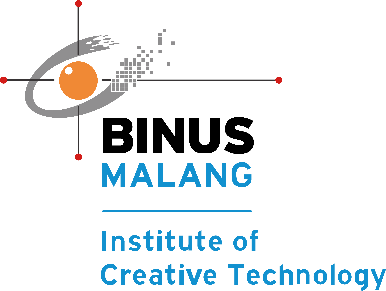
**JATIM PARK 1 TICKETING DATABASE SYSTEM**

**LAPORAN PROYEK AKHIR**

COMP6852004 – Database Technology

KELAS LB-20



Oleh:

2602166640 – SEKAR AYU NADITA

2602085920 – RAHEL ANDREA ARABELLA

2602204310 – NABILA VEBYKA ZHAFIRA

2602098614 – REGINA PATRICIA HARTAWAN

Semester Ganjil 2023/2024

MALANG

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc153833555)

[BAB I 3](#_Toc153833556)

[1.1 Latar Belakang 3](#_Toc153833557)

[1.2 Rumusan Masalah 4](#_Toc153833558)

[1.3 Tujuan 4](#_Toc153833559)

[1.4 Manfaat 4](#_Toc153833560)

[BAB II 6](#_Toc153833561)

[2.1 ERD 6](#_Toc153833562)

[2.2 XAMPP 7](#_Toc153833563)

[2.3 MySQL 7](#_Toc153833564)

[BAB III 8](#_Toc153833565)

[3.1 Rancangan Database (ERD) 8](#_Toc153833566)

[3.2 List of Attributes 9](#_Toc153833567)

[3.3 List Pertanyaan dan Jawaban 14](#_Toc153833568)

[BAB IV 28](#_Toc153833569)

[DAFTAR PUSTAKA 29](#_Toc153833570)

# BAB I

**PENDAHULUAN**

## 1.1 Latar Belakang

Pariwisata merupakan kegiatan yang berkaitan dengan perjalanan dan kunjungan ke destinasi tertentu dengan tujuan untuk liburan, rekreasi, dan lain-lain. Pariwisata juga melibatkan berbagai aspek, seperti jasa akomodasi, makanan, transportasi, dan aspek lainnya. Pariwisata di Indonesia, khususnya Jawa Timur telah berkembang dari tahun ke tahun dan menjadi destinasi yang diminati oleh berbagai kalangan pengunjung. Salah satu objek wisata yang ada di Jawa Timur adalah Jatim Park 1.

Jawa Timur Park 1 atau biasa dikenal dengan Jatim Park 1 merupakan salah satu destinasi wisata yang terkenal di Jawa Timur yang memiliki wahana permainan seru bagi anak-anak dan orang dewasa . Hal ini terbukti dengan peningkatan jumlah pengunjungnya dari tahun ke tahun. Namun karena popularitasnya yang semakin meningkat, objek wisata ini dihadapkan pada tekanan untuk mengoptimalkan pengalaman pengunjung. Adanya peningkatan jumlah pengunjung memberikan tantangan dalam manajemen antrian, pengelolaan kapasitas wahana, penyediaan informasi yang kurang jelas, antrian yang panjang, kesulitan dalam pemesanan tiket dan paket, serta pengelolaan staff dan operasional yang kompleks sehingga menjadi beberapa masalah yang perlu diatasi.

Oleh karena itu, kami membuat sebuah sistem database yang dapat memberikan keunggulan kompetitif bagi Jatim Park 1. Sistem yang memadai ini dapat memberikan aksesibilitas yang lebih baik, memberikan informasi yang lengkap, mengatur klasifikasi wahana, serta memberikan analisis data untuk perbaikan layanan yang berkelanjutan bagi Jatim Park 1. Tidak hanya itu, pengelolaan staff dan operasional juga menjadi aspek krusial dalam kesuksesan objek wisata tersebut. Maka sistem ini juga akan membantu penjadwalan staff yang efisien serta pengelolaan tugas harian untuk meningkatkan kualitas layanan kepada pengunjung.

Dengan penekanan pada pengelolaan pengunjung, efisiensi operasional, dan penerapan teknologi terkini, sistem database diharapkan dapat memberikan solusi yang baik bagi Jatim Park 1 secara responsif, terintegrasi, dan inovatif dalam mengatasi tantangan yang dihadapi serta meningkatkan kualitas layanan dan pengalaman pengunjung secara keseluruhan.

## 1.2 Rumusan Masalah

Adapun rumusan masalah adalah sebagai berikut :

1. Bagaimana cara mengelola data dan riwayat pengunjung dengan efisien untuk memastikan efisiensi kualitas layanan yang lebih baik lagi?
2. Bagaimana cara menyusun informasi terkait variasi dan kategori paket tiket secara terstruktur untuk memberikan pilihan yang lebih luas kepada pengunjung?
3. Bagaimana cara mengintegrasikan sistem pembayaran agar dapat melacak transaksi dan pembayaran tiket dengan aman dan efisien?
4. Apa yang harus dilakukan dalam mengelola data wahana agar dapat memastikan pengelolaan yang tepat?

## 1.3 Tujuan

Berdasarkan latar belakang dan rumusan masalah tersebut, proyek akhir ini memiliki tujuan sebagai berikut :

1. Database Jatim Park 1 Ticketing ini dibuat dengan tujuan sebagai media bagi pengusaha untuk melakukan pencatatan data dan riwayat pengunjung tempat wisata.
2. Database Jatim Park 1 Ticketing bertujuan untuk efisiensi manajemen informasi tiket dan transaksi untuk memudahkan ketersediaan dan pemesanan tiket.
3. Database Jatim Park 1 Ticketing dibuat untuk mendukung analisis data guna memberikan wawasan tentang tren pengunjung, preferensi tiket, dan efektivitasnya dalam menunjang strategi promosi tempat wisata.

## 1.4 Manfaat

1. Membantu dalam pengelolaan antrian pengunjung, kapasitas paket dan wahana, pengaturan staff sehingga meningkatkan efisiensi operasional Jatim Park 1.
2. Memberikan aksesibilitas yang lebih baik kepada pengunjung terkait informasi variasi paket dan kategori wahana di dalam Jatim Park 1.
3. Menyediakan analisis data yang dapat digunakan untuk mengetahui preferensi pengunjung dan keterkaitannya dalam meningkatkan efektivitas strategi promosi.
4. Memudahkan integrasi sistem pembayaran dan keuangan dari transaksi yang telah dilakukan oleh pengunjung Jatim Park 1.
5. Memberikan pemahaman lebih baik mengenai sistem operasional wahana dan staff yang bekerja di Jatim Park 1.

# BAB II

**TINJAUAN PUSTAKA**

## 2.1 ERD

Diagram Hubungan Entitas (Entity Relationship Diagram atau ERD) adalah representasi grafis yang digunakan dalam pengembangan basis data untuk menggambarkan hubungan antara berbagai entitas atau data. Ini merupakan alat bantu penting dalam proses pembuatan basis data, memberikan gambaran visual tentang bagaimana data saling terkait dan berinteraksi. Sumber Dosen Online tahun 2017 menyebutkan bahwa ERD membantu dalam merancang struktur basis data dan memberikan pemahaman yang jelas tentang bagaimana database yang akan dibuat akan beroperasi. Di dalam ERD terdapat 3 komponen, diantaranya sebagai berikut.

1. Entitas

Entitas dalam konteks basis data merujuk pada objek yang menjadi fokus perhatian. Entitas ini bisa berupa manusia, tempat, benda, atau kondisi yang terkait dengan data yang diperlukan dalam database. Simbol untuk entitas biasanya berbentuk persegi panjang.

1. Atribut

Atribut merujuk pada informasi yang ada dalam suatu entitas. Setiap entitas harus memiliki primary key sebagai ciri khasnya, dan atribut deskriptif. Atribut ini umumnya terdapat dalam tabel entitas atau bisa juga terpisah dalam tabel sendiri. Simbol untuk atribut biasanya berbentuk elips.

1. Relasi

Relasi dalam ERD adalah keterhubungan antara dua atau lebih entitas. Simbol relasi biasanya berbentuk belah ketupat. Terdapat beberapa macam relasi dalam ERD, antara lain:

1. One-to-One: Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.
2. One-to-Many: Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain.
3. Many-to-Many: Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

## 2.2 XAMPP

XAMPP merupakan perangkat lunak gratis yang kompatibel dengan berbagai sistem operasi. Ini merupakan kumpulan dari beberapa program yang berperan sebagai server independen (localhost). XAMPP mencakup Apache HTTP Server, MySQL database, dan penerjemah bahasa yang dibuat dengan PHP dan Perl.

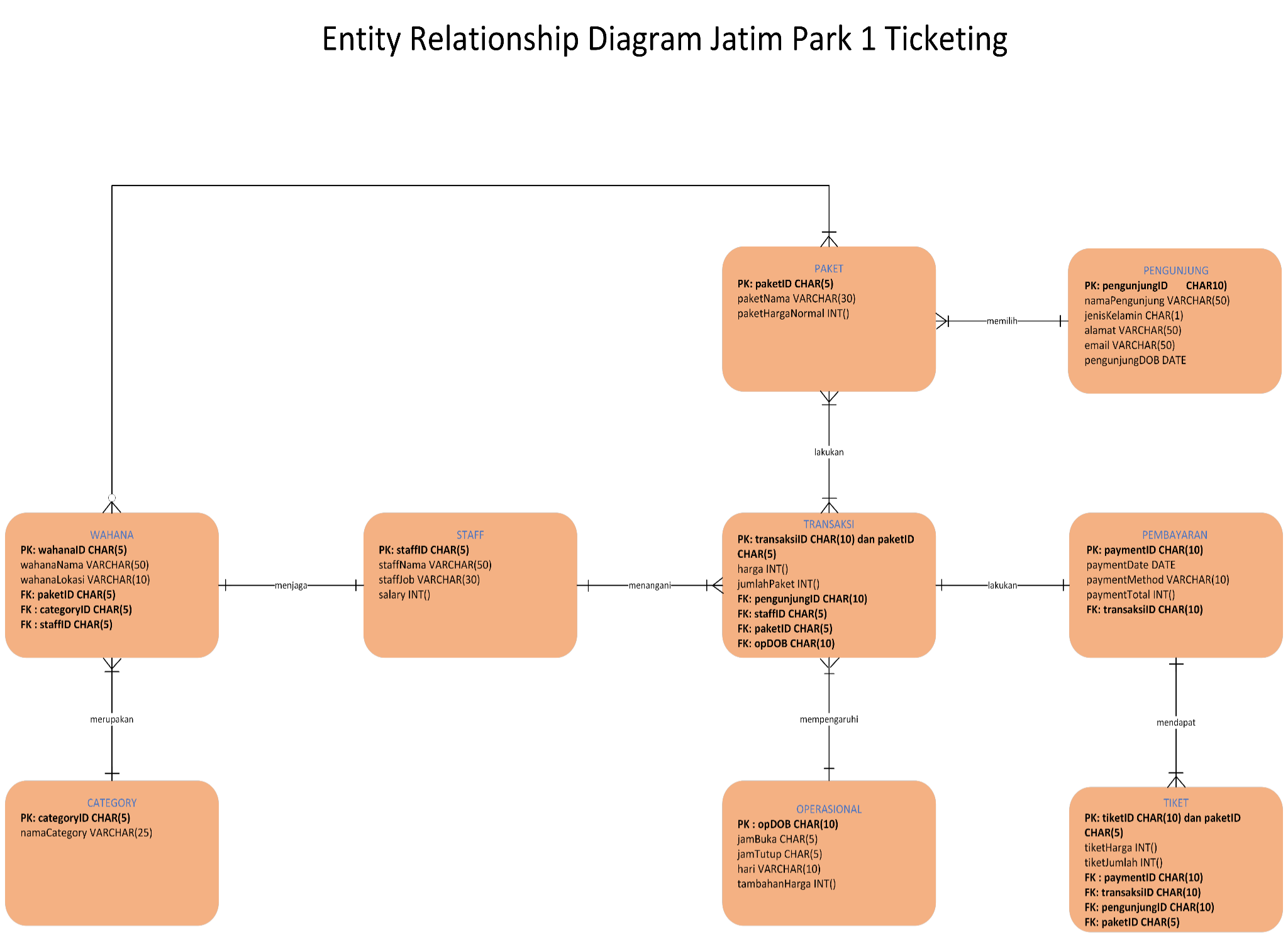
## 2.3 MySQL

MySQL pada dasarnya merupakan derivatif dari salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (Structured Query Language). SQL merupakan suatu konsep operasi basis data, terutama untuk pemilihan atau seleksi serta pemasukan data. Hal ini memungkinkan pengelolaan data dilakukan dengan mudah secara otomatis, sebagaimana dijelaskan oleh Alan Nur Aditya (2011:61).

# BAB III

**HASIL DAN PEMBAHASAN**

## 3.1 Rancangan Database (ERD)

****

Berdasarkan ERD diatas, terdapat 9 tabel yang mencakup ERD Jatim Park 1 diantaranya tabel pengunjung, tabel paket, tabel operasional, tabel wahana, tabel staff, tabel transaksi, tabel pembayaran, tabel tiket, dan tabel category. Pertama, terdapat tabel pengunjung yang memiliki relationship ke tabel paket. Disini, pengunjung bisa memilih paket yang tersedia. Tabel transaksi dipengaruhi oleh tabel operasional yang nantinya akan menentukan harga tambahan dari paket yang dipilih user. Lalu, lakukan transaksi yang akan ditangani oleh staff untuk dilanjutkan ke pembayaran sehingga pengunjung akan mendapatkan tiket. Disini, staff juga memiliki relasi dengan wahana yaitu staff akan menjaga keamanan para pengunjung saat menaiki wahana dengan kategori tertentu.

## 3.2 List of Attributes

**Tabel Operasional**

1. PK : opDOB CHAR(10)
2. jamBuka CHAR(5)
3. jamTutup CHAR(5)
4. hari VARCHAR(10)
5. tambahanHarga INT()

Constraint :

1. opDOB : yyyy-mm-dd
2. jamBuka dan jamTutup : hh:mm

**Tabel Paket**

1. PK : paketID CHAR(5)
2. paketNama VARCHAR(30)
3. paketHargaNormal INT()

Constraint :

1. paketID : PA[num][num][num], mis : PA001

**Tabel Pengunjung**

1. PK : pengunjungID CHAR(10)
2. namaPengunjung VARCHAR(50)
3. jenisKelamin CHAR(1)
4. alamat VARCHAR(50)
5. email VARCHAR(50)
6. pengunjungDOB DATE

Constraint :

1. pengunjungID : num sebanyak 10 kali, mis : 1234567890
2. email : dimulai dengan karakter alphanumeric (tidak boleh symbol), selanjutnya dapat menggunakan #, ., \_, dan tentunya alphanumeric, lalu simbol @ yang diikuti karakter alphanumeric maupun titik
3. jenisKelamin : P atau L

**Tabel Wahana**

1. PK : wahanaID CHAR(5)
2. wahanaNama VARCHAR(50)
3. wahanaLokasi VARCHAR(10)
4. ***FK : paketID CHAR(5)***
5. ***FK : categoryID CHAR(5)***
6. ***FK : staffID CHAR(5)***

Constraint :

* + - 1. wahanaID : WA[num][num][num], mis : WA001
      2. wahanaLokasi : Zone [A-Z], mis : Zone A atau Zone C, dll
      3. paketID : PA[num][num][num], mis : PA001

**Tabel Staff**

1. PK : staffID CHAR(5)
2. staffNama VARCHAR(50)
3. staffJob VARCHAR(30)
4. salary INT()

Constraint :

* + - 1. staffID : ST[num][num][num], mis : ST001

**Tabel Transaksi**

1. PK : transaksiID CHAR(10)
2. PK : paketID CHAR(5)
3. harga INT()
4. jumlahPaket INT()
5. ***FK : pengunjungID CHAR(10)***
6. ***FK : staffID CHAR(5)***
7. ***FK : paketID CHAR(5)***
8. ***FK : opDOB CHAR(10)***

Constraint :

* + - 1. transaksiID : num sebanyak 10 kali, mis : 1234567890
      2. pengunjungID : num sebanyak 10 kali, mis : 1234567890
      3. staffID : ST[num][num][num], mis : ST001
      4. paketID : PA[num][num][num], mis : PA001
      5. opDOB : yyyy-mm-dd

**Tabel Pembayaran**

1. PK : paymentID CHAR(10)
2. paymentDate DATE
3. paymentMethod VARCHAR(10)
4. paymentTotal INT()
5. ***FK: transaksiID CHAR(10)***

Constraint :

* + - 1. paymentID : alphanumeric sebanyak 10 kali, mis : ABC1234567
      2. transaksiID : num sebanyak 10 kali, mis : 1234567890

**Tabel Category**

1. PK : categoryID CHAR(5)
2. namaCategory VARCHAR(25)

Constraint :

* + - 1. categoryID : CA[num][num][num], mis : CA001

**Tabel Tiket**

1. PK : tiketID CHAR(10)
2. PK : paketID CHAR(5)
3. tiketHarga INT()
4. tiketJumlah INT()
5. ***FK : paymentID CHAR(10)***
6. ***FK : transaksiID CHAR(10)***
7. ***FK : pengunjungID CHAR(10)***
8. ***FK : paketID CHAR(5)***

Constraint :

* + - 1. tiketID : num sebanyak 10 kali, mis : 1234567890
      2. paymentID : alphanumeric sebanyak 10 kali, mis : ABC1234567
      3. transaksiID : num sebanyak 10 kali, mis : 1234567890
      4. pengunjungID : num sebanyak 10 kali, mis : 1234567890
      5. paketID : PA[num][num][num], mis : PA001

**Store Procedure doTransaksi**

DELIMITER //

CREATE PROCEDURE doTransaksi(

IN ID\_Transaksi CHAR(10),

IN ID\_Pengunjung CHAR(10),

IN ID\_Staff CHAR(5),

IN ID\_Paket CHAR(5),

IN opDate CHAR(10),

IN jumlah\_paket INT

)

BEGIN

-- Memasukkan data ke dalam tabel transaksi

INSERT INTO transaksi

VALUES (ID\_Transaksi, 1, jumlah\_paket, ID\_Pengunjung, ID\_Staff, ID\_Paket, opDate);

-- Menghitung total harga dari transaksi dan memperbarui kolom harga pada tabel transaksi

UPDATE transaksi T

SET T.harga = (

SELECT ((P.paketHargaNormal + O.tambahanHarga) \* jumlah\_paket)

FROM operasional O

JOIN paket P ON T.paketID = P.paketID

WHERE O.opDOB = opDate AND P.paketID = ID\_Paket

)

WHERE T.transaksiID = ID\_Transaksi AND T.paketID = ID\_Paket;

END //

DELIMITER ;

**Store Procedure doPembayaran**

DELIMITER //

CREATE PROCEDURE doPembayaran(

IN ID\_Bayar CHAR(10),

IN ID\_Transaksi CHAR(10)

)

BEGIN

INSERT INTO pembayaran

VALUES (ID\_Bayar,

(SELECT DISTINCT(T.opDOB) FROM transaksi T WHERE T.transaksiID = ID\_Transaksi),

'Online',

(SELECT SUM(T.harga) FROM transaksi T WHERE T.transaksiID = ID\_Transaksi GROUP BY T.transaksiID),

ID\_Transaksi);

END //

DELIMITER ;

**Store Procedure Tiket**

DELIMITER //

CREATE PROCEDURE doTiket(

IN ID\_Tiket CHAR(10),

IN ID\_Pembayaran CHAR(10),

IN ID\_Transaksi CHAR(10),

IN ID\_Pengunjung CHAR(10),

IN ID\_Paket CHAR(5) )

BEGIN

INSERT INTO tiket VALUES

(ID\_Tiket, 1, 1, ID\_Pembayaran, ID\_Transaksi, ID\_Pengunjung , ID\_Paket);

UPDATE tiket Ti

SET Ti.tiketHarga = (SELECT T.harga

FROM transaksi T

WHERE T.paketID = ID\_Paket AND T.transaksiID = ID\_Transaksi),

Ti.tiketJumlah = (SELECT T.jumlahPaket

FROM transaksi T

WHERE T.paketID = ID\_Paket AND T.transaksiID = ID\_Transaksi)

WHERE Ti.paketID = ID\_Paket AND Ti.transaksiID = ID\_Transaksi;

END //

DELIMITER ;

**View ListWahanaSesuaiPaket**

CREATE VIEW ListWahanaSesuaiPaket

AS

SELECT P.paketID AS ID\_Paket, P.paketNama AS Nama\_Paket, GROUP\_CONCAT(C.namaCategory SEPARATOR '\n') AS Nama\_Kategori, GROUP\_CONCAT(W.wahanaNama SEPARATOR '\n') AS List\_Wahana, GROUP\_CONCAT(W.wahanaLokasi SEPARATOR '\n') AS Lokasi\_Wahana

FROM paket P JOIN wahana W ON W.paketID = P.paketID

JOIN category C ON W.categoryID = C.categoryID

GROUP BY P.paketID

**View ListPenjualanPaket**

CREATE VIEW ListPenjualanPaket

AS

SELECT P.paketID AS ID\_Paket, P.paketNama AS Nama\_Paket, SUM(T.jumlahPaket) AS Jumlah\_Paket, SUM(T.harga) AS Total\_Harga

FROM paket P JOIN transaksi T ON P.paketID = T.paketID

GROUP BY P.paketID, P.paketNama

ORDER BY Jumlah\_Paket DESC

## 3.3 List Pertanyaan dan Jawaban

<https://binusianorg-my.sharepoint.com/personal/adeva_wibawa_binus_ac_id/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fadeva%5Fwibawa%5Fbinus%5Fac%5Fid%2FDocuments%2FSoal%20Quiz%20Database%20Technology%20BD20%2FFILE%20PERTANYAAN%20QUIZ%20DATABASE%20TECHNOLOGY%20BD20%2FKelompok%20Badutz>

Tampilkan daftar wahana beserta lokasinya dan nama kategorinya

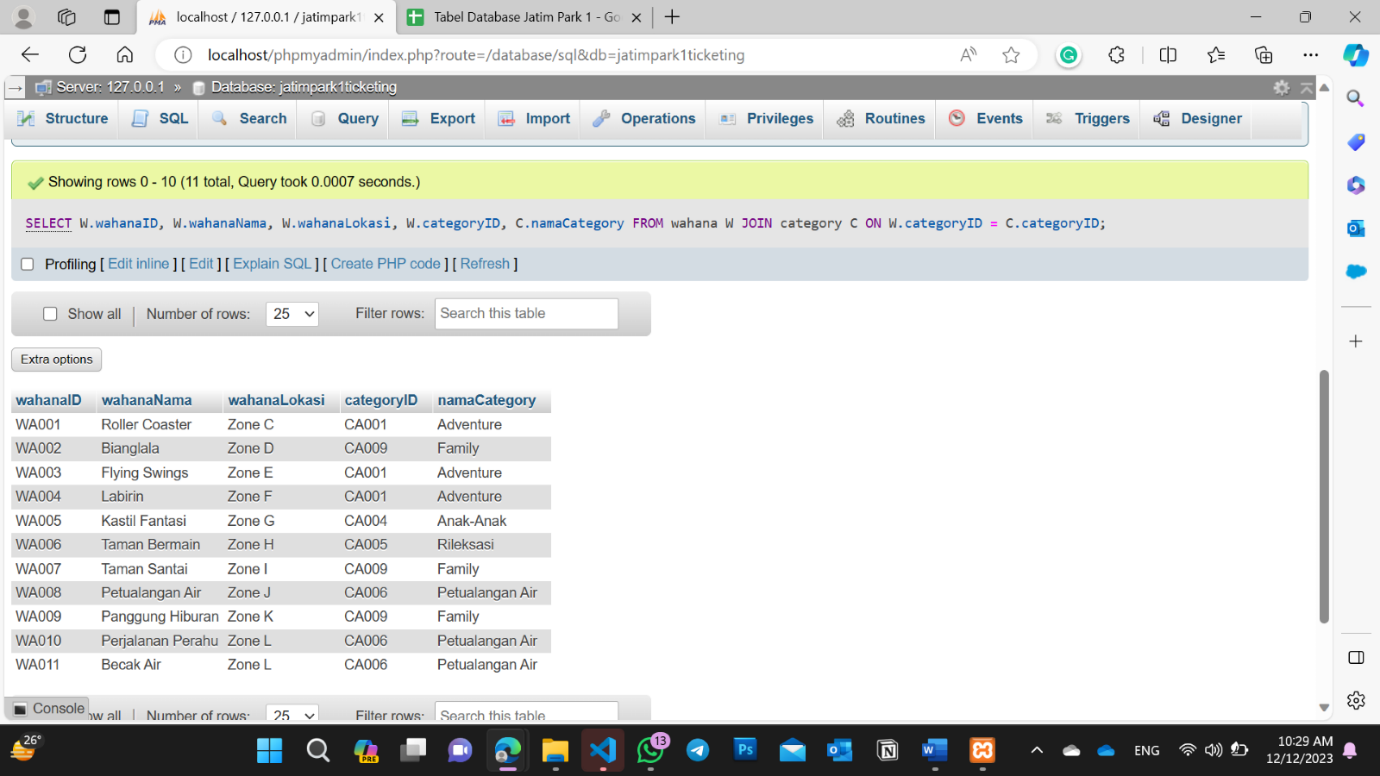
**(SELECT, FROM , JOIN)**

**(40 Point Pertanyaan kelompok).**

**Jawaban :**

SELECT W.wahanaID, W.wahanaNama, W.wahanaLokasi, W.categoryID, C.namaCategory

FROM wahana W JOIN category C ON W.categoryID = C.categoryID



Penjelasan :

Kita akan select semua detail wahana, mulai dari ID, nama, lokasi, categoryID dan nama category yang kita dapat dari join tabel wahana dan tabel category, dimana penghubungnya merupakan categoryID nya, sehingga keluar hasil dimana kita bisa tahu juga categorynama nya apa berdasarkan join ini

1. Tampilkan nama pengunjung, tanggal lahir, dan Alamat pengunjung yang telah melakukan pembayaran menggunakan metode ‘Online’.

**(SELECT , JOIN, WHERE)**

**(5 point pertanyaan individu) Rahel**

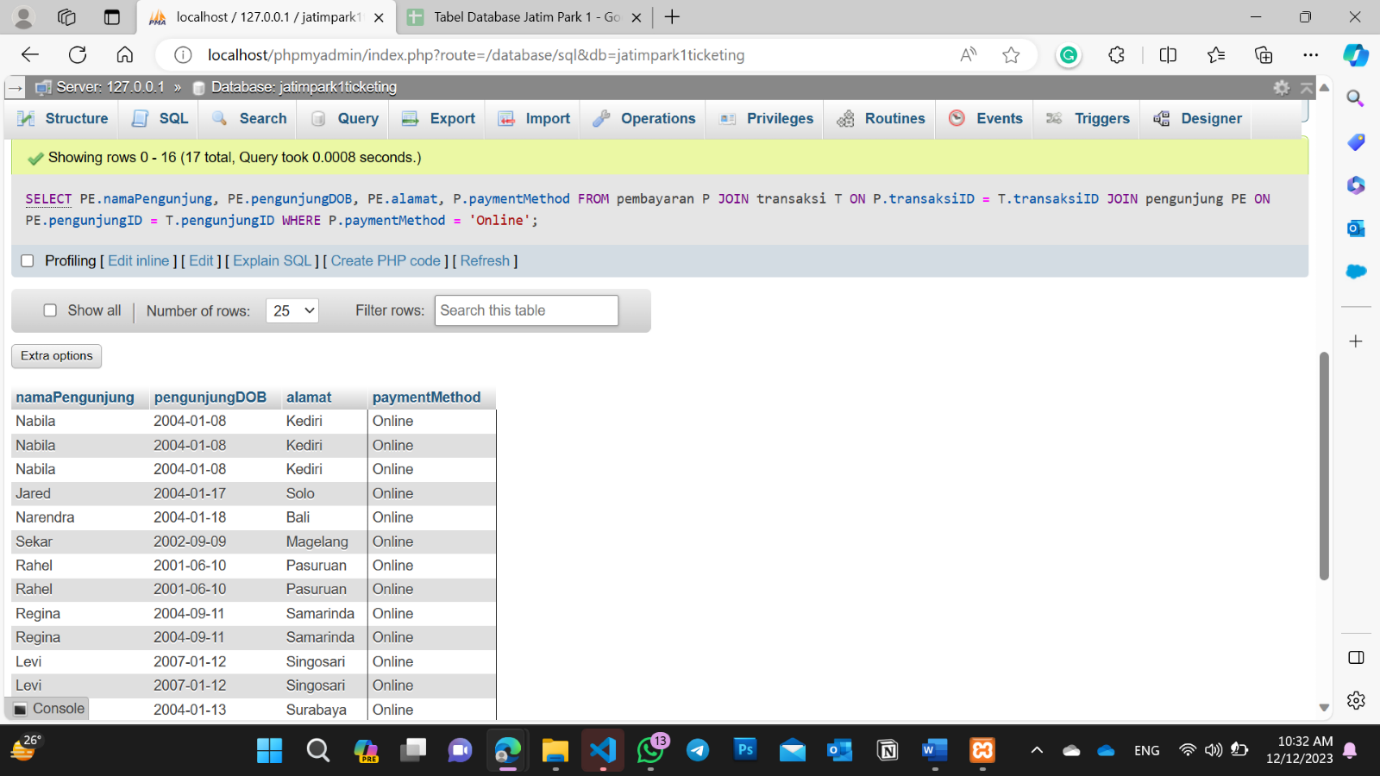
**Jawaban :**

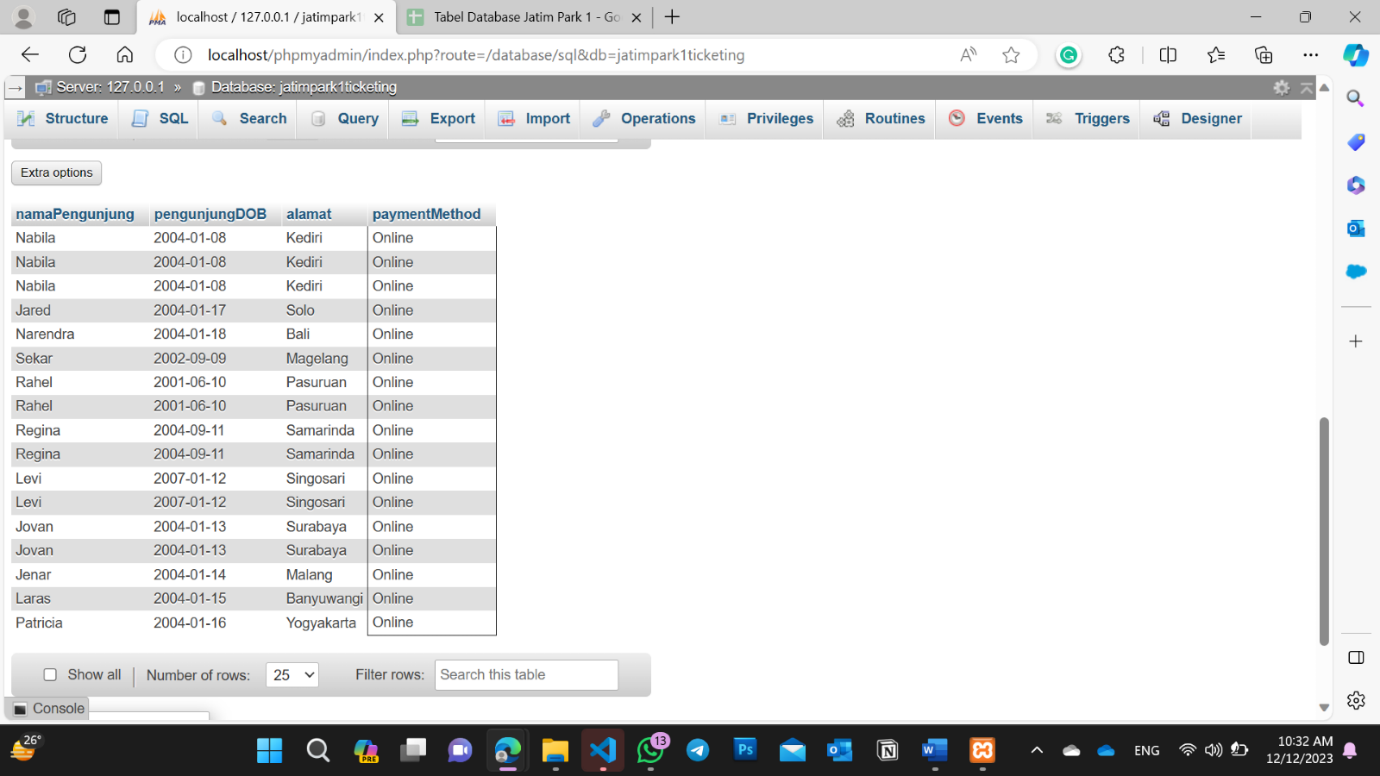
SELECT PE.namaPengunjung, PE.pengunjungDOB, PE.alamat, P.paymentMethod

FROM pembayaran P JOIN transaksi T ON P.transaksiID = T.transaksiID

JOIN pengunjung PE ON PE.pengunjungID = T.pengunjungID

WHERE P.paymentMethod = 'Online'





Penjelasan : Kita akan select semua info mengenai pengunjung, mulai dari nama, DOB, alamat, lalu kita akan mengeluarkan paymentMethod, jadi kita join transaksi karena kita akan mencari pengunjung yang sudah melakukan transaksi, lalu kita hubungkan ke tabel pembayaran, karena kita ingin mengetahui metode pembayaran mereka, setelah itu, kita join tabel juga ke pengunjung karena kita ingin mengetahui siapa yang melakukan transaksi dan pembayaran tersebut. Setelah itu, kita beri kondisi, bahwa yang dikeluarkan adalah detail dengan paymentMethodnya ‘Online’

1. Tampilkan jumlah total pengunjung berdasarkan jenis kelamin

**(SELECT, COUNT, GROUP BY)**

**(5 point pertanyaan individu) (SEKAR AYU NADITA)**

**Jawab :**

**SCREENSHOOT :**

A screenshot of a computer

Description automatically generated

QUERY

SELECT COUNT(DISTINCT p.pengunjungID) as JumlahPengunjung, p.jenisKelamin FROM pengunjung p

WHERE p.jenisKelamin = 'P' OR p.jenisKelamin = 'L'

GROUP BY (p.jenisKelamin)

Penjelasan:

Pertama melakukan select dengan isian pengunjung ide dan jenis kelamin. Pengaplikasian COUNT(DISTINT p.pengunjung ID) digunakan untuk menghitung banyak total pengunjung berdasarkan ketentuan jenis kelaminnya P(Perempuan) atau L(Laki-laki). Kemudian di group berdasarkan jenis kelamin pengunjung.

1. Tampilkan nama staff, pekerjaan, dan gaji yang bekerja di wahana dengan ketentuan ID “WA001”.

**(SELECT , JOIN, WHERE)**

**(5 point pertanyaan individu) Nabila**

A screenshot of a computer

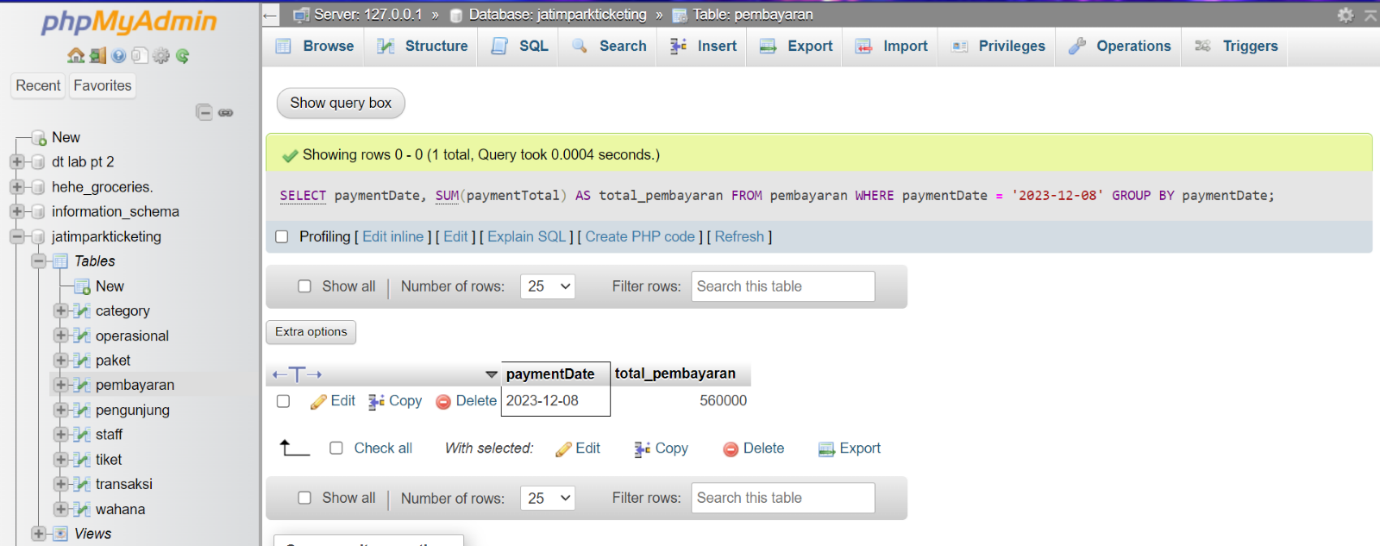
Description automatically generated

Menggunakan query select untuk menampilkan staffNama, staffJob, dan salary. Di join dengn tabel wahana untuk menampilkan staff yang ada di wahana WA001 saja.

1. Tampilkan daftar pembayaran yang dilakukan pada tanggal tertentu beserta total pembayarannya.

**(SELECT, SUM, GROUP BY).**

**(5 point pertanyaan individu) Regina**



**QUERY :**

SELECT paymentDate, SUM(paymentTotal) AS total\_pembayaran

FROM pembayaran

WHERE paymentDate = '2023-12-08'

GROUP BY paymentDate

**Penjelasan Query :**

Menggunakan query ‘SELECT’ untuk menampilkan ‘paymentDate’, lalu dijumlahkan total pembayaran dari ‘paymentTotal’ menggunakan query ‘SUM’ sebagai ‘total\_pembayaran’. Dan diambil dari table ‘pembayaran’ dimana disini bisa milih salah satu tanggal paymentnya. Disini saya mengambil **paymentDate = ‘2023-12-08'.** Lalu dikelompokkan berdasarkan ‘paymentDate’ dengan menggunakan query ‘GROUP BY’

1. Tampilkan jumlah tiket yang terjual untuk setiap transaksi yang dilakukan.

**(SELECT , SUM, JOIN, GROUP BY)**

**(5 point pertanyaan individu) Nabila**

A screenshot of a computer

Description automatically generated

Kita akan menampilkan sum tiket.tiketJumlah untuk menjumlahkan jumlah tiket pada tiap transaksi pengunjung dan transaksiID yang mana disini mewakilkan pengunjung. Join dengan tabel transaksi dengan transaksiID sebagai parameter. Lalu group by berdasarkan transaksiID.

1. Tampilkan daftar paket yang memiliki harga di atas rata-rata harga paket.

**(SELECT, WHERE, AVG).**

**(5 point pertanyaan individu) Rahel**

**Jawaban :**

SELECT paketID, paketNama, paketHargaNormal

FROM paket

WHERE paketHargaNormal > (SELECT AVG(paketHargaNormal) FROM paket)

A screenshot of a computer

Description automatically generated

Dimana rata-rata paket :

A screenshot of a computer

Description automatically generated

Penjelasan :

Kita akan select paketID, paketNama, paketHargaNormal, namun dengan kondisi bahwa paket yang akan ditampilkan tersebut harus memiliki paketHargaNormal yang lebih besar dari average semua paketHargaNormal yang ada.

Oleh karena itu, kita akan select paketID, paketNama, paketHargaNormal, yang merupakan detail dari tabel paket, lalu kita beri kondisi dimana akan menampilkan paketHargaNormal yang lebih dari rata-rata saja (menggunakan WHERE paketHargaNormal > (SELECT AVG(paketHargaNormal) FROM paket))

1. Tampilkan jumlah transaksi yang dilakukan oleh setiap pengunjung.

**(SELECT, FROM , GROUP BY).**

**(5 point pertanyaan individu) Regina**

A screenshot of a computer

Description automatically generated

**QUERY :**

SELECT pengunjungID, COUNT(transaksiID) AS jumlah\_transaksi

FROM transaksi

GROUP BY pengunjungID

**Penjelasan Query :**

Menggunakan Query ‘SELECT’ untuk menampilkan ID pengunjung, lalu menghitung jumlah transaksi dari ‘transaksiID’ sebagai ‘jumlah\_transaksi’, yang berasal dari table ‘transaksi’ menggunakan ‘FROM’. Lalu dikelompokkan berdasarkan ‘pengunjungID’ menggunakan query ‘GROUP BY’.

1. Tampilkan daftar transaksi beserta nama pengunjung, nama staff, dan total pembayaran. Jika total pembayaran lebih dari 100000, berikan label ‘Pembelian Besar’, dan sebaliknya.

**(SELECT, SUM, CASE, JOIN, LEFT JOIN, GROUP BY)**

**(5 point pertanyaan individu) Regina**

A screenshot of a computer

Description automatically generated

**QUERY :**

SELECT T.transaksiID, P.namaPengunjung, S.staffNama, SUM(T.harga) AS total\_pembayaran,

CASE

WHEN SUM(T.harga) > 100000 THEN 'Pembelian Besar' ELSE 'Pembelian Kecil'

END AS label\_pembelian

FROM transaksi T

JOIN pengunjung P ON T.pengunjungID = P.pengunjungID

LEFT JOIN staff S ON T.staffID = S.staffID

GROUP BY T.transaksiID, P.namaPengunjung, S.staffNama

**Penjelasann Query :**

Disini pertama menggunakan query ‘SELECT’ untuk menampilkan transaksiID dari ‘transaksi’, namaPengunjung dari ‘pengunjung’, staffNama dari ‘staff’, menghitung total pembayaran dari harga dengan tabel ‘Transaksi’ dengan menggunakan query ‘SUM’. Namun ada pengecualian, dimana menggunakan query ‘CASE’ untuk jika total\_pembayaran > 100000, maka dilabel ‘Pembelian Besar’, begitu juga sebaliknya menggunakan query ‘ELSE’. Lalu diakhiri dengan ‘label\_pembelian” menggunakan query ‘END AS’. Dimana berasal dari tabel ‘transaksi’. Lalu menggabungkan tabel pengunjung dan transaksi melalui ‘pengunjungID’ menggunakan query ‘JOIN’. Lalu menggunakan query ‘LEFT JOIN’ untuk menggabungkan tabel staff dan transaksi melalui ‘staffID’ dimana untuk memastikan semua data dari transaksi tetap ditampilkan walaupun tidak ada kaitannya dengan data staff. Terakhir, dikelompokkan berdasarkan ‘T.transaksiID’, ‘P.namaPengunjung’, dan ‘S.staffNama’ menggunakan query ‘GROUP BY’.

1. Tampilkan daftar wahana yang memiliki paket dengan harga di atas rata-rata harga paket.

**(SELECT , DISTINCT, JOIN, WHERE, AVG).**

**(5 point pertanyaan individu) (SEKAR AYU NADITA)**

SCREENSHOOT:

A screenshot of a computer

Description automatically generated

QUERY:

SELECT DISTINCT W.wahanaID, W.wahanaNama, P.paketNama, P.paketHargaNormal

FROM Wahana W

JOIN Paket P ON W.paketID = P.paketID

WHERE P.paketHargaNormal > (SELECT AVG(paketHargaNormal) FROM Paket);

Penjelasan Querry:

Awalnya saya melakukan SELECT DISTINCT untuk memastikan tidak ada duplikasi data wahanaID, wahanaNama yang berasal dari tabel wahana dengan aliases w dan paketNama, paketHargaNormal yang berasal dari tabel paket dengan aliases P. Kedua tabel tersebut kemudian di gabungkan dengan meggunakan JOIN melalui paketID masing” tabel. Kemudian untuk menampilkan sesuai yang diminta soal, saya menggunakan WHERE untuk membuat ketentuan bahwa yang akan ditampilkan hanyalah wahana yang harga paketnya diatas rata” harga paket lainnya yang dapat kita ketahui menggunakan SELECT AVG untuk mengetahui rata-rata harga paket dari tabel paket (Yang secara manual adalah 82.272,72727272. Lalu hasil yag ditampilkan adalah seperti diatas, mengapa hanya paket Couples Retreat saja yang ada disana? Karena untuk paket lainnya yang harganya diatas rata” belum memiliki wahana terdaftar didalamnya.



Pada data wahana yang terdaftar yang memiliki paket diatas rata-rata hanyalah pada paket 7 yaitu Couples Retreat saja yang muncul pada hasil ss an tadi.

1. Tampilkan total gaji yang dikeluarkan per jenis pekerjaan (staffJob). Sertakan juga rata-rata gaji, gaji tertinggi, dan gaji terendah.

**(SELECT, SUM, AVG, MAX, MIN, GROUP BY).**

**(5 point pertanyaan individu) (SEKAR AYU NADITA)**

A screenshot of a computer

Description automatically generated

Querry:

SELECT

    staffJob,

    SUM(salary) AS GajiTotal,

    ROUND(AVG(salary), 1) AS RataRataGaji,

    MAX(salary) AS GajiTertinggi,

    MIN(salary) AS GajiTerendah

FROM Staff

GROUP BY staffJob;

Penjelasan:

Pertama saya melakukan SELECT untuk menampilkan nama Job Staff yang ada pada tabel Staff. Lalu, seusai dengan permintaan soal untuk menampilkan total gaji, rata-rata gaji, gaji maksimal dan minimum tiap jenis pekerjaan maka saya menggunakan:

a. SUM(salary) AS GajiTotal, untuk menampilkan total gaji

b. ROUND(AVG(salary), 1) AS RataRataGaji,  untuk menampilkan Rata-rata gaji berdasarkan total dibagi jumlah karyawan yang bekerja pada jenis pekerjaan dengan pembulatan angka 1 dibelakang koma

c. MAX(salary) AS GajiTertinggi,  untuk menampilkan gaji maksimal

d. MIN(salary) AS GajiTerendah, untuk menampilkan gaji minimum

Kemudian di group by nama job staff untuk masing” select diatas.

1. Tampilkan total jumlah tiket yang terjual per paket. Sertakan juga jumlah transaksi dan rata-rata jumlah tiket per transaksi.

**(SELECT, COUNT, AVG, JOIN, GROUP BY).**

**(5 point pertanyaan individu) Rahel**

**Jawaban :**

SELECT P.paketID, P.paketNama, SUM(T.jumlahPaket) AS total\_tiketTerjualPerPaket, COUNT(T.paketID) AS jumlah\_Transaksi, AVG(T.jumlahPaket) AS rata\_rataJumlahTiketPerTransaksi

FROM transaksi T JOIN paket P ON T.paketID = P.paketID

GROUP BY T.paketID

A screenshot of a computer

Description automatically generated

Penjelasan :

Kita akan mengeluarkan total jumlah tiket yang terjual (berdasarkan paketnya), di transaksi, terdapat berbagai transaksi yang juga membeli paket yang sama, itu mengapa kita juga menghitung jumlah\_transaksi per penjualan paketnya, setelah itu kita menghitung berapa rata-rata orang membeli paket tersebut dalam 1 transaksi.

Sehingga kita akan select paketID serta paketnama nya, lalu kita tampilkan di paket itu, berapa tiket yang terjual (menggunakan **SUM(T.jumlahPaket))**, dan berapa kali transaksi yang terjadi setiap paket tersebut (menggunakan **COUNT(T.paketID))** , dan berapa rata-rata penjualan per transaksinya (menggunakan **AVG(T.jumlahPaket))**

**Kita GROUP BY T.paketID karena kita meghitung jumlah maupun average berdasarkan tipe paket ID nya.**

1. Tampilkan daftar paket dengan harga di atas rata-rata harga paket, sertakan juga paket dengan harga tertinggi dan terendah.

**(SELECT, AVG, MAX, MIN, GROUP BY).**

**(5 point pertanyaan individu) Nabila**

A screenshot of a computer

Description automatically generated



Kita akan menghitung paket mana yang akan ditampilkan dengan kondisi harga diatas rata-rata. Jadi kita menggunakan AVG untuk menghitung rata-rata dari paketHargaNormal lalu menggunakan perbandingan untuk menentukan harga diatas rata-rata. Lalu di query yang kedua kita menggunakan “Max” untuk menampilkan hargaTertinggi dari paketHargaNormal, dan ‘MIN” untuk menampilkan hargaTerendah dari tabel paket.

# BAB IV

**KESIMPULAN**

Pariwisata di daerah Jawa Timur kini semakin berkembang pesat, khususnya objek wisata Jatim Park 1 yang banyak menarik perhatian pengunjung untuk datang berwisata. Namun, seiring pertumbuhan tersebut, terdapat tantangan baru dalam pengelolaan serta penyediaan informasi operasional yang begitu kompleks. Pembuatan proyek database Jatim Park 1 Ticketing ini dapat memudahkan pengelola dalam melakukan pencatatan pengunjung, pemantauan sistem pembayaran, sistem paket dan wahana, serta informasi tiket. Selain itu, database ini dapat dijadikan media untuk melakukan pengelolaan serta meningkatkan efisiensi strategi promosi Jatim Park 1 melalui analisis data yang telah didapat. Secara keseluruhan, kami berharap dengan adanya penekanan pada efisiensi tempat wisata ini melalui sistem database ini, dapat memberikan solusi yang responsif dan inovatif untuk meningkatkan kualitas pelayanan serta pengalaman pengunjung di objek wisata Jatim Park 1.

# DAFTAR PUSTAKA

E. Sipasi and E. M. Y. E. M. Yasak, “STRATEGI BRANDING JAWA TIMUR PARK GROUP SEBAGAI IKON PARIWISATA KOTA BATU,” *Jurnal Ilmu Sosial dan Ilmu Politik (JISIP)*, vol. 6, no. 3, Jan. 2019, Accessed: Dec. 04, 2023. [Online]. Available: https://publikasi.unitri.ac.id/index.php/fisip/article/view/1460/1034

‌H. Ardeena, Wasino, D. Arisandi, K. Kunci, and J. Perjalanan, “DESAIN SISTEM INFORMASI WISATA DI JALUR PERJALANAN YOGYAKARTA-MAGELANG- SEMARANG BERBASIS WEBSITE.”

K. Afiifah, Z. F. Azzahra, and A. D. Anggoro, “Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review,” *INTECH*, vol. 3, no. 1, pp. 8–11, Apr. 2022, doi: https://doi.org/10.54895/intech.v3i1.1261.

‌“Layanan Registrasi Nama Domain ID Internasional Murah Full Control Panel,” *www.iixplanet.com*. https://ijns.org/journal/index.php/ijns/article/view/181/176 (accessed Dec. 05, 2023).