**KETENTUAN PENGERJAAN TEST**

1. Pertanyaan ini wajib dikerjakan berurutan dari Kategori A sampai dengan Kategori B.
2. Setiap ketegori sudah memiliki batas waktu masing-masing, yaitu :

Kategori A : 45 Menit

Kategori B : 60 Menit

1. Kandidat diperbolehkan melakukan browsing.
2. Seluruh hasil pengerjaan dapat dikirimkan ke email : [hrd@aptika.co.id](mailto:hrd@aptika.co.id) dengan subject : Nama\_Skill Test\_Kategori (A/B).
3. **Jawaban dan screenshoot langsung dilampirkan di file ini!**

Informasi tambahan :

1. Jawaban kode A bisa menggunakan nodejs / php
2. Jawaban DB A dan DB B wajib menggunakan SQL command
3. Jawaban Code B bisa menggunakan scenario flow/activity diagram/flow chart.
4. Jawaban bisa di kerjakan di mana saja (online tools hanya alat bantu tambahan jika di perlukan)
5. **Jawaban dan screenshoot langsung dilampirkan di halaman ini**

**Kategori A (Waktu pengerjaan : 45 Menit)**

1. **Bobot nilai 20  
   Database:** <https://paiza.io/en/projects/new?language=mysql>

Anda mempunyai skema tabel sebagai berikut:

create table data (

data\_id int primary key auto\_increment,

data\_nama varchar(100) not null

);

INSERT INTO data VALUES (1, 'Test 1');

INSERT INTO data VALUES(2, 'Test 2');

create table data\_detail(

data\_id int not null,

data\_key varchar(50) not null,

data\_val varchar(100) not null

);

INSERT INTO data\_detail VALUES (1, 'key 1', 'Val 1');

Table Data

|  |  |
| --- | --- |
| Data\_id | Data\_nama |
| 1 | Test 1 |
| 2 | Test 2 |

Tabel data\_detail

|  |  |  |
| --- | --- | --- |
| Data\_id | Data\_key | Data\_val |
| 1 | Key 1 | Val 1 |

Buatkan query dengan hasil akhir dengan header dan value sebagai Berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Nama | Data\_Key | Data\_Value |
| 1 | Test 1 | Key 1 | Val 1 |
| 2 | Test 2 | No data | No Data |

**Jawaban kode yang belum dijalankan (bukan screenshoot)** :

CREATE TABLE `data` (

`data\_id` INT NOT NULL AUTO\_INCREMENT,

`data\_nama` VARCHAR(100) NOT NULL,

PRIMARY KEY (`data\_id`));

CREATE TABLE `data\_detail` (

`data\_id` INT NOT NULL,

`data\_key` VARCHAR(50) NOT NULL DEFAULT 'No Data',

`data\_val` VARCHAR(100) NOT NULL DEFAULT 'No Data');

INSERT INTO data (data\_nama) VALUES ('test 1');

INSERT INTO data (data\_nama) VALUES ('test 2');

INSERT INTO data\_detail (data\_id, data\_key, data\_val) VALUES (1, 'Key 1', 'Val 1');

INSERT INTO data\_detail (data\_id) VALUES (2);

SELECT \* FROM data;

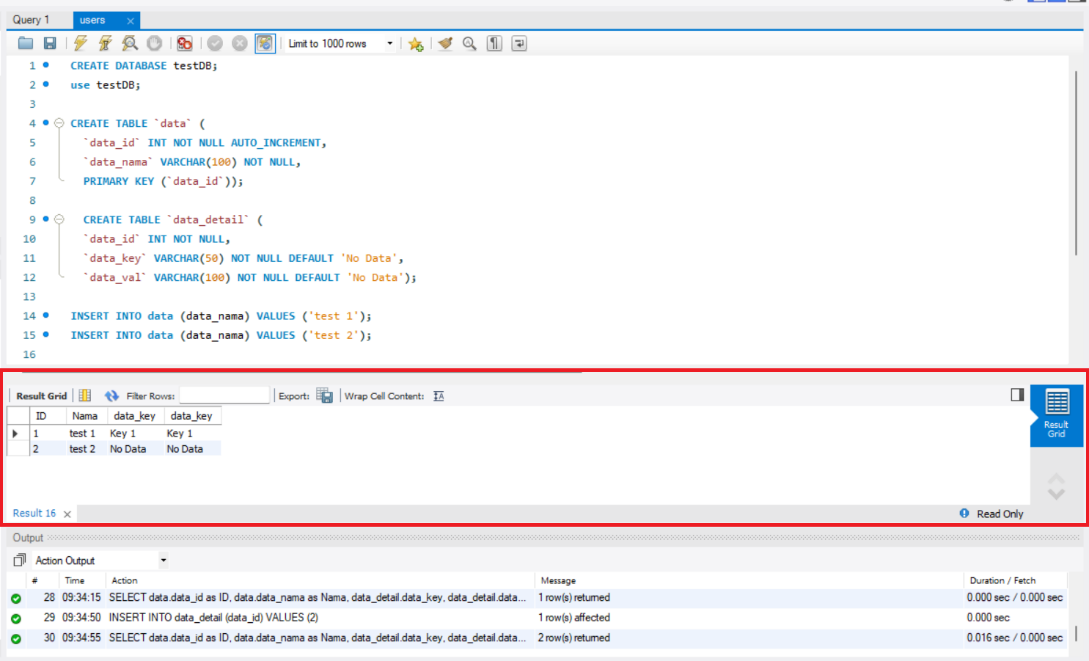
SELECT \* FROM data\_detail;

SELECT data.data\_id as ID, data.data\_nama as Nama, data\_detail.data\_key, data\_detail.data\_key

FROM data

INNER JOIN data\_detail ON data.data\_id = data\_detail.data\_id;

**Hasil screenshoot kode setelah dijalankan:**



1. **Bobot Nilai 30  
   Sort and Looping**

**Code :** <https://paiza.io/en/projects/new?language=php>

<https://paiza.io/en/projects/new?language=javascript>

Ada sebuah array menu

/\*

Index 0 = id Menu

Index 1 = Parent Menu ( 0 artinya paling atas)

Index 2 = Nama Menu

Index 3 = Ordering Menu

\*/

[

[1, 0, "Beranda", 0],

[2, 0, "Product", 2],

[3, 2, "Product 1", 0],

[4, 2, "Product 2", 1],

[5, 2, "Product 3", 2],

[6, 0, "About Us", 1],

[7, 6, "Contact Us", 0],

[8, 5, "Product 3.1", 0],

[9, 2, "Product 4", 3]

]

Buatlah sebuah fungsi untuk mencetak menu, untuk halaman menu tanpa batas children dan sort sesuai dengan value Ordering index ke 3, hasil akhir yang diinginkan beruta text dengan indentasi spasi.

- Beranda (0)

- About Us (1)

- Contact Us (0)

- Product (2)

- Product 1 (0)

- Product 2 (1)

- Product 3 (2)

- Product 3.1 (0)

- Product 4 (3)

**NB:**

**Recursive functions, Sort & Multidimensional Array**

**Jawaban kode yang belum dijalankan (bukan screenshoot)** :

const array = [

[1, 0, "Beranda", 0],

[2, 0, "Product", 2],

[3, 2, "Product 1", 0],

[4, 2, "Product 2", 1],

[5, 2, "Product 3", 2],

[6, 0, "About Us", 1],

[7, 6, "Contact Us", 0],

[8, 5, "Product 3.1", 0],

[9, 2, "Product 4", 3]

]

console.log('result', array.sort(function(a, b){

if (a[1] == b[1]) {

return b[3] - a[3];

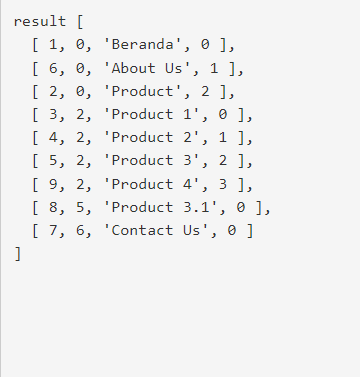
}

return b[1] - a[1];

}).reverse()

)

**Hasil screenshoot kode setelah dijalankan:**

****

**Kategori B (Waktu pengerjaan : 60 menit)**

1. **Bobot nilai 20  
   Database Test Insert:** <https://paiza.io/en/projects/new?language=mysql>

Anda mempunyai tabel sebagai berikut :

create table saldo (

saldo\_id int primary key auto\_increment,

saldo\_tanggal datetime not null,

user\_id int not null,

saldo\_value int not null

);

INSERT INTO saldo VALUES (1, '2020-10-01 13:00', 1, 10000);

INSERT INTO saldo VALUES (2, '2020-10-01 14:00', 1, 15000);

INSERT INTO saldo VALUES (3, '2020-10-01 15:00', 1, 20000);

INSERT INTO saldo VALUES (4, '2020-10-01 13:00', 2, 25000);

INSERT INTO saldo VALUES (5, '2020-10-01 13:00', 3, 30000);

INSERT INTO saldo VALUES (6, '2020-10-01 14:00', 3, 50000);

Tabel saldo

|  |  |  |  |
| --- | --- | --- | --- |
| Saldo\_id | Saldo\_tanggal | User\_id | Saldo\_value |
| 1 | 2020-10-01 13:00 | 1 | 10000 |
| 2 | 2020-10-01 14:00 | 1 | 15000 |
| 3 | 2020-10-01 15:00 | 1 | 20000 |
| 4 | 2020-10-01 13:00 | 2 | 25000 |
| 5 | 2020-10-01 13:00 | 3 | 30000 |
| 6 | 2020-10-01 14:00 | 3 | 50000 |

create table projection(

projection\_id int primary key auto\_increment,

projection\_tanggal datetime not null,

user\_id int not null,

saldo\_id int not null,

projection\_value int not null

);

Anda harus melakukan insert ke table project untuk setiap unique user\_id dengan last\_id terakhir saldo itu di jumlah, dengan single query akan menghasilkan values table serperti di bawah ini:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Projection\_id | Projection\_tanggal | User\_id | Saldo\_id | Projection\_value |
| 1 | 2020-11-02 03:20:38 | 1 | 3 | 45000 |
| 2 | 2020-11-02 03:20:38 | 2 | 4 | 25000 |
| 3 | 2020-11-02 03:20:38 | 3 | 6 | 80000 |

**Jawaban kode yang belum dijalankan (bukan screenshoot)** :

INSERT INTO projection (projection\_tanggal, user\_id, saldo\_id, projection\_value) VALUES

(

'2020-11-02 03:20:38',

1,

(

SELECT MAX(saldo\_id)

FROM saldo

WHERE projection.user\_id = saldo.user\_id

),

(

SELECT SUM(saldo\_value)

FROM saldo

WHERE projection.user\_id = saldo.user\_id

)

);

INSERT INTO projection (projection\_tanggal, user\_id, saldo\_id, projection\_value) VALUES

(

'2020-11-02 03:20:38',

2,

(

SELECT MAX(saldo\_id)

FROM saldo

WHERE projection.user\_id = saldo.user\_id

),

(

SELECT SUM(saldo\_value)

FROM saldo

WHERE projection.user\_id = saldo.user\_id

)

);

INSERT INTO projection (projection\_tanggal, user\_id, saldo\_id, projection\_value) VALUES

(

'2020-11-02 03:20:38',

3,

(

SELECT MAX(saldo\_id)

FROM saldo

WHERE projection.user\_id = saldo.user\_id

),

(

SELECT SUM(saldo\_value)

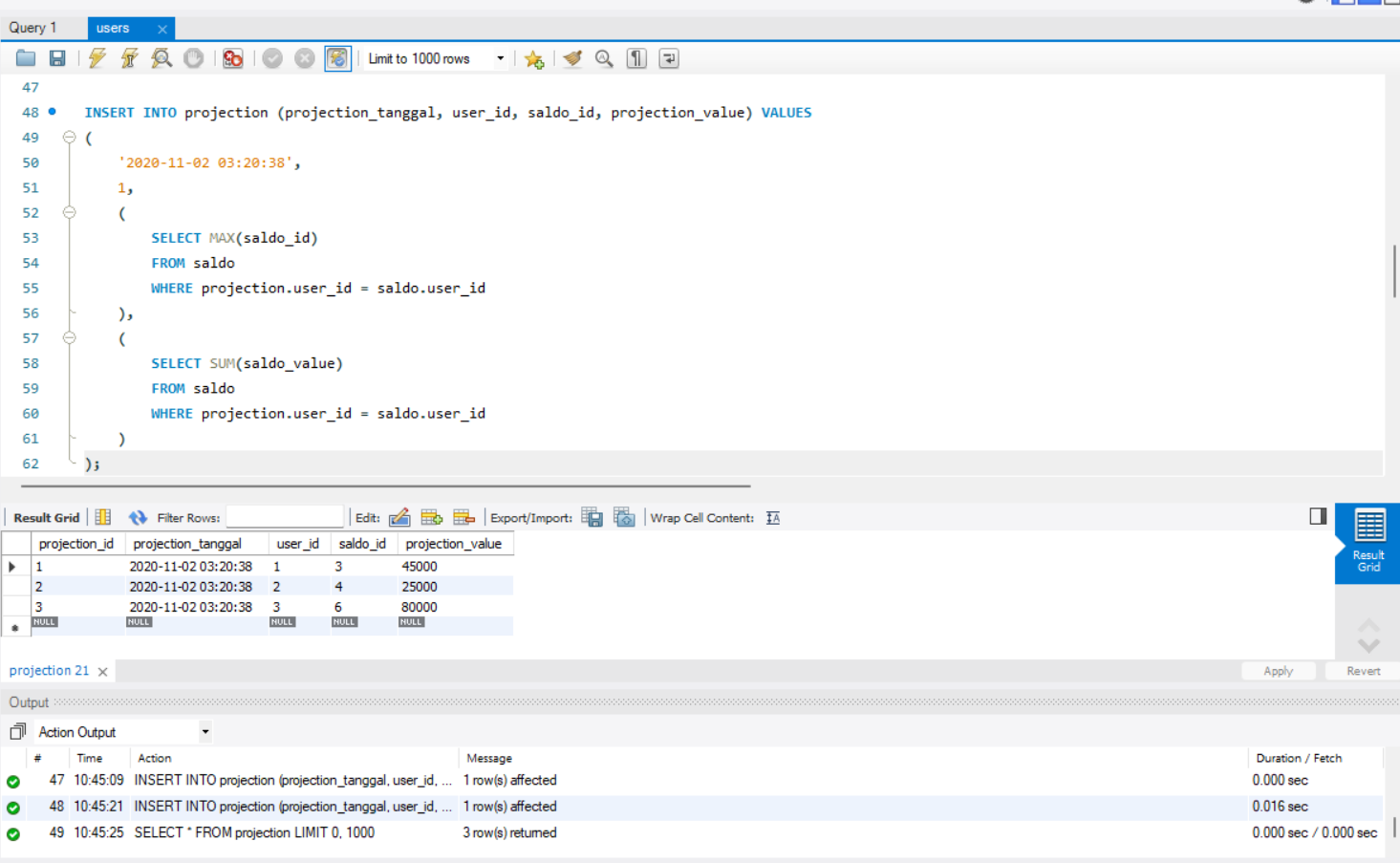
FROM saldo

WHERE projection.user\_id = saldo.user\_id

)

);

**Hasil screenshoot kode setelah dijalankan:**

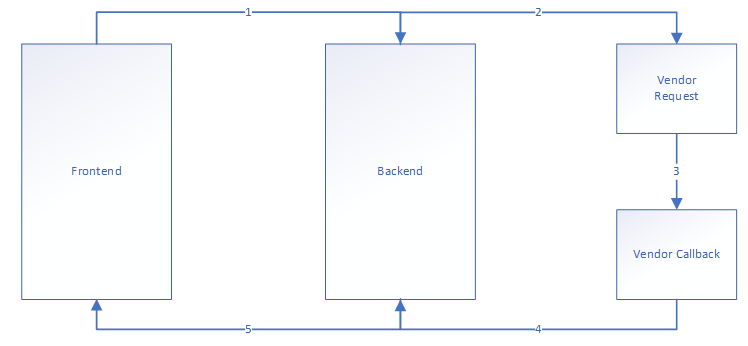


1. **Bobot Nilai 30  
   Scenario Code & test**

**Code :** <https://paiza.io/en/projects/new?language=php>

<https://paiza.io/en/projects/new?language=javascript>

Jika Anda harus menhubungkan vendor melalui REST API dengan metode callback hasil success dan gagal, dan harus menyampaikan ke bagian frontend secara realtime bukan menunggu callback, bagaimana flow yang anda akan buat dan scenario test apa saja yang harus Anda siapkan.



Flow koneksi vendor

1. Frontend melakukan request ke backend.

2. Backend melakukan REST APi ke vendor.

3. Vendor Callback akan Memanggil RESTAPI backend untuk notifikasi sukses/gagal

4. Backend melakukan response ke frontend.

Note: Request no 1 dan 5 merupakan Request dan Response secara langsung ke user.

Output

1. Jelaskan flow transaksi yang akan di buat untuk membuat callback transaction menjadi

realltime transaction!

2. Tulisan 5 case yang akan Anda coba agar tidak terjadi BUG dalam system.

**Jawab :**

**Case :**



* **G O O D L U C K ! -**