Project Case	BINUS UNIVERSITY Software Laboratory Center
ISYS6169 | ISYS6279 | ISYS6280 | T0206 Database Systems	
Information Systems	**O212-ISYS6169-AT01-00**
Valid on *Odd Semester Year 2020/2021*	**Revision 00**

1. Seluruh kelompok tidak diperkenankan untuk:
   The whole group is not allowed to:
   - Melihat sebagian atau seluruh proyek kelompok lain,
     Seeing a part or the whole project from other groups
   - Menyadur sebagian maupun seluruh proyek dari buku,
     Adapted a part or the whole project from the book
   - Mendownload sebagian maupun seluruh proyek dari internet,
     Downloading a part or the whole project from the internet,
   - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,
     Working with another theme which is not in accordance with the existing theme in the matter of the project,
   - Melakukan tindakan kecurangan lainnya,
     Committing other dishonest actions,
   - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.
     Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.

2. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.
   If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be "Zero"

3. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.
   Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted

4. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:
   Marking percentage for this subject is described as follows:

Tugas Mandiri *Assignment*	**Proyek** *Project*	**UAP** *Final Exam*
30%	30%	40%

5. Software yang digunakan pada matakuliah ini adalah sebagai berikut:
Software will be used in this subject are described as follows:

Software *Software*
SQL Server Management Studio 18.5.1 SQL Server Developer 2019 Microsoft Office 365 Visual Paradigm Community Edition 16.1

6. Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, proyek dan uap untuk matakuliah ini adalah sebagai berikut:
File extensions should be included in assignment, project, and final exam collection for this subject are described as follows:

Tugas Mandiri *Assignment*	**Proyek** *Project*	**UAP** *Final Exam*
SQL	SQL, VPP, Image Files (JPG / PNG)	SQL

Soal

Case

Catering delVia

Catering delVia is a catering services managed by your friend, Via. Via manage all of activities that belongs to **Catering delVia** like **providing catering services to customer** and **purchasing ingredients from vendors**.

To improve **Catering delVia's** management, Via hired you as a database administrator are required to analyze and design a database for Bob's shop using **SQL Server Management Studio**.

Every staff that hired by **Catering delVia** have a task to **serve a customer who use the service** and **purchase an ingredient from a vendor**. Every staff must be following the procedures to become a staff, which are:

- Every **staff** hired must have a personal information like **name, gender, email, phone number, address,** and **salary**. Every **staff** has an identification number with the following format:

  "STXXX"
  X => number between 0 – 9

- Every staff has its own **position** information like **position name** and has an identification number with the following format:

  "SPXXX"
  X => number between 0 – 9

- Every **purchase** made with the vendor have all the information about **staff, vendor, purchase date, ingredient(s) purchased,** and **the quantity of each ingredient**. Every **purchase transaction** has an identification number with the following format:

  "PUXXX"
  X => number between 0 – 9

- Every **ingredient** purchased from supplier have its own **name, stock**, and **price**. Every **ingredient** has an identification number with the following format:

  "IDXXX"
  X => number between 0 – 9

- Every **vendor** that wants to sell their ingredient must already completed vendor information like **name, phone number,** and **address**. Every **vendor** has an identification number with the following format:

“VEXXX”
X => number between 0 – 9

- Vendor can sell **more than one ingredient** in every transaction.

Every customer that wants to get a service at **Catering delVia** must be following the **transaction procedures**, those are:

- Every **customer** that wants to purchase a product must already completed personal information like **name, phone number, address, gender,** and **email**. Every customer has an identification number with the following format:

“CUXXX”
X => number between 0 – 9

- Every **services transaction** made by the customer have all the information about **staff, customer, transaction date, reservation type, reservation address, menu(s) sold,** and **the pax of each menu**. Every **services transaction** has an identification number with the following format:

“TRXXX”
X => number between 0 – 9

- Customer can purchase **more than one menu** in every transaction.
- Every **Menu** sold by **Catering delVia** have its own **name, description**, and **price**. Every **menu** has an identification number with the following format:

“MEXXX”
X => number between 0 – 9

Notes:

- Staff phone number must be numeric and starts with ‘08’ (without quote).
- Staff email must not start with ‘@’ and must ends with ‘@gmail.com’ or ‘@yahoo.com’ or ‘@yahoo.co.id’ (without quote).
- Staff salary must between 500000 and 5000000.
- Menu name must be more than 5 characters.

- Customer phone number must be numeric and starts with ‘08’.
- Customer email must not start with ‘@’ and must ends with ‘@gmail.com’ or ‘@yahoo.com’ or ‘@yahoo.co.id’ (without quote).
- Vendor name must start with ‘PT. ’ (without quote).

Now **Catering delVia** still using manual management system to maintain the **services** and **purchase transactions**. You as her precious friend wants to help **Catering delVia** to create a database system that can store data and maintain the **services** and **purchase transactions**. The tasks that you must do are:

a. Create **Entity Relationship Diagram** to maintain **services** and **purchase transactions**.
b. Create a database system using **DDL syntax** that relevant with **services** and **purchase transactions**.
c. Create query using **DML syntax** to fill the tables in database systems with data based on the following conditions:
    - **Master** table must be filled with more than or equals 10 data.
    - **Transaction** table must be filled with more than or equals 15 data.
    - **Transaction detail** table must be filled with more than or equals 25 data.
    - For the **Staff Position** table, the table must be filled with the following data:

Staff Position Names	
CEO	Secretary
Treasurer	Promotion
Marketing	Inventory Manager
Head Chef	Chef
Supervisor	Cashier

d. Create query using **DML syntax** to **simulate** the transactions process for **services** and **purchase transactions**.

   Note: DML syntax to **fill database** and DML syntax to **simulate** the **transactions process** should be a **different query**.

e. To support database management process in **Catering delVia**, Rika asked you to provide some query that resulting important data. The requirements that asked from her are:

1. Display StaffName, StaffPositionName, and Total Activity (obtained by counting the services and purchase transaction made by the staff) for every staff whose salary is between 1000000 and 4000000 and has total activity of more than 2.

2. Display CustomerID, CustomerName, and Pax Bought (obtained by calculating the sum of pax in each transaction) for every male customer and every transaction that occurred within the first half of the year.

3. Display IngredientName, Ingredient Bought (obtained by calculating the sum of quantity of each purchase), Purchase Count (obtained by counting the number of purchase), and Total Expenses (obtained by calculating the sum of quantity * ingredient price) for every ingredient that is purchased in even month and happened in day between 2 and 5 of the week.

4. Display Staff First Name (obtained from staff's first name), Transaction Count (obtained by counting the number of transaction), and Pax Sold (obtained by calculating the sum of pax sold from each transaction) for every staff that has odd id and serves customer with even id.

5. Select Vendor Name (obtained from VendorName without 'PT. '), IngredientName, and Ingredient Price (obtained by adding 'Rp. ' to IngredientPrice) for every purchase with ingredient price more than or equal to the average of all ingredient, and ingredient stock lower than 250.
   (**alias subquery**)

6. Display CustomerName, TransactionDate (obtained from PurchaseDate with 'Mon dd, yyyy' format), MenuName, MenuPrice, Brief Description (obtained from the first 2 words from MenuDescription), Pax, Total Price (obtained by calculating MenuPrice times Pax) for every menu with price more than the average of all menu and pax sold more than 100 in each transaction.
   (**alias subquery**)

7. Display Staff Name (obtained from StaffName in uppercase letter), Purchase Date (obtained Purchase date in ‘Mon dd, yyyy’ format), Quantity Bought (obtained by calculating sum of quantity and adding ‘ pcs’ at the end) for every purchase that occurred in even month with ingredient that has price lower than the maximum price of all ingredient.
   (**alias subquery**)

8. Display CustomerName, Email (obtained from CustomerEmail before ‘@’), Menu Name (obtained from MenuName in lowercase), and Pax Bought (obtained by calculating the sum of pax) for every transaction that serves male customer and pax more than or equal to the average of all pax.
   (**alias subquery**)

9. Create a view named ‘**LoyalCustomerView**’ to display CustomerName, Total Transaction (obtained by counting the customer id), Total Pax Purchased (obtained by calculating the sum of pax), and Total price (obtained by calculating the sum of pax times menuPrice) for every female customer that that has made a transaction more than two times.

10. Create a view named ‘**VendorRecapView**’ to display VendorName, Purchases Made (obtained by counting the vendorID), and Ingredients Purchased (obtained by calculating the sum of quantity) for every vendor that has made a transaction more than once and ingredients with stock more than 150.

File that must be collected:

1. Entity Relationship Diagram (.vpp, .png)
2. Query to create the database system. (.sql)
3. Query to insert data into tables. (.sql)
4. Query to simulate the transactions processes. (.sql)
5. Query to answer the 10 cases. (.sql)

Here are the rules that you must follow to create your project:

1. Use appropriate software for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
2. Use the techniques taught during practicum.
3. Collect appropriate files for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
4. Include the other files that can support your project, such as:
   - All files in your project
   - Other files (image, audio, video, etc.) used in your project
   - *.DOC file (documentation of your project) that contains the reference links of additional files (image, audio, video, etc.) used in your project.