

LAPORAN PROYEK PENGENALAN BASIS DATA

PROYEK MERANCANG, MEMBANGUN, DAN MENGELOLA BASIS DATA DI MYSQL



Agus Pranata Marpaung

13323033

DIII TEKNOLOGI KOMPUTER

**INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI**

Judul Proyek

Minggu/Sesi	:	XV/2
Kode Mata Kuliah	:	1131205
Nama Mata Kuliah	:	PENGENALAN BASIS DATA
Setoran	:	Softcopy
Batas Waktu Setoran	:	<i>Senin, 13 Mei 2024 jam 21:00</i>
Tujuan	:	<i>1. Tujuan dari proyek ini adalah mahasiswa mampu menerapkan konsep perancangan dan pengembangan basisdata dengan benar menggunakan DBMS MYSQL.</i>

Petunjuk

Dalam pengerjaan proyek ini mahasiswa akan melakukan perancangan dan pengembangan basisdata untuk studi kasus yang dikembangkan di PA1, dengan scope analisis dan implementasi basisdata mencakup penanganan pemesanan produk/layanan dari customer.

Penugasan

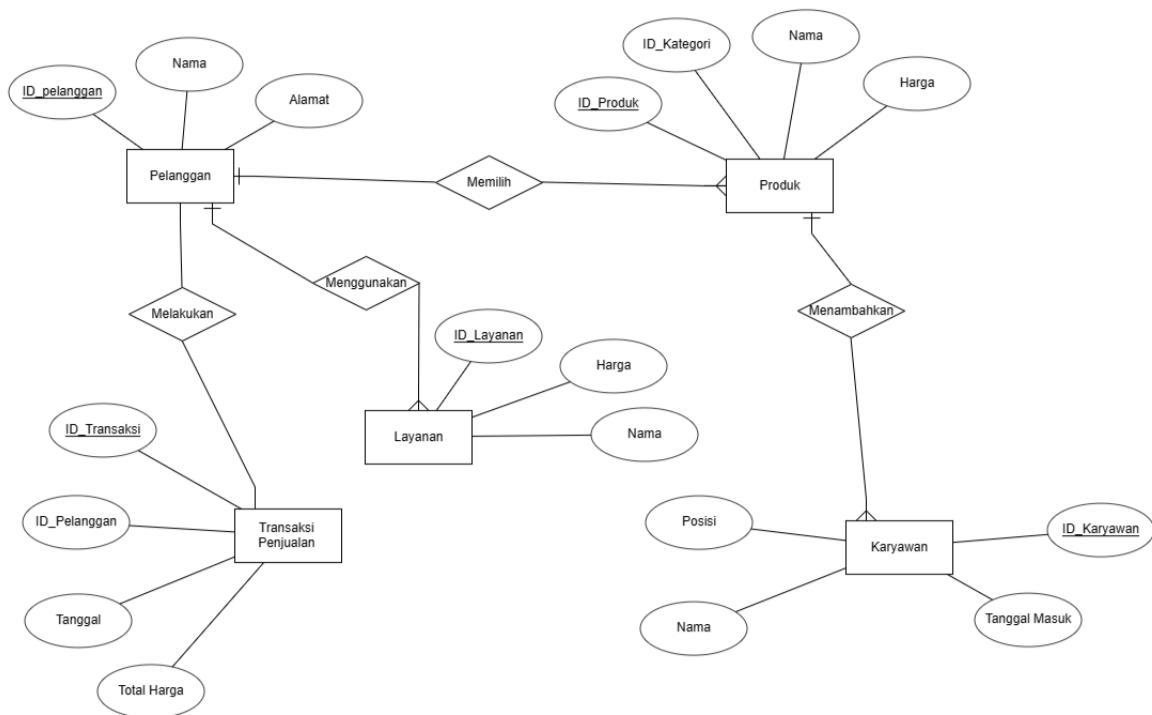
1. Laporan proyek, dengan format nama file : Laporan Proyek_NIM.pdf (Semua yang dilakukan dari no 1 s.d 4 didokumentasikan dalam laporan proyek)
2. File SQL, dengan format nama file : Syntax_NIM.sql

Tugas

Tugas mahasiswa adalah sebagai berikut:

- Buat rancangan basisdata (buat ERD dan lakukan translasi ERD menjadi table-table yang sesuai)

Jawab:



Penjelasan:

1. Pelanggan

Entitas ini mempresentasikan pelanggan dalam sistem. Setiap pelanggan memiliki atribut:

- **ID_Pelanggan** : Identifikasi unik untuk setiap pelanggan.
- **Nama** : Nama Pelanggan.
- **Alamat** : Alamat Pelanggan.

2. Transaksi_Penjualan

Entitas ini mempresentasikan transaksi penjualan yang dilakukan oleh pelanggan. Atributnya yaitu:

- **ID_Transaksi** : Identifikasi unik untuk setiap transaksi.
- **ID_Pelanggan** : Identifikasi pelanggan yang melakukan transaksi. Ini adalah foreign key yang menghubungkan ke entitas Pelanggan.
- **Tanggal** : Tanggal transaksi dilakukan.
- **Total Harga** : Total harga dari transaksi.

3. Produk

Entitas ini mempresentasikan produk yang tersedia untuk dibeli. Atributnya adalah:

- **ID_Prod** : Identifikasi unik untuk setiap produk.
- **ID_Kategori** : Identifikasi kategori produk.
- **Nama** : Nama produk.
- **Harga** : Harga produk.

4. Layanan

Entitas ini mempresentasikan produk yang tersedia. Atributnya adalah:

- **ID_Layanan** : Identifikasi unik untuk setiap layanan
- **Harga** : Harga layanan.
- **Nama** : Nama layanan.

5. Karyawan

Entitas ini mempresentasikan karyawan yang berkerja. Atributnya adalah:

- **ID_Karyawan** : Identifikasi unik untuk setiap karyawan.
- **Posisi** : Posisi karyawan.
- **Nama** : Nama karyawan.
- **Tanggal_Masuk** : Tanggal karyawan mulai bekerja.

Berikut hubungan antara Entitasnya:

- **Melakukan** : Hubungan antara pelanggan dan Transaksi Penjualan. Menunjukkan bahwa pelanggan melakukan transaksi penjualan.
- **Memilih** : Hubungan antara Transaksi Penjualan dan Produk. Menunjukkan bahwa dalam sebuah transaksi penjualan, produk dipilih.
- **Menggunakan** : Hubungan antara Transaksi Penjualan dan Layanan. Menunjukkan bahwa dalam sebuah transaksi penjualan, layanan digunakan.
- **Menambahkan** : Hubungan antara Karyawan dan Layanan. Menunjukkan bahwa karyawan menambahkan Layanan.

2. Lakukan validasi terhadap rancangan anda dengan konsep normalisasi.

Jawab:

- **Tabel Karyawan**

The screenshot shows the MySQL Workbench interface with the 'karyawan' table selected. The table has four columns: Karyawan_id, Nama, Posisi, and Tanggal_masuk. The data consists of three rows:

	Karyawan_id	Nama	Posisi	Tanggal_masuk
<input type="checkbox"/>	1	Dewi	Kasir	2022-01-01
<input type="checkbox"/>	2	Ani	Karyawati	2022-01-02
<input type="checkbox"/>	3	Lini	Karyawati	2022-01-03

- Data tersebut sudah 1NF karena setiap selnya hanya diisi satu nilai.
- Data tersebut tidak lagi dinormalisasi untuk 2NF atau 3NF karena tidak adanya dependensi yang kompleks.

• Tabel Layanan

Layanan_id	Nama	Harga
1	Manikur	100000
2	Makeup Untuk Acara Khusus	250000
3	Facial Perawatan Jerawat	100000

- Data tersebut sudah 1NF karena setiap sel nya hanya diisikan satu nilai.
- Data tersebut tidak lagi dinormalisasi untuk 2NF atau 3NF karena tidak adanya dependensi yang kompleks.

• Tabel Pelanggan

ID_Pelanggan	Nama	Alamat
1	Aisyah	Jl. Durian
2	Budi	Jl. Mangga
3	Citra	Jl. Apel
4	James	Jl. Sutomo
5	Karen	Jl. Simpang Mansur
6	Lini	Jl. Pangandaran
7	Kesya	Jl. Belawan
8	Winner	Jl. Pahlawan
9	Juna	Jl. Prismatama
10	Hana	Jl. Mulia
11	Tina	Jl. Bintaro

- Data tersebut sudah 1NF karena setiap sel nya hanya diisikan satu nilai.
- Data tersebut tidak lagi dinormalisasi untuk 2NF atau 3NF karena tidak adanya dependensi yang kompleks.

• Tabel Produk

The screenshot shows the MySQL Workbench interface for the 'produk' table. The table has columns: Produk_id, Kategori_id, Nama, and Harga. The data is as follows:

Produk_id	Kategori_id	Nama	Harga
1	101	Cream	10000
2	102	Shampoo	10000
3	103	Conditioner	20000
4	104	Face Wash	10000
5	105	Body Wash	10000
6	106	Minyak Rambut	25000
7	107	Semir Bigen	30000
8	108	Anti Aging	20000
9	109	Conditioner + Minyak Rambut	20000
10	110	Face Wash + Body Wash	20000
11	111	Shampoo Anti Rontok	30000

- Data tersebut sudah 1NF karena setiap sel nya hanya diisikan satu nilai.
- Data tersebut tidak lagi dinormalisasi untuk 2NF atau 3NF karena tidak adanya dependensi yang kompleks.

• Tabel Transaksi_Penjualan

The screenshot shows the MySQL Workbench interface for the 'transaksi_penjualan' table. The table has columns: Transaksi_id, Pelanggan_id, Produk_id, Tanggal, and Total_harga. The data is as follows:

Transaksi_id	Pelanggan_id	Produk_id	Tanggal	Total_harga
1001	1	1	2022-03-01	100000
1002	2	2	2022-03-02	250000
1003	3	3	2022-03-03	100000
1004	4	4	2022-03-04	120000
1005	5	5	2022-03-05	260000
1006	6	6	2022-03-07	140000
1007	7	7	2022-03-09	130000
1008	8	8	2022-03-11	210000
1009	9	9	2022-03-12	112000
1010	10	10	2022-03-13	101000
1011	11	11	2022-03-14	190000

- Data tersebut sudah 1NF karena setiap sel nya hanya diisikan satu nilai.
- Data tersebut tidak lagi dinormalisasi untuk 2NF atau 3NF karena tidak adanya dependensi yang kompleks.

3. Lakukan pengembangan basisdata sesuai rancangan final (menerapkan Data Definition Language pada SQL)

- a) Pembuatan basisdata beserta tabel-tabelnya

Jawab:

```

63 -- Data Dummy Masing Masing Tabel
64 INSERT INTO `Pelanggan` ('ID_Pelanggan', 'Nama', 'Alamat') VALUES
65 (1, 'Aisyah', 'Jl. Durian'),
66 (2, 'Budi', 'Jl. Mangga'),
67 (3, 'Citra', 'Jl. Apel'),
68 (4, 'James', 'Jl. Sutomo'),
69 (5, 'Karen', 'Jl. Simpang Mansur'),
70 (6, 'Lini', 'Jl. Pangandaran'),
71 (7, 'Kesya', 'Jl. Belawan'),
72 (8, 'Winner', 'Jl. Pahlawan'),
73 (9, 'Junia', 'Jl. Prismatama'),
74 (10, 'Hana', 'Jl. Mulia'),
75 (11, 'Tina', 'Jl. Bintaro');
76
77 INSERT INTO `Produk` ('Produk_id', 'Kategori_id', 'Nama', 'Harga') VALUES
78 (1, 101, 'Cream', 10000),
79 (2, 102, 'Shampoo', 10000),
80 (3, 103, 'Conditioner', 20000);
81
82 INSERT INTO `Karyawan` ('Karyawan_id', 'Nama', 'Posisi', 'Tanggal_masuk') VALUES
83 (1, 'Dewi', 'Kasir', '2022-01-01'),
84 (2, 'Ani', 'Karyawati', '2022-01-02'),
85 (3, 'Lini', 'Karyawati', '2022-01-03');
86
87 INSERT INTO `Layanan` ('Layanan_id', 'Nama', 'Harga') VALUES
88 (1, 'Manikur', 10000),
89 (2, 'Makeup Untuk Acara Khusus', 25000),
90 (3, 'Facial Perawatan Jerawat', 10000);
91
92 INSERT INTO `Transaksi_Penjualan` ('Transaksi_id', 'Pelanggan_id', 'Produk_id', 'Tanggal', 'Total_harga') VALUES
93 (1001, 1, 1, '2022-03-01', 10000),
94 (1002, 2, 2, '2022-03-02', 25000),
95 (1003, 3, 3, '2022-03-03', 10000),
96 (1004, 4, 4, '2022-03-04', 12000),
97 (1005, 5, 5, '2022-03-05', 26000),
98 (1006, 6, 6, '2022-03-07', 14000),
99 (1007, 7, 7, '2022-03-09', 13000),
100 (1008, 8, 8, '2022-03-11', 21000),
101 (1009, 9, 9, '2022-03-12', 112000),
102 (1010, 10, 10, '2022-03-13', 101000),
103 (1011, 11, 11, '2022-03-14', 190000);
104

```

Hasil:

1. Pelanggan

ID_Pelanggan	Nama	Alamat
1	Aisyah	Jl. Durian
2	Budi	Jl. Mangga
3	Citra	Jl. Apel
4	James	Jl. Sutomo
5	Karen	Jl. Simpang Mansur
6	Lini	Jl. Pangandaran
7	Kesya	Jl. Belawan
8	Winner	Jl. Pahlawan
9	Junia	Jl. Prismatama
10	Hana	Jl. Mulia
11	Tina	Jl. Bintaro

Penjelasan:

Pada query Data Dummy untuk tabel Pelanggan dengan atribut ID_Pelanggan, Nama, dan Alamat. Query tersebut akan dimasukkan ke Database dengan Nama Tabel Pelanggan.

2. Produk

Produk_Id	Kategori_Id	Nama	Harga
1	101	Cream	10000
2	102	Shampoo	10000
3	103	Conditioner	20000

Penjelasan:

Pada query Data Dummy untuk tabel Produk dengan atribut Produk_id, Kategori_id, Nama, dan Harga. Query tersebut akan dimasukkan ke Database dengan Nama Tabel Produk.

3. Karyawan

Karyawan_Id	Nama	Posisi	Tanggal_masuk
1	Dewi	Kasir	2022-01-01
2	Ani	Karyawati	2022-01-02
3	Lini	Karyawati	2022-01-03

Penjelasan:

Pada query Data Dummy untuk tabel Karyawan dengan atribut Karyawan_id, Nama, Posisi, dan Tanggal_masuk. Query tersebut akan dimasukkan ke Database dengan Nama Tabel Karyawan.

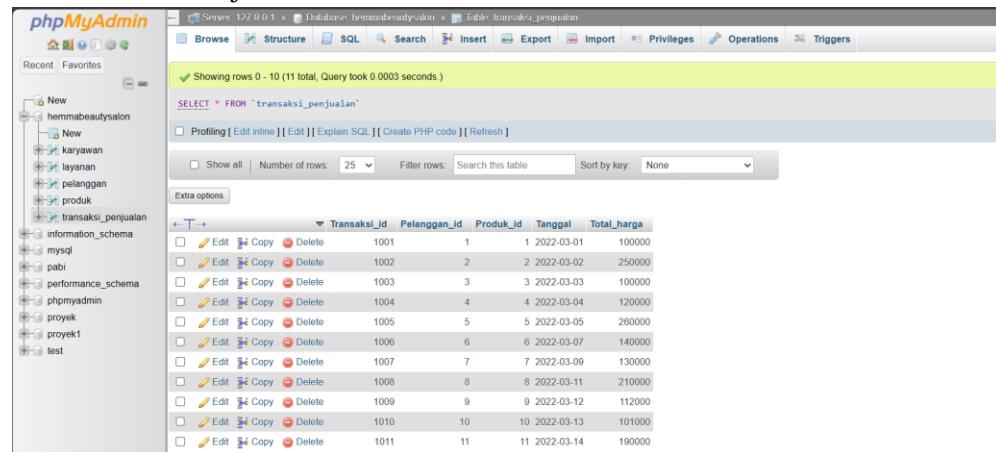
4. Layanan

Layanan_Id	Nama	Harga
1	Manikur	10000
2	Makeup Untuk Acara Khusus	250000
3	Facial Perawatan Jerawat	100000

Penjelasan:

Pada query Data Dummy untuk tabel Layanan dengan atribut Layanan_id, Nama, dan Harga. Query tersebut akan dimasukkan ke Database dengan Nama Tabel Layanan.

5. Transaksi_Penjualan



	Transaksi_id	Pelanggan_id	Produk_id	Tanggal	Total_harga
1	1001	1	1	2022-03-01	100000
2	1002	2	2	2022-03-02	250000
3	1003	3	3	2022-03-03	100000
4	1004	4	4	2022-03-04	120000
5	1005	5	5	2022-03-05	260000
6	1006	6	6	2022-03-07	140000
7	1007	7	7	2022-03-09	130000
8	1008	8	8	2022-03-11	210000
9	1009	9	9	2022-03-12	112000
10	1010	10	10	2022-03-13	101000
11	1011	11	11	2022-03-14	190000

Penjelasan:

Pada query Data Dummy untuk tabel transaksi_penjualan dengan atribut Transaksi_id, Pelanggan_id, Produk_id, Tanggal, dan Total_harga. Query tersebut akan dimasukkan ke Database dengan Nama Tabel transaksi_penjualan.

- b) Enforcing Data Integrity dengan menerapkan jenis-jenis constraint yang sudah anda pelajari.

Jawab:

```
16 -- Membuat tabel Pelanggan
17 CREATE TABLE `Pelanggan` (
18     `ID_Pelanggan` INT NOT NULL,
19     `Nama` VARCHAR(255) NOT NULL,
20     `Alamat` VARCHAR(255) NOT NULL,
21     PRIMARY KEY (`ID_Pelanggan`)
22 );
23
24
25 -- Membuat tabel Produk
26 CREATE TABLE `Produk` (
27     `Produk_id` INT NOT NULL,
28     `Kategori_id` INT,
29     `Nama` VARCHAR(255) NOT NULL,
30     `Harga` INT,
31     PRIMARY KEY (`Produk_id`)
32 );
33
34
35 -- Membuat tabel Karyawan
36 CREATE TABLE `Karyawan` (
37     `Karyawan_id` INT NOT NULL,
38     `Nama` VARCHAR(255) NOT NULL,
39     `Posisi` VARCHAR(255),
40     `Tanggal_masuk` DATE,
41     PRIMARY KEY (`Karyawan_id`)
42 );
43
44 -- Membuat tabel Layanan
45 CREATE TABLE `Layanan` (
46     `Layanan_id` INT NOT NULL,
47     `Nama` VARCHAR(255) NOT NULL,
48     `Harga` INT,
49     PRIMARY KEY (`Layanan_id`)
50 );
51
52 -- Membuat tabel Transaksi Penjualan
53 CREATE TABLE `Transaksi_Penjualan` (
54     `Transaksi_id` INT NOT NULL,
55     `Pelanggan_id` INT,
56     `Produk_id` INT,
57     `Tanggal` DATE,
58     `Total_harga` INT,
59     PRIMARY KEY (`Transaksi_id`),
60     FOREIGN KEY (`Pelanggan_id`) REFERENCES `Pelanggan`(`ID_Pelanggan`),
61     FOREIGN KEY (`Produk_id`) REFERENCES `Produk`(`Produk_id`)
62 );
```

Hasil:

1. Pelanggan

The screenshot shows the phpMyAdmin interface for the 'hemmabeautysalon' database. The left sidebar lists databases and tables. The main area shows a query result for a SELECT query on the 'pelanggan' table, which returns zero rows. The table structure is defined by the query: `CREATE TABLE `pelanggan` (`ID_Pelanggan` int(11) NOT NULL, `Nama` varchar(50) NOT NULL, `Alamat` varchar(100) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1`. The table has three columns: ID_Pelanggan, Nama, and Alamat.

Penjelasan:

Pada query dibagian Membuat Tabel Pelanggan yaitu untuk membuat tabel baru yang bernama pelanggan di PHPMyAdmin yang mana atributnya ada ID_Pelanggan, Nama, dan Alamat di dalam tabel tersebut.

2. Produk

The screenshot shows the phpMyAdmin interface for the 'hemmabeautysalon' database. The left sidebar lists databases and tables. The main area shows a query result for a SELECT query on the 'produk' table, which returns zero rows. The table structure is defined by the query: `CREATE TABLE `produk` (`Produk_id` int(11) NOT NULL, `Kategori_id` int(11) NOT NULL, `Nama` varchar(50) NOT NULL, `Harga` int(11) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1`. The table has four columns: Produk_id, Kategori_id, Nama, and Harga.

Penjelasan:

Pada query dibagian Membuat Tabel Produk yaitu untuk membuat tabel baru yang bernama produk di PHPMyAdmin yang mana atributnya ada Produk_id, Kategori_id, Nama dan Harga di dalam tabel tersebut.

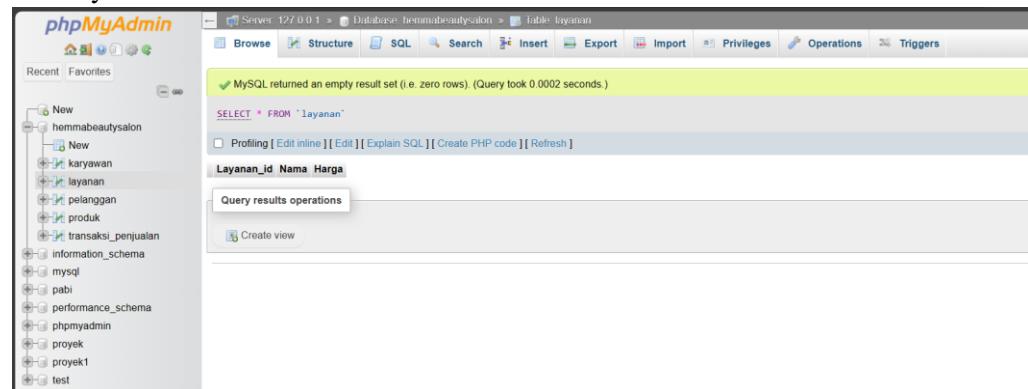
3. Karyawan

The screenshot shows the phpMyAdmin interface for the 'hemmabeautysalon' database. The left sidebar lists databases and tables. The main area shows a query result for a SELECT query on the 'karyawan' table, which returns zero rows. The table structure is defined by the query: `CREATE TABLE `karyawan` (`Karyawan_id` int(11) NOT NULL, `Nama` varchar(50) NOT NULL, `Posisi` varchar(50) NOT NULL, `Tanggal_masuk` date NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1`. The table has four columns: Karyawan_id, Nama, Posisi, and Tanggal_masuk.

Penjelasan:

Pada query dibagian Membuat Tabel Karyawan yaitu untuk membuat tabel baru yang bernama karyawan di PHPMyAdmin yang mana atributnya ada Karyawan_id, Nama, Posisi, dan Tanggal_masuk di dalam tabel tersebut.

4. Layanan



The screenshot shows the phpMyAdmin interface for the 'hemmabeautysalon' database. The left sidebar lists various schemas and tables. In the main area, a query is being run:

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)  
SELECT * FROM `layanan`
```

The results pane shows a table structure:

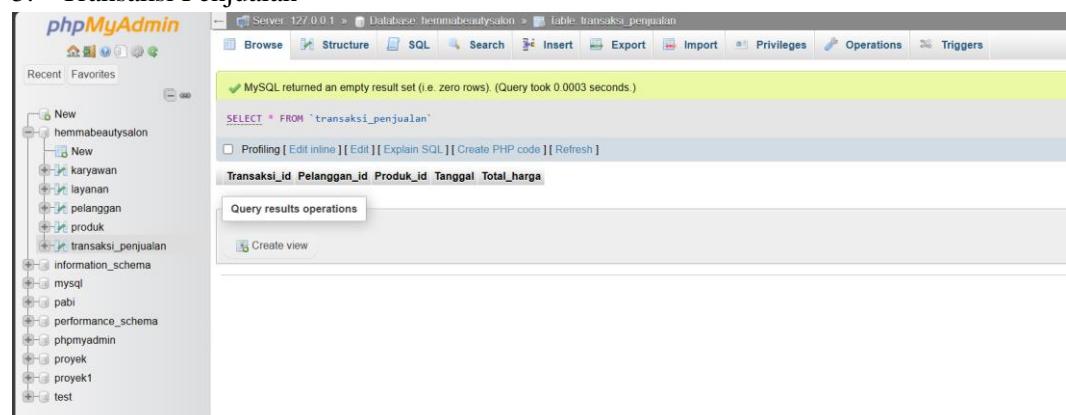
Layanan_id	Nama	Harga

Below the table, there are buttons for 'Query results operations' and 'Create view'.

Penjelasan:

Pada query dibagian Membuat Tabel Layanan yaitu untuk membuat tabel baru yang bernama layanan di PHPMyAdmin yang mana atributnya ada Layanan_id, Nama, dan Harga di dalam tabel tersebut.

5. Transaksi Penjualan



The screenshot shows the phpMyAdmin interface for the 'hemmabeautysalon' database. The left sidebar lists various schemas and tables. In the main area, a query is being run:

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)  
SELECT * FROM `transaksi_penjualan`
```

The results pane shows a table structure:

Transaksi_id	Pelanggan_id	Produk_id	Tanggal	Total_harga

Below the table, there are buttons for 'Query results operations' and 'Create view'.

Penjelasan:

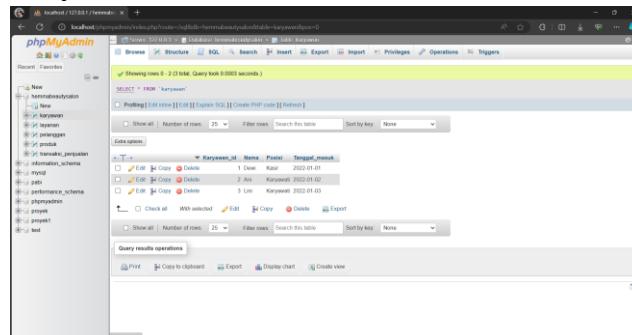
Pada query dibagian Membuat Tabel Transaksi_Penjualan yaitu untuk membuat tabel baru yang bernama transaksi_penjualan di PHPMyAdmin yang mana atributnya ada Transaksi_id, Pelanggan_id, Produk_id, Tanggal, dan Total_harga di dalam tabel tersebut.

4. Lakukan aktifitas pengelolaan basisdata (menerapkan Data Manipulation Language pada SQL)

- a) Lakukan pengisian data dummy pada basisdata yang anda kembangkan

Jawab:

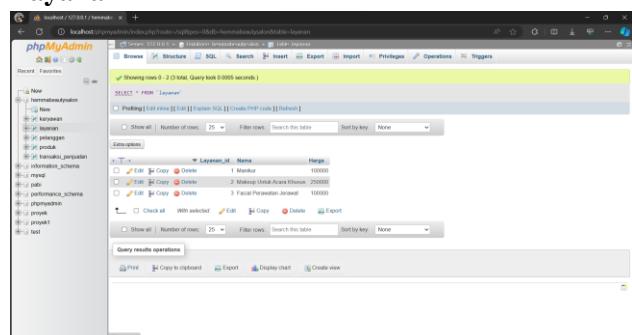
• Karyawan



The screenshot shows the 'karyawan' table in phpMyAdmin. The table has columns: Karyawan_id, Name, Posisi, Tanggal_mulai. There are three rows of data:

Karyawan_id	Name	Posisi	Tanggal_mulai
1	Dave	Kasir	2022-01-01
2	Ari	Kasir	2022-01-02
3	Karyaw	Kasir	2022-01-03

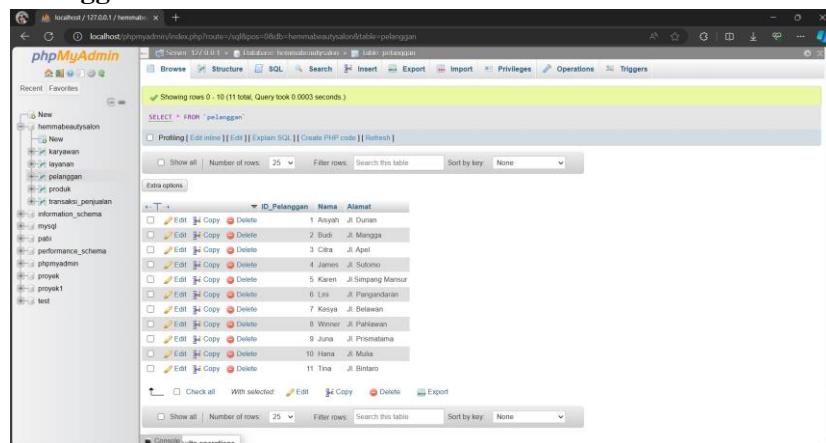
• Layanan



The screenshot shows the 'layanan' table in phpMyAdmin. The table has columns: Layanan_id, Name, Harga. There are three rows of data:

Layanan_id	Name	Harga
1	Murah	10000
2	Makmur Untuk Acara Khusus	25000
3	Fasilitasi Persewaan	10000

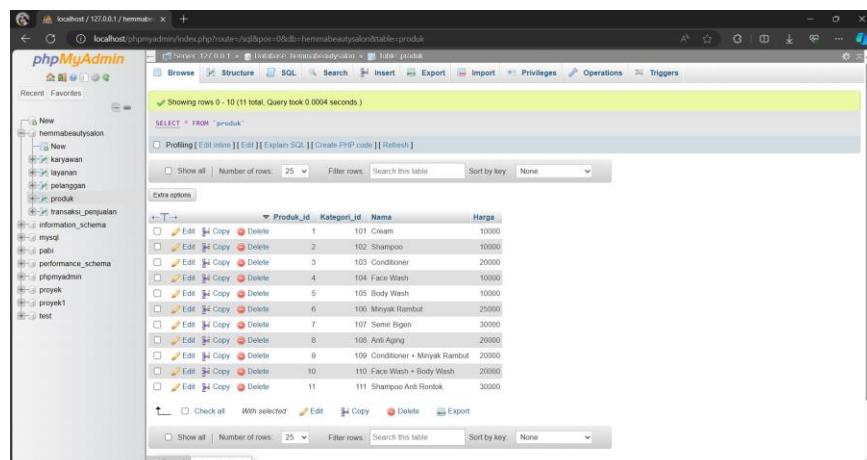
• Pelanggan



The screenshot shows the 'pelanggan' table in phpMyAdmin. The table has columns: ID_Pelanggan, Name, Alamat. There are eleven rows of data:

ID_Pelanggan	Name	Alamat
1	Asyay	Jl. Duran
2	Budi	Jl. Mangga
3	Citra	Jl. Apel
4	James	Jl. Sutomo
5	Karen	Jl. Simpang Mansur
6	Lia	Jl. Pangandaran
7	Keyse	Jl. Belawan
8	Winner	Jl. Palihawan
9	Juna	Jl. Prismatama
10	Hana	Jl. Mula
11	Tina	Jl. Bintaro

• Produk



The screenshot shows the 'produk' table in phpMyAdmin. The table has columns: Produk_id, Kategori_id, Name, Harga. There are eleven rows of data:

Produk_id	Kategori_id	Name	Harga
1	101	Cream	10000
2	102	Shampoo	10000
3	103	Conditioner	20000
4	104	Face Wash	10000
5	105	Body Wash	10000
6	106	Minyak Rambut	25000
7	107	Semir Rambut	30000
8	108	Anti-Aging	20000
9	109	Conditioner + Minyak Rambut	20000
10	110	Face Wash + Body Wash	20000
11	111	Shampoo Anti Rontok	30000

- **Transaksi_Penjualan**

Transaksi_id	Pelanggan_id	Produk_id	Tanggal	Total_harga
1001	1	1	2022-03-01	100000
1002	2	2	2022-03-02	250000
1003	3	3	2022-03-03	100000
1004	4	4	2022-03-04	120000
1005	5	5	2022-03-05	260000
1006	6	6	2022-03-07	140000
1007	7	7	2022-03-09	130000
1008	8	8	2022-03-11	210000
1009	9	9	2022-03-12	112000
1010	10	10	2022-03-13	101000
1011	11	11	2022-03-14	190000

- b) Buat 10 query untuk menampilkan data dari 1 table

Jawab:

```
-- 10 query menampilkan data dari tabel Pelanggan

-- 1. Menampilkan semua data pelanggan
SELECT * FROM `Pelanggan`;

-- 2. Menampilkan nama dan alamat pelanggan dengan ID 1
SELECT `Nama`, `Alamat` FROM `Pelanggan` WHERE `ID_Pelanggan` = 1;

-- 3. Menampilkan semua data pelanggan dengan nama 'Aisyah'
SELECT * FROM `Pelanggan` WHERE `Nama` = 'Aisyah';

-- 4. Menampilkan ID dan nama pelanggan yang alamatnya di 'Jl. Mangga'
SELECT `ID_Pelanggan`, `Nama` FROM `Pelanggan` WHERE `Alamat` = 'Jl. Mangga';

-- 5. Menampilkan semua pelanggan yang ID_Pelanggan lebih besar dari 10
SELECT * FROM `Pelanggan` WHERE `ID_Pelanggan` > 10;

-- 6. Menampilkan nama semua pelanggan yang alamatnya mengandung kata 'Jl.'
SELECT `Nama` FROM `Pelanggan` WHERE `Alamat` LIKE '%Jl.%';

-- 7. Menampilkan ID dan nama semua pelanggan yang namanya diawali dengan 'A'
SELECT `ID_Pelanggan`, `Nama` FROM `Pelanggan` WHERE `Nama` LIKE 'A%';

-- 8. Menampilkan ID dan nama semua pelanggan yang namanya diakhiri dengan 'i'
SELECT `ID_Pelanggan`, `Nama` FROM `Pelanggan` WHERE `Nama` LIKE '%i';

-- 9. Menampilkan jumlah total pelanggan
SELECT COUNT(*) FROM `Pelanggan`;

-- 10. Menampilkan nama pelanggan dengan alamat terpanjang
SELECT `Nama` FROM `Pelanggan` ORDER BY LENGTH(`Alamat`) DESC LIMIT 1;
```

Hasil:

1. Menampilkan semua data pelanggan

ID_Pelanggan	Nama	Alamat
1	Aisyah	Jl. Durian
2	Budi	Jl. Mangga
3	Citra	Jl. Apel
4	James	Jl. Sutomo
5	Karen	Jl. Simpang Mansur
6	Lini	Jl. Pangandaran
7	Kesya	Jl. Belawan
8	Winner	Jl. Pahlawan
9	Juna	Jl. Prismatama
10	Hana	Jl. Mulia
11	Tina	Jl. Bintaro

Penjelasan:

Pada query nomor 1, menampilkan semua data pelanggan yang ada di tabel Pelanggan.

2. Menampilkan nama dan alamat pelanggan dengan ID 1

	Nama	Alamat
	Aisyah	Jl. Durian

Penjelasan:

Pada query nomor 2, menampilkan nama dan alamat pelanggan yang memiliki ID 1 di tabel Pelanggan.

3. Menampilkan semua data pelanggan dengan nama 'Aisyah'

	ID_Pelanggan	Nama	Alamat
	1	Aisyah	Jl. Durian

Penjelasan:

Pada query nomor 3, menampilkan semua data pelanggan yang hanya bernama **Aisyah**.

4. Menampilkan ID dan nama pelanggan yang alamatnya di 'Jl. Mangga'

	ID_Pelanggan	Nama
	2	Budi

Penjelasan:

Pada query nomor 4, menampilkan ID dan Nama Pelanggan yang hanya beralamat di **Jl.Mangga**

5. Menampilkan semua pelanggan yang ID_Pelanggan lebih besar dari 10

ID_Pelanggan	Nama	Alamat
11	Tina	Jl. Bintaro

Penjelasan:

Pada query nomor 5, menampilkan semua data pelanggan yang ID nya itu lebih besar dari 10. Maka dari itu, data yang lebih dari 10 yaitu Tina yang beralamat di Jl. Bintaro.

6. Menampilkan nama semua pelanggan yang alamatnya mengandung kata 'Jl.'

Nama
Aisyah
Budi
Citra
James
Karen
Lini
Kesya
Winner
Juna
Hana
Tina

Penjelasan:

Pada query nomor 6, menampilkan semua data yang mana Alamat nya itu hanya mengandung kata **Jl.**

7. Menampilkan ID dan nama semua pelanggan yang namanya diawali dengan 'A'

ID_Pelanggan	Nama
1	Aisyah

Penjelasan:

Pada query nomor 7, menampilkan data yang mana Namanya itu diawali dengan huruf **A**.

8. Menampilkan ID dan nama semua pelanggan yang namanya diakhiri dengan 'i'

ID_Pelanggan	Nama
2	Budi
6	Lini

Penjelasan:

Pada query nomor 8, menampilkan ID dan nama semua pelanggan yang mana Namanya diakhiri dengan huruf I.

9. Menampilkan jumlah total pelanggan

COUNT (*)
11

Penjelasan:

Pada query nomor 9, data yang ada di tabel pelanggan dihitung dan dijumlahkan berapa pelanggan yang ada di tabel pelanggan.

10. Menampilkan nama pelanggan dengan alamat terpanjang

Nama
Karen

Penjelasan:

Pada query nomor 10, menampilkan nama pelanggan yang mana nama nya itu ditampilkan berdasarkan alamat yang terpanjang dari semua data yang ada di tabel itu.

- c) Buat 10 query untuk menampilkan data dari beberapa tabel (harus mencakup contoh inner JOIN, left/right JOIN)

Jawab:

```

13/
138 -- Buat 10 query untuk menampilkan data dari beberapa tabel (harus mencakup contoh inner JOIN, left/right JOIN)
139
140 -- 1. INNER JOIN
141 SELECT 'P'.Nama , 'T'.Total_harga
142 FROM `Pelanggan` AS 'P'
143 INNER JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id';
144
145 -- 2. LEFT JOIN
146 SELECT 'P'.Nama , 'T'.Total_harga
147 FROM `Pelanggan` AS 'P'
148 LEFT JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id';
149
150 -- 3. RIGHT JOIN
151 SELECT 'P'.Nama , 'T'.Total_harga
152 FROM `Pelanggan` AS 'P'
153 RIGHT JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id';
154
155 -- 4. INNER JOIN dengan kondisi
156 SELECT 'P'.Nama , 'T'.Total_harga
157 FROM `Pelanggan` AS 'P'
158 INNER JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
159 WHERE 'T'.Total_harga > 100000;
160
161 -- 5. LEFT JOIN dengan kondisi
162 SELECT 'P'.Nama , 'T'.Total_harga
163 FROM `Pelanggan` AS 'P'
164 LEFT JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
165 WHERE 'P'.Nama LIKE 'A%';
166
167
168 -- 6. RIGHT JOIN dengan kondisi
169 SELECT 'P'.Nama , 'T'.Total_harga
170 FROM `Pelanggan` AS 'P'
171 RIGHT JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
172 WHERE 'T'.Total_harga > 100000;
173
174 -- 7. INNER JOIN dengan pengurutan hasil
175 SELECT 'P'.Nama , 'T'.Total_harga
176 FROM `Pelanggan` AS 'P'
177 INNER JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
178 ORDER BY 'T'.Total_harga DESC;
179
180 -- 8. LEFT JOIN dengan pengurutan hasil
181 SELECT 'P'.Nama , 'T'.Total_harga
182 FROM `Pelanggan` AS 'P'
183 LEFT JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
184 ORDER BY 'P'.Nama ;
185
186 -- 9. RIGHT JOIN dengan pengurutan hasil
187 SELECT 'P'.Nama , 'T'.Total_harga
188 FROM `Pelanggan` AS 'P'
189 RIGHT JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
190 ORDER BY 'T'.Total_harga;
191
192 -- 10. INNER JOIN dengan pengelompokan hasil
193 SELECT 'P'.Nama , COUNT('T'.Transaksi_id ) AS 'Jumlah_Transaksi'
194 FROM `Pelanggan` AS 'P'
195 INNER JOIN `Transaksi_Penjualan` AS 'T' ON 'P'.ID_Pelanggan = 'T'.Pelanggan_id'
196 GROUP BY 'P'.Nama ;
197

```

Hasil:

1. INNER JOIN

Nama	Total_harga
Aisyah	100000
Budi	250000
Citra	100000

Penjelasan:

Pada query nomor 1 menggabungkan tabel Pelanggan dan Transaksi_Penjualan berdasarkan kolom ID_Pelanggan dan Pelanggan_ID. Hanya baris yang memiliki nilai yang sama di kedua tabel yang akan ditampilkan.

2. LEFT JOIN

Nama	Total_harga
Budi	250000
Citra	100000
James	120000
Karen	260000
Lini	140000
Kesya	130000
Winner	210000
Juna	112000
Hana	101000
Tina	190000

Penjelasan:

Pada query nomor 2 menggabungkan tabel Pelanggan dan Transaksi_Penjualan, tetapi akan menampilkan semua baris dari tabel Pelanggan (tabel sebelah kiri), dan baris dari tabel Transaksi_Penjualan yang cocok. Jika tidak ada kecocokan, hasilnya adalah NULL di sisi kanan.

3. RIGHT JOIN

Nama	Total_harga
Aisyah	100000
Budi	250000
Citra	100000
James	120000
Karen	260000
Lini	140000
Kesya	130000
Winner	210000
Juna	112000
Hana	101000
Tina	190000

Penjelasan:

Pada query nomor 3, mirip dengan **LEFT JOIN**, tetapi query ini akan menampilkan semua baris dari tabel Transaksi_Penjualan (tabel sebelah kanan), dan baris dari tabel **Pelanggan** yang cocok. Jika tidak ada kecocokan, hasilnya adalah NULL di sebelah kiri.

4. INNER JOIN dengan kondisi

The screenshot shows the SSMS interface with the 'Result' tab selected. The results grid displays two columns: 'Nama' and 'Total_harga'. The data is as follows:

Nama	Total_harga
Budi	250000
James	120000
Karen	260000
Lini	140000
Kesya	130000
Winner	210000
Juna	112000
Hana	101000
Tina	190000

Penjelasan:

Pada query nomor 4, sama seperti **INNER JOIN**, tetapi hanya menampilkan baris dimana Total_harga lebih dari 100000.

5. LEFT JOIN dengan kondisi

The screenshot shows the SSMS interface with the 'Result' tab selected. The results grid displays two columns: 'Nama' and 'Total_harga'. The data is as follows:

Nama	Total_harga
Aisyah	100000

Penjelasan:

Pada query nomor 5, sama seperti **LEFT JOIN**, tetapi hanya menampilkan baris dimana Nama di Tabel Pelanggan dimulai dengan huruf A.

6. RIGHT JOIN dengan kondisi

Nama	Total_harga
Budi	250000
James	120000
Karen	260000
Lini	140000
Kesya	130000
Winner	210000
Juna	112000
Hana	101000
Tina	190000

Penjelasan:

Pada query nomor 6, ini sama seperti **RIGHT JOIN**, tetapi hanya menampilkan baris dimana Total_harga lebih dari 100000.

7. INNER JOIN dengan pengurutan hasil

Nama	Total_harga
Karen	260000
Budi	250000
Winner	210000
Tina	190000
Lini	140000
Kesya	130000
James	120000
Juna	112000
Hana	101000
Citra	100000
Aisyah	100000

Penjelasan:

Pada query nomor 7, sama seperti **INNER JOIN**, tetapi hasilnya diurutkan berdasarkan Total_harga dalam urutan **DESCENDING** (menurun).

8. LEFT JOIN dengan pengurutan hasil

Nama	Total_harga
Aisyah	100000
Budi	250000
Citra	100000
Hana	101000
James	120000
Juna	112000
Karen	260000
Kesya	130000
Lini	140000
Tina	190000
Winner	210000

Penjelasan:

Pada query nomor 8, sama seperti **RIGHT JOIN**, tetapi hasilnya diurutkan berdasarkan Nama dalam urutan **ASCENDING** (default).

9. RIGHT JOIN dengan pengurutan hasil

Nama	Total_harga
Aisyah	100000
Citra	100000
Hana	101000
Juna	112000
James	120000
Kesya	130000
Lini	140000
Tina	190000
Winner	210000
Budi	250000
Karen	260000

Penjelasan:

Pada query nomor 9, sama seperti **RIGHT JOIN**, tetapi hasilnya diurutkan berdasarkan Total_harga dalam urutan **ASCENDING** (default).

10. INNER JOIN dengan pengelompokan hasil

Nama	Jumlah_Transaksi
Aisyah	1
Budi	1
Citra	1
Hana	1
James	1
Juna	1
Karen	1
Kesya	1
Lini	1
Tina	1
Winner	1

Penjelasan:

Pada query nomor 10, sama seperti **INNER JOIN**, tetapi hasilnya dikelompokkan berdasarkan Nama di tabel Pelanggan, dan menampilkan jumlah transaksi untuk setiap pelanggan.

- d) Buat 5 penggunaan subquery

Jawab:

```
198 -- Buat 5 penggunaan subquery
199
200 -- 1. Menampilkan nama pelanggan yang melakukan transaksi dengan total harga tertinggi
201 SELECT 'Nama' FROM 'Pelanggan' WHERE 'ID_Pelanggan' = (SELECT 'Pelanggan_id' FROM 'Transaksi_Penjualan' ORDER BY 'Total_harga' DESC LIMIT 1);
202
203 -- 2. Menampilkan nama produk yang memiliki harga lebih rendah dari harga produk tertinggi
204 SELECT 'Nama' FROM 'Produk' WHERE 'Harga' < (SELECT MAX('Harga') FROM 'Produk');
205
206 -- 3. Menampilkan nama karyawan yang tidak melakukan layanan
207 SELECT 'Nama' FROM 'Karyawan' WHERE 'Karyawan_id' NOT IN (SELECT DISTINCT 'Karyawan_id' FROM 'Layanan');
208
209 -- 4. Menampilkan nama layanan yang harganya lebih tinggi dari rata-rata harga layanan
210 SELECT 'Nama' FROM 'Layanan' WHERE 'Harga' > (SELECT AVG('Harga') FROM 'Layanan');
211
212 -- 5. Menampilkan nama pelanggan yang tidak melakukan transaksi penjualan
213 SELECT 'Nama' FROM 'Pelanggan' WHERE 'ID_Pelanggan' NOT IN (SELECT DISTINCT 'Pelanggan_id' FROM 'Transaksi_Penjualan');
214
```

Hasil:

1. Menampilkan nama pelanggan yang melakukan transaksi dengan total harga tertinggi

Nama
Karen

Penjelasan:

Pada query nomor 1, query ini mencari pelanggan yang melakukan transaksi dengan total harga tertinggi. Subquery di dalamnya mencari Pelanggan_id dari tabel Transaksi_Penjualan dengan total harga tertinggi (Total_harga), dan query utamanya mencari Nama pelanggan dengan ID_Pelanggan yang sama.

2. Menampilkan nama produk yang memiliki harga lebih rendah dari harga produk tertinggi

Nama
Cream
Shampoo
Conditioner
Face Wash
Body Wash
Minyak Rambut
Anti Aging
Conditioner + Minyak Rambut
Face Wash + Body Wash

Penjelasan:

Pada query nomor 2, query ini mencari semua produk yang harganya (Harga) lebih rendah dari harga produk tertinggi. Subquery di dalamnya mencari harga tertinggi dari tabel Produk.

3. Menampilkan nama karyawan yang tidak melakukan layanan

Nama

Penjelasan:

Pada query nomor 3, query ini mencari semua karyawan yang tidak terdaftar dalam tabel Layanan. Subquery di dalamnya mencari semua Karyawan_id yang unik dari tabel Layanan, dan query utamanya mencari Nama Karyawan yang Karyawan_id nya tidak ada diantara mereka.

4. Menampilkan nama layanan yang harganya lebih tinggi dari rata-rata harga layanan

Nama
Makeup Untuk Acara Khusus

Penjelasan:

Pada query nomor 4, query ini mencari semua layanan yang harganya (Harga) lebih tinggi dari rata-rata harga layanan. Subquery di dalamnya mencari rata-rata harga dari tabel Layanan.

5. Menampilkan nama pelanggan yang tidak melakukan transaksi penjualan

1 Result	2 Profiler	3 Messages
(Read Only)		
Nama		

Penjelasan:

Pada query nomor 5, query ini mencari semua pelanggan yang tidak terdaftar dalam tabel Transaksi_Penjualan. Subquery di dalamnya mencari semua Pelanggan_id yang unik dari tabel Transaksi_Penjualan, dan query utamanya mencari Nama Pelanggan yang ID_Pelanggan nya tidak ada diantara mereka.

Tautan Video:

Selamat pagi ibu dan kakak.

Perkenalkan nama saya Agus Pranata Marpaung dengan NIM 13323033.

Saya telah mengirim video presentasi Laporan Proyek Ke YouTube.

Berikut linknya bu/kak:

<https://youtu.be/lSwWBNVwNKE?si=eJLoOesRzMxK38d2>

Terima kasih bu/kak 🌟