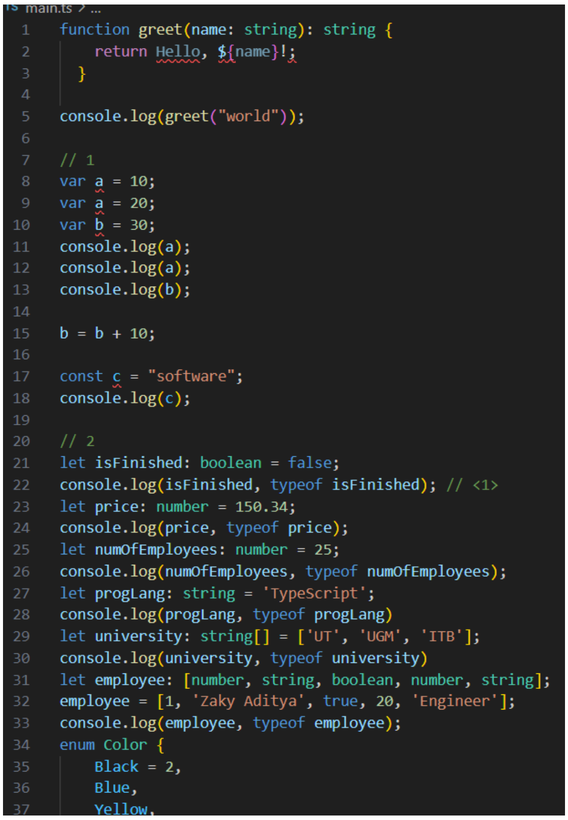
Nama : Ulfah Siti Nurhidayah

NIM : 22060002

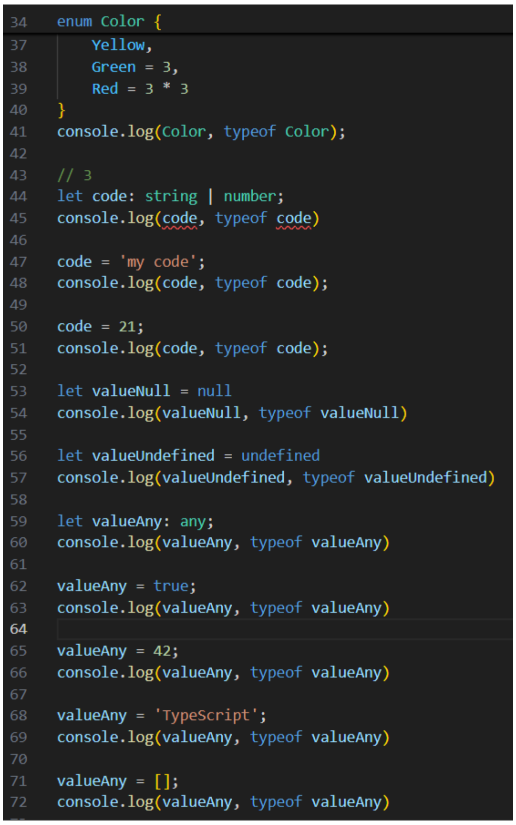
Tugas : UTS

Latihan !!

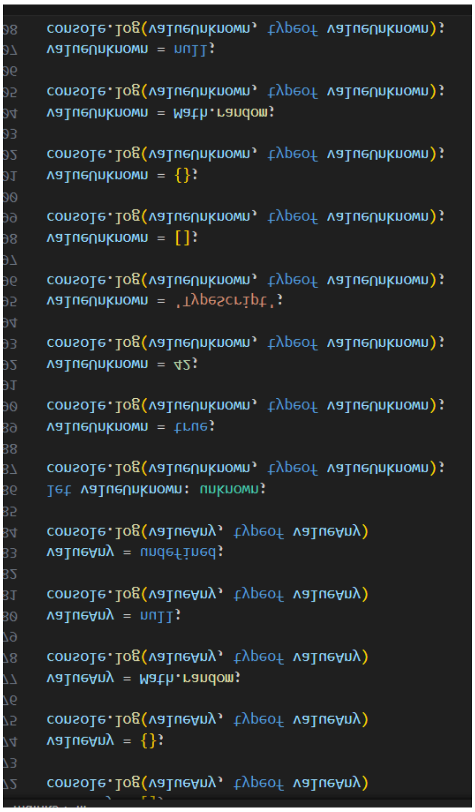
Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut! Selain menggunakan tsc, cobalah semua kode sumber yang ada pada kegiatan belajar ini dengan menggunakan Deno. Perhatikan baik-baik, adakah kode sumber mana yang tidak dapat dijalankan dengan menggunakan Deno? Petunjuk Jawaban Latihan Kunjungi web Deno di https://deno.land/. Install Deno sesuai petunjuk pada URL tersebut. Setelah itu, jalankan berbagai kode sumber TypeScript yang ada pada kegiatan belajar ini dengan menggunakan petunjuk di manual (https://deno.land/manual), khususnya pada Getting Started - Command line interface.



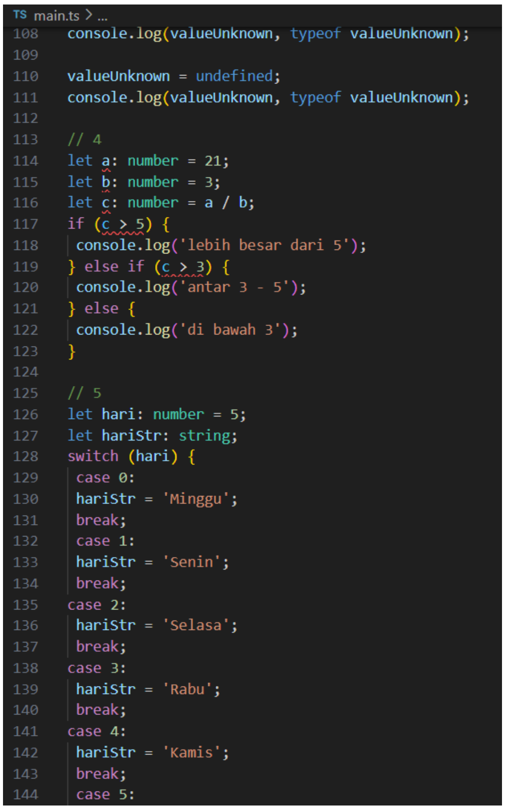
```
function greet(name: string): string {
    return Hello, ${name}!;
  }

console.log(greet("world"));

// 1
var a = 10;
var a = 20;
var b = 30;
console.log(a);
console.log(a);
console.log(b);

b = b + 10;

const c = "software";
console.log(c);

// 2
let isFinished: boolean = false;
console.log(isFinished, typeof isFinished); // <1>
let price: number = 150.34;
console.log(price, typeof price);
let numOfEmployees: number = 25;
console.log(numOfEmployees, typeof numOfEmployees);
let progLang: string = 'TypeScript';
console.log(progLang, typeof progLang)
let university: string[] = ['UT', 'UGM', 'ITB'];
console.log(university, typeof university)
let employee: [number, string, boolean, number, string];
employee = [1, 'Zaky Aditya', true, 20, 'Engineer'];
console.log(employee, typeof employee);
enum Color {
    Black = 2,
    Blue,
    Yellow,
```



```
enum Color {
    Yellow,
    Green = 3,
    Red = 3 * 3
}
console.log(Color, typeof Color);

// 3
let code: string | number;
console.log(code, typeof code)

code = 'my code';
console.log(code, typeof code);

code = 21;
console.log(code, typeof code);

let valueNull = null
console.log(valueNull, typeof valueNull)

let valueUndefined = undefined
console.log(valueUndefined, typeof valueUndefined)

let valueAny: any;
console.log(valueAny, typeof valueAny)

valueAny = true;
console.log(valueAny, typeof valueAny)

valueAny = 42;
console.log(valueAny, typeof valueAny)

valueAny = 'TypeScript';
console.log(valueAny, typeof valueAny)

valueAny = [];
console.log(valueAny, typeof valueAny)
```

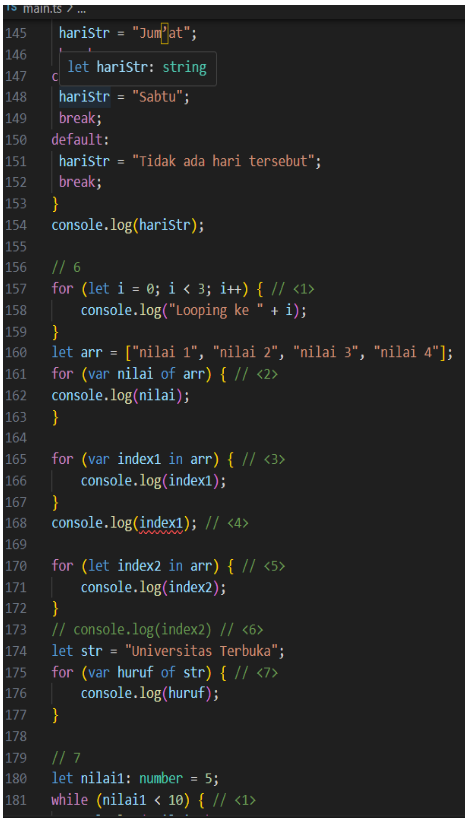


```
console.log(valueAny, typeof valueAny)
valueAny = {};
console.log(valueAny, typeof valueAny)

valueAny = Math.random;
console.log(valueAny, typeof valueAny)

valueAny = null;
console.log(valueAny, typeof valueAny)

valueAny = undefined;
console.log(valueAny, typeof valueAny)

let valueUnknown: unknown;
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = true;
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = 42;
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = 'TypeScript';
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = [];
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = {};
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = Math.random;
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = null;
console.log(valueUnknown, typeof valueUnknown);
```



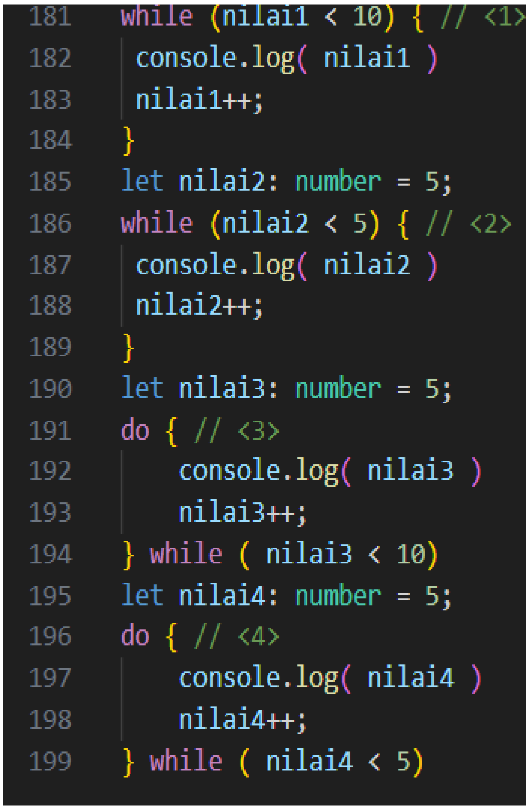
main.ts > ...

```
console.log(valueUnknown, typeof valueUnknown);

valueUnknown = undefined;
console.log(valueUnknown, typeof valueUnknown);

// 4
let a: number = 21;
let b: number = 3;
let c: number = a / b;
if (c > 5) {
 console.log('lebih besar dari 5');
} else if (c > 3) {
 console.log('antar 3 - 5');
} else {
 console.log('di bawah 3');
}

// 5
let hari: number = 5;
let hariStr: string;
switch (hari) {
 case 0:
 hariStr = 'Minggu';
 break;
 case 1:
 hariStr = 'Senin';
 break;
case 2:
 hariStr = 'Selasa';
 break;
case 3:
 hariStr = 'Rabu';
 break;
case 4:
 hariStr = 'Kamis';
 break;
 case 5:
```

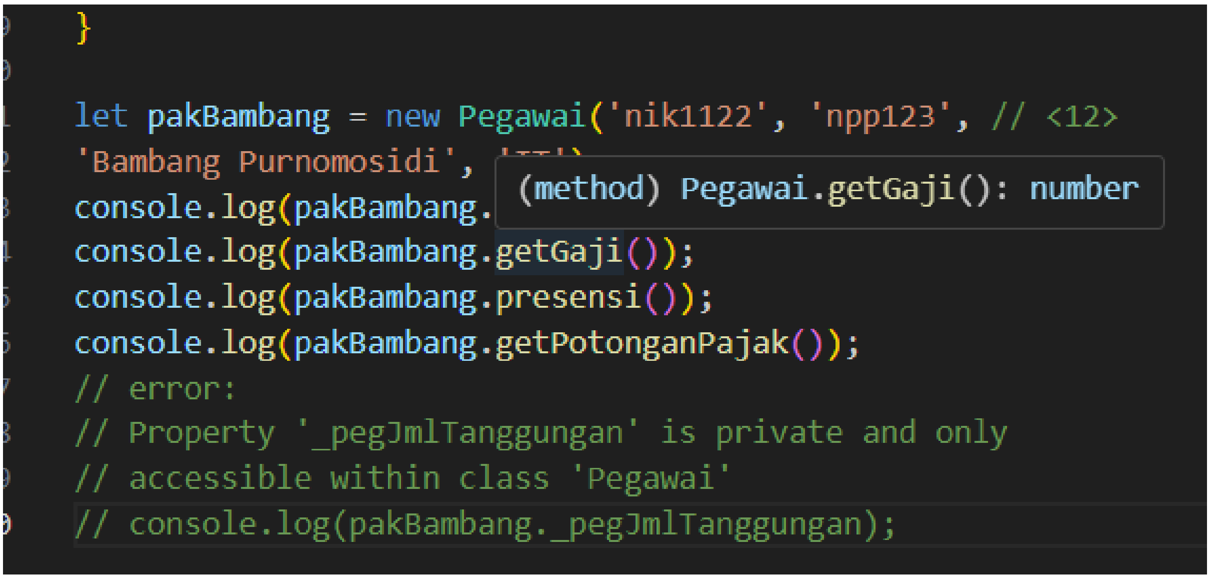


main.ts > ...

```
 hariStr = "Jum'at";
 
c  let hariStr: string
 hariStr = "Sabtu";
 break;
default:
 hariStr = "Tidak ada hari tersebut";
 break;
}
console.log(hariStr);

// 6
for (let i = 0; i < 3; i++) { // <1>
    console.log("Looping ke " + i);
}
let arr = ["nilai 1", "nilai 2", "nilai 3", "nilai 4"];
for (var nilai of arr) { // <2>
console.log(nilai);
}

for (var index1 in arr) { // <3>
    console.log(index1);
}
console.log(index1); // <4>

for (let index2 in arr) { // <5>
    console.log(index2);
}
// console.log(index2) // <6>
let str = "Universitas Terbuka";
for (var huruf of str) { // <7>
    console.log(huruf);
}

// 7
let nilai1: number = 5;
while (nilai1 < 10) { // <1>
```

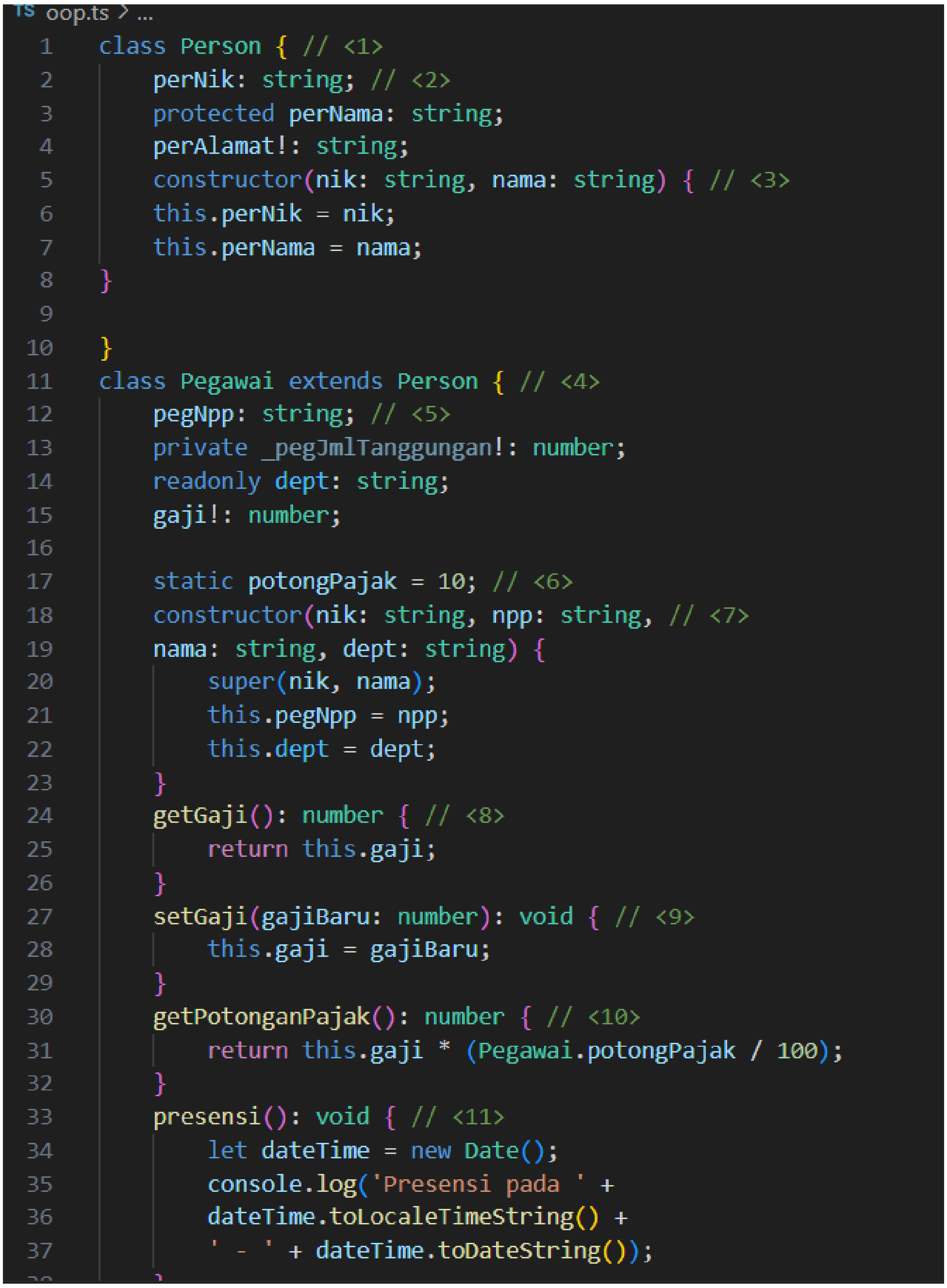


```
while (nilai1 < 10) { // <1>
 console.log( nilai1 )
 nilai1++;
}
let nilai2: number = 5;
while (nilai2 < 5) { // <2>
 console.log( nilai2 )
 nilai2++;
}
let nilai3: number = 5;
do { // <3>
    console.log( nilai3 )
    nilai3++;
} while ( nilai3 < 10)
let nilai4: number = 5;
do { // <4>
    console.log( nilai4 )
    nilai4++;
} while ( nilai4 < 5)
```

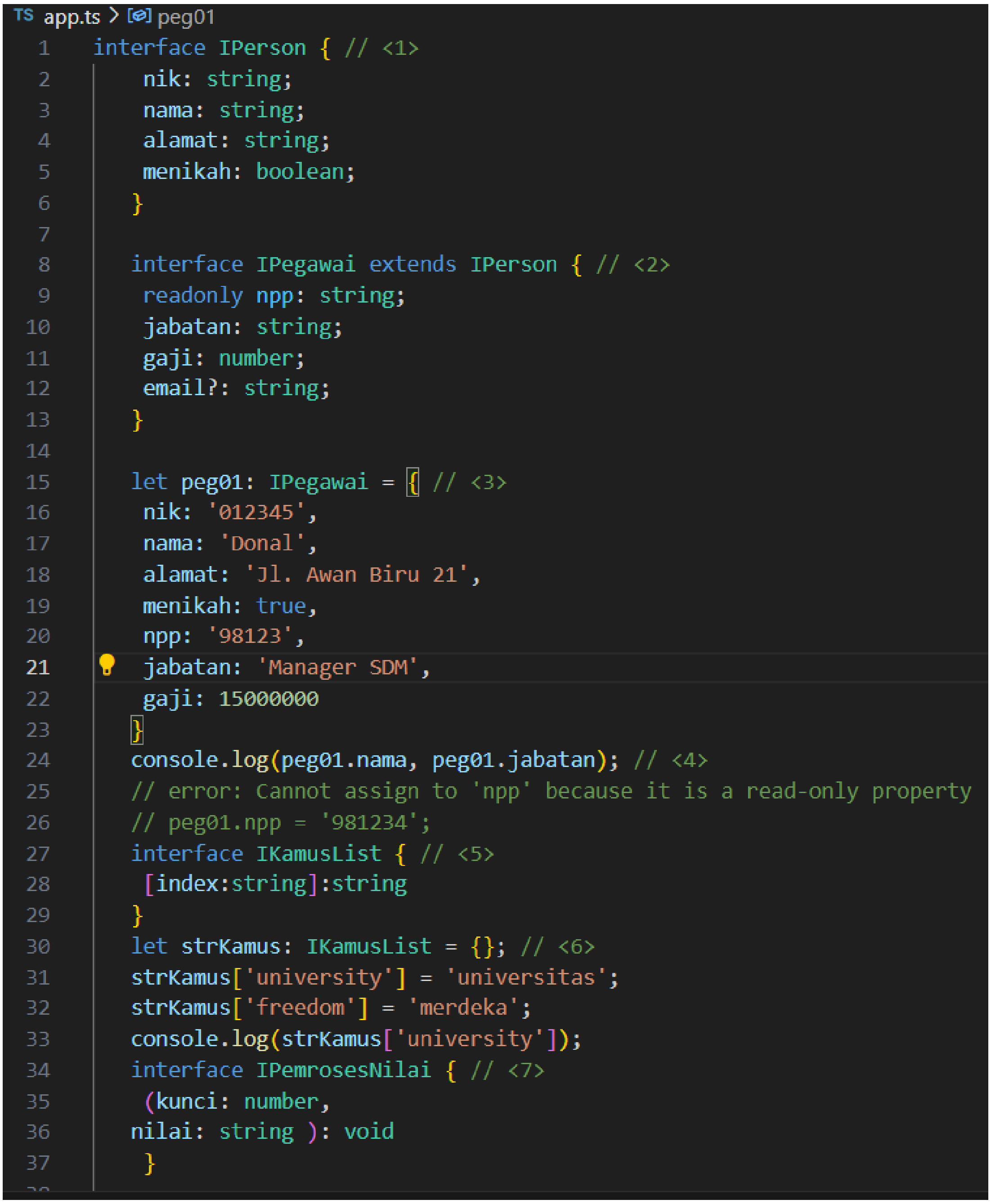


```
}

let pakBambang = new Pegawai('nik1122', 'npp123', // <12>
'Bambang Purnomosidi', 'IT');
console.log(pakBambang.
console.log(pakBambang.getGaji());
console.log(pakBambang.presensi());
console.log(pakBambang.getPotonganPajak());
// error:
// Property '_pegJmlTanggungan' is private and only
// accessible within class 'Pegawai'
// console.log(pakBambang._pegJmlTanggungan);
```

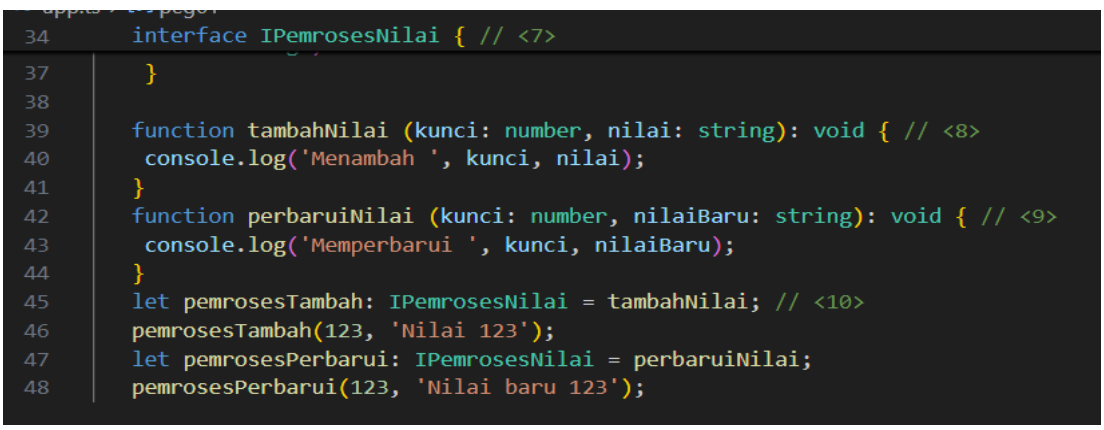
```
(method) Pegawai.getGaji(): number
```



```
class Person { // <1>
    perNik: string; // <2>
    protected perNama: string;
    perAlamat!: string;
    constructor(nik: string, nama: string) { // <3>
    this.perNik = nik;
    this.perNama = nama;
}

}
class Pegawai extends Person { // <4>
    pegNpp: string; // <5>
    private _pegJmlTanggungan!: number;
    readonly dept: string;
    gaji!: number;

    static potongPajak = 10; // <6>
    constructor(nik: string, npp: string, // <7>
    nama: string, dept: string) {
        super(nik, nama);
        this.pegNpp = npp;
        this.dept = dept;
    }
    getGaji(): number { // <8>
        return this.gaji;
    }
    setGaji(gajiBaru: number): void { // <9>
        this.gaji = gajiBaru;
    }
    getPotonganPajak(): number { // <10>
        return this.gaji * (Pegawai.potongPajak / 100);
    }
    presensi(): void { // <11>
        let dateTime = new Date();
        console.log('Presensi pada ' +
        dateTime.toLocaleTimeString() +
        ' - ' + dateTime.toDateString());
    }
```



```
interface IPerson { // <1>
    nik: string;
    nama: string;
    alamat: string;
    menikah: boolean;
   }

   interface IPegawai extends IPerson { // <2>
    readonly npp: string;
    jabatan: string;
    gaji: number;
    email?: string;
   }

   let peg01: IPegawai = { // <3>
    nik: '012345',
    nama: 'Donal',
    alamat: 'Jl. Awan Biru 21',
    menikah: true,
    npp: '98123',
    jabatan: 'Manager SDM',
    gaji: 15000000
   }
   console.log(peg01.nama, peg01.jabatan); // <4>
   // error: Cannot assign to 'npp' because it is a read-only property
   // peg01.npp = '981234';
   interface IKamusList { // <5>
    [index:string]:string
   }
   let strKamus: IKamusList = {}; // <6>
   strKamus['university'] = 'universitas';
   strKamus['freedom'] = 'merdeka';
   console.log(strKamus['university']);
   interface IPemrosesNilai { // <7>
    (kunci: number,
   nilai: string ): void
    }
```



```
   interface IPemrosesNilai { // <7>
    }

   function tambahNilai (kunci: number, nilai: string): void { // <8>
    console.log('Menambah ', kunci, nilai);
   }
   function perbaruiNilai (kunci: number, nilaiBaru: string): void { // <9>
    console.log('Memperbarui ', kunci, nilaiBaru);
   }
   let pemrosesTambah: IPemrosesNilai = tambahNilai; // <10>
   pemrosesTambah(123, 'Nilai 123');
   let pemrosesPerbarui: IPemrosesNilai = perbaruiNilai;
   pemrosesPerbarui(123, 'Nilai baru 123');
```

Tes Formatif 2

1) B
2) C
3) D
4) D
5) B
6) A
7) B
8) D
9) A
10) C