

UTS Basis Data Semester 3

Aldi Maulana Iqbal – 20210801222

1. Kelemahan Traditional File System

- a. Timbulnya data rangkap.

Karena file dan program dikembangkan oleh pemrogram yang berbeda, beberapa informasi dalam beberapa file dapat terduplikasi.
- b. Data yang tidak konsisten.

Data tidak konsisten dalam file yang dihubungkan.
- c. Kesulitan mengakses data.

Munculnya permintaan baru yang tidak diantisipasi pada saat pengembangan aplikasi.
- d. Data terisolir.

Karena data tersebar di beberapa file dan mungkin dalam format yang berbeda, sulit untuk menulis program baru untuk mendapatkan data yang sesuai.
- e. Masalah keamanan.

Pengguna tidak dapat mengakses semua data.
- f. Ketergantungan data.

Jika aplikasi berubah atau ada bug, pengguna akan kehilangan akses ke data.
- g. Tidak mendukung sharing data.

Tidak dapat berbagi elemen database dengan pengguna lain.
- h. Biaya.

Biaya pengembangan yang tinggi tidak memungkinkan untuk menanggapi perubahan kebutuhan.
- i. Object Oriented.

Program menjadi lebih besar dalam ukuran, sehingga proses eksekusi memakan waktu lama.
- j. Record File.

Membutuhkan waktu lama karena sulitnya mengakses data yang akan dicari.

2. Pentingnya Perancangan Basis Data

Basis data yang dirancang dengan baik memfasilitasi pengelolaan data dan menghasilkan informasi yang berharga. Sebaliknya, jika basis data dirancang dengan buruk, itu menjadi sumber redundansi data yang tidak terkendali dan dapat menyebabkan kesalahan yang mengarah pada pengambilan keputusan yang buruk.

3. Redudansi Terkontrol dan Redudansi Tidak Terkontrol

- a. Redudansi Terkontrol

Penggunaan atribut bersama-sama membuat database relasional bekerja dengan baik.

Contoh: Membuat beberapa tabel untuk menyimpan berbagai informasi tentang perusahaan, seperti tabel yang menyimpan data departemen, data karyawan, dan data gaji.

b. Redudansi Tidak Terkontrol

Redundansi yang tidak terkendali dapat menyebabkan masalah seperti inkonsistensi data dan anomali data.

Contoh: Dalam sistem identitas mahasiswa, input menggunakan identitas saja.

4. DML Procedural dan non-Procedural

a. DML Procedural.

Untuk bahasa pemrograman tingkat tinggi. Digunakan oleh pengguna untuk memberikan spesifikasi data kepada sistem tentang apa yang dibutuhkan dan bagaimana mendapatkannya.

Contoh: Java, PHP, C

b. DML non-Procedural.

Untuk bahasa pemrograman deklaratif. Digunakan dengan cara pengguna memberikan spesifikasi data yang diinginkan tanpa mengetahui cara mendapatkan data tersebut.

Contoh: SQL, QBE

5. Arsitektur DBMS

a. Centralized DBMS Architecture.

Sistem basis data yang berjalan pada satu sistem komputer dan bukan sistem komputer lain. Pengguna terhubung ke komputer pusat melalui terminal.

b. Basic Client/Server Architecture.

Sistem basis data yang memisahkan program pengguna dari program basis data pada sistem yang berbeda. Pengguna terhubung ke komputer pusat data yang disebut sistem server melalui sistem antarmuka pengguna pada komputer pribadi.

c. Two-Tier Client/Server Architecture untuk DBMS.

Arsitektur ini membagi proses loading menjadi dua bagian. Logika aplikasi utama berjalan pada klien, dan klien biasanya mengirimkan permintaan ke server basis data sebagai media penyimpanan data dalam bentuk sintaks SQL.

d. Three-Tier and n-Tier Architecture untuk Aplikasi Web.

Arsitektur ini membagi proses loading menjadi tiga bagian yaitu:

- 1) Komputer klien menjalankan logika Graphical User Interface (GUI).
- 2) Aplikasi server menjalankan logika bisnis.
- 3) Database legacy application.

6. Jika Anda merancang sistem berbasis web untuk membuat reservasi penerbangan dan menjual tiket pesawat, arsitektur DBMS mana yang akan Anda pilih? Jelaskan

n-Tier/Multi-Tier Architecture karena memerlukan database dengan banyak bagian untuk memesan tiket. Data yang diperlukan adalah identitas penumpang, nomor kursi, nomor penerbangan, dll.

Normalisasi Database “rs_ueu”