



BELAJAR BASIS DATA RELASIONAL MYSQL

MELALUI STUDI KASUS



Andharini Dwi Cahyani, S.Kom., M.Kom., PhD

BELAJAR BASIS DATA RELASIONAL MYSQL MELALUI STUDI KASUS

Andharini Dwi Cahyani, S.Kom., M.Kom., PhD



BELAJAR BASIS DATA RELASIONAL MYSQL MELALUI STUDI KASUS

Penulis:

Andharini Dwi Cahyani, S.Kom., M.Kom., PhD

Editor:

Erik Santoso

Layouter :

Tim Kreatif PRCI

Cover:

Rusli

Cetakan Pertama : Agustus 2023

Hak Cipta 2022, pada Penulis. Diterbitkan pertama kali oleh:

**Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT**

Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org

E-mail : rumahcemerlangindonesia@gmail.com

Copyright © 2023 by Perkumpulan Rumah Cemerlang Indonesia
All Right Reserved

- Cet. I – : Perkumpulan Rumah Cemerlang Indonesia, 2023
; 18 x 25 cm
ISBN : -

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

Kata Pengantar

Puji syukur kami panjatkan kepada Allah SWT yang telah memberikan rahmat dan karuniaNya sehingga kami dapat menyelesaikan penulisan buku ini.

Buku ini menjelaskan konsep dasar basis data relasional yang relevan dengan mata kuliah Basis Data dan dapat digunakan sebagai referensi bagi dosen dan mahasiswa yang menekuni bidang IT. Semoga buku ini dapat memudahkan pembaca memahami konsep basis data relasional melalui studi kasus yang diangkat.

Penulis menyampaikan terimakasih sebanyak-banyaknya kepada seluruh pihak yang membantu penulisan buku ini. Penulis menyadari bahwa buku ini masih belum sempurna, karenanya saran dan kritik yang membangun dari semua pihak diharapkan demi penyempurnaan buku ini.

Sidoarjo, Agustus 2023

Daftar Isi

1	Konsep DBMS dan Penyimpanan Data Secara Tradisional	1
1.1	<i>Basic File System</i>	1
1.2	Basis Data	3
1.3	<i>Database Management System</i>	4
1.4	Tujuan <i>DBMS</i>	6
1.5	Bahasa Basis Data	8
1.6	Basis Data Relasional	11
1.6.1	Komponen Penyusun Basis Data.	12
1.6.2	<i>Relational Keys</i>	13
2	Normalisasi Data.....	18
2.1	Pengertian.....	19
2.2	Ketergantungan dalam Normalisasi	20
2.3	Tahapan Normalisasi.....	22
2.4	Latihan Soal	28
3	<i>Constraint</i>	33
3.1	Pengertian <i>Constraint</i>	33
3.2	<i>Integrity Constraint</i>	33
3.3	<i>Constraint</i> pada Perancangan Model Hubungan Basis Data	35
3.3.1	<i>Cardinality Constraint</i> :.....	35
3.3.2	<i>Participation Constraint</i> :.....	37
3.4	Perancangan Basis Data.....	38
3.4.1	<i>ERD</i>	38
3.4.2	<i>CDM</i>	39
3.4.3	<i>PDM</i>	40
3.4.4	Contoh Transformasi <i>ERD</i> ke <i>CDM</i> dan <i>PDM</i>	41
3.5	Latihan soal	43
4	Studi Kasus: Sistem Penjualan Barang di Toko.....	47
4.1	Proses Normalisasi.....	48
4.1.1	Tidak Normal (<i>Unnormalized</i>)	48
4.1.2	Normal Satu (1NF)	49
4.1.3	Normal Dua (2NF)	50
4.1.4	Normal Tiga (3NF)	51

4.2	Perancangan <i>Entity Relationship Diagram</i>	52
4.3	<i>Conceptual Data Model</i>	54
4.4	<i>Physical Data Model</i>	54
4.5	Penerapan Basis Data di <i>MySQL</i>	55
5	Studi kasus perpustakaan.....	66
5.1	Proses Normalisasi.....	70
5.1.1	Tidak Normal (<i>Unnormalized</i>)	70
5.1.2	Normal Satu (1NF)	71
5.1.3	Normal Dua (2NF)	72
5.1.4	Normal Tiga (3NF)	74
5.2	Perancangan Dalam Basis Data	76
6	Penerapan GROUP BY pada Kedua Studi Kasus.....	90
6.1	Penjelasan Group By.....	91
6.2	Contoh Penggunaan GROUP BY dalam SQL.....	91
7	Penerapan JOIN pada Studi Kasus Sistem Penjualan Barang.....	99
7.1	Penjelasan JOIN	99
7.2	Jenis-jenis JOIN.....	99
7.3	Contoh Penerapan JOIN Pada Studi Kasus Sistem Penjualan Barang	101
8	Penerapan JOIN pada Studi Kasus Sistem Perpustakaan	105
9	Penerapan <i>Nested Query</i> pada Studi Kasus Sistem Penjualan Barang	115
10	Penerapan <i>Nested Query</i> pada Studi Kasus Sistem Perpustakaan.....	122
11	Penerapan CREATE PROCEDURE untuk Studi Kasus Sistem Penjualan Barang.....	127
11.1	Penjelasan PROCEDURE	127
11.2	Contoh Penerapan PROCEDURE pada Studi Kasus Sistem Penjualan Barang	127
12	Penerapan CREATE PROCEDURE untuk Studi Kasus Sistem Perpustakaan.....	133
13	Penerapan CREATE TRIGGER untuk Studi Kasus Sistem Penjualan Barang.....	137
13.1	Penjelasan TRIGGER.....	137
13.2	Penerapan CREATE TRIGGER untuk Studi Kasus Sistem Penjualan Barang	138

14	Penerapan CREATE TRIGGER untuk Studi Kasus Sistem Perpustakaan	142
----	--	-----

Daftar Gambar

Gambar 1.1 Komponen Penyusun Tabel pada Basis Data	12
Gambar 2.1 Tahapan Normalisasi	22
Gambar 3.1 Relasi <i>One to One</i>	35
Gambar 3.2 Relasi <i>One to Many</i>	36
Gambar 3.3 Relasi <i>Many to Many</i>	36
Gambar 3.4 <i>ERD</i> Sistem Penjualan Motor	42
Gambar 3.5 <i>CDM</i> Sistem Penjualan Motor	42
Gambar 3.6 <i>PDM</i> Sistem Penjualan Motor	43
Gambar 4.1 Nota Tranksaksi Pembelian Barang	48
Gambar 4.2 <i>ERD</i> Sistem Penjualan Barang	53
Gambar 4.3 <i>CDM</i> Sistem Penjualan Barang	54
Gambar 4.4 <i>PDM</i> Sistem Penjualan Barang	54
Gambar 4.5 <i>Web Download XAMPP</i>	56
Gambar 4.6 <i>File Installer XAMPP</i>	56
Gambar 4.7 Langkah Pertama <i>Install XAMPP</i>	56
Gambar 4.8 Fitur-fitur <i>XAMPP</i>	57
Gambar 4.9 Lokasi <i>Install XAMPP</i>	57
Gambar 4.10 Proses <i>Install XAMPP</i>	57
Gambar 4.11 <i>Control Panel XAMPP</i>	58
Gambar 4.12 Mengakses <i>phpMyAdmin</i>	58
Gambar 4.13 Tampilan <i>phpMyAdmin</i>	59
Gambar 4.14 Membuat Basis Data Sistem Penjualan Barang	59
Gambar 4.15 Tab <i>SQL</i> Untuk Membuat <i>Query</i> di Basis Data	60
Gambar 4.16 Notifikasi Tabel Berhasil Dibuat	60
Gambar 4.17 Tabel Barang Sudah Dibuat di Basis Data Sistem Penjualan Barang	61
Gambar 4.18 Seluruh Tabel Selesai Dibuat	61
Gambar 4.19 Relasi Tabel Berhasil Dibuat	62
Gambar 5.1 Keterangan Data Buku	67
Gambar 5.2 Kartu Anggota Perpustakaan	67
Gambar 5.3 Data Petugas Perpustakaan	67
Gambar 5.4 Struk Peminjaman dari Perpustakaan	68
Gambar 5.5 Struk Pengembalian Perpustakaan	69
Gambar 5.6 <i>ERD</i> Studi Kasus Perpustakaan	80
Gambar 5.7 <i>CDM</i> Studi Kasus Perpustakaan	81

Gambar 5.8 <i>PDM</i> Studi Kasus Perpustakaan	82
Gambar 6.1 Tabel Transaksi.....	92
Gambar 6.2 Contoh Penggunaan GROUP BY pada Tabel Transaksi ...	92
Gambar 6.3 ORDER BY Membuat Urutan transaksi.total dari Terbesar ke Terkecil.....	93
Gambar 6.4 Membuat GROUP BY pada Tabel Transaksi dan Kasir	93
Gambar 6.5 VIEW transaksi_tiapkasir di <i>Database</i> Sistem Penjualan Barang	95
Gambar 6.6 Hasil <i>Query</i> VIEW transaksi_tiapkasir.....	95
Gambar 6.7 Hasil VIEW totalPenjualanBarang	97
Gambar 6.8 Hasil Pemanggilan VIEW tahunTerbitBuku.....	98
Gambar 7.1 INNER JOIN	99
Gambar 7.2 LEFT JOIN.....	100
Gambar 7.3 RIGHT JOIN.....	100
Gambar 7.4 FULL JOIN.....	100
Gambar 7.5 Contoh Hasil Penerapan RIGHT JOIN	102
Gambar 7.6 Hasil Penerapan INNER JOIN pada Sistem Penjualan Barang.....	104
Gambar 8.1 Hasil INNER JOIN untuk Tabel Detail Peminjaman.....	107
Gambar 8.2 Hasil INNER JOIN pada Tabel Pengembalian Buku dengan Tabel Petugas	108
Gambar 8.3 Hasil RIGHT JOIN pada Tabel Pengembalian Buku dan Tabel Petugas	109
Gambar 8.4 Hasil Penerapan LEFT JOIN pada Tabel Pengembalian Buku dan Tabel Petugas	110
Gambar 8.5 <i>Error</i> Karena <i>MySQL</i> Tidak Mendukung FULL JOIN.....	111
Gambar 8.6 FULL JOIN pada <i>MySQL</i> Menggunakan Istilah CROSS JOIN	112
Gambar 8.7 Hasil Penerapan CROSS JOIN.....	112
Gambar 8.8 Penjelasan CROSS JOIN dari w3school	113
Gambar 8.9 Hasil Penggunaan UNION untuk Menggabungkan LEFT JOIN dan RIGHT JOIN.....	114
Gambar 9.1 Contoh Hasil <i>Dependent Nested Query</i>	116
Gambar 9.2 Contoh Hasil <i>Independent Nested Query</i>	117
Gambar 9.3 Jumlah Transaksi Dibawah Rata-rata	119

Gambar 9.4 Hasil Nested Query Dengan Kriteria Nama Barang Tertentu	121
Gambar 10.1 Hasil <i>Nested Query</i> Pada detail_pengembalian_buku Dengan Kriteria Tahun Terbit Buku	125
Gambar 11.1 Hasil Pemanggilan PROCEDURE GetDetailTransbyNoBon	129
Gambar 11.2 Hasil Pemanggilan PROCEDURE SetNewBarang.....	131
Gambar 12.1 Hasil Pemanggil PROCEDURE GetDetailPeminjaman	135
Gambar 12.2 Hasil PROCEDURE GetDetailBuku.....	135
Gambar 12.3 <i>Record</i> Anggota Baru Berhasil Ditambahkan	136
Gambar 13.1 Hasil TRIGGER Kembalian.....	139
Gambar 13.2 Hasil TRIGGER inputDate	141
Gambar 14.1 Hasil Penggunaan TRIGGER Pada Tabel pengembalian_buku.....	147
Gambar 14.2 Hasil TRIGGER inputWaktuPeminjaman.....	149

Daftar Tabel

Tabel 1.1 Tabel Mahasiswa.....	16
Tabel 2.1 Seluruh Data dari Studi Kasus Nilai Siswa.....	23
Tabel 2.2 Bentuk <i>Unnormalized</i> Data Nilai Siswa.....	24
Tabel 2.3 Bentuk Normal Satu Data Nilai Siswa.....	24
Tabel 2.4 Ketergantungan Fungsional di Basis Data Siswa.....	25
Tabel 2.5 Tabel Siswa.....	26
Tabel 2.6 Tabel Matapelajaran.....	26
Tabel 2.7 Tabel Nilai.....	26
Tabel 2.8 Tabel Siswa.....	27
Tabel 2.9 Tabel Guru.....	27
Tabel 2.10 Tabel Mata Pelajaran.....	27
Tabel 2.11 Tabel Nilai.....	28
Tabel 2.12 Tabel Mahasiswa untuk Latihan Soal.....	29
Tabel 2.13 Tabel Mahasiswa.....	31
Tabel 2.14 Tabel Mahasiswa.....	31
Tabel 2.15 Tabel Matakuliah.....	32
Tabel 2.16 Tabel Mahasiswa.....	32
Tabel 2.17 Tabel Matakuliah.....	32
Tabel 2.18 Tabel Pendaftaran Matakuliah.....	32
Tabel 3.1 Tabel <i>Customer</i>	44
Tabel 3.2 Tabel <i>Order</i>	44
Tabel 4.1 Seluruh Data dari Nota Transaksi.....	49
Tabel 4.2 Bentuk Satu Normalisasi Database Sistem Penjualan Barang.....	49
Tabel 4.3 Tabel Barang pada Normalisasi 2NF.....	50
Tabel 4.4 Tabel Kasir pada Normalisasi 2NF.....	50
Tabel 4.5 Tabel Transaksi pada Normalisasi 2NF.....	50
Tabel 4.6 Tabel Barang Pada Normalisasi 3NF.....	51
Tabel 4.7 Tabel Kasir Pada Normalisasi 3NF.....	51
Tabel 4.8 Tabel Detail Transaksi Pada Normalisasi 3NF.....	51
Tabel 4.9 Tabel Transaksi pada Normalisasi 3NF.....	52
Tabel 5.1 Daftar Atribut Studi Kasus Perpustakaan.....	70
Tabel 5.2 Tahap Normalisasi Bentuk 1NF.....	71
Tabel 5.3 Tabel Buku Pada Tahap Normalisasi 2NF.....	72
Tabel 5.4 Tabel Anggota Pada Tahap Normalisasi 2NF.....	73

Tabel 5.5 Tabel Petugas Pada Tahap Normalisasi 2NF	73
Tabel 5.6 Tabel Peminjaman Pada Tahap Normalisasi 2NF	73
Tabel 5.7 Tabel Pengembalian Pada Tahap Normalisasi 2NF	74
Tabel 5.8 Tabel Buku pada Tahap Normalisasi 3NF	74
Tabel 5.9 Tabel Anggota Perpustakaan Pada Tahap Normalisasi 3NF	75
Tabel 5.10 Tabel Petugas Perpustakaan Pada Tahap Normalisasi 3NF	75
Tabel 5.11 Tabel Peminjaman Pada Tahap Normalisasi 3NF	75
Tabel 5.12 Tabel Detail Peminjaman Pada Tahap Normlaisasi 3NF ..	75
Tabel 5.13 Tabel Pengembalian Pada Tahap Normalisasi 3NF	75
Tabel 5.14 Tabel Detail Pengembalian Pada Tahap Normalisasi 3NF	76
Tabel 5.15 Tabel Anggota Perpustakaan Pada Tahap Normalisasi 3NF	77
Tabel 5.16 Tabel Petugas Perpustakaan Pada Tahap Normalisasi 3NF	77
Tabel 5.17 Tabel Peminjaman Pada Tahap Normalisasi 3NF	77
Tabel 5.18 Tabel Detail Peminjaman Pada Tahap Normalisasi 3NF ..	78
Tabel 5.19 Tabel Pengembalian Pada Tahap Normalisasi 3NF	78
Tabel 5.20 Tabel Detail Pengembalian Pada Tahap Normalisasi 3NF	78

Daftar Kode Program

Kode Program 1.1 Perintah Membuat Tabel Baru	9
Kode Program 1.2 Perintah Menambahkan Kolom Baru Pada Suatu Tabel	9
Kode Program 1.3 Perintah Menghapus Tabel	9
Kode Program 1.4 Perintah Menambahkan Data Pada Suatu Tabel	10
Kode Program 1.5 Perintah Mengubah Data Pada Suatu Tabel	10
Kode Program 1.6 Perintah Mengambil Data Dari Suatu Tabel	10
Kode Program 1.7 Perintah Menghapus Data Dari Suatu Tabel	10
Kode Program 1.8 Perintah Memberikan Izin Pada Suatu Peran	11
Kode Program 1.9 Perintah Mencabut Izin Pada Suatu Peran	11
Kode Program 4.1 Membuat Tabel Barang	60
Kode Program 4.2 Membuat Relasi antara Tabel Transaksi dan Kasir	62
Kode Program 4.3 Memasukkan Data Ke Tabel Kasir	62
Kode Program 4.4 Seluruh Query Untuk Menyusun Basis Data Sistem Penjualan Barang	65
Kode Program 5.1 Query Penyusun Database Studi Kasus Perpustakaan	89
Kode Program 6.1 Syntax Group By	91
Kode Program 6.2 Contoh GROUP BY pada Tabel Transaksi	92
Kode Program 6.3 Menggunakan ORDER BY untuk Mengurutkan Data	93
Kode Program 6.4 Contoh Penggunaan GROUP BY	93
Kode Program 6.5 Membuat VIEW transaksi_tiapkasir	94
Kode Program 6.6 Memanggil VIEW transaksi_tiapkasir	95
Kode Program 6.7 Membuat VIEW totalPenjualanBarang	97
Kode Program 6.8 Membuat VIEW tahunTerbitBuku	98
Kode Program 6.9 Memanggil VIEW tahunTerbitBuku	98
Kode Program 7.1 Menambah Data Kasir Baru	101
Kode Program 7.2 Contoh Penerapan JOIN pada Tabel Transaksi dan Tabel Kasir	101
Kode Program 7.3 Penerapan INNER JOIN pada Sistem Penjualan Barang	103

Kode Program 8.1 Penerapan INNER JOIN Pada Tabel Detail Peminjaman.....	106
Kode Program 8.2 Contoh Penerapan INNER JOIN pada Studi Kasus Sistem Perpustakaan	107
Kode Program 8.3 Contoh Penerapan RIGHT JOIN pada Studi Kasus Sistem Perpustakaan	108
Kode Program 8.4 Contoh Penerapan RIGHT JOIN pada Studi Kasus Perpustakaan	110
Kode Program 8.5 Contoh Penerapan CROSS JOIN pada Studi Kasus Sistem Perpustakaan	112
Kode Program 8.6 UNION untuk Menggabungkan LEFT JOIN dan RIGHT JOIN	113
Kode Program 9.1 Contoh Dependent Nested Query	115
Kode Program 9.2 Contoh Independent Nested Query	116
Kode Program 9.3 Nested Query Untuk Menentukan Total Transaksi Dibawah Rata-rata	119
Kode Program 9.4 Nested Query Untuk Menampilkan Transaksi Dengan Nama Barang Tertentu.....	120
Kode Program 10.1 Nested Query Pada detail_peminjaman_buku Dengan Kriteria Tahun Terbit Buku	124
Kode Program 10.2 Nested Query Pada detail_pengembalian_buku Dengan Kriteria Tahun Terbit Buku.....	125
Kode Program 11.1 Syntax Membuat PROCEDURE	127
Kode Program 11.2 Membuat PROCEDURE Untuk Melihat Detail Transaksi	128
Kode Program 11.3 Memanggil PROCEDURE GetDetailTransbyNoBon	128
Kode Program 11.4 Membuat PROCEDURE SetNewBarang	130
Kode Program 11.5 Memanggil PROCEDURE SetNewBarang	130
Kode Program 11.6 Membuat PROCEDURE GetKinerjaKasir	131
Kode Program 12.1 Membuat Procedure GetDetailPeminjaman	134
Kode Program 12.2 Memanggil Procedure GetDetailPeminjaman	135
Kode Program 12.3 Membuat PROCEDURE GetDetailBuku	135

Kode Program 12.4 Memanggil PROCEDURE GetDetailBuku	135
Kode Program 12.5 PROCEDURE SetNewAnggota	136
Kode Program 12.6 Memanggil PROCEDURE SetNewAnggota ...	136
Kode Program 13.1 Syntax Membuat TRIGGER	137
Kode Program 13.2 Membuat TRIGGER Menghitung Kembalian	138
Kode Program 13.3 Memasukkan Data Baru di Tabel Transaksi	139
Kode Program 13.4 Membuat TRIGGER Untuk Memasukkan Data Waktu	140
Kode Program 13.5 Input Data Pada Tabel Transaksi	141
Kode Program 14.1 TRIGGER Untuk Menghitung Denda	144
Kode Program 14.2 Memasukkan Data Pada Tabel peminjaman_buku	146
Kode Program 14.3 Memasukkan Data Pada Tabel pengembalian_buku	147
Kode Program 14.4 Membuat TRIGGER inputWaktuPeminjaman	148
Kode Program 14.5 Input Data Pada Tabel peminjaman_buku	148

1 Konsep DBMS dan Penyimpanan Data Secara Tradisional

1.1 Basic File System

Basic File System (BFS) adalah sebuah sistem *file* yang digunakan untuk mengatur dan menyimpan data dalam sebuah komputer atau perangkat penyimpanan lainnya. *BFS* menyediakan metode untuk mengorganisir, menyimpan, mengakses, dan mengelola *file* dan direktori. *BFS* biasanya digunakan dalam sistem operasi untuk mengatur direktori di penyimpanan seperti *hard drive*, *flash drive* dan sistem berkas *online*. Contoh dari *BFS* termasuk *File Allocation Table*, *New Technology File System* dan *Fourth Extended File System*.

Sistem basis data adalah hasil pengembangan dari *BSF* yang berfungsi untuk menyimpan data dalam waktu yang relatif lama dan memungkinkan penyimpanan data dalam jumlah yang besar. Namun, meskipun *BSF* dapat menampung banyak data, mereka tidak dapat menjamin keamanan data jika tak mendapat dukungan serta tidak efisien dalam mengakses data yang terletak pada beberapa *file* yang tidak dapat diketahui. Selain itu, *BSF* tidak memiliki dukungan langsung untuk perintah pengambilan data dalam *file* dan hanya mampu memberikan dukungan skema terbatas melalui penyusunan struktur direktori. Ada risiko bagi pengguna untuk kehilangan data yang tidak ter-*backup*. Selain itu, *BSF* tidak memuaskan karena tidak dapat mencegah situasi di mana dua pengguna atau proses mengakses *file* yang sama pada waktu bersamaan, mengakibatkan perubahan yang dibuat oleh salah satu *user* mungkin tidak terlihat pada *file*.

Menyimpan data dalam bentuk *file* sistem operasi memiliki beberapa kelemahan, terutama jika digunakan untuk perusahaan besar dengan skala operasional yang besar dan kompleks. Beberapa kelemahan utama termasuk:

1. Tidak efisien untuk skala besar

Saat perusahaan tumbuh, jumlah data yang harus disimpan dan dikelola juga meningkat. Penggunaan *file* sistem operasi tradisional tidak efisien untuk menangani volume data yang besar. Proses akses dan pencarian data mungkin menjadi lambat dan tidak optimal.

2. Sulit untuk pengelolaan dan organisasi

Dalam sistem *file*, data disimpan dalam hierarki *folder* dan *file*. Manajemen data dalam bentuk struktur direktori ini bisa menjadi rumit, terutama ketika datanya sangat beragam dan tersebar di berbagai bagian sistem.

3. Kurangnya keamanan

File sistem operasi biasanya tidak menyediakan tingkat keamanan yang tinggi untuk melindungi data dari akses yang tidak sah atau upaya pencurian data. Kredensial akses dan izin pengguna dapat mudah ditembus oleh pihak yang tidak berwenang.

4. Masalah konsistensi

Dalam lingkungan dengan banyak pengguna yang mengakses dan memodifikasi data secara bersamaan, ada risiko kehilangan konsistensi data. Misalnya, jika beberapa pengguna menyimpan data yang berbeda dengan nama yang sama di lokasi yang berbeda, bisa sulit untuk mengintegrasikan informasi tersebut.

5. Tidak mendukung kemudahan berbagi

File sistem tradisional cenderung tidak mendukung berbagi data dengan mudah di seluruh organisasi. Kolaborasi dan akses bersama bisa menjadi sulit dan membatasi efisiensi tim.

6. Tidak cocok untuk data terstruktur dan terhubung

Beberapa data dalam perusahaan berhubungan dengan data lainnya, seperti dalam basis data relasional. *File* sistem operasi biasanya tidak mendukung struktur data ini dan sulit untuk mencari dan memperbarui data yang saling terkait.

7. Tidak tahan terhadap kegagalan

Dalam beberapa kasus, jika ada kegagalan pada media penyimpanan fisik, data yang disimpan dalam *file* sistem bisa hilang atau rusak, terutama jika tidak ada sistem backup yang baik.

Oleh karena itu, perusahaan besar lebih sering memilih untuk menggunakan basis data relasional atau sistem manajemen

basis data (*DBMS*) yang canggih untuk mengatasi kelemahan-kelemahan ini. Basis data menyediakan solusi yang lebih efisien, aman, dan dapat diandalkan untuk mengelola dan menyimpan data perusahaan dengan cara yang lebih terstruktur, terhubung, dan terkendali.

1.2 Basis Data

Basis data (*Database*) adalah kumpulan informasi yang terstruktur dan terorganisir dengan cara tertentu, yang disimpan dalam media penyimpanan komputer dan dapat diakses serta dikelola secara elektronik. Basis data menyimpan data dalam format yang terstruktur, memungkinkan pengguna atau aplikasi untuk dengan mudah menambahkan, mengakses, memodifikasi, dan menghapus data.

Tujuan utama dari basis data adalah untuk menyediakan mekanisme efisien dalam pengelolaan dan pengolahan data. Dengan menggunakan basis data, organisasi dapat mengumpulkan, menyimpan, dan mengelola data dengan cara yang lebih terstruktur dan terpusat, yang pada gilirannya membantu meningkatkan efisiensi operasional, kemudahan akses, dan akurasi informasi.

Basis data umumnya terdiri dari dua komponen utama:

1. Sistem Manajemen Basis Data (*Database Management System - DBMS*)

Ini adalah perangkat lunak yang berfungsi untuk mengelola dan menyediakan akses ke basis data. *DBMS* bertanggung jawab untuk memfasilitasi operasi data, termasuk pengelolaan penambahan, pengambilan, pembaruan, dan penghapusan data.

2. Skema Basis Data

Skema adalah struktur atau format data yang didefinisikan secara matematis, yang menentukan bagaimana data akan disimpan, diorganisasi, dan berinteraksi dalam basis data. Skema basis data mendefinisikan tabel, kolom, kunci, hubungan antartabel, dan aturan integritas data.

Beberapa keuntungan menggunakan basis data meliputi:

- **Konsistensi data**
Basis data memastikan data konsisten dan terintegrasi, menghindari duplikasi dan inkonsistensi informasi.
- **Efisiensi penyimpanan dan akses data**
Basis data dirancang untuk memberikan akses data yang cepat dan efisien melalui indeks dan optimisasi kueri.
- **Keamanan data**
Basis data menyediakan mekanisme untuk mengatur hak akses dan keamanan data, membatasi akses ke informasi yang sensitif.
- **Keteraturan dan kelengkapan data**
Basis data membantu menjaga integritas dan keteraturan data melalui aturan dan batasan yang telah ditetapkan.
- **Kemampuan berbagi data**
Dalam basis data, data dapat diakses dan digunakan oleh banyak pengguna dan aplikasi secara bersamaan.

Dengan adanya basis data, organisasi dapat mengelola data dengan lebih baik, membuat keputusan yang lebih baik berdasarkan informasi yang akurat, dan meningkatkan produktivitas serta efisiensi proses bisnis.

1.3 *Database Management System*

DBMS adalah kependekan dari "*Database Management System*" atau Sistem Manajemen Basis Data. Ini adalah perangkat lunak yang bertugas untuk mengelola dan menyediakan akses ke basis data. *DBMS* berfungsi sebagai perantara antara pengguna atau aplikasi dengan basis data, memungkinkan mereka untuk berinteraksi dengan data dengan cara yang terstruktur dan efisien.

Fungsi utama dari *DBMS* adalah untuk mengatur cara data disimpan, diorganisir, diakses, dan dikelola dalam basis data. *DBMS* menyediakan berbagai layanan dan fungsi untuk pengguna atau aplikasi, termasuk:

- **Pembuatan dan Pengelolaan Basis Data**
DBMS memungkinkan pembuatan basis data dan struktur skema yang mendefinisikan tabel, kolom, indeks, dan aturan integritas data.
- **Penyimpanan Data**
DBMS bertanggung jawab untuk menyimpan data dalam format yang optimal dan efisien di media penyimpanan fisik seperti *hard disk*.
- **Manipulasi Data**
DBMS menyediakan operasi untuk menambahkan (*insert*), mengambil (*select*), memperbarui (*update*), dan menghapus (*delete*) data dalam basis data melalui bahasa kueri seperti *SQL (Structured Query Language)*.
- **Pengelolaan Kunci dan Indeks**
DBMS mendukung pembuatan indeks dan kunci yang membantu meningkatkan kinerja akses data dengan cara mengurangi waktu pencarian data.
- **Keamanan**
DBMS menyediakan mekanisme untuk mengatur hak akses pengguna dan hak akses pada tingkat kolom atau baris tertentu, sehingga data sensitif dapat dijaga dari akses yang tidak sah.
- **Integritas Data**
DBMS memastikan data tetap konsisten dan sesuai dengan aturan integritas yang telah ditetapkan.
- **Transaksi dan Pengendalian Kesalahan**
DBMS mendukung transaksi yang aman dan pengendalian kesalahan untuk memastikan keandalan data.
Setiap *DBMS* memiliki kelebihan dan kelemahan tertentu, sehingga pilihan *DBMS* yang tepat harus didasarkan pada kebutuhan dan karakteristik aplikasi atau sistem yang akan digunakan.
Beberapa jenis software *DBMS* yang dapat digunakan meliputi: