

**LAPORAN PRAKTIKUM
WEBSITE ‘KOMPETISI/LOMBA MAHASISWA’**

LAPORAN PRAKTIKUM



Nama : Stevani Maria Ayu Rajagukguk

NPM : 230210038

Dosen : Ellbert Hutabri, S.Kom., M.Kom

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM
2025**

KATA PENGANTAR

Puji syukur atas limpahan rahmat dan karunia-NYA sehingga Pedoman Penulisan Tugas Mandiri ini dapat diselesaikan dengan baik walaupun masih terdapat kekurangan namun diharapkan dapat diperbaiki kedepannya.

Tugas Mandiri (TM) disusun menurut kaidah keilmuan dan ditulis berdasarkan kaidah Bahasa Indonesia di bawah pengawasan atau pengarahan dosen pengampu untuk memenuhi kriteria-kriteria kualitas yang telah ditetapkan sesuai keilmuannya masing-masing. Tugas Mandiri dibuat sebagai salah satu persyaratan untuk menyelesaikan suatu mata kuliah di Universitas Putera Batam (UPB). Dalam upaya mendokumentasikan seluruh Tugas Mandiri mahasiswa, diperlukan Pedoman Penulisan Tugas Mandiri yang dapat digunakan di semua fakultas di lingkungan Universitas Putera Batam. Pedoman ini disusun oleh Tim Penyusun Pedoman Penulisan Tugas Mandiri dari LPPM UPB dengan tujuan memberikan tuntunan kepada mahasiswa dan pedoman bagi semua fakultas. Tim Penyusun memberi kesempatan kepada Program Studi/Fakultas untuk membuat petunjuk tambahan mengenai hal-hal yang tidak diatur dalam pedoman ini.

Semoga dengan adanya Panduan Penulisan Tugas Mandiri ini maka penyelesaian Tugas Mandiri bagi setiap mahasiswa UPB dapat lebih lancar.

Batam, 27 June 2025

Ketik nama penyusun.

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
1.5 Ruang Lingkup.....	2
BAB II	3
LANDASAN TEORI.....	3
2.1 Konsep Dasar Rekayasa Perangkat Lunak.....	3
2.2 Unified Modeling Language (UML).....	3
2.2.1 Use Case Diagram (Analisis Fungsionalitas Sistem).....	3
2.2.2 Activity Diagram (Pemodelan Alur Proses)	6
2.2.3 Sequence Diagram (Pemodelan Interaksi Objek).....	7
2.2.4 Class Diagram (Pemodelan Struktur Sistem)	10
2.3 Konsep Sistem Informasi Berbasis Web	13
2.4 Konsep Database.....	13
2.5 Lomba Mahasiswa Online.....	13

DAFTAR GAMBAR

GAMBAR 1. USE CASE DIAGRAM.....	5
GAMBAR 2. ACTIVITY DIAGRAM.....	7
GAMBAR 3. SEQUENCE DIAGRAM	9
GAMBAR 4. CLASS DIAGRAM	12

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi telah berkembang pesat dan memberikan dampak yang signifikan terhadap berbagai aspek kehidupan, termasuk dalam bidang pendidikan. Perguruan tinggi sebagai institusi pendidikan tinggi dituntut untuk mampu mengikuti perkembangan teknologi guna mendukung proses belajar mengajar serta kegiatan non-akademik, seperti perlombaan antar mahasiswa. Lomba atau kompetisi menjadi salah satu sarana penting dalam mengembangkan potensi, kreativitas, dan inovasi mahasiswa.

Namun demikian, penyelenggaraan lomba secara manual masih sering dijumpai di berbagai kampus. Proses pendaftaran peserta, pengumpulan karya, penilaian, hingga pengumuman pemenang seringkali dilakukan secara terpisah, tidak efisien, dan rentan terhadap kesalahan administrasi. Hal ini dapat menghambat kelancaran kegiatan dan menurunkan minat mahasiswa untuk berpartisipasi.

Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah sistem informasi yang dapat mendukung pelaksanaan lomba secara digital, terpusat, dan terintegrasi. Oleh karena itu, dibangunlah **Sistem Informasi Lomba Mahasiswa Berbasis Web** yang dapat digunakan oleh admin, peserta, dan juri dalam satu platform terpadu. Sistem ini memanfaatkan teknologi PHP sebagai bahasa pemrograman dan MySQL sebagai sistem basis data, sehingga bersifat ringan dan dapat dioperasikan melalui jaringan lokal (localhost) maupun internet.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian atau proyek ini adalah:

1. Bagaimana merancang dan membangun sistem informasi lomba mahasiswa berbasis web?
2. Bagaimana sistem dapat memfasilitasi admin dalam mengelola data lomba dan peserta?
3. Bagaimana sistem memungkinkan peserta mengunggah karya secara online?
4. Bagaimana sistem mendukung proses penilaian oleh juri secara terstruktur?

1.3 Tujuan Penelitian

Adapun tujuan dari pembangunan sistem informasi lomba mahasiswa ini adalah:

1. Untuk membangun sistem informasi berbasis web yang dapat memfasilitasi pengelolaan lomba secara digital.
2. Untuk menyediakan fitur unggah karya secara daring oleh peserta lomba.
3. Untuk menyederhanakan proses penilaian oleh juri melalui platform yang terpusat.
4. Untuk meningkatkan efisiensi dan transparansi dalam pelaksanaan lomba mahasiswa.

1.4 Manfaat Penelitian

Manfaat dari pengembangan sistem informasi lomba mahasiswa ini antara lain:

- **Bagi Admin:** Mempermudah pengelolaan lomba, data peserta, dan dokumentasi hasil lomba.
- **Bagi Peserta:** Memberikan kemudahan dalam proses pendaftaran dan pengumpulan karya..
- **Bagi Juri:** Menyediakan antarmuka penilaian yang efisien dan terintegrasi.
- **Bagi Institusi:** Meningkatkan kualitas pelayanan dan dokumentasi kegiatan mahasiswa secara digital

1.5 Ruang Lingkup

Penelitian atau pengembangan sistem ini difokuskan pada:

- Sistem berbasis web yang dibangun dengan PHP dan MySQL.
- Fungsi utama meliputi: pengelolaan data lomba, unggah karya oleh peserta, dan penilaian oleh juri.
- Sistem berjalan pada lingkungan lokal (localhost) menggunakan XAMPP.
- Tidak mencakup sistem pembayaran, integrasi API eksternal, atau pengumuman otomatis melalui media sosial.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (RPL) adalah disiplin ilmu yang sistematis, terukur, dan terorganisir yang digunakan untuk mengembangkan, mengoperasikan, dan memelihara perangkat lunak. Tujuan utama RPL adalah menghasilkan perangkat lunak berkualitas tinggi, efisien, dan dapat diandalkan yang memenuhi kebutuhan pengguna dalam batas waktu dan anggaran yang telah ditentukan. Dalam konteks pengembangan sistem informasi, RPL menyediakan kerangka kerja metodologis untuk memastikan bahwa proses pengembangan perangkat lunak dilakukan secara terstruktur dan profesional.

2.2 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan dalam rekayasa perangkat lunak untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan artefak dari sistem yang berorientasi objek. UML menyediakan serangkaian diagram yang komprehensif, masing-masing dengan fokus pada aspek tertentu dari sistem. Penggunaan UML sangat vital untuk meningkatkan komunikasi antar tim pengembangan, mempermudah pemahaman sistem yang kompleks, dan memastikan konsistensi dalam desain.

2.2.1 Use Case Diagram (Analisis Fungsionalitas Sistem)

Use Case Diagram memodelkan fungsionalitas sistem dari perspektif pengguna atau entitas eksternal (aktor). Diagram ini berfokus pada "apa yang dilakukan sistem" tanpa merinci "bagaimana sistem melakukannya".

Berdasarkan perancangan **Website Kompetisi/Lomba Mahasiswa**, Use Case Diagram yang disajikan (Lihat Gambar 2.1: Use Case Diagram Website Kompetisi/Lomba Mahasiswa) mengidentifikasi aktor dan fungsionalitas utama sebagai berikut:

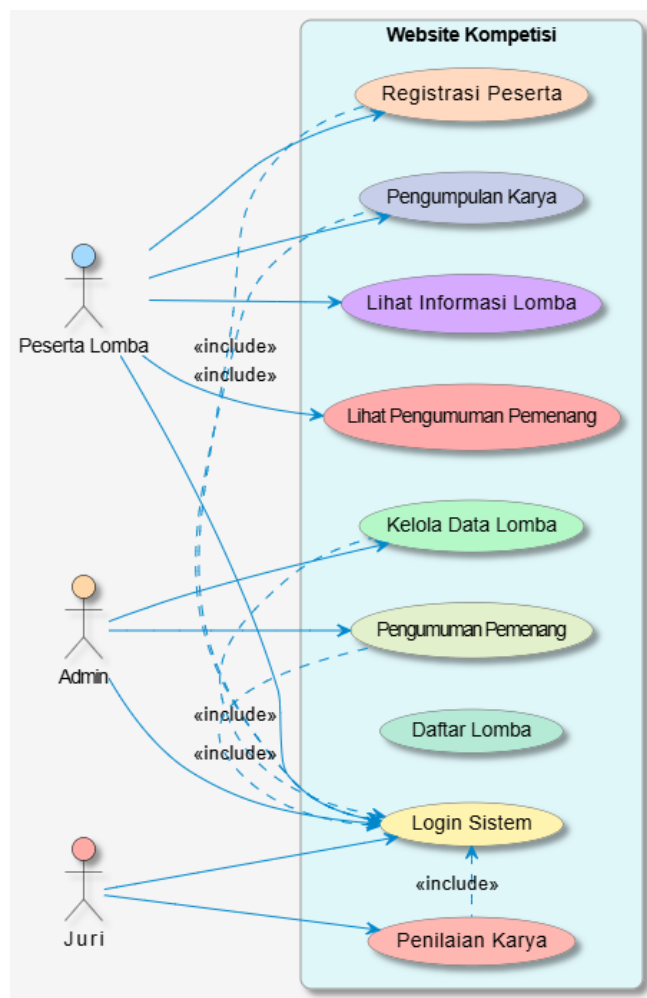
- **Aktor:**

- **Peserta Lomba:** Mahasiswa yang berpartisipasi dalam kompetisi. Bertanggung jawab untuk melakukan registrasi, pengumpulan karya, dan melihat informasi terkait lomba serta pengumuman.
- **Admin:** Pengelola sistem yang memiliki hak akses untuk mengatur dan memanajemen seluruh data dan proses lomba.
- **Juri:** Pihak yang bertugas untuk melakukan penilaian terhadap karya-karya yang telah dikumpulkan oleh peserta.

- **Use Case Utama (Fungsionalitas Sistem):**

- **Login Sistem:** Merupakan fungsionalitas dasar yang **di-include** (disertakan) dalam hampir semua use case lain yang membutuhkan otentikasi. Ini memungkinkan Peserta Lomba, Admin, dan Juri untuk masuk ke sistem.
- **Registrasi Peserta:** Fungsionalitas bagi Peserta Lomba untuk mendaftarkan diri ke sistem dan/atau lomba tertentu.
- **Pengumpulan Karya:** Fungsionalitas bagi Peserta Lomba untuk mengunggah dan mengirimkan hasil karya mereka sesuai dengan persyaratan lomba.
- **Lihat Informasi Lomba:** Memungkinkan Peserta Lomba untuk mengakses dan membaca detail serta peraturan lomba yang tersedia.
- **Lihat Pengumuman Pemenang:** Fungsionalitas bagi Peserta Lomba untuk melihat daftar pemenang yang telah diumumkan.
- **Kelola Data Lomba:** Fungsionalitas utama bagi Admin untuk menambah, mengubah, atau menghapus data lomba, termasuk mengatur jadwal dan detail lainnya.
- **Pengumuman Pemenang:** Fungsionalitas bagi Admin untuk secara resmi mengumumkan hasil dan daftar pemenang lomba.

- **Daftar Lomba:** Ini dapat diartikan sebagai bagian dari fungsionalitas Admin dalam mengelola ketersediaan dan detail lomba di dalam sistem.
- **Penilaian Karya:** Fungsionalitas bagi Juri untuk mengakses karya peserta dan memberikan nilai serta komentar sesuai kriteria yang ditentukan.
- **Hubungan Include (<<include>>):** Diagram ini secara eksplisit menunjukkan bahwa Login Sistem adalah use case yang wajib diakses sebelum melakukan sebagian besar fungsionalitas lain oleh aktor terkait (Registrasi Peserta, Pengumpulan Karya, Lihat Informasi Lomba, Lihat Pengumuman Pemenang, Kelola Data Lomba, Pengumuman Pemenang, Daftar Lomba, Penilaian Karya). Hal ini menekankan pentingnya otentikasi pengguna dalam sistem.



gambar 1. Use Case Diagram

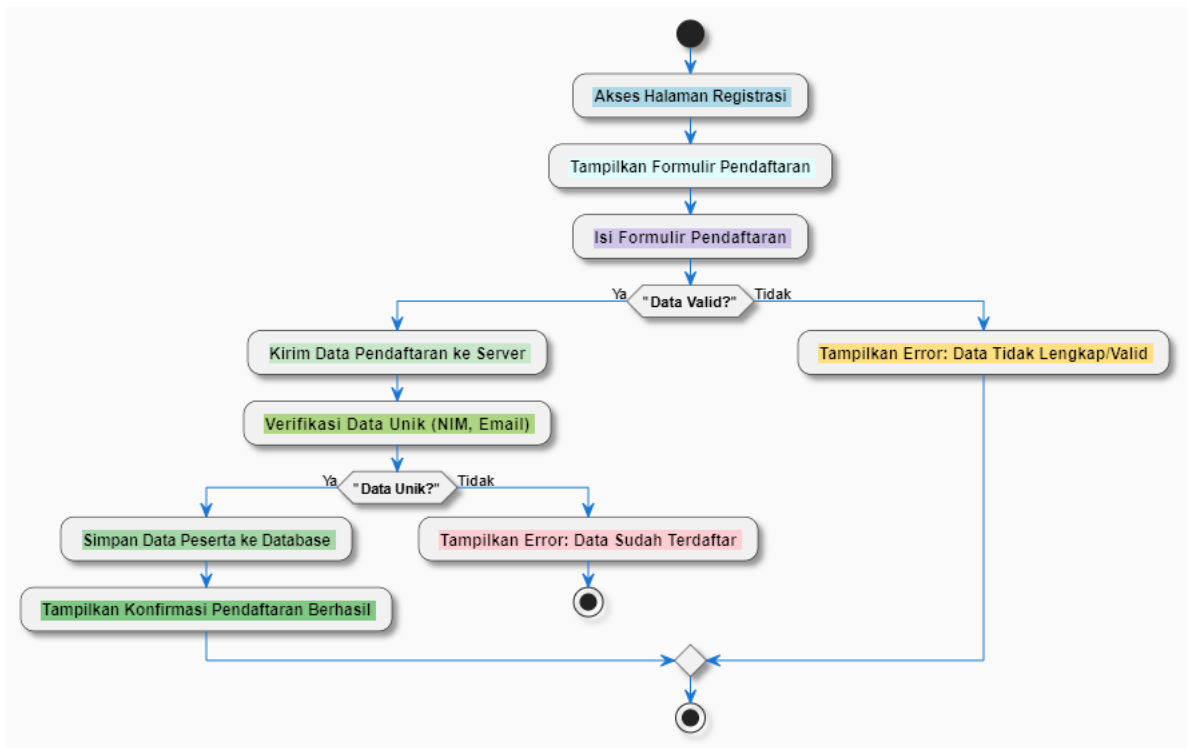
2.2.2 Activity Diagram (Pemodelan Alur Proses)

Activity Diagram digunakan untuk memodelkan alur kerja atau proses bisnis secara langkah demi langkah, termasuk percabangan dan aktivitas paralel.

Berdasarkan perancangan **Website Kompetisi/Lomba Mahasiswa**, Activity Diagram untuk proses "Registrasi Peserta" (Lihat Gambar 2.2: Activity Diagram Registrasi Peserta) menjelaskan alur sebagai berikut:

- **Node Awal (Start):** Proses dimulai dari sini.
- **Aktivitas Berurutan:**
 1. **Akses Halaman Registrasi:** Pengguna memulai dengan mengakses halaman pendaftaran.
 2. **Tampilkan Formulir Pendaftaran:** Sistem merespons dengan menampilkan formulir yang harus diisi.
 3. **Isi Formulir Pendaftaran:** Pengguna memasukkan data-data yang diminta pada formulir.
- **Node Keputusan "Data Valid?":** Setelah pengisian formulir, sistem melakukan validasi awal.
 - **Jalur "Ya":** Jika data yang diisi valid (sesuai format, tidak kosong), alur berlanjut ke:
 - **Kirim Data Pendaftaran ke Server:** Data dikirimkan ke server untuk diproses lebih lanjut.
 - **Verifikasi Data Unik (NIM, Email):** Server memeriksa apakah Nomor Induk Mahasiswa (NIM) atau alamat email sudah terdaftar dalam sistem.
 - **Node Keputusan "Data Unik?":**
 - **Jalur "Ya":** Jika NIM dan email belum terdaftar (unik), alur berlanjut ke:
 - **Simpan Data Peserta ke Database:** Informasi peserta baru disimpan ke basis data sistem.
 - **Tampilkan Konfirmasi Pendaftaran Berhasil:** Peserta menerima notifikasi bahwa pendaftaran sukses.
 - **Jalur "Tidak":** Jika NIM atau email sudah terdaftar, alur akan menampilkan:
 - **Tampilkan Error: Data Sudah Terdaftar:** Peserta diberitahu bahwa data mereka sudah ada. Proses berhenti setelah ini.

- **Jalur "Tidak":** Jika data awal yang diisi tidak valid (misalnya, format salah, ada kolom kosong), alur akan menampilkan:
- **Tampilkan Error: Data Tidak Lengkap/Valid:** Peserta menerima pesan kesalahan. Proses berhenti setelah ini.
- **Node Akhir (End):** Menunjukkan titik akhir dari alur proses, baik sukses maupun mengalami kesalahan.



gambar 2.Activity Diagram

2.2.3 Sequence Diagram (Pemodelan Interaksi Objek)

Sequence Diagram memodelkan interaksi antar objek atau komponen dalam sistem secara kronologis, menunjukkan urutan pesan yang dipertukarkan untuk menyelesaikan suatu skenario.

Berdasarkan perancangan **Website Kompetisi/Lomba Mahasiswa**, Sequence Diagram untuk skenario "Pengumpulan Karya" (Lihat Gambar 2.3: Sequence Diagram Pengumpulan Karya) merinci interaksi sebagai berikut:

- **Peserta (Aktor):** Menginisiasi interaksi.
- **Halaman Pengumpulan Karya (UI):** Antarmuka pengguna yang diakses oleh peserta.
- **Kontroler Pengumpulan:** Komponen logika bisnis di sisi server yang menangani permintaan pengumpulan karya.
- **Layanan Penyimpanan File:** Modul atau service yang bertanggung jawab menyimpan file karya ke tempat penyimpanan fisik (misalnya, cloud storage, server filesystem).
- **Database Sistem (DB):** Basis data tempat metadata karya disimpan.

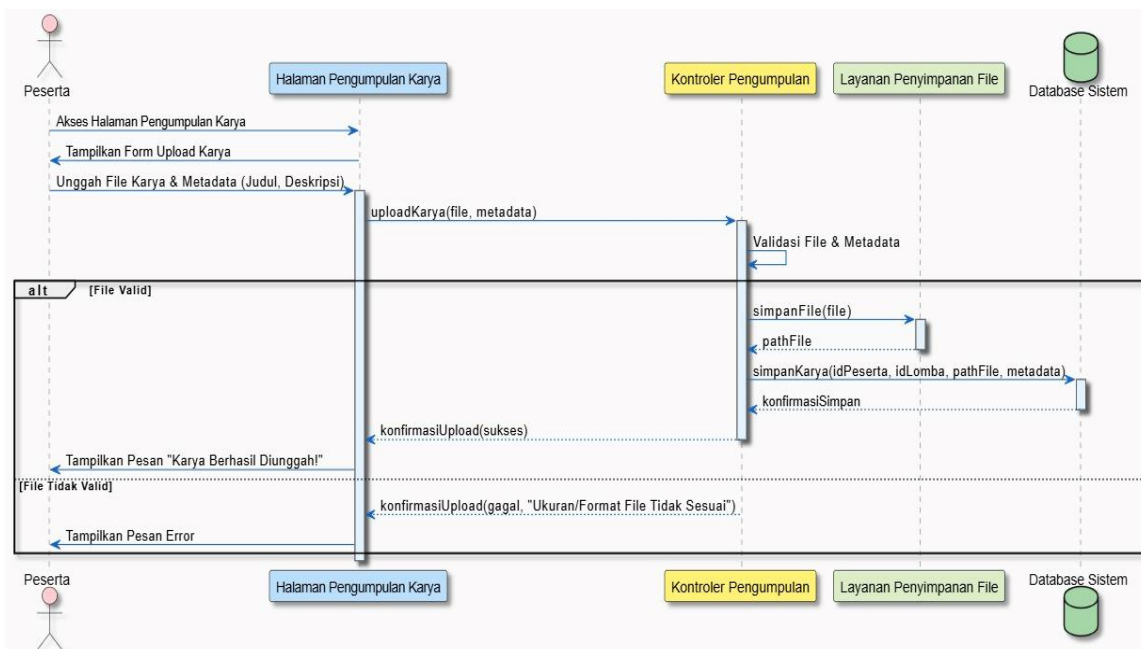
Urutan Interaksi:

1. **Akses Halaman Pengumpulan Karya:** Peserta mengirim permintaan untuk mengakses halaman pengumpulan karya ke Halaman Pengumpulan Karya.
2. **Tampilkan Form Upload Karya:** Halaman Pengumpulan Karya merespons dengan menampilkan formulir upload kepada Peserta.
3. **Unggah File & Metadata (Judul, Deskripsi):** Peserta melakukan upload file dan menyertakan metadata (judul, deskripsi) ke Halaman Pengumpulan Karya.
4. **uploadKarya(file, metadata):** Halaman Pengumpulan Karya meneruskan file dan metadata ke Kontroler Pengumpulan.
5. **Validasi File & Metadata:** Kontroler Pengumpulan secara internal melakukan validasi terhadap file (misalnya, format, ukuran) dan metadata yang diterima.
6. **Alternatif (Alt) - File Valid:**
 - Jika file dan metadata valid:
 - **simpanFile(file):** Kontroler Pengumpulan mengirim permintaan penyimpanan file ke Layanan Penyimpanan File.
 - **pathFile:** Layanan Penyimpanan File mengembalikan lokasi (path) di mana file disimpan kepada Kontroler Pengumpulan.

- **simpanKarya(idPeserta, idLomba, pathFile, metadata):** Kontroler Pengumpulan kemudian menyimpan metadata karya, termasuk pathFile yang baru diterima, ke Database Sistem.
- **konfirmasiSimpan:** Database Sistem mengkonfirmasi bahwa data telah berhasil disimpan.
- **konfirmasiUpload(sukses):** Kontroler Pengumpulan mengirimkan status sukses kembali ke Halaman Pengumpulan Karya.
- **Tampilkan Pesan "Karya Berhasil Diunggah!":** Halaman Pengumpulan Karya menampilkan pesan sukses kepada Peserta.

7. Alternatif (Alt) - File Tidak Valid:

- Jika validasi gagal (misalnya, ukuran atau format file tidak sesuai):
 - **konfirmasiUpload(gagal, "Ukuran/Format File Tidak Sesuai"):** Kontroler Pengumpulan mengirimkan status gagal bersama pesan error ke Halaman Pengumpulan Karya.
 - **Tampilkan Pesan Error:** Halaman Pengumpulan Karya menampilkan pesan kesalahan kepada Peserta.



gambar 3.Sequence Diagram

2.2.4 Class Diagram (Pemodelan Struktur Sistem)

Class Diagram memodelkan struktur statis sistem dengan menunjukkan kelas-kelas, atribut, metode, dan hubungan antar kelas. Ini adalah fondasi untuk desain basis data dan implementasi kode.

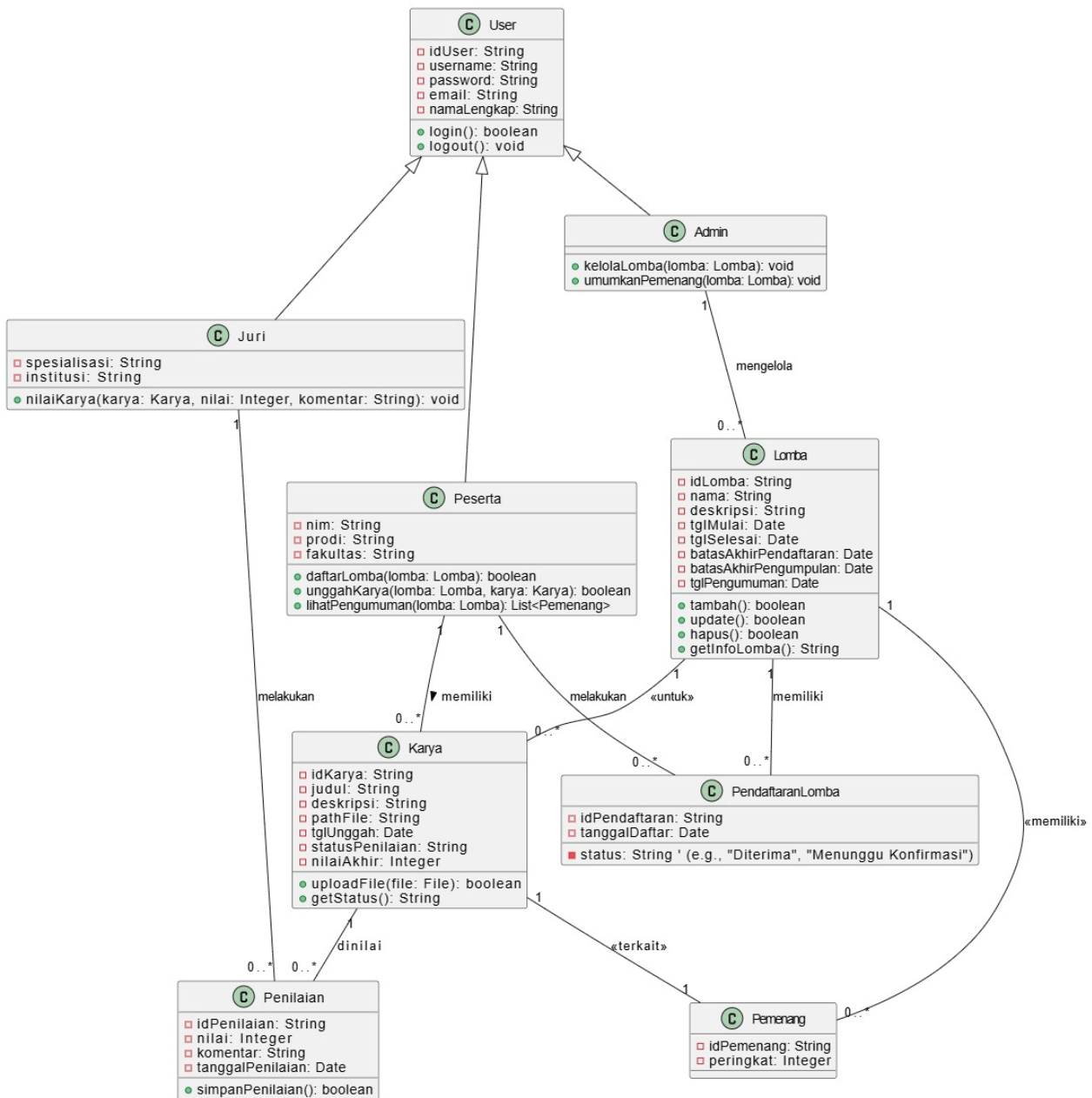
Berdasarkan perancangan **Website Kompetisi/Lomba Mahasiswa**, Class Diagram yang disajikan (Lihat Gambar 2.4: Class Diagram Website Kompetisi/Lomba Mahasiswa) mengidentifikasi entitas dan hubungan sebagai berikut:

- **Kelas Utama:**

- **User:** Kelas dasar yang merepresentasikan semua pengguna sistem. Atributnya meliputi -idUser, -username, -password, -email, dan -namaLengkap. Memiliki metode umum +login() dan +logout().
- **Admin:** Merupakan spesialisasi dari User (Admin extends User). Memiliki metode khusus +kelolaLomba(lomba: Lomba) dan +umumkanPemenang(lomba: Lomba).
- **Juri:** Merupakan spesialisasi dari User (Juri extends User). Memiliki atribut -spesialisasi dan -institusi. Memiliki metode +nilaiKarya(karya: Karya, nilai: Integer, komentar: String).
- **Peserta:** Merupakan spesialisasi dari User (Peserta extends User). Memiliki atribut khusus -nim, -prodi, dan -fakultas. Memiliki metode +daftarLomba(lomba: Lomba), +unggahKarya(lomba: Lomba, karya: Karya), dan +lihatPengumuman().
- **Lomba:** Merepresentasikan detail sebuah kompetisi. Atributnya mencakup -idLomba, -nama, -deskripsi, serta tanggal-tanggal penting seperti -tglMulai, -tglSelesai, -batasAkhirPendaftaran, -batasAkhirPengumpulan, dan -tglPengumuman. Memiliki metode +tambah(), +update(), +hapus(), dan +getInfoLomba().
- **Karya:** Merepresentasikan karya yang diunggah oleh peserta. Atributnya: -idKarya, -judul, -deskripsi, -pathFile (lokasi penyimpanan), -tglUnggah, -statusPenilaian, dan -nilaiAkhir. Metode: +uploadFile() dan +getStatus().
- **PendaftaranLomba:** Merupakan kelas asosiasi yang menghubungkan Peserta dengan Lomba untuk mencatat detail pendaftaran (-idPendaftaran, -tanggalDaftar, -status).

- **Penilaian:** Merupakan kelas asosiasi yang menghubungkan Juri dengan Karya untuk mencatat hasil penilaian (-idPenilaian, -nilai, -komentar, -tanggalPenilaian). Memiliki metode +simpanPenilaian().
- **Pemenang:** Kelas yang merepresentasikan entitas pemenang. Atributnya -idPemenang dan -peringkat.
- **Hubungan Antar Kelas:**
 - **Generalisasi (Inheritance):** Ditunjukkan dengan panah berongga dari subclass ke superclass. Admin, Juri, dan Peserta adalah subclass dari User, mewarisi karakteristik dasar pengguna.
 - **Asosiasi:** Ditunjukkan dengan garis lurus yang menghubungkan kelas-kelas, seringkali dengan label dan multiplicitas (kardinalitas) di kedua ujungnya:
 - Peserta (1) -- "melakukan" -- (0..*) PendaftaranLomba: Satu peserta dapat melakukan nol atau lebih pendaftaran.
 - Lomba (1) -- "memiliki" -- (0..*) PendaftaranLomba: Satu lomba dapat memiliki nol atau lebih pendaftaran.
 - Peserta (1) -- "memiliki" -- (0..*) Karya: Satu peserta dapat memiliki nol atau lebih karya.
 - Lomba (1) -- "<<untuk>>" -- (0..*) Karya: Satu lomba dapat memiliki nol atau lebih karya yang terkait dengannya.
 - Juri (1) -- "melakukan" -- (0..*) Penilaian: Satu juri dapat melakukan nol atau lebih penilaian.
 - Karya (1) -- "dinilai" -- (0..*) Penilaian: Satu karya dapat dinilai oleh nol atau lebih penilaian (dari juri berbeda).
 - Admin (1) -- "mengelola" -- (0..*) Lomba: Satu admin dapat mengelola nol atau lebih lomba.
 - Lomba (1) -- "<<memiliki>>" -- (0..*) Pemenang: Satu lomba dapat memiliki nol atau lebih pemenang.

- Karya (1) -- "terkait" -- (1) Pemenang: Satu karya dapat terkait dengan tepat satu pemenang (jika karya tersebut adalah pemenang).



gambar 4. Class Diagram

2.3 Konsep Sistem Informasi Berbasis Web

Sistem informasi berbasis web adalah aplikasi perangkat lunak yang diakses melalui peramban web (web browser) dan beroperasi menggunakan protokol komunikasi internet. Sistem ini umumnya mengadopsi arsitektur client-server, di mana klien (web browser) mengirimkan permintaan ke server web, yang kemudian berinteraksi dengan server aplikasi dan database untuk memproses data dan mengembalikan respons yang ditampilkan di browser. Keunggulan utama sistem berbasis web meliputi jangkauan akses yang luas (dari mana saja dengan koneksi internet), kemudahan pembaruan dan pemeliharaan (cukup dilakukan di server), serta biaya distribusi yang rendah.

2.4 Konsep Database

Database, atau basis data, adalah kumpulan data terorganisir yang disimpan secara elektronik dan terstruktur sehingga mudah untuk diakses, dikelola, dan diperbarui. Dalam konteks sistem informasi, database berfungsi sebagai gudang data utama yang mendukung seluruh operasional aplikasi. Sistem Manajemen Database (DBMS) adalah perangkat lunak yang memungkinkan pengguna untuk berinteraksi dengan database, seperti MySQL, PostgreSQL, atau SQL Server. Konsep-konsep dasar dalam database meliputi tabel (entitas), kolom (atribut), baris (record), kunci primer (pengidentifikasi unik), kunci asing (penghubung antar tabel), dan normalisasi (proses desain untuk mengurangi redundansi data).

2.5 Lomba Mahasiswa Online

Sistem lomba mahasiswa online merupakan platform digital yang dirancang untuk mengelola dan memfasilitasi seluruh siklus hidup sebuah kompetisi atau lomba yang melibatkan mahasiswa, mulai dari pendaftaran hingga pengumuman pemenang. Tujuan dari sistem ini adalah untuk mengotomatisasi proses manual, meningkatkan efisiensi administrasi, memperluas jangkauan partisipasi, dan memberikan transparansi dalam setiap tahapan lomba. Fitur-fitur yang terintegrasi dalam sistem ini, sebagaimana dimodelkan dalam diagram UML, mencakup manajemen pengguna (admin, juri, peserta), pengelolaan data lomba, proses pendaftaran online, modul pengumpulan karya digital, fasilitas penilaian terpusat untuk juri, serta publikasi pengumuman hasil lomba.