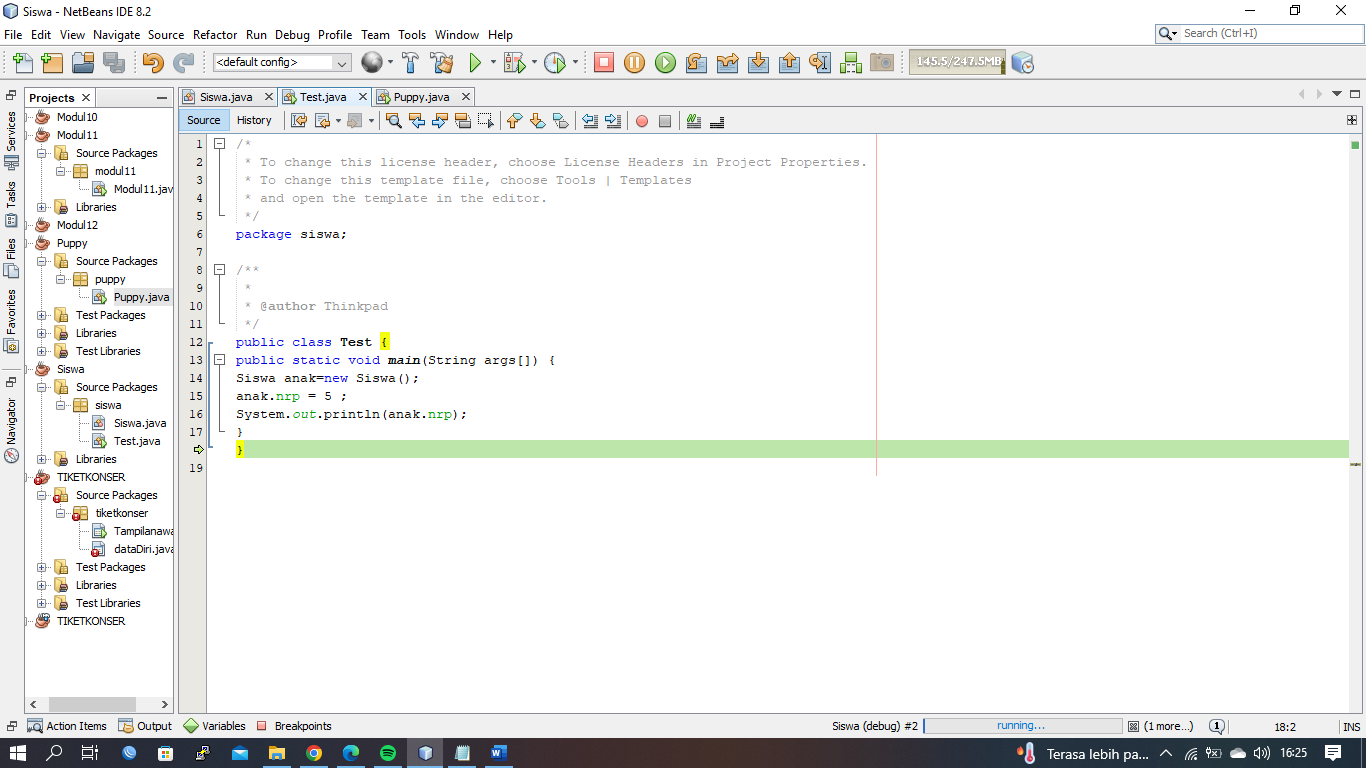
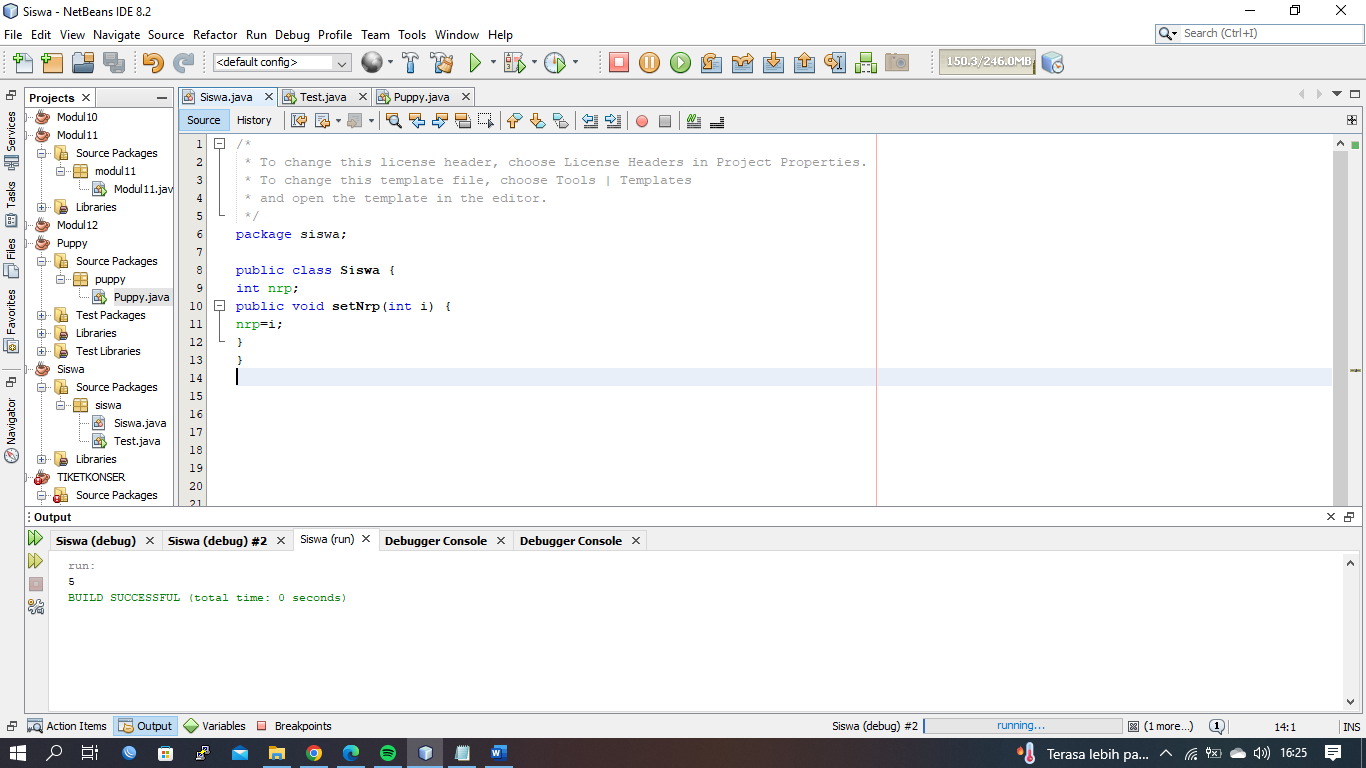
Nama : Ifan Ali Murtadho

Nim : 4123014

Kelas B Semester 2

**D. PERCOBAAN**

-Percobaan 1



-Kelas **Siswa**: Ini adalah kelas yang mewakili entitas siswa. sebagai argumen **i**. Dalam kasus ini, metode ini mengatur nilai **nrp** menjadi nilai **i**.

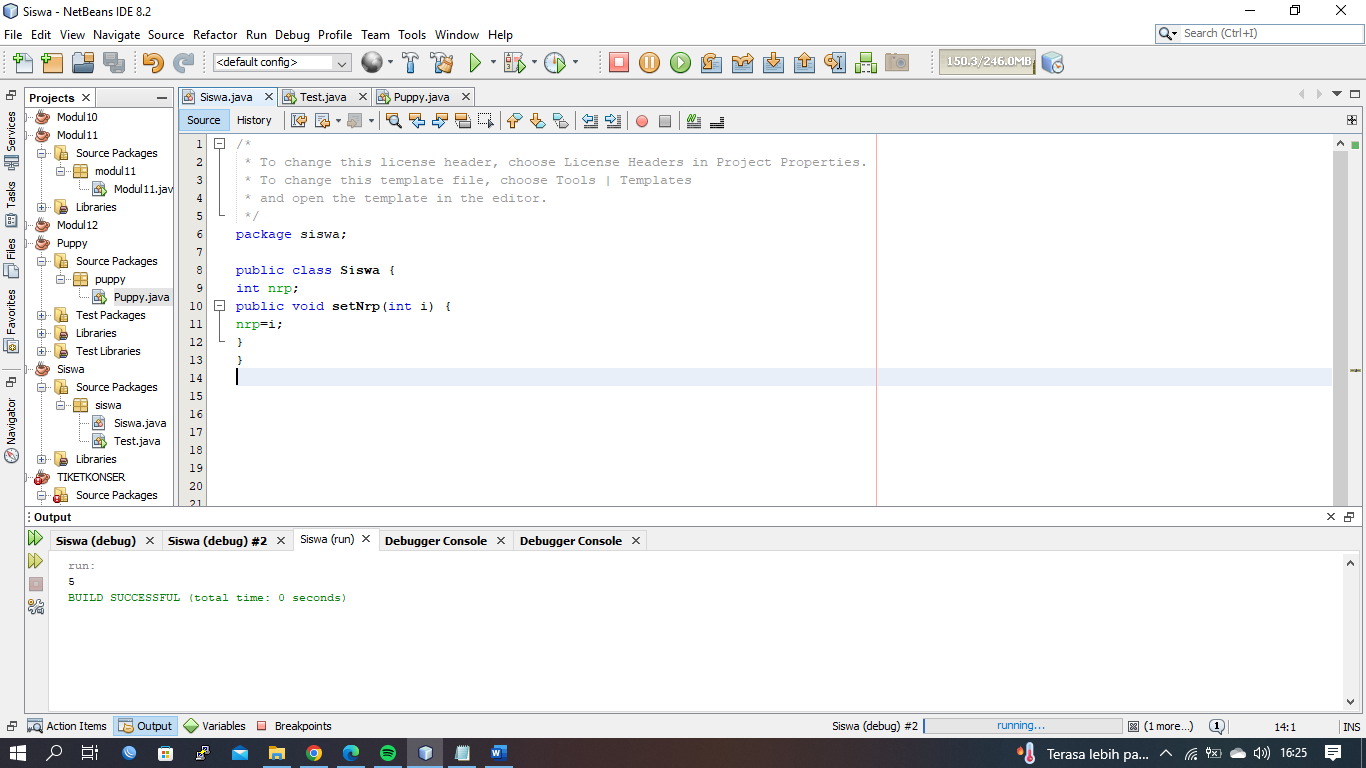
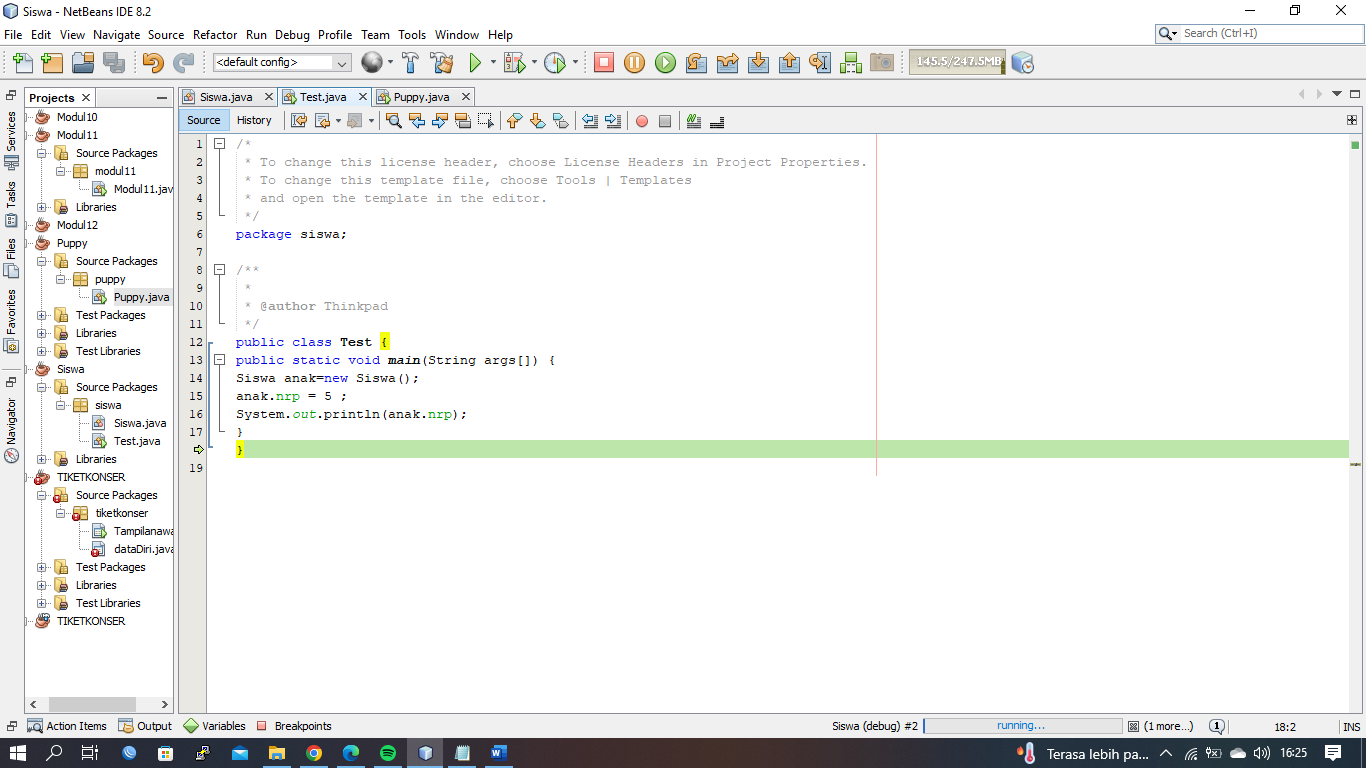
-Kelas **Test**: Ini adalah kelas yang berisi metode main yang merupakan titik masuk utama program.

-Dalam metode main, sebuah objek dari kelas Siswa dibuat dengan nama anak.

-Kemudian, nilai 5 langsung diberikan ke variabel nrp dari objek anak menggunakan operator

-Nilai nrp kemudian dicetak menggunakan System.out.println().

-Percobaan 2

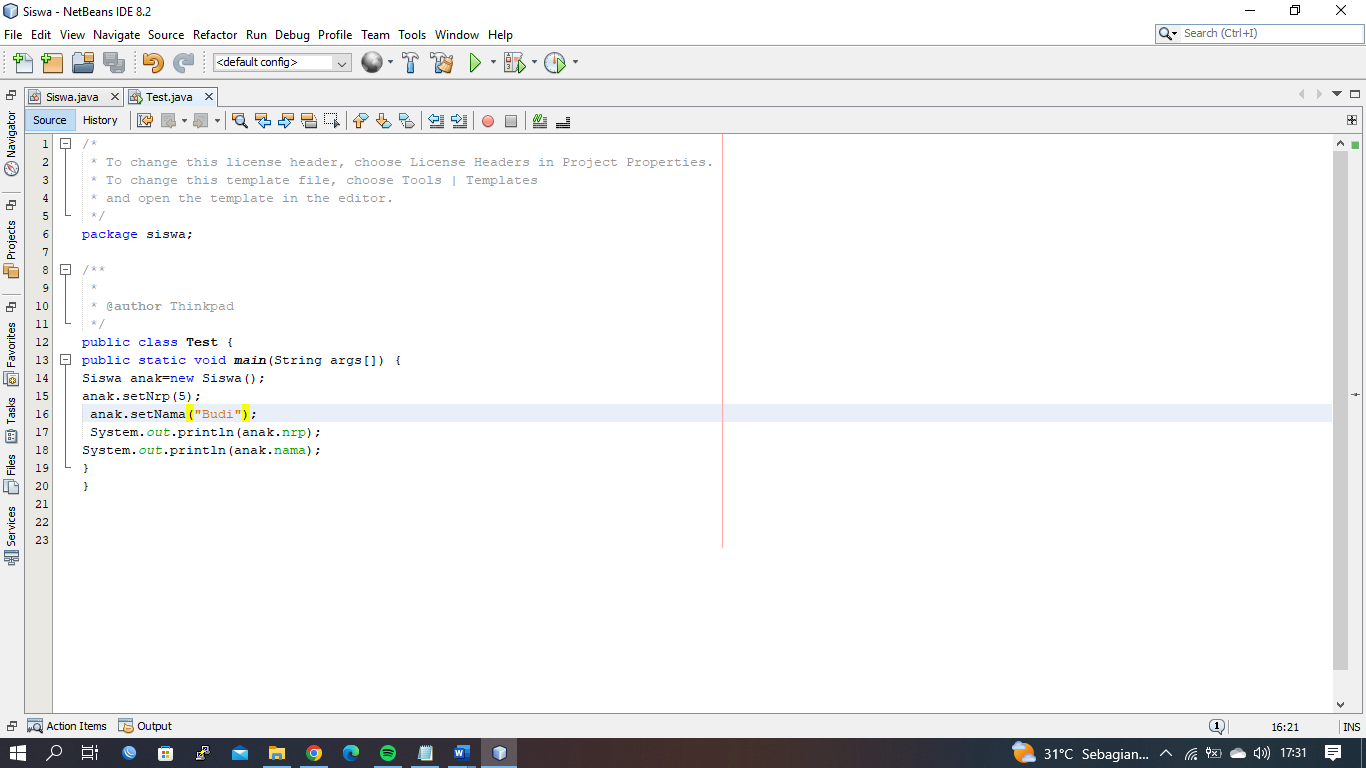
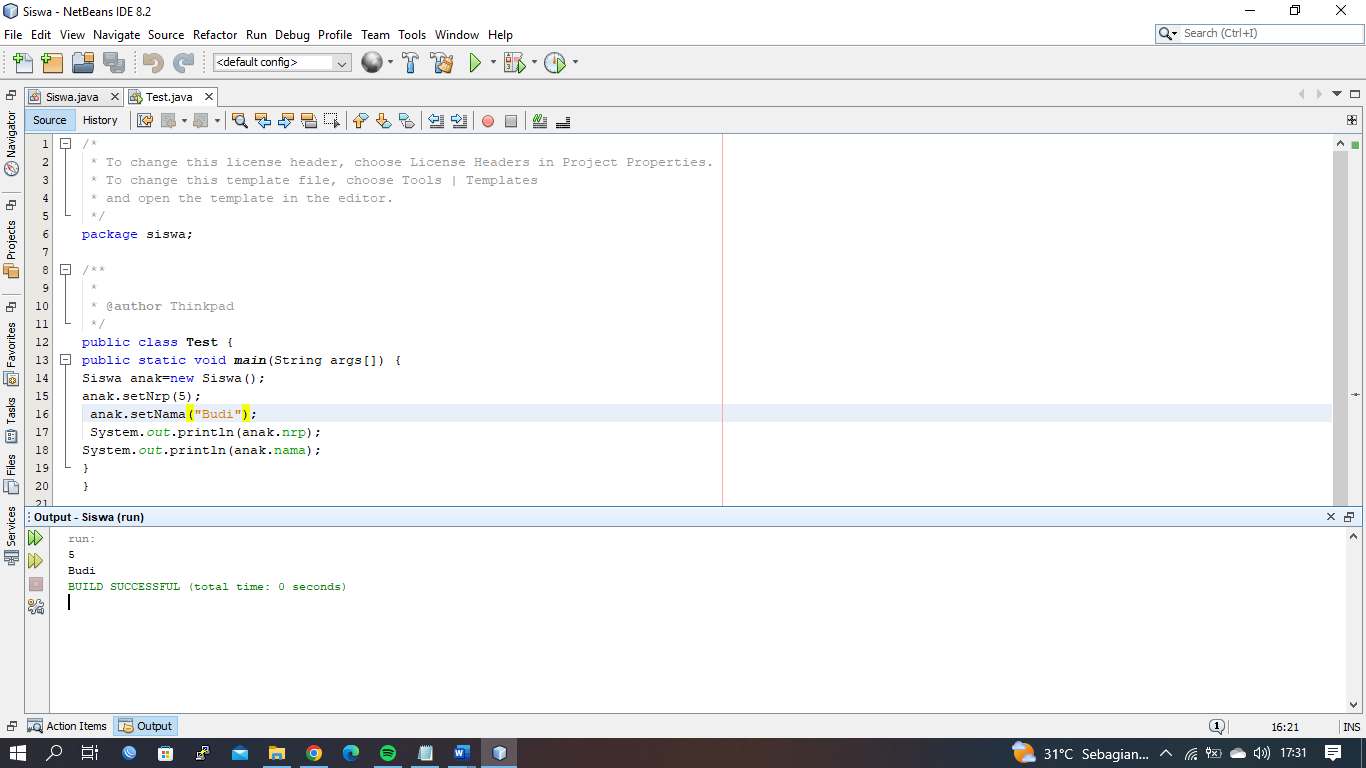


Kelas **Siswa**: Ini adalah kelas yang mewakili entitas siswa.

* Variabel **nrp**: Ini adalah variabel instance yang mewakili Nomor Registrasi Siswa (NRP).
* Metode **setNrp(int i)**: Ini adalah metode untuk menetapkan nilai **nrp** dengan nilai yang diberikan sebagai argumen **i**. Dalam hal ini, metode ini mengatur nilai **nrp** menjadi nilai **i**.

Kelas **Siswa**: Ini adalah kelas yang mewakili entitas siswa.

* Variabel **nrp**: Ini adalah variabel instance yang mewakili Nomor Registrasi Siswa (NRP).
* Metode **setNrp(int i)**: Ini adalah metode untuk menetapkan nilai **nrp** dengan nilai yang diberikan sebagai argumen **i**. Dalam kasus ini, metode ini mengatur nilai **nrp** menjadi nilai **i**.

Percobaan 3

Kelas **Siswa**: Ini adalah kelas yang mewakili entitas siswa.

* Variabel **nrp**: Ini adalah variabel instance yang mewakili Nomor Registrasi Siswa (NRP).
* Variabel **nama**: Ini adalah variabel instance yang mewakili nama siswa.
* Metode **setNrp(int i)**: Ini adalah metode untuk menetapkan nilai **nrp** dengan nilai yang diberikan sebagai argumen **i**.
* Metode **setNama(String i)**: Ini adalah metode untuk menetapkan nilai **nama** dengan nilai yang diberikan sebagai argumen **i**.

Kelas **Test**: Ini adalah kelas yang berisi metode **main** yang merupakan titik masuk utama program.

* Dalam metode **main**, sebuah objek dari kelas **Siswa** dibuat dengan nama **anak**.
* Kemudian, metode **setNrp(int i)** dipanggil untuk menetapkan nilai **nrp** menjadi **5**.
* Selanjutnya, metode **setNama(String i)** dipanggil untuk menetapkan nilai **nama** menjadi **"Budi"**.
* Nilai **nrp** dan **nama** kemudian dicetak menggunakan **System.out.println()**.

Percobaan 4.

class Puppy {

private int puppyAge;

public Puppy(String name){

// This constructor has one parameter, name.

System.out.println("Passed Name is :" + name );

}

public void setAge( int age ){

puppyAge = age;

}

public int getAge( ){

System.out.println("Puppy's age is :" + puppyAge );

return puppyAge;

}

public static void main(String []args){

Puppy myPuppy = new Puppy( "tommy" );

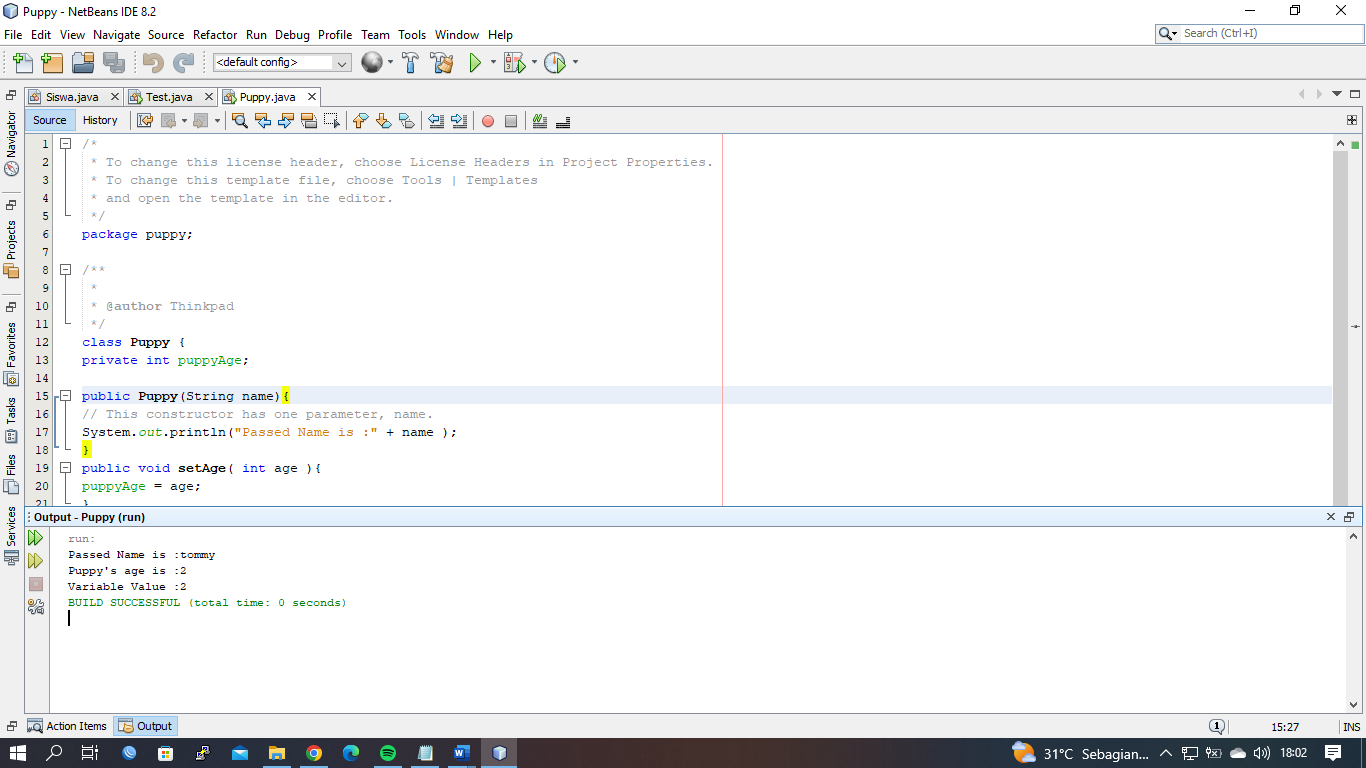
myPuppy.setAge( 2 );

myPuppy.getAge( );

System.out.println("Variable Value :" + myPuppy.puppyAge );

}

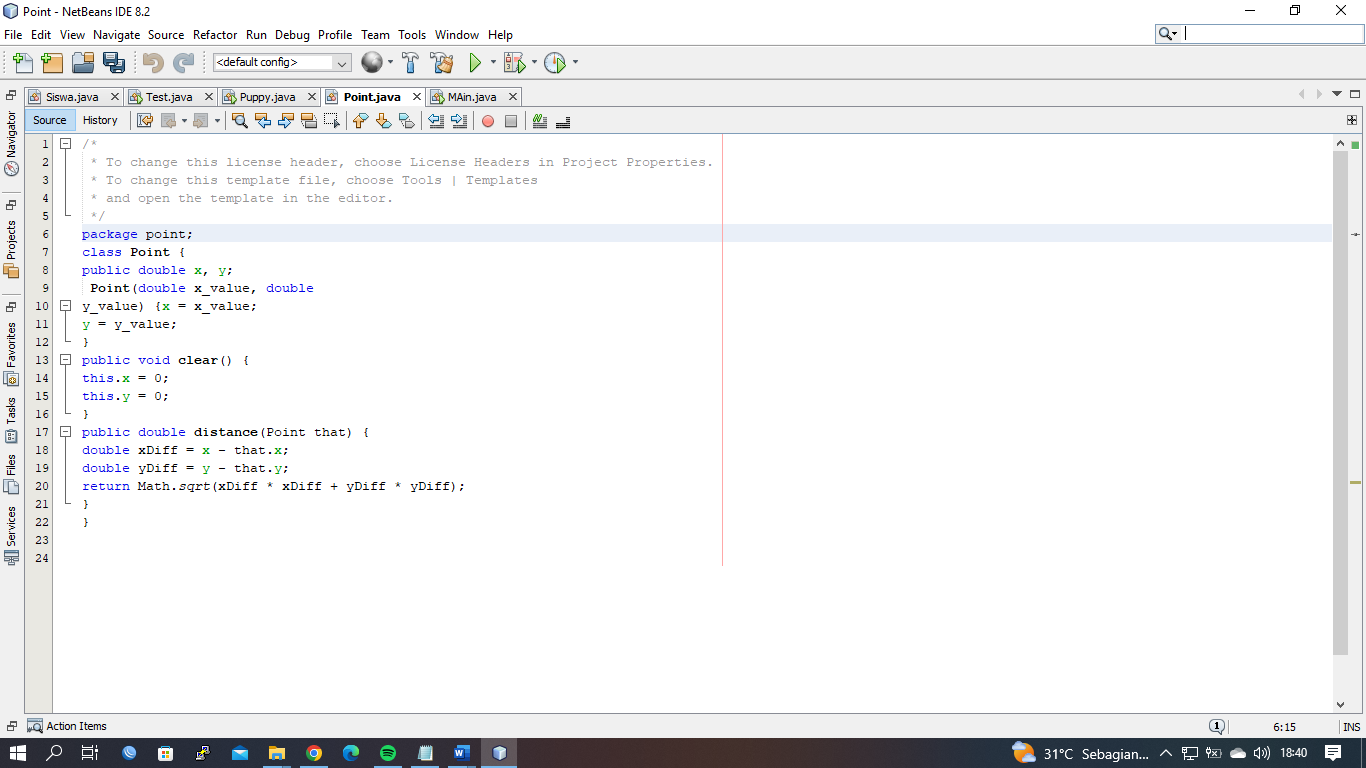
}

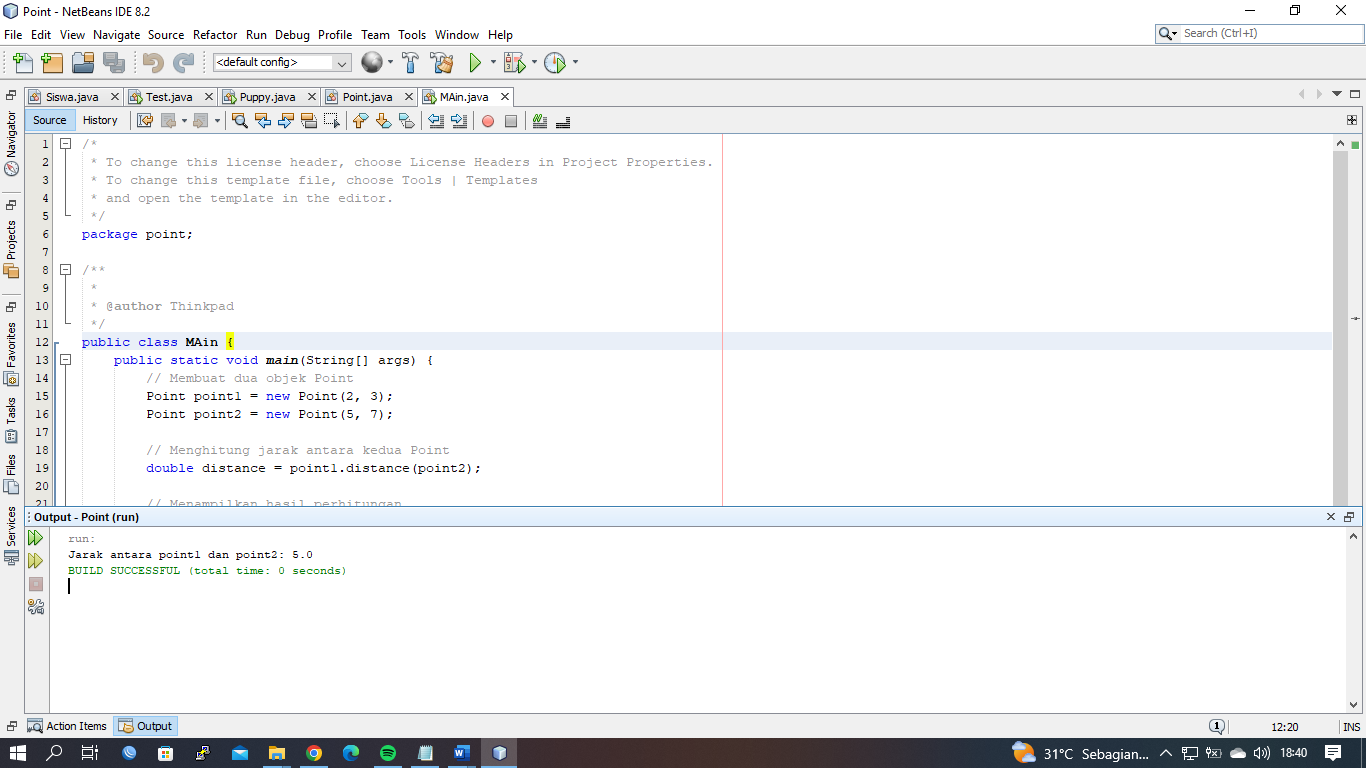


Kelas **Puppy**: Ini adalah kelas yang mewakili entitas anjing.

* Variabel **puppyAge**: Ini adalah variabel instance yang mewakili umur anjing.
* Konstruktor **Puppy(String name)**: Ini adalah konstruktor kelas yang memiliki satu parameter, yaitu **name**. Konstruktor ini dipanggil ketika objek **Puppy** dibuat, dan mencetak pesan dengan nama anjing yang diberikan.
* Metode **setAge(int age)**: Ini adalah metode untuk menetapkan umur anjing dengan nilai yang diberikan sebagai argumen **age**.
* Metode **getAge()**: Ini adalah metode untuk mendapatkan umur anjing. Metode ini mencetak umur anjing dan mengembalikan nilai umur tersebut.
* Metode **main(String[] args)**: Ini adalah metode utama program. Di dalamnya, sebuah objek **Puppy** dibuat dengan nama "tommy", kemudian umur anjing tersebut diatur menjadi 2 menggunakan metode **setAge(int age)**, dan umur anjing dicetak menggunakan metode **getAge()**. Selain itu, nilai variabel **puppyAge** juga dicetak langsung.

Percobaan 5.





public class MAin {

public static void main(String[] args) {

// Membuat dua objek Point

Point point1 = new Point(2, 3);

Point point2 = new Point(5, 7);

// Menghitung jarak antara kedua Point

double distance = point1.distance(point2);

// Menampilkan hasil perhitungan

System.out.println("Jarak antara point1 dan point2: " + distance);

}

}

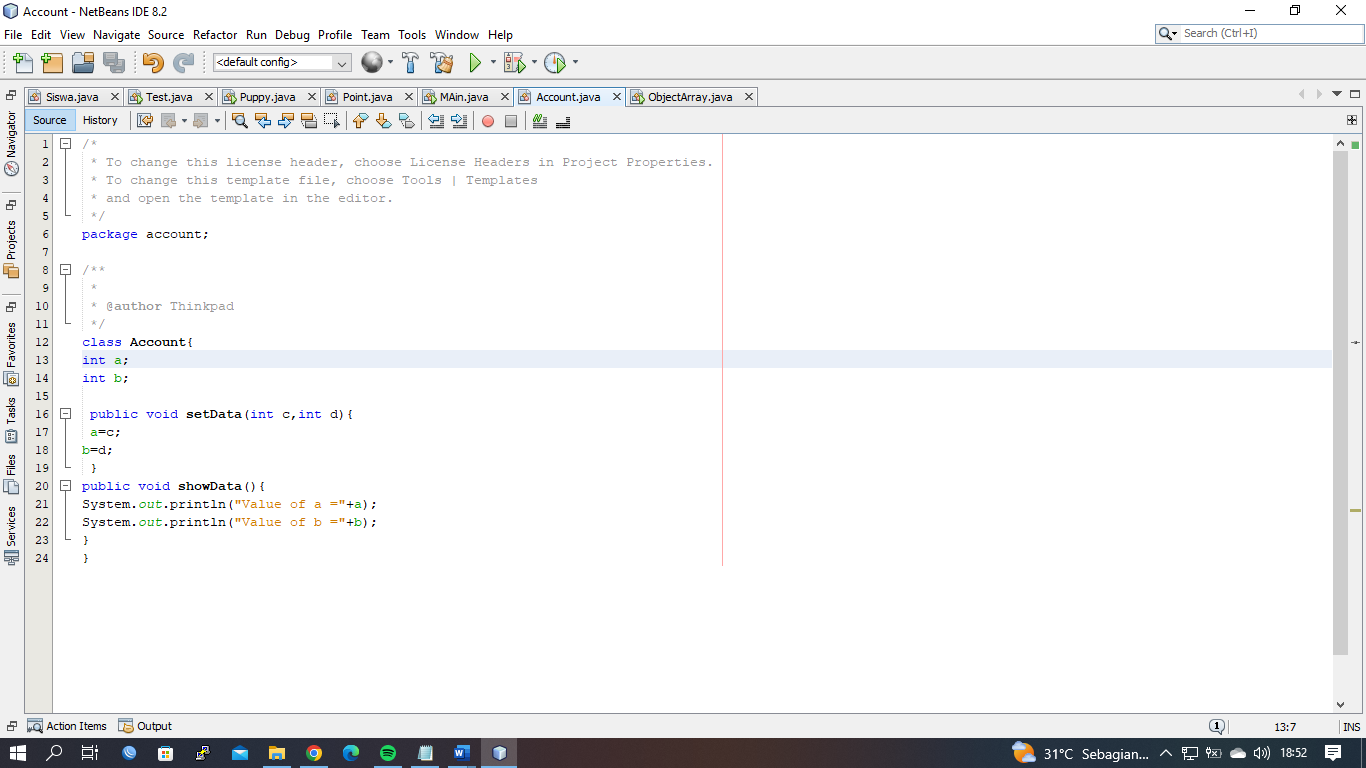
Kelas **Point**: Ini adalah kelas yang merepresentasikan titik dalam sistem koordinat kartesian.

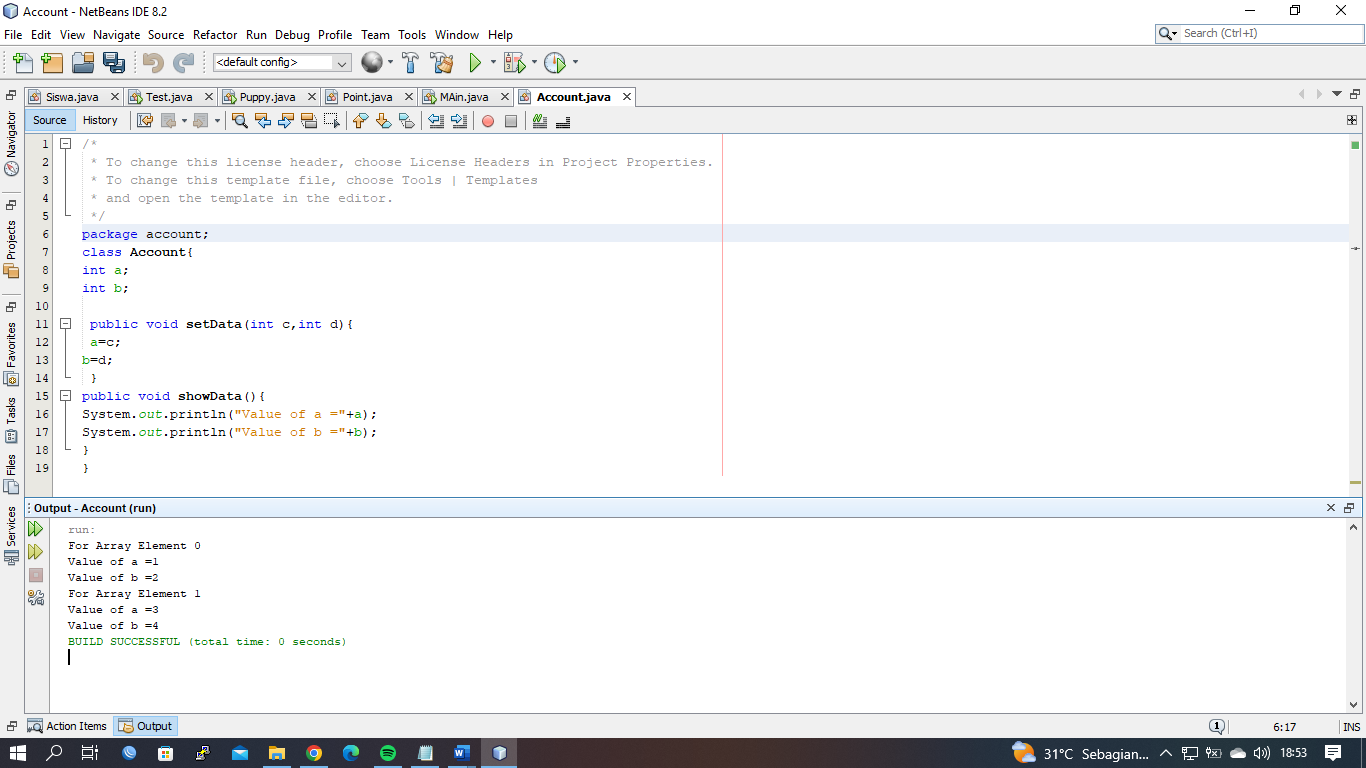
* Variabel **x** dan **y**: Ini adalah variabel instance yang menyimpan koordinat titik.
* Konstruktor **Point(double x\_value, double y\_value)**: Ini adalah konstruktor kelas yang menginisialisasi nilai **x** dan **y** saat objek **Point** dibuat.
* Metode **clear()**: Ini adalah metode yang digunakan untuk mengatur kedua nilai **x** dan **y** menjadi 0.
* Metode **distance(Point that)**: Ini adalah metode yang menghitung jarak antara titik saat ini dan titik yang diberikan sebagai argumen **that**. Metode ini menghitung jarak menggunakan rumus jarak Euclidean (jarak antara dua titik dalam ruang dua dimensi) dan mengembalikan nilai jarak tersebut.

Kelas **MAin**: Ini adalah kelas utama yang memiliki metode **main** sebagai titik masuk utama program.

* Metode **main(String[] args)**: Ini adalah metode yang dieksekusi pertama kali saat program dijalankan.
  + Membuat dua objek **Point**, yaitu **point1** dengan koordinat (2, 3) dan **point2** dengan koordinat (5, 7).
  + Menghitung jarak antara dua titik menggunakan metode **distance(Point that)** dari objek **point1** dengan **point2** sebagai argumen.
  + Menampilkan hasil perhitungan jarak ke layar menggunakan **System.out.println()**.

Percobaan 6





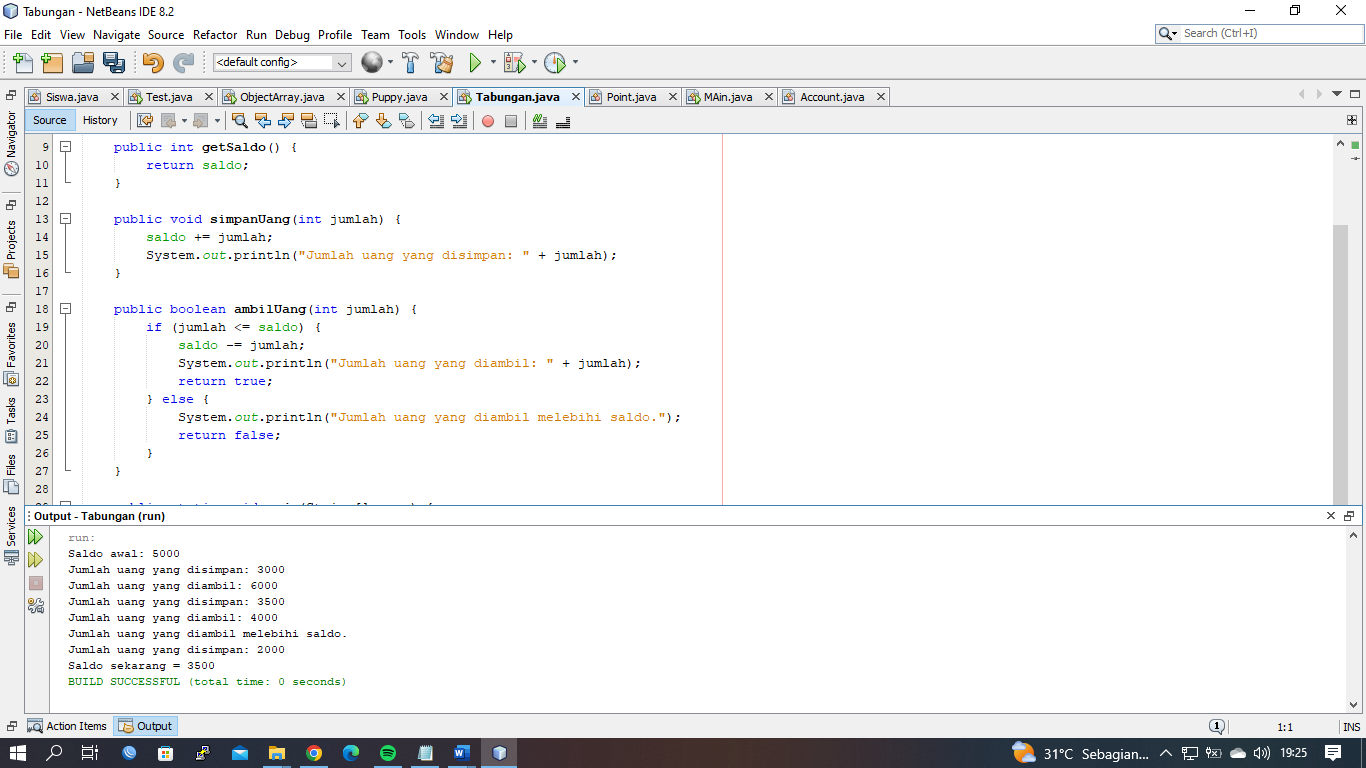
Kelas **Account**: Ini adalah kelas yang memiliki dua variabel instance **a** dan **b**, yang mungkin mewakili data akun atau informasi lainnya.

* Metode **setData(int c, int d)**: Ini adalah metode yang digunakan untuk mengatur nilai variabel instance **a** dan **b** dengan nilai yang diberikan sebagai argumen **c** dan **d**.
* Metode **showData()**: Ini adalah metode yang digunakan untuk menampilkan nilai dari variabel instance **a** dan **b** ke layar.

Kelas **ObjectArray**: Ini adalah kelas yang memiliki metode **main** sebagai titik masuk utama program.

* Metode **main(String args[])**: Ini adalah metode yang dieksekusi pertama kali saat program dijalankan.
* Membuat sebuah array **obj** yang terdiri dari dua elemen yang bertipe **Account**. Array ini akan menampung dua objek dari kelas **Account**.
* Membuat dua objek dari kelas **Account** menggunakan konstruktor default dan menyimpannya di dalam array **obj**.
* Mengatur data untuk masing-masing objek yang disimpan dalam array **obj**, yaitu objek pada indeks 0 dan 1, menggunakan metode **setData(int c, int d)**.
* Menampilkan data untuk masing-masing objek yang disimpan dalam array **obj** menggunakan metode **showData()**. Ini dilakukan dengan mengakses objek pada indeks 0 dan 1 dari array **obj**.

**E.LATIHAN**



package tabungan;

public class Tabungan {

private int saldo;

public Tabungan(int initsaldo) {

saldo = initsaldo;

}

public int getSaldo() {

return saldo;

}

public void simpanUang(int jumlah) {

saldo += jumlah;

System.out.println("Jumlah uang yang disimpan: " + jumlah);

}

public boolean ambilUang(int jumlah) {

if (jumlah <= saldo) {

saldo -= jumlah;

System.out.println("Jumlah uang yang diambil: " + jumlah);

return true;

} else {

System.out.println("Jumlah uang yang diambil melebihi saldo.");

return false;

}

}

public static void main(String[] args) {

Tabungan tabungan = new Tabungan(5000);

System.out.println("Saldo awal: " + tabungan.getSaldo());

tabungan.simpanUang(3000);

tabungan.ambilUang(6000);

tabungan.simpanUang(3500);

tabungan.ambilUang(4000);

tabungan.ambilUang(1600);

tabungan.simpanUang(2000);

System.out.println("Saldo sekarang = " + tabungan.getSaldo());

}

}

Kelas **Tabungan**: Ini adalah kelas yang memiliki atribut **saldo** yang hanya dapat diakses secara internal.

* Konstruktor **Tabungan(int initsaldo)**: Digunakan untuk menginisialisasi saldo awal.
* Metode **getSaldo()**: Digunakan untuk mendapatkan nilai saldo.
* Metode **simpanUang(int jumlah)**: Digunakan untuk menambahkan jumlah uang ke saldo dan mencetak pesan berapa jumlah uang yang disimpan.
* Metode **ambilUang(int jumlah)**: Digunakan untuk mengambil uang dari saldo, memeriksa apakah saldo mencukupi, dan mencetak pesan berapa jumlah uang yang diambil atau pesan bahwa jumlah uang yang diambil melebihi saldo.
* Metode **main(String[] args)**: Digunakan untuk menguji kelas **Tabungan** dengan skenario yang diberikan. Membuat objek **Tabungan** dengan saldo awal 5000, melakukan operasi penyimpanan dan pengambilan uang, dan menampilkan saldo akhir

Latihan 2.

public class Tabungan {

private int saldo;

// Konstruktor untuk menginisialisasi saldo awal

public Tabungan(int initsaldo) {

saldo = initsaldo;

}

// Metode untuk mendapatkan saldo

public int getSaldo() {

return saldo;

}

// Metode untuk menyimpan uang ke saldo

public void simpanUang(int jumlah) {

saldo += jumlah;

System.out.println("Jumlah uang yang disimpan: " + jumlah);

}

// Metode untuk mengambil uang dari saldo

public boolean ambilUang(int jumlah) {

if (jumlah <= saldo) {

saldo -= jumlah;

System.out.println("Jumlah uang yang diambil: " + jumlah);

return true;

} else {

System.out.println("Jumlah uang yang diambil melebihi saldo.");

return false;

}

}

// Metode untuk mentransfer sejumlah uang ke tabungan lain

public boolean transfer(Tabungan tab, int jumlah) {

if (ambilUang(jumlah)) {

tab.simpanUang(jumlah);

return true;

} else {

System.out.println("Transfer gagal: Saldo tidak mencukupi.");

return false;

}

}

public static void main(String[] args) {

// Membuat objek tabungan

Tabungan tabungan1 = new Tabungan(5000);

Tabungan tabungan2 = new Tabungan(0); // Saldo awal tabungan kedua 0

// Menampilkan saldo awal tabungan

System.out.println("Saldo awal tabungan 1: " + tabungan1.getSaldo());

System.out.println("Saldo awal tabungan 2: " + tabungan2.getSaldo());

// Menyimpan uang, mengambil uang, dan melakukan transfer

tabungan1.simpanUang(3000);

tabungan1.ambilUang(6000);

tabungan1.simpanUang(3500);

tabungan1.ambilUang(4000);

tabungan1.ambilUang(1600);

tabungan1.simpanUang(2000);

// Menampilkan saldo tabungan setelah operasi

System.out.println("Saldo sekarang tabungan 1: " + tabungan1.getSaldo());

// Melakukan transfer antar tabungan

tabungan1.transfer(tabungan2, 1500);

tabungan1.transfer(tabungan2, 5000);

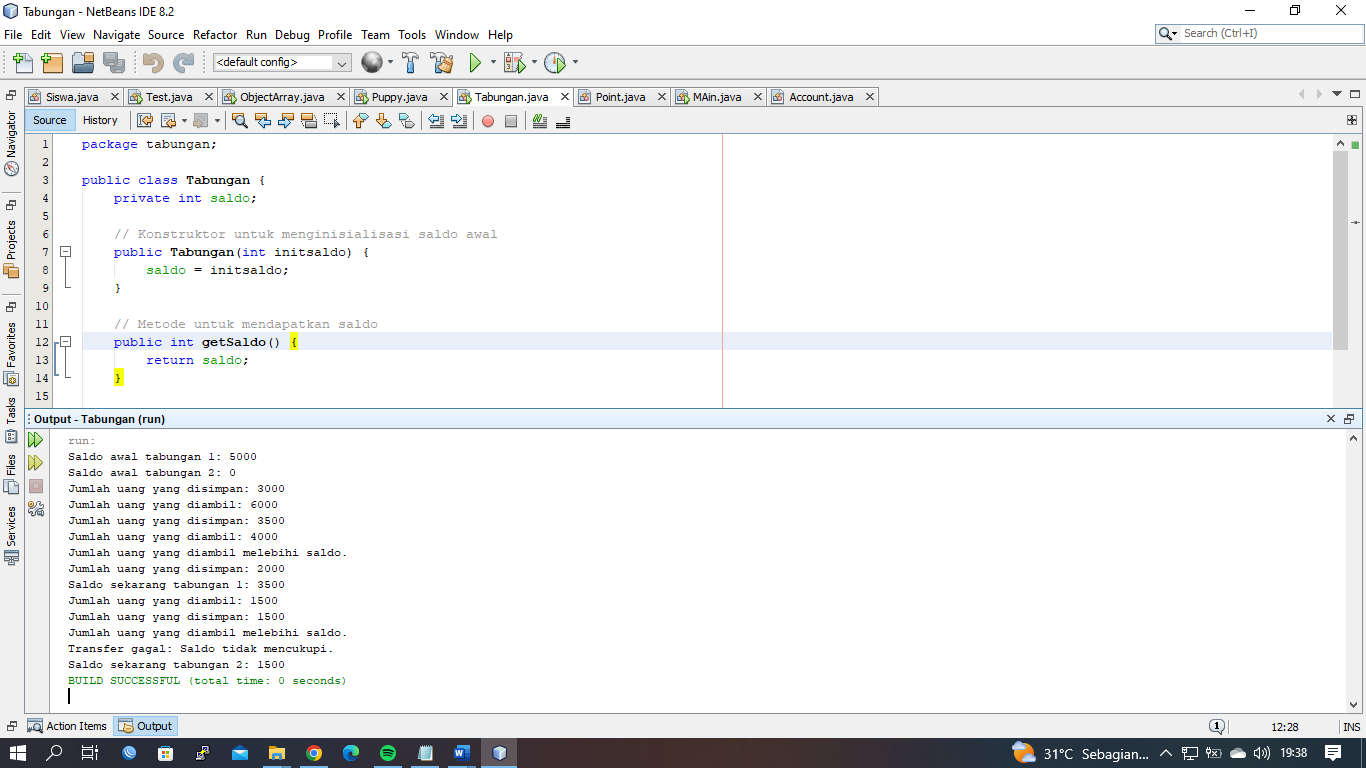
// Menampilkan saldo tabungan kedua setelah transfer

System.out.println("Saldo sekarang tabungan 2: " + tabungan2.getSaldo());

}

}

Hasilnya adalah :



Kelas **Tabungan**: Ini adalah kelas yang mengelola saldo tabungan.

* Atribut **saldo**: Menyimpan saldo tabungan.
* Konstruktor **Tabungan(int initsaldo)**: Digunakan untuk menginisialisasi saldo awal tabungan.
* Metode **getSaldo()**: Mengembalikan nilai saldo tabungan.
* Metode **simpanUang(int jumlah)**: Menambahkan jumlah uang ke saldo tabungan.
* Metode **ambilUang(int jumlah)**: Mengurangi jumlah uang dari saldo tabungan jika saldo mencukupi.
* Metode **transfer(Tabungan tab, int jumlah)**: Mentransfer sejumlah uang dari tabungan saat ini ke tabungan lain jika saldo mencukupi.