

Perancangan Database Inventaris Laboratorium Menggunakan Semantic Modeling Dengan Pendekatan Top Down

Hendra Kurniawan
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
hendrakurniawan@amikom.ac.id

Abstract— Perancangan database yang tepat merupakan salah satu fondasi penting dalam membangun sistem informasi yang handal dan efisien, terutama dalam pengelolaan data inventaris laboratorium yang memiliki banyak entitas dan relasi yang kompleks. Proses pencatatan inventaris laboratorium di Universitas Amikom Yogyakarta masih dilakukan secara manual, sehingga rentan terhadap human error, duplikasi data, dan aksesibilitas data yang terbatas. Penelitian ini bertujuan untuk merancang database inventaris laboratorium menggunakan semantic modeling dengan pendekatan top down. Pendekatan ini dimulai dengan menganalisis kebutuhan sistem secara menyeluruh dan mengidentifikasi entitas serta hubungan antar entitasnya berdasarkan pemahaman terhadap struktur dan aktivitas nyata pengelolaan inventaris di laboratorium. Model konseptual yang dihasilkan kemudian ditransformasi ke dalam model logis dan fisik untuk membentuk struktur database yang terintegrasi. Hasil dari penelitian ini adalah rancangan database yang terstruktur, konsisten, dan dapat digunakan sebagai dasar pengembangan sistem informasi inventaris laboratorium yang adaptif, scalable, dan mudah dalam pengelolaan data inventaris di kemudian hari. Diharapkan database ini mampu meningkatkan efisiensi pengelolaan data, mempercepat proses pencatatan dan pelacakan aset, serta menunjang pengambilan keputusan manajerial berdasarkan data yang lebih akurat.

Kata kunci — database, inventaris, laboratorium, semantic modeling, top down

I. PENDAHULUAN

Laboratorium/studio adalah unsur penunjang yang harus dimiliki untuk penyelenggaraan perguruan tinggi di Indonesia [1]. Universitas Amikom Yogyakarta merupakan perguruan tinggi yang telah memiliki 16 laboratorium dan 5 studio. Fasilitas ini ditujukan untuk mendukung pembelajaran interaktif dan inovatif bagi mahasiswa, serta sebagai ruang untuk pelaksanaan Tri Dharma Perguruan Tinggi yang meliputi kegiatan pendidikan, penelitian, dan pengabdian kepada masyarakat. Akan tetapi, pengelolaan barang inventaris atau aset dalam laboratorium/studio tersebut masih dilakukan secara manual dengan mencatat data pada lembaran formulir fisik, sehingga rentan terhadap timbulnya masalah, seperti *human error*, duplikasi data, dan aksesibilitas data yang terbatas.

Pada tahun 2019 sub bagian laboratorium Universitas Amikom Yogyakarta merencanakan untuk membangun sistem informasi inventaris laboratorium. Sistem ini bertujuan untuk menggantikan metode pencatatan manual dengan teknologi berbasis digital yang terintegrasi. Salah satu tahap untuk memulainya adalah merancang database, yang berfungsi sebagai tempat penyimpanan, pengelolaan, dan pengolahan data yang digunakan dalam sistem. Database yang dirancang secara baik dapat meningkatkan efisiensi dan kualitas dalam pengembangan *software* [2], serta dapat menjamin keamanan data inventaris laboratorium.

Beberapa masalah yang sering muncul dalam perancangan database meliputi ketidaklengkapan data, redundansi data [3], inkonsistensi data [4], dan sulitnya penyusunan relasi antar entitas yang sesuai kebutuhan. Solusi atas masalah tersebut dapat menggunakan pemodelan semantik, yang berfokus pada pemahaman mendalam terhadap makna data dan hubungan antar entitas untuk menghasilkan desain database yang optimal [5]. Pemodelan ini dimulai dengan mengidentifikasi kebutuhan bisnis hingga menghasilkan model konseptual yang merepresentasikan entitas, atribut, dan relasinya secara logis. Melalui pemodelan ini dapat dipastikan bahwa setiap entitas memiliki makna yang jelas dan relasi antar entitasnya dapat dirancang secara logis sebelum implementasi teknis dilakukan.

Pada penelitian ini akan mengkombinasikan pemodelan semantik (*semantic modeling*) dengan pendekatan *top down* untuk merancang database inventaris laboratorium. Pendekatan *top down* dipilih karena memberikan struktur yang lebih sistematis, mulai dari identifikasi kebutuhan bisnis hingga menghasilkan entitas, atribut, dan relasi antar tabel. Hal ini tentunya dapat membantu meminimalisir kesalahan dalam perancangan database, terutama bila dibandingkan dengan pendekatan *bottom up* [6]. Penelitian ini diharapkan dapat menghasilkan rancangan database yang tidak hanya lebih terstruktur, tetapi juga mampu meningkatkan efisiensi dalam pencatatan, pengelolaan, dan aksesibilitas data inventaris laboratorium.

II. TINJAUAN PUSTAKA

Beberapa penelitian menunjukkan bahwa pemodelan semantik dapat membantu menyelaraskan antara kebutuhan fungsional sistem dengan implementasi teknis database dan memastikan bahwa struktur yang dirancang mendukung efisiensi pengelolaan data dalam jangka panjang. Penelitian oleh Krakowiak [7] menyajikan penerapan kerangka SERM (*Structured Entity Relationship Model*) untuk mengoptimalkan pemodelan entitas. Kerangka ini merupakan bagian dari ERM (*Entity Relationship Model*) klasik yang menggunakan tata letak hierarkis untuk membangun struktur data semantik, mengidentifikasi relasi, dan memastikan integritas data.

Penelitian ini menyimpulkan bahwa kerangka SERM mampu mendukung analisis dan perancangan sistem informasi, khususnya dalam studi kasus Sistem Layanan Pasien.

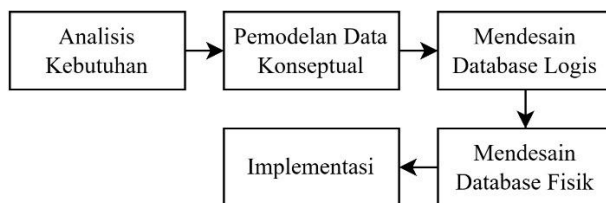
Penelitian oleh Cohen dan Gil [8] menghasilkan model hubungan entitas (*Entity-Relationship Model*) untuk menyelaraskan dan mengintegrasikan informasi tentang pengelolaan limbah dan sumber daya di wilayah kota. Pengumpulan data dilakukan melalui wawancara semi terstruktur dengan pejabat pemerintah, pengembang aplikasi, dan akademisi untuk mendapatkan wawasan dan memvalidasi model. Penelitian ini mengilustrasikan penerjemahan *Entity-Relationship Model* ke dalam skema database relasional dan membuat contoh kasus pengelolaan limbah dan simbiosis industri di Buenos Aires (ARG) dan Helsingborg (SWE).

Penelitian oleh Paneque, dkk [9] menghasilkan kerangka kerja berbasis ontologi untuk merancang database berbasis grafik yang aman yang didukung oleh kerangka kerja TITAN. Kerangka ini dirancang untuk mengintegrasikan aspek keamanan secara otomatis dalam pengelolaan database, sehingga memudahkan penerapan kebijakan yang terkait dengan keamanan. Selanjutnya, hasil penelitian diimplementasikan menggunakan database NoSQL yang mendukung fleksibilitas dan skalabilitasnya dalam menangani data besar dan kemampuannya dalam memenuhi persyaratan keamanan data dan informasi.

Penelitian oleh Javed dan Lin [10] menghasilkan sebuah pendekatan otomatis yang diberi nama iMER yang dirancang untuk mengekstraksi hubungan antar entitas dan membangun model proses bisnis dengan memanfaatkan teknik pemrosesan bahasa alami (*natural language processing*). Beberapa langkah utama dalam penelitian ini meliputi proses identifikasi dan ekstraksi entitas serta atribut data yang ada di dalam dokumen kebutuhan, penentuan kardinalitas untuk menggambarkan keterkaitan antara entitas, serta pembuatan model proses bisnis yang menggambarkan alur kerja berdasarkan informasi yang telah diolah.

III. METODE PENELITIAN

Metode dalam penelitian ini mengadopsi metode *database life cycle* untuk perancangan database inventaris laboratorium menggunakan *semantic modeling* dengan pendekatan *top down*. Tahapan pada metode penelitian ini ditunjukkan pada gambar 1.



Gambar 1 Tahapan Penelitian

1) Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan untuk memahami kebutuhan pengguna dan sistem secara menyeluruh. Pada tahap ini dilakukan pengumpulan data dan informasi melalui wawancara, observasi, dan analisis dokumen terkait pengelolaan inventaris laboratorium. Hasil dari tahap ini adalah spesifikasi kebutuhan data yang akan menjadi dasar dalam perancangan database.

2) Pemodelan Data Konseptual

Tahap pemodelan data konseptual berfokus pada pembuatan model konseptual database berdasarkan hasil analisis kebutuhan. Pendekatan *Semantic Modeling* digunakan untuk menggambarkan entitas utama, atribut, serta hubungan antar entitas dalam sistem. *Entity Relationship Diagram* (ERD) dibuat untuk memberikan representasi visual terkait pengorganisasian data. Pada tahap ini menghasilkan entitas berserta hubungan semantik antar datanya, sehingga dapat dipahami keterkaitan logis dari setiap elemen informasinya.

3) Mendesain Database Logis

Mendesain database logis merupakan proses mengubah model konseptual menjadi skema database relasional yang lebih terstruktur. Pada tahap ini setiap entitas dikonversi menjadi tabel dalam database, setiap atribut ditentukan tipe datanya, serta hubungan antar tabel didefinisikan dengan *primary key* dan *foreign key* untuk menjaga integritas data. Proses normalisasi juga diterapkan untuk menghindari redundansi dan memastikan efisiensi penyimpanan data.

4) Mendesain Database Fisik

Mendesain database fisik merupakan proses mengkonversi model logis ke bentuk yang dapat diimplementasikan dalam *Database Management System* (DBMS) seperti MySQL. Pada tahap ini, skema database dirancang dengan lebih rinci menggunakan bahasa SQL (*Structured Query Language*) yang mencakup pembuatan struktur tabel, penentuan tipe data untuk setiap atribut, serta pembentukan relasi antar tabel melalui *primary key* dan *foreign key*. Selain itu, proses ini juga mencakup pembuatan aturan validasi seperti *constraint* (NOT NULL, UNIQUE, atau CHECK) untuk memastikan kualitas datanya. Dengan demikian, desain database fisik tidak hanya memastikan bahwa database dapat menangani data secara efisien, tetapi juga menjaga keamanan, konsistensi, dan keandalan dalam operasional sehari-hari.

5) Implementasi

Implementasi merupakan proses penerapan database sebagai tempat untuk penyimpanan data inventaris laboratorium. Database yang telah dibuat akan diuji dengan data nyata untuk memastikan bahwa sistem dapat berjalan sesuai dengan kebutuhan pengguna. Pengujian mencakup validasi integritas data dan uji coba skenario operasional seperti pencatatan barang, peminjaman,

pengembalian, perawatan, dan lain-lain.

IV. HASIL DAN PEMBAHASAN

Semantic modeling dalam perancangan database inventaris laboratorium digunakan untuk memastikan bahwa setiap entitas, atribut, dan relasinya di dalam sistem memiliki makna yang jelas dan sesuai dengan kebutuhan bisnis. Pada pendekatan ini setiap elemen dalam database tidak hanya direpresentasikan dalam bentuk tabel, tetapi juga dihubungkan secara konseptual berdasarkan proses bisnis yang terjadi di laboratorium. Pendekatan *top down* dimulai dari identifikasi kebutuhan sistem, mendefinisikan entitas utama, serta mendefinisikan atribut dan relasi antar entitas secara lebih rinci.

4.1 Analisis Kebutuhan

Sistem informasi inventaris laboratorium dirancang untuk meningkatkan efisiensi dalam pencatatan, pengelolaan, serta aksesibilitas data terkait barang inventaris laboratorium. Melalui sistem ini, maka proses pencatatan barang, peminjaman, pengembalian, perawatan, dan pelaporan dapat dilakukan dengan lebih cepat dan akurat dibandingkan dengan metode pencatatan manual. Pengguna utama sistem ini adalah admin, laboran, dan pimpinan. Admin dan laboran bertanggung jawab mencatat barang masuk dan keluar. Pimpinan dapat menampilkan laporan inventaris dan menggunakannya untuk pengambilan keputusan strategis terkait pengadaan dan pemeliharaan barang laboratorium.

Data yang perlu dikelola dalam database inventaris laboratorium meliputi barang, pengguna, laboratorium, peminjaman, pengembalian, alokasi, penganggaran, dan perawatan seperti yang ditunjukkan pada tabel 1.

Tabel 1 Kebutuhan data dalam database

No	Data	Keterangan
1	Peran	Menyimpan jenis peran pengguna dalam sistem, seperti Admin, Laboran, dan Pimpinan
2	Pengguna	Menyimpan informasi pengguna sistem yang dapat mengakses fitur tertentu sesuai perannya
3	Gudang	Menyimpan informasi lokasi penyimpanan barang sebelum dialokasikan
4	Barang	Menyimpan informasi aset yang tersedia, seperti kamera, komputer, speaker, dan lain-lain
5	Penganggaran	Mencatat pengajuan anggaran untuk pembelian barang baru, serta persetujuan dari pimpinan
6	Alokasi	Menyimpan informasi distribusi barang dari Gudang ke Laboratorium atau Kantor
7	Laboratorium	Menyimpan informasi tentang laboratorium
8	Kantor	Menyimpan informasi tentang kantor yang digunakan oleh staff laboratorium
9	Peminjam	Menyimpan informasi peminjam yang berupa mahasiswa, dosen, atau staff
10	Peminjaman	Mencatat transaksi peminjaman barang oleh peminjam
11	Detail_Peminjaman	Menghubungkan peminjaman dengan barang yang dipinjam dalam satu transaksi
12	Pengembalian	Mencatat pengembalian barang yang telah dipinjam
13	Perawatan	Mencatat informasi pemeliharaan atau perbaikan barang yang rusak

Alur data dalam database inventaris laboratorium dimulai dari laboran mengajukan penganggaran barang yang dibutuhkan untuk mendukung operasional laboratorium. Apabila barang disetujui, maka admin atau laboran akan memasukkan data barang ke dalam database sebagai inventaris baru. Setiap barang yang telah terdaftar dapat langsung dialokasikan ke laboratorium/studio tertentu dan kantor untuk penggunaan jangka panjang, serta dapat dipinjam oleh staff, dosen, atau mahasiswa untuk keperluan akademik, penelitian, atau pengabdian kepada masyarakat. Jika terjadi kerusakan pada barang, maka admin/laboran akan mencatat barang tersebut dalam modul perawatan. Setelah perawatan selesai, maka status barang dapat diperbarui menjadi baik (jika berhasil diperbaiki) atau tetap rusak (jika tidak dapat diperbaiki).

4.2 Pemodelan Data Konseptual

Pemodelan Data Konseptual merupakan tahap awal dalam perancangan database yang berfokus pada identifikasi entitas, atribut, dan hubungan antar entitas tanpa mempertimbangkan detail teknis seperti tipe data atau implementasi dalam DBMS. Model ini digunakan untuk memberikan gambaran menyeluruh tentang bagaimana data dalam sistem akan diorganisasikan dan berinteraksi, sehingga memastikan bahwa database yang dibangun sesuai dengan kebutuhan bisnis dan operasional.

Pemodelan data konseptual dirancang dalam bentuk *Entity Relationship Diagram* (ERD) yang mencakup entitas, atribut, dan hubungan antar entitas. Berdasarkan kebutuhan data, maka terdapat 13 entitas utama yang membentuk struktur database, yaitu peran, pengguna, gudang, barang, penganggaran, alokasi, laboratorium, kantor, peminjam, peminjaman, detail_peminjaman, pengembalian, dan perawatan seperti yang ditunjukkan pada tabel 2.

Tabel 2 Entitas dalam database

No	Entitas	Primary Key	Atribut
1	Peran	ID_Peran	Nama_Peran
2	Pengguna	ID_Pengguna	Nama, Username, Password
3	Gudang	ID_Gudang	Nama_Gudang

No	Entitas	Primary Key	Atribut
4	Barang	ID_Barang	Nama_Barang, Kategori, Spesifikasi, Jumlah, Kondisi, Tahun_Pengadaan, Status
5	Penganggaran	ID_Penganggaran	Harga_Per_Unit, Jumlah_Barang, Total_Anggaran, Tanggal_Pengajuan, Status, Tanggal_Keputusan
6	Laboratorium	ID_Laboratorium	Nama_Laboratorium
7	Kantor	ID_Kantor	Nama_Kantor
8	Alokasi	ID_Alokasi	Jumlah_Barang, Tanggal_Alokasi, Keterangan_Alokasi
9	Peminjam	ID_Peminjam	NIM_NIK, Nama_Peminjam, Jenis_Peminjam, No_Telepon, Email
10	Peminjaman	ID_Peminjaman	Tanggal_Peminjaman, Tanggal_Kembali, Keperluan_Peminjaman
11	Detail_Peminjaman	ID_Detail_Peminjaman	Jumlah_Barang_Dipinjam
12	Pengembalian	ID_Pengembalian	Tanggal_Pengembalian, Kondisi_Saat_Kembali, Status_Pengembalian
13	Perawatan	ID_Perawatan	Tanggal_Perawatan, Jenis_Perawatan, Biaya_Perawatan, Status_Perawatan

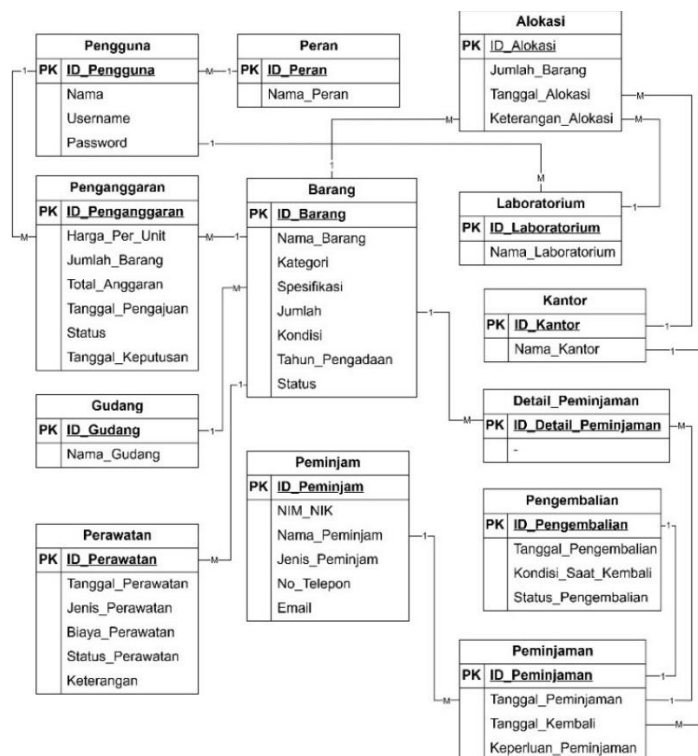
Tabel di atas merepresentasikan struktur entitas dalam database inventaris laboratorium, di mana setiap entitas memiliki *Primary Key* (PK) yang digunakan sebagai pengidentifikasi unik dan sejumlah atribut yang menyimpan informasi terkait entitas tersebut. Sebagai contoh pada entitas Barang memiliki PK berupa ID_Barang dan atribut berupa Nama_Barang, Kategori, Spesifikasi, Jumlah, Kondisi, Tahun_Pengadaan, dan Status.

Setiap entitas dalam database inventaris laboratorium tidak berdiri sendiri, tetapi juga memiliki hubungan yang bermakna dengan entitas lain sesuai dengan proses bisnis yang terjadi di laboratorium. Hubungan ini disebut dengan relasi antar entitas yang bertujuan untuk memastikan bahwa struktur data mencerminkan hubungan nyata antar objek dalam sistem, sehingga memungkinkan pengelolaan data yang lebih efisien, terorganisir, dan mudah diakses. Hasil relasi antar entitas ditunjukkan pada tabel 3.

Tabel 3 Relasi antar entitas

No	Hubungan	Kardinalitas	Makna relasi antar entitas
1	Peran : Pengguna	One to Many	Satu peran bisa dimiliki oleh banyak pengguna, tetapi satu pengguna hanya memiliki satu peran
2	Pengguna : Penganggaran	One to Many	Seorang pengguna dapat membuat banyak pengajuan anggaran, tetapi satu pengajuan anggaran dibuat oleh satu pengguna
3	Pengguna : Penganggaran	One to Many	Seorang pengguna dengan peran pimpinan dapat menyetujui banyak anggaran
4	Pengguna : Laboratorium	One to Many	Seorang pengguna dengan peran laboran dapat menjadi penanggung jawab lebih dari satu laboratorium
5	Barang : Penganggaran	One to Many	Satu barang dapat diajukan untuk penganggaran beberapa kali
6	Barang : Alokasi	One to Many	Satu barang dapat dialokasikan ke beberapa tempat (kantor dan laboratorium)
7	Gudang : Barang	One to Many	Satu gudang dapat menyimpan banyak barang, tetapi satu barang hanya ada di satu gudang
8	Alokasi : Laboratorium	Many to One	Banyak alokasi bisa terjadi di satu laboratorium
9	Alokasi : Kantor	Many to One	Banyak alokasi bisa terjadi di satu kantor
10	Peminjam : Peminjaman	One to Many	Satu peminjam bisa melakukan banyak peminjaman, tetapi satu peminjaman hanya dilakukan oleh satu peminjam
11	Peminjaman : Detail_Peminjaman	One to Many	Satu peminjaman bisa terdiri dari banyak detail peminjaman
12	Detail_Peminjaman : Barang	Many to One	Banyak detail peminjaman bisa merujuk ke satu barang
13	Kantor : Peminjaman	One to Many	Satu kantor bisa menyediakan banyak barang yang dapat dipinjam
14	Peminjaman : Pengembalian	One to One	Satu peminjaman memiliki satu pengembalian
15	Barang : Perawatan	One to Many	Satu barang bisa mendapatkan banyak perawatan selama masa pakainya

Hubungan antar entitas di atas dirancang berdasarkan aturan bisnis dan proses operasional yang terjadi dalam pengelolaan inventaris laboratorium. Relasi antara barang dan laboratorium memiliki hubungan *many to one* yang berarti setiap barang dapat berada di satu laboratorium tertentu, tetapi satu laboratorium dapat memiliki banyak barang. Sebagai contoh, laboratorium pemrograman dapat memiliki 80 unit CPU, monitor, keyboard, dan lain-lain. Berdasarkan relasi antar entitas di atas, maka dihasilkan model ERD seperti yang ditunjukkan pada gambar 2.



Gambar 2 Perancangan ERD

Entity Relationship Diagram (ERD) di atas menggambarkan struktur data dalam sistem pengelolaan laboratorium yang mencakup berbagai entitas dan relasinya. Pengguna memiliki atribut dasar seperti nama, *username*, dan *password*, serta berelasi dengan entitas Peran. Relasi ini untuk menentukan apakah pengguna bertindak sebagai Admin, Laboran, atau Pimpinan. Barang sebagai entitas utama yang mencatat informasi barang laboratorium yang memiliki atribut nama_barang, kategori, spesifikasi, jumlah, kondisi, tahun_pengadaan, dan status. Setiap barang tersebut dapat dialokasikan ke laboratorium atau kantor untuk penggunaan jangka panjang.

Proses peminjaman dikelola oleh entitas Peminjam yang berisi 3 jenis peminjam, yaitu informasi Mahasiswa, Dosen, atau Staf. Entitas Peminjaman mencatat transaksi peminjaman barang dengan atribut tanggal_peminjaman, tanggal_kembali, dan keperluan. Relasi Peminjaman dengan Barang dipisah menggunakan entitas Detail_Peminjaman yang memungkinkan satu transaksi peminjaman dapat melibatkan beberapa barang. Setelah barang dikembalikan, entitas Pengembalian mencatat tanggal pengembalian, kondisi barang saat dikembalikan, dan status pengembalian.

Perancangan database juga mencakup Penganggaran yang memungkinkan pengguna Admin atau Laboran untuk mengajukan anggaran pengadaan barang baru, sementara Pimpinan bertindak sebagai pemberi keputusan dengan status Disetujui, Ditolak, atau Menunggu. Selain itu, terdapat entitas Perawatan yang mencatat tanggal perawatan, jenis perawatan (perbaikan, pengecekan rutin, penggantian suku cadang), biaya, dan status perawatan.

4.3 Mendesain Database Logis

Mendesain Database Logis dalam database inventaris laboratorium bertujuan untuk menyusun struktur data yang optimal guna memastikan efisiensi, keakuratan, dan keamanan dalam pengelolaan aset laboratorium sebelum diterapkan dalam *Database Management System* (DBMS) MySQL. Pada tahap ini, model konseptual yang dibuat sebelumnya dikonversi menjadi entitas, atribut, relasi, dan aturan normalisasi untuk memastikan efisiensi dan integritas data. Normalisasi data digunakan untuk mengurangi redundansi dan meningkatkan integritas data melalui penggunaan *Primary Key* (PK) dan *Foreign Key* (FK).

Tabel 4 Desain database logis entitas Peran

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Peran (PK)	INT (AUTO)	<i>Primary Key</i> , identitas unik peran
2	Nama_Peran	ENUM('Admin', 'Laboran', 'Pimpinan')	Jenis peran pengguna

Tabel 5 Desain database logis entitas Pengguna

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Pengguna (PK)	INT (AUTO)	<i>Primary Key</i> , identitas unik pengguna
2	ID_Peran (FK)	INT	<i>Foreign Key</i> , menghubungkan ke tabel Peran
3	Nama	VARCHAR(100)	Nama lengkap pengguna
4	Username	VARCHAR(50)	Nama pengguna untuk login
5	Password	VARCHAR(255)	Kata sandi pengguna

Tabel 6 Desain database logis entitas Gudang

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Gudang (PK)	INT (AUTO)	Primary Key, identitas unik gudang
2	Nama_Gudang	VARCHAR(100)	Nama gudang

Tabel 7 Desain database logis entitas Barang

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Barang (PK)	INT (AUTO)	Primary Key, identitas unik barang
2	ID_Gudang (FK)	INT	Foreign Key, menghubungkan ke tabel Gudang
3	Nama_Barang	VARCHAR(100)	Nama barang
4	Kategori	VARCHAR(50)	Jenis atau kategori barang
5	Spesifikasi	TEXT	Deskripsi spesifikasi barang
6	Jumlah	INT	Jumlah barang tersedia
7	Kondisi (Baik, Rusak)	ENUM('Baik', 'Rusak')	Kondisi barang
8	Tahun_Pengadaan	YEAR	Tahun pengadaan barang
9	Status	ENUM('Tersedia', 'Dialokasikan', 'Dipinjam', 'Dihapus')	Status ketersediaan barang

Tabel 8 Desain database logis entitas Penganggaran

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Penganggaran	INT (AUTO)	Primary Key, identitas unik penganggaran
2	ID_Barang (FK)	INT	Foreign Key, menghubungkan ke tabel Barang
3	ID_Pengaju (FK)	INT	Foreign Key, pengguna yang mengajukan anggaran
4	ID_Pimpinan (FK)	INT	Foreign Key, pengguna pimpinan yang menyetujui
5	Harga_Per_Unit	DECIMAL(10,2)	Harga satuan barang
6	Jumlah_Barang	INT	Jumlah barang diajukan
7	Total_Anggaran	DECIMAL(15,2)	Total anggaran barang
8	Tanggal_Pengajuan	DATE	Tanggal pengajuan anggaran
9	Status	ENUM('Disetujui', 'Ditolak', 'Menunggu')	Status pengajuan anggaran
10	Tanggal_Keputusan	DATE	Tanggal keputusan anggaran

Tabel 9 Desain database logis entitas Laboratorium

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Laboratorium (PK)	INT (AUTO)	Primary Key, identitas unik laboratorium
2	ID_Laboran	INT	Foreign Key, menghubungkan ke tabel Pengguna dengan peran Laboran
3	Nama_Laboratorium	VARCHAR(100)	Nama laboratorium

Tabel 10 Desain database logis entitas Kantor

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Kantor (PK)	INT (AUTO)	Primary Key, identitas unik kantor
2	Nama_Kantor	VARCHAR(100)	Nama kantor

Tabel 11 Desain database logis entitas Alokasi

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Alokasi (PK)	INT (AUTO)	Primary Key, identitas unik alokasi
2	ID_Barang (FK)	INT	Foreign Key, menghubungkan ke tabel Barang
3	ID_Kantor (FK)	INT (NULLABLE)	Foreign Key, menghubungkan ke tabel Kantor, NULL jika dialokasikan hanya ke Laboratorium
4	ID_Laboratorium (FK)	INT (NULLABLE)	Foreign Key, menghubungkan ke tabel Laboratorium, NULL jika dialokasikan hanya ke Kantor
5	Jumlah_Barang	INT	Jumlah barang yang dialokasikan
6	Tanggal_Alokasi	DATE	Tanggal barang dialokasikan
7	Keterangan_Alokasi	TEXT	Keterangan alokasi barang

Tabel 12 Desain database logis entitas Peminjam

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Peminjam (PK)	INT (AUTO)	Primary Key, identitas unik peminjam

No	Nama Atribut	Tipe Data	Keterangan
2	NIM_NIK	VARCHAR(20)	Nomor identitas peminjam
3	Nama_Peminjam	VARCHAR(100)	Nama peminjam
4	Jenis_Peminjam	ENUM('Mahasiswa', 'Dosen', 'Staff')	Jenis peminjam
5	No_Telepon	VARCHAR(20)	Nomor telepon peminjam
6	Email	VARCHAR(100)	Email peminjam

Tabel 13 Desain database logis entitas Peminjaman

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Peminjaman (PK)	INT (AUTO)	Primary Key, identitas unik peminjaman
2	ID_Peminjam (FK)	INT	Foreign Key, menghubungkan ke tabel Peminjam
3	ID_Kantor	INT	Foreign Key, menghubungkan ke tabel Kantor
4	Tanggal_Peminjaman	DATE	Tanggal barang dipinjam
5	Tanggal_Kembali	DATE	Tanggal barang harus dikembalikan
6	Keperluan_Peminjaman	TEXT	Tujuan peminjaman barang

Tabel 14 Desain database logis entitas Detail_Peminjaman

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Detail_Peminjaman (PK)	INT (AUTO)	Primary Key, identitas unik transaksi detail peminjaman
2	ID_Peminjaman (FK)	INT	Foreign Key, menghubungkan ke tabel Peminjaman
3	ID_Barang (FK)	INT	Foreign Key, menghubungkan ke tabel Barang
4	Jumlah_Barang_Dipinjam	INT	Jumlah barang yang dipinjam dalam transaksi

Tabel 15 Desain database logis entitas Pengembalian

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Pengembalian (PK)	INT (AUTO)	Primary Key, identitas unik pengembalian
2	ID_Peminjaman (FK)	INT	Foreign Key, menghubungkan ke tabel Peminjaman
3	Tanggal_Pengembalian	DATE	Tanggal barang dikembalikan
4	Kondisi_Saat_Kembali	ENUM('Baik', 'Rusak')	Kondisi barang saat dikembalikan
5	Status_Pengembalian	ENUM('Dikembalikan', 'Terlambat')	Status pengembalian

Tabel 16 Desain database logis entitas Perawatan

No	Nama Atribut	Tipe Data	Keterangan
1	ID_Perawatan	INT (AUTO)	Primary Key, identitas unik perawatan
2	ID_Barang (FK)	INT	Foreign Key, menghubungkan ke tabel Barang
3	Tanggal_Perawatan	DATE	Tanggal perawatan dilakukan
4	Jenis_Perawatan	ENUM('Perbaikan', 'Pengecekan Rutin', 'Penggantian Suku Cadang')	Jenis perawatan
5	Biaya_Perawatan	DECIMAL(10,2)	Biaya perawatan
6	Status_Perawatan	ENUM('Selesai', 'Dalam Proses')	Status perawatan
7	Keterangan	TEXT	Keterangan barang

Hasil desain database logis di atas mencerminkan struktur data yang sistematis untuk mengelola inventaris laboratorium yang meliputi penganggaran, pengalokasian, peminjaman, pengembalian, dan perawatan barang. Setiap entitas memiliki atribut yang mendukung kebutuhan sistem dengan penggunaan *primary key* dan *foreign key* (FK). Entitas Barang menyimpan informasi barang laboratorium, sementara entitas Pengguna dan entitas Peran menentukan hak akses dalam sistem. Entitas Penganggaran mencatat proses pengajuan dan persetujuan anggaran barang, sedangkan Entitas Alokasi, Laboratorium, dan Kantor ditujukan untuk mengelola proses pengalokasian barang. Entitas Peminjam, Peminjaman, dan Detail_Peminjaman mencatat transaksi peminjaman, sedangkan entitas Pengembalian mengelola proses pengembalian barang. Terakhir, entitas Perawatan digunakan untuk memastikan pemeliharaan barang supaya dalam kondisi optimal.

4.4 Mendesain Database Fisik

Mendesain database fisik merupakan tahap akhir dalam perancangan database yang bertujuan untuk mengimplementasikan struktur data ke dalam sistem manajemen database (DBMS) secara konkret. Tahap ini bertujuan untuk mengoptimalkan penyimpanan, memastikan integritas data, serta meningkatkan efisiensi akses dan manipulasi data.

Tabel 17 Pembuatan tabel Peran

```
CREATE TABLE Peran (
  ID_Peran INT AUTO_INCREMENT PRIMARY KEY,
  Nama_Peran ENUM('Admin', 'Laboran', 'Pimpinan') NOT NULL
);
```

Tabel 18 Pembuatan tabel Pengguna

```
CREATE TABLE Pengguna (  
  ID_Pengguna INT AUTO_INCREMENT PRIMARY KEY,  
  ID_Peran INT,  
  Nama VARCHAR(100) NOT NULL,  
  Username VARCHAR(50) NOT NULL UNIQUE,  
  Password VARCHAR(255) NOT NULL,  
  FOREIGN KEY (ID_Peran) REFERENCES Peran(ID_Peran)  
);
```

Tabel 19 Pembuatan tabel Gudang

```
CREATE TABLE Gudang (  
  ID_Gudang INT AUTO_INCREMENT PRIMARY KEY,  
  Nama_Gudang VARCHAR(100) NOT NULL  
);
```

Tabel 20 Pembuatan tabel Barang

```
CREATE TABLE Barang (  
  ID_Barang INT AUTO_INCREMENT PRIMARY KEY,  
  ID_Gudang INT,  
  Nama_Barang VARCHAR(100) NOT NULL,  
  Kategori VARCHAR(50),  
  Spesifikasi TEXT,  
  Jumlah INT,  
  Kondisi ENUM('Baik', 'Rusak') NOT NULL,  
  Tahun_Pengadaan YEAR,  
  Status ENUM('Tersedia', 'Dialokasikan', 'Dipinjam', 'Dihapus') NOT NULL,  
  FOREIGN KEY (ID_Gudang) REFERENCES Gudang(ID_Gudang)  
);
```

Tabel 21 Pembuatan tabel Penganggaran

```
CREATE TABLE Penganggaran (  
  ID_Penganggaran INT AUTO_INCREMENT PRIMARY KEY,  
  ID_Barang INT,  
  ID_Pengaju INT,  
  ID_Pimpinan INT,  
  Harga_Per_Unit DECIMAL(10,2),  
  Jumlah_Barang INT,  
  Total_Anggaran DECIMAL(15,2),  
  Tanggal_Pengajuan DATE,  
  Status ENUM('Disetujui', 'Ditolak', 'Menunggu'),  
  Tanggal_Keputusan DATE,  
  FOREIGN KEY (ID_Barang) REFERENCES Barang(ID_Barang),  
  FOREIGN KEY (ID_Pengaju) REFERENCES Pengguna(ID_Pengguna),  
  FOREIGN KEY (ID_Pimpinan) REFERENCES Pengguna(ID_Pengguna)  
);
```

Tabel 22 Pembuatan tabel Laboratorium

```
CREATE TABLE Laboratorium (  
  ID_Laboratorium INT AUTO_INCREMENT PRIMARY KEY,  
  ID_Laboran INT,  
  Nama_Laboratorium VARCHAR(100) NOT NULL,  
  FOREIGN KEY (ID_Laboran) REFERENCES Pengguna(ID_Pengguna)  
);
```

Tabel 23 Pembuatan tabel Kantor

```
CREATE TABLE Kantor (  
  ID_Kantor INT AUTO_INCREMENT PRIMARY KEY,  
  Nama_Kantor VARCHAR(100) NOT NULL  
);
```

Tabel 24 Pembuatan tabel Alokasi

```
CREATE TABLE Alokasi (  
  ID_Alokasi INT AUTO_INCREMENT PRIMARY KEY,
```

```
ID_Barang INT NOT NULL,  
ID_Kantor INT NULL,  
ID_Laboratorium INT NULL,  
Jumlah_Barang INT NOT NULL,  
Tanggal_Alokasi DATE NOT NULL,  
Keterangan_Alokasi TEXT,  
FOREIGN KEY (ID_Barang) REFERENCES Barang(ID_Barang),  
FOREIGN KEY (ID_Kantor) REFERENCES Kantor(ID_Kantor),  
FOREIGN KEY (ID_Laboratorium) REFERENCES Laboratorium(ID_Laboratorium)  
);
```

Tabel 25 Pembuatan tabel Peminjam

```
CREATE TABLE Peminjam (  
ID_Peminjam INT AUTO_INCREMENT PRIMARY KEY,  
NIM_NIK VARCHAR(20) NOT NULL,  
Nama_Peminjam VARCHAR(100) NOT NULL,  
Jenis_Peminjam ENUM('Mahasiswa', 'Dosen', 'Staff') NOT NULL,  
No_Telepon VARCHAR(20),  
Email VARCHAR(100)  
);
```

Tabel 26 Pembuatan tabel Peminjaman

```
CREATE TABLE Peminjaman (  
ID_Peminjaman INT AUTO_INCREMENT PRIMARY KEY,  
ID_Peminjam INT NOT NULL,  
ID_Kantor INT NOT NULL,  
Tanggal_Peminjaman DATE,  
Tanggal_Kembali DATE,  
Keperluan_Peminjaman TEXT,  
FOREIGN KEY (ID_Peminjam) REFERENCES Peminjam(ID_Peminjam),  
FOREIGN KEY (ID_Kantor) REFERENCES Kantor(ID_Kantor)  
);
```

Tabel 27 Pembuatan tabel Detail_Peminjaman

```
CREATE TABLE Detail_Peminjaman (  
ID_Detail_Peminjaman INT AUTO_INCREMENT PRIMARY KEY,  
ID_Peminjaman INT,  
ID_Barang INT,  
Jumlah_Barang_Dipinjam INT,  
FOREIGN KEY (ID_Peminjaman) REFERENCES Peminjaman(ID_Peminjaman),  
FOREIGN KEY (ID_Barang) REFERENCES Barang(ID_Barang)  
);
```

Tabel 28 Pembuatan tabel Pengembalian

```
CREATE TABLE Pengembalian (  
ID_Pengembalian INT AUTO_INCREMENT PRIMARY KEY,  
ID_Peminjaman INT UNIQUE,  
Tanggal_Pengembalian DATE,  
Kondisi_Saat_Kembali ENUM('Baik', 'Rusak') NOT NULL,  
Status_Pengembalian ENUM('Dikembalikan', 'Terlambat'),  
FOREIGN KEY (ID_Peminjaman) REFERENCES Peminjaman(ID_Peminjaman)  
);
```

Tabel 29 Pembuatan tabel Perawatan

```
CREATE TABLE Perawatan (  
ID_Perawatan INT AUTO_INCREMENT PRIMARY KEY,  
ID_Barang INT,  
Tanggal_Perawatan DATE,  
Jenis_Perawatan ENUM('Perbaikan', 'Pengecekan Rutin', 'Penggantian Suku Cadang'),  
Biaya_Perawatan DECIMAL(10,2),  
Status_Perawatan ENUM('Selesai', 'Dalam Proses'),  
Keterangan TEXT,  
FOREIGN KEY (ID_Barang) REFERENCES Barang(ID_Barang)  
);
```

Setiap entitas direpresentasikan sebagai tabel dengan atributnya masing-masing menggunakan tipe data tertentu agar efisien dalam penyimpanan dan pengolahan data. Sebagai contoh, kode SQL pada tabel Barang digunakan untuk menyimpan data barang inventaris laboratorium. Setiap barang diidentifikasi secara unik melalui kolom ID_Barang yang bertipe INT dan diatur sebagai PRIMARY KEY dengan jenis AUTO_INCREMENT agar nilainya bertambah secara otomatis. Tabel barang terhubung dengan tabel gudang melalui kolom ID_Gudang yang bertindak sebagai FOREIGN KEY. Kolom Nama_Barang bertipe VARCHAR(100) menyimpan nama barang, sedangkan kolom Kategori bertipe VARCHAR(50) mencatat jenis atau kelompok barang tersebut. Kolom Spesifikasi bertipe TEXT yang digunakan untuk menjelaskan secara rinci karakteristik barang. Kolom Jumlah bertipe INT menyimpan informasi jumlah stok barang. Kolom Kondisi bertipe ENUM mencatat kondisi fisik barang dan hanya bernilai 'Baik' atau 'Rusak'. Atribut Tahun_Pengadaan bertipe YEAR mencatat tahun pengadaan barang. Terakhir, kolom Status bertipe ENUM mencatat status keberadaan atau penggunaan barang dan hanya bernilai 'Tersedia', 'Dialokasikan', 'Dipinjam', atau 'Dihapus'.

4.5 Implementasi

Implementasi adalah proses memasukkan data nyata ke dalam database yang telah dirancang. Tujuannya untuk memastikan bahwa semua tabel berfungsi dengan baik dan sesuai dengan perancangan awal database. Selain itu, implementasi juga bertujuan untuk memastikan database dapat memenuhi kebutuhan operasional laboratorium. Sebagai contoh, implementasi data ke dalam tabel Barang ditunjukkan pada gambar 3.

	ID_Barang	ID_Gudang	Nama_Barang	Kategori	Spesifikasi	Jumlah	Kondisi	Tahun_Pengadaan	Status
Edit Copy Delete	1	1	Dell Vostro 3510	Laptop	Intel Core i5, RAM 16GB, SSD 256GB	6	Baik	2018	Dialokasikan
Edit Copy Delete	2	1	Dell Inspiron 5400	Personal Computer	Intel Core i5, RAM 8GB, SSD 256GB	80	Baik	2017	Tersedia
Edit Copy Delete	3	1	Kursi Lipat Chitose	Kursi	Kaki lipat biasa	160	Baik	2022	Tersedia
Edit Copy Delete	4	2	Proyektor Epson EB-X500	Proyektor	3600 lumens, HDMI/VGA input	10	Baik	2021	Dialokasikan
Edit Copy Delete	5	2	Proyektor BenQ MX560	Proyektor	4000 lumens, HDMI/VGA input	6	Baik	2022	Dialokasikan
Edit Copy Delete	6	1	Mouse Logitech B100	Mouse	Kabel port USB	80	Baik	2020	Tersedia
Edit Copy Delete	7	1	Keyboard Logitech MK215	Keyboard	Kabel port USB	80	Baik	2020	Tersedia
Edit Copy Delete	8	1	Lenovo LED L24i-4A	Monitor	24 inch, IPS Flat, 1920 x 1080 Full HD	80	Baik	2020	Dialokasikan

Gambar 3 Implementasi tabel Barang

Gambar di atas menampilkan isi tabel Barang dalam database inventaris laboratorium. Tabel ini berfungsi untuk mencatat seluruh informasi penting mengenai barang-barang inventaris laboratorium. Setiap entri dalam tabel merepresentasikan satu jenis barang dengan atribut-atribut utama seperti ID_Barang sebagai identifikasi unik, ID_Gudang sebagai tempat penyimpanan barang, Nama_Barang sebagai nama suatu barang, dan Kategori sebagai pengelompokan jenis barang seperti Laptop, Proyektor, Kursi, dan lainnya.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan database inventaris laboratorium menggunakan *semantic modeling* dengan pendekatan metode *top down*, maka dapat disimpulkan bahwa:

1. Metode *semantic modeling* dengan pendekatan *top down* efektif untuk merancang database inventaris laboratorium, karena dimulai dari pemahaman menyeluruh terhadap sistem yang akan dikembangkan melalui pemetaan entitas dan relasinya secara semantik dan terstruktur.
2. Model relasional yang diperoleh dari transformasi model konseptual telah memenuhi kebutuhan pengelolaan data inventaris laboratorium yang efisien dan sistematis.

Adapun saran dalam penelitian ini adalah sebagai berikut:

1. Implementasi database yang telah dirancang perlu dilakukan pengujian bersamaan dengan pengujian sistem informasi inventaris laboratorium untuk memastikan fungsionalitas dan kinerjanya dapat berjalan sesuai dengan skenario nyata.
2. Perlu dilakukan evaluasi berkala terhadap kebutuhan pengguna supaya database yang dikembangkan dapat terus relevan dan menyesuaikan dengan perkembangan teknologi maupun kebijakan institusi.

DAFTAR PUSTAKA

- [1] P. R. Indonesia, *Peraturan Pemerintah Republik Indonesia Nomor 60 Tahun 1999 tentang Perguruan Tinggi Pasal 56*. Jakarta: Sekretariat Negara, 1999.
- [2] X. Peng, S. Wang, and Y. Zhang, "Automatic Generation Technology of Table Structure for Software Development Based on Oracle Database System," *Proc. - 2022 Int. Symp. Adv. Informatics, Electron. Educ. ISAIEE 2022*, pp. 342–346, 2022, doi: 10.1109/ISAIEE57420.2022.00077.
- [3] T. D. T. Nguyen, V. Nguyen, V. N. Pham, L. N. T. Huynh, M. D. Hossain, and E. N. Huh, "Modeling Data Redundancy and Cost-Aware Task Allocation in MEC-Enabled Internet-of-Vehicles Applications," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1687–1701, Feb. 2021, doi: 10.1109/JIOT.2020.3015534.

- [4] J. Grant and F. Parisi, "On measuring inconsistency in graph databases with regular path constraints," *Artif. Intell.*, vol. 335, p. 104197, Oct. 2024, doi: 10.1016/J.ARTINT.2024.104197.
- [5] P. Alexopoulos, *Semantic Modeling for Data*, First Edit. United States of America: O'Reilly Media, Inc, 2020.
- [6] H.-J. Kung, L. Kung, and A. Gardiner, "Comparing Top-down with Bottom-up Approaches: Teaching Data Modeling," *nformation Syst. Educ. J.*, pp. 14–24, 2013.
- [7] M. Krakowiak, "A SERM based framework to optimize the modelling of entity life history," *Procedia Comput. Sci.*, vol. 246, pp. 3898–3907, Jan. 2024, doi: 10.1016/J.PROCS.2024.09.163.
- [8] J. Cohen and J. Gil, "An entity-relationship model of the flow of waste and resources in city-regions: Improving knowledge management for the circular economy," *Resour. Conserv. Recycl. Adv.*, vol. 12, p. 200058, Dec. 2021, doi: 10.1016/J.RCRADV.2021.200058.
- [9] M. Paneque, M. del M. Roldán-García, C. Blanco, A. Maté, D. G. Rosado, and J. Trujillo, "An ontology-based secure design framework for graph-based databases," *Comput. Stand. Interfaces*, vol. 88, p. 103801, Mar. 2024, doi: 10.1016/J.CSI.2023.103801.
- [10] M. Javed and Y. Lin, "iMER: Iterative process of entity relationship and business process model extraction from the requirements," *Inf. Softw. Technol.*, vol. 135, p. 106558, Jul. 2021, doi: 10.1016/J.INFSOF.2021.106558.