

Nama : Gibran Fasha Ghazanfar
E-mail : gibran.fsh@gmail.com
No. WA : +62 813 1122 9890

SS HASIL TES LOGIKA

Buat fungsi berikut menggunakan **bahasa pemrograman yang paling anda kuasai** (boleh js, python, php, java, dll.)

1. Buat fungsi dengan menampilkan bilangan cacah kelipatan 3 atau 7 sebanyak N, serta menampilkan huruf Z saat bilangan tersebut kelipatan 3 dan 7.

Contoh :

N = 13

Output : 3, 6, 7, 9, 12, 14, 15, 18, **Z**, 24, 27, 28, 30

SS Hasil Kode :

```
A:\FinplanIDBackendTest\Tes Logika>javac BilanganCacah.java
A:\FinplanIDBackendTest\Tes Logika>java BilanganCacah
13
3, 6, 7, 9, 12, 14, 15, 18, Z, 24, 27, 28, 30
A:\FinplanIDBackendTest\Tes Logika>
```

2. Buat fungsi pencarian 'sang gajah', 'serigala', 'harimau'.
Dengan contoh masukan dan keluaran sebagai berikut :

Input : Berikut adalah kisah sang gajah. Sang gajah memiliki teman serigala bernama DoeSang. Gajah sering dibela oleh serigala ketika harimau mendekati gajah.

Output : sang gajah - sang gajah - serigala - serigala - harimau

SS Hasil Kode :

```
A:\FinplanIDBackendTest\Tes Logika>javac PencarianKata.java
A:\FinplanIDBackendTest\Tes Logika>java PencarianKata
Berikut adalah kisah sang gajah. Sang gajah memiliki teman serigala bernama DoeSang. Gajah sering dibela oleh serigala
ketika harimau mendekati gajah.
sang gajah - sang gajah - serigala - serigala - harimau
A:\FinplanIDBackendTest\Tes Logika>
```

3. Buatlah fungsi pengecekan kata sandi, dengan ketentuan sebagai
 - a. Kata sandi minimal 8 karakter
 - b. Kata sandi maksimal 32 karakter
 - c. Karakter awal tidak boleh angka
 - d. Harus memiliki angka

- e. Harus memiliki huruf kapital dan huruf kecil

Contoh

Input : Sandiwara

Output : Karakter awal tidak boleh angka

Input : sandiwar4

Output : Harus memiliki huruf kapital dan huruf kecil

Input : Sandiwar4

Output : Kata sandi valid

SS Hasil Kode :

```
A:\FinplanIDBackendTest\Tes Logika>javac PasswordChecker.java
A:\FinplanIDBackendTest\Tes Logika>java PasswordChecker
Sandiwara
Karakter awal tidak boleh angka
A:\FinplanIDBackendTest\Tes Logika>java PasswordChecker
sandiwar4
Harus memiliki huruf kapital dan huruf kecil
A:\FinplanIDBackendTest\Tes Logika>java PasswordChecker
Sandiwar4
Kata sandi valid
A:\FinplanIDBackendTest\Tes Logika>
```

4. Buat fungsi pengecekan bilangan cacah terkecil yang tidak ada dari data yang diinputkan. Dengan contoh input dan output sebagai berikut :

| Contoh 1 | Contoh 2 | Contoh 3 |
|-------------------------------------------|------------------------------------|-------------------------------------|
| Input : [5, 2, 8, 4, 3, 10] Output : 6 | Input : [2, 3, 4, 6] Output : 5 | Input : [8, 6, 7, 12] Output : 9 |

SS Hasil Kode :

```

A:\FinplanIDBackendTest\Tes Logika>javac BilanganCacahTerkecil.java

A:\FinplanIDBackendTest\Tes Logika>java BilanganCacahTerkecil
Masukkan jumlah elemen: 6
Masukkan elemen (Input):
5 2 8 4 3 10
Output: 6

A:\FinplanIDBackendTest\Tes Logika>java BilanganCacahTerkecil
Masukkan jumlah elemen: 4
Masukkan elemen (Input):
2 3 4 6
Output: 5

A:\FinplanIDBackendTest\Tes Logika>java BilanganCacahTerkecil
Masukkan jumlah elemen: 4
Masukkan elemen (Input):
8 6 7 12
Output: 9

A:\FinplanIDBackendTest\Tes Logika>

```

5. Buat pola berikut sesuai inputan N, dengan N adalah bilangan ganjil

**catatan : nilai lebih jika pakai rekursif*

| Input | N = 5 | N = 3 | N = 7 | N = 2 |
|--------|-------------------------------------------|-------------------|-----------------------------------------------------------------------------|-----------------------|
| Output | XXXXX XOOXX XOXOX XXOOX XXXXX | XXX XXX XXX | XXXXXXXX XOOOOXX XOOOXOX XOOXOOX XOXOOOX XXOOOOX XXXXXXXX | Harus bilangan ganjil |

SS Hasil Kode :

```

A:\FinplanIDBackendTest\Tes Logika>javac Pattern.java

A:\FinplanIDBackendTest\Tes Logika>java Pattern
5
XXXXX
X00XX
X0X0X
XX00X
XXXXX

A:\FinplanIDBackendTest\Tes Logika>java Pattern
3
XXX
XXX
XXX

A:\FinplanIDBackendTest\Tes Logika>java Pattern
7
XXXXXXX
X0000XX
X000X0X
X00X00X
X0X000X
XX0000X
XXXXXXX

A:\FinplanIDBackendTest\Tes Logika>java Pattern
2
Harus bilangan ganjil

A:\FinplanIDBackendTest\Tes Logika>_

```

Notes :

- **Catat semua query search yang anda ketikkan di mesin pencari (simpan dlm bentuk .txt)**
- Pembuatan boleh dalam bentuk (.js .php .java dll.) sesuai **bahasa yang paling anda kuasai**
- Buat repository di github.com / gitlab.com
- Invite github / gitlab repository ke gasur.galih@gmail.com
- Buat README.md, dengan ketentuan :
 - Cara menjalankan apps / website
 - Screenshot hasil

SS HASIL TES BACKEND (MySQL Query)

Terdapat 3 tabel berikut

users

| id | name | gender | status |
|----|-----------|--------|--------|
| 1 | Frasch | F | active |
| 2 | Garmuth | M | active |
| 3 | Goliath | M | active |
| 4 | Luna | F | active |
| 5 | Zeus | M | active |
| 6 | Aphrodite | F | active |
| 7 | Ares | M | active |
| 8 | Lina | F | active |
| 9 | Lanaya | F | active |
| 10 | Hades | M | active |

hobbies

| id | name | level |
|----|----------|-------|
| 1 | Running | 8 |
| 2 | Skipping | 5 |
| 3 | Push Up | 10 |

map_user_hobby

| id | id_user | id_hobby | status |
|----|---------|----------|---------|
| 1 | 1 | 1 | active |
| 2 | 3 | 1 | active |
| 3 | 8 | 3 | deleted |
| 4 | 2 | 2 | active |
| 5 | 4 | 1 | deleted |
| 6 | 6 | 2 | active |
| 7 | 5 | 3 | active |
| 8 | 8 | 1 | active |
| 9 | 7 | 2 | active |
| 10 | 4 | 2 | active |
| 11 | 9 | 3 | deleted |
| 12 | 10 | 2 | deleted |
| 13 | 3 | 2 | active |
| 14 | 2 | 3 | active |
| 15 | 10 | 2 | active |

Soal :

1. Buat query untuk membuat tabel di atas dengan indexing.

SS Hasil Query:

A. Tabel Users :

```

mysql> CREATE TABLE Users (
  ->   id INT PRIMARY KEY,
  ->   name VARCHAR(255),
  ->   gender CHAR(1),
  ->   status VARCHAR(20)
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> INSERT INTO Users (id, name, gender, status) VALUES
  -> (1, 'Frasch', 'F', 'active'),
  -> (2, 'Garmuth', 'M', 'active'),
  -> (3, 'Goliath', 'M', 'active'),
  -> (4, 'Luna', 'F', 'active'),
  -> (5, 'Zeus', 'M', 'active'),
  -> (6, 'Aphrodite', 'F', 'active'),
  -> (7, 'Ares', 'M', 'active'),
  -> (8, 'Lina', 'F', 'active'),
  -> (9, 'Lanaya', 'F', 'active'),
  -> (10, 'Hades', 'M', 'active');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> create index idx_name_users on User (name);
ERROR 1146 (42S02): Table 'finplan.user' doesn't exist
mysql> create index idx_name_users on Users (name);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>

```

B. Tabel Hobbies :

```

mysql> CREATE TABLE Hobbies (
  ->   id INT PRIMARY KEY,
  ->   name VARCHAR(255),
  ->   level INT
  -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> INSERT INTO Hobbies (id, name, level) VALUES
  -> (1, 'Running', 8),
  -> (2, 'Skipping', 5),
  -> (3, 'Push Up', 10);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> create index idx_name_hobbies on Hobbies (name);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

```

C. Tabel map_user_hobby :

```
mysql> CREATE TABLE map_user_hobby (  
-> id INT PRIMARY KEY,  
-> id_user INT,  
-> id_hobby INT,  
-> status VARCHAR(20)  
-> );  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>  
mysql> INSERT INTO map_user_hobby (id, id_user, id_hobby, status) VALUES  
-> (1, 1, 1, 'active'),  
-> (2, 3, 1, 'active'),  
-> (3, 8, 3, 'deleted'),  
-> (4, 2, 2, 'active'),  
-> (5, 4, 1, 'deleted'),  
-> (6, 6, 2, 'active'),  
-> (7, 5, 3, 'active'),  
-> (8, 8, 1, 'active'),  
-> (9, 7, 2, 'active'),  
-> (10, 4, 2, 'active'),  
-> (11, 9, 3, 'deleted'),  
-> (12, 10, 2, 'deleted'),  
-> (13, 3, 2, 'active'),  
-> (14, 2, 3, 'active'),  
-> (15, 10, 2, 'active');  
Query OK, 15 rows affected (0.00 sec)  
Records: 15 Duplicates: 0 Warnings: 0  
  
mysql> create index idx_userid_hobbyid_map_user_hobby on map_user_hobby (id_user, id_hobby);  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>
```

2. Buat query untuk menghitung jumlah user masing-masing gender pada suatu hobby tertentu
- Input / filter : skipping
 - Output :

| gender | total |
|--------|-------|
| M | 5 |
| F | 1 |

SS Hasil Query :

Query untuk mendapatkan jumlah user dari masing-masing gender untuk hobby Skipping :

```
mysql> select gender, count(gender) as total from users join map_user_hobby on users.id = map_user_hobby.id_user join hobbies on hobbies.id = map_user_hobby.id_hobby where hobbies.name = 'Skipping' group by gender order by total desc;
```

| gender | total |
|--------|-------|
| M | 5 |
| F | 2 |

2 rows in set (0.00 sec)

3. Buat query untuk menghitung jumlah hobby dari setiap user yang masih aktif
 - Output kolom : name_user, total

SS Hasil Query :

```
mysql> select u.name as name_user, count(m.id_hobby) as total from users u join map_user_hobby m on u.id = m.id_user where u.status = 'active' group by u.id, u.name;
```

| name_user | total |
|-----------|-------|
| Frasch | 1 |
| Garmuth | 2 |
| Goliath | 2 |
| Luna | 2 |
| Zeus | 1 |
| Aphrodite | 1 |
| Ares | 1 |
| Lina | 2 |
| Lanaya | 1 |
| Hades | 2 |

10 rows in set (0.00 sec)

```
mysql> _
```

4. Hitung level rata - rata tiap user dari hobi yang dimiliki, ketika user memiliki hobi lebih dari 1
 - Output kolom : name, level_avg

SS Hasil Query :

```
mysql> select u.name as name, avg(h.level) as level_avg from users u join map_user_hobby m on u.id = m.id_user join hobbies h on h.id = m.id_hobby group by u.id, u.name having count(h.id) > 1;
```

| name | level_avg |
|---------|-----------|
| Garmuth | 7.5000 |
| Goliath | 6.5000 |
| Luna | 6.5000 |
| Lina | 9.0000 |
| Hades | 5.0000 |

5 rows in set (0.00 sec)

```
mysql>
```

Notes :

- **Catat semua query search yang anda ketikkan di mesin pencari (simpan dlm bentuk .txt)**
- Pembuatan boleh dalam query (.sql)
- Buat repository di github.com / gitlab.com

- Invite github / gitlab repository ke gasur.galih@gmail.com
- Buat README.md, dengan ketentuan :
 - Nama lengkap / email yg dapat dihubungi
 - Cara menjalankan apps / website
 - Screenshot hasil