

Database Analysis and Design Techniques

Shaesaryan Firmansyah - 11210910000008

PEMBAHASAN

01

The Information
System Lifecycle

02

Database System
Development Lifecycle

03

Database Planning

04

System Definition

05

Requirement Collection
and Analysis

06

Database Design

PEMBAHASAN

07

DBMS Selection

08

Applications Design

09

Prototyping

10

Implementation

11

Data Conversion and
Loading

12

Testing

PEMBAHASAN

13

Operational
Maintenance

14

CASE Tools

15

Data Administration
and Database
Administration

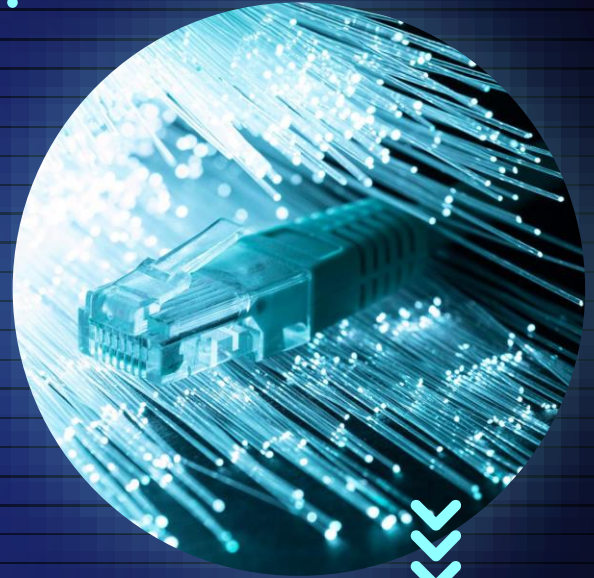


01

>>>> THE INFORMATION SYSTEM
LIFECYCLE

The Information System Lifecycle

Database system lifecycle(siklus dari sistem database) menggabungkan langkah-langkah dasar yang terlibat dalam merancang skema global dari basis data logis, mengalokasikan data di seluruh jaringan komputer, dan menentukan skema khusus DBMS lokal. Setelah desain selesai, siklus hidup akan berlanjut dengan implementasi dan Maintenance database.



◀◀◀◀

02

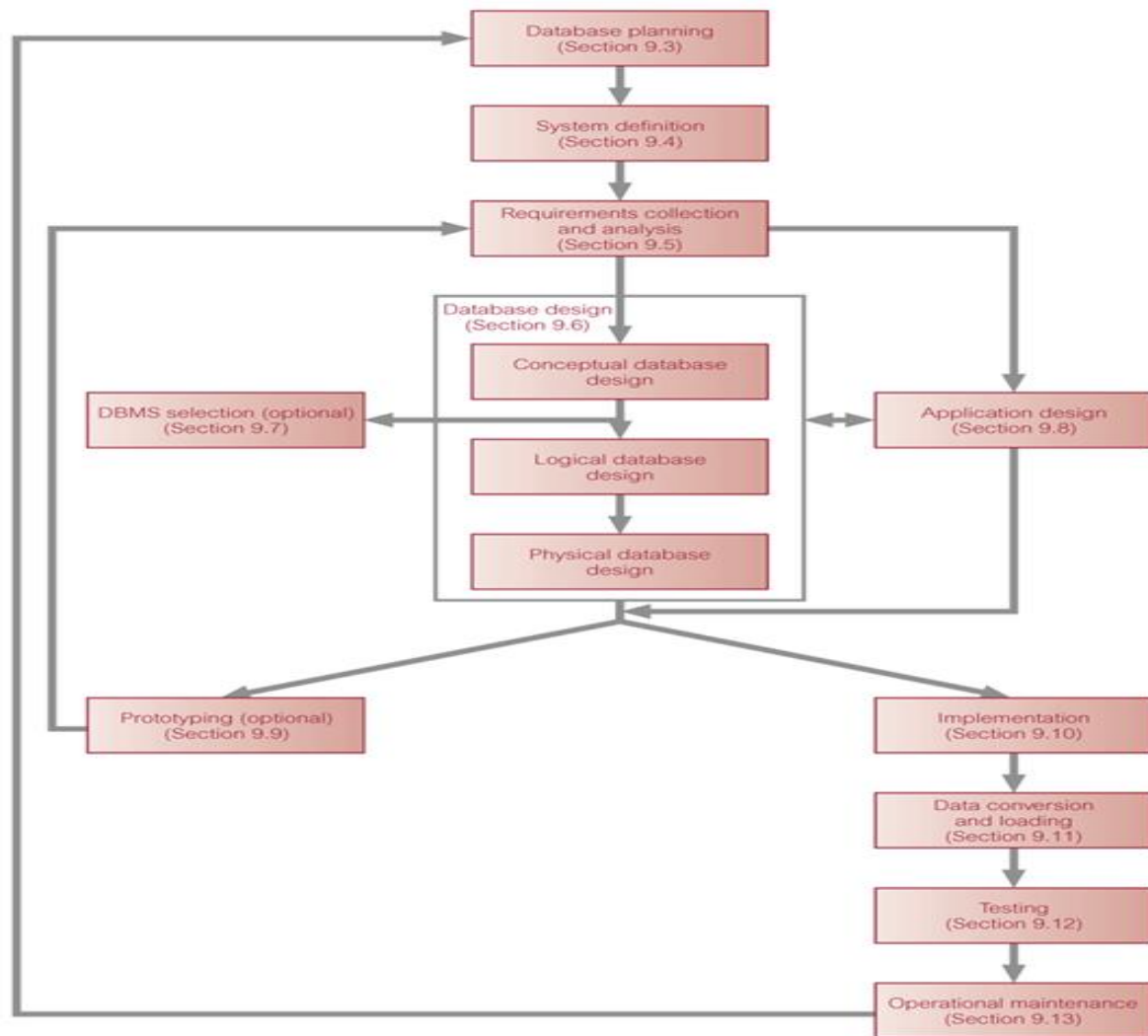
Database System

Development Lifecycle

▶▶▶▶▶

Database System Development Lifecycle

Dalam database system development lifecycle akan membahas tentang langkah-langkah khusus dalam merancang database. Tahapan siklus hidup pengembangan sistem basis data tidak sepenuhnya berurutan, tetapi melibatkan sejumlah pengulangan dari tahapan sebelumnya melalui feedback loops. Misalnya, masalah yang dihadapi selama desain database mungkin memerlukan pengumpulan dan analisis persyaratan tambahan.





03

Database Planning

Database Planning

Di langkah ini pembuat database membuat gambaran besar apa yang akan dilakukan dalam mengembangkan sebuah sistem database. Hasil dari Database Planning berupa dua hal, yaitu mission statement dan mission objective.



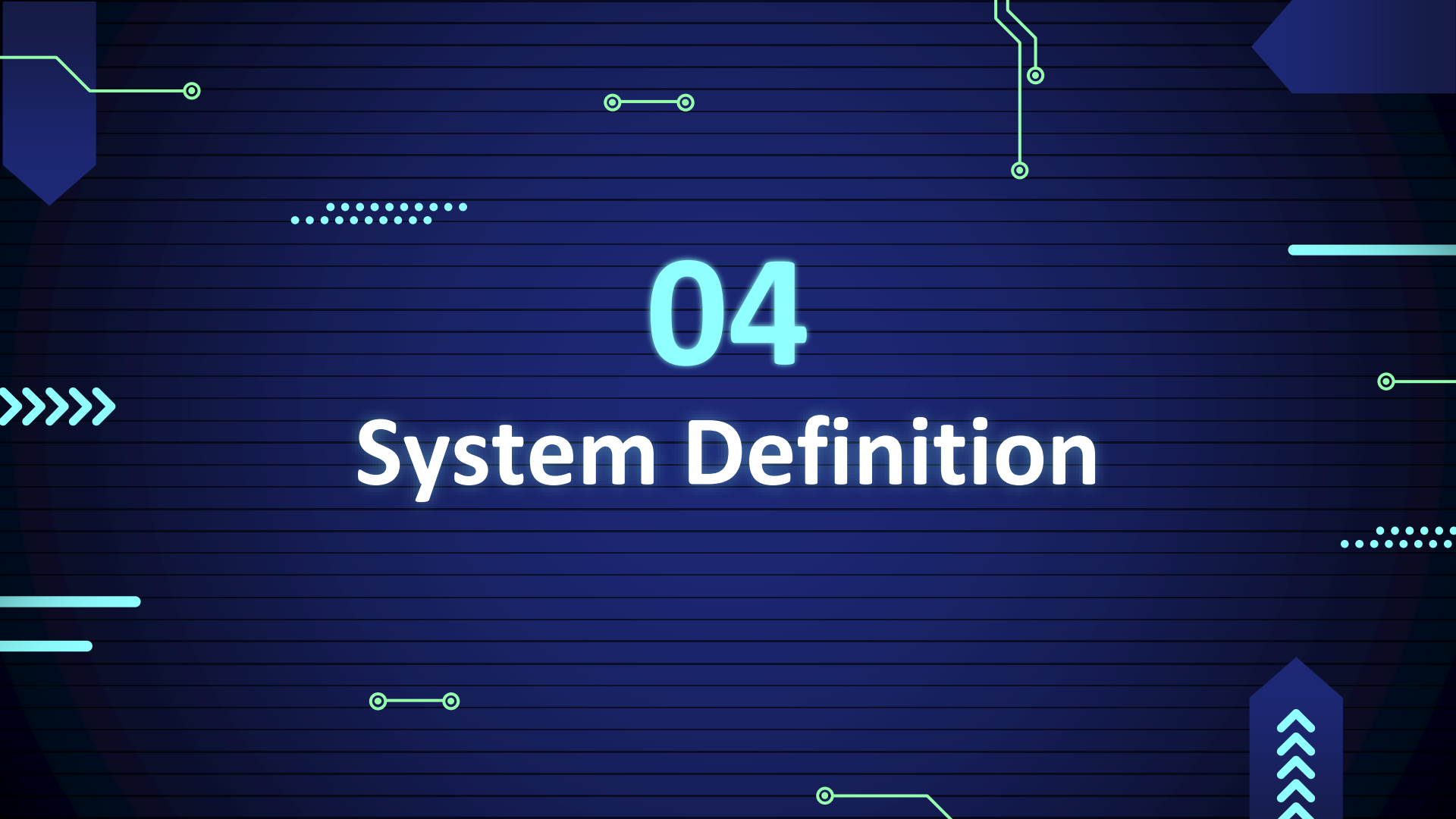
Mission Statement

Dapat berupa satu kalimat yang menjelaskan sistem yang akan dikembangkan.

Mission Objective

Dapat Berupa beberapa hal yang akan ada didalam sistem, seperti data apa saja yang akan ada sistem, perintah apa saja yang bisa dilakukan terhadap data tersebut, bisa berupa flag, atau hasil laporan yang nanti akan dikeluarkan oleh sistem.





04

System Definition

System Definition

Menjelaskan ruang lingkup dan batasan dalam pembuatan sistem dan user view yang akan berhadapan langsung dengan sistem. User view nantinya akan membantu kita untuk menetapkan apa saja yang dibutuhkan dalam sistem. User view dapat dilihat dari job role, seperti manager, staff atau supervisor atau application area dalam perusahaan, seperti marketing, finance atau sales. Menentukan user view yang tepat akan membantu kita membangun sistem, terutama dalam mengembangkan kebutuhan.



User View

Mendefinisikan apa yang diperlukan dari sistem basis data dari perspektif peran pekerjaan tertentu (seperti Manajer atau Supervisor) atau area aplikasi Perusahaan (seperti pemasaran, Personalia, atau kontrol stok). User view nantinya akan membantu kita untuk menetapkan apa saja yang dibutuhkan dalam sistem.



05

Requirement Collection and Analysis



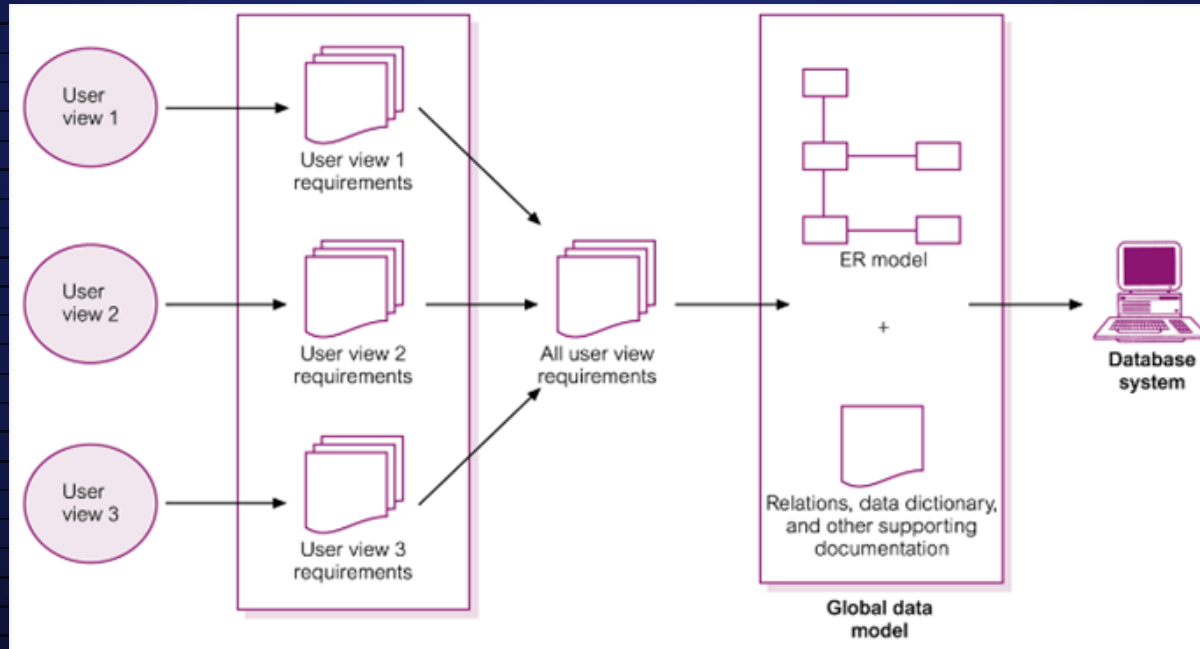
Requirement Collection and Analysis

Langkah ini mengharuskan kita mencari informasi kebutuhan dari user view yang sudah ditentukan. Informasi dapat berupa gambaran data yang digunakan atau dihasilkan, bagaimana data tersebut digunakan dan dihasilkan, dan kebutuhan lain dalam sistem baru. Nantinya informasi tersebut ditelaah untuk menentukan kebutuhan dalam sistem baru. Ada satu aktivitas penting dalam langkah ini, yaitu bagaimana informasi kebutuhan ini diolah. Pendekatannya ada 3, yaitu centralized, view integration, atau kombinasi dari kedua pendekatan tersebut.



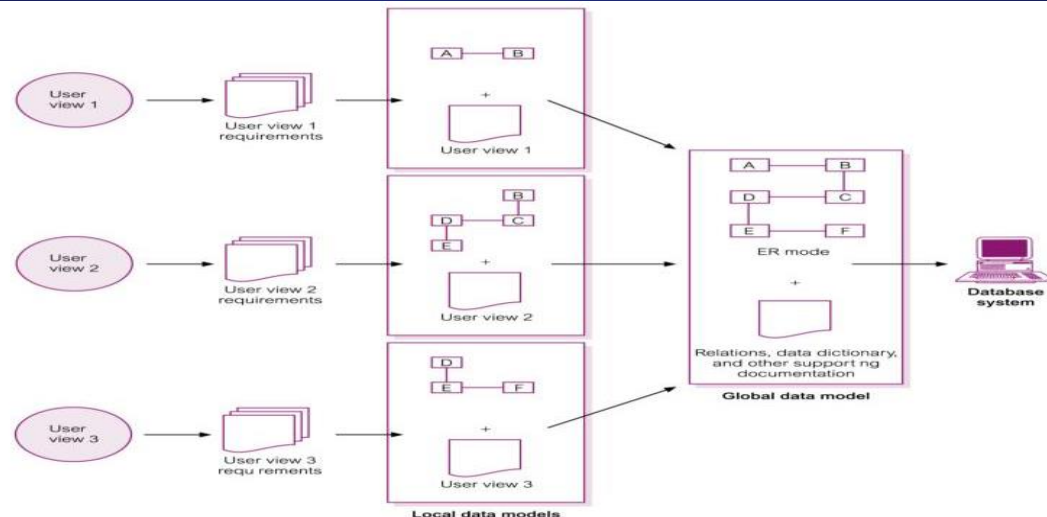
Centralized Approach

Pendekatan centralized mengumpulkan terlebih dahulu informasi kebutuhan dari user view lalu digabungkan lalu diproses menjadi data model dan dokumentasi terhadap data model tersebut



View Integrated Approach

View Integration Approach, setiap informasi kebutuhan langsung diproses menjadi data model yang disebut local data model. Setiap data model berisi diagram model dan dokumentasi terhadap model tersebut. Lalu local data model digabungkan pada tahap akhir dalam langkah Database Design menjadi global data model yang akan merepresentasikan semua user view.





06

Database Design

Database Design

Didalam langkah ini kita akan membuat rancangan database untuk nantinya digunakan dalam sistem. Database design dibagi menjadi 3 langkah, yang pertama Conceptual database design, Logical database design, dan Physical database design.

Pada saat Conceptual database design ke Logical database design, kita dapat melakukan langkah DBMS selection. Langkah ini bersifat optional, jadi boleh kita lakukan boleh juga tidak. Langkah ini dilakukan supaya DBMS yang digunakan sesuai dengan kebutuhan sistem.

Pada saat Conceptual database design ke Logical database design, kita dapat melakukan langkah DBMS selection. Langkah ini bersifat optional, jadi boleh kita lakukan boleh juga tidak. Langkah ini dilakukan supaya DBMS yang digunakan sesuai dengan kebutuhan sistem.

DATABASE DESIGN

Conceptual Database

Conceptual database design merupakan proses membangun sebuah model dari data yang nantinya akan digunakan. Data model dalam Conceptual database design dirancang dari spesifikasi kebutuhan pengguna system. Nantinya data model yang dibuat dalam Conceptual database design ini akan digunakan pada langkah berikutnya, yaitu Logical database design.



Logical Database

Logical database design merupakan proses penyempurnaan data model yang sudah dibuat dalam Conceptual database design. Data model dari Logical database design akan menjadi bentuk data yang akan digunakan dalam sistem.



DATABASE DESIGN

Physical Database

Dalam Physical database design, kita akan membuat deskripsi dari data model yang sudah dibuat dalam langkah Logical database design. Physical Database Design menjelaskan bagaimana data nantinya akan disimpan kedalam tempat penyimpanan. Desainnya akan ini disesuaikan dengan DBMS yang digunakan.



07

>>>>>

DBMS SELECTION

.....

>>>>>

DBMS COLLECTION

Pemilihan DBMS yang tepat untuk mendukung aplikasi database dapat dilakukan kapanpun sebelum menuju desain logikal asalkan terdapat cukup informasi mengenai kebutuhan sistem

4 Langkah Utama Saat Memilih DB yang Tepat

- Define Terms of Reference of Study

Tujuan dari langkah ini adalah untuk Mendeskripsikan dan mengidentifikasi Tujuan dan Ruang Lingkup Studi dan tugas yang perlu dilakukan. Karena setiap perusahaan memiliki tujuan yang berbeda, maka tujuan yang berbeda memerlukan perlakuan/pelayanan yang unik dari database. Lingkup tugas yang akan dilakukan juga penting untuk dipelajari. Baik Tujuan dan Ruang Lingkup Studi akan dibuat berdasarkan kriteria kebutuhan pengguna.



DBMS COLLECTION

- Shortlist two or three products

Setelah konsep DBMS yang akan dibutuhkan ditentukan, langkah selanjutnya dari pemilihan DBMS adalah membuat daftar beberapa produk yang sesuai dengan konsep utama yang telah ditentukan pada langkah sebelumnya. Kita juga perlu mempertimbangkan anggaran yang tersedia berdasarkan Lingkup Tugas yang akan dilakukan.

- Evaluate Product

Tujuan dari langkah ini adalah untuk melakukan analisis yang lebih dalam tentang DBMS yang telah terdaftar sebagai opsi yang memungkinkan, analisis yang lebih dalam ini terdiri dari aksesibilitas Produk, kinerja Produk itu sendiri, Keamanan dan seberapa jauh Produk memberikan keamanan, dll.

- Recommend Selection and Produce Report

Terakhir, kami akan membuat dokumentasi tentang temuan saat menganalisis DBMS yang tepat yang dibutuhkan perusahaan. Dokumentasi terdiri dari proses saat memberikan pernyataan temuan dan rekomendasi untuk Produk DBMS tertentu. Rekomendasi adalah pendapat para analis ketika memilih DBMS.





08

Application Design

Application Design

Menurut Connolly, T.M., et al. (2010), application design adalah desain user interface dan program aplikasi yang menggunakan dan memproses database.

Dua aspek application design, yaitu transaction design dan user interface design.

- Transaction Design

Transaksi sebagai tindakan, atau serangkaian tindakan yang dilakukan oleh pengguna tunggal atau program aplikasi, yang mengakses atau mengubah isi database.

Sebuah transaksi dapat terdiri dari beberapa operasi, seperti transfer uang dari satu account ke account lainnya.

Tujuan dari desain transaksi adalah untuk menetapkan dan mendokumentasikan karakteristik tingkat tinggi dari transaksi yang dibutuhkan pada database, termasuk:

- Data yang akan digunakan oleh transaksi;
- Karakteristik fungsional dari transaksi;
- Output transaksi;
- Kepentingan bagi pengguna;
- Tingkat pengembalian yang diharapkan dari penggunaan.

Application Design

Menurut Connolly, T.M., et al. (2010), application design adalah desain user interface dan program aplikasi yang menggunakan dan memproses database.

Dua aspek application design, yaitu transaction design dan user interface design.

❑ Transaction Design

Transaksi sebagai tindakan, atau serangkaian tindakan yang dilakukan oleh pengguna tunggal atau program aplikasi, yang mengakses atau mengubah isi database.

Sebuah transaksi dapat terdiri dari beberapa operasi, seperti transfer uang dari satu account ke account lainnya.

Tujuan dari desain transaksi adalah untuk menetapkan dan mendokumentasikan karakteristik tingkat tinggi dari transaksi yang dibutuhkan pada database, termasuk:

- Data yang akan digunakan oleh transaksi;
- Karakteristik fungsional dari transaksi;
- Output transaksi;
- Kepentingan bagi pengguna;
- Tingkat pengembalian yang diharapkan dari penggunaan.



Application Design

Ada tiga jenis utama transaksi:

- Retrieval transactions

Digunakan untuk mengambil data untuk ditampilkan di layar atau dalam produksi laporan. Sebagai contoh, operasi untuk mencari dan menampilkan rincian dari properti (diberi nomor properti) adalah contoh dari transaksi pengambilan.

- Update transactions

Digunakan untuk menyisipkan catatan baru, menghapus catatan lama, atau memodifikasi catatan yang ada dalam database. Sebagai contoh, operasi untuk memasukkan rincian properti baru ke dalam database adalah contoh dari sebuah transaksi update.

- Mixed transactions

Melibatkan baik pengambilan dan memperbarui data. Sebagai contoh, operasi untuk mencari dan menampilkan rincian dari properti (diberi nomor properti) dan kemudian memperbarui nilai sewa bulanan adalah contoh dari transaksi campuran.

Application Design

❑ User Interface Design Guidelines

Sebelum menerapkan suatu bentuk atau laporan, adalah penting bahwa pertama-tama merancang tata letak. Berikut panduan yang berguna (User Interface Design Guidelines) untuk diikuti saat merancang bentuk atau laporan:

- Judul yang bermakna
- Instruksi yang bisa dipahami
- Pengelompokan logis dan pengurutan bidang
- Tata letak formulir/laporan yang menarik secara visual
- Label bidang yang familier
- Terminologi dan singkatan yang konsisten
- Penggunaan warna yang konsisten
- Ruang dan batas yang terlihat untuk bidang entri data
- Gerakan kursor yang nyaman
- Koreksi kesalahan untuk karakter individu dan seluruh bidang
- Pesan kesalahan untuk nilai yang tidak dapat diterima
- Bidang opsional ditandai dengan jelas
- Pesan penjelasan untuk bidang
- Sinyal penyelesaian



09

Prototyping

Prototyping

Merupakan langkah yang optional atau boleh dilakukan boleh juga tidak dilakukan.

Dalam langkah ini kita membuat sistem database yang dapat digunakan dengan beberapa tujuan, yaitu mengenali bahwa fitur-fitur dalam sistem berjalan dengan baik atau sebaliknya, dapat menyarankan peningkatan dalam sistem atau menambahkan fitur, klarifikasi kebutuhan user, dan mengevaluasi sistem yang kita buat layak atau tidak.

Terdapat dua macam strategi prototyping yang digunakan saat ini, yaitu :

- ❑ **Requirement Prototyping**, menggunakan prototype untuk menentukan kebutuhan dari usulan sistem basis data yang diinginkan dan ketika kebutuhan itu terpenuhi maka prototype tidak akan dipakai lagi.
- ❑ **Evolutionary Prototyping**, digunakan untuk tujuan yang sama. Perbedaanya, prototype tidak akan dibuang tetapi dengan pengembangan lanjutan menjadi aplikasi basis data yang digunakan.



10

Implementation

Implementation

Menurut Connolly, T.M., et al. (2010), implementation adalah proses realisasi fisik dari database dan desain aplikasi. Setelah tahapan desain dirasa sudah selesai selanjutnya masuk ke tahapan implementasi database di program aplikasi.

Implementasi database dicapai dengan menggunakan Data Definition Language (DDL) dari DBMS yang dipilih atau Graphical User Interface (GUI), yang menyediakan fungsi yang sama sambil menyembunyikan laporan DDL tingkat rendah. Laporan DDL digunakan untuk membuat struktur database dan file database kosong. Setiap user view tertentu juga diimplementasikan pada tahap ini

11

Data Conversion and Loading

Data Conversion and Loading

Data Conversion and Loading adalah proses transfer data yang sudah ada kedalam basis data dan convert aplikasi yang sudah ada agar berjalan di basis data yang baru.

Menurut Connolly dan Begg (2010:284), proses ini membutuhkan spesifikasi dari Database sumber menuju ke Database tujuan, dan melakukan konversi data ke format yang pada File basis data yang baru secara otomatis



12



Testing



Testing

Proses menjalankan sistem basis data dengan tujuan menemukan error

Evaluasi yang ideal pada saat melakukan Testing adalah:

- ❑ Learnability

Waktu yang dihabiskan oleh pengguna untuk dapat menyesuaikan diri dalam menggunakan aplikasi ini

- ❑ Robustness

Toleransi dari Error pada system

- ❑ Adaptability Kedekatan sistem pada model pekerjaan yang sesungguhnya.

- ❑ Performance

Performa aplikasi yang memiliki sistem respon yang sesuai dengan pekerjaan pengguna.

- ❑ Recoverability

Kualitas sistem saat terjadi error pada pengguna saat melakukan proses recovery



13

Operational Maintenance

Operational Maintenance

Operational Maintenance merupakan proses terakhir dimana penggunaan database dimonitor dan terjadinya proses pemantauan dan pemeliharaan sistem basis data berikut instalasi.

Tahap operational maintenance memiliki dua tahap yaitu:

- ❖ Memantau performa dari sistem. Jika tingkat performa berada di bawah dari seharusnya, maka penyusunan ulang sistem basis data perlu dilakukan.
- ❖ Memantau dan melakukan Upgrade pada sistem basis data jika diperlukan.



14

CASE Tools



CASE Tools

- Dalam software engineering telah dikenal banyak tools (computer-base system) yang dikenal dengan Computer-Aided Software Engineering (CASE).
- CASE merupakan suatu teknik yang digunakan untuk membantu satu atau beberapa fase dalam life-cycle software, termasuk fase analisis, desain, implementasi dan maintenance dari software tersebut

CASE Tools

diklasifikasikan sebagai berikut:

Upper Case

CASE tools yang didesain untuk mendukung perencanaan, identifikasi, dan seleksi proyek (permulaan dari perencanaan proyek), tepatnya pada fase analisis dan desain dari suatu system development life cycle (SDLC). Contoh CASE tools : Cradle, PRO-IV, Workbench, ProKit*Workbrnch.

Lower Case

CASE tools yang didesain untuk mendukung tahap implementasi dan maintenance dari SDLC. Tools yang termasuk kelas ini adalah jenis Code generators. Contoh CASE tools: Level/I-User Sensitive CASE, PRO-IV application Development.

CASE TOOLS

Integrated Case

CASE tools yang dirancang untuk mendukung aktifitas-aktifitas yang terjadi pada beberapa fase dari SDLC. Mengkombinasikan Upper dan Lower CASE menjadi satu. Tools yang termasuk kelas ini adalah jenis Project management tools.

Contoh CASE tools: Rational Rose, Poseidon, ArgoUML, Catalyze, inStep, Juggler, PRINCE

Manfaat CASE Tools

CASE Tools memberikan manfaat berikut yang meningkatkan produktivitas :

- ❑ **Standards**, CASE tools membantu menegakkan standar pada proyek perangkat lunak atau di seluruh organisasi. Mereka mendorong produksi komponen pengujian standar yang dapat digunakan kembali, sehingga menyederhanakan pemeliharaan dan meningkatkan produktivitas.
- ❑ **Integration**, CASE tools menyimpan semua informasi yang dihasilkan dalam repository, atau data dictionary. Dengan demikian, data yang dikumpulkan harus dapat disimpan selama semua tahap siklus hidup pengembangan sistem basis data. Data kemudian dapat dihubungkan bersama untuk memastikan bahwa semua bagian dari sistem terintegrasi.
- ❑ **Support for standard methods**, Teknik terstruktur memanfaatkan diagram secara signifikan, yang sulit untuk digambar dan dipelihara secara manual. CASE Tools menyederhanakan proses ini, menghasilkan dokumentasi yang benar dan lebih terkini.



Manfaat CASE Tools



- ❑ **Consistency**, Karena semua informasi dalam kamus data saling terkait, CASE Tools dapat memeriksa konsistensinya.
- ❑ **Automation**, Beberapa CASE Tools dapat secara otomatis mengubah bagian dari spesifikasi desain menjadi kode yang dapat dieksekusi. Ini mengurangi pekerjaan yang diperlukan untuk menghasilkan sistem yang diimplementasikan, dan dapat menghilangkan kesalahan yang muncul selama proses pengkodean



«««««

15

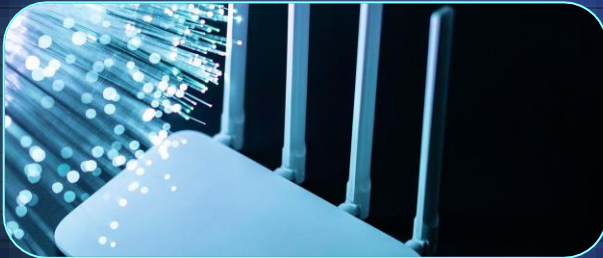
Data Administration and Database Administration

»»»»»

Data Administration and Database Administration

Pengelolaan sumber daya data, yang meliputi basis database administration planning, development, maintenance of standards, kebijakan dan prosedur, dan desain database konseptual dan logis

Data Administration




Database Administration

Pengelolaan realisasi fisik sistem basis data, yang meliputi desain dan implementasi basis data fisik, mengatur kontrol keamanan dan integritas, memantau kinerja sistem, dan mengatur ulang database, jika diperlukan

Perbedaan Data Administration and Database Administration

Data administration	Database administration
Terlibat dalam IS planning strategis	Mengevaluasi DBMS baru
Menentukan tujuan jangka panjang	Jalankan rencana untuk mencapai tujuan
Menegakkan standar, kebijakan, dan prosedur	Menegakkan standar, kebijakan, dan prosedur
Menentukan persyaratan data	Menerapkan persyaratan data
Mengembangkan desain database konseptual dan logis	Mengembangkan desain database logis dan fisik
Mengembangkan dan memelihara model data perusahaan	Mengimplementasikan desain basis data fisik
Mengkoordinasikan pengembangan sistem	Memantau dan mengontrol database
Managerial orientation	Technical orientation
DBMS independent	



Thanks!

Credits: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

Please keep this slide for attribution

