

|                       |   |
|-----------------------|---|
| <b>Nama</b>           | Agung Dwi Sahputra  |
| <b>Email</b>          | agungdwisahputra@gmail.com  |
| <b>Github account</b> | <a href="https://github.com/AgungDwiSahputra/php-fullstack-test">https://github.com/AgungDwiSahputra/php-fullstack-test</a> |

#### Ketentuan ujian IT Fullstack PHP - React:

1. Waktu mengerjakan adalah dalam waktu 125 menit termasuk push ke git.
2. Soal nomor 1 hingga 15 dapat dijawab pada lembar soal ini dengan cara menyilang pilihan jawaban dan dikembalikan kepada pengawas. Jangan lupa isikan nama dan alamat email anda.
3. Jawaban code dapat di-push ke [github](#) dengan invite user "[asiasiapac](#)" dengan nama repository **[you-github-account]/php-fullstack-test.git**.
4. Diperkenankan untuk menggunakan *tools* kerja digital yang biasa digunakan.
5. Silahkan eksplorasi internet untuk menjawab pertanyaan
6. Kecurangan menggunakan kecerdasan buatan seperti ChatGPT, Gemini, Perplexity, dan lainnya akan didiskualifikasi dari proses penerimaan.

#### Pilihan jawaban yang tepat

1. Dalam eloquent (laravel) ORM, mengapa penting untuk menggunakan properti `$fillable`?
  - a. Untuk memberikan default nilai pada kolom-kolom model.
  - b. Untuk memastikan bahwa hanya kolom-kolom tertentu yang dapat diisi melalui mass assignment.**
  - c. Agar model dapat menyimpan nilai-nilai yang bersifat rahasia.
  - d. Untuk memberikan batasan pada jumlah data yang dapat diambil dari database.
  - e. Untuk mencegah redudansi insert
2. Manakah dari berikut ini yang **bukan** fitur Redis di Laravel?
  - a. Pub/Sub (Publisher/Subscriber).**
  - b. Manajemen antrian pekerjaan.
  - c. Penyimpanan file secara terdistribusi
  - d. Caching variabel PHP secara langsung.
  - e. Data json sementara secara langsung
3. Apa yang dimaksud dengan "Cross-Origin Resource Sharing" (CORS) dalam konteks RESTful API?

- a. Teknik enkripsi data untuk mengamankan komunikasi.
- b. Mekanisme untuk membagi sumber daya antar domain.
- c. Metode otentikasi klien untuk mengakses sumber daya.
- d. Format data standar untuk pertukaran pesan.
- e. Optimasi koneksi antar layanan server.

4. Apa itu "autoload" dalam konteks Composer?

- a. Proses menginstal dependensi.
- b. Mekanisme untuk muatan otomatis class PHP.
- c. Fitur untuk mematikan cache Composer.
- d. Langkah-langkah untuk menyusun file `composer.json`.
- e. Semua jawaban benar

Untuk soal nomer 6 hingga 8, perhatikan kode berikut:

```

001 class UserController extends Controller {
002
003     public function index() {
004         $users
005         =
006         User::all();
007         return response()->json($users);
008     }
009
010     public function store(Request $request) {
011         $validatedData = $request->validate([
012             'name' =>
013             'required|string|max:255',
014             'email' => 'required|email|unique:users|max:255',
015             'password' => 'required|string|min:6',
016         ]);
017         $user = User::create($validatedData);
018
019         return response()->json($user, 201);
020     }
021
022
023
024
025     public function show($id) {
026         $user = User::find($id);
027         if ($user) {
028             return response()->json($user);
029         } else {
030             return response()->json(
031                 ['message' => 'User not found'], 404);
032         }
033     }

```

6. Apa yang dilakukan metode `store` pada kode diatas?
- Menghapus pengguna dari basis data.
  - Menambahkan data pengguna baru ke dalam memory index.
  - Memperbarui informasi pengguna yang sudah ada.
  - Mengambil data satu pengguna berdasarkan ID.
  - e.** Menambahkan pengguna baru ke dalam database berdasarkan input yang divalidasi.
7. Apa yang dihasilkan oleh metode `show` jika pengguna dengan ID yang diberikan tidak ditemukan?
- Respons JSON dengan data pengguna yang ditemukan.
  - b.** Respons 404 dengan pesan 'User not found'.
  - Respons header kosong dengan kode 404 tanpa data pengguna.
  - Respons 500 dengan pesan kesalahan server.
  - Respons null dengan parameter http 404
8. Jika model user diatas tersedia, maka untuk mempersiapkan model tersebut dapat menggunakan perintah
- a.** `use App\Models\User;`
  - `use \App\Models\User;`
  - `use App/Models/User;`
  - `use App\User->load();`
  - `use Models\User;`
9. Apa arti karakter "+" dalam pola regex `[a-zA-Z0-9._-]+?`
- Karakter "+" menandakan bahwa setiap karakter khusus harus diulangi tepat satu kali.
  - Karakter "+" menandakan bahwa setiap karakter khusus bersifat opsional.
  - c.** Karakter "+" menandakan bahwa setiap karakter tersebut dapat diulangi lebih dari 1 kali.
  - Karakter "+" menandakan bahwa setiap karakter khusus bersifat eksklusif.
  - Karakter "+" menandakan bahwa setiap karakter khusus bersifat sementara dan repetisi.

10. `preg_match("/^\+?[0-9]$/", $nomor)`  
pada potongan kode diatas, maka bernilai **TRUE** jika isi dari variable nomor adalah.
- a. +123.456.789
  - b. +1.2.3.4.5.6.7.8.9
  - c. 1.2.3.4.5.6.7.8.9
  - d. +1,2,3,4,5,6,7,8,9
  - e. +123456789

Untuk soal nomor 11, perhatikan potongan kode berikut

```
1 import React, { useState } from 'react';
2
3 const Counter: React.FC = () => {
4   const [count, setCount] = useState<number>(0);
5
6   const increment = () => setCount(count + 1);
7
8   return (
9     <div>
10      <h1>Count: {count}</h1>
11      <button onClick={increment}>Increment</button>
12    </div>
13  );
14 };
15 export default Counter;
16
```

11. Pada baris 4, Apa yang terjadi jika kita menghapus tipe number pada penggunaan `useState`, menjadi

```
const [count, setCount] = useState(0);
```

- a. TypeScript akan memberikan error karena tidak ada tipe yang ditentukan
- b. TypeScript akan secara otomatis mendeteksi tipe count sebagai number berdasarkan nilai awal 0
- c. Kompiler TypeScript akan mengubah `count` menjadi tipe `any`
- d. Kompiler TypeScript akan memberikan peringatan bahwa tipe `count` adalah `any`
- e. `useState` tidak dapat bekerja tanpa tipe yang ditentukan dalam TypeScript

Untuk soal nomor 12 hingga 14, perhatikan kode kompleks berikut

```
1 import React, { useState, useEffect } from 'react';
2
3 // Interface untuk props komponen
4 interface Todo {
5   id: number;
6   text: string;
7   completed: boolean;
8 }
9
10 interface TodoAppProps {
11   initialTodos: Todo[];
12 }
13
14 const TodoApp: React.FC<TodoAppProps> = ({ initialTodos })
15 => {
16   const [todos, setTodos] = useState<Todo[]>(initialTodos);
17   const [newTodo, setNewTodo] = useState<string>('');
18
19   useEffect(() => {
20     console.log('Todos updated:', todos);
21   }, [todos]);
22
23   const handleAddTodo = () => {
24     if (newTodo.trim()) {
25       const newTodoObj: Todo = {
26         id: Date.now(),
27         text: newTodo,
28         completed: false,
29       };
30       setTodos([...todos, newTodoObj]);
31       setNewTodo('');
32     }
33   };
34
35   const toggleTodoCompletion = (id: number) => {
36     const updatedTodos = todos.map(todo =>
37       todo.id === id ? { ...todo, completed: !todo.completed
38     } : todo
39   );
40   setTodos(updatedTodos);
41
42   return (
43     <div>
44       <h1>Todo App</h1>
45       <input
46         type="text"
```

```

47     value={newTodo}
48     onChange={(e) => setNewTodo(e.target.value)}
49     placeholder="Add new todo"
50   />
51   <button onClick={handleAddTodo}>Add Todo</button>
52   <ul>
53     {todos.map(todo => (
54       <li key={todo.id}>
55         <span
56           style={{ textDecoration: todo.completed ?
57 'line-through' : 'none' }}
58           onClick={() => toggleTodoCompletion(todo.id)}
59         > {todo.text}
60       </span>
61     </li>
62     )}}
63   </ul>
64 </div>
65 );
66 };
67 export default TodoApp;

```

12. Pada kode di atas, komponen `TodoApp` menggunakan `useState` untuk menyimpan list `todos`. Apa tipe data yang diterima oleh `useState` pada baris 16 ?
- `Todo[]`
  - `string[]`
  - `object[]`
  - `any[]`
  - `number[]`
13. Dalam kode di atas, `useEffect` digunakan untuk melacak perubahan pada array `todos`. Apa yang akan terjadi jika kita menghapus `todos` dari dependency array dalam `useEffect` pada baris 19 sampai 21?:
- `useEffect` tidak akan dipanggil sama sekali.
  - `useEffect` hanya akan dipanggil sekali setelah komponen dirender pertama kali
  - `useEffect` akan dipanggil setiap kali ada perubahan pada state atau props komponen.
  - `useEffect` akan dipanggil setiap kali ada perubahan pada `todos` dan `newTodo`

- e. `useEffect` akan dipanggil tanpa mempedulikan perubahan pada state atau props.

14. Perhatikan baris 35 sampai 40.

Apa yang terjadi jika kita memanggil `toggleTodoCompletion` dengan `id` yang tidak ada dalam array `todos`?

- a. Array `todos` akan tetap tidak berubah.
- b. Array `todos` akan terhapus
- c. Array `todos` akan menambah elemen baru dengan status `completed` menjadi `true`
- d. Array `todos` akan melempar error
- e. Status `completed` untuk semua `todo` akan berubah menjadi `true`

15. Bagaimana cara mengidentifikasi unik suatu objek di dalam bucket AWS S3?

- a. Etag (Entity Tag)
- b. Nama objek
- c. URL objek
- d. Domain region
- e. ID pengguna AWS

Untuk tugas code perhatikan sql berikut:

```
CREATE TABLE my_client (  
  id int NOT NULL GENERATED ALWAYS AS IDENTITY,  
  name char(250) NOT  
  NULL, slug char(100)  
  NOT NULL,  
  is_project varchar(30) check (is_project in ('0','1')) NOT NULL DEFAULT '0',  
  self_capture char(1) NOT NULL DEFAULT '1',  
  client_prefix char(4) NOT NULL,  
  client_logo char(255) NOT NULL DEFAULT 'no-image.jpg',  
  address text DEFAULT NULL,  
  phone_number char(50) DEFAULT NULL,  
  city char(50) DEFAULT NULL,  
  created_at timestamp(0) DEFAULT  
  NULL, updated_at timestamp(0)  
  DEFAULT NULL, deleted_at  
  timestamp(0) DEFAULT NULL, PRIMARY  
  KEY (id)  
);
```

Dari table yang dibuat:

- a. Silakan buat CRUD table ini ke dalam database postgresql
- b. Setiap yang tersimpan akan di generate redis dengan isi redis json data yang dimaksud, dan tersimpan secara persisten dengan key adalah isi slug
- c. Field column `client_logo` adalah sebuah URL image file yang di upload dan tersimpan dalam S3.
- d. Ketika update, redis akan terdelete dan di generate baru.
- e. Ketika delete data hanya membuat update kolom `deleted_at` dengan menghapus redis