

**TUGAS Software Requirements Specification
Frontiers Informations System Management
(Workingpaper)**

Mata kuliah : Analisis dan Perancangan Sistem Informasi

Dosen Pengampu : Dr. Nova Suparmanto M.Sc.



Disusun oleh :

Nama Anggota 1 : Sesia Nugroho Satriva (23051430031)

Nama Anggota 2 : M Lathif Ramadhani (23051430040)

PROGRAM STUDI TEKNIK INDUSTRI

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2025

Daftar Isi

| | |
|---|----|
| 1. Deskripsi Keseluruhan | 1 |
| 1.1 Deskripsi Produk..... | 1 |
| 1.2 Fungsi Produk | 1 |
| 1.3 Pengembangan produk | 3 |
| 1.4 Lingkungan Operasi | 6 |
| 1.5 Batasan dan Implementasi..... | 7 |
| 2. Deskripsi Kebutuhan Fungsional | 9 |
| 2.1 Proses Kegiatan..... | 9 |
| 2.2 Karakteristik Pengguna | 10 |
| 2.3 Use Case setiap Aktor | 11 |
| 2.4 Fungsi Silang..... | 14 |
| 2.5 Kebutuhan Perangkat Lunak dan Perangkat Keras..... | 15 |
| 3. .Desain dan Perancangan Sistem | 17 |
| 3.1 Site Map | 17 |
| 3.2 Use Case..... | 18 |
| 3.3 Activity Diagram..... | 19 |
| 3.4 Data Dictionary | 22 |
| 3.5 Data Flow Diagram (DFD) | 27 |
| 3.6 Entity Relationship Diagram (ERD) | 30 |
| 4. Database MySql | 36 |
| 4.1 Persiapan membuat database MySql..... | 36 |
| 4.2 Pembuatan database dan tabel data menggunakan metode <i>Graphical User Interface(GUI)</i> dengan PHPMyAdmin MySQL | 37 |
| 4.3 Struktur dan Ekspor coding MySQLPhp | 44 |
| 4.4 Membuat database dengan SQL Code | 45 |
| 5. Perancangan Antarmuka Pengguna (UI) dan Pengalaman Pengguna (UX) Menggunakan Figma..... | 50 |
| 5.1 Pengantar UI dan UX | 50 |
| 5.2 Landasan Perancangan UI/UX..... | 50 |
| 5.3 Tools Desain Figma | 51 |
| 5.4 Pemanfaatan plugin icon8 | 51 |

| | |
|--|----|
| 5.5 Tampilan halaman publik..... | 51 |
| 5.6 Tampilan Dashboard Pengguna Internal..... | 54 |
| 5.7 Prototyping dan Uji Alur UX..... | 56 |
| 5.8 Kesimpulan Perancangan UI/UX..... | 56 |
| 6. Implementasi Slicing HTML CSS | 58 |
| 6.1 Pendahuluan | 58 |
| 6.2 Teknologi yang Digunakan | 58 |
| 6.3 Penjelasan Bahasa Pemrograman HTML dan CSS | 58 |
| 6.4 Struktur Folder dan File Website | 59 |
| 6.5 Komponen Dasar HTML yang Digunakan | 60 |
| 6.6 Komponen dan Properti CSS yang Digunakan..... | 63 |
| 6.7 Proses Slicing dari Figma ke HTML dan CSS..... | 66 |
| 6.8 Menjalankan Website Menggunakan XAMPP | 67 |
| 6.9 Kesimpulan | 67 |
| Refrensi | 68 |
| Lampiran | 69 |

Daftar Tabel

| | |
|---|----|
| Tabel 1. Requirement | 4 |
| Tabel 2. Design | 4 |
| Tabel 3. Implementation | 5 |
| Tabel 4. Testing..... | 5 |
| Tabel 5. Maintenance | 6 |
| Tabel 6. Deskripsi fitur aktivitas..... | 9 |
| Tabel 7. Karakteristik Pengguna | 10 |
| Tabel 8. Use case untuk umum | 12 |
| Tabel 9. Use case untuk mahasiswa..... | 12 |
| Tabel 10. Use case untuk dosen | 13 |
| Tabel 11. Use case untuk admin | 13 |
| Tabel 12. Fungsi Silang | 14 |
| Tabel 13. Simbol Activity Diagram | 20 |
| Tabel 14. Tipe data dictionary | 22 |
| Tabel 15. Simbol ERD | 32 |
| Tabel 16. Hubungan relasi MYSQL | 42 |
| Tabel 17. Komponen teknologi yang digunakan | 58 |
| Tabel 18.Komponen HTML yang digunakan | 60 |
| Tabel 19. Komponen CSS yang digunakan | 64 |

Daftar Gambar

| | |
|---|----|
| Gambar 1. Sitemap..... | 18 |
| Gambar 2. Use Case | 19 |
| Gambar 3. Activity Diagram..... | 21 |
| Gambar 4. Data dictionary Email, Login, Profil Dosen..... | 25 |
| Gambar 5. Data dictionary profil mahasiswa, artikel, komentar | 26 |
| Gambar 6. Data dictionary update artikel, publish jurnal dan artikel, indekstasi jurnal | 27 |
| Gambar 7. Notasi DFD | 28 |
| Gambar 8. DFD Level 0..... | 29 |
| Gambar 9. DFD Level 1..... | 30 |
| Gambar 10. Simbol ERD | 32 |
| Gambar 11. Crow's Foot Notation..... | 33 |
| Gambar 12. Kardinalitas ERD | 34 |
| Gambar 13. ERD | 35 |
| Gambar 14. Instalasi XAMPP..... | 36 |
| Ganbar 15. Koneksi XAMPP | 37 |
| Gambar 16. Tampilan awal MySQL..... | 38 |
| Gambar 17. Pembuatan database | 39 |
| Gambar 18. Pembuatan tabel | 39 |
| Gambar 19. Pembuatan tabel data..... | 40 |
| Gambar 20. Hasil pembuatan database dan tabel data..... | 40 |
| Gambar 21. Memasukan <i>Value</i> data | 41 |
| Gambar 22. Hasil penambahan data..... | 41 |
| Gambar 23. Hasil Indeks..... | 42 |
| Gambar 24. Penambahan relasi..... | 44 |
| Gambar 25. Struktur desain MySQL | 45 |
| Gambar 26. Ekspor data MySQL..... | 45 |
| Gambar 27. Pembuatan database SQL code | 46 |
| Gambar 28. Pembuatan tabel data dengan sql code..... | 46 |
| Gambar 29. Pembuatan primary key dan foreign key dengan sql code..... | 47 |
| Gambar 30. Penambahan relasi dengan sql code | 47 |
| Gambar 31. Penambahan data ke tabel dengan sql code | 48 |
| Gambar 32. Menampilkan keseluruhan isi tabel data dengan sql code | 48 |

| | |
|---|----|
| Gambar 33. Menampilkan isi data tertentu dari tabel data dengan sql code..... | 49 |
| Gambar 34. Tampilan beranda..... | 52 |
| Gambar 35. Detail jurnal dan workingpaper..... | 53 |
| Gambar 36. Profil dosen | 54 |
| Gambar 37. Dashboard Mahasiswa..... | 55 |
| Gambar 38. Dashboard Dosen | 55 |
| Gambar 39. Dashboard Admin | 56 |
| Gambar 40. Contoh struktur dasar HTML | 59 |
| Gambar 41. Contoh penggunaan CSS..... | 59 |
| Gambar 42. Struktur folder dan file website..... | 60 |
| Gambar 43. Hasil menjalankan local host..... | 67 |

1. Deskripsi Keseluruhan

1.1 Deskripsi Produk

Dokumen ini merupakan Spesifikasi Kebutuhan Perangkat Lunak (SKPL) yang disusun untuk mendukung pengembangan Sistem Informasi Manajemen (SIM). Tujuan penyusunan dokumen adalah memberikan penjelasan menyeluruh mengenai kebutuhan perangkat lunak, baik berupa gambaran umum maupun detail proses yang terlibat.

Kebutuhan sistem dirumuskan berdasarkan identifikasi aktor dan aktivitas yang terekam dalam data analisis pada tabel analisis kebutuhan.

Beberapa aktivitas utama mencakup pembuatan dan pengelolaan dokumen, pengunggahan data pendukung, pengaturan akses pengguna, kolaborasi antar pihak terkait, pencatatan (version control), hingga proses persetujuan dan arsip data. Dengan demikian, dokumen SKPL ini tidak hanya mendeskripsikan fungsi teknis sistem, tetapi juga mengakomodasi alur kerja pengguna aktor seperti mahasiswa (author), dosen (reviewer/author), admin, maupun publik sesuai kebutuhan.

Dokumen ini ditujukan untuk pengembang (developer) sebagai acuan teknis dalam membangun sistem, serta pengguna (user) sebagai panduan interaksi dengan perangkat lunak. Dengan adanya SKPL, pengembangan SIM diharapkan lebih terarah, sistematis, sehingga mendukung pencapaian tujuan organisasi dalam pengelolaan logistik maupun manajemen dokumen akademik.

1.2 Fungsi Produk

Fungsi Produk untuk Program Studi Teknik Industri

Produk ini pada dasarnya merupakan sebuah website yang dirancang untuk mendukung manajemen penulisan dan pengelolaan jurnal mahasiswa. Fungsi utama bagi mahasiswa antara lain sebagai berikut:

1. Sebagai wadah penyimpanan seluruh dokumen penulisan

Aplikasi ini dapat digunakan untuk membuat draft artikel, mengunggah data penelitian, gambar, maupun file lain yang mendukung

penulisan jurnal. Dengan demikian, seluruh dokumen dapat tersimpan secara terorganisasi dalam satu sistem, tidak lagi tersebar secara acak di perangkat pribadi mahasiswa.

2. Mempermudah kolaborasi dengan dosen pembimbing

Penulisan jurnal pada umumnya memerlukan bimbingan dosen. Aplikasi ini memberikan kemudahan bagi mahasiswa untuk memberikan akses kepada dosen atau rekan, baik hanya untuk melihat, memberikan komentar, maupun menyunting dokumen. Dengan adanya fitur ini, proses komunikasi menjadi lebih efektif tanpa perlu bergantung pada pertukaran file melalui surat elektronik.

3. Mendukung kolaborasi penulisan

Aplikasi ini dilengkapi dengan fitur kolaborasi yang memungkinkan mahasiswa dan dosen melakukan penulisan secara bersama dalam satu dokumen secara langsung. Dengan demikian, revisi maupun masukan dapat segera direspon tanpa harus menunggu lama.

4. Kontrol dan pelacakan revisi dokumen

Aplikasi menyediakan fitur version control sehingga setiap perubahan dokumen tercatat dengan baik. Jika diperlukan, pengguna dapat kembali ke versi sebelumnya. Fitur ini sangat penting untuk memantau perkembangan penulisan dari tahap awal hingga selesai.

5. Dashboard sebagai pusat informasi

Halaman utama (dashboard) memberikan ringkasan status seluruh artikel yang dimiliki mahasiswa, seperti artikel yang masih dalam bentuk draft, sedang dalam proses peninjauan, maupun yang telah selesai. Dengan fitur ini, mahasiswa dapat lebih mudah mengontrol perkembangan penulisan tanpa harus membuka dokumen satu per satu.

6. Meningkatkan efisiensi dan profesionalitas

Aplikasi ini membantu mahasiswa dalam menyusun dokumen secara lebih terarah, terstruktur, dan profesional. Dengan demikian,

proses penulisan tidak hanya sekadar menghasilkan karya tulis, tetapi juga mengajarkan manajemen dokumen secara sistematis.

Fungsi Produk untuk Publik

Selain memberikan manfaat bagi mahasiswa dan dosen, aplikasi ini juga memiliki fungsi yang penting bagi publik, khususnya bagi masyarakat akademik dan dunia industri. Beberapa fungsi tersebut antara lain:

1. Sebagai repositori jurnal mahasiswa

Publik dapat mengakses kumpulan jurnal atau working paper yang telah selesai dan dipublikasikan. Dengan demikian, karya mahasiswa tidak hanya menjadi arsip internal kampus, tetapi juga dapat dibaca dan dimanfaatkan oleh masyarakat luas.

2. Sebagai media berbagi hasil penelitian

Melalui aplikasi ini, mahasiswa dapat membagikan hasil penelitian kepada publik sehingga penelitian yang telah dilakukan dapat memberikan dampak nyata. Publik yang membutuhkan referensi, inspirasi, maupun data penelitian dapat langsung mencari melalui sistem ini.

3. Membangun citra dan reputasi kampus maupun mahasiswa

Dengan keterbukaan akses jurnal mahasiswa, reputasi kampus dan mahasiswa dapat meningkat. Publik dapat menilai kualitas penelitian yang dihasilkan, yang pada gilirannya dapat meningkatkan citra positif kampus sekaligus membuka peluang kolaborasi dengan pihak eksternal, baik akademisi maupun pelaku industri.

1.3 Pengembangan produk

1. Definitif

Proses repository jurnal memiliki batasan yang jelas, dimulai dari penulis mengunggah artikel ilmiah, melalui tahap review dan revisi, hingga artikel diterima lalu disimpan dalam sistem repository agar dapat diakses publik.

2. Urutan

Proses terdiri dari tahapan yang runtut: penyerahan artikel, pemeriksaan kelengkapan, review, revisi, publikasi akhir, artikel dapat diakses dan diindeks.

3. Pengguna

Pengguna utama repository adalah pembaca (mahasiswa, dosen, peneliti) yang mengakses artikel sebagai referensi ilmiah. Penulis mendapatkan manfaat berupa publikasi karya ilmiah, sementara pengelola bertanggung jawab menjaga kualitas repository.

4. Value

Repository jurnal online memberikan nilai berupa kemudahan akses pengetahuan. Penulis dapat menyebarkan hasil riset secara luas, sementara pembaca memperoleh sumber ilmiah tanpa harus datang langsung ke perpustakaan fisik. Institusi juga memperoleh peningkatan visibilitas akademik.

5. Ketertarikan

Repository jurnal terhubung dengan pihak lain seperti reviewer, editor, institusi pendidikan, serta layanan indeksasi (misalnya DOI atau Google Scholar). Sistem juga terintegrasi dengan infrastruktur teknologi untuk memastikan artikel terdokumentasi dengan baik.

6. Requirement

Tabel 1. Requirement

| | | |
|-----------|---|--|
| Tujuan | : | <ul style="list-style-type: none">Mengidentifikasi kebutuhan pengguna (mahasiswa, dosen, admin, publik).Mendefinisikan fungsi utama sistem (upload artikel, review, publikasi, indeksasi, akses jurnal).Menetapkan batasan sistem agar sesuai dengan tujuan prodi. |
| Aktivitas | : | <ul style="list-style-type: none">Analisis alur kerja dari activity diagram.Penyusunan dokumen kebutuhan fungsional (login, upload, review, publikasi).Penyusunan kebutuhan non-fungsional (keamanan, performa, aksesibilitas). |

7. Design

Tabel 2. Design

| | | |
|--------|---|--|
| Tujuan | : | <ul style="list-style-type: none">Merancang arsitektur sistem yang sesuai dengan kebutuhan.Membuat desain antarmuka pengguna dan struktur basis data. |
|--------|---|--|

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • Menyusun alur navigasi sistem sesuai sitemap. |
| Aktivitas : | <ul style="list-style-type: none"> • Membuat diagram UML (use case, class, sequence). • Mendesain struktur basis data untuk artikel, jurnal, reviewer, indeksasi. • Membuat prototipe antarmuka (UI/UX) untuk halaman login, upload artikel, publikasi. • Menentukan teknologi yang digunakan (framework, database, server). |

8. Implementation

Tabel 3. Implementation

| | |
|-------------|---|
| Tujuan : | <ul style="list-style-type: none"> • Menerjemahkan desain ke dalam kode program. • Membangun modul fungsional sesuai dengan kebutuhan. • Mengintegrasikan komponen sistem menjadi satu kesatuan. |
| Aktivitas : | <ul style="list-style-type: none"> • Coding modul autentikasi, upload artikel, manajemen reviewer, publikasi. • Integrasi dengan sistem basis data. • Pengembangan API untuk akses jurnal online. • Melakukan code review untuk memastikan kualitas implementasi. |

9. Testing

Tabel 4. Testing

| | |
|----------|---|
| Tujuan : | <ul style="list-style-type: none"> • Memastikan sistem bekerja sesuai kebutuhan. • Menemukan dan memperbaiki bug sebelum sistem dirilis. • Menguji keamanan dan performa sistem. |
|----------|---|

| | | |
|-----------|---|--|
| Aktivitas | : | <ul style="list-style-type: none"> • Unit testing untuk setiap modul (login, upload, indeksasi). • Integration testing untuk memastikan modul dapat saling berkomunikasi. • System testing untuk keseluruhan sistem. • User acceptance testing (UAT) dengan mahasiswa, dosen, admin. • Uji performa (jumlah artikel besar, akses serentak). |
|-----------|---|--|

10. Maintenance

Tabel 5. Maintenance

| | | |
|-----------|---|---|
| Tujuan | : | <ul style="list-style-type: none"> • Menjamin sistem tetap berjalan sesuai kebutuhan setelah implementasi. • Melakukan perbaikan jika ditemukan bug baru. • Mengembangkan sistem agar adaptif terhadap kebutuhan baru (misalnya integrasi ke database indeksasi nasional). |
| Aktivitas | : | <ul style="list-style-type: none"> • Monitoring performa sistem secara berkala. • Penerapan patch keamanan. • Update fitur berdasarkan feedback pengguna. • Backup data rutin dan pemulihan bencana (disaster recovery). • Dokumentasi versi sistem dan perubahan fitur. |

1.4 Lingkungan Operasi

1. Platform

Platform ini dikembangkan dalam bentuk website berbasis web application sehingga dapat diakses langsung melalui browser tanpa memerlukan instalasi tambahan. Website ini juga responsif, sehingga bisa

dijalankan dengan baik baik melalui perangkat desktop maupun perangkat mobile.

2. Sistem Operasi dan Versi

Website dapat berjalan di hampir semua OS pada sisi pengguna, mulai dari Windows, macOS, Linux, hingga Android dan iOS, karena hanya memerlukan browser modern seperti Google Chrome, Mozilla Firefox, Safari, atau Microsoft Edge. Sedangkan untuk server, sistem operasi berbasis Linux seperti Ubuntu atau Debian sangat direkomendasikan karena stabilitasnya, walaupun Windows Server juga tetap dapat digunakan.

3. Aplikasi yang Berdampingan

Dalam penggunaannya, perangkat lunak ini bisa beroperasi berdampingan dengan aplikasi lain, seperti Microsoft Office Online atau Google Docs untuk proses pengeditan dokumen, Google Scholar, Mendeley, atau Zotero untuk manajemen referensi, serta platform komunikasi seperti WhatsApp Web, Telegram, dan email untuk mendukung koordinasi di luar sistem. Dengan integrasi tersebut, website tidak hanya berdiri sendiri, tetapi juga menjadi bagian dari ekosistem digital yang lebih luas dalam mendukung kegiatan akademik.

1.5 Batasan dan Implementasi

1. Kebijakan Kampus dan Regulasi Akademik

Karena aplikasi ini ditujukan untuk mahasiswa dalam penulisan jurnal atau working paper, maka sistem harus mengikuti kebijakan akademik yang berlaku di kampus. Seperti format penulisan jurnal yang sudah ditentukan, aturan kolaborasi mahasiswa-dosen, serta aturan publikasi. Hal ini membatasi fleksibilitas pengembang, karena sistem harus menyesuaikan dengan standar kampus, bukan sepenuhnya bebas.

2. Keterbatasan Perangkat Keras (Hardware Requirements)

Target utama pengguna adalah mahasiswa yang biasanya menggunakan laptop standar dengan RAM terbatas (4–8 GB). Oleh karena itu, pengembang harus membuat aplikasi yang ringan, tidak terlalu banyak

memakan memori, dan tetap responsif meskipun dijalankan di laptop dengan spesifikasi menengah.

3. Teknologi dan Database yang Digunakan

Pengembangan sistem dilakukan dengan memanfaatkan database MySQL sebagai basis penyimpanan data utama. Pemilihan MySQL didasarkan pada kestabilan, skalabilitas, serta dukungan luas dari komunitas pengembang. Struktur database dirancang menggunakan model relasional dengan normalisasi agar data lebih terorganisir, konsisten, dan mudah diakses oleh aplikasi.

Untuk mendukung akses aplikasi, digunakan web server standar seperti Apache. Apache dipilih karena sifatnya yang open-source, fleksibel, serta memiliki kompatibilitas tinggi dengan berbagai bahasa pemrograman, khususnya PHP yang umum digunakan bersama MySQL. Server ini akan bertugas melayani permintaan dari client (browser), mengolah request, kemudian menghubungkannya dengan database agar data dapat ditampilkan sesuai kebutuhan pengguna.

Dengan kombinasi MySQL sebagai database dan Apache sebagai web server, sistem dapat dikembangkan secara modular, aman, serta mudah diintegrasikan dengan framework berbasis web. Hal ini juga memungkinkan sistem untuk dikelola baik secara lokal maupun pada server hosting berbasis Linux/Windows.

4. Keamanan

Karena website ini menyimpan dokumen akademik penting (draft jurnal, data penelitian, dsb.), keamanan menjadi batasan utama:

- Sistem harus punya autentikasi (login dengan akun mahasiswa/dosen).
- Data harus disimpan secara terenkripsi untuk mencegah kebocoran.

Hal ini membatasi pengembang untuk tidak sembarangan dalam menyimpan file, dan harus menambahkan modul keamanan tambahan.

5. Standar Pemrograman dan Kualitas Kode

Untuk menjamin keberlanjutan, pengembang harus mengikuti standar pemrograman yang konsisten (misalnya clean code dan coding convention tertentu). Ini menjadi batasan karena meskipun ada cara cepat untuk coding, pengembang harus tetap mengikuti aturan standar supaya mudah dipelihara.

2. Deskripsi Kebutuhan Fungsional

2.1 Proses Kegiatan

Tabel 6. Deskripsi fitur aktivitas

| Proses/Kegiatan | Deskripsi |
|---------------------------------|---|
| Profil | User dapat mengelola profil |
| Upload dokumen | Upload dokumen link sharepoint dan pdf |
| Registrasi akun | Pengguna baru (mahasiswa/dosen) membuat akun untuk mengakses sistem jurnal. |
| Login sistem | Pengguna masuk menggunakan akun terdaftar. |
| Kolaborasi komentar | Membalas komentar reviewer/editor langsung di dokumen. |
| Tracking progress | Memantau status (submitted, review, revisi, accepted, published). |
| Melihat arsip revisi | Melihat arsip revisi |
| Update revisi | Mengupdate revisi |
| Menambah dan mengelola refensi | Menambah dan mengelola refensi |
| Setting affiliasi/akses dokumen | Menyimpan artikel yang sudah publish untuk portofolio pribadi. |
| Approve/Reject via dashboard | Memberi keputusan akhir (accept, revisi, reject). |
| Dashboard Admin | Mengelola seluruh artikel (draft, review, publish) melalui dashboard pusat. |
| Validasi akun | Memverifikasi akun mahasiswa & dosen. |
| Assign reviewer | Menetapkan reviewer untuk artikel. |

| | |
|--|--|
| Manage folder jurnal | Mengatur struktur folder per edisi/volume jurnal. |
| Publikasi artikel | Artikel yang lolos review dipindahkan ke folder publik & tampil di website. |
| Statistik dashboard | Memantau submission, revisi, approval, publish, dan trafik akses. |
| Indeksasi jurnal (SINTA, DOAJ, GS, Scopus, dll.) | Mengajukan jurnal ke lembaga pengindeks agar publikasi lebih kredibel & terakui. |
| Monitoring indeksasi | Memantau perkembangan akreditasi dan peringkat jurnal di sistem indeksasi. |
| Dashboard publik | Publik mengakses daftar artikel terbaru, populer, atau berdasarkan bidang. |
| Pencarian artikel | Mencari artikel berdasarkan judul, penulis, kata kunci. |
| View online | Membaca artikel langsung di browser (viewer mirip Google Docs). |
| Download PDF | Mengunduh artikel dalam format PDF. |
| Share DOI/link | Membagikan artikel dengan permalink atau DOI. |
| Melihat statistik artikel | Melihat jumlah view, download, atau sitasi. |

2.2 Karakteristik Pengguna

Tabel 7. Karakteristik Pengguna

| Aktor | Deskripsi |
|-----------|---|
| Umum | Publik dapat mencari artikel berdasarkan berbagai kriteria, membaca atau mengunduh sesuai hak akses, memfilter artikel, melihat statistik, serta membagikan artikel yang sudah diterbitkan. |
| Mahasiswa | Mahasiswa bertugas membuat akun dan login, mengunggah dokumen, memantau |

| | |
|-------|--|
| | perkembangan artikel, melakukan revisi, mengelola referensi, serta menyimpan artikel yang telah diterbitkan untuk portofolio. |
| Dosen | Dosen bertugas membuat akun dan login, mengunggah dokumen, memantau perkembangan artikel, melakukan revisi, mengelola referensi, serta menyimpan artikel yang telah diterbitkan untuk portofolio. serta memiliki tugas tambahan sebagai reviewer memantau artikel mahasiswa, memberikan bimbingan dan komentar, serta menyetujui revisi yang dilakukan mahasiswa. |
| Admin | Admin memiliki tugas mengelola seluruh aktivitas sistem melalui dashboard admin, termasuk melakukan manajemen pengguna untuk menambah, mengedit, atau menghapus akun, serta manajemen artikel untuk mengatur, menghapus, atau memulihkan artikel. Selain itu, admin juga bertanggung jawab melakukan backup dan restore data sebagai bentuk pengamanan, serta menjalankan monitoring sistem guna memantau aktivitas pengguna dan memastikan kelancaran operasional platform. |

2.3 Use Case setiap Aktor

Use case digunakan untuk menjelaskan interaksi antara aktor dengan sistem berdasarkan fungsi yang tersedia. Pada sistem repository jurnal, aktor yang terlibat meliputi mahasiswa, dosen, admin, serta publik, masing-masing dengan peran dan hak akses yang berbeda. Melalui use case, kebutuhan fungsional sistem seperti registrasi akun, unggah artikel, proses review,

publikasi, hingga akses jurnal dapat digambarkan secara terstruktur sesuai alur penggunaannya.

Use case untuk umum

Tabel 8. Use case untuk umum

| | |
|---------------------------|--|
| Dashboard publik | Publik mengakses daftar artikel terbaru, populer, atau berdasarkan bidang. |
| Pencarian artikel | Mencari artikel berdasarkan judul, penulis, kata kunci. |
| View online | Membaca artikel langsung di browser (viewer mirip Google Docs). |
| Download PDF | Mengunduh artikel dalam format PDF. |
| Share DOI/link | Membagikan artikel dengan permalink atau DOI. |
| Melihat statistik artikel | Melihat jumlah view, download, atau sitasi. |

Use case untuk Mahasiswa

Tabel 9. Use case untuk mahasiswa

| | |
|----------------------|---|
| Profil | Mahasiswa dapat mengelola profil |
| Upload dokumen | Upload dokumen link sharepoint dan pdf |
| Registrasi akun | Pengguna baru (mahasiswa/dosen) membuat akun untuk mengakses sistem jurnal. |
| Login sistem | Pengguna masuk menggunakan akun terdaftar. |
| Kolaborasi komentar | Membalas komentar reviewer/editor langsung di dokumen. |
| Tracking progress | Memantau status (submitted, review, revisi, accepted, published). |
| Melihat arsip revisi | Melihat arsip revisi |

| | |
|---------------------------------|--|
| Update revisi | Mengupdate revisi |
| Menambah dan mengelola refrensi | Menambah dan mengelola refrensi |
| Setting affiliasi/akses dokumen | Menyimpan artikel yang sudah publish untuk portofolio pribadi. |

Use case untuk dosen

Tabel 10. Use case untuk dosen

| | |
|---------------------------------|---|
| Profil | Dosen dapat mengelola profil |
| Upload dokumen | Upload dokumen link sharepoint dan pdf |
| Registrasi akun | Pengguna baru (mahasiswa/dosen) membuat akun untuk mengakses sistem jurnal. |
| Login sistem | Pengguna masuk menggunakan akun terdaftar. |
| Kolaborasi komentar | Membalas komentar reviewer/editor langsung di dokumen. |
| Tracking progress | Memantau status (submitted, review, revisi, accepted, published). |
| Melihat arsip revisi | Melihat arsip revisi |
| Update revisi | Mengupdate revisi |
| Menambah dan mengelola refrensi | Menambah dan mengelola refrensi |
| Setting affiliasi/akses dokumen | Menyimpan artikel yang sudah publish untuk portofolio pribadi. |
| Approve/Reject via dashboard | Memberi keputusan akhir (accept, revisi, reject). |

Use case untuk admin

Tabel 11. Use case untuk admin

| | |
|-----------------|---|
| Dashboard Admin | Mengelola seluruh artikel (draft, review, publish) melalui dashboard pusat. |
|-----------------|---|

| | |
|--|--|
| Validasi akun | Memverifikasi akun mahasiswa & dosen. |
| Assign reviewer | Menetapkan reviewer untuk artikel. |
| Manage folder jurnal | Mengatur struktur folder per edisi/volume jurnal. |
| Publikasi artikel | Artikel yang lolos review dipindahkan ke folder publik & tampil di website. |
| Statistik dashboard | Memantau submission, revisi, approval, publish, dan trafik akses. |
| Indeksasi jurnal (SINTA, DOAJ, GS, Scopus, dll.) | Mengajukan jurnal ke lembaga pengindeks agar publikasi lebih kredibel & terakui. |
| Monitoring indeksasi | Memantau perkembangan akreditasi dan peringkat jurnal di sistem indeksasi. |

2.4 Fungsi Silang

Fungsi silang atau *cross-functional requirements* mendeskripsikan hubungan antar fitur yang saling terkait dan melintasi berbagai proses dalam sistem. Fungsi silang ini memastikan setiap aktivitas tidak berjalan secara terpisah, melainkan saling mendukung mulai dari tahap penyusunan draft artikel, penugasan reviewer, kolaborasi komentar, publikasi, indeksasi, hingga akses jurnal oleh publik. Dengan demikian, fungsi silang melengkapi use case dalam menjamin keterpaduan proses bisnis sistem.

Tabel 12. Fungsi Silang

| | | |
|--------------------------------|---|--|
| Fungsi Draft Artikel | : | Assign Reviewer, Kolaborasi Komentar, Update Artikel |
| Fungsi Assign Reviewer (Admin) | : | Draft Artikel, Kolaborasi Komentar, Update Artikel |

| | | |
|--------------------------------------|---|---|
| Fungsi Kolaborasi Komentar | : | Draft Artikel, Assign Reviewer, Update Artikel |
| Fungsi Update Artikel | : | Draft Artikel, Kolaborasi Komentar, Assign Reviewer, Status Artikel |
| Fungsi Status Artikel | : | Update Artikel |
| Fungsi Publikasi Artikel | : | Update Artikel, Manage Dokumen Jurnal Artikel, Indeksasi Jurnal |
| Fungsi Manage Dokumen Jurnal Artikel | : | Publikasi Artikel, Indeksasi Jurnal |
| Fungsi Indeksasi Jurnal | : | Publikasi Artikel, Manage Dokumen Jurnal Artikel, Update Status Indeksasi, Lihat Status Indeksasi |
| Fungsi Update Status Indeksasi | : | Indeksasi Jurnal, Lihat Status Indeksasi |
| Fungsi Lihat Status Indeksasi | : | Indeksasi Jurnal, Update Status Indeksasi |
| Fungsi Akses Jurnal | : | Publikasi Artikel, Indeksasi Jurnal, Statistik Jurnal |
| Fungsi Statistik Jurnal | : | Akses Jurnal |

2.5 Kebutuhan Perangkat Lunak dan Perangkat Keras

1. Perangkat Keras (Hardware)

Karena target utamanya mahasiswa dan dosen, spesifikasi perangkat keras yang dibutuhkan tidak terlalu tinggi, hanya standar laptop atau PC yang biasa dipakai untuk keperluan akademik. Perkiraannya:

Kebutuhan pengguna tidak terlalu tinggi karena sebagian besar proses dijalankan di server. Laptop atau komputer standar dengan

prosesor minimal setara Intel i3, RAM 4 GB, dan resolusi layar 1366x768 sudah cukup untuk mengakses sistem. Sementara di sisi server, dibutuhkan spesifikasi yang lebih besar seperti prosesor multi-core, RAM minimal 8 GB, kapasitas penyimpanan ratusan gigabyte sesuai jumlah dokumen, serta jaringan internet yang stabil untuk melayani banyak pengguna.

2. Komponen Perangkat Lunak Lain

Database server seperti MySQL atau PostgreSQL digunakan untuk menyimpan akun, dokumen, dan riwayat aktivitas. Website di-host melalui web server seperti Apache atau Nginx yang dipadukan dengan framework web sesuai bahasa pemrograman yang dipilih, serta penyimpanan tambahan dapat menggunakan cloud storage atau server lokal untuk menampung file seperti gambar, dataset, maupun lampiran jurnal.

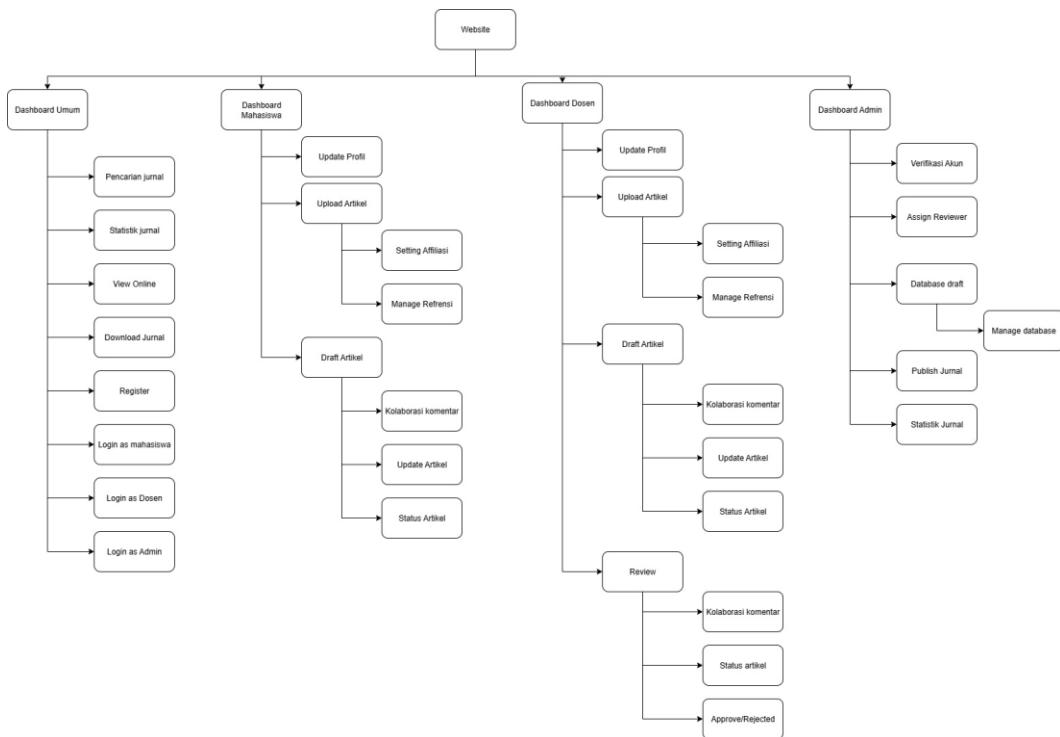
3. Desain dan Perancangan Sistem

3.1 Site Map

Sitemap sistem repository jurnal ini menggambarkan struktur navigasi utama yang terbagi ke dalam empat dashboard sesuai dengan jenis pengguna, yaitu Dashboard Umum, Dashboard Mahasiswa, Dashboard Dosen, dan Dashboard Admin. Pada Dashboard Umum, pengguna publik dapat melakukan pencarian jurnal, melihat statistik, membaca artikel secara daring, mengunduh jurnal dalam format PDF, melakukan registrasi akun, serta login sesuai perannya, baik sebagai mahasiswa, dosen, maupun admin. Selanjutnya, Dashboard Mahasiswa menyediakan fitur pengelolaan profil, unggah artikel, pengaturan afiliasi, manajemen referensi, serta pengelolaan draft artikel yang mencakup kolaborasi komentar, pembaruan artikel, dan pemantauan status artikel.

Pada Dashboard Dosen, fitur yang tersedia hampir serupa dengan mahasiswa, meliputi pengelolaan profil, unggah artikel, pengaturan afiliasi, manajemen referensi, serta pengelolaan draft artikel. Namun, dosen memiliki tambahan peran dalam proses review artikel, termasuk memberikan komentar, memantau status artikel, hingga melakukan persetujuan atau penolakan. Sementara itu, Dashboard Admin memiliki fungsi yang lebih luas, mencakup verifikasi akun, penugasan reviewer, pengelolaan database draft, publikasi jurnal, serta pemantauan statistik. Admin juga bertanggung jawab terhadap manajemen database agar sistem tetap terstruktur dan berfungsi dengan baik.

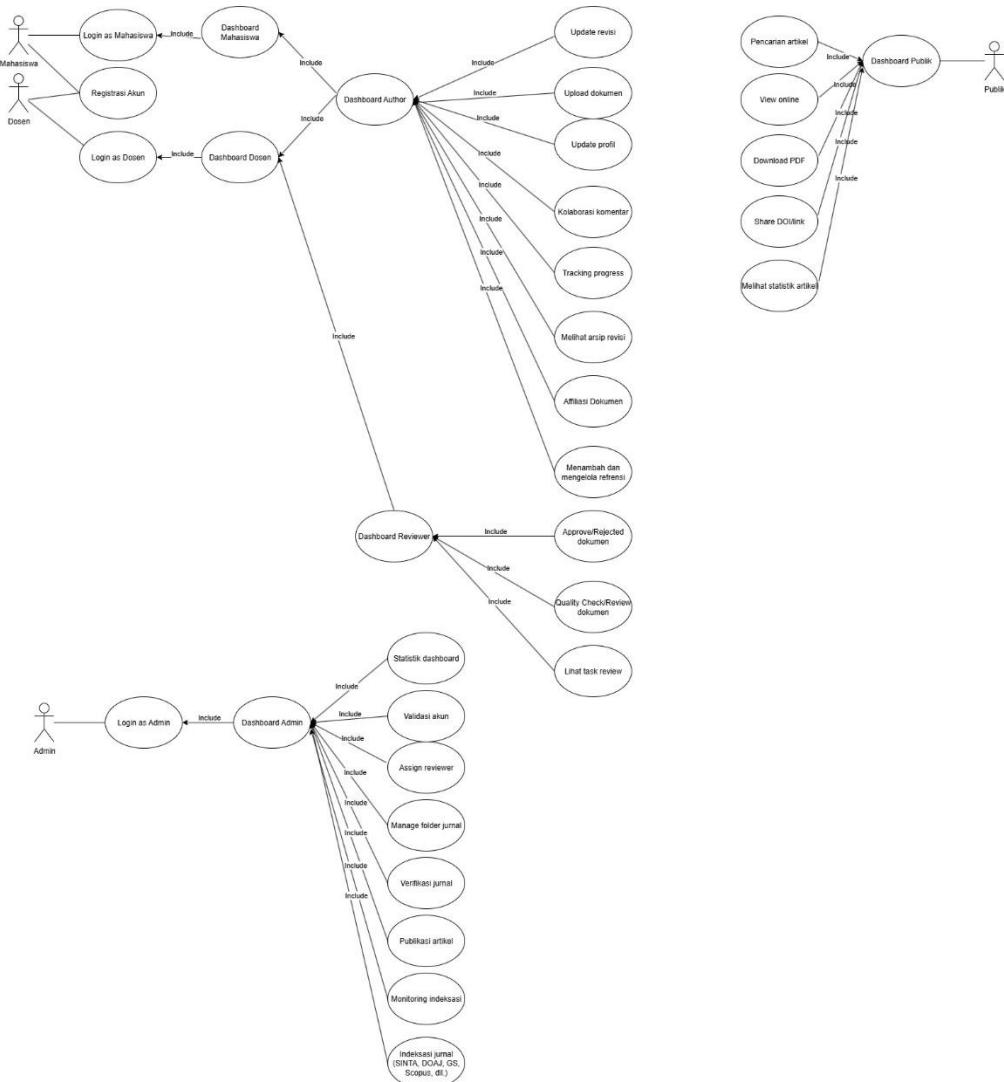
Secara keseluruhan, sitemap ini menunjukkan keterhubungan antara berbagai modul sistem sesuai dengan kebutuhan tiap jenis pengguna. Struktur hierarki yang jelas memungkinkan navigasi lebih mudah sekaligus mendukung alur proses bisnis mulai dari registrasi dan unggah artikel, proses review dan revisi, hingga publikasi dan akses jurnal oleh publik.



Gambar 1. Sitemap

3.2 Use Case

Diagram use case ini menggambarkan hubungan antara aktor dengan sistem repository jurnal berdasarkan peran dan hak akses yang dimiliki. Sebagaimana telah dijelaskan pada bab sebelumnya, use case berfungsi untuk mendeskripsikan kebutuhan fungsional sistem melalui interaksi antara pengguna dan fitur yang tersedia. Pada sistem ini, mahasiswa dan dosen berinteraksi melalui dashboard author untuk mengelola artikel, sementara dosen juga memiliki peran tambahan sebagai reviewer. Publik mengakses sistem melalui dashboard publik dengan fungsi pencarian dan pembacaan artikel, sedangkan admin menggunakan dashboard admin untuk mengelola akun, artikel, dan proses publikasi hingga indeksasi. Dengan demikian, diagram ini memberikan gambaran umum mengenai fungsi utama yang tersedia dalam sistem serta keterhubungan antar aktivitas yang mendukung proses bisnis secara menyeluruh.



Gambar 2. Use Case

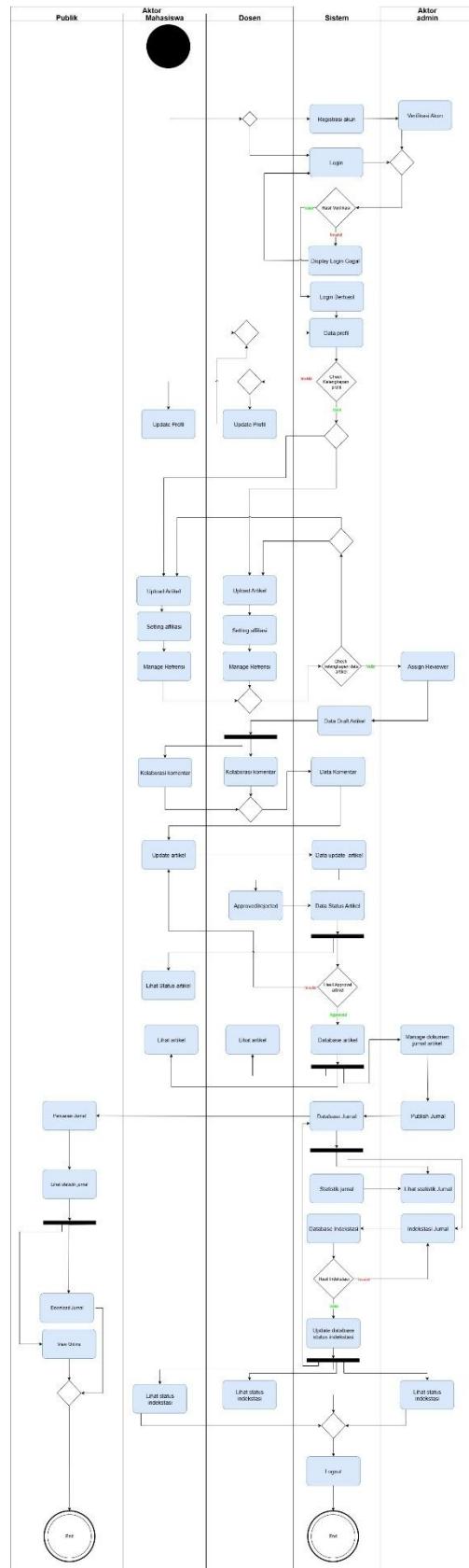
3.3 Activity Diagram

Activity diagram pada sistem repository jurnal ini digunakan untuk memodelkan alur aktivitas yang melibatkan empat aktor utama, yaitu publik, mahasiswa, dosen, dan admin. Sebagai contoh, proses diawali dengan registrasi akun dan login, kemudian sistem melakukan verifikasi. Jika validasi berhasil, pengguna dapat melanjutkan aktivitas sesuai perannya, misalnya mahasiswa dan dosen mengunggah artikel atau mengelola referensi, sementara admin memverifikasi akun dan menetapkan reviewer. Proses selanjutnya berjalan hingga artikel dipublikasikan dan dapat diakses publik, lalu diindeks pada lembaga pengindeks.

Dalam penyusunannya, activity diagram menggunakan simbol-simbol standar UML untuk memperjelas alur proses. Setiap simbol memiliki arti tersendiri sebagaimana ditunjukkan pada tabel berikut

Tabel 13. Simbol Activity Diagram

| Simbol | Nama Simbol | Fungsi/Arti |
|--|-----------------------|--|
| ● (lingkaran hitam penuh) | Initial Node | Menandakan titik awal dari sebuah aktivitas/proses. |
| ◎ (lingkaran dengan garis tebal di luar) | Final Node | Menandakan titik akhir dari aktivitas/proses. |
| □ (persegi panjang dengan sudut tumpul) | Activity/Action | Menggambarkan kegiatan atau proses yang dilakukan sistem atau aktor. |
| ◇ (belah ketupat) | Decision / Merge Node | Decision: memisahkan alur berdasarkan kondisi (Ya/Tidak, Valid/Invalid). Merge: menggabungkan kembali alur alternatif menjadi satu jalur. |
| → (garis panah) | Control Flow | Menunjukkan urutan eksekusi dari satu aktivitas ke aktivitas lainnya. |
| —— (garis tebal horizontal) | Fork/Join Node | Fork: membagi alur menjadi paralel. Join: menggabungkan kembali alur menjadi satu. |



Gambar 3. Activity Diagram

3.4 Data Dictionary

Data dictionary merupakan kumpulan informasi terstruktur yang berfungsi untuk mendefinisikan, mendokumentasikan, dan menjelaskan elemen data yang digunakan dalam suatu sistem. Keberadaan data dictionary membantu pengembang, analis, maupun pengguna sistem dalam memahami arti, format, sumber, dan hubungan antar data. Dengan adanya kamus data, proses komunikasi antar tim menjadi lebih konsisten karena semua pihak merujuk pada definisi data yang sama.

Dalam konteks perancangan sistem, data dictionary memuat rincian setiap entitas, atribut, serta relasi yang telah digambarkan pada model konseptual seperti ERD (Entity Relationship Diagram). Setiap entitas dijabarkan dalam bentuk tabel yang mencantumkan nama atribut, tipe data, panjang data, domain nilai, hingga deskripsi penggunaannya. Dengan struktur ini, data dictionary berperan sebagai jembatan antara rancangan logis dan implementasi fisik di basis data.

Selain sebagai dokumentasi teknis, data dictionary juga mendukung proses pengendalian kualitas data. Adanya standar definisi atribut dapat mengurangi terjadinya redundansi, inkonsistensi, maupun ambiguitas pada sistem. Oleh karena itu, data dictionary tidak hanya menjadi referensi awal pada tahap pengembangan, tetapi juga dipakai sebagai acuan pemeliharaan sistem di masa mendatang agar integritas data tetap terjaga.

Tabel 14. Tipe data dictionary

| Kategori | Tipe Data | Keterangan |
|-------------------|------------|--|
| String (Karakter) | CHAR(n) | Menyimpan string dengan panjang tetap (1–255 karakter). |
| | VARCHAR(n) | Menyimpan string dengan panjang variabel (1–65535 karakter). |
| | TEXT | Menyimpan teks panjang hingga 65.535 karakter. |

| | | |
|-----------------------------|-------------------|--|
| | TINYTEXT | Menyimpan teks pendek hingga 255 karakter. |
| | MEDIUMTEXT | Menyimpan teks hingga 16.777.215 karakter. |
| | LONGTEXT | Menyimpan teks sangat panjang hingga 4.294.967.295 karakter. |
| | ENUM('a','b',...) | Menyimpan satu nilai dari daftar nilai yang telah ditentukan. |
| | SET('a','b',...) | Menyimpan satu atau lebih nilai dari daftar nilai yang ditentukan. |
| BLOB (Binary Large Object) | TINYBLOB | Menyimpan data biner hingga 255 byte. |
| | BLOB | Menyimpan data biner hingga 65.535 byte. |
| | MEDIUMBLOB | Menyimpan data biner hingga 16.777.215 byte. |
| | LONGBLOB | Menyimpan data biner hingga 4.294.967.295 byte. |
| Numeric / Integer | TINYINT | Bilangan bulat sangat kecil (-128 s/d 127). |
| | SMALLINT | Bilangan bulat kecil (-32.768 s/d 32.767). |
| | MEDIUMINT | Bilangan bulat menengah (-8.388.608 s/d 8.388.607). |
| | INT / INTEGER | Bilangan bulat standar (-2.147.483.648 s/d 2.147.483.647). |
| | BIGINT | Bilangan bulat sangat besar (-9.22e18 s/d 9.22e18). |
| Numeric / Desimal & Pecahan | DECIMAL(p,s) | Bilangan desimal presisi tinggi, cocok untuk data keuangan. |

| | | |
|-----------------|---------------|--|
| | NUMERIC(p,s) | Sama dengan DECIMAL, digunakan untuk angka presisi tetap. |
| | FLOAT | Bilangan pecahan presisi tunggal (~7 digit). |
| | DOUBLE / REAL | Bilangan pecahan presisi ganda (~16 digit). |
| Tanggal & Waktu | DATE | Format 'YYYY-MM-DD'. |
| | DATETIME | Format 'YYYY-MM-DD HH:MM:SS'. |
| | TIMESTAMP | Format 'YYYY-MM-DD HH:MM:SS', mengikuti zona waktu server. |
| | TIME | Format 'HH:MM:SS'. |
| | YEAR | Format 'YYYY'. |

REGISTRASI

Premary Key : Email

Foregin Key : -

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|-----------|--------------|-------------------|---|
| 1 | Email | Varchar (255) | Berisi email yang diinput oleh aktor |
| 2 | Password | Varchar (255) | Berisi password yang diinput oleh aktor |
| 3 | Jabatan | Enum | Berisi jabatan dari aktor |

LOGIN

Premary Key : Email

Foregin Key : -

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|-----------|--------------|-------------------|---|
| 1 | Email | Varchar (255) | Berisi email yang diinput oleh aktor |
| 2 | Password | Varchar (255) | Berisi password yang diinput oleh aktor |
| 3 | Jabatan | Enum | Berisi jabatan dari aktor |

PROFIL DOSEN

Premary Key : NIP

Foregin Key : -

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|-----------|--------------------|-------------------|--------------------------------------|
| 1 | NIP | Integer (30) | Berisi NIP dari dosen |
| 2 | Nama Dosen | String (255) | Berisi nama dari dosen |
| 3 | Riwayat Pendidikan | String (255) | Berisi riwayat pendidikan dari dosen |

Gambar 4. Data dictionary Email, Login, Profil Dosen

PROFIL MAHASISWA

Premary Key : NIM

Foregin Key : -

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|----|----------------|--------------|--------------------------------|
| 1 | NIM | Integer (30) | Berisi NIM dari mahasiswa |
| 2 | Nama Mahasiswa | String (255) | Berisi nama dari mahasiswa |
| 3 | Angkatan | Integer (30) | Berisi angkatan dari mahasiswa |

ARTIKEL

Premary Key : Draft Artikel

Foregin Key : -

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|----|---------------|---------------|-------------------------------|
| 1 | Draft Artikel | LONGBLOB (4) | Berisi judul dan file artikel |
| 2 | Afiliasi | String (255) | Berisi data afiliasi artikel |
| 3 | Referensi | Varchar (255) | Berisi data referensi artikel |

KOMENTAR

Premary Key : Waktu komentar

Foregin Key : -

Jumlah Filed : 2

| NO | Filed | Type/Lebar | Keterangan |
|----|----------------|----------------------------------|------------------------------------|
| 1 | Waktu Komentar | Timestamp (hh:mm:ss/yy:mm:dd) | Berisi waktu diberikannya komentar |
| 2 | Isi Komentar | Varchar (255) | Berisi komentar yang diinput aktor |

Gambar 5. Data dictionary profil mahasiswa, artikel, komentar

UPDATE ARTIKEL

Premary Key : Waktu update artikel

Foregin Key :-

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|----|----------------------|-------------------------------|--|
| 1 | Waktu Update Artikel | Timestamp (hh:mm:ss/yy:mm:dd) | Berisi waktu terupdatenya artikel |
| 2 | File Update Artikel | LONGBLOB (Size) | Berisi file artikel yang telah di update |
| 3 | Status Approval | Enum | Berisi status approval dari artikel |

PUBLISH JURNAL DAN ARTIKEL

Premary Key : Seri Jurnal

Foregin Key :-

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|----|------------------------------------|-------------------|---|
| 1 | Seri Jurnal | Integer (30) | Berisi nomor seri jurnal dan artikel |
| 2 | Judul Jurnal dan Artikel | String (255) | Berisi file artikel yang telah di update |
| 3 | Tanggal Publish Jurnal dan Artikel | Date (yyyy-mm-dd) | Berisi tanggal dipulishnya jurnal dan artikel |

INDEKSTASI JURNAL

Premary Key : Id Indekstasi

Foregin Key :-

Jumlah Filed : 3

| NO | Filed | Type/Lebar | Keterangan |
|----|--------------------------|--------------|--|
| 1 | Id indekstasi | Integer (30) | Berisi nomor id indekstasi dari jurnal |
| 2 | Jenis Indekstasi Jurnal | String (255) | Berisi jenis indekstasi dari jurnal |
| 3 | Status Indekstasi Jurnal | enum | Berisi status indekstasi dari jurnal |

Gambar 6. Data dictionary update artikel, publish jurnal dan artikel, indekstasi jurnal

3.5 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan alat bantu analisis yang digunakan untuk memodelkan alur data dalam sebuah sistem secara grafis. DFD membantu memperlihatkan bagaimana data mengalir dari entitas eksternal ke dalam sistem, bagaimana data diproses, disimpan, dan dihasilkan kembali sebagai output. DFD umumnya digunakan dalam tahap analisis kebutuhan sistem karena fokus pada alur data dan proses logis, bukan implementasi teknis.

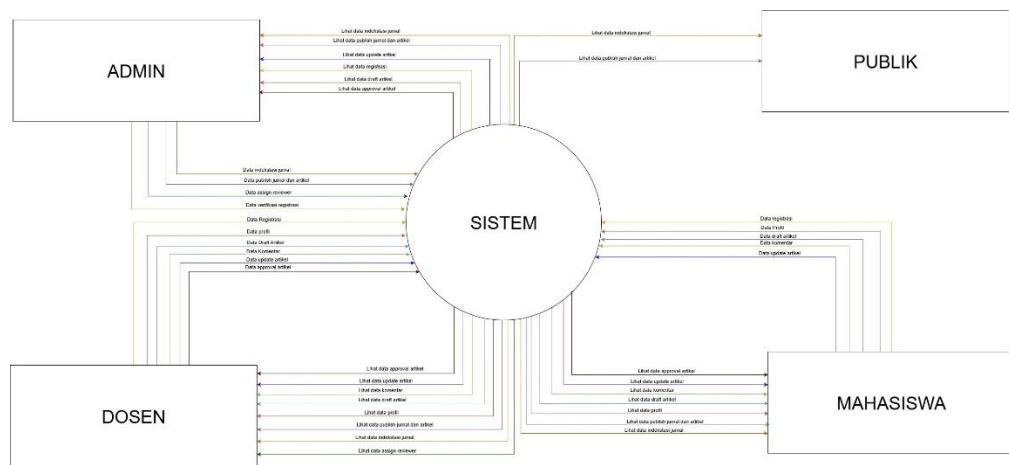
DFD biasanya ditulis dengan menggunakan notasi standar. Dua notasi yang paling populer adalah **Gane & Sarson** serta **Yourdon & DeMarco**.

| Notasi Yourdon DeMarco | Notasi Gane & Sarson | Deskripsi |
|---------------------------|----------------------|---|
| | | Simbol Entitas Eksternal / Terminator menggambarkan asal atau tujuan data di luar system |
| | | Simbol lingkaran menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar |
| | | Simbol aliran data menggambarkan aliran data |
| | | Simbol file menggambarkan tempat data disimpan |

Gambar 7. Notasi DFD

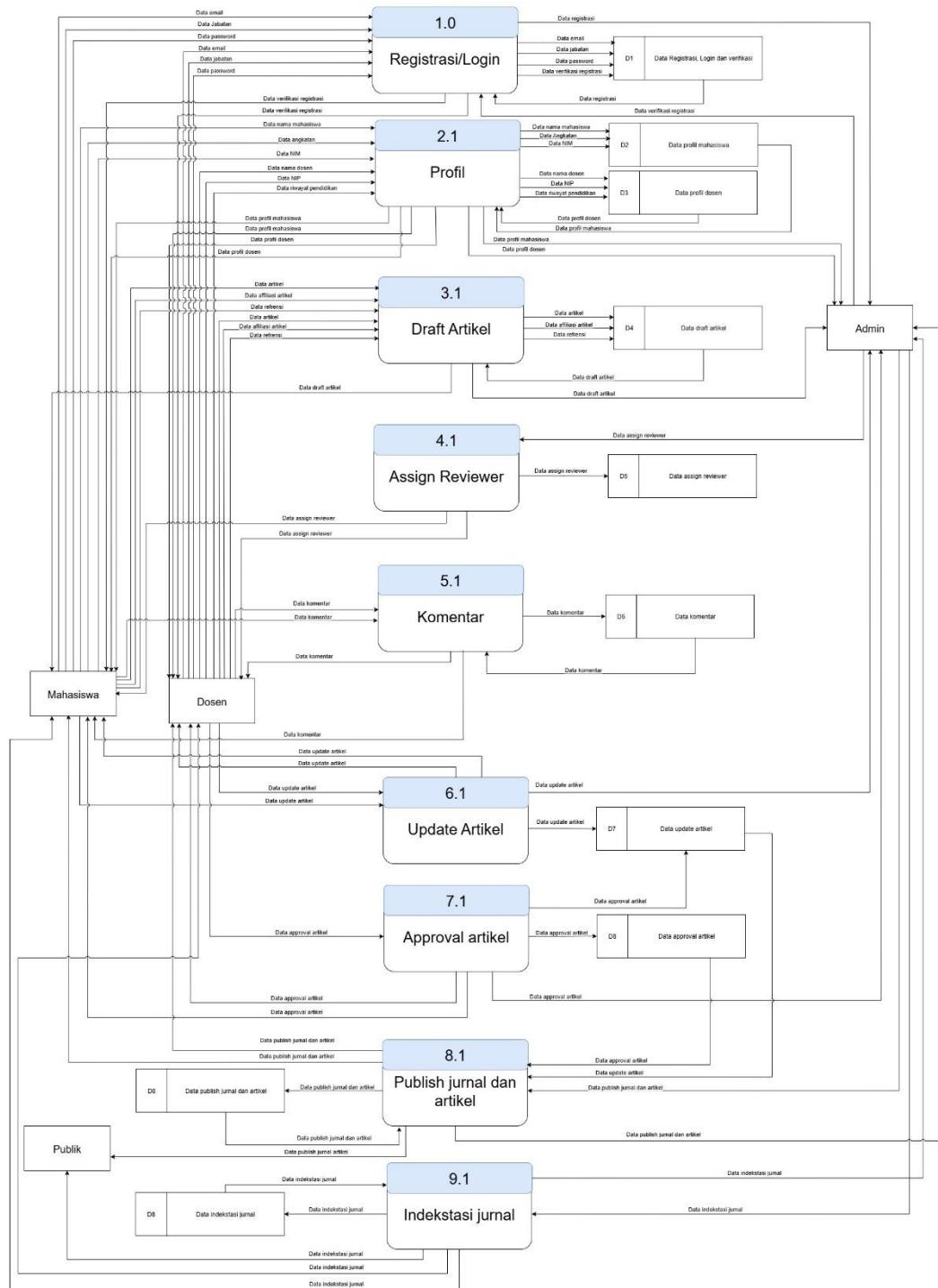
Perancangan DFD pada kasus kali ini menggunakan notasi menurut **Gane & Yarson** serta melakukan perincian sampai DFD Level 1

Data Flow Diagram (DFD) digunakan untuk menggambarkan aliran data dalam suatu sistem, termasuk hubungan antara entitas eksternal, proses, dan penyimpanan data. Pada DFD Level 0, yang sering disebut juga sebagai context diagram, sistem repository jurnal digambarkan secara menyeluruh sebagai satu proses tunggal yang berinteraksi dengan empat entitas utama, yaitu **Admin, Mahasiswa, Dosen, dan Publik**. Setiap entitas melakukan pertukaran data dengan sistem, misalnya admin mengelola data verifikasi, reviewer, hingga publikasi, mahasiswa dan dosen mengelola artikel, sementara publik dapat melihat data publikasi dan indeksasi. DFD Level 0 ini memberikan gambaran umum bagaimana sistem berhubungan dengan lingkungan eksternalnya.



Gambar 8. DFD Level 0

Sementara itu, DFD Level 1 menyajikan rincian lebih detail dari proses utama yang ada pada Level 0. Sistem repository jurnal diuraikan ke dalam beberapa proses inti, seperti registrasi, pengelolaan profil, draft artikel, penugasan reviewer, komentar, update artikel, approval artikel, publikasi artikel, dan indeksasi jurnal. Pada level ini terlihat jelas bagaimana masing-masing proses saling terhubung, data apa saja yang diproses, serta bagaimana data disimpan di dalam data store. Dengan adanya dekomposisi ini, alur data menjadi lebih transparan sehingga memudahkan pemahaman terhadap kebutuhan sistem.



Gambar 9. DFD Level 1

3.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah model konseptual yang digunakan untuk merepresentasikan struktur logis dari basis data dengan memvisualisasikan entitas, atribut, dan hubungan antar entitas. ERD

mempermudah pengembang dalam merancang basis data karena mampu menggambarkan hubungan data secara sistematis dan jelas.

1. Entitas (Entity)

Entitas adalah objek nyata atau konsep penting yang datanya perlu disimpan dalam sistem. Entitas digambarkan dalam bentuk persegi panjang. Contoh pada sistem repository jurnal adalah Mahasiswa, Dosen, Admin, Artikel, dan Indeksasi Jurnal. Masing-masing entitas merepresentasikan tabel dalam basis data relasional.

2. Atribut (Attribute)

Atribut merupakan karakteristik atau informasi yang dimiliki oleh suatu entitas. Dalam diagram, atribut digambarkan dengan elips yang terhubung ke entitas. Misalnya entitas Mahasiswa memiliki atribut seperti NIM, Nama, Email, dan Jurusan. Atribut inilah yang nantinya menjadi kolom dalam tabel basis data.

3. Relasi (Relationship)

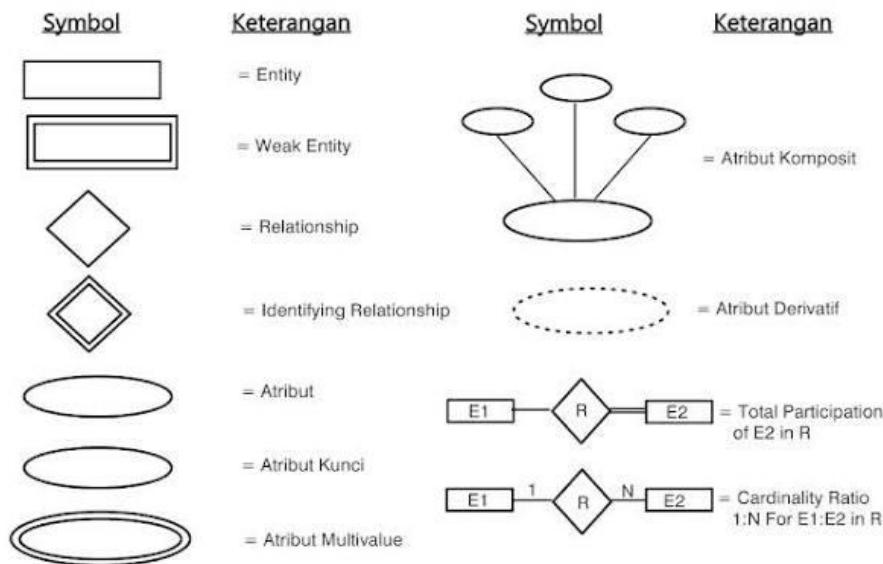
Relasi menunjukkan bagaimana dua atau lebih entitas saling berhubungan dalam sistem. Relasi digambarkan dengan belah ketupat. Contohnya, entitas Mahasiswa memiliki relasi dengan Artikel melalui hubungan “Mengunggah”, atau entitas Admin berelasi dengan Artikel melalui hubungan “Memverifikasi”. Relasi ini akan menjadi dasar pembentukan foreign key dalam implementasi basis data.

4. Kardinalitas (Cardinality)

Kardinalitas menunjukkan jumlah hubungan yang dapat terjadi antara entitas satu dengan entitas lainnya. Dalam ERD, kardinalitas digambarkan dengan notasi angka atau simbol, dan salah satu yang paling banyak digunakan adalah Crow’s Foot Notation. Notasi ini menggunakan garis dengan tanda khusus di ujung relasi untuk menunjukkan jenis hubungan.

- One to One (1:1) Artinya, satu entitas hanya berhubungan dengan satu entitas lainnya. Contoh: Satu akun admin hanya dimiliki oleh satu orang admin.

- One to Many (1:N) Artinya, satu entitas berhubungan dengan banyak entitas lain. Contoh: Satu dosen dapat mereview banyak artikel, tetapi satu artikel hanya direview oleh satu dosen tertentu.
- Many to Many (M:N) Artinya, banyak entitas berhubungan dengan banyak entitas lain. Contoh: Banyak mahasiswa dapat menulis banyak artikel, dan satu artikel juga bisa ditulis oleh lebih dari satu mahasiswa.

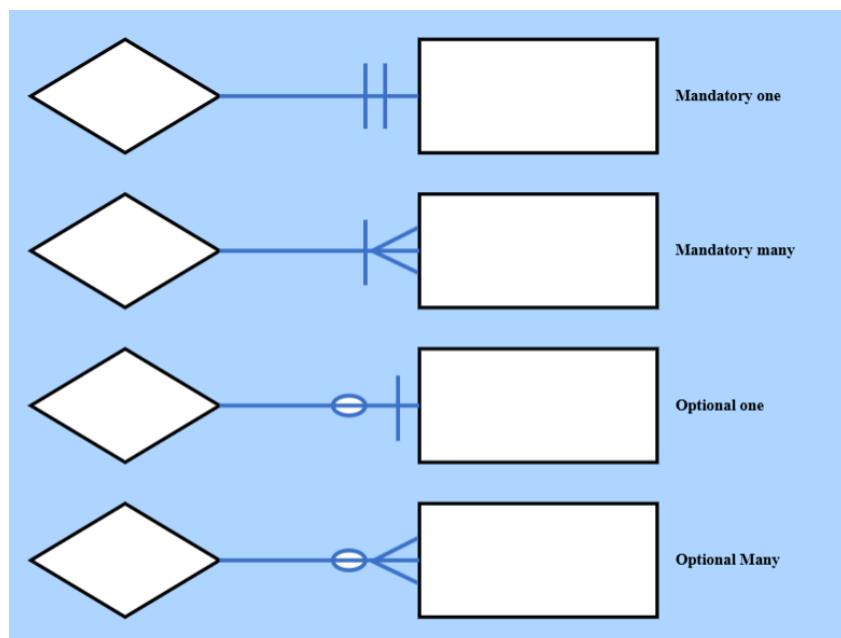


Gambar 10. Simbol ERD

Tabel 15. Simbol ERD

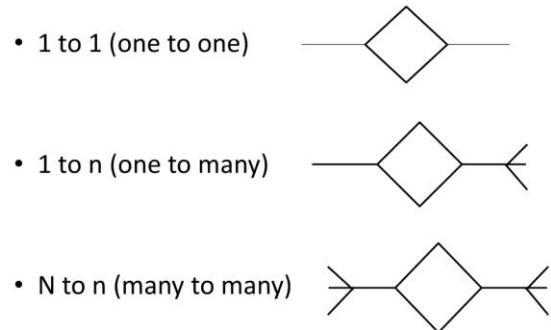
| Simbol | Nama | Keterangan / Pengertian |
|-----------------------|--------------------------|--|
| Persegi panjang | Entity | Menunjukkan himpunan objek yang datanya disimpan, misalnya Mahasiswa, Pelanggan. |
| Persegi panjang ganda | Weak Entity | Entitas lemah yang keberadaannya tergantung pada entitas lain (memerlukan identifying relationship). |
| Belah ketupat | Relationship | Menggambarkan hubungan antar entitas, misalnya Mengajar, Memesan. |
| Belah ketupat ganda | Identifying Relationship | Relasi yang menghubungkan entitas lemah dengan entitas pendukungnya. |

| | | |
|-----------------------------------|---------------------|---|
| Oval | Atribut | Menjelaskan karakteristik/field dari entitas, misalnya Nama, Alamat. |
| Oval bergaris bawah | Atribut Kunci | Atribut unik yang digunakan sebagai identifier, misalnya NIM atau ID Transaksi. |
| Oval ganda | Atribut Multivalue | Atribut yang dapat memiliki lebih dari satu nilai, misalnya Nomor Telepon. |
| Oval bercabang | Atribut Komposit | Atribut yang masih bisa diuraikan lagi, misalnya Alamat → Jalan, Kota, Kode Pos. |
| Oval putus-putus | Atribut Derivatif | Atribut yang nilainya diperoleh dari atribut lain, misalnya Umur dari Tanggal Lahir. |
| Garis dengan notasi penuh | Total Participation | Menunjukkan bahwa semua entitas pada himpunan berpartisipasi penuh dalam relasi. |
| Garis dengan notasi 1:1, 1:N, M:N | Cardinality Ratio | Menjelaskan jumlah maksimum keterhubungan antar entitas, misalnya One to One, One to Many, atau Many to Many. |



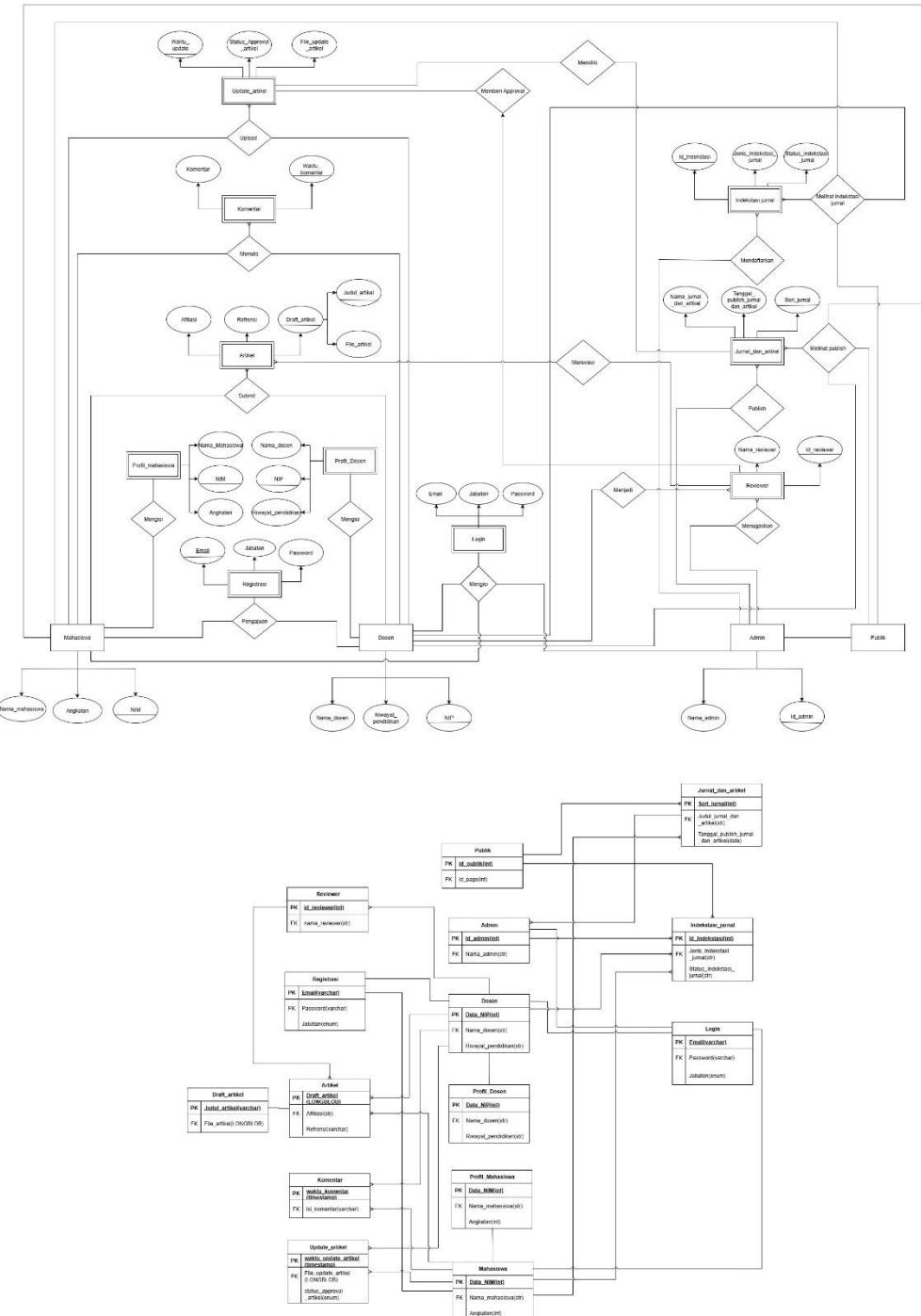
Gambar 11. Crow's Foot Notation

Simbol Kardinalitas ERD



Gambar 12. Kardinalitas ERD

Entity Relationship Diagram (ERD) pada sistem repository jurnal ini digunakan untuk memodelkan hubungan antar data yang dikelola dalam sistem. Diagram ini memperlihatkan entitas utama seperti Mahasiswa, Dosen, Admin, Publik, Registrasi, Profil Mahasiswa, Profil Dosen, Draft Artikel, Update Artikel, Komentar, Assign Reviewer, Approval Artikel, Publikasi Jurnal dan Artikel, serta Indeksasi Jurnal. Setiap entitas memiliki atribut yang merepresentasikan data yang disimpan, misalnya entitas Mahasiswa menyimpan data email, nama, jurusan, hingga NIM, sedangkan entitas Draft Artikel memuat data artikel, afiliasi, referensi, dan status revisi. Hubungan antar entitas digambarkan dengan garis relasi yang menunjukkan bagaimana data saling terhubung, contohnya antara mahasiswa dengan draft artikel atau antara admin dengan proses publikasi jurnal. Dengan demikian, ERD ini memberikan gambaran menyeluruh mengenai struktur basis data yang mendukung operasional sistem repository jurnal.

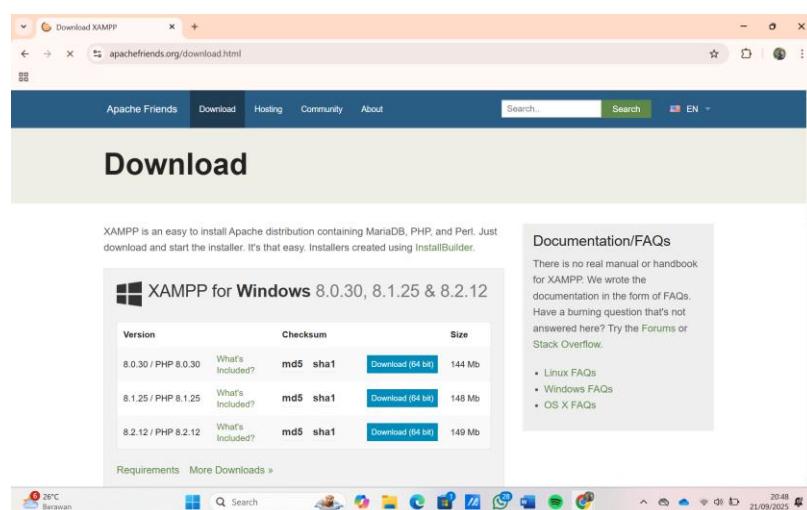


Gambar 13. ERD

4. Database MySql

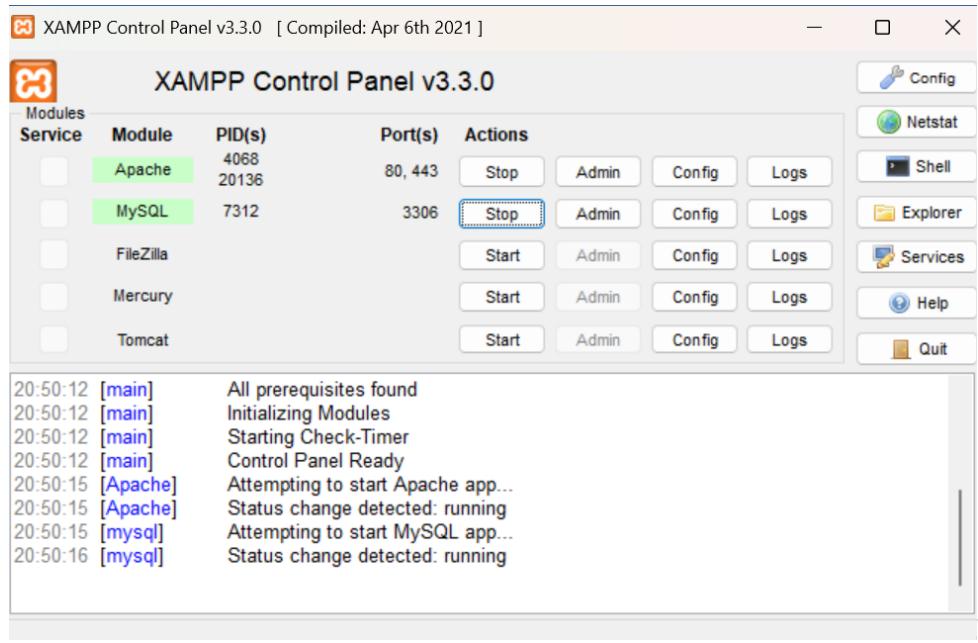
4.1 Persiapan membuat database MySql

Instalasi XAMPP merupakan tahapan fundamental dalam penyediaan lingkungan pengembangan aplikasi berbasis web yang membutuhkan integrasi dengan database MySQL. XAMPP, yang terdiri atas komponen utama Apache sebagai web server dan MySQL sebagai sistem manajemen basis data relasional, memberikan kemudahan bagi pengguna dalam membangun serta menguji aplikasi secara lokal. Proses koneksi antara aplikasi dengan MySQL diawali dengan aktivasi modul Apache dan MySQL melalui XAMPP Control Panel. Selanjutnya, pengelolaan basis data dapat dilakukan melalui antarmuka phpMyAdmin yang diakses menggunakan peramban dengan alamat <http://localhost/phpmyadmin>. Pada tahap ini, pengguna berwenang untuk melakukan pembuatan basis data, pengaturan struktur tabel, hingga manipulasi data. Integrasi antara aplikasi dan basis data kemudian diwujudkan melalui skrip PHP dengan memanfaatkan fungsi koneksi, seperti `mysqli_connect()` atau PHP Data Object (PDO). Dengan demikian, instalasi dan konfigurasi XAMPP tidak hanya menyediakan sarana praktis untuk menjalankan server-side scripting, tetapi juga memastikan ketersediaan basis data yang terintegrasi secara optimal dalam mendukung kebutuhan pengembangan sistem informasi berbasis web.



Gambar 14. Instalasi XAMPP

Proses koneksi antara aplikasi dengan MySQL diawali dengan aktivasi modul Apache dan MySQL melalui XAMPP Control Panel. Selanjutnya, pengelolaan basis data dapat dilakukan melalui antarmuka phpMyAdmin yang diakses menggunakan peramban dengan alamat <http://localhost/phpmyadmin>. Pada tahap ini, pengguna berwenang untuk melakukan pembuatan basis data, pengaturan struktur tabel, hingga manipulasi data. Integrasi antara aplikasi dan basis data kemudian diwujudkan melalui skrip PHP dengan memanfaatkan fungsi koneksi, seperti `mysqli_connect()` atau PHP Data Object (PDO). Dengan demikian, instalasi dan konfigurasi XAMPP tidak hanya menyediakan sarana praktis untuk menjalankan server-side scripting, tetapi juga memastikan ketersediaan basis data yang terintegrasi secara optimal dalam mendukung kebutuhan pengembangan sistem informasi berbasis web.



Ganbar 15. Koneksi XAMPP

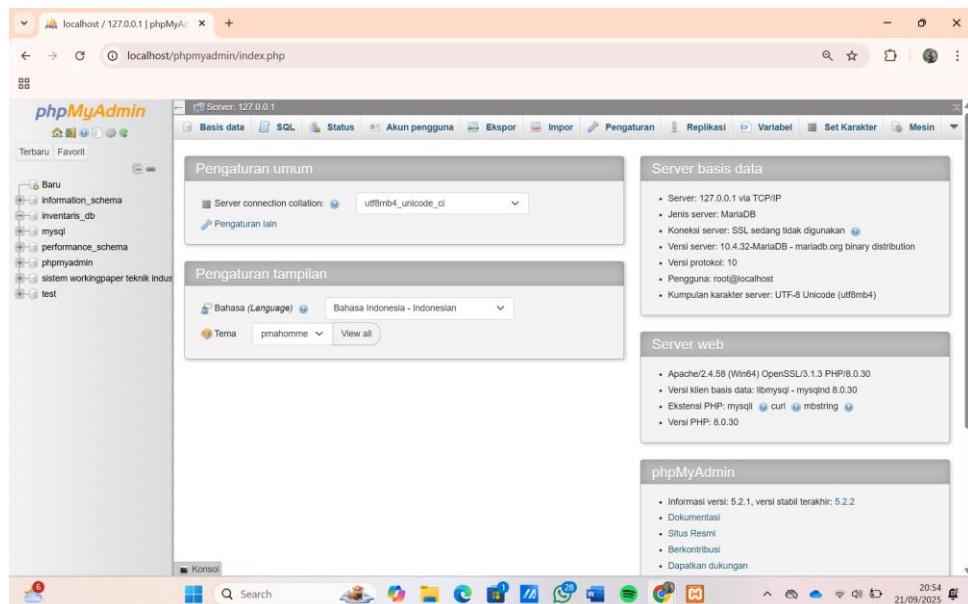
4.2 Pembuatan database dan tabel data menggunakan metode *Graphical User Interface(GUI)* dengan PHPMyAdmin MySQL

Pembuatan database dengan metode **GUI (Graphical User Interface)** menggunakan alat **phpMyAdmin** merupakan salah satu cara yang paling mudah dan efisien dalam pengelolaan basis data, khususnya bagi pengguna yang belum terbiasa dengan perintah SQL secara langsung. Melalui antarmuka grafis yang intuitif, phpMyAdmin memungkinkan pengguna untuk melakukan

berbagai aktivitas seperti membuat database, menambahkan tabel, mengatur relasi antar tabel, serta melakukan manajemen data tanpa perlu menulis sintaks SQL secara manual. Metode ini tidak hanya mempercepat proses pembuatan database, tetapi juga meminimalkan kesalahan penulisan perintah yang sering terjadi pada mode teks. Selain itu, phpMyAdmin memiliki berbagai fitur tambahan seperti impor dan ekspor data, pengaturan hak akses pengguna, serta analisis struktur database yang sangat membantu dalam pengembangan aplikasi berbasis web, terutama yang menggunakan **bahasa pemrograman PHP** dan **sistem manajemen basis data MySQL**.

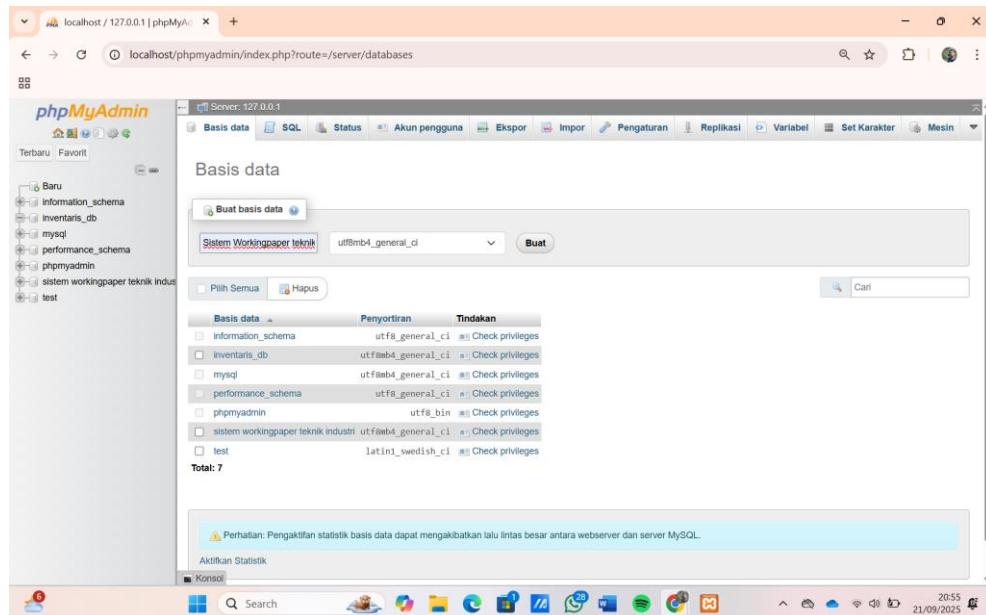
Proses awal dalam penyetelan basis data pada MySQL adalah pembuatan basis data baru. Langkah ini dapat dilakukan melalui antarmuka phpMyAdmin dengan memilih menu Databases kemudian memberikan nama sesuai kebutuhan. Basis data berfungsi sebagai wadah utama yang akan menampung berbagai tabel dan objek lain yang mendukung sistem. Tahap ini penting karena menjadi fondasi bagi keseluruhan pengelolaan data yang akan digunakan pada aplikasi atau sistem informasi.

Pastikan koneksi ke apache dan MySQL sudah berjalan kemudian buka browser dan akses halaman berikut <http://localhost/phpmyadmin/index.php>



Gambar 16. Tampilan awal MySQL

Pilih baru kemudian masukkan nama database yang digunakan.



Gambar 17. Pembuatan database

Tahap kedua adalah pembuatan tabel dalam basis data yang telah dibuat. Pada langkah ini, pengguna menentukan jumlah kolom yang diperlukan, memberi nama atribut, memilih tipe data yang sesuai, serta menetapkan kunci primer (*primary key*) untuk memastikan setiap baris data memiliki identitas unik. Misalnya, sebuah tabel mahasiswa dapat terdiri dari atribut NIM (INT, Primary Key), nama (VARCHAR), dan angkatan (VARCHAR). Desain tabel yang baik akan mempengaruhi efisiensi pengelolaan data dan konsistensi integritas basis data secara keseluruhan.

Pembuatan isi tabel data mengacu pada dictionary yang telah dibuat pada bab sebelumnya

| Create new table | |
|-------------------------------------|--------------|
| Nama tabel | Jumlah kolom |
| profil_mahasiswa | 3 |
| <input type="button" value="Buat"/> | |

Gambar 18. Pembuatan tabel

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---------------------|--------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| 1 | id_profil_mahasiswa | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | id_mahasiswa | int(11) | | | No | None | | | Change Drop More |
| 3 | nama | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | nim | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | angkatan | int(11) | | | Yes | NULL | | | Change Drop More |

Gambar 19. Pembuatan tabel data

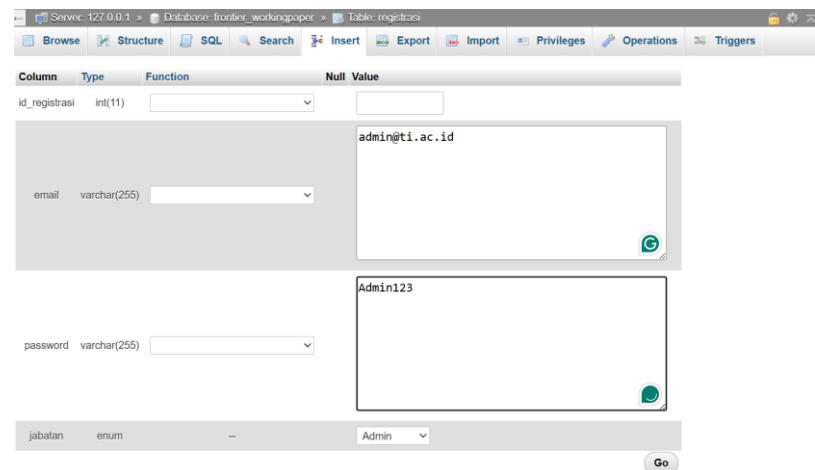
Hasil pembuatan basis data dengan nama **frontier_workingpaper** pada phpMyAdmin menunjukkan bahwa struktur basis data telah berhasil dibangun secara sistematis dan sesuai dengan kebutuhan sistem pengelolaan *working paper* Teknik Industri. Pada basis data tersebut, telah dibuat sebelas tabel utama yang saling terhubung untuk mendukung proses pengelolaan data, yaitu **admin**, **arsip_workingpaper**, **dosen**, **draft_workingpaper**, **login**, **mahasiswa**, **profil_dosen**, **profil_mahasiswa**, **publish_workingpaper**, **registrasi**, dan **review_draft**.

Masing-masing tabel memiliki struktur dan fungsi yang berbeda sesuai perannya, seperti tabel **draft_workingpaper** yang menyimpan data naskah awal (*draft*) mahasiswa, tabel **review_draft** untuk menampung hasil penilaian dosen pembimbing, serta tabel **publish_workingpaper** yang berfungsi sebagai penyimpanan data naskah yang telah diterbitkan. Selain itu, tabel **profil_mahasiswa** dan **profil_dosen** digunakan untuk mengelola informasi identitas pengguna, sedangkan tabel **login** dan **registrasi** mendukung proses autentikasi serta pendaftaran akun.

| Tabel | Tindakan | Baris | Seimbang | Pembuktian | Ukuran | Beban |
|----------------------------|----------|--------|----------|------------|--------|--|
| artikel | | 16.0 | 0 | 43 | - | |
| indeksasi_jurnal | | 16.0 | 0 | 43 | - | |
| komentar | | 16.0 | 0 | 43 | - | |
| login | | 16.0 | 0 | 43 | - | |
| profil_dosen | | 16.0 | 0 | 43 | - | |
| profil_mahasiswa | | 16.0 | 0 | 43 | - | |
| publish_jurnal_dan_artikel | | 16.0 | 0 | 43 | - | |
| registrasi | | 16.0 | 0 | 43 | - | |
| update_artikel | | 16.0 | 0 | 43 | - | |
| update_artikel | | 16.0 | 0 | 43 | - | |
| 9 tabel | | Jumlah | | | | e InnoDB utf8mb4_general_ci 144.0 kB 0 B |

Gambar 20. Hasil pembuatan database dan tabel data

Langkah selanjutnya adalah menambahkan data pada tabel data yang sudah dibuat. Cara nya adalah menekan tombol “tambahkan di bagian tabel data yang ingin ditambahkan kemudian isi *value* sesuai dengan isi data yang diinginkan



Gambar 21. Memasukan *Value* data

Setelah data berhasil dimasukkan, hasil data dapat dilihat pada menu jelajahi pada tabel yang telah ditambahkan.

| | T | id_registrasi | email | password | jabatan |
|--------------------------|----------------------|-------------------------------|-----------------------|--------------------------|-------------------------|
| <input type="checkbox"/> | Edit | 1 | admin@ti.ac.id | admin123 | Admin |
| <input type="checkbox"/> | Edit | 2 | dosen1@ti.ac.id | dosen123 | Dosen |
| <input type="checkbox"/> | Edit | 3 | mahasiswa1@ti.ac.id | mhs123 | Mahasiswa |
| <input type="checkbox"/> | Edit | 4 | mahasiswa2@ti.ac.id | mhs456 | Mahasiswa |

Gambar 22. Hasil penambahan data

Langkah selanjutnya adalah membuat relasi antar tabel, sebelum membuat relasi, pastikan seluruh kolom foreign key telah di indeks dengan cara memilih menu struktur dan centang kolom foreign key kemudian klik indeks

The screenshot shows the 'login' table structure in phpMyAdmin. The table has five columns: id_login, id_registrasi, email, password, and jabatan. Primary keys are defined for both id_login and id_registrasi. An index is also present on the jabatan column.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------------------|---|--------------------|------------|------|---------|----------|----------------|--------------|
| 1 | id_login | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop |
| 2 | id_registrasi | int(11) | | | Yes | NULL | | | Change Drop |
| 3 | email | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop |
| 4 | password | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop |
| 5 | jabatan | enum('dosen', 'mahasiswa', 'admin', '') | utf8mb4_general_ci | | No | None | | | Change Drop |

Indexes

| Action | Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|------------|----------------------|-------|--------|--------|---------------|-------------|-----------|------|---------|
| Edit Drop | PRIMARY | BTREE | Yes | No | id_login | 4 | A | No | |
| Edit Drop | id_registrasi | BTREE | No | No | id_registrasi | 4 | A | Yes | |

Create an index on 1 columns Go

Gambar 23. Hasil Indeks

Selanjutnya adalah membuat relasi antar tabel dengan cara menghubungkan *primary key* pada *foreign key* dengan hubungan relasi dijelaskan pada tabel 16

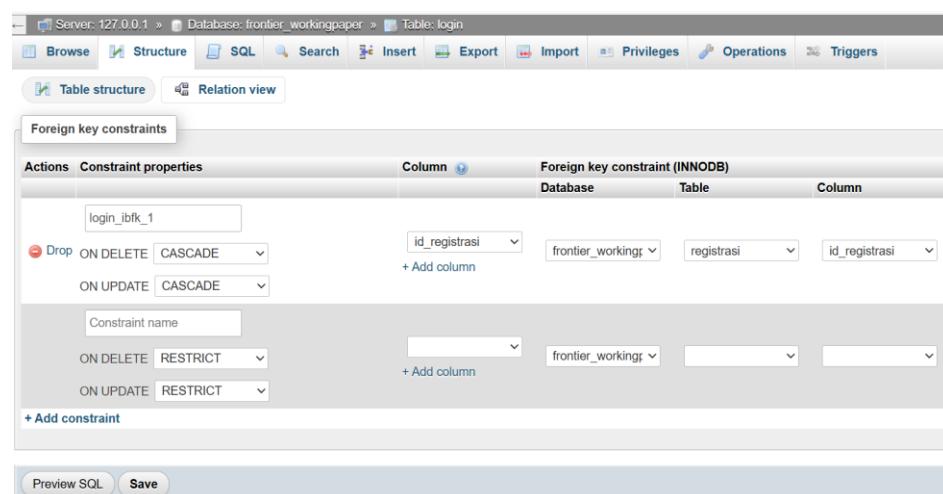
Tabel 16. Hubungan relasi MYSQL

| Nama Atribut / Opsi | Keterangan / Arti | Fungsi dalam Relasi / Integritas Data | Contoh Kasus di phpMyAdmin |
|---------------------------|---|---|---|
| ON DELETE CASCADE | Jika baris data pada tabel induk dihapus, maka semua baris data terkait di tabel anak juga ikut terhapus otomatis. | Menjaga konsistensi data agar tidak ada data yatim (orphan data) di tabel anak. | Jika data pelanggan dihapus, maka semua transaksi yang berhubungan dengan pelanggan tersebut juga terhapus. |
| ON UPDATE CASCADE | Jika nilai kunci pada tabel induk diubah, maka nilai yang berelasi pada tabel anak ikut diperbarui secara otomatis. | Memastikan kesesuaian nilai referensi antar tabel setelah pembaruan data. | Jika id_pelanggan berubah dari C01 menjadi C05, maka nilai id_pelanggan di tabel transaksi juga ikut berubah. |
| ON DELETE SET NULL | Jika data di tabel induk dihapus, maka kolom foreign key di | Data anak tetap ada, namun hubungan ke | Saat pelanggan dihapus, id_pelanggan di tabel transaksi |

| | | | |
|----------------------------|---|---|---|
| | tabel anak akan diset menjadi NULL. | induk dihapus dengan aman. | menjadi NULL (tidak terhubung ke pelanggan mana pun). |
| ON UPDATE SET NULL | Jika data induk diperbarui, maka kolom anak yang berelasi akan diubah menjadi NULL. | Menghindari perubahan otomatis pada tabel anak saat nilai induk berubah. | Saat <code>id_pelanggan</code> berubah di tabel induk, nilai di tabel anak diset NULL. |
| ON DELETE RESTRICT | Menolak penghapusan data di tabel induk jika masih ada data terkait di tabel anak. | Melindungi data agar tidak terhapus secara tidak sengaja ketika masih dipakai tabel lain. | phpMyAdmin akan menolak penghapusan pelanggan yang masih memiliki transaksi. |
| ON UPDATE RESTRICT | Menolak pembaruan data di tabel induk jika ada tabel anak yang masih berelasi. | Menjaga stabilitas referensi antar tabel. | Tidak bisa mengubah <code>id_pelanggan</code> jika masih dipakai pada tabel transaksi. |
| ON DELETE NO ACTION | Mirip dengan RESTRICT maka perubahan atau penghapusan tidak diizinkan jika masih ada relasi aktif. | Menegakkan integritas referensial secara ketat. | phpMyAdmin akan memunculkan error jika penghapusan pelanggan melanggar relasi. |
| ON UPDATE NO ACTION | Tidak mengizinkan pembaruan nilai kunci pada tabel induk selama masih ada data yang bergantung padanya. | Menghindari inkonsistensi akibat pembaruan data di tabel induk. | Tidak dapat memperbarui <code>id_pelanggan</code> karena masih digunakan di tabel lain. |
| NOT NULL | Menentukan bahwa kolom tidak boleh kosong (NULL). | Menjamin bahwa setiap entri data memiliki nilai yang pasti (tidak kosong). | Kolom <code>nama_pelanggan</code> diset NOT NULL agar setiap |

| | | | |
|------|--|---|---|
| | | | pelanggan harus memiliki nama. |
| NULL | Mengizinkan kolom berisi nilai kosong. | Memberikan fleksibilitas jika ada data yang belum tersedia. | Kolom alamat bisa dibiarkan kosong (NULL) bila pelanggan belum memberikan alamat. |

Untuk menambahkan relasi, pilih struktur pada tabel data, kemudian tampilkan hubungan, lalu hubungkan antara primary key dan foreign key antar tabel.

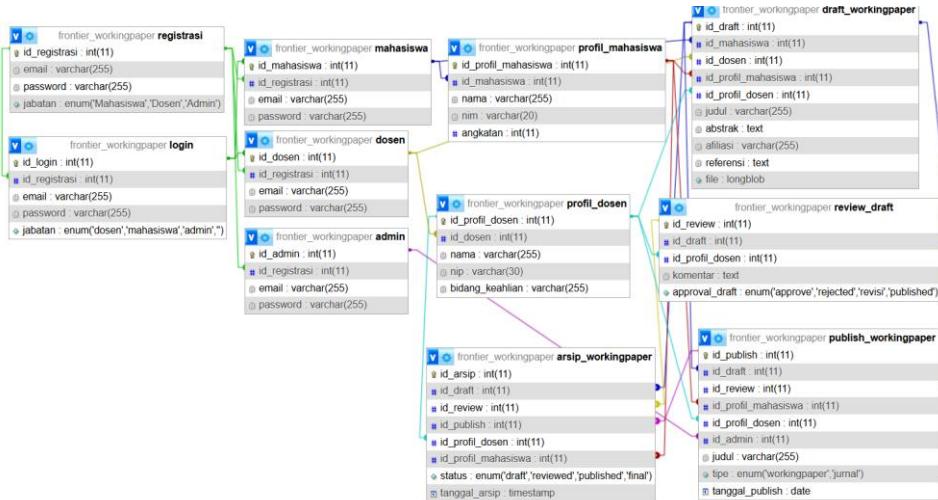


Gambar 24. Penambahan relasi

Dengan adanya struktur tabel yang terorganisasi dan memiliki relasi antar entitas tersebut, basis data **frontier_workingpaper** mampu menampung, mengolah, serta menyajikan data secara terintegrasi, sehingga mendukung kinerja sistem informasi publikasi *working paper* yang efisien dan terstruktur.

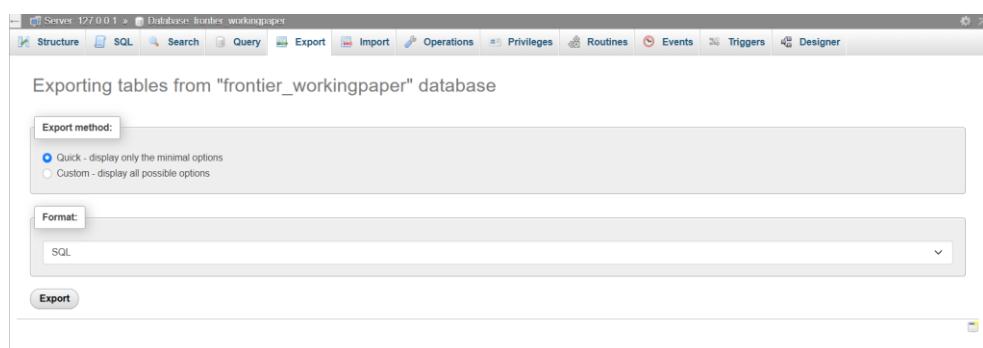
4.3 Struktur dan Ekspor coding MySQLPhp

Untuk meninjau rancangan struktur tabel pada phpMyAdmin, pengguna dapat memilih opsi Struktur yang tersedia pada masing-masing tabel dalam basis data. Melalui menu ini, ditampilkan detail desain tabel yang meliputi nama atribut, tipe data, panjang karakter, pengaturan kunci primer, serta sifat tambahan seperti auto increment atau not null. Informasi tersebut memberikan gambaran yang jelas mengenai desain data yang telah direncanakan serta memungkinkan pengguna untuk melakukan modifikasi apabila diperlukan.



Gambar 25. Struktur desain MySQL

Selanjutnya, apabila rancangan maupun data dalam basis data ingin disimpan atau dipindahkan, phpMyAdmin menyediakan fasilitas Ekspor. Melalui menu ini, pengguna dapat mengekspor basis data atau tabel tertentu ke dalam format skrip MySQL. File hasil ekspor ini berisi perintah-perintah SQL seperti CREATE TABLE dan INSERT INTO yang dapat dijalankan kembali pada server lain untuk merekonstruksi struktur maupun isi basis data. Dengan demikian, fasilitas ekspor tidak hanya berfungsi sebagai sarana pencadangan, tetapi juga sebagai cara untuk mendistribusikan atau memigrasikan basis data secara efisien.



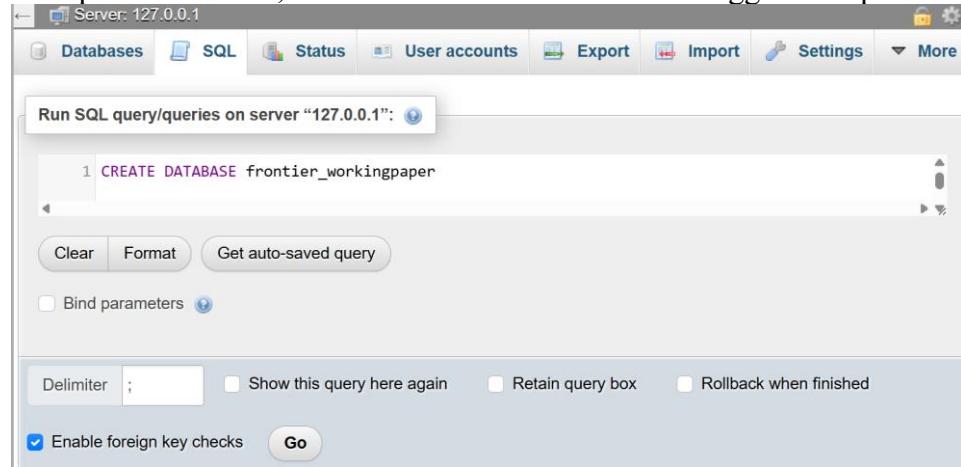
Gambar 26. Eksport data MySQL

4.4 Membuat database dengan SQL Code

1. Membuat database

Tahap pertama adalah mendefinisikan basis data baru dengan perintah CREATE DATABASE frontier_workingpaper;. Perintah ini akan

membuat wadah utama tempat seluruh tabel dan objek penyusun sistem disimpan. Setelah itu, basis data tersebut diaktifkan menggunakan perintah



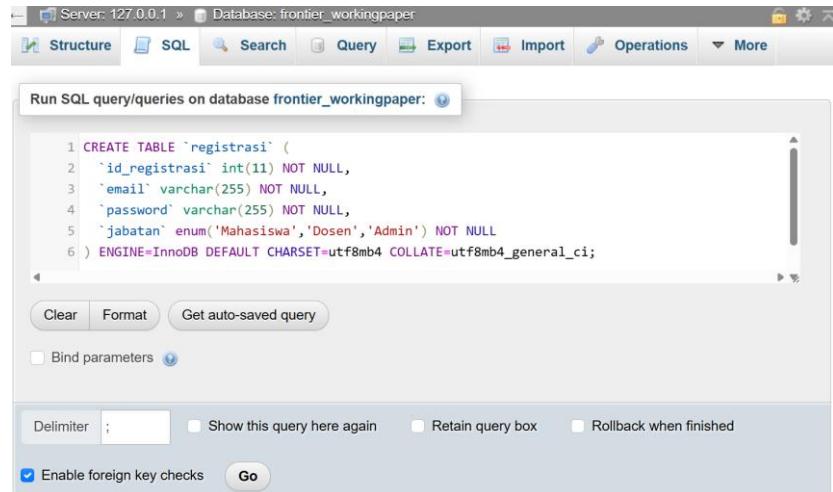
The screenshot shows the MySQL Workbench interface. At the top, it says "Server: 127.0.0.1". Below that is a toolbar with icons for Databases, SQL, Status, User accounts, Export, Import, Settings, and More. The main area is titled "Run SQL query/queries on server '127.0.0.1'". It contains a single line of SQL code: "1 CREATE DATABASE frontier_workingpaper". Below the code are several buttons: Clear, Format, Get auto-saved query, Bind parameters, Delimiter (set to ;), Show this query here again, Retain query box, Rollback when finished, and Enable foreign key checks (which is checked). A "Go" button is also present.

Gambar 27. Pembuatan database SQL code

2. Membuat tabel data

Selanjutnya dilakukan pembuatan tabel-tabel utama seperti admin, dosen, mahasiswa, profil_dosen, profil_mahasiswa, draft_workingpaper, review_draft, publish_workingpaper, arsip_workingpaper, login, dan registrasi.

Masing-masing tabel dibuat menggunakan perintah CREATE TABLE, dengan mendefinisikan nama kolom, tipe data, panjang karakter, serta atribut tambahan seperti NOT NULL dan AUTO_INCREMENT. Misalnya, tabel mahasiswa memiliki atribut id_mahasiswa, id_registrasi, email, dan password yang berfungsi untuk menyimpan data akun pengguna mahasiswa.



The screenshot shows the MySQL Workbench interface with the database "frontier_workingpaper" selected. The toolbar includes Structure, SQL, Search, Query, Export, Import, Operations, and More. The main area is titled "Run SQL query/queries on database 'frontier_workingpaper':". It contains the following SQL code for creating a table:

```

1 CREATE TABLE `registrasi` (
2     `id_registrasi` int(11) NOT NULL,
3     `email` varchar(255) NOT NULL,
4     `password` varchar(255) NOT NULL,
5     `jabatan` enum('Mahasiswa','Dosen','Admin') NOT NULL
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Below the code are standard MySQL Workbench controls: Clear, Format, Get auto-saved query, Bind parameters, Delimiter (set to ;), Show this query here again, Retain query box, Rollback when finished, and Enable foreign key checks (checked).

Gambar 28. Pembuatan tabel data dengan sql code

3. Membuat primary key dan foreign key/index

Setelah tabel selesai dibuat, setiap tabel diberi kunci primer (PRIMARY KEY) untuk memastikan keunikan data. Selanjutnya, relasi antar tabel dibangun melalui kunci tamu (FOREIGN KEY) yang menghubungkan tabel satu dengan tabel lainnya.

Contohnya, id_registrasi pada tabel mahasiswa menjadi *foreign key* yang terhubung dengan id_registrasi pada tabel registrasi. Proses ini menggunakan perintah ALTER TABLE ... ADD CONSTRAINT ... FOREIGN KEY ... REFERENCES Dengan adanya indeks dan relasi ini, integritas referensial antar tabel dapat terjaga dengan baik.

```

1 ALTER TABLE `login`
2   ADD PRIMARY KEY (`id_login`),
3   ADD KEY `id_registrasi` (`id_registrasi`);

```

Run SQL query/queries on database **frontier_workingpaper**:

Bind parameters

Delimiter : Show this query here again Retain query box Rollback when finished

Enable foreign key checks

Gambar 29. Pembuatan primary key dan foreign key dengan sql code

4. Menambahkan relasi antar tabel

Langkah berikutnya adalah mengatur perilaku relasi dengan opsi seperti ON DELETE CASCADE, ON UPDATE CASCADE, dan ON DELETE SET NULL.

Pengaturan ini memastikan bahwa perubahan atau penghapusan data di tabel induk akan otomatis memengaruhi tabel anak yang terhubung. Misalnya, jika data dosen dihapus, maka data draft yang berelasi juga ikut terhapus untuk mencegah *orphan data*.

```

1 ALTER TABLE `login`
2   ADD CONSTRAINT `login_ibfk_1` FOREIGN KEY (`id_registrasi`) REFERENCES
`registrasi` (`id_registrasi`) ON DELETE CASCADE ON UPDATE CASCADE;

```

Run SQL query/queries on database **frontier_workingpaper**:

Bind parameters

Delimiter : Show this query here again Retain query box Rollback when finished

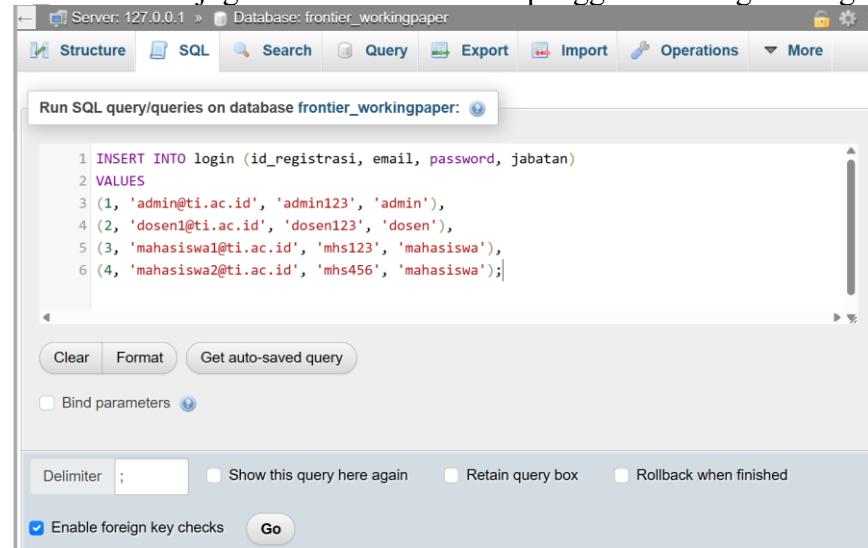
Enable foreign key checks

Gambar 30. Penambahan relasi dengan sql code

5. Menambahkan data ke tabel

Setelah struktur tabel selesai, proses selanjutnya adalah memasukkan data awal (*dummy data*) ke setiap tabel menggunakan perintah INSERT INTO.

Contohnya, tabel registrasi diisi dengan data pengguna awal seperti admin, dosen, dan mahasiswa. Data pada tabel lain seperti profil_dosen dan profil_mahasiswa juga diisi sesuai identitas pengguna masing-masing.



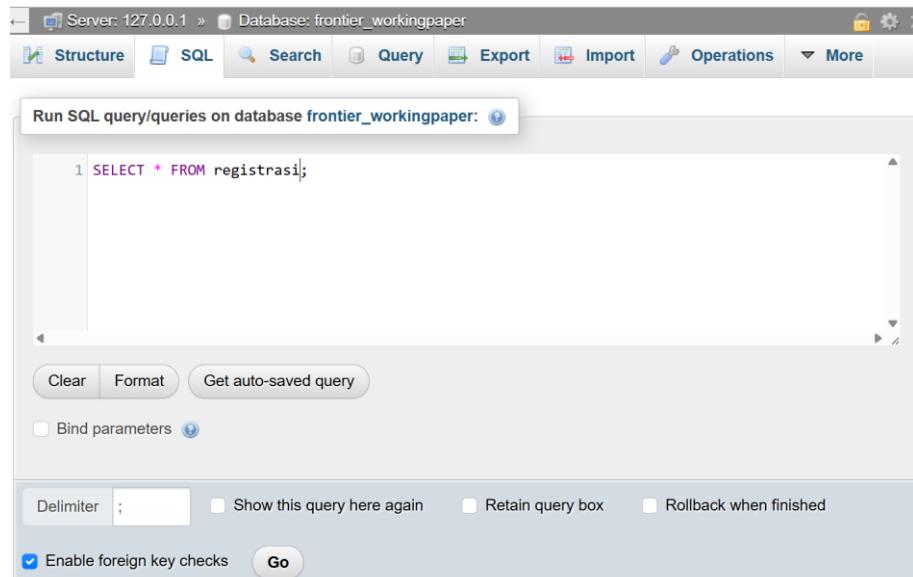
```
1 INSERT INTO login (id_registrasi, email, password, jabatan)
2 VALUES
3 (1, 'admin@ti.ac.id', 'admin123', 'admin'),
4 (2, 'dosen1@ti.ac.id', 'dosen123', 'dosen'),
5 (3, 'mahasiswa1@ti.ac.id', 'mhs123', 'mahasiswa'),
6 (4, 'mahasiswa2@ti.ac.id', 'mhs456', 'mahasiswa');
```

Gambar 31. Penambahan data ke tabel dengan sql code

6. Menampilkan keseluruhan isi tabel data

Untuk memastikan data telah tersimpan dengan benar, digunakan perintah SELECT * FROM nama_tabel;

Perintah ini menampilkan seluruh isi tabel yang telah dibuat sehingga pengguna dapat melakukan pengecekan dan verifikasi hasil input data.

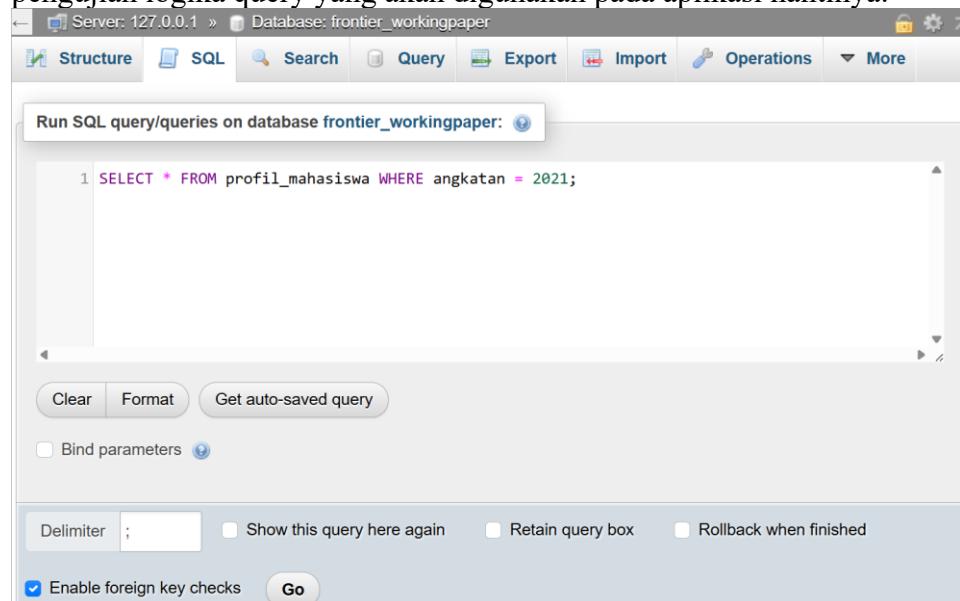


```
1 SELECT * FROM registrasi;
```

Gambar 32. Menampilkan keseluruhan isi tabel data dengan sql code

7. Menampilkan isi data tertentu dari tabel data

Terakhir, untuk melihat data spesifik, digunakan perintah SELECT dengan klausa WHERE. Misalnya, SELECT * FROM mahasiswa WHERE email='mahasiswa1@ti.ac.id'; akan menampilkan hanya data mahasiswa dengan email tertentu. Langkah ini penting dalam proses validasi serta pengujian logika query yang akan digunakan pada aplikasi nantinya.



The screenshot shows the MySQL Workbench interface. At the top, it displays 'Server: 127.0.0.1' and 'Database: frontier_workingpaper'. Below the header are tabs for 'Structure', 'SQL' (which is selected), 'Search', 'Query', 'Export', 'Import', 'Operations', and 'More'. The main area is titled 'Run SQL query/queries on database frontier_workingpaper'. Inside, there is a code editor containing the following SQL query:

```
1 SELECT * FROM profil_mahasiswa WHERE angkatan = 2021;
```

Below the code editor are several buttons: 'Clear', 'Format', 'Get auto-saved query', and 'Bind parameters'. At the bottom of the editor are settings for 'Delimiter' (set to ;), 'Show this query here again' (unchecked), 'Retain query box' (unchecked), 'Rollback when finished' (unchecked), and 'Enable foreign key checks' (checked). A 'Go' button is located at the bottom right of the editor.

Gambar 33. Menampilkan isi data tertentu dari tabel data dengan sql code

5. Perancangan Antarmuka Pengguna (UI) dan Pengalaman Pengguna (UX) Menggunakan Figma

5.1 Pengantar UI dan UX

Perancangan antarmuka pengguna (*User Interface / UI*) dan pengalaman pengguna (*User Experience / UX*) merupakan tahapan penting dalam proses pengembangan sistem informasi berbasis web. UI berfokus pada aspek visual yang dilihat dan digunakan oleh pengguna, sedangkan UX menekankan bagaimana pengguna berinteraksi, merasakan kemudahan, serta memperoleh manfaat dari sistem yang dibuat.

Penerapan prinsip UI/UX yang baik dapat meningkatkan efisiensi kerja pengguna dan memperkuat kesan profesional pada sistem yang dikembangkan. Pada sistem **Working Paper Teknik Industri**, konsep UI/UX dirancang untuk mendukung kemudahan mahasiswa, dosen, dan admin dalam mengelola publikasi ilmiah, mulai dari proses unggah naskah (*draft*), peninjauan (*review*), hingga publikasi akhir (*publish*).

Desain ini tidak hanya menampilkan tampilan visual yang menarik, tetapi juga mengutamakan **alur kerja yang logis, navigasi yang konsisten, dan aksesibilitas bagi pengguna publik** yang ingin menelusuri hasil publikasi tanpa harus login terlebih dahulu. Dengan demikian, sistem ini diharapkan dapat menjadi platform ilmiah yang informatif, ramah pengguna, dan terintegrasi.

5.2 Landasan Perancangan UI/UX

Dalam proses perancangan UI/UX, digunakan prinsip dasar berikut:

1. Konsistensi (Consistency) : Setiap elemen seperti warna, ikon, dan tipografi dibuat seragam untuk membangun identitas visual sistem.
2. Keterbacaan (Readability) : Pemilihan font dan ukuran teks disesuaikan agar informasi dapat dibaca dengan nyaman oleh pengguna di berbagai perangkat.
3. Kesederhanaan (Simplicity) : Desain diupayakan minimalis dengan fokus pada fungsi utama, sehingga pengguna tidak merasa terbebani oleh elemen visual yang berlebihan.
4. Navigasi yang Mudah (Ease of Navigation) : Menu dan tombol ditempatkan pada posisi yang intuitif sehingga pengguna dapat menjelajah sistem tanpa kebingungan.
5. Responsif dan Aksesibel (Responsive & Accessible) : Desain menyesuaikan tampilan di berbagai ukuran layar (desktop, tablet, dan smartphone).

Prinsip-prinsip tersebut menjadi dasar dalam setiap tahapan perancangan di aplikasi Working Paper Teknik Industri.

5.3 Tools Desain Figma

Perancangan antarmuka sistem dilakukan menggunakan Figma, sebuah aplikasi berbasis cloud yang memungkinkan kolaborasi desain secara real-time.

Figma dipilih karena memiliki fitur unggulan berikut:

- Frame dan Auto Layout : untuk menyusun struktur halaman yang adaptif terhadap ukuran layar pengguna.
- Component dan Variants : untuk membuat elemen yang dapat digunakan kembali seperti tombol, form login, card publikasi, dan navigasi header.
- Prototype Mode : untuk mensimulasikan interaksi antarhalaman, seperti klik tombol login menuju dashboard.
- Style System : untuk mengatur warna utama, font, dan ikon agar seragam di seluruh halaman.
- Commenting and Collaboration Tools : untuk memberikan catatan desain dan umpan balik secara langsung antaranggota tim.

Penggunaan Figma dalam proyek ini mempercepat tahapan perancangan, memperkecil kesalahan visual, serta memastikan keselarasan antara rancangan desain dan implementasi kode di tahap selanjutnya.

5.4 Pemanfaatan plugin icon8

Untuk memperkaya elemen visual, digunakan plugin Icon8 (Icons8) yang menyediakan ribuan ikon dengan berbagai gaya.

Plugin ini sangat membantu dalam memperindah antarmuka tanpa mengganggu kesederhanaan tampilan.

Ikon-ikon yang diambil dari Icon8 digunakan pada bagian:

- Navigasi utama (ikon rumah, profil, upload).
- Form aksi seperti tombol kirim, unggah, dan simpan.
- Halaman publikasi sebagai indikator status working paper (approved, rejected, published).

Penggunaan plugin Icon8 tidak hanya menambah kejelasan fungsi tiap tombol atau menu, tetapi juga memperkuat visualisasi makna setiap elemen dalam sistem. Gaya ikon yang seragam menciptakan konsistensi visual di seluruh halaman, sehingga meningkatkan pengalaman pengguna secara keseluruhan.

5.5 Tampilan halaman publik

Sistem ini menyediakan halaman publik (public view) yang dapat diakses tanpa login. Tujuannya adalah agar pengunjung umum, seperti mahasiswa lain, dosen, atau masyarakat, dapat melihat publikasi yang telah diterbitkan.

Adapun tampilan publik mencakup beberapa bagian utama berikut:

1. Beranda (Home Page)

Beranda berfungsi sebagai pintu utama sistem yang menampilkan identitas jurnal dan working paper Teknik Industri. Di bagian atas terdapat header dengan menu navigasi yang terdiri dari:

- Logo dan judul sistem (“Jurnal & Working Paper Teknik Industri”)
- Menu: Beranda, Jurnal, Working Paper, Login, dan Registrasi
- Pilihan peran pengguna (Mahasiswa, Dosen, Admin)

Bagian utama beranda menampilkan highlight publikasi terbaru dalam bentuk card grid, berisi judul, penulis, tanggal publikasi, serta tombol View untuk melihat detail jurnal atau working paper tertentu.

Tampilan ini memberikan kesan profesional dan informatif bagi pengunjung pertama kali.



Gambar 34. Tampilan beranda

2. Halaman Detail Jurnal dan Working Paper

Halaman ini menyajikan informasi lengkap dari setiap publikasi, meliputi:

- Judul, nama penulis, dan afiliasi institusi

- Abstrak penelitian dan referensi
- Tanggal publikasi, tipe publikasi (jurnal/working paper), serta E-ISBN dan volume
- Tombol File untuk mengunduh atau melihat dokumen (jika tersedia publik).

Desain difokuskan pada keterbacaan isi dan penataan informasi yang rapi. Warna latar dibuat netral dengan tipografi yang lembut untuk meningkatkan kenyamanan membaca.

The screenshot shows the homepage of the 'Workingpaper Teknik Industri' website. At the top, there is a navigation bar with icons for Home, Beranda Jurnal&workingpaper, Dosen, Mahasiswa, and Login. Below the navigation bar, there is a banner for 'JURNAL TEKNIK INDUSTRI' featuring a group photo and some text. To the right of the banner, there is a section titled 'Judul jurnal' with details like Published date: dd/mm/yyyy, E-ISBN: xxxxxxxxx, and Volume: xxx. Below this, there is a section titled 'About Jurnal' with a short paragraph of placeholder text. The main content area is divided into sections for 'Artikel' and 'Indekstasi'. The 'Artikel' section lists five articles with titles like 'Judul artikel 1', authors like 'Penulis artikel 1, penulis artikel 2, penulis artikel 3', abstract views like 'Abstract view: xxx', file download links like 'File 102-103', and a small thumbnail. The 'Indekstasi' section lists various indexing services with their logos: Sinta, ProQuest, Dimensions, DOAJ, WorldCat, INDEX COPERNICUS, EBSCOhost, Google Scholar, and ASEAN CITATION INDEX.

Gambar 35. Detail jurnal dan workingpaper

3. Halaman Profil Mahasiswa dan Dosen

Publik dapat mengakses profil penulis untuk melihat karya ilmiah yang telah mereka hasilkan.

- Profil Mahasiswa menampilkan nama lengkap, NIM, program studi, angkatan, serta daftar working paper yang telah dipublikasikan.
- Profil Dosen menampilkan nama, NIP, bidang keahlian, serta daftar publikasi yang pernah dibimbing atau ditulis sendiri.

Fitur ini memperkuat transparansi publikasi ilmiah dan memberikan pengakuan terhadap kontribusi akademik civitas akademika.



NAMA DOSEN
NIP : 2301220211
Bidang Keahlian : Sistem Informasi dan enterprise

| No | Judul karya | Tahun | Detail |
|----|--|-------|--------|
| 1 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 2 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 3 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 4 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 5 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 6 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 7 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2022 | |
| 8 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2025 | |
| 9 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2023 | |
| 10 | Lorem ipsum dolor sit amet, consectetur adipiscing Lorem ipsum dolor sit amet, consectetur adipiscing | 2024 | |

Gambar 36. Profil dosen

4. Fitur Navigasi dan Responsivitas

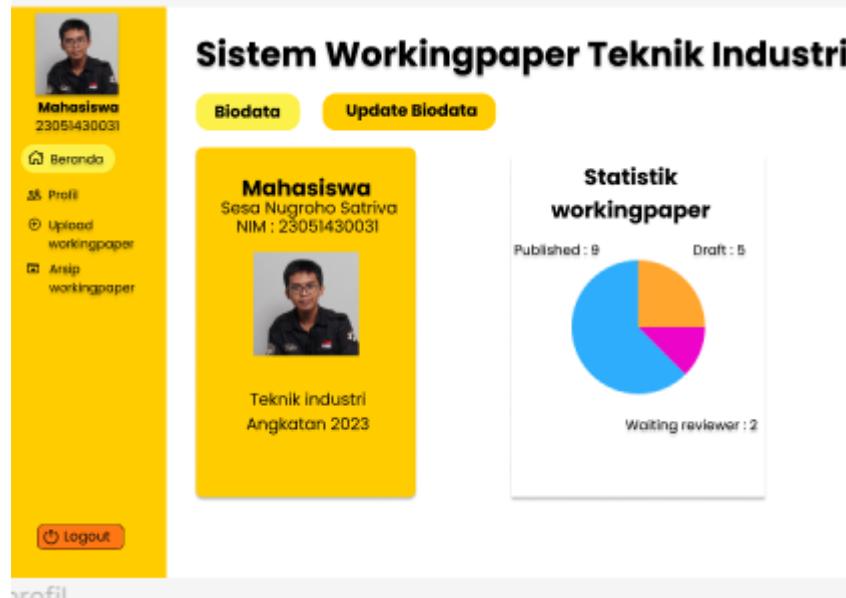
Tiap halaman publik dirancang responsif agar tampil optimal di berbagai perangkat. Navigasi utama ditempatkan di bagian atas (sticky header) untuk memudahkan pengguna berpindah antarhalaman.

Desain interaktif sederhana diterapkan melalui efek hover, transisi lembut, dan penggunaan warna aksen untuk tombol aktif.

5.6 Tampilan Dashboard Pengguna Internal

1. Dashboard Mahasiswa

- Menyediakan menu *Upload Working Paper, Profil, Statistik, dan Arsip*.
- Mahasiswa dapat mengunggah file, menambahkan abstrak, referensi, dan afiliasi.
- Tersedia panel statistik jumlah *draft, waiting review, dan published paper*.



Gambar 37. Dashboard Mahasiswa

2. Dashboard Dosen

- Dosen dapat meninjau dan memberikan *review* terhadap *draft working paper*.
- Tersedia menu *Profil Dosen*, *Riwayat Review*, dan *Daftar Working Paper Mahasiswa*.
- Desain dibuat dengan tampilan formal dan dominasi warna netral untuk menjaga kesan profesional.

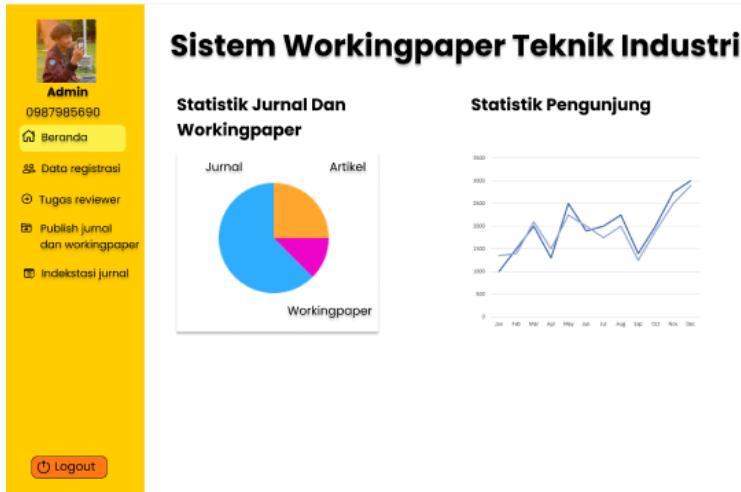


Gambar 38. Dashboard Dosen

3. Dashboard Admin

- Memiliki akses penuh untuk mengelola data pengguna, verifikasi registrasi, serta pengaturan publikasi dan indeksasi jurnal.

- Admin juga dapat melihat statistik sistem dan aktivitas pengguna.



Gambar 39. Dashboard Admin

Ketiga dashboard ini menggunakan gaya visual yang konsisten dengan halaman publik agar transisi penggunaan tetap nyaman bagi semua peran pengguna.

5.7 Prototyping dan Uji Alur UX

Figma digunakan tidak hanya untuk desain statis, tetapi juga untuk membangun prototipe interaktif.

Melalui fitur Prototype Mode, seluruh halaman dihubungkan untuk mensimulasikan:

- Proses login dan registrasi
- Navigasi dari beranda menuju halaman detail jurnal
- Proses unggah file oleh mahasiswa
- Alur review dosen hingga publikasi oleh admin

Simulasi ini memungkinkan pengujian awal terhadap kenyamanan pengguna (UX Testing) sebelum implementasi sebenarnya dilakukan.

Umpulan balik dari pengguna uji digunakan untuk memperbaiki posisi tombol, kontras warna, dan kejelasan ikon sebelum desain dikonversi menjadi kode front-end.

5.8 Kesimpulan Perancangan UI/UX

Perancangan UI/UX menggunakan Figma dengan dukungan plugin Icon8 menghasilkan antarmuka sistem Working Paper Teknik Industri yang modern, terstruktur, dan mudah digunakan.

Tampilan publik memungkinkan masyarakat umum untuk mengakses informasi jurnal dan publikasi secara terbuka, sedangkan dashboard internal memberikan pengalaman kerja yang efisien bagi mahasiswa, dosen, dan admin.

Desain ini menggabungkan estetika dan fungsionalitas melalui prinsip kesederhanaan, konsistensi, serta aksesibilitas. Dengan rancangan UI/UX yang matang, sistem ini siap diimplementasikan pada tahap berikutnya menggunakan framework front-end berbasis HTML, CSS, dan JavaScript.

6. Implementasi Slicing HTML CSS

6.1 Pendahuluan

Tahap implementasi merupakan proses pengubahan hasil rancangan User Interface (UI) dan User Experience (UX) menjadi sistem nyata berbasis web. Dalam proyek ini, implementasi dilakukan menggunakan bahasa HTML (HyperText Markup Language) dan CSS (Cascading Style Sheets) yang merupakan fondasi utama dalam pembuatan halaman web statis.

Seluruh proses implementasi dilakukan secara offline menggunakan XAMPP, yaitu aplikasi server lokal yang menggabungkan Apache (web server), MySQL (basis data), dan PHP (server-side scripting). Folder proyek disimpan pada direktori htdocs agar dapat diakses melalui alamat:

http://localhost/frontier_workingpaper/

C:\xampp\htdocs

Implementasi ini bertujuan untuk mengkonversi hasil desain Figma menjadi tampilan web yang identik secara visual, dengan memastikan setiap elemen : warna, posisi, dan ikon sesuai dengan rancangan awal.

6.2 Teknologi yang Digunakan

Dalam proses implementasi website ini, digunakan beberapa teknologi utama, yaitu:

Tabel 17. Komponen teknologi yang digunakan

| Komponen | Keterangan |
|--|--|
| HTML (HyperText Markup Language) | Bahasa markah standar untuk membuat struktur halaman web. |
| CSS (Cascading Style Sheets) | Bahasa pemformatan untuk mengatur tampilan visual dari elemen HTML. |
| Figma | Alat desain yang digunakan untuk membuat rancangan UI/UX yang kemudian di-slice menjadi HTML & CSS. |
| XAMPP | Aplikasi server lokal yang digunakan untuk menjalankan website secara <i>offline</i> pada komputer pengembang. |
| Browser (Google Chrome/Mozilla Firefox) | Alat untuk menampilkan hasil implementasi website. |

6.3 Penjelasan Bahasa Pemrograman HTML dan CSS

1. HTML (HyperText Markup Language)

HTML merupakan bahasa dasar untuk menyusun struktur halaman web. HTML menggunakan tag untuk menandai setiap elemen seperti teks, gambar, tautan, tabel, atau formulir.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |  <title>Beranda Working Paper</title>
5  </head>
6  <body>
7  |  <h1>Selamat Datang di Sistem Working Paper</h1>
8  |  <p>Website publikasi ilmiah mahasiswa dan dosen Teknik Industri.</p>
9  </body>
10 </html>
```

Gambar 40. Contoh struktur dasar HTML

2. CSS (Cascading Style Sheets)

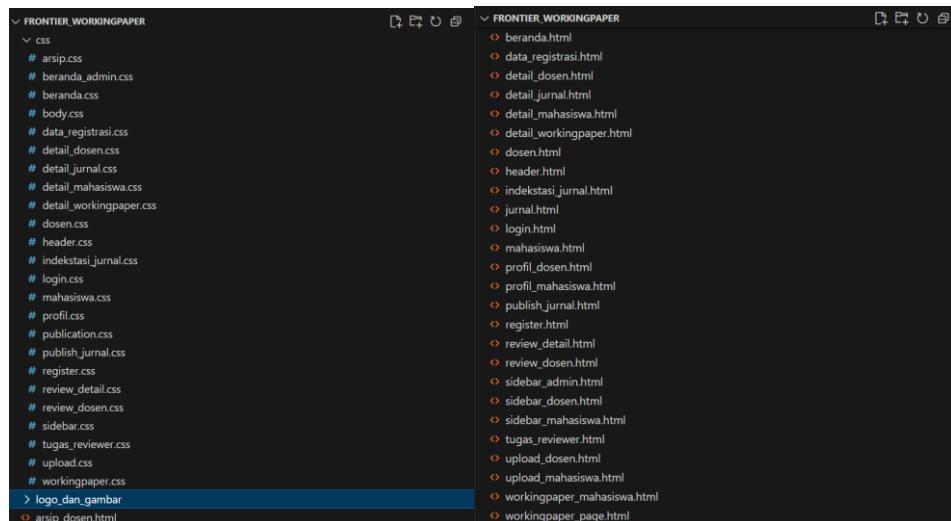
CSS digunakan untuk mengatur tampilan visual halaman HTML seperti warna, ukuran teks, jarak, dan tata letak. CSS dapat ditulis secara *inline*, *internal*, atau *eksternal* melalui file terpisah (.css).

```
1  <style>
2  body {
3  |  background-color: #f8f8f8;
4  |  font-family: 'Poppins', sans-serif;
5  }
6  h1 {
7  |  color: #004AAD;
8  }
9  </style>
```

Gambar 41. Contoh penggunaan CSS

6.4 Struktur Folder dan File Website

Struktur proyek website yang digunakan adalah sebagai berikut:



Gambar 42. Struktur folder dan file website

Struktur ini menggambarkan sistem yang modular dan mudah dikelola, di mana setiap halaman HTML memiliki file CSS pendukung yang mengatur tampilannya secara spesifik.

6.5 Komponen Dasar HTML yang Digunakan

Berikut penjelasan elemen HTML utama yang digunakan dalam proyek ini:

Tabel 18.Komponen HTML yang digunakan

| No | Tag Atribut HTML / | Fungsi / Penjelasan | Contoh Penggunaan dalam Kode |
|----|---------------------|--|-------------------------------------|
| 1 | <!DOCTY PE html> | Mendeklarasikan bahwa dokumen ini menggunakan standar HTML5. | <!DOCTYPE html> |
| 2 | <html> | Elemen pembungkus seluruh dokumen HTML. | <html lang="id"> ... </html> |
| 3 | <head> | Menyimpan metadata seperti judul, CSS, dan <i>favicon</i> . | <head><title>Beranda</title></head> |
| 4 | <meta> | Menentukan pengaturan karakter dan responsivitas halaman. | <meta charset="UTF-8"> |

| | | | |
|----|-----------|--|---|
| 5 | <title> | Menampilkan nama halaman di tab browser. | <title>Frontier Working Paper</title> |
| 6 | <link> | Menghubungkan file eksternal seperti CSS atau ikon. | <link rel="stylesheet" href="css/beranda.css"> |
| 7 | <body> | Berisi seluruh isi halaman web. | <body> ... </body> |
| 8 | <header> | Bagian atas halaman berisi logo dan navigasi utama. | <header>...</header> |
| 9 | <nav> | Menampung daftar tautan menu utama. | <nav>Login</nav> |
| 10 | <main> | Elemen utama berisi konten inti halaman. | <main><section>...</section></main> |
| 11 | <section> | Mengelompokkan konten ke dalam bagian tematik. | <section class="publication-grid">...</section> |
| 12 | <article> | Digunakan untuk menampilkan satu unit konten seperti satu artikel atau satu publikasi. | <article><h3>Judul Paper</h3></article> |
| 13 | <aside> | Menampilkan konten tambahan seperti panel samping atau informasi tambahan. | <aside class="sidebar">Menu Admin</aside> |
| 14 | <div> | Elemen <i>container</i> serbaguna untuk mengelompokkan elemen-elemen HTML. | <div class="card">...</div> |
| 15 | <a> | Menautkan halaman atau dokumen lain menggunakan atribut href. | Beranda |

| | | | |
|----|--------------|---|--|
| 16 | | Menampilkan gambar dari folder asset. | |
| 17 | <p> | Menampilkan teks paragraf. | <p>Selamat datang di website publikasi.</p> |
| 18 | <h1> <h6> | Menentukan tingkat judul dari besar (h1) ke kecil (h6). | <h2>Daftar Working Paper</h2> |
| 19 | <form> | Membuat formulir untuk input pengguna. | <form action="login_process.php">...</form> |
| 20 | <input> | Menyediakan kolom input teks, email, password, atau file. | <input type="email" placeholder="Masukkan email"> |
| 21 | <label> | Memberi keterangan pada kolom input dalam form. | <label for="email">Email:</label> |
| 22 | <button> | Menjalankan aksi seperti login, kirim data, atau unggah file. | <button type="submit">Masuk</button> |
| 23 | <select> | Menyediakan pilihan dropdown. | <select><option>Mahasiswa</option></select> |
| 24 | <option> | Menampilkan pilihan dalam dropdown <select>. | <option value="admin">Admin</option> |
| 25 | <table> | Menampilkan data dalam bentuk tabel. | <table><tr><td>Judul</td></tr></table> |
| 26 | <tr> | Menentukan baris dalam tabel (<i>table row</i>). | <tr><td>Nama</td><td>Sesa Nugroho</td></tr> |
| 27 | <th> | Menentukan <i>header cell</i> atau judul kolom pada tabel. | <th>Nama</th> |
| 28 | <td> | Menentukan <i>data cell</i> atau isi kolom pada tabel. | <td>Teknik Industri</td> |

| | | | |
|----|----------|--|--|
| 29 | | Menyusun daftar tanpa urutan (bullet list). | LoginRegister |
| 30 | | Menyusun daftar dengan urutan angka. | Langkah 1 |
| 31 | | Menentukan item dalam daftar (atau). | Publikasi |
| 32 | | Menandai bagian kecil teks di dalam elemen lain untuk styling. | <p>Oleh: Sesa Nugroho</p> |
| 33 | <iframe> | Menyematkan file eksternal seperti PDF atau halaman web lain. | <iframe src="file.pdf" width="100%"></iframe> |
| 34 | | Memberikan baris baru tanpa membuat paragraf baru. | Nama Teknik Industri |
| 35 | <hr> | Membuat garis pemisah horizontal antar bagian. | <hr> |
| 36 | <style> | Menyisipkan CSS langsung di dalam file HTML. | <style>body {background:#fff;}</style> |
| 37 | <script> | Menyisipkan file JavaScript eksternal (opsional). | <script src="script.js"></script> |
| 38 | <footer> | Tidak digunakan dalam sistem ini, karena bagian bawah halaman diganti dengan <div class="bottom-bar"> untuk fleksibilitas layout. | — |

6.6 Komponen dan Properti CSS yang Digunakan

Berikut penjelasan elemen HTML utama yang digunakan dalam proyek ini:

Tabel 19. Komponen CSS yang digunakan

| No | Properti / Selektor CSS | Fungsi / Penjelasan | Contoh Penggunaan dalam Kode |
|----|-------------------------|--|---|
| 1 | body | Mengatur tampilan global halaman, termasuk warna latar dan jenis font. | body { background:#f9f9f9; font-family:'Poppins'; } |
| 2 | .class | Selektor untuk elemen dengan atribut class. | .card { border-radius:10px; } |
| 3 | #id | Selektor untuk elemen dengan atribut id. | #header { background:#004AAD; } |
| 4 | * | Selektor universal untuk seluruh elemen. | * { box-sizing:border-box; } |
| 5 | color | Mengatur warna teks. | h1 { color:#004AAD; } |
| 6 | background-color | Menentukan warna latar belakang elemen. | .btn { background-color:#00D26A; } |
| 7 | font-family | Mengatur jenis font teks. | p { font-family:'Inter',sans-serif; } |
| 8 | font-size | Menentukan ukuran teks. | .title { font-size:20px; } |
| 9 | font-weight | Menentukan ketebalan teks (bold, normal, light). | h2 { font-weight:600; } |
| 10 | text-align | Menentukan posisi teks (left, center, right). | p { text-align:center; } |
| 11 | margin | Memberi jarak luar elemen. | .card { margin:20px; } |
| 12 | padding | Memberi jarak antara isi dan tepi elemen. | .content { padding:15px; } |
| 13 | border | Membuat garis tepi pada elemen. | .input { border:1px solid #ccc; } |

| | | | |
|----|-----------------------|---|--|
| 14 | border-radius | Membulatkan sudut elemen. | .btn { border-radius:8px; } |
| 15 | box-shadow | Memberikan efek bayangan untuk kedalaman visual. | .card { box-shadow:0 0 10px rgba(0,0,0,0.1); } |
| 16 | width / height | Menentukan ukuran elemen. | img { width:120px; height:auto; } |
| 17 | display | Menentukan cara elemen ditampilkan (block, flex, grid, inline). | .header { display:flex; } |
| 18 | flex | Mengatur ukuran fleksibel pada elemen dalam flexbox. | .nav { flex:1; } |
| 19 | justify-content | Menentukan posisi horizontal dalam flex container. | .header { justify-content:space-between; } |
| 20 | align-items | Menentukan posisi vertikal elemen dalam flex container. | .header { align-items:center; } |
| 21 | gap | Mengatur jarak antar elemen di flex/grid. | .grid { gap:25px; } |
| 22 | grid-template-columns | Menentukan jumlah kolom dalam grid layout. | .grid { grid-template-columns:repeat(3,1fr); } |
| 23 | position | Menentukan posisi elemen relatif, absolut, atau tetap. | .nav { position:fixed; top:0; } |
| 24 | overflow | Mengatur tampilan jika isi melebihi batas elemen. | .table-wrapper { overflow:auto; } |
| 25 | z-index | Menentukan lapisan elemen (mana yang berada di depan). | header { z-index:10; } |

| | | | |
|----|-------------------------|---|---|
| 26 | cursor | Menentukan bentuk kursor saat diarahkan ke elemen. | a:hover { cursor:pointer; } |
| 27 | text-decoration | Mengatur garis pada teks (misal: menghapus garis bawah tautan). | a { text-decoration:none; } |
| 28 | hover | Pseudo-class yang aktif saat elemen disentuh kursor. | .btn:hover { background:#00B35B; } |
| 29 | transition | Menambahkan efek halus pada perubahan gaya. | .btn:hover { transition:0.3s; } |
| 30 | background-image | Menambahkan gambar sebagai latar belakang. | body { background-image:url('bg.png'); } |
| 31 | object-fit | Mengatur cara gambar menyesuaikan ukuran kotak. | img { object-fit:cover; } |
| 32 | opacity | Menentukan tingkat transparansi elemen. | .overlay { opacity:0.8; } |
| 33 | overflow-x / overflow-y | Menentukan scroll horizontal atau vertikal. | .table { overflow-y:scroll; } |
| 34 | box-sizing | Mengatur perhitungan lebar dan tinggi elemen. | * { box-sizing:border-box; } |
| 35 | align-self | Menyesuaikan posisi individu dalam flex container. | .item:last-child { align-self:flex-end; } |

6.7 Proses Slicing dari Figma ke HTML dan CSS

Tahap slicing dilakukan dengan cara mengubah komponen desain Figma menjadi elemen HTML. Setiap bagian desain diidentifikasi, diekspor (gambar/logo), kemudian dikodekan sesuai dengan struktur web.

Langkah-langkah yang dilakukan:

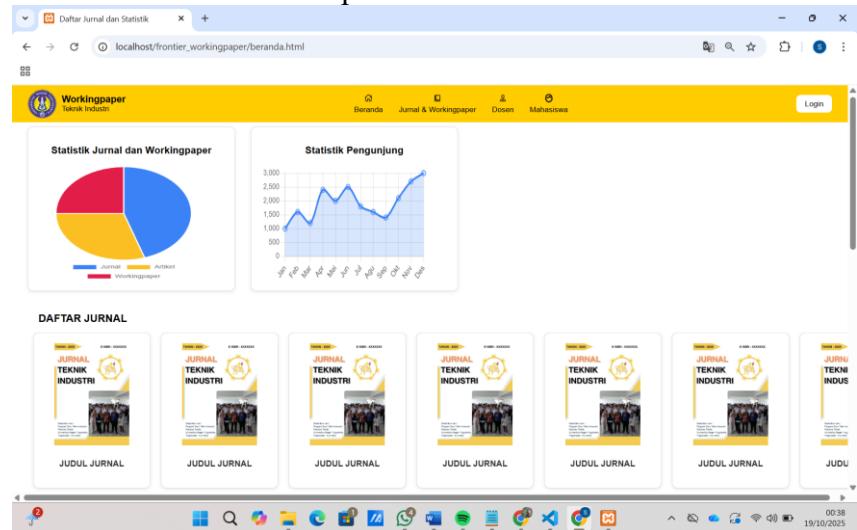
- Mengekspor komponen visual seperti logo dan ikon dari Figma.
- Menyalin nilai warna, ukuran font, dan margin dari design inspector.
- Menyusun layout menggunakan struktur <div> dan flexbox/grid.
- Mengatur gaya di file CSS eksternal agar sesuai dengan desain.

- Menguji tampilan pada browser untuk memastikan kesesuaian.

6.8 Menjalankan Website Menggunakan XAMPP

Langkah-langkah menjalankan sistem:

1. Buka aplikasi XAMPP Control Panel.
2. Aktifkan modul Apache dan MySQL.
3. Salin folder proyek frontier_workingpaper ke:
C:\xampp\htdocs\
4. Jalankan browser dan ketik alamat:
http://localhost/frontier_workingpaper/beranda.html
5. Website akan terbuka dan dapat diakses secara lokal.



Gambar 43. Hasil menjalankan local host

6.9 Kesimpulan

Implementasi website Sistem Working Paper Teknik Industri berhasil mengonversi hasil rancangan Figma ke bentuk nyata menggunakan HTML dan CSS. Struktur folder yang terorganisasi, penggunaan CSS eksternal, serta integrasi di XAMPP localhost menjadikan sistem mudah dijalankan dan dikembangkan.

Dengan hasil ini, tahap implementasi telah memenuhi standar antarmuka profesional, serta siap untuk dilanjutkan ke tahap integrasi fungsional dengan PHP dan database MySQL.

Refrensi

<https://www.uml-diagrams.org/activity-diagrams-reference.html>

https://sparxsystems.com/enterprise_architect_user_guide/17.1/guide_books/tools_ba_uml_activity_diagram.html

<https://arxiv.org/abs/1604.02386>

<https://www.pro.co.id/pengertian-dan-contoh-data-flow-diagram-dfd/>

<https://www.edrawmax.com/templates/1012160/>

<https://www.domainesia.com/berita/pengertian-erd-adalah/>

Lampiran

[https://drive.google.com/drive/folders/1DZvm98DtKMPQofn6c-kMVfgx49YjRnAN?usp=sharing](https://drive.google.com/drive/folders/1DZvm98DtKMPQofn6ckMVfgx49YjRnAN?usp=sharing)

<https://www.figma.com/design/GIgy52Q7BRXX0tAW340vru/Frontiers?t=kqkwm3ZzrXtTkxOi-0>