan pengguna berkolaborasi untuk membuat alur kerja sistem hingga

antarmuka pengguna dalam bentuk prototipe sistem. Pada model *Rapid Application Development* (RAD), tahap tersebut umumnya terdiri atas tiga langkah:

- (1) perancangan alur kerja sistem,
- (2) pengembangan prototipe, dan
- (3) revisi berdasarkan masukan pengguna.

Dalam penelitian ini, teknik Data Flow Diagram (DFD) digunakan untuk menggambarkan desain sistem karena mampu membantu pengembang dan pengguna memvisualisasikan, mendokumentasikan, serta membangun sistem perangkat lunak secara terstruktur. Selain itu, sebagai media prototipe antarmuka digunakan Figma yang memudahkan pengguna memberikan umpan balik terhadap tampilan sebelum tahap konstruksi lebih lanjut.

3.4.1. Proses Bisnis



Gambar 3. 2. Proses Bisnis

Gambar di atas menunjukkan alur proses bisnis dari sistem Manajemen Link dan *Tools Online* yang terdiri dari dua aktor utama, yaitu User dan Admin. Setiap aktor memiliki peran dan aktivitas berbeda dalam sistem, yang saling terhubung untuk mendukung fungsionalitas utama platform.

1) Aktor: User

Aktor User merupakan pengguna umum yang dapat mendaftar, login, dan menggunakan berbagai layanan yang tersedia di dalam sistem.

Langkah-langkah proses bisnis untuk User sebagai berikut:

1. Login / Signup, Pengguna terlebih dahulu melakukan proses login atau pendaftaran akun untuk dapat mengakses layanan sistem.
2. *Tools Online*, Setelah login berhasil, pengguna dapat menggunakan berbagai tools yang tersedia seperti link management, QR code generator, dan file transfer.
3. Setting Akun, Pengguna dapat mengatur profil dan keamanan akunnya, termasuk pengaktifan fitur *Two Factor Authentication (2FA)*.
4. API Key, Sistem menyediakan API Key untuk integrasi layanan otomatis dengan sistem eksternal.
5. Buat Link, Pengguna dapat membuat berbagai jenis tautan, meliputi:
 - *Short Link* – untuk pemendekan URL.
 - Biolink – untuk membuat halaman profil berisi kumpulan tautan.
 - *File Transfer* – untuk berbagi berkas melalui tautan.
6. Pembuatan Kode QR, Setiap tautan dapat dikonversi menjadi Kode QR. Sistem akan mengecek apakah pengguna ingin membuat *Dynamic QR Code* (yang datanya dapat diubah) atau *Static QR Code* (yang bersifat tetap).
 - Jika *Dynamic*, sistem membuat *Dynamic QR Code* yang terhubung ke data *shortlink* di database.
 - Jika *Static*, sistem membuat *Static QR Code* tanpa keterhubungan data dinamis.
7. Semua aktivitas pengguna terhubung ke server utama untuk penyimpanan data dan validasi keamanan.

2) Aktor: Admin

Aktor Admin memiliki hak akses khusus untuk mengelola data, pengguna, serta konfigurasi sistem.

Proses bisnis untuk Admin mencakup:

1. Hak Akses Admin, Sistem akan memverifikasi apakah pengguna memiliki hak akses admin. Jika tidak, sistem akan menolak akses ke halaman pengaturan admin.
2. User Management, Admin dapat mengelola data pengguna, termasuk aktivasi akun, pengaturan peran, dan penghapusan akun.
3. *Link Management*, Admin memantau serta mengelola link yang dibuat oleh pengguna untuk mencegah penyalahgunaan.
4. *QR Code Management*, Admin mengawasi data QR Code yang dibuat, baik static maupun dynamic, untuk keperluan validasi sistem.
5. Template Biolink, Admin dapat membuat dan mengatur template tampilan biolink yang dapat digunakan oleh pengguna.
6. Settings Website, Admin mengatur konfigurasi umum sistem, seperti nama website, logo, API, serta fitur tambahan lainnya.

3) Integrasi dan Alur Keseluruhan

- Proses dimulai dari login/signup oleh pengguna.
- Sistem akan mengecek status aktivasi dan hak akses.
- Pengguna umum dapat membuat link, QR, dan file; sementara admin memiliki kontrol penuh atas semua data dalam sistem.
- Semua proses tersimpan dan diatur dalam server manajemen link dan tools online, yang menjadi pusat pengelolaan data serta autentikasi.

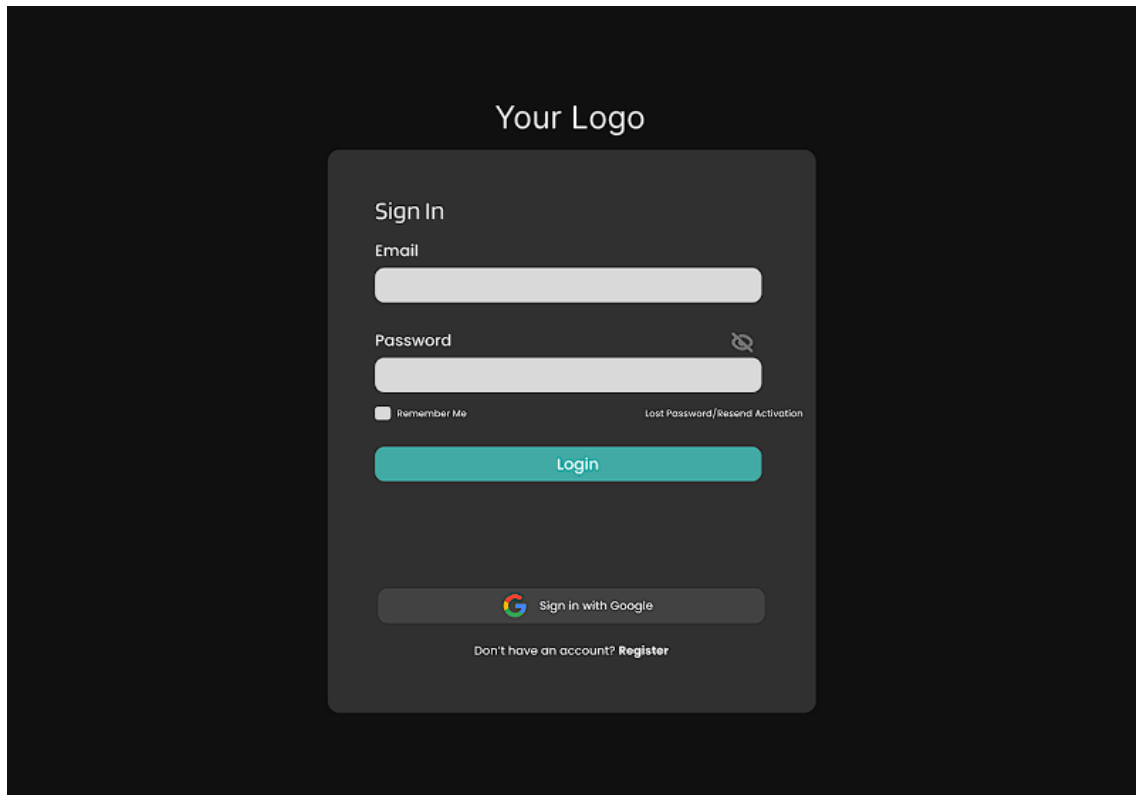
3.4.2. UI dan UX

UI dan UX adalah singkatan dari *User Interface* dan *User experience* yakni merupakan sebuah tampilan visual dalam sebuah aplikasi atau alat pemasaran digital

dalam bentuk Website yang dapat meningkatkan brand yang dimiliki oleh bisnis atau perusahaan (Ikhsan, 2023).

Pada tahap ini UI/UX dibuat menggunakan figma terdiri dari,

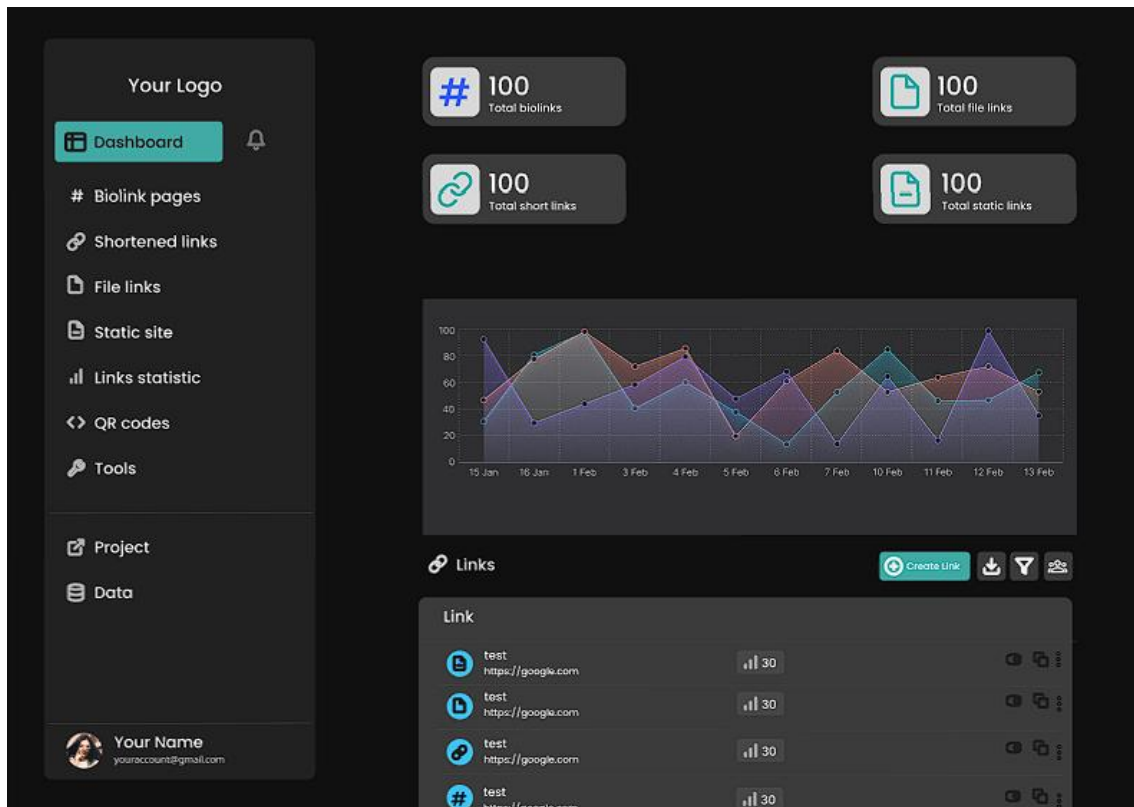
a) Halaman Login



Gambar 3. 3. Halaman Login

Halaman ini berfungsi untuk mengautentikasi pengguna sebelum masuk ke dalam sistem. Pengguna harus mengisi email dan kata sandi yang terdaftar untuk dapat mengakses dashboard utama. Desain halaman dibuat sederhana dengan fokus pada bidang input agar proses login cepat dan intuitif.

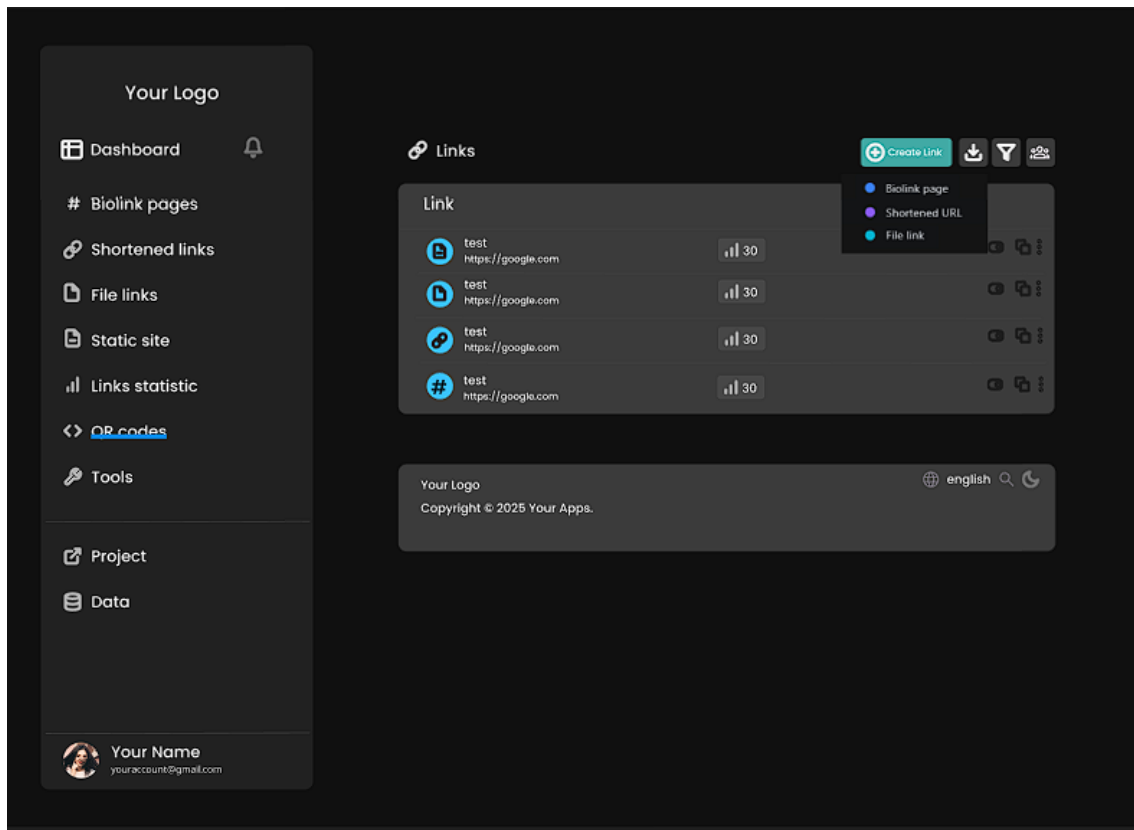
b) Halaman Dashboard



Gambar 3. 4. Halaman Dashboard

Setelah berhasil masuk, pengguna diarahkan ke halaman Dashboard yang menampilkan ringkasan data seperti jumlah link, jumlah kunjungan, dan statistik penggunaan sistem. Grafik interaktif digunakan untuk memvisualisasikan performa tautan dalam periode waktu tertentu.

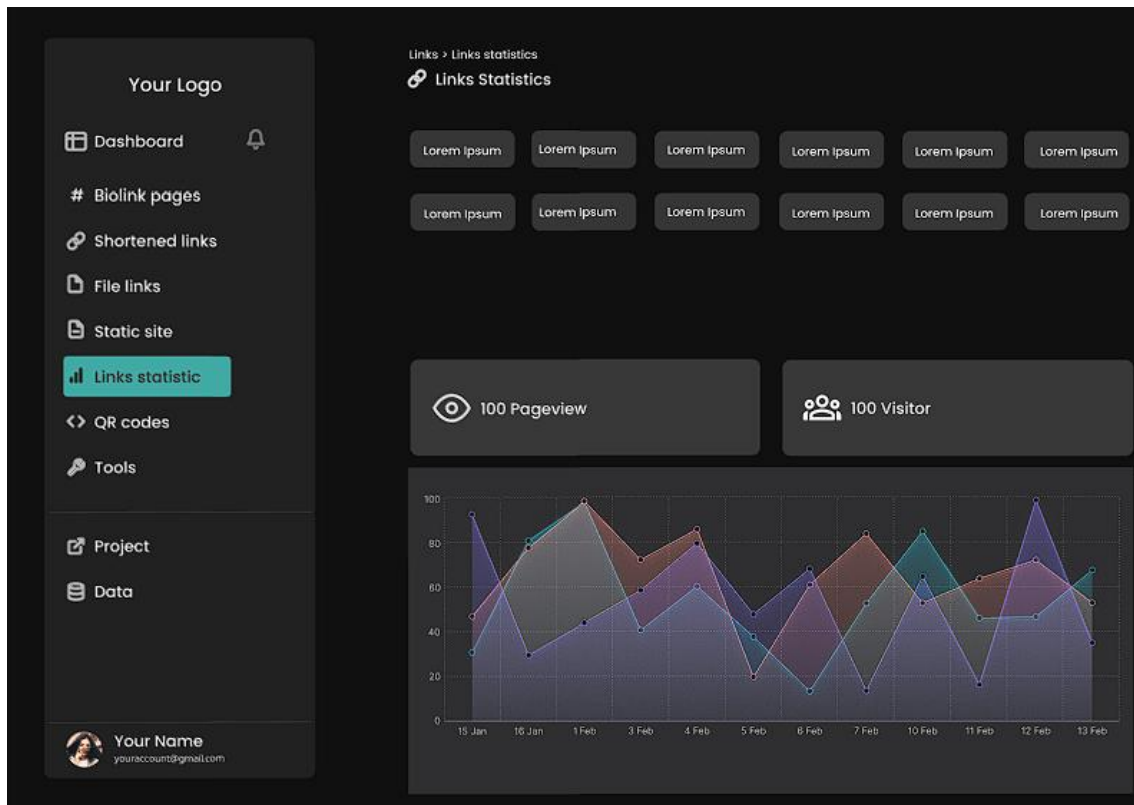
c) Halaman Link



Gambar 3. 5. Halaman Link

Pada halaman ini, pengguna dapat mengelola berbagai tautan, baik short link, file link, maupun biolink. Tersedia fitur untuk menambah, mengedit, filter berdasarkan jenis dan menghapus link, serta menampilkan status aktif atau nonaktif dari setiap tautan.

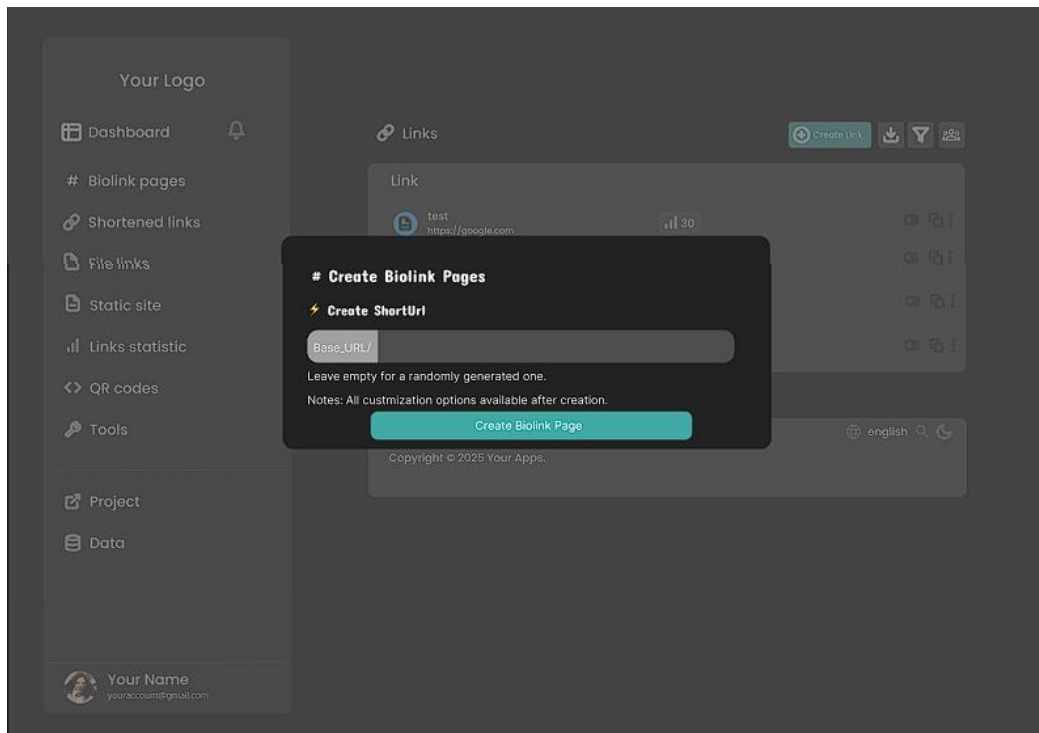
d) Halaman Statistik



Gambar 3. 6. Halaman Statistik

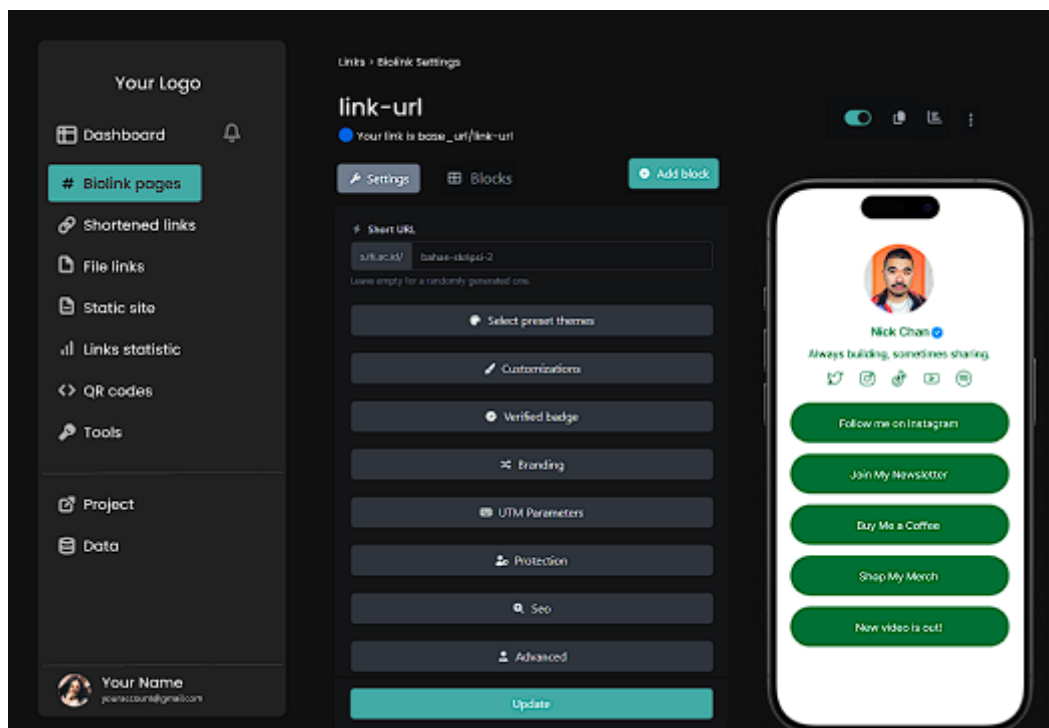
Menampilkan data analitik berupa grafik interaktif yang menggambarkan jumlah pengunjung, lokasi, perangkat, serta waktu akses tautan. Tujuan halaman ini adalah membantu pengguna memahami performa link yang telah dibuat.

e) Halaman Biolink



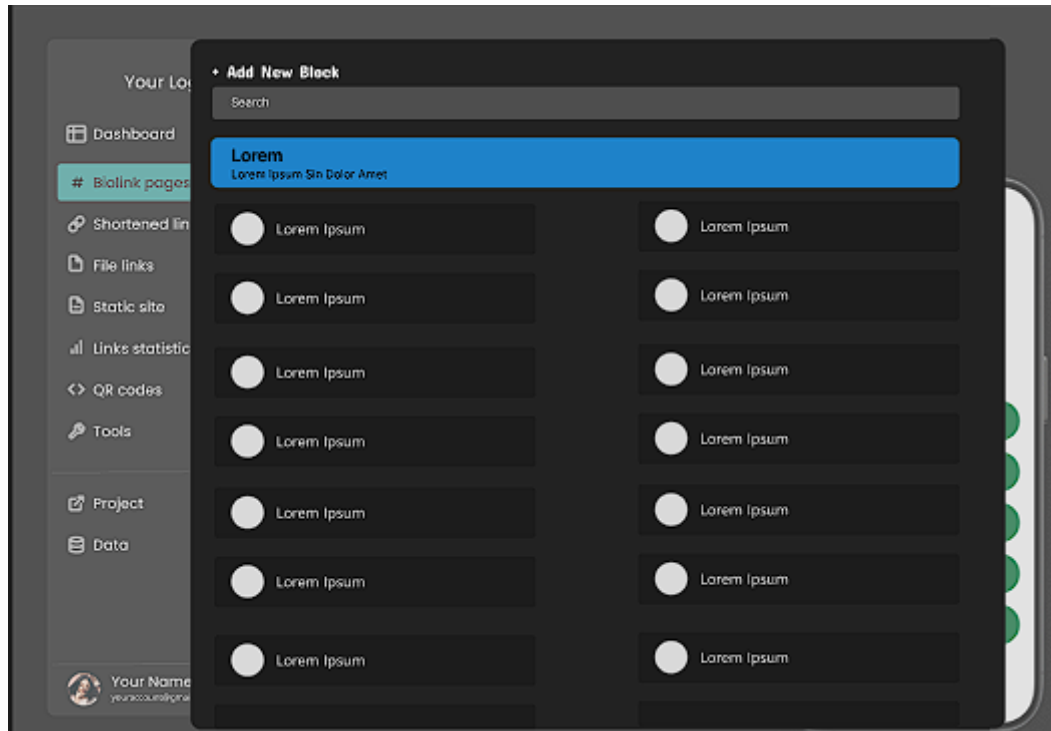
Gambar 3. 7. Halaman Buat Biolink

Halaman ini digunakan untuk membuat biolink tautan yang menampung beberapa link dalam satu halaman profil.



Gambar 3. 8. Halaman biolink

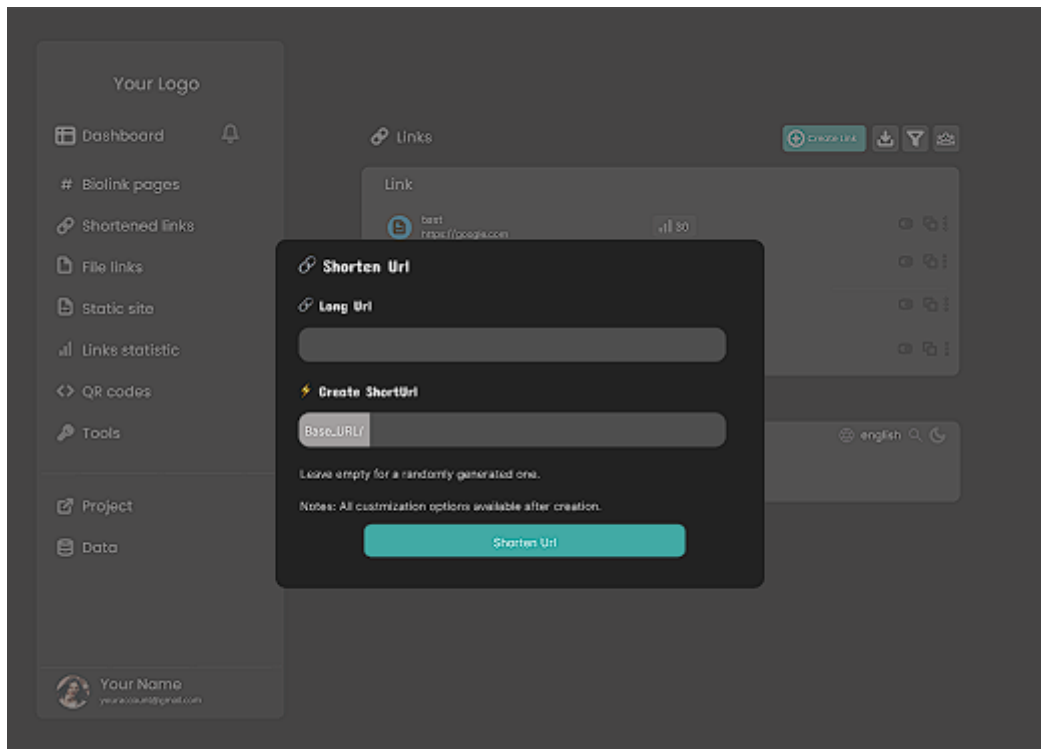
Halaman ini digunakan untuk mengelola biolink tautan yang menampung beberapa link dalam satu halaman profil. Pengguna dapat membuat dan menyesuaikan tampilan biolink dengan menambahkan elemen seperti tombol, teks, dan media.



Gambar 3. 9. Halaman Blok Biolink

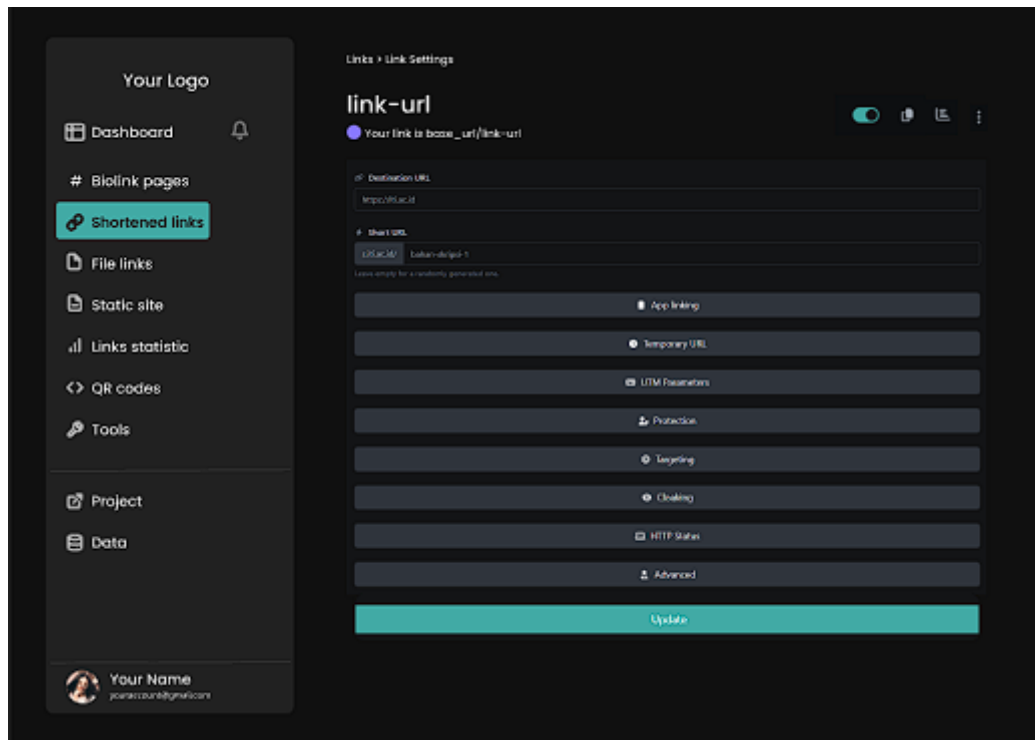
Berfungsi untuk menambah atau mengatur blok konten pada biolink, seperti tombol sosial media, video, gambar, atau teks. Pengguna dapat menyusun urutan blok sesuai kebutuhan melalui tampilan interaktif dan drag-and-drop.

f) Halaman Shortlink



Gambar 3. 10. Halaman Buat Link Pendek

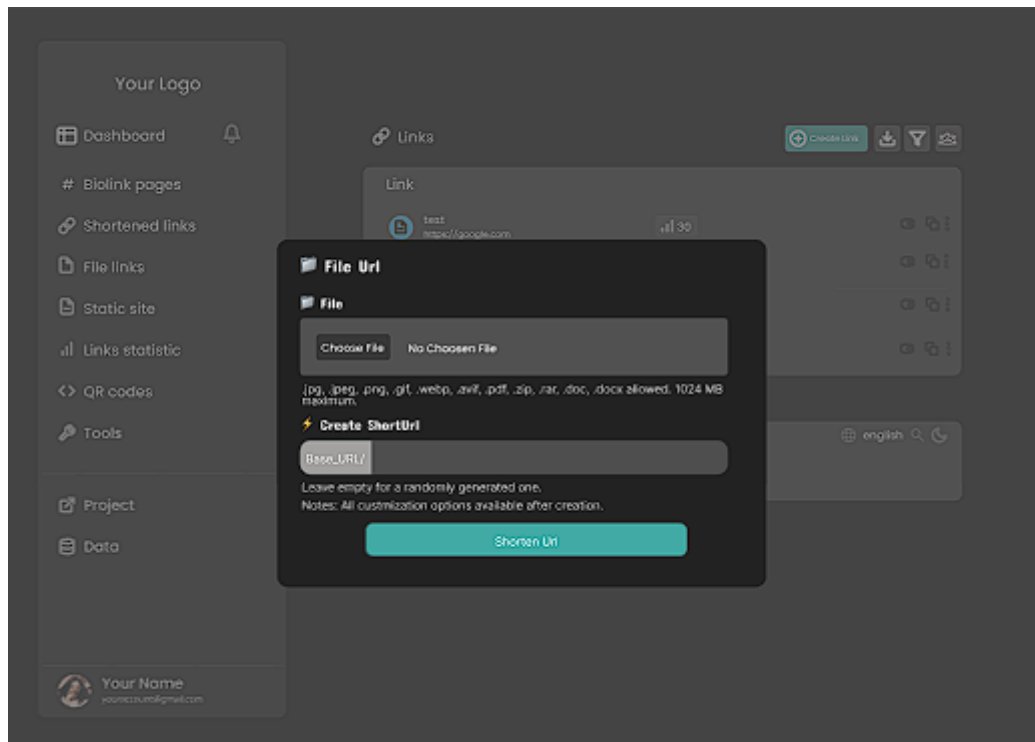
Halaman ini digunakan untuk membuat *short url* yang menampung link tujuan dalam satu halaman profil.



Gambar 3. 11. Halaman Link Pendek

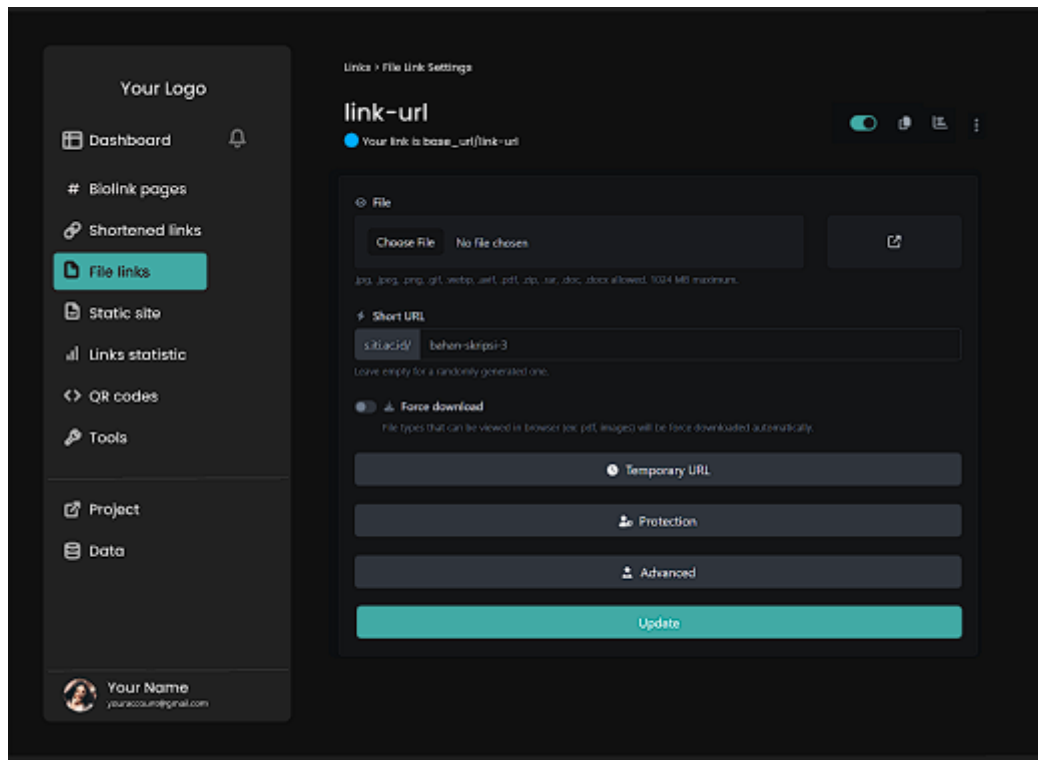
Fitur ini digunakan untuk mempersingkat URL panjang menjadi link yang lebih ringkas dan mudah dibagikan. Desain form dibuat ringkas agar pengguna dapat melakukan konversi URL dengan cepat.

g) Halaman File link



Gambar 3. 12. Halaman buat file link

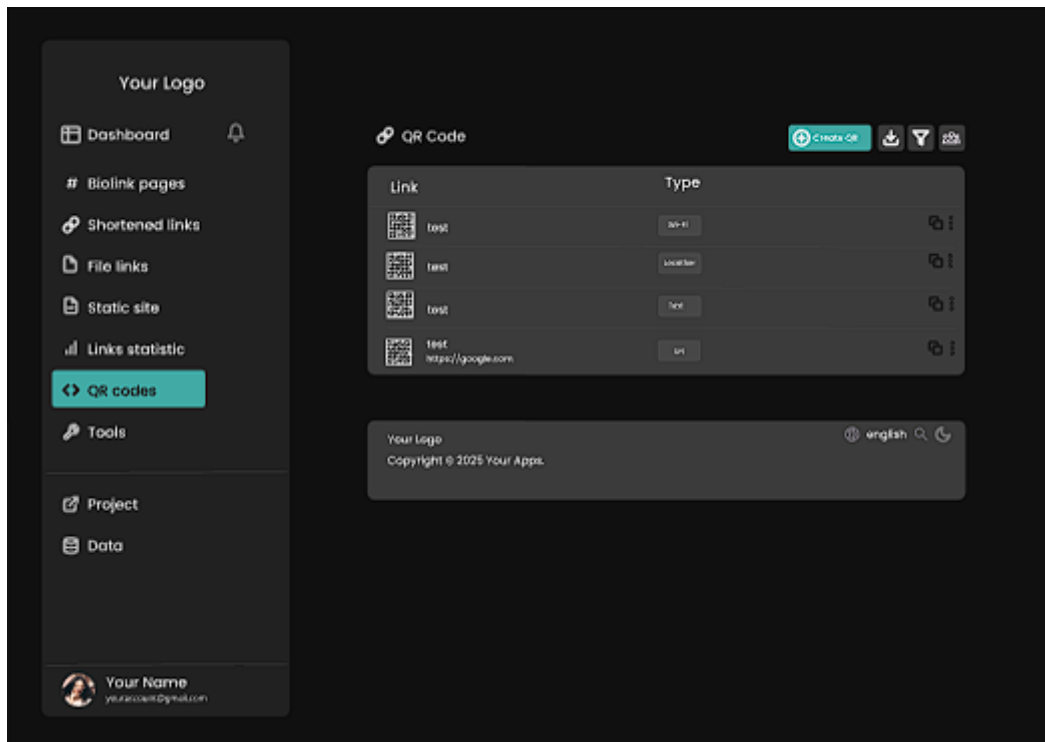
Halaman ini digunakan untuk membuat *short url* yang menampung file dalam sistem.



Gambar 3. 13. Halaman file link

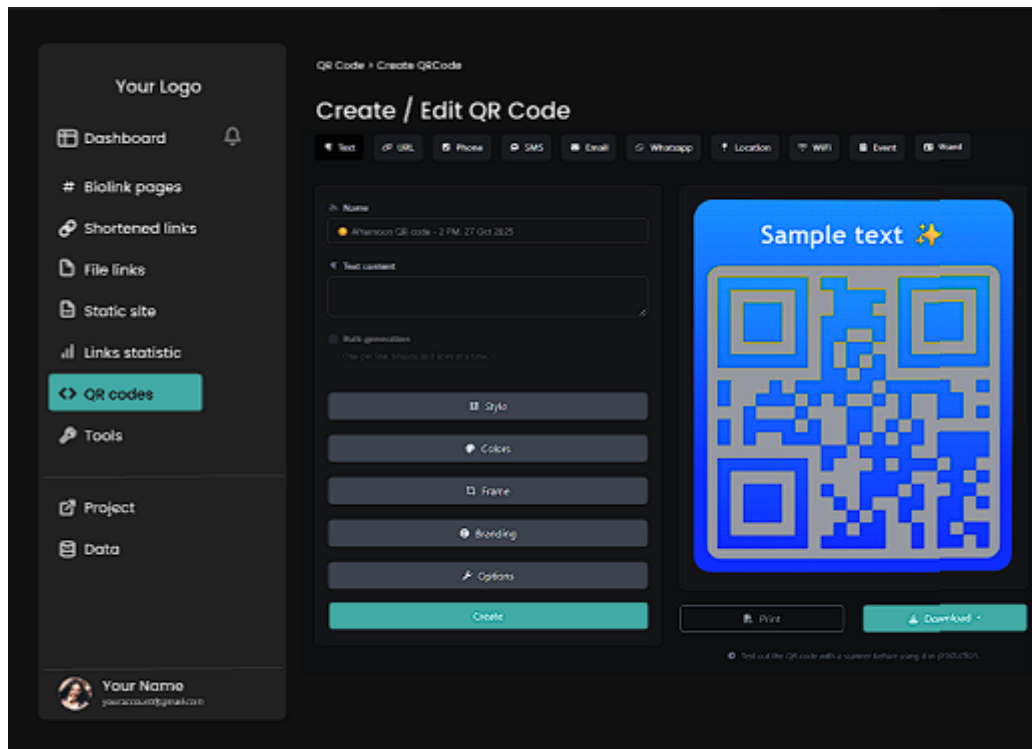
Fitur ini digunakan untuk menyimpan file menjadi link yang lebih ringkas dan mudah dibagikan. Desain form dibuat ringkas agar pengguna dapat melakukan manajemen file dengan cepat.

h) Halaman QR Code



Gambar 3. 14. Halaman List Qr Code

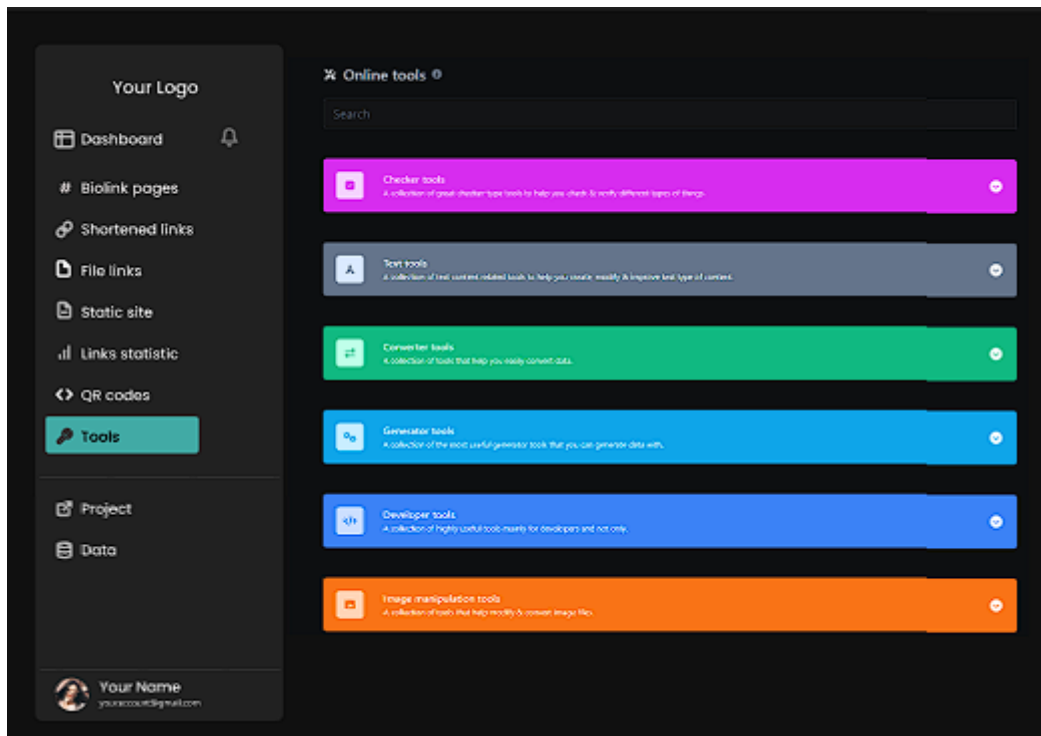
Menyediakan fitur untuk menghasilkan QR Code dari tautan yang dibuat. Pengguna dapat menyesuaikan ukuran, warna, serta format gambar QR sebelum menyimpannya. Halaman ini juga menampilkan daftar QR Code yang telah dibuat sebelumnya.



Gambar 3. 15. Halaman QR Code

Fitur ini digunakan untuk membuat QrCode secara statik maupun dynamic lebih beragam dan mudah dibagikan.

i) Halaman Tools Online/Utilitas Online



Gambar 3. 16. Halaman Tools Online

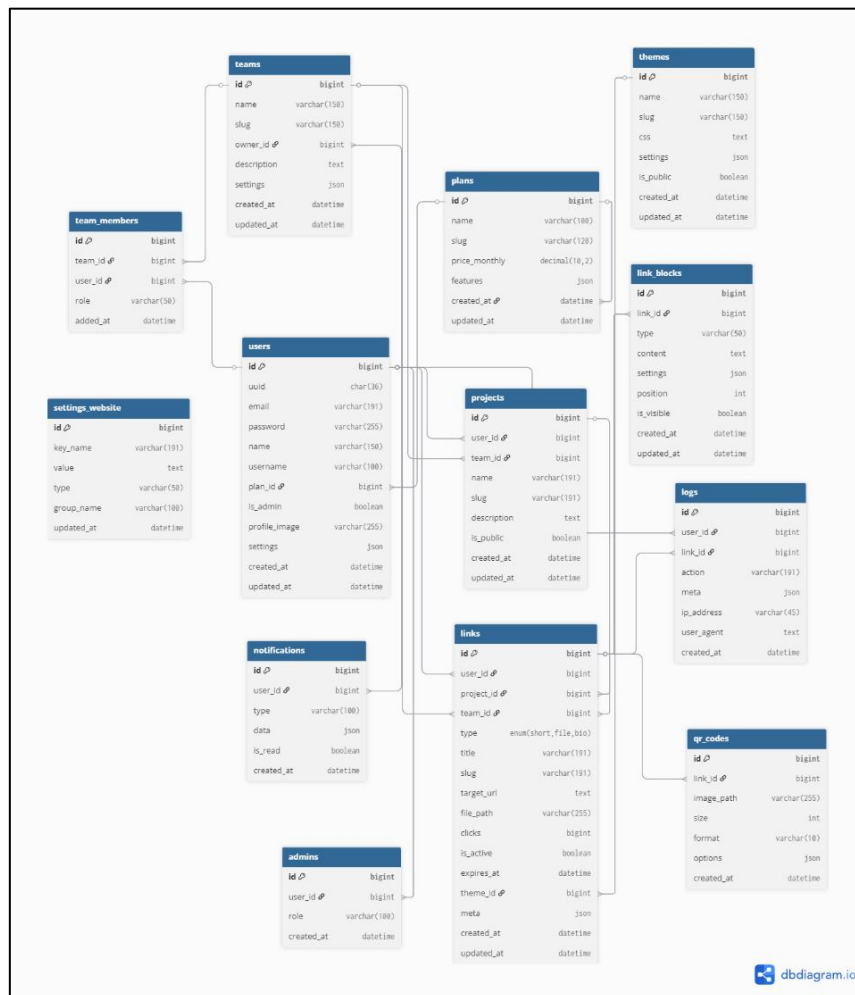
Fitur ini digunakan untuk users yang ingin menggunakan tools yang bekerja secara *client-side* sehingga tidak membebani kinerja server.

3.4.3. Entity Relation Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang menggambarkan hubungan antar entitas dalam sistem basis data. ERD digunakan untuk memodelkan struktur data dan hubungan antar tabel yang dibutuhkan agar sistem dapat berjalan dengan baik. Melalui ERD, pengembang dapat memahami bagaimana data saling berinteraksi, sehingga memudahkan proses perancangan maupun implementasi basis data di tahap berikutnya.

Pada penelitian ini, ERD menggambarkan hubungan antara entitas utama seperti User, Team, Project, Link, QR Code, dan Theme. Masing-masing entitas memiliki atribut dan relasi yang mendukung proses manajemen tautan, pembuatan QR Code, dan pengelolaan konten biolink. Secara umum, setiap pengguna (User) dapat memiliki banyak proyek (Project) atau tim (Team), di mana di dalamnya terdapat berbagai tautan (Link) yang dapat berupa short link, file link, atau biolink. Setiap tautan juga dapat memiliki blok konten (Link Block) dan QR Code yang dihasilkan secara dinamis. Selain itu, entitas

seperti Plan, Notification, dan Log mendukung aktivitas administratif dan pelacakan sistem.



Gambar 3. 17. Entity Relationship Diagram

Penjelasan Gambar 3.17:

1. **Entitas *User***, Menyimpan data pengguna aplikasi seperti nama, email, kata sandi, dan rencana langganan (*Plan*). Setiap pengguna dapat memiliki lebih dari satu *Project* dan *Link*.
2. **Entitas *Plan***, Berisi informasi paket langganan yang menentukan fitur yang dapat diakses oleh pengguna. Satu *Plan* dapat dimiliki oleh banyak *User*.
3. **Entitas *Team* dan *Team Member***, *Team* digunakan untuk kolaborasi antar pengguna. Satu tim dapat memiliki beberapa anggota (*Team Member*) yang masing-masing memiliki peran (role) tertentu.

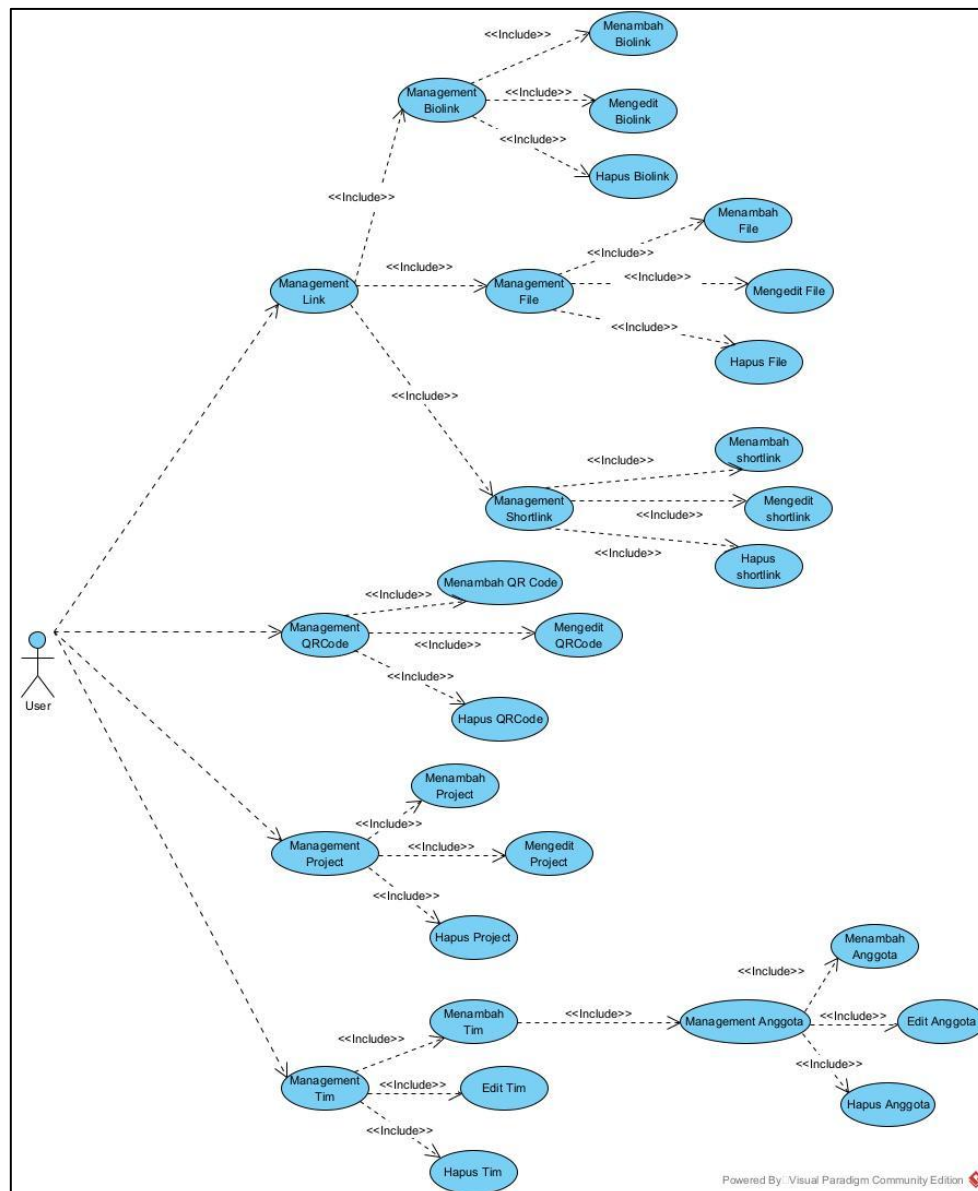
4. **Entitas *Project***, Mewakili kumpulan tautan yang dimiliki oleh seorang pengguna atau tim. Satu *Project* dapat berisi banyak *Link*.
5. **Entitas *Link***, Merupakan inti dari sistem yang dapat berupa *short link*, *file link*, atau *biolink*. *Link* memiliki relasi dengan *User*, *Project*, *Team*, dan *Theme*.
6. **Entitas *Link Block***, Digunakan untuk menyimpan elemen konten pada *biolink* seperti tombol, teks, atau video. Satu *Link* dapat memiliki beberapa *Link Block*.
7. **Entitas *QR Code***, Menyimpan data hasil generate QR Code dari suatu tautan. Setiap *Link* memiliki satu *QR Code* yang berisi pengaturan seperti ukuran dan format gambar.
8. **Entitas *Theme***, Menyimpan desain tampilan biolink yang bisa digunakan oleh beberapa tautan. Pengguna dapat memilih tema sesuai preferensi.
9. **Entitas *Log dan Notification***, *Log* berfungsi mencatat aktivitas pengguna, sedangkan *Notification* memberikan pemberitahuan terkait sistem, misalnya link kedaluwarsa atau perubahan plan.
10. **Entitas *Setting Website***, Berisi konfigurasi global sistem seperti pengaturan tampilan, API, dan metadata lainnya.

Dengan adanya ERD ini, hubungan antar tabel menjadi lebih jelas dan terstruktur sehingga proses implementasi ke dalam basis data MySQL dapat dilakukan dengan lebih efisien. Diagram ini juga memudahkan dalam proses normalisasi dan pengujian struktur data.

3.4.4. *Unified Modeling Language (UML)*

3.4.4.1. *Use Case Diagram Sistem*

1. *Use Case Diagram Sistem User*



Gambar 3. 18 Use Case Diagram User

Use Case Diagram pada Gambar 3.18 menggambarkan interaksi antara aktor User dengan seluruh fungsi yang tersedia pada sistem manajemen link, biolink, file, shortlink, QR Code, project, tim, dan anggota. Diagram ini memperlihatkan bahwa aktor utama, yaitu User, memiliki hak penuh untuk melakukan pengelolaan (management) terhadap berbagai komponen sistem. Setiap aktivitas pengelolaan terdiri dari tiga fungsi utama, yaitu menambah data (create), mengedit data (update), dan menghapus data (delete), yang masing-masing dihubungkan menggunakan relasi <<include>> untuk

menunjukkan bahwa proses-proses tersebut merupakan bagian yang tidak terpisahkan dari aktivitas manajemen utama.

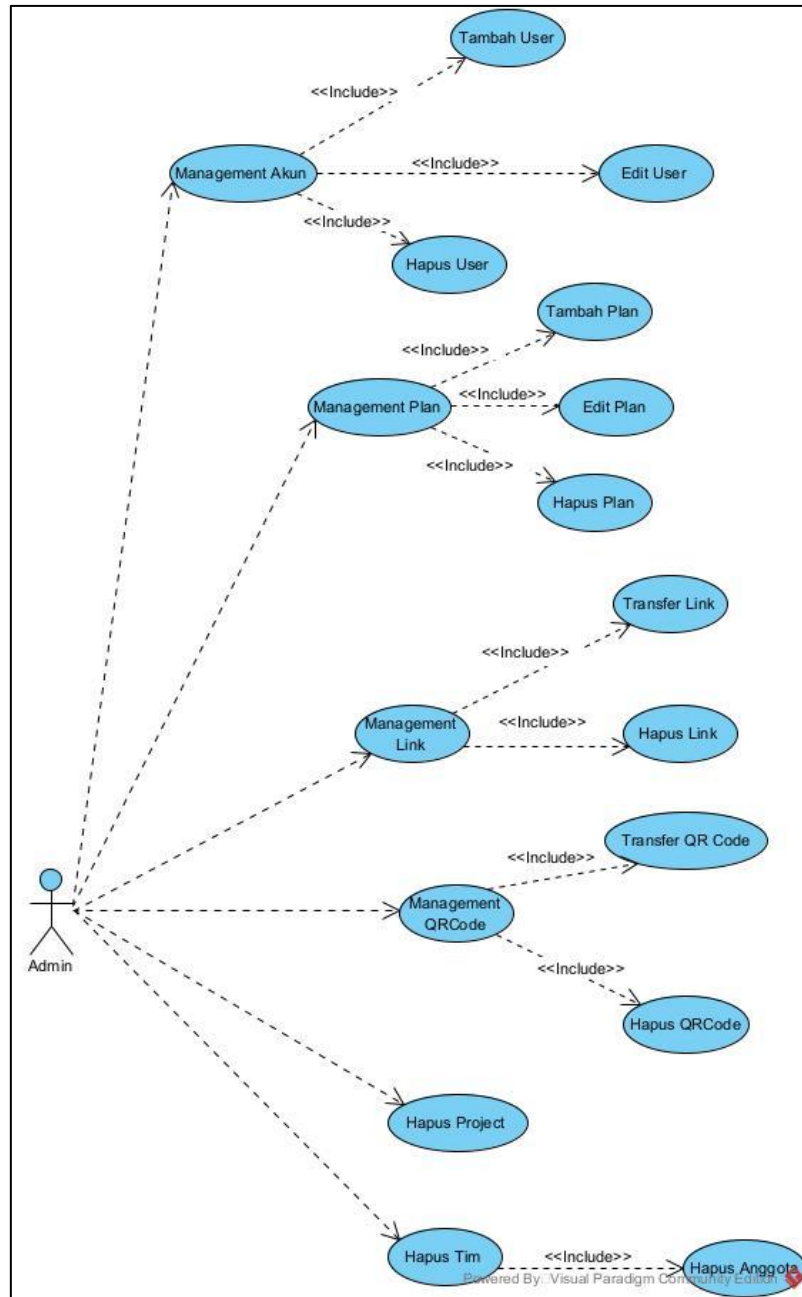
Sistem ini diawali dengan use case Management Link, yang berfungsi sebagai pusat pengelolaan berbagai jenis tautan. Di dalamnya, terdapat tiga sub-fungsi, yaitu Management Biolink, Management File, dan Management Shortlink. Setiap sub-fungsi memiliki kemampuan untuk menambah, mengedit, dan menghapus data sesuai jenisnya. Misalnya, pada Management Biolink, User dapat membuat biolink baru yang berisi beberapa tautan, memperbarui informasi biolink yang telah dibuat, maupun menghapus biolink yang sudah tidak diperlukan. Hal yang sama berlaku pada Management File, di mana User dapat menambah file baru, mengedit file yang telah diunggah, serta menghapus file dari sistem. Sementara itu, pada Management Shortlink, User dapat membuat tautan pendek baru, melakukan perubahan terhadap tautan pendek yang sudah ada, serta menghapusnya apabila tidak lagi digunakan.

Selain itu, sistem juga menyediakan fitur Management QRCode, yang mencakup fungsi menambah, mengedit, dan menghapus QR Code. Fitur ini memungkinkan User untuk membuat QR Code yang terhubung ke link atau halaman tertentu, serta memperbarui atau menghapusnya sesuai kebutuhan. Selanjutnya terdapat Management Project, yang berfungsi untuk mengelola proyek-proyek yang dibuat oleh User. Di dalamnya, User dapat menambah project baru, mengedit detail project, maupun menghapus project yang sudah selesai atau tidak lagi relevan.

Pada bagian lain dari sistem, terdapat Management Tim yang memungkinkan User untuk membuat tim kerja baru, mengedit informasi tim, serta menghapus tim yang sudah tidak aktif. Use case ini juga terhubung dengan Management Anggota, di mana User dapat menambah anggota ke dalam tim, memperbarui data anggota, maupun menghapus anggota dari daftar. Relasi <<include>> antara Management Tim dan Management Anggota menunjukkan bahwa setiap proses pengelolaan tim mencakup juga proses pengelolaan anggotanya.

Secara keseluruhan, diagram ini menjelaskan bahwa sistem dirancang untuk memberikan kemudahan bagi User dalam melakukan pengelolaan berbagai komponen digital secara terintegrasi, mulai dari link, file, dan QR Code, hingga ke pengaturan project, tim, dan anggota. Struktur hubungan <<include>> digunakan secara konsisten untuk menunjukkan keterkaitan antar proses yang bersifat wajib dalam satu alur aktivitas. Dengan demikian, Use Case Diagram ini menggambarkan ruang lingkup fungsionalitas sistem secara menyeluruh dan menjadi dasar bagi perancangan aktivitas serta alur sistem pada tahap selanjutnya.

2. *Use Case* Diagram Sistem Admin



Gambar 3. 19. Use Case Diagram Admin

Use Case Diagram pada Gambar 3.19 menggambarkan interaksi antara aktor Admin dengan seluruh fungsi sistem yang berhubungan dengan pengelolaan akun pengguna, paket layanan (plan), link, QR Code, serta penghapusan data tertentu seperti proyek, tim, dan anggota. Diagram ini menunjukkan bahwa Admin memiliki hak akses penuh terhadap data dan aktivitas di dalam sistem, termasuk hak untuk menambah, mengubah, memindahkan, dan menghapus data pengguna maupun komponen lainnya.

Setiap fungsi utama dalam diagram dipecah menjadi beberapa use case yang saling terhubung dengan relasi <<include>>, menandakan bahwa fungsi tersebut merupakan bagian dari proses manajemen yang lebih besar.

Fungsi pertama yang dapat diakses oleh Admin adalah Management Akun, yang digunakan untuk melakukan pengelolaan data pengguna sistem. Di dalamnya terdapat tiga use case utama, yaitu Tambah User, Edit User, dan Hapus User. Melalui fungsi ini, Admin dapat menambahkan akun baru untuk pengguna, mengubah data pengguna yang sudah ada (misalnya memperbarui email, nama, atau status akun), serta menghapus akun pengguna yang tidak aktif atau melanggar kebijakan sistem. Relasi <<include>> menunjukkan bahwa ketiga aktivitas tersebut merupakan bagian integral dari proses manajemen akun.

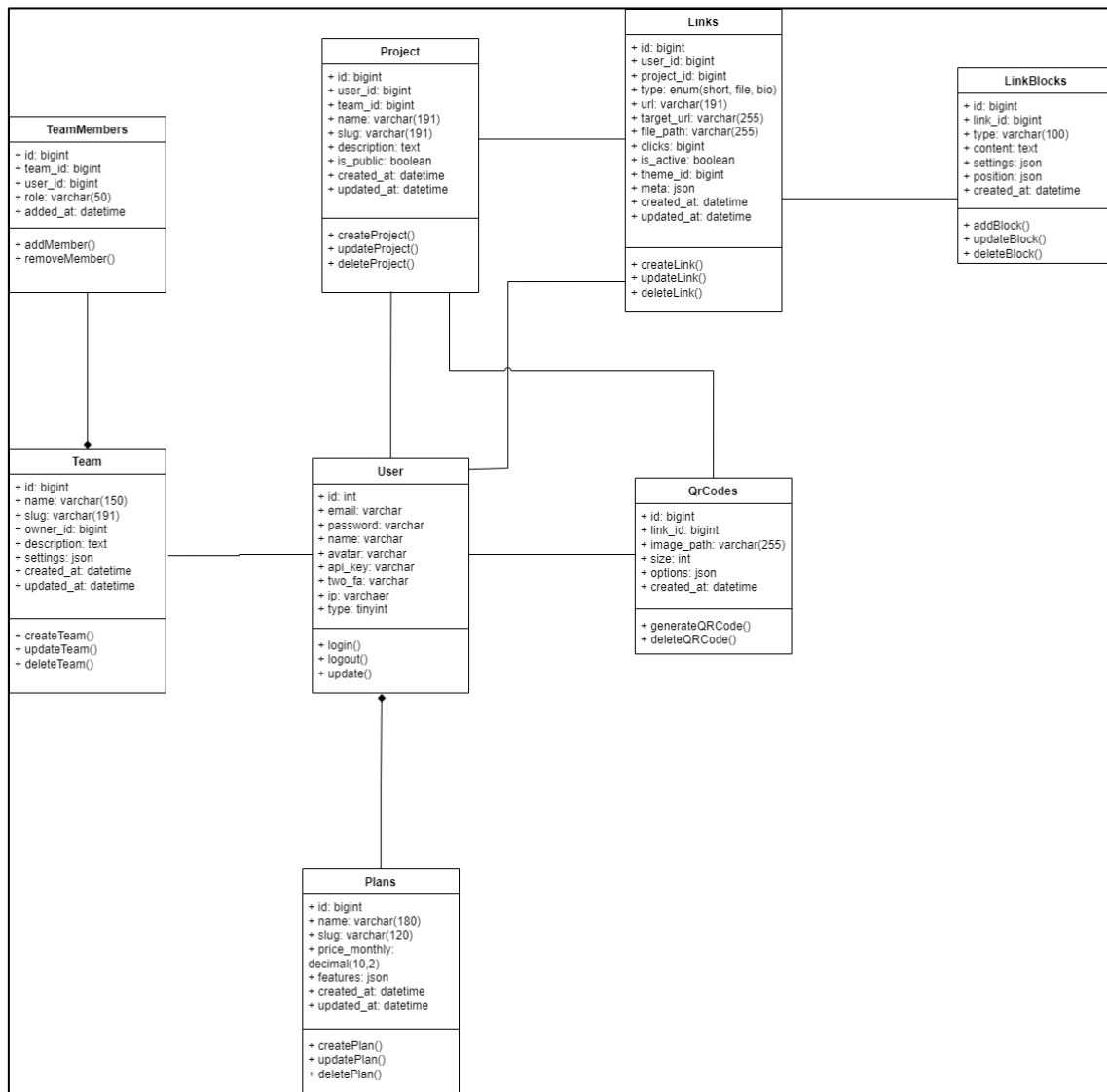
Selanjutnya terdapat fungsi Management Plan, yang berperan dalam pengelolaan paket atau rencana layanan (plan) yang tersedia di sistem. Admin dapat menambah plan baru, melakukan perubahan terhadap plan yang sudah ada, serta menghapus plan yang sudah tidak digunakan. Masing-masing fungsi tersebut digambarkan dalam use case Tambah Plan, Edit Plan, dan Hapus Plan, yang semuanya terhubung ke use case utama Management Plan melalui relasi <<include>>. Dengan demikian, pengelolaan plan dilakukan secara terstruktur dan dapat disesuaikan dengan kebutuhan sistem atau kebijakan layanan yang berlaku.

Selain itu, Admin juga memiliki kemampuan untuk melakukan pengelolaan terhadap link dan QR Code. Pada use case Management Link, Admin dapat menghapus link tertentu melalui use case Hapus Link, maupun memindahkan atau mentransfer link kepada pengguna lain menggunakan use case Transfer Link. Begitu pula dengan Management QRCode, di mana Admin dapat melakukan penghapusan QR Code (Hapus QRCode) serta melakukan proses pemindahan QR Code ke akun lain melalui use case Transfer QR Code. Kedua fungsi ini menunjukkan bahwa Admin tidak hanya bertanggung jawab untuk pengawasan data, tetapi juga dapat melakukan tindakan administratif yang berkaitan dengan redistribusi aset digital antar pengguna.

Selain aktivitas manajemen di atas, Admin juga memiliki hak untuk melakukan penghapusan langsung terhadap entitas lain yang terdapat di dalam sistem, seperti proyek, tim, dan anggota. Use case Hapus Project memungkinkan Admin untuk menghapus proyek yang sudah tidak aktif atau melanggar ketentuan. Sementara use case Hapus Tim digunakan untuk menghapus data tim tertentu, dan melalui relasi <<include>>, use case ini juga mencakup fungsi Hapus Anggota, yang menunjukkan bahwa penghapusan tim secara otomatis akan mencakup penghapusan anggota yang tergabung di dalamnya.

Secara keseluruhan, diagram ini memperlihatkan bahwa Admin berperan sebagai pengelola utama sistem dengan wewenang untuk melakukan kontrol penuh terhadap seluruh entitas data. Hubungan <<include>> dalam diagram digunakan untuk memperjelas bahwa fungsi-fungsi turunan seperti tambah, edit, hapus, dan transfer merupakan bagian dari proses manajemen utama yang tidak dapat dipisahkan. Dengan demikian, Use Case Diagram Sistem Admin ini menjadi representasi struktur fungsional sistem yang menunjukkan cakupan dan tanggung jawab administrator dalam menjaga keteraturan, keamanan, dan integritas data di dalam sistem.

3.4.4.2. Class Diagram

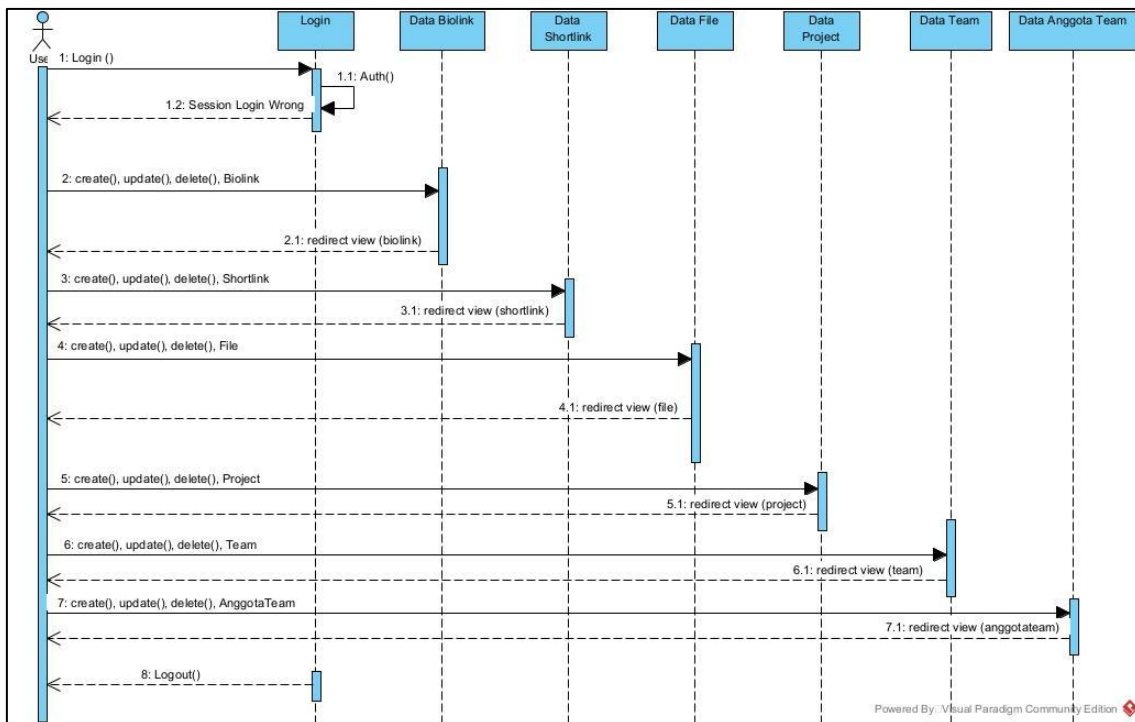


Gambar 3. 20. Class Diagram

Gambar 3.20 Class Diagram menggambarkan sistem manajemen proyek dan tautan digital yang terintegrasi, di mana pengguna dapat membuat tim, mengelola proyek, membuat tautan (link), menambahkan blok konten, serta menghasilkan kode QR. Diagram ini juga menunjukkan keterkaitan antara langganan pengguna (plans) dan kolaborasi tim (team & members).

3.4.4.3. Sequence Diagram

1. Sequence Diagram User



Gambar 3. 21. Sequence Diagram User

Sequence Diagram pada Gambar 3.X menggambarkan urutan interaksi antara aktor User dengan berbagai komponen sistem dalam proses pengelolaan data, mulai dari tahap login hingga logout. Diagram ini menunjukkan aliran pesan (message) secara berurutan dari atas ke bawah yang menggambarkan bagaimana User berinteraksi dengan beberapa objek utama dalam sistem, yaitu Login, Data Biolink, Data Shortlink, Data File, Data Project, Data Team, dan Data Anggota Team. Setiap objek mewakili modul atau entitas data yang dapat dikelola oleh pengguna, sedangkan garis vertikal pada diagram menunjukkan lifeline dari masing-masing objek selama proses berlangsung.

Proses dimulai ketika User melakukan aksi Login() ke dalam sistem. Sistem kemudian memanggil proses otentikasi melalui pesan Auth() (1.1) untuk memverifikasi kredensial pengguna. Apabila informasi login tidak valid, sistem akan menampilkan pesan Session Login Wrong (1.2) yang menandakan bahwa proses autentikasi gagal. Namun, apabila proses autentikasi berhasil, User dapat melanjutkan ke tahap berikutnya untuk mengelola berbagai data yang tersedia di sistem.

Selanjutnya, User dapat melakukan operasi `create()`, `update()`, `delete()` terhadap beberapa jenis data. Proses pertama ditunjukkan oleh interaksi pada modul Data Biolink, di mana pengguna dapat menambah, memperbarui, atau menghapus data biolink sesuai kebutuhan. Setelah operasi selesai, sistem akan menampilkan tampilan terbaru melalui proses `redirect view (biolink)` (2.1) sebagai umpan balik visual bagi pengguna. Proses serupa terjadi pada modul Data Shortlink, di mana pengguna dapat melakukan pengelolaan data shortlink melalui operasi `create()`, `update()`, `delete()`, kemudian sistem mengarahkan tampilan ke halaman hasil pembaruan melalui `redirect view (shortlink)` (3.1).

Proses berikutnya berlanjut pada modul Data File, di mana User dapat melakukan tindakan pengelolaan file, baik menambah file baru, memperbarui file yang sudah ada, maupun menghapus file dari sistem. Setelah proses berhasil dilakukan, sistem mengirimkan respon berupa `redirect view (file)` (4.1) yang menampilkan daftar file terbaru. Proses yang sama juga diterapkan pada modul Data Project, Data Team, dan Data Anggota Team, yang masing-masing memiliki fungsi pengelolaan data dengan pola operasi `create()`, `update()`, `delete()`, diikuti oleh pengalihan tampilan atau `redirect view` untuk memastikan pengguna melihat hasil perubahan data secara langsung.

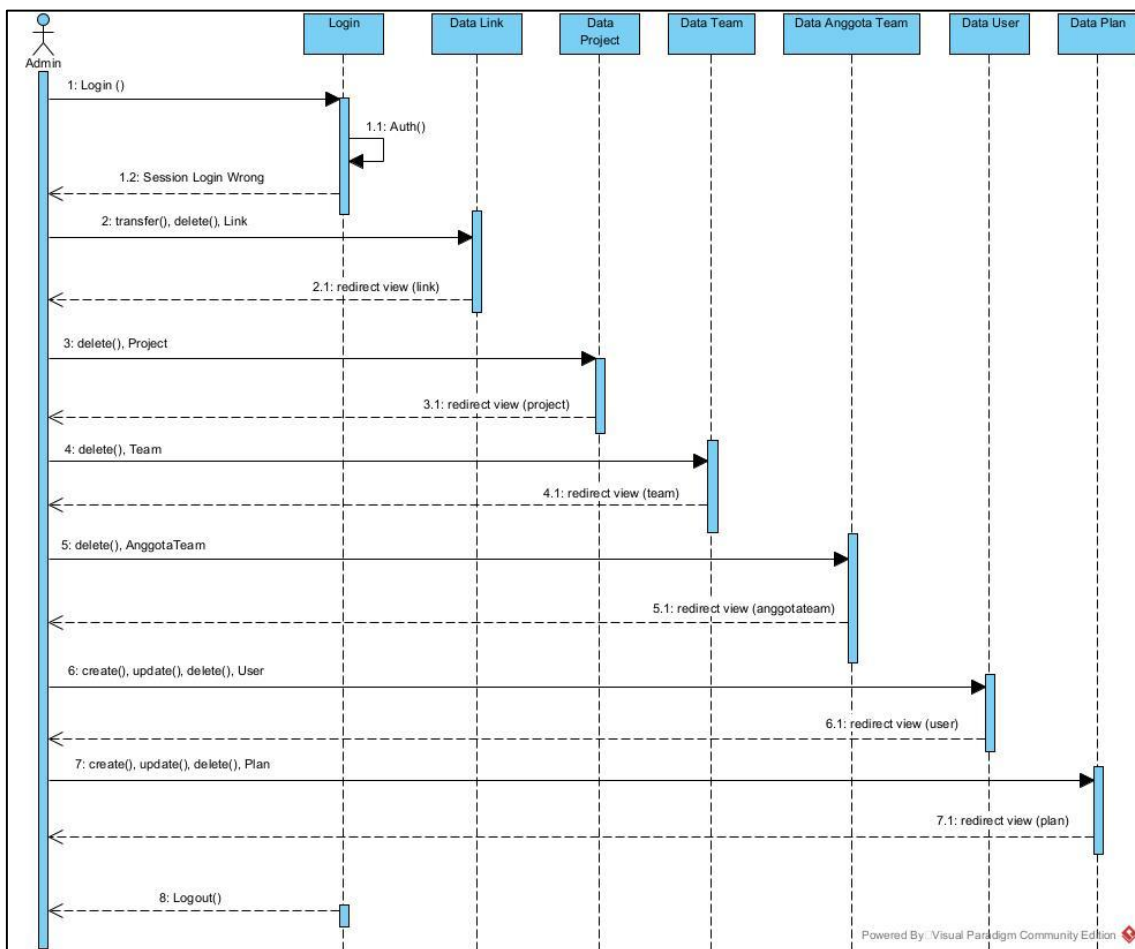
Lebih lanjut, pada Data Project, sistem akan melakukan `redirect` setelah operasi selesai melalui `redirect view (project)` (5.1). Hal ini menandakan bahwa setiap perubahan data proyek langsung disinkronkan ke tampilan antarmuka pengguna. Pada Data Team, proses serupa terjadi dengan pesan `redirect view (team)` (6.1), sedangkan pada Data Anggota Team, sistem juga melakukan pengalihan tampilan melalui `redirect view (anggotateam)` (7.1) setelah proses penambahan, pembaruan, atau penghapusan data anggota dilakukan. Keseluruhan alur ini memperlihatkan bahwa sistem bekerja secara sinkron dan setiap perubahan data langsung dikembalikan dalam bentuk tampilan yang diperbarui.

Sebagai tahap akhir, User melakukan aksi `Logout()` (8), yang menandakan bahwa sesi pengguna telah berakhir dan sistem menghentikan semua aktivitas yang berkaitan dengan akun tersebut. Proses `logout` ini juga

berfungsi sebagai mekanisme keamanan untuk mencegah akses tidak sah setelah sesi penggunaan selesai.

Secara keseluruhan, Sequence Diagram ini menunjukkan bagaimana sistem merespons setiap tindakan pengguna secara berurutan dan terkoordinasi. Hubungan antar objek dalam diagram mencerminkan komunikasi logis antara antarmuka pengguna, modul pengelolaan data, dan proses kontrol sistem. Dengan demikian, diagram ini memberikan gambaran jelas mengenai dinamika interaksi yang terjadi selama penggunaan sistem, serta memastikan bahwa setiap fungsi berjalan secara konsisten dan sesuai dengan urutan logika yang telah dirancang.

2. Sequence Diagram Admin



Gambar 3. 22. Sequence Diagram Admin

Proses dimulai ketika Admin melakukan login ke dalam sistem. Sistem kemudian memanggil fungsi Auth() untuk melakukan autentikasi. Jika login

gagal, sistem menampilkan pesan “Session Login Wrong”. Setelah berhasil login, admin dapat melakukan berbagai operasi pengelolaan data. Misalnya, admin dapat melakukan transfer, penghapusan, atau pengelolaan link melalui fungsi `transfer()`, `delete()`, dan sistem akan menampilkan halaman `redirect view (link)` sebagai umpan balik.

Selanjutnya, admin dapat menghapus data project, dan sistem akan memproses permintaan tersebut melalui `delete(Project)` lalu mengarahkan kembali ke tampilan proyek (`redirect view (project)`). Hal serupa juga berlaku untuk penghapusan data team dan data anggota team, di mana setiap tindakan `delete()` diikuti oleh proses pengalihan tampilan ke halaman yang sesuai (`redirect view (team)` dan `redirect view (anggotateam)`).

Admin juga memiliki hak untuk membuat, memperbarui, atau menghapus data user dan data plan melalui fungsi `create()`, `update()`, dan `delete()`. Setiap kali perubahan dilakukan, sistem akan menampilkan tampilan terbaru dengan `redirect view (user)` atau `redirect view (plan)` untuk memastikan data yang dimodifikasi terlihat oleh admin.

Terakhir, setelah seluruh proses manajemen data selesai, admin dapat melakukan logout dari sistem untuk mengakhiri sesi penggunaannya.

3.5. Pengembangan Sistem

Tahapan pengembangan sistem adalah langkah untuk menerapkan hasil dari analisis dan desain yang telah dilakukan sebelumnya. Dalam tahap ini, semua bagian dari sistem mulai dibuat sesuai dengan kebutuhan yang telah ditentukan, baik yang berhubungan dengan fungsi maupun yang tidak.

Sistem ini dibuat dengan menggunakan bahasa pemrograman PHP (tanpa menggunakan *framework*) dan menerapkan Model-View-Controller (MVC) sebagai cara utama dalam pengembangan. Selain itu, Bootstrap dipakai untuk memperindah tampilan antarmuka (bagian depan), sementara MySQL berfungsi sebagai tempat penyimpanan utama untuk mengelola informasi.

3.5.1. Model Pengembangan Model-View-Controller (MVC)

Model MVC dipilih karena memberikan struktur yang jelas, memisahkan logika bisnis, tampilan, dan pengelolaan data, sehingga pengembangan jadi lebih teratur dan mudah diatur. Berikut adalah penjelasan tentang tiap komponen:

a) Model

Komponen Model bertugas mengelola data dan logika bisnis dari sistem. Semua aktivitas yang berhubungan dengan database, seperti menyimpan, memperbarui, menghapus, dan mengambil data dari MySQL dilakukan di bagian ini. Contohnya mencakup pengelolaan data tautan pendek, biolink, kode QR, serta pencatatan statistik kunjungan pengguna.

b) View

Komponen View memiliki tugas untuk menampilkan data kepada pengguna. Tampilan dibuat menggunakan HTML, Bootstrap, dan JavaScript agar lebih interaktif dan responsif di berbagai perangkat. Desain antarmuka dirancang agar sederhana dan mudah digunakan oleh pengguna umum tanpa perlu pelatihan khusus.

c) Controller

Komponen Controller berfungsi sebagai penghubung antara Model dan View. Controller menangani permintaan dari pengguna, memprosesnya melalui Model, lalu mengembalikan hasilnya dalam bentuk tampilan View. Misalnya, ketika pengguna membuat tautan pendek, Controller akan menerima input, memvalidasi data, menyimpannya lewat Model, lalu menunjukkan hasilnya di View.

3.5.2. Bahasa Pemrograman dan Teknologi Pendukung

Dalam proses pengembangan, digunakan beberapa teknologi utama sebagai berikut:

Tabel 3. 1. TechStack

Komponen	Teknologi yang Digunakan	Fungsi Utama
Bahasa Pemrograman	PHP 8.4	Logika utama sistem dan pengelolaan proses server-side

Struktur Aplikasi	Model-View-Controller (MVC)	Pemisahan logika, tampilan, dan data untuk memudahkan pemeliharaan
Basis Data	MySQL	Penyimpanan data tautan, QR, biolink, file, dan statistik
Front-End	HTML, CSS, JavaScript, Bootstrap	Tampilan antarmuka yang responsif dan mudah digunakan
Server Lokal	XAMPP	Menyediakan lingkungan pengujian dan server Apache untuk PHP
Browser	Google Chrome, Mozilla Firefox	Media untuk menjalankan dan menguji aplikasi

3.6. Pengujian dan Penyelesaian

Pada tahapan ini, aplikasi yang telah dikembangkan diuji menggunakan metode *black-box* testing. Pengujian ini berfokus untuk memvalidasi kesesuaian fungsionalitas aplikasi dengan kebutuhan (*requirements*) serta mengidentifikasi kendala atau permasalahan yang ada. Proses pengujian *black-box* dilaksanakan dengan menjalankan unit program, kemudian mengamati hasil yang diperoleh untuk memastikan kesesuaiannya dengan proses bisnis yang diharapkan. Selain itu, dilakukan pula *User Acceptance Testing* (UAT) atau Uji Penerimaan Pengguna. Pengujian ini melibatkan pengguna akhir (*end-user*) secara langsung untuk memvalidasi bahwa aplikasi telah sesuai dengan alur kerja (*workflow*) bisnis di dunia nyata dan dapat diterima untuk digunakan. Temuan dari keseluruhan proses pengujian ini selanjutnya didokumentasikan dalam tabel pengujian.

BAB 4

PEMBAHASAN

4.1. Pengaturan Sistem

4.2. Tampilan Halaman

4.3. Pengujian BlackBox

4.4. UAT

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

5.2. Saran

DAFTAR REFERENSI (JANGAN LUPA FORMAT APA & CTRL + T)

Andriani, D. A., & Prasetyo, H. (2022). Rancang bangun aplikasi Lihat.in berbasis web di Fakultas Ilmu Komputer UPN “Veteran” Jawa Timur dengan algoritma Base64 serta implementasi metode QR-Code. *Jurnal Ilmu Komputer dan Teknologi Informasi*, 6(1), 12–21. <https://repository.upnjatim.ac.id/14662>

Bitly. (2022). Bitly company facts and statistics. <https://bitly.com>

Capriotti, P., Zeler, I., & Oliveira, A. (2023). University communication on social media: An analysis of organizational strategies. *Public Relations Review*, 49(1), Article 102117. <https://doi.org/10.1016/j.pubrev.2022.102117>

DataIntel. (2024). Link-in-bio market size, share & forecast report 2024–2032. <https://dataintel.com/report/link-in-bio-platform-market>

Fukushi, N., Koide, T., Chiba, D., Nakano, H., & Akiyama, M. (2021). Analyzing security risks of ad-based URL shortening services caused by users’ behaviors. In *International Conference on Security and Privacy in Communication Systems (SecureComm 2021)* (pp. 3–21). Springer. https://doi.org/10.1007/978-3-030-90022-9_1

Fukushi, N., Koide, T., Chiba, D., Nakano, H., & Akiyama, M. (2022). Understanding security risks of ad-based URL shortening services caused by users’ behaviors. *Journal of Information Processing*, 30, 865–876. <https://doi.org/10.2197/ipsjip.30.865>

Adhitama, R., & Jauhari, A. (2022). Rancang Bangun Tools URL Shortener Berbasis Website Menggunakan Framework Laravel (Studi Kasus Diskominfo Kalimantan Barat). *Jurnal Teknologi dan Sistem Informasi, Diskominfo Kalimantan Barat*.

Mota, J. S., Tives, H. A., & Canedo, E. D. (2021). Tool for measuring productivity in software development teams. *Information*, 12(10), 396. <https://doi.org/10.3390/info12100396>

Hidayat, R., & Sari, P. (2022). Aplikasi sistem informasi pemendek URL (SI SOUP) berbasis web. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 9(1), 89–98. <https://d1wqtxts1xzle7.cloudfront.net/79965164/4084-libre.pdf>

Langgam.id. (2025, Januari 24). [Platform Digital S.id Tembus 1,5 Juta Pengguna](#).
Langgam.id.

Linktree. (2023). Linktree company facts and usage statistics. <https://linktr.ee>

Permana, I. M., & Lestari, D. (2023). Perancangan prototipe UI/UX pada pemesanan menu di kedai kopi Rumiko berbasis mobile microsite. *Jurnal Wanarupa: Desain, Komunikasi Visual, dan Media*, 2(1), 34–42.
<https://jurnal.umt.ac.id/index.php/WARNARUPA/article/view/13694>

Ramadhan, A., & Susanto, H. (2021). Rancang bangun aplikasi pemandu wisata Museum Sumbawa berbasis Android dengan memanfaatkan quick response code (QR code). *Jurnal Informatika dan Teknologi Sistem (JINTEKS)*, 2(2), 101–110.
<https://www.jurnal.uts.ac.id/index.php/JINTEKS/article/view/596>

Reynolds, J., Samarin, N., Barnes, J., Judd, T., Mason, J., Bailey, M., & Egelman, S. (2020). Empirical measurement of systemic 2FA usability. In 29th USENIX Security Symposium (USENIX Security '20) (pp. 127–143). USENIX Association.
<https://www.usenix.org/system/files/sec20-reynolds.pdf>

Q. Xingguo dkk.. (2020). *Design and Implementation of Toolbox Based on Component Technology*, 835-838. <https://www.computer.org/csdl/proceedings-article/iccea/2020/09103839/1kesDoMfYAg>

StackOverflow. (2024). What is Bitly and how is it used? <https://stackoverflow.com>
<https://dibimbing-cdn.sgp1.cdn.digitaloceanspaces.com/media-strap/287291ec889bac46d683df95f95196e8.webp> (gambar 2.1)

Chen, Y., Huang, Z., & Lee, J. (2021). *Integrated Short URL System Based on Cloud Architecture*. *Journal of Web Engineering and Technology*, 19(3), 205–218.

Prajapati, R., & Jain, D. (2025). *URL Shortener Using Python Flask Framework*. *International Journal of Creative Research Thoughts (IJCRT)*, Volume 13, Issue 5, Paper ID: IJCRT2505498.

Tersedia di: <https://www.ijcrt.org/papers/IJCRT2505498.pdf>

Rif'ah, F., Nurcholis, A., & Mulyadi, D. (2024). *Perancangan Aplikasi Microsite Untuk Mengelola Link Pada Media Sosial Menggunakan Framework Laravel*. *Jurnal Inovasi*

Penelitian, Teknologi Informasi dan Komunikasi (JIPTIKA), 3(2), 45–52.
Tersedia di: <https://e-journal.ivet.ac.id/index.php/jiptika/article/view/3920>

Sulistyo, D. (2023). *Integrasi Microsite dalam Sistem Pelayanan Digital Publik*. Jakarta: CV. Media Sains Indonesia.
Tersedia di: https://www.google.co.id/books/edition/Integrasi_Microsite_dalam_Sistem_Pelayan/xTNoEQAAQBAJ?hl=en&gbpv=1&dq=microsite+buku&pg=PR3&printsec=frontcover

Gunawan, W., & Hidayat, R. (2021). *Implementasi QR-Code pada Aplikasi E-Market Mandiri untuk Pemberdayaan Ekonomi Kreatif Berbasis Android*. Tersedia di: https://www.researchgate.net/publication/349633308_Implementasi_QR-Code_pada_Aplikasi_E-Market_Mandiri_untuk_Pemberdayaan_Ekonomi_Kreatif_Berbasis_Android

Rachman, A. (2023). *Implementasi QR (Quick Response) Code pada Sertifikat Tanah Elektronik*. Bandung: Media Sains Indonesia.
Tersedia di: https://www.google.co.id/books/edition/Implementasi_Qr_Quick_Response_Code_pada/ArcpEQAAQBAJ

Taufik, M., Lee, D., & Park, S. (2022). *Integrated QR Payment System (QRIS): Cashless Payment Solution in Developing Country from Merchant Perspective*. *Asia Pacific Journal of Information Systems (APJIS)*, 32(3), 630–651.
Tersedia di: https://www.apjis.or.kr/pdf/APJIS_32_3_630.pdf

Ali, E. (2020). *Rekayasa Perangkat Lunak*. Tersedia di: https://www.researchgate.net/publication/342914703_Rekayasa_Perangkat_Lunak

MDPI. (2022). *Theories and Applications of Human-Computer Interaction*. Tersedia di: <https://www.mdpi.com/topics/U2530G5HII>

Suharyadi, B. (2023). *Sistem Utilitas*. Bandung: Penerbit Media Sains Indonesia.
Tersedia di: https://www.google.co.id/books/edition/SISTEM_UTILITAS/Gb1HEQAAQBAJ

IBM. (2024). What is a file transfer. <https://www.ibm.com/think/topics/file-transfer>

Webasha. (2025). Which Protocols Are Used for File Sharing? A 2025 Guide to FTP, SFTP, SMB, and More. <https://www.webasha.com/blog/which-protocols-are-used-for-file-sharing-guide-to-ftp-sftp-smb-and-more>

Usman K. (2024). *Enhancing File Transfer Security and Efficiency through Compression, Fragmentation and Reassembling Techniques*, Volume 186, No 39, Paper ID: IJCA 0975 – 8887

Redwood. (2024). Limitations of SFTP for modern data transfer use cases. <https://www.redwood.com/article/sftp-limitations>

Ikhsan, A. A., Mufti P., S., Listiyanto., & Safrudin. (2023). *UI/UX Developer dan perannya dalam pengembangan produk di industri digital*. JATIMIKA : Jurnal Kreativitas Mahasiswa Informatika, Vol. 4(1).

Kamba, M. I., & Dauda, A. (2025) - The Role of Multi-Factor Authentication (MFA) in Preventing Cyber Attacks. *International Journal of Research Publication and Reviews*, 6(7), 445-448.

Syahreen, M., Hafizah, N., Maarop, N., & Maslinan, M. (2024) - A Systematic Review on Multi-Factor Authentication Framework. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 15(5), 1043-1050.

Garfinkel, S. (2021) - The Challenges of Biometric Security. *Communications of the ACM*, 64(3), 27-29.

Ramcharan, H. (2025) - The Effective Integration of Multi-Factor Authentication (MFA) with Zero Trust Security. *American Journal of Mathematical and Computer Modelling*, 10(1), 1-5.

Triawan, A. (2021). Penerapan Application Programming Interface (API) Pada Push Notification Untuk Informasi Monitoring Stok Barang Minim. *Teknois. Jurnal. Ilmiah. Teknologi. Informasi. dan. Sains*, 11(2), 120-128. <https://doi.org/10.36350/jbs.v11i2.120>

Gopal, S. (2025). Building a Robust OAuth Token Based API Security: A High Level Overview. *arXiv Preprint*, 2507.16870, 1-28.

Ma'arif, M. R. (2010). Implementation of Model View Controller (MVC) Architecture in Web-Based Application. *SNATI: Seminar Nasional Aplikasi Teknologi Informasi*, 1-6. <https://doi.org/10.1109/icse.2010.5543579>

Sihombing, E. D. C. (2021). Penerapan Framework Model-View-Controller (MVC) Pada Sistem Informasi Manajemen Data Jemaat Berbasis Web (Studi Kasus GKI Maranatha Kampung Harapan). *Journal of Information System, Applied, Management, Accounting and Research (JISAMAR)*, 5(1), 152-160. <https://doi.org/10.52362/jisamar.v5i1.353>

Niarman, A., Iswandi, & Candri, A. K. (2023). Comparative Analysis of PHP Frameworks for Development of Academic Information System Using Load and Stress Testing. *International Journal of Software Engineering and Computer Science*, 3(3), 1-15. <https://doi.org/10.35870/ijsecs.v3i3.1850>

Siregar, U. K. (2024). Pengembangan Database Management System Menggunakan MySQL. *SAINTEK: Jurnal Sains, Teknologi & Komputer*, 5(1), 8-12. <https://doi.org/10.55123/storage.v4i2.5182>

LAMPIRAN

