

1. GIT Merupakan VCS atau Version Control System yang digunakan untuk mengetahui perubahan kode yang terjadi selama proses pengembangan perangkat lunak.

GIT dapat membantu developer untuk bekerja sama pada proyek yang sama tanpa harus khawatir mengganggu pekerjaan satu sama lain. Setiap developer dapat membuat salinan proyek ke lokal dan saling membagi tugas, kemudian menggabungkan hasil masing-masing ke repositori pusat

Perintah-perintah yang biasanya digunakan yaitu :

### **1. Membuat Repository**

Membuat repository dapat dilakukan dengan membuat repository baru di direktori dengan `git init`

atau mengkloning repository yang sudah ada dengan

`git clone https://github.com/user/repo.git`

### **2. Branching**

Perintah GIT untuk membuat cabang sendiri pada repository yang telah dibuat untuk mengerjakan bagian individu masing-masing

dimana ketika selesai dapat digabungkan dengan perintah `merge` ke cabang utama/main

### **3. Commit and push**

Commit merupakan titik perubahan yang dilakukan oleh developer yang kemudian di push ke repository agar dapat diakses oleh anggota tim lainnya.

### **4. Merge and Pull Request**

setelah menyelesaikan branch maka perlu dilakukan `merge` untuk menggabungkan bagian antar tim yang kemudian diperlukan `pull request` untuk

melihat perubahan yang dilakukan oleh anggota tim lain sebelum digabungkan

### **5. Resolusi Konflik**

ketika melakukan `merge`, biasanya terjadi konflik apabila terjadi kesalahan dalam mengedit baris yang sama, maka GIT akan meminta

resolusi konflik sebelum melanjutkan

### **6. git status**

Menampilkan status dari repository, termasuk perubahan yang belum di-commit.

### **7. git pull**

Mengambil perubahan terbaru dari repository

## 8. git checkout feature-branch

Berpindah ke cabang yang berbeda.

2. <?php

```
for ($i = 1; $i <= 100; $i++) {  
    if ($i % 3 == 0 && $i % 5 == 0) {  
        echo "FizzBuzz<br>";  
    } elseif ($i % 3 == 0) {  
        echo "Fizz<br>";  
    } elseif ($i % 5 == 0) {  
        echo "Buzz<br>";  
    } else {  
        echo $i . "<br>";  
    }  
}  
?  
?>
```

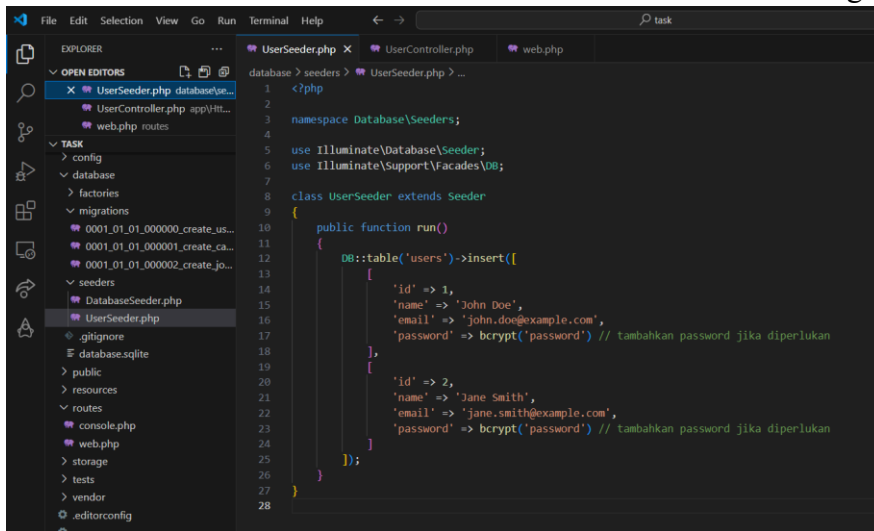
3. <?php

```
use Illuminate\Database\Migrations\Migration;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Support\Facades\Schema;
```

return new class extends Migration

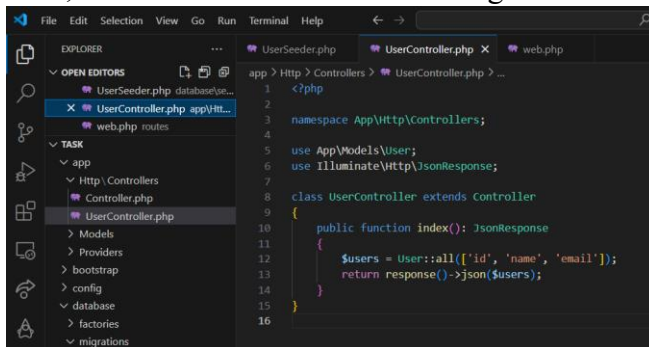
```
{  
    /**  
     * Run the migrations.  
     */  
    public function up(): void  
    {  
        Schema::create('users', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->string('email')->unique();  
            $table->timestamp('email_verified_at')->nullable();  
            $table->string('password');  
            $table->rememberToken();  
            $table->timestamps();  
        });  
    }  
};
```

#### 4. Memasukkan data tersebut ke dalam database terlebih dahulu dengan userseeder



```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6 use Illuminate\Support\Facades\DB;
7
8 class UserSeeder extends Seeder
9 {
10     public function run()
11     {
12         DB::table('users')->insert([
13             [
14                 'id' => 1,
15                 'name' => 'John Doe',
16                 'email' => 'john.doe@example.com',
17                 'password' => bcrypt('password') // tambahkan password jika diperlukan
18             ],
19             [
20                 'id' => 2,
21                 'name' => 'Jane Smith',
22                 'email' => 'jane.smith@example.com',
23                 'password' => bcrypt('password') // tambahkan password jika diperlukan
24             ]
25         ]);
26     }
27 }
28
```

Lalu, Membuat Controller untuk mengembalikan data pengguna dengan format json



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use Illuminate\Http\JsonResponse;
7
8 class UserController extends Controller
9 {
10     public function index(): JsonResponse
11     {
12         $users = User::all(['id', 'name', 'email']);
13         return response()->json($users);
14     }
15 }
16
```

Output :



#### 5. Berikut adalah daftar kelas

	id	class_name
<input type="checkbox"/> Edit Copy Delete	1	Kelas A
<input type="checkbox"/> Edit Copy Delete	2	Kelas B
<input type="checkbox"/> Edit Copy Delete	3	Kelas C
With selected: Edit Copy		

## Nama Siswa

				id	name	class_id
<input type="checkbox"/>		Edit		Copy		Delete
	1	Alice	1			
<input type="checkbox"/>		Edit		Copy		Delete
	2	Bob	1			
<input type="checkbox"/>		Edit		Copy		Delete
	3	Charlie	1			
<input type="checkbox"/>		Edit		Copy		Delete
	4	David	2			
<input type="checkbox"/>		Edit		Copy		Delete
	5	Eve	2			
<input type="checkbox"/>		Edit		Copy		Delete
	6	Frank	2			
<input type="checkbox"/>		Edit		Copy		Delete
	7	Grace	2			
<input type="checkbox"/>		Edit		Copy		Delete
	8	Hannah	2			
<input type="checkbox"/>		Edit		Copy		Delete
	9	Ivy	2			
<input type="checkbox"/>		Edit		Copy		Delete
	10	Jack	3			

## Pendaftaran

				student_id	class_id	enrollment_date
<input type="checkbox"/>		Edit		Copy		Delete
	1	1	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	2	1	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	3	1	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	4	2	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	5	2	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	6	2	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	7	2	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	8	2	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	9	2	2024-08-20			
<input type="checkbox"/>		Edit		Copy		Delete
	10	3	2024-08-20			

Query :

```
SELECT
    c.class_name,
    COUNT(e.student_id) AS student_count,
    (SELECT COUNT(student_id) FROM enrollments) AS total_students
FROM
    classes c
JOIN
    enrollments e ON c.id = e.class_id
GROUP BY
    c.class_name
HAVING
    COUNT(e.student_id) > 5;
```

6. <?php

```
namespace App\Models;
```

```
use Illuminate\Foundation\Auth\User as Authenticatable;
```

```
use Laravel\Passport\HasApiTokens;
```

```
class User extends Authenticatable
```

```
{
```

```
    use HasApiTokens, Notifiable;
```

```
}
```