



JOBSHEET

Nama Program Pelatihan	: Pengembangan Web dengan Node.js dan React
Kode Unit Kompetensi	: J.620100.018.02
Judul Unit Kompetensi	: Mengimplementasikan Pemrograman Berorientasi Objek
Kejuruan	: Teknologi Informasi dan Komunikasi
No. Jobsheet	:
<p>A. Judul Penerapan Konsep Pemrograman Berorientasi Objek (OOP) Menggunakan Class dalam Bahasa Pemrograman JavaScript</p> <p>B. Tujuan Peserta kompeten mengimplementasikan pemrograman berorientasi objek (PBO) menggunakan class dalam bahasa pemrograman javascript menggunakan Node.js dan React sesuai standar pemrograman berorientasi objek yang ada.</p> <p>C. Peralatan dan Bahan</p> <ol style="list-style-type: none">1. Laptop / PC2. Text Editor (Visual Code)3. Terminal / Command Prompt4. Node.js versi terbaru <p>D. Petunjuk Kegiatan Praktik</p> <ol style="list-style-type: none">1. Buka text editor dan buat folder bernama latihan_pbo di Local Disk (D:)2. Simpan file latihan dengan ekstensi .js3. Jalankan program menggunakan perintah node nama-file.js4. Periksa output program di terminal.5. Hindari kesalahan sintaks. <p>E. Langkah kerja</p> <ol style="list-style-type: none">1. Membuka code editor Visual Studio Code.	

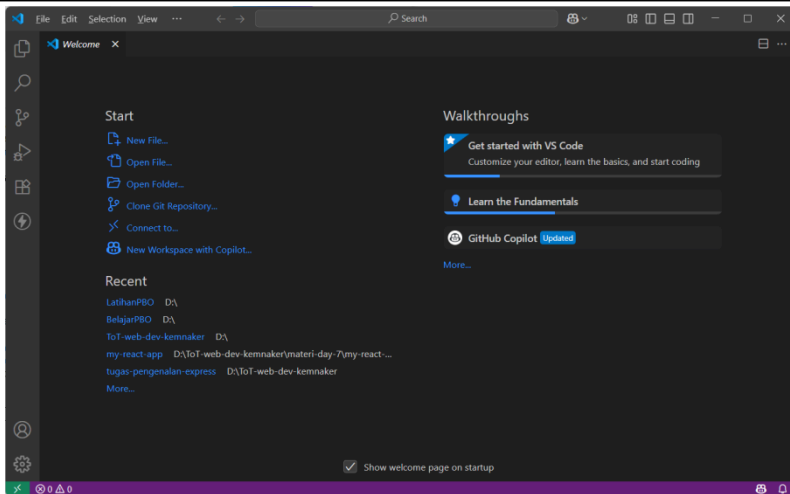


BBPVP dan BPVP

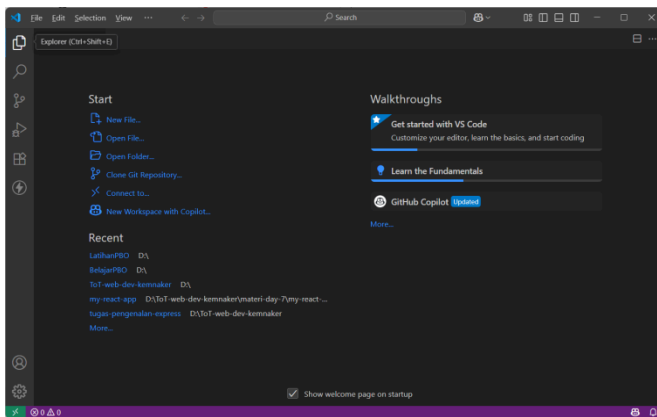
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas

Kementerian Ketenagakerjaan R.I

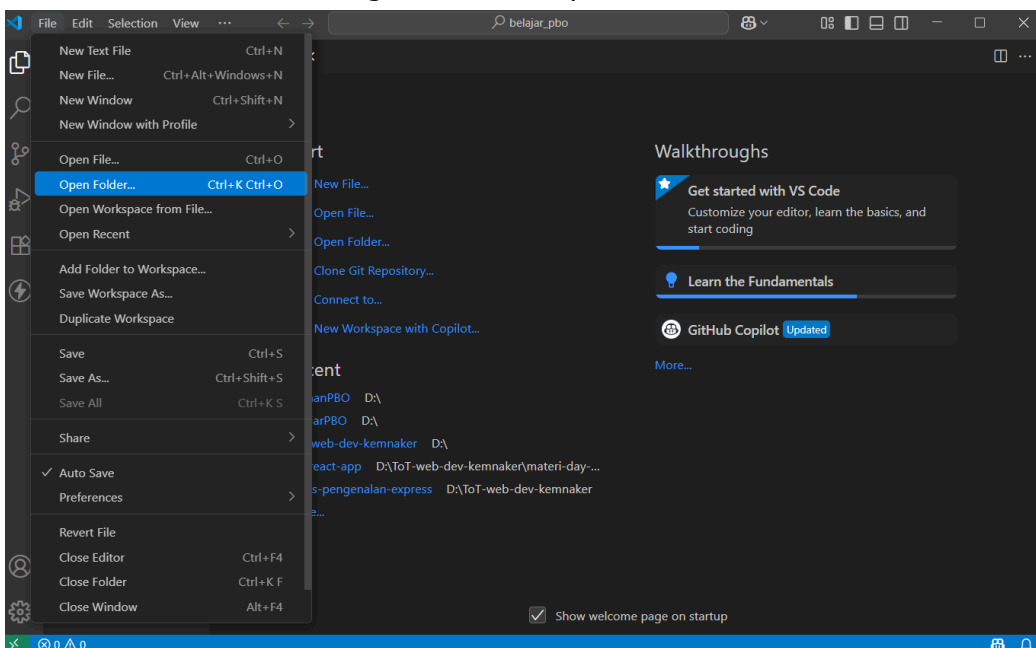
- Smart Creative Skills -



2. Klik Explorer atau Ctrl + Shift + E untuk membuka jendela explorer



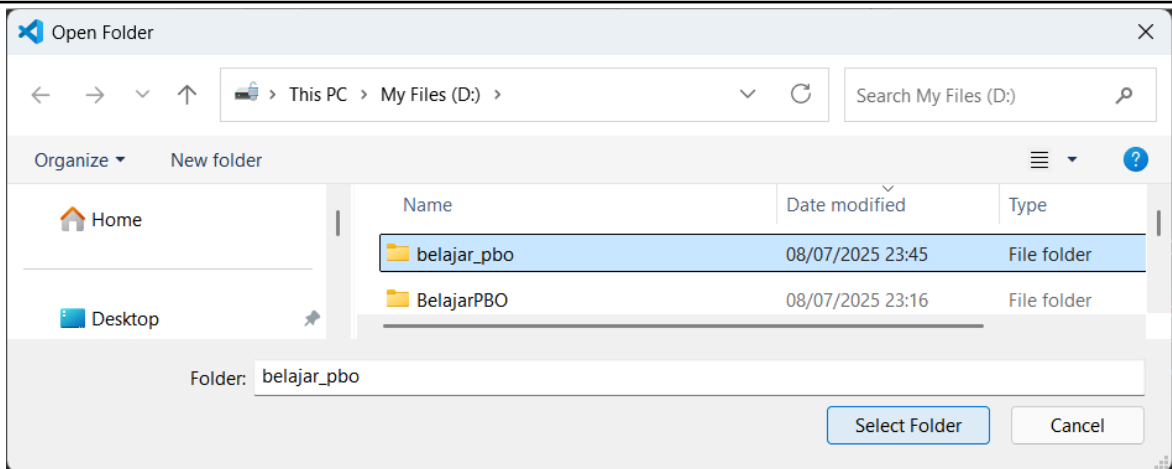
3. Membuat folder baru dengan klik File - Open Folder atau Ctrl + K Ctrl + O



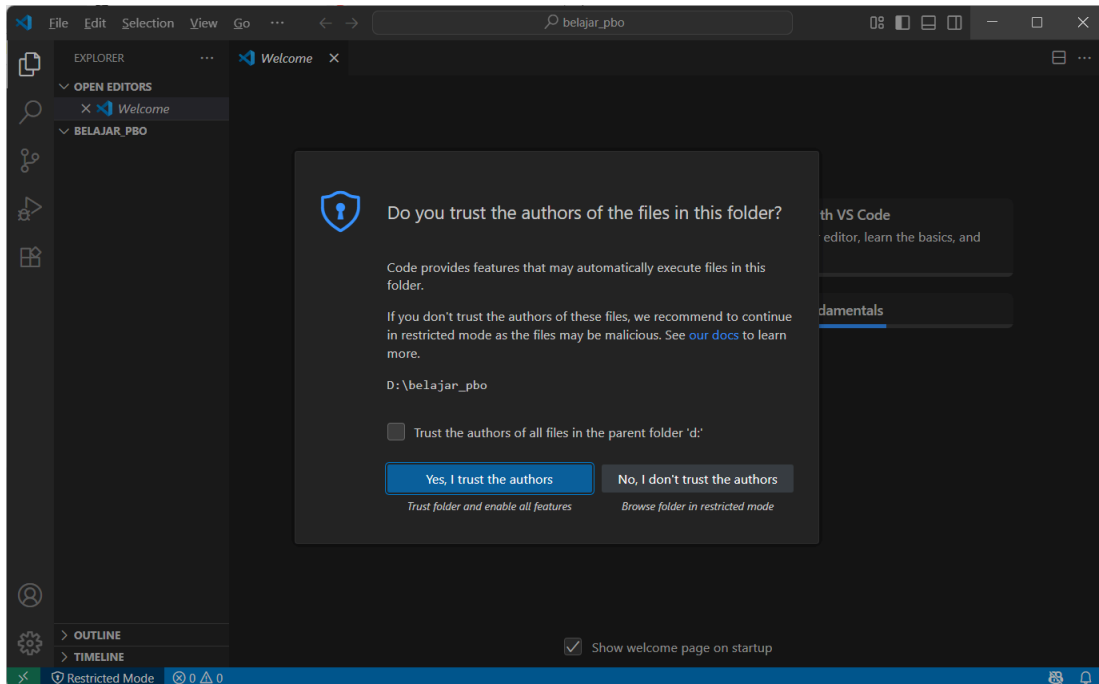
4. Membuat nama folder baru dengan nama belajar_pbo



BBPVP dan BPVP
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas
Kementerian Ketenagakerjaan R.I
- Smart Creative Skills -



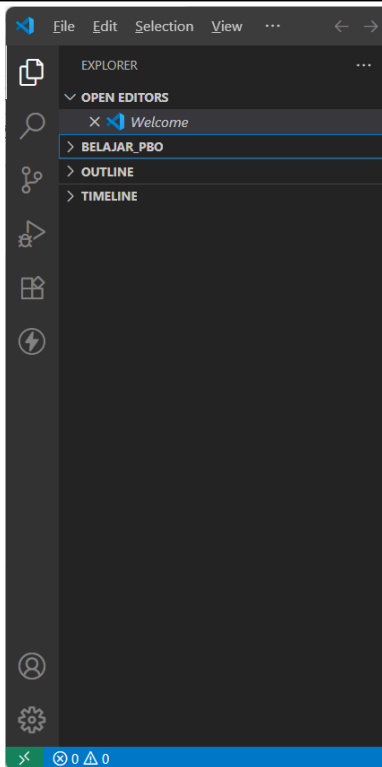
5. Pilih Yes, I trust the author



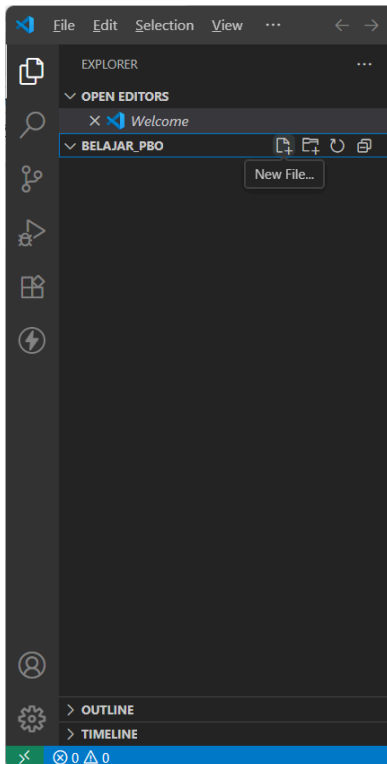
6. Pada jendela explorer tampil folder belajar_pbo.



BBPVP dan BPVP
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas
Kementerian Ketenagakerjaan R.I
- Smart Creative Skills -



7. Di explorer folder belajar_pbo, klik New File untuk membuat halaman baru.



8. Beri nama pada file class.js

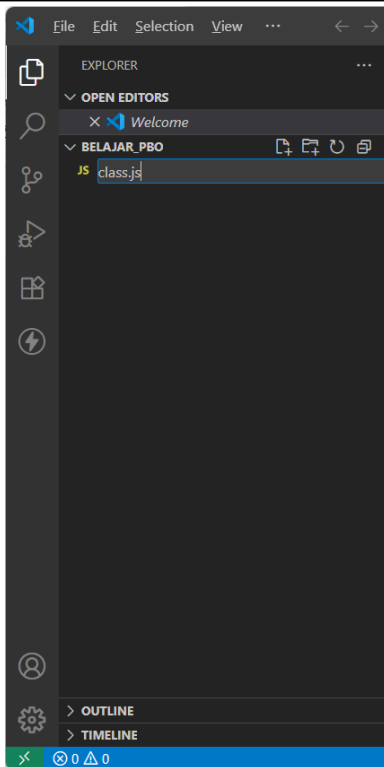


BBPVP dan BPVP

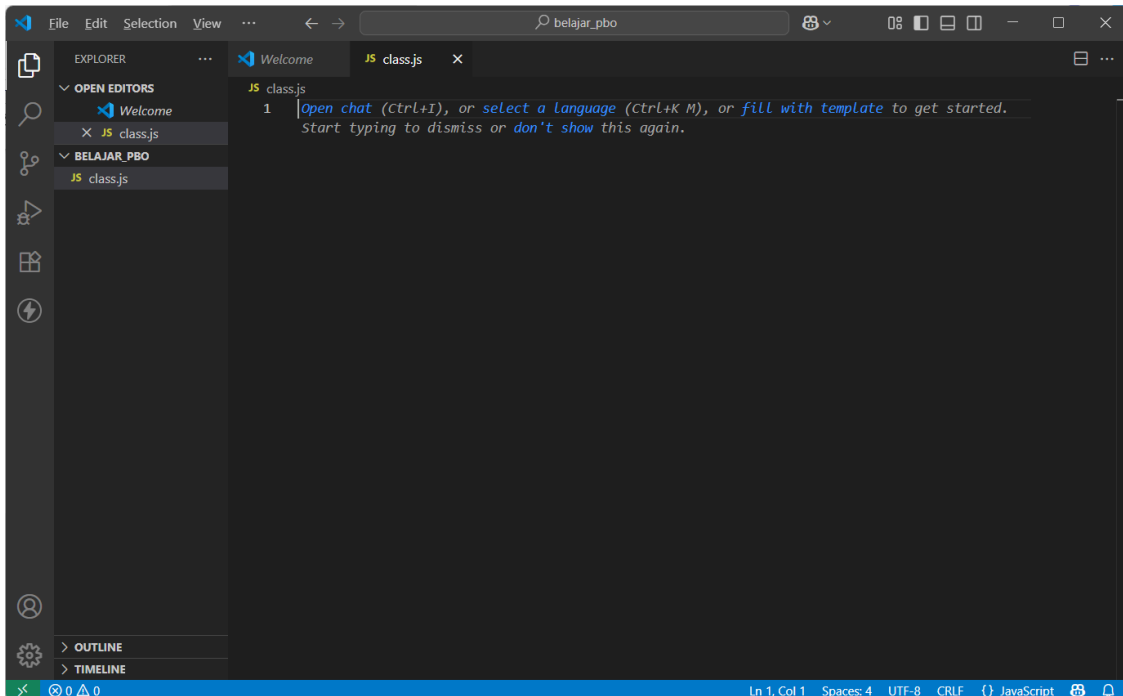
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas

Kementerian Ketenagakerjaan R.I

- Smart Creative Skills -



9. Klik pada tab class.js untuk mengetik perintah.



10. Membuat class dengan nama Person yang memiliki constructor dan property (name, age)

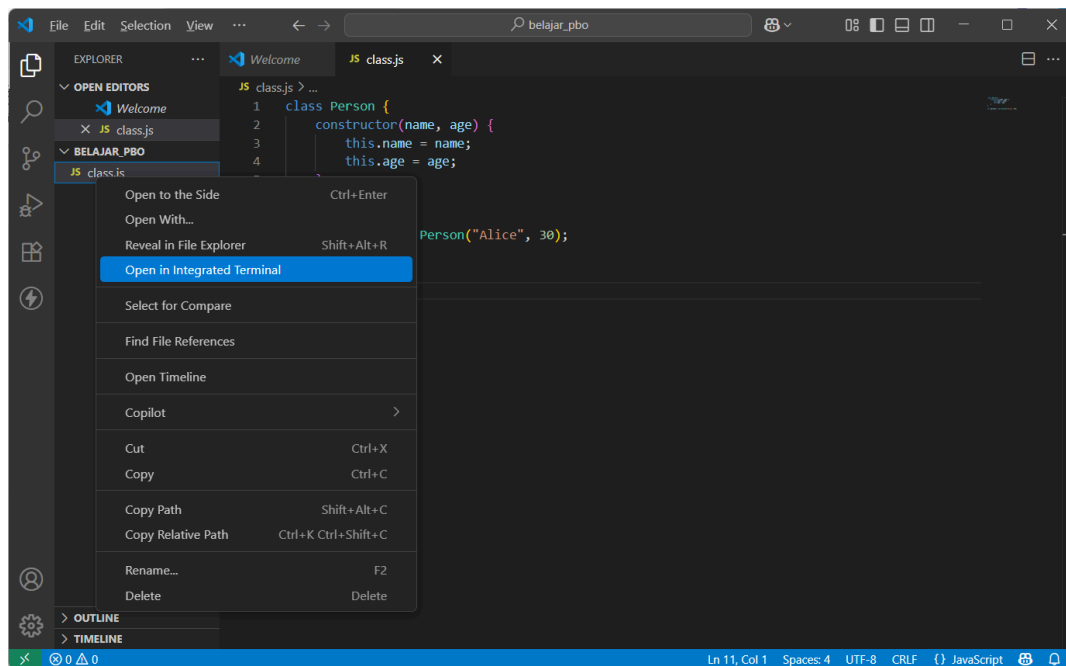


```
class Person {  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
}
```

11. Membuat objek (instansiasi) dari class Person dengan menambahkan baris perintah ini dibawah class.

```
class Person {  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
}  
  
var person1 = new Person("Alice", 30);  
console.log(person1.name); // Output: Alice  
console.log(person1.age);  // Output: 30
```

12. Jalankan perintah dengan cara klik kanan pada file class.js, pilih Open in integrated Terminal



13. Ketikkan perintah berikut pada terminal

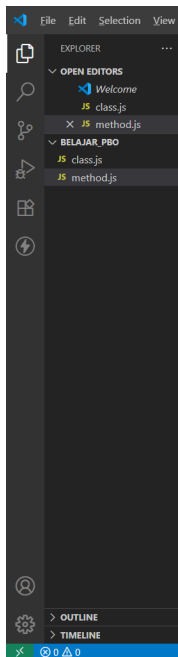


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\belajar_pbo> node class.js
```

14. Output yang ditampilkan adalah Alice dan 30

```
PS D:\belajar_pbo> node class.js  
Alice  
30
```

15. Membuat file baru dengan nama method.js



16. Ketikkan perintah berikut ini pada file method.js untuk membuat method say dalam class

```
// Method  
class Person {  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
  
  // method  
  say() {  
    return "Hello, my name is " + this.name + " and I am " + this.age + " years old.";  
  }  
}
```

17. Membuat objek (instansiasi) dari class Person dengan menambahkan baris perintah ini dibawah class.



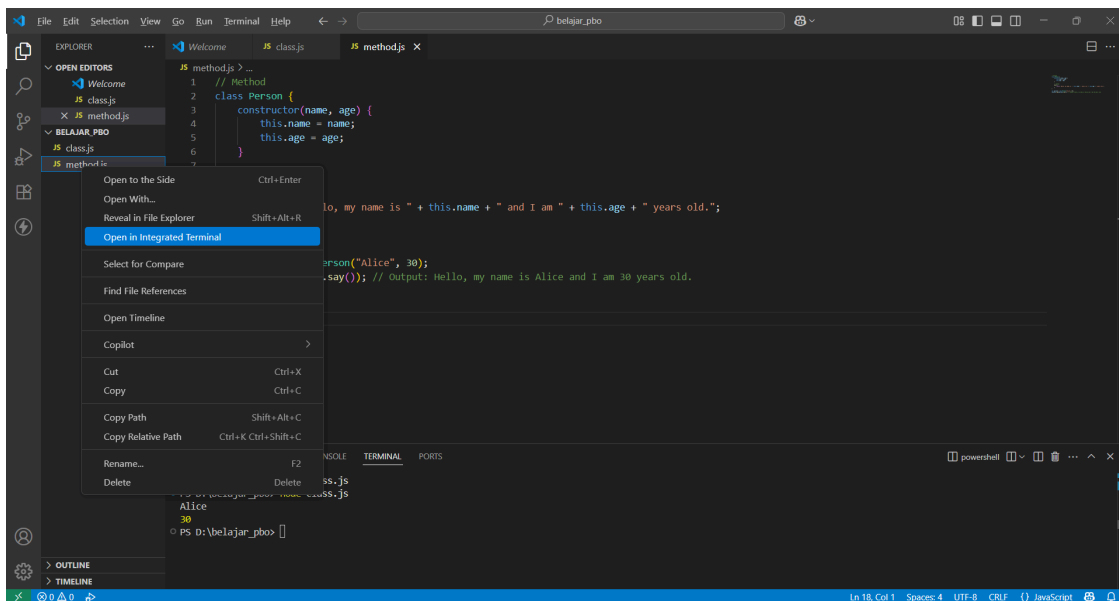
BBPVP dan BPVP
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas
Kementerian Ketenagakerjaan R.I
- Smart Creative Skills -

```
// Method
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

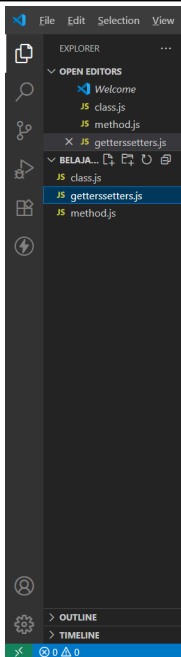
  // method
  say() {
    return "Hello, my name is " + this.name + " and I am " + this.age + " years old.";
  }
}

var person1 = new Person("Alice", 30);
console.log(person1.say()); // Output: Hello, my name is Alice and I am 30 years old.
```

18. Jalankan perintah dengan cara klik kanan pada file method.js, pilih Open in integrated Terminal



19. Ketikkan perintah berikut pada terminal
PS D:\belajar_pbo> node method.js
20. Output yang ditampilkan pada terminal.
Hello, my name is Alice and I am 30 years old.
21. Membuat file baru dengan nama getterssetters.js



22. Ketikkan perintah berikut ini pada file getterssetters.js untuk membuat dapat mengambil property brand dan factory dari objek dengan perintah get dan mengubah nilai objek dengan set dalam class

```
class Car{
  constructor(brand,factory) {
    this._carname = brand;
    this._factory = factory;
  }
  get carname() {
    return this._carname;
  }
  set carname(x) {
    this._carname = x;
  }
  get factory() {
    return this._factory;
  }
  set factory(x) {
    this._factory = x;
  }
}
```

23. Instansiasi objek dengan nama objek newcar dengan perintah berikut.



BBPVP dan BPVP

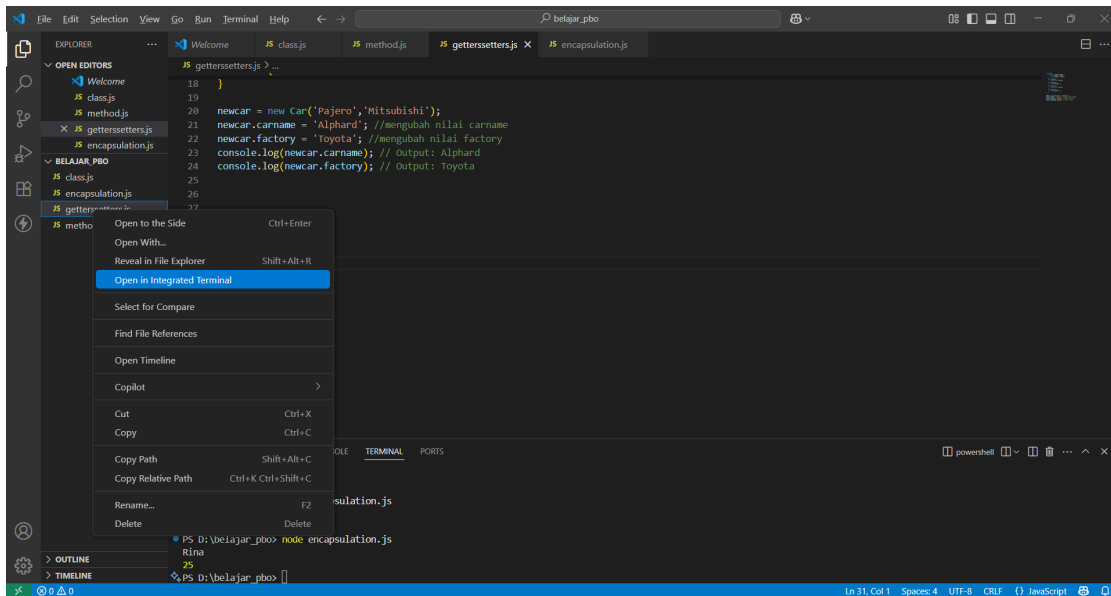
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas

Kementerian Ketenagakerjaan R.I

- Smart Creative Skills -

```
newcar = new Car('Pajero','Mitsubishi');  
newcar.carname = 'Alphard'; //mengubah nilai carname  
newcar.factory = 'Toyota'; //mengubah nilai factory  
console.log(newcar.carname); // Output: Alphard  
console.log(newcar.factory); // Output: Toyota
```

24. Jalankan perintah dengan cara klik kanan pada file method.js, pilih Open in integrated Terminal.



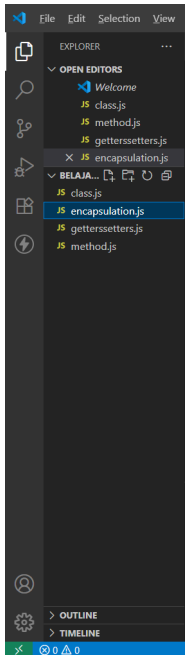
25. Ketikkan perintah node getterssetters.js pada terminal

```
PS D:\belajar_pbo> node getterssetters.js
```

26. Output akan ditampilkan di terminal

```
Alphard  
Toyota
```

27. Membuat file baru dengan nama encapsulation.js



28. Ketikkan perintah berikut pada file encapsulation.js untuk membuat property age menjadi private dengan menambahkan tanda # sebelum age.

```
class Person {  
  #age; // Properti benar-benar private  
  
  constructor(name, age) {  
    this.name = name; // Public  
    this.#age = age;   // Private  
  }  
  
  // Getter  
  getAge() {  
    return this.#age;  
  }  
  
  // Setter dengan validasi  
  setAge(newAge) {  
    if (newAge < 0 || newAge > 150) {  
      console.log('Umur tidak valid');  
    } else {  
      this.#age = newAge;  
    }  
  }  
}
```

29. Instansiasi objek dengan nama objek newcar dengan perintah berikut.



BBPVP dan BPVP

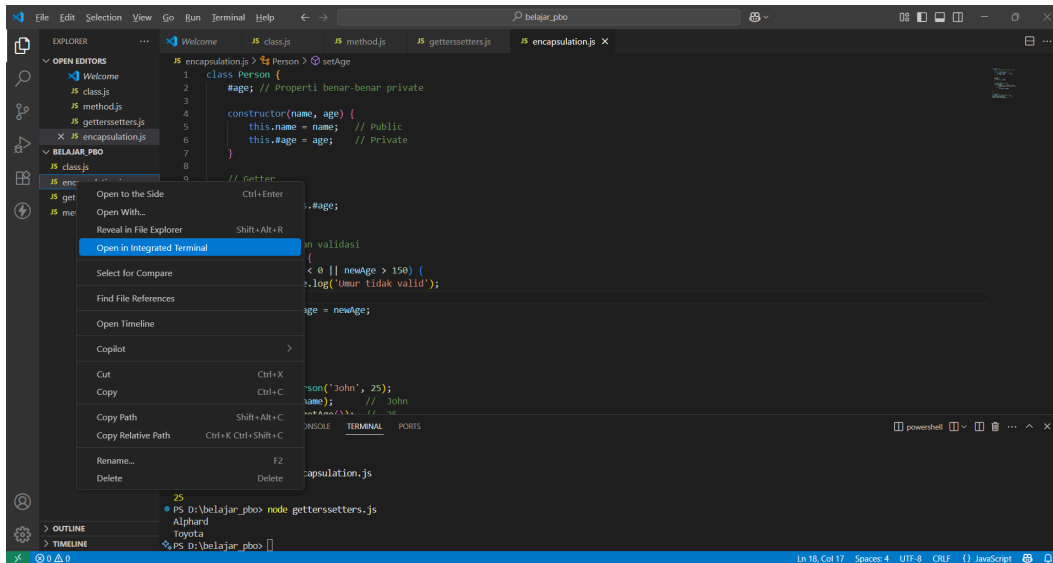
Direktorat Jenderal Pembinaan Pelatihan Vokasi dan Produktivitas

Kementerian Ketenagakerjaan R.I

- Smart Creative Skills -

```
var orang = new Person('John', 25);  
console.log(orang.name);           // John  
console.log(orang.getAge());       // 25
```

30. Jalankan perintah dengan cara klik kanan pada file encapsulation.js, pilih Open in integrated Terminal.



31. Ketikkan perintah node encapsulation.js pada terminal

```
PS D:\belajar_pbo> node encapsulation.js
```

32. Output akan ditampilkan di terminal.

```
John  
25
```