**LAPORAN RESMI**

**MODUL II**

**VIEW**

**SISTEM MANAJEMEN BASIS DATA**



**NAMA : Agus Putra Singkih**

**N.R.P : 200442200153**

**DOSEN : Fitri Damayanti, S.Kom., M.Kom.**

**ASISTEN : Muhammad Iqbal Firmansyah**

**TGL PRAKTIKUM : Jum’at, 29 Maret 2024**

**Disetujui : .. ……… 2024**

**Asisten**

**Muhammad Iqbal Firmansyah**

**21.04.411.00084**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Database atau basis data sangat penting dalam dunia teknologi informasi karena dapat membantu pengguna untuk mengelola dan memanipulasi data secara efisien dan efektif. Dalam sebuah database yang besar dan kompleks, seringkali sulit untuk melihat dan memfokuskan pada beberapa data tertentu tanpa harus membuka keseluruhan database. Dalam hal ini, View adalah salah satu tool penting dalam SQL yang memungkinkan pengguna untuk membuat tampilan atau pandangan khusus dari database yang terdiri dari beberapa tabel yang saling berhubungan. View dapat mempercepat proses pengambilan informasi yang diinginkan dari database dan meminimalkan kemungkinan terjadinya kesalahan karena memungkinkan pengguna untuk fokus pada data tertentu tanpa harus mengakses keseluruhan database. Dengan memahami konsep dan penggunaan View dalam SQL, pengguna dapat memaksimalkan potensi dari basis data yang mereka kelola.

View juga membantu pengguna untuk menjaga keamanan data dan privasi dengan memberikan akses terbatas hanya untuk data yang dibutuhkan, tanpa harus memberikan akses penuh ke seluruh database. Selain itu, View juga dapat digunakan untuk menggabungkan data dari beberapa tabel yang saling terkait menjadi satu tampilan yang lebih mudah dipahami dan digunakan oleh pengguna. Dalam praktiknya, pengguna dapat menggunakan View untuk membuat laporan, grafik, atau dashboard yang memudahkan dalam memahami data dan membuat keputusan bisnis yang lebih baik. Dalam hal ini, pemahaman tentang View menjadi penting untuk setiap orang yang bekerja dengan basis data dan mengelola data dalam skala besar dan kompleks.

Selain itu, penggunaan View dalam SQL juga memungkinkan pengguna untuk menyederhanakan struktur database dengan menggabungkan beberapa tabel menjadi satu tampilan logis, sehingga dapat mempermudah penggunaan dan manajemen database. Hal ini dapat membantu pengguna dalam menghindari kesalahan dalam pemrograman, mempercepat kinerja database, dan meminimalkan kebutuhan akan pembaruan struktur database ketika terjadi perubahan dalam sistem atau data yang disimpan. Dalam konteks pemrograman, View juga memungkinkan pengguna untuk membagi tugas antara tim pemrogram dan mengatur hak akses terhadap data dalam database. Dengan demikian, penggunaan View dalam SQL menjadi sangat penting bagi siapa saja yang ingin mengelola dan memanipulasi data dalam skala besar dan kompleks, dan dapat membantu dalam mengoptimalkan kinerja dan keamanan database.

## **Tujuan**

1. Mampu memahami konsep dasar view didalam basis data.
2. Mampu memahami penerapan view.
3. Mampu menyelesaikan pengambilan data dengan menggunakan pendekatan view

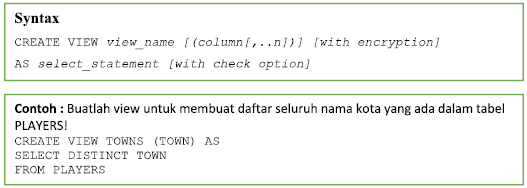
# **BAB II DASAR TEORI**

## **Dasar Teori**

### **Pengertian View**

View adalah tabel virtual yang terbuat dari suatu query terhadap satu tabel atau beberapa tabel. View tidak ada secara nyata dalam database. View hanya digunakan untuk menyederhanakan dan mempermudah persepsi pengguna dalam database. Tidak seperti pada umumnya tabel di dalam basis data relasional, view bukanlah bagian dari skema fisik. View bersifat dinamis, ia mengandung data dari tabel yang direpresentasikannya. Dengan demikian, ketika tael menjadi sumber datanya berubah, data di view juga akan berubah. Kegunaan view antara lain :

1. Memfokuskan pada data tertentu
2. Penyederhanaan manipulasi data
3. Menyesuaikan data dengan kebutuhan user
4. Export dan impor data
5. Mengkombinasikan data terpartisi



### **Updatable**

View dapat berisi read-only atau updatable. Kondisi ini sangat dipengaruhi oleh adanya pendefinisian view itu sendiri. Bagaimanapun, untuk menciptakan updatable view, pernyataan SELECT yang didefinisikan di view haru mengikuti aturan-aturan sebagai berikut :

* Pernyataan SELECT tidak boleh merujuk ke lebih dari satu tabel.
* Pernyataan SELECT tidak boleh menggunakan klausa GROUP BY atau HAVING.
* Pernyataan SELECT harus tidak menggunakkan DISTINCT.
* Pernyataan SELECT harus tidak merujuk ke view lain yang tidak updatable.
* Pernyataan SELECT tidak boleh mengandung ekspresi apa pun, misalnya fungsi agregat.

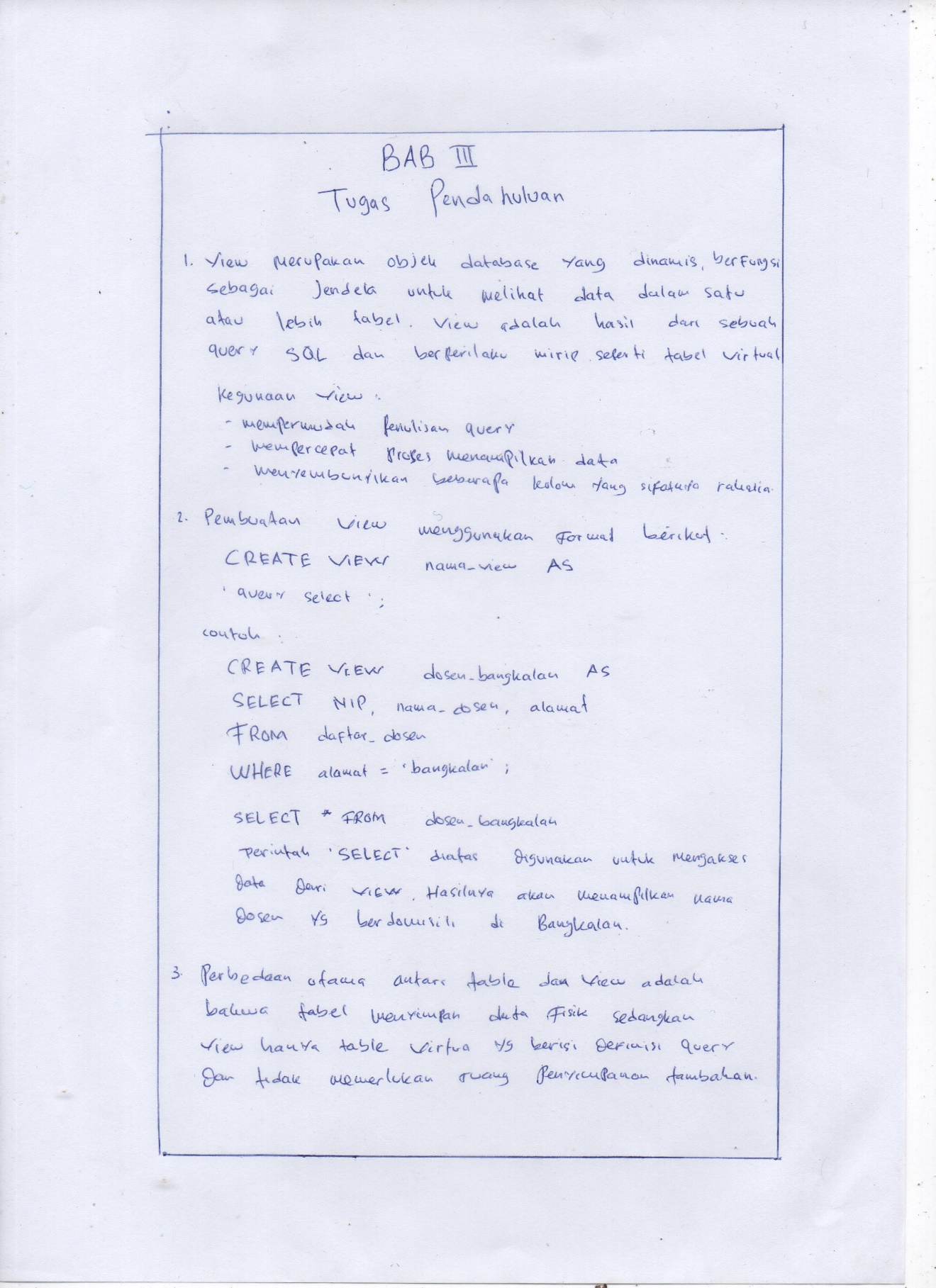
Pada hakikatnya, jika sistem database mampu menentukan pemetaan balik dari skema view ke skema tabel dasar, maka view memungkinkan untuk di update. Dalam kondisi ini, operasi-operasi INSERT, UPDATE dan DELETE dapat diterapkan pada view.

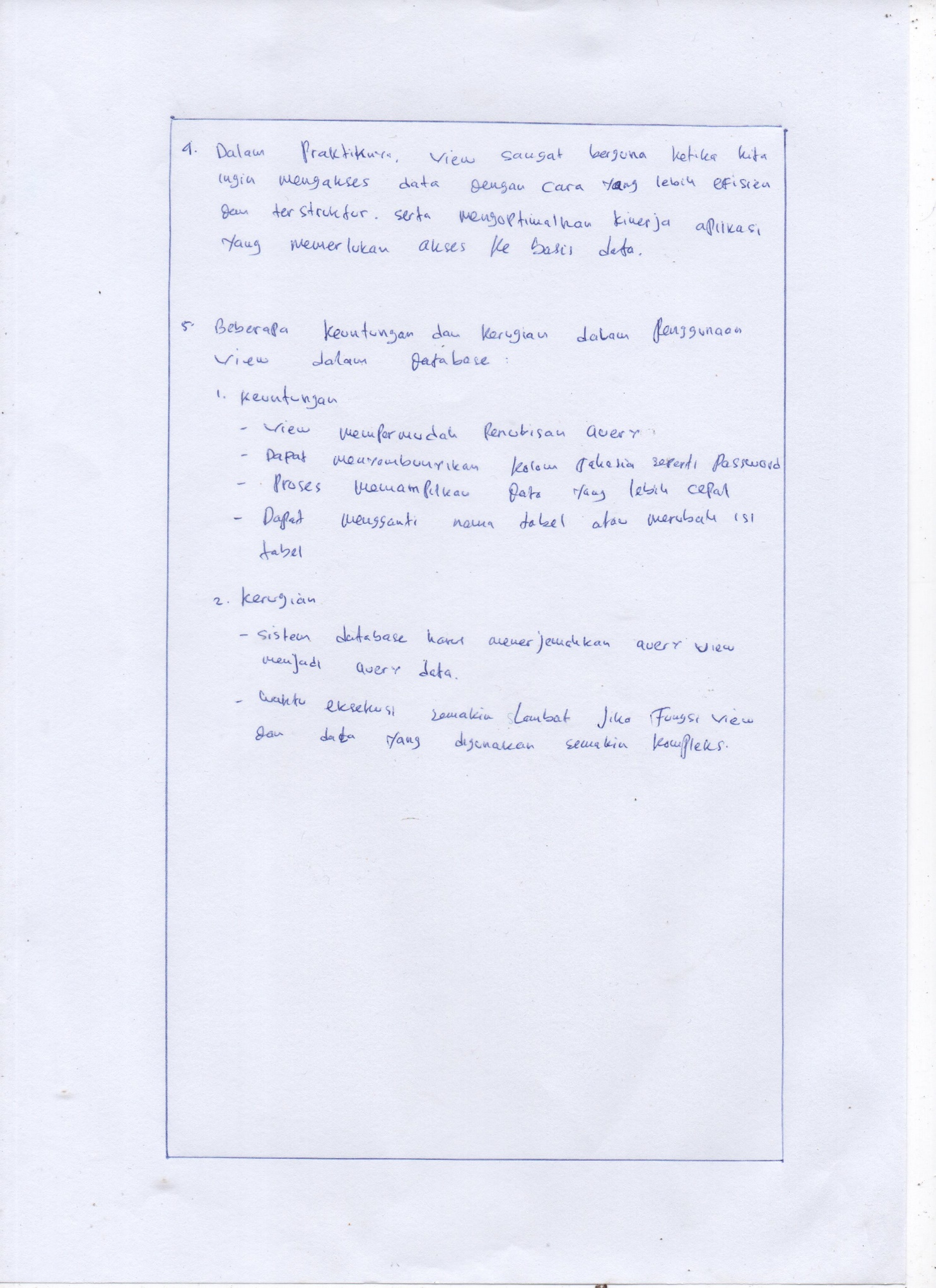
### **Hapus View**

Suatu view dari query selalu menampilkan data yang terbaru, sebagai contoh bila kita memodifikasi sebagian tupel dalam tabel dasarnya dimana view tersebut didefinisikan maka secara otomatis akan berpengaruh pada view di query. Jika tidak membutuhkan view lagi, kita bisa menggunakan perintah :



# **BAB III TUGAS PENDAHULUAN**

****

****

# **BAB IV IMPLEMENTASI**

## **Implementasi**

1. Definisikan view untuk menampilkan nama pelanggan, total, dan tanggal pesanan untuk semua pesanan yang memiliki total pesanan lebih dari rata-rata total pesanan?

Jawaban :

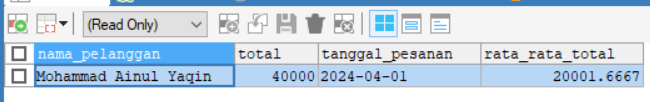
Query :

|  |
| --- |
| CREATE VIEW IF NOT EXISTS pesanan\_diatas\_rata AS  SELECT pelanggan.nama\_pelanggan, pesanan.total, pesanan.tanggal\_pesanan  FROM pelanggan  JOIN pesanan ON pelanggan.id\_pelanggan = pesanan.id\_pesanan  WHERE pesanan.total > (  SELECT \* FROM pesanan\_diatas\_rata  INNER JOIN (  SELECT AVG(total) AS rata\_rata\_total  FROM pesanan  ) AS subquery ON pesanan\_diatas\_rata.total > subquery.rata\_rata\_total; |

Penjalasan :

Pernyataan SQL di atas membuat sebuah "view" baru yang disebut "pesanan\_diatas\_rata". View ini menampilkan daftar nama pelanggan yang telah melakukan pesanan di atas rata-rata, dengan menghitung jumlah pesanan yang dilakukan oleh setiap pelanggan. Dalam view ini menggunakan syntax “JOIN” untuk menggabungkan tabel pesanan dengan tabel pelannggan. Syntax “SELECT AVG” digunakan untuk memfilter kolom-kolom yang ditampilkan dalam view. Pernyataan mengenai “SELECT pelanggan.nama\_pelanggan, pesanan.total,pesanan.tanggal\_pesanan” adalah fungsi SELECT yang akan menentukan kolom-kolom yang akan ditampilkan. Dalam hal ini view akan menampilkan nama pelanggan, total, dan tanggal pesanan.

Output :



1. Buatlah view yang menampilkan nama produk, harga satuan, jumlah produk terjual, dan total pendapatan untuk setiap produk yang telah terjual dalam setiap pesanan!

Jawaban :

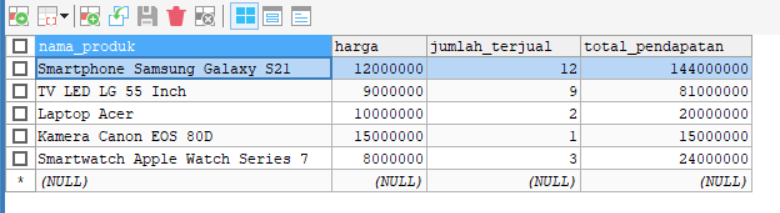
Query :

|  |
| --- |
| CREATE VIEW IF NOT EXISTS detail\_penjualan AS  SELECT p.nama\_produk, p.harga, dp.jumlah AS jumlah\_terjual, (p.harga \* dp.jumlah) AS total\_pendapatan  FROM produk p  JOIN detail\_pesanan dp ON p.id\_produk = dp.id\_produk; |

Penjelasan :

Pernyataan SQL di atas membuat sebuah "view" baru yang disebut "detail\_penjualan". View ini menampilkan nama produk, harga, jumlah terjual, dan total pendapatan. Kolom nama\_produk di dapat dari table produk yang merupakan harga dari setiap produk (Menggunakan fungsi p.produk). kemudian, kolom total\_pendapatan didapat dari hasil penghitungan harga dari tabel produk dan jumlah dari tabel pesanan. Pernyataan ini ditandai dengan perintah “(p.harga \* dp.jumlah)” sehingga terbentuk kolom baru yaitu total\_pendapatan. Pada query di atas juga digunakan fungsi JOIN yaitu untuk menggabungkan tabel detail\_pesanan dengan tabel produk.

Output :



1. Definisikan view untuk menampilkan nama produk dan jumlah stok yang tersisa untuk produk-produk yang memiliki stok kurang dari 5.

Jawaban :

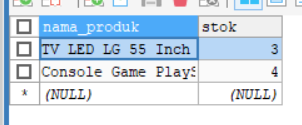
Query :

|  |
| --- |
| CREATE VIEW IF NOT EXISTS stok\_kurang\_dari\_5 AS  SELECT nama\_produk, stok  FROM produk  WHERE stok < 5 ; |

Penjelasan :

Pernyataan pertama dalam syntax di atas adalah membuat view dengan nama “stok\_kurang\_dari\_5”. Kemudian digunakan perintah SELECT untuk menentukan kolom yang akan ditampilan dalam view. Dalam hal ini menampilkan dua kolom saja yaitu nama\_produk dan stok dari table produk. Kemudian kita gunakan perintah WHERE untuk memfilter stok yang kurang dari 5.

Ouput :



1. Buatlah view yang menampilkan nama pelanggan dan jumlah total pesanan yang telah dilakukan oleh setiap pelanggan dalam satu bulan terakhir.

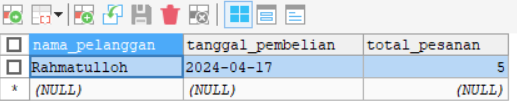
Jawaban :

|  |
| --- |
| CREATE VIEW view\_total\_pesanan\_perpelanggan AS  SELECT p.nama\_pelanggan,  pes.tanggal\_pesanan AS tanggal\_pembelian,  SUM(pes.total) AS total\_pesanan  FROM pesanan pes  JOIN pelanggan p ON pes.id\_pelanggan = p.id\_pelanggan  WHERE pes.tanggal\_pesanan >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH)  GROUP BY p.nama\_pelanggan, pes.tanggal\_pesanan; |

Penjelasan :

Pernyataan di atas digunakan untuk membuah sebuah view yang digunakan untuk menampilkan jumlah pesanan yang dilakukan dalam satu bulan terakhir. Kolom nama\_pelanggan diambil dari table pelanggan pada kolom nama\_pelanggan, sedangkan jumla\_pesanan didapat dari penghitungan COUNT dari table pesanan kolom id\_pesanan. Kemudian diberi alias yaitu jumlah\_pesanan. Klause LEFT JOIN digunakan untuk menggabungkan dua table antara table pesanan dan tabel pelanggan. Kemudian fungsi WHERE digunakan untuk memfilter hasil perhitungan di atas sesuai tanggal selama satu bulan terakhir. Kemudian GROUP BY digunakan untuk mengelompokkan hasil berdasarkan perhitungan pelanggan, sehingga view dapat menampilkan jumlah dari pesanan pelanggan.

Output :



## **Penjelasan Hasil Praktikum**

Berdasarkan hasil praktikum yang dilakukan, kita dapat mengetahui fungsi view dan menjalankannya. View berfungsi untuk menampilkan data yang kita butuhkan tanpa membuat tabel baru. Penggunaan view bisa kita lihat pada fitur *sort by* dalam sebuah website. Kemudian penggunaan view juga berguna pada efesiensi data dan kecepatan pemrosesan data. Sehingga program yang kita bangun dapat berjalan lebih optimal.

# **BAB V PENUTUP**

## **Analisa**

Dari hasil praktikum, praktikan menganalisa bahwa tools view ini merupakan sebuah tools yang sangat berguna apabila kita sedang ingin fokus pada beberapa data saja yang ada di dalam database, dimana tujuan dari pembuatan view ini sendiri adalah untuk kenyamanan (mempermudah penulisan query), untuk keamanan (menyembunyikan beberapa kolom yang bersifat rahasia), atau dalam beberapa kasus bisa digunakan untuk mempercepat proses menampilkan data (terutama jika kita akan menjalankan query tersebut secara berulang).

## **Kesimpulan**

View adalah tabel virtual (bukan tabel sebenarnya) yang dibuat dari beberapa tabel atau view lain. View tidak memiliki data sendiri, tetapi data-datanya berasal dari tabel-tabel atau view lain. View dibuat atau dibangun dengan SELECT statement. Di dalam query SELECT tersebut kita bisa menambahkan ORDER BY statement untuk mengurutkan data atau tidak sama sekali. View dapat digunakan untuk memudahkan atau menyederhanakan data yang ditampilkan, mencegah menampilkan data yang penting dengan menyembunyikan nama kolomnya.