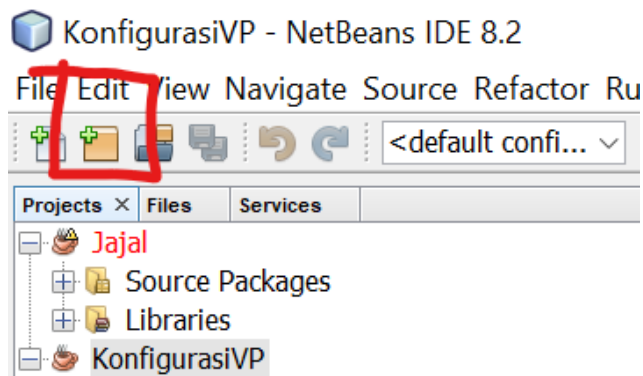


Nama : Imanuddin
No absen : 08
Kelas : TI-1B
NIM : 2131740036

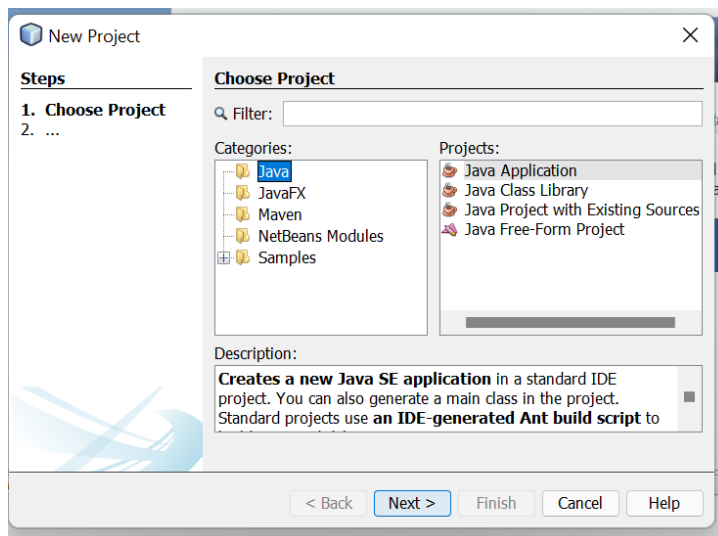
PRAKTIKUM CLASS DIAGRAM LINGKARAN DAN PERSEGI PANJANG

A. Membuat project baru

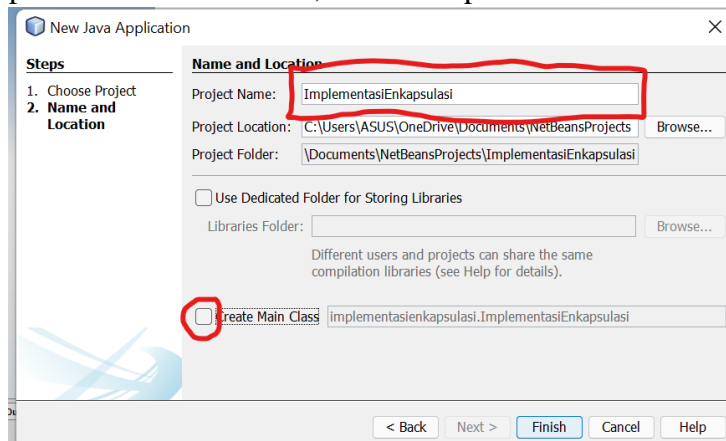
1. Jalankan Netbeans, kemudian pilih **New Project**.



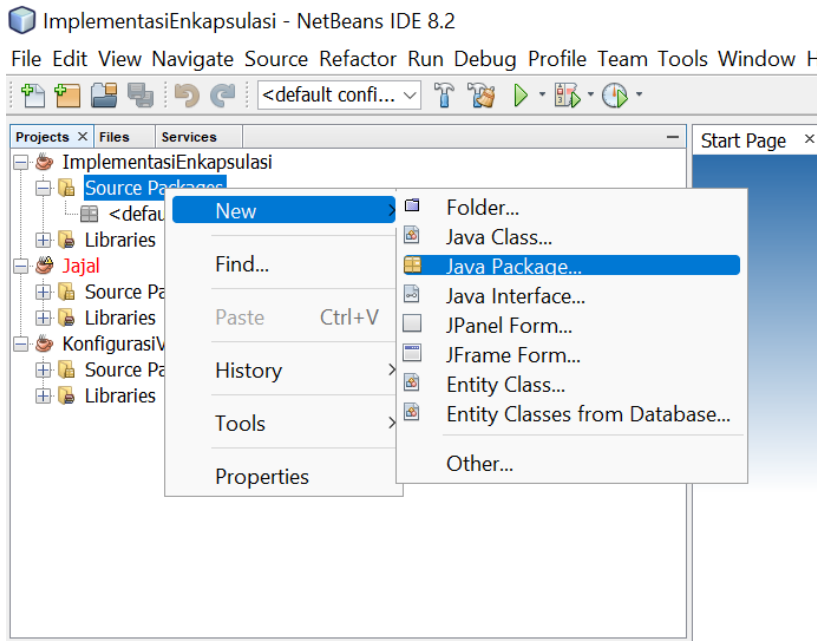
2. Pada Categories java, kemudian pada Projects pilih Java Application, setelah itu pilih **next**.



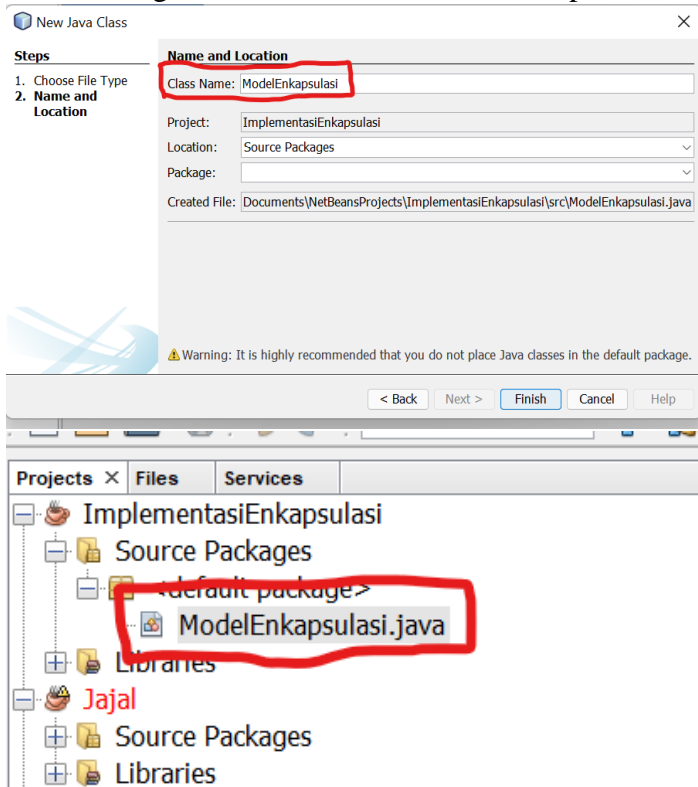
3. Pada Project Name beri nama: **ImplementasiEnkapsulasi** dan hilangkan checklist pada Craete Main Class, kemudian pilih Finish.



- Setelah itu pada Project ImplementasiEnkapsulasi pilih Source Package kemudian **klik kanan** pilih **New -> Java Package** seperti pada tampilan berikut.

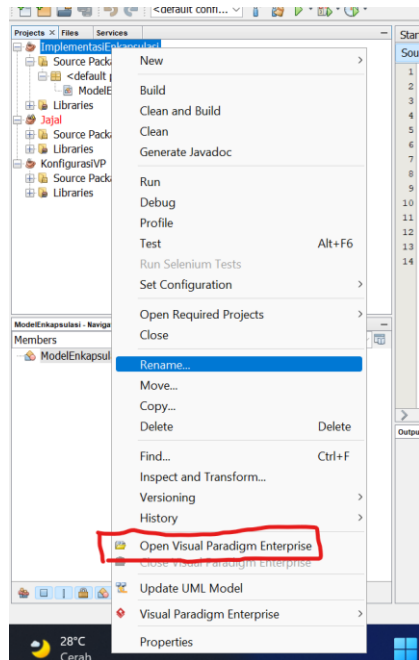


- Pada Package Name beri nama: ModelEnkapsulasi, setelah itu klik **Finish**.

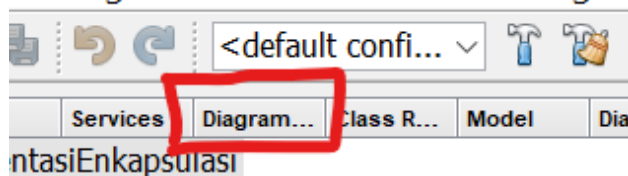


B. Memanggil Plug-in Visual Paradigm

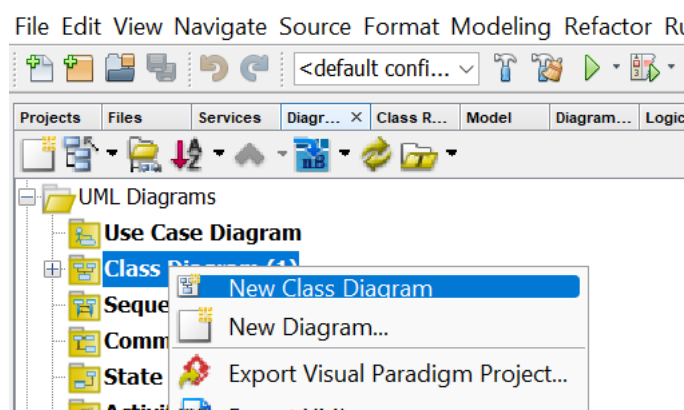
- Pada Project ImplementasiEnkapsulasi klik kanan, kemudian pilih Open Visual Paradigm Enterprise dan tunggu beberapa saat.



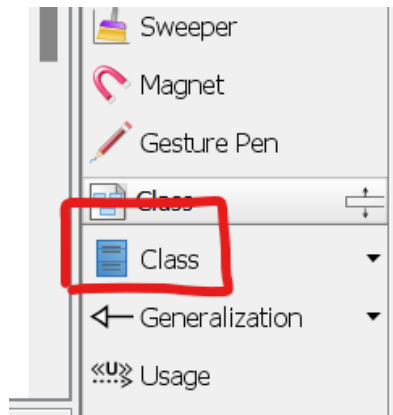
2. Setelah tampil panel visual paradigm, pilih Diagram pada panel.



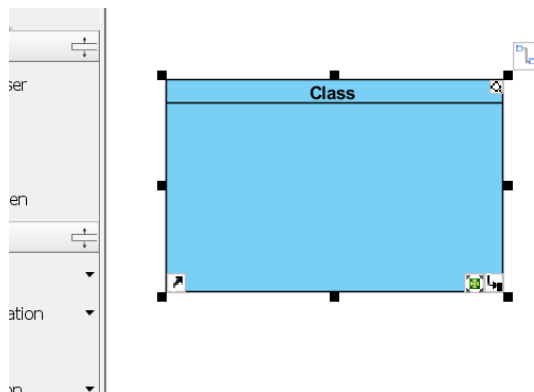
3. Pada UML Diagram pilih Class Diagram, kemudian klik kanan pada Class Diagram dan pilih New Class Diagram.



4. Klik pad panel Class.



5. Setelah di klik kemudian akan muncul UML Diagram Class seperti berikut ini.



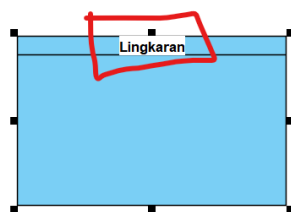
C. Memodelkan Class Diagram menggunakan Visual Paradigm

1. Membuat Class Diagram dengan spesifikasi sebagai berikut.

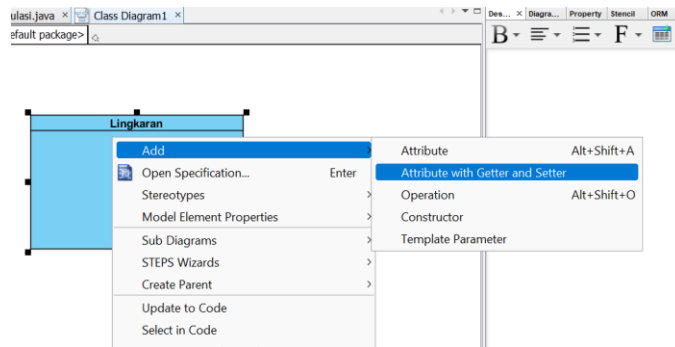
- Nama Class: Lingkaran

Nama Attribute	Access Modifier	Tipe Data	Value
phi	private	double	3.14
r	private	double	-
Nama Method/oparation	Acces Modifier	Type Method	Value
setPhi	public	void	phi
getPhi	public	double	phi
getR	public	void	r
getR	public	double	R
keliling	public	double	$2 * \text{phi} * r$
luas	public	double	$\text{phi} * r * r$

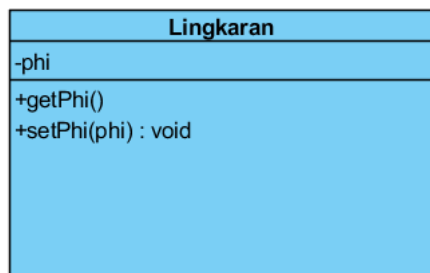
2. Langkah pertama double klik pada nama class kemudian ganti dengan nama Lingkaran.



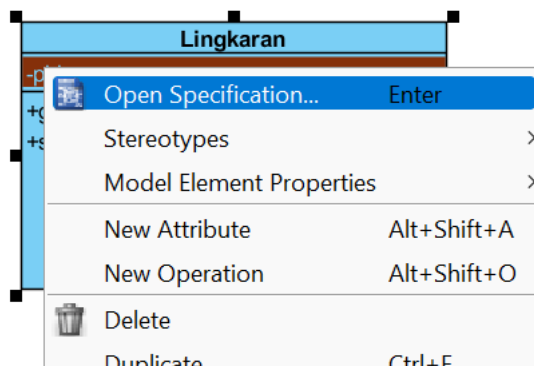
3. Klik kanan pada Class Diagram kemudian pilih add -> attribute with Getter and Setter.



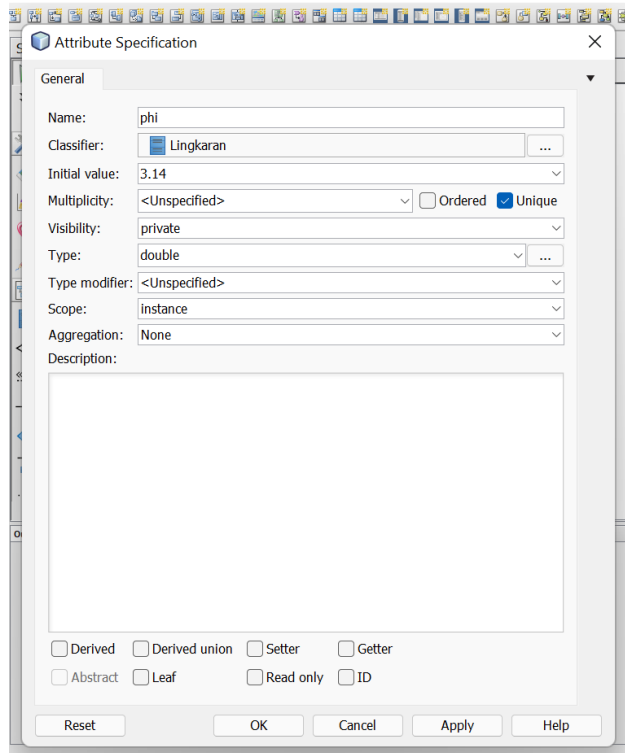
4. Tambahkan attribute sesuai dengan spesifikasi Class Diagram yang diinginkan (lihat pada table).



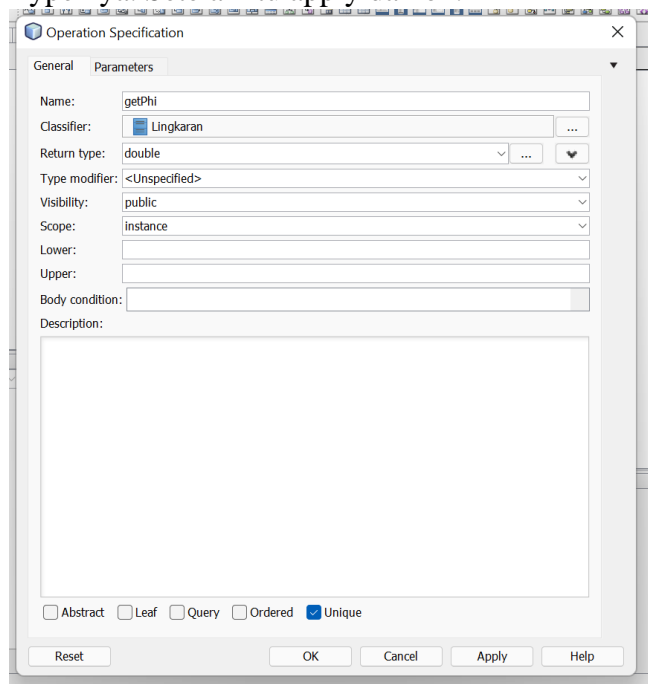
5. Pada attribute phi klik kanan dan pilih Open Specification.



6. Pada attribute specification tambahkan Initial value menjadi = 3.14 dan pada Type pilih type data double, setelah itu klik **apply** dan **ok**.



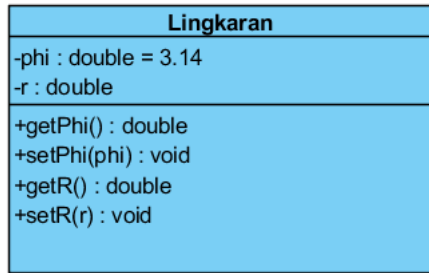
7. Ulangi cara yang sama pada method getPhi dan isikan type data double pada Return Type nya. Setelah itu apply dan ok.



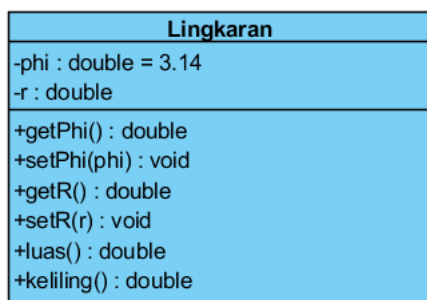
8. Hasilnya.

Lingkaran
-phi : double = 3.14
+getPhi() : double +setPhi(phi) : void

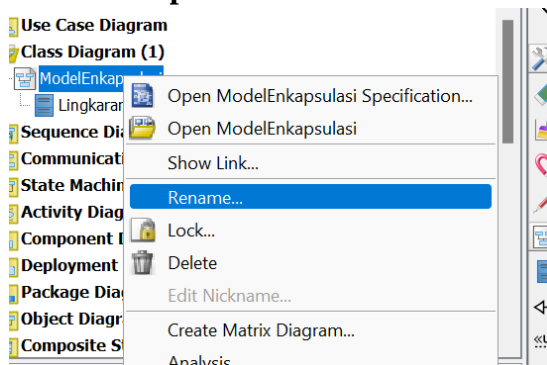
9. Kemudian tambahkan attribute berikutnya sesuai dengan yang ada di properties table.
10. Hasilnya setelah ditambahkan semua attribute pada Class Diagram.



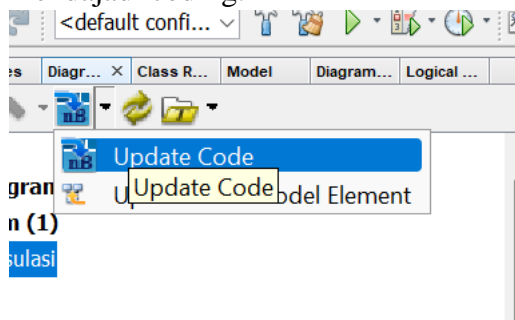
11. Tambahkan **operation** untuk menambahkan operation/method keliling dan luas, pastikan operation/ method mempunyai **return type** double.
12. Hasil setelah dimasukkan semua method dan attribute.



13. Setelah semua method/operasi dan attribute sudah dimasukkan, langkah berikutnya adalah kita ganti nama package yang ada di UML Diagram menjadi **ModelEnkapsulasi**.

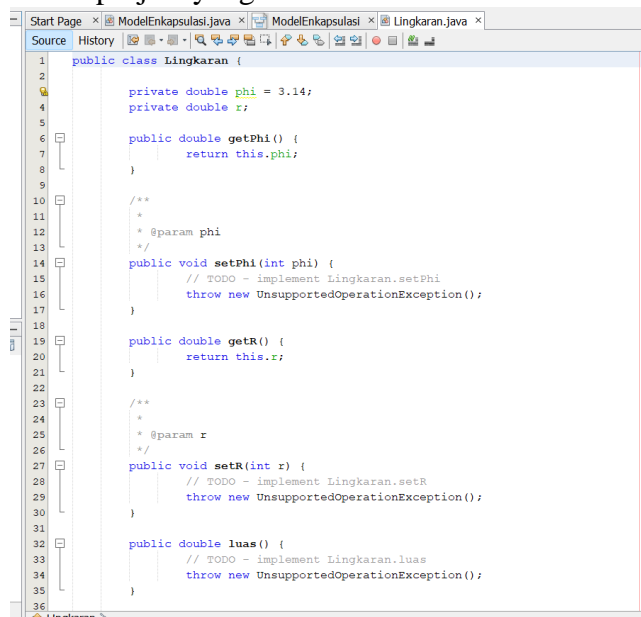


14. Setelah itu pilih update code untuk melakukan generate dari visual paradigm menjadi coding.



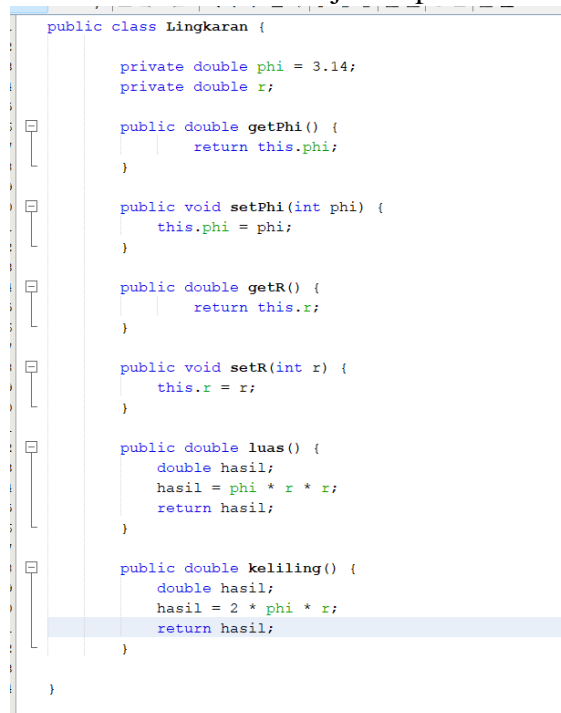
15. Setelah dilakukan update code, maka secara otomatis kita bisa melihat hasil generate code dari Class Diagram yang sudah kita buat.

16. Berikut adalah hasil dari generate Code yang secara otomatis akan ditampilkan di dalam project yang kita buat.



```
1 public class Lingkaran {
2
3     private double phi = 3.14;
4     private double r;
5
6     public double getPhi() {
7         return this.phi;
8     }
9
10
11     /**
12      *
13      * @param phi
14      */
15     public void setPhi(int phi) {
16         // TODO - implement Lingkaran.setPhi
17         throw new UnsupportedOperationException();
18     }
19
20     public double getR() {
21         return this.r;
22     }
23
24     /**
25      *
26      * @param r
27      */
28     public void setR(int r) {
29         // TODO - implement Lingkaran.setR
30         throw new UnsupportedOperationException();
31     }
32
33     public double luas() {
34         // TODO - implement Lingkaran.luas
35         throw new UnsupportedOperationException();
36     }
37 }
```

17. Perbaiki kode diatas menjadi seperti ini.

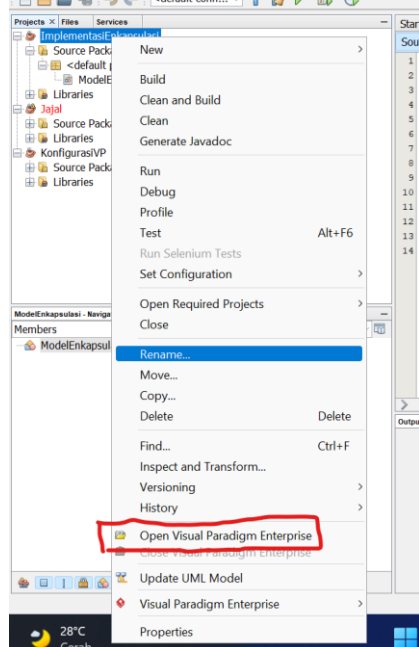


```
1 public class Lingkaran {
2
3     private double phi = 3.14;
4     private double r;
5
6     public double getPhi() {
7         return this.phi;
8     }
9
10     public void setPhi(int phi) {
11         this.phi = phi;
12     }
13
14     public double getR() {
15         return this.r;
16     }
17
18     public void setR(int r) {
19         this.r = r;
20     }
21
22     public double luas() {
23         double hasil;
24         hasil = phi * r * r;
25         return hasil;
26     }
27
28     public double keliling() {
29         double hasil;
30         hasil = 2 * phi * r;
31         return hasil;
32     }
33 }
```

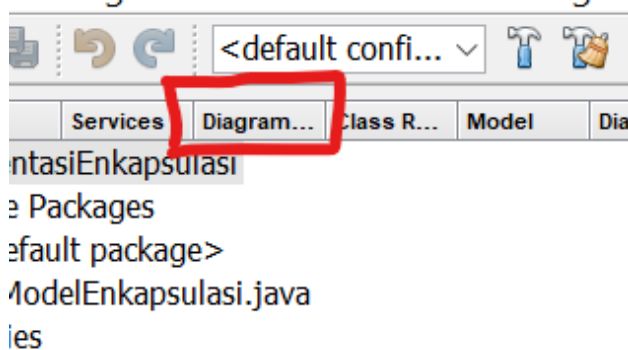

PRAKTIKUM PERSEGI PANJANG

A. Memanggil Plug-in Visual Paradigm

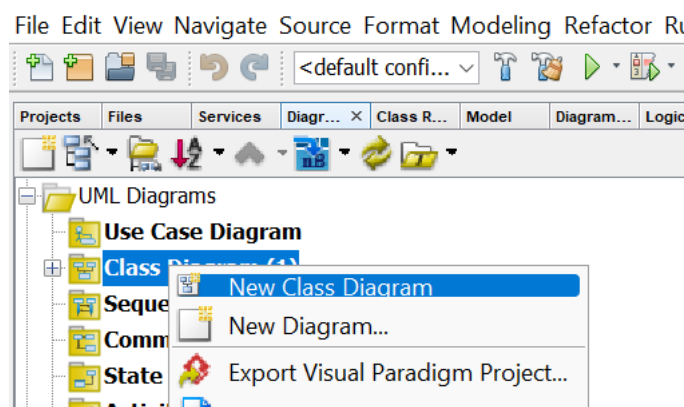
1. Pada Project ImplementasiEnkapsulasi klik kanan, kemudian pilih Open Visual Paradigm Enterprise dan tunggu beberapa saat.



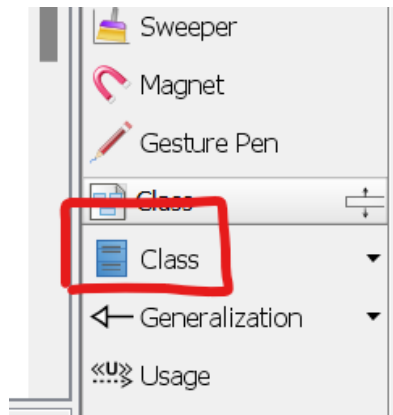
2. Setelah tampil panel visual paradigm, pilih Diagram pada panel.



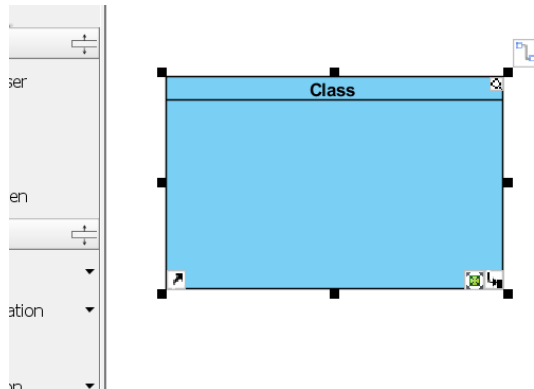
3. Pada UML Diagram pilih Class Diagram, kemudian klik kanan pada Class Diagram dan pilih New Class Diagram.



4. Klik pad panel Class.



5. Setelah di klik kemudian akan muncul UML Diagram Class seperti berikut ini.



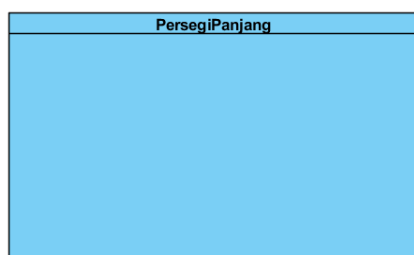
B. Memodelkan Class Diagram menggunakan Visual Paradigm

1. Membuat Class Diagram dengan spesifikasi sebagai berikut.

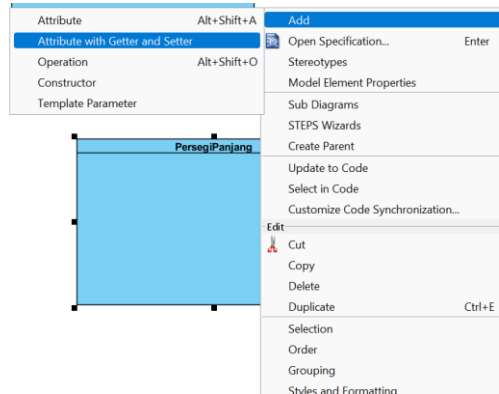
- Nama Class: PersegiPanjang

Nama Attribute	Access Modifier	Type Data	Value
P	private	double	-
l	private	double	-
Nama Method/oparation	Acces Modifier	Type Method	Value
setP	public	void	p
getP	public	double	p
getL	public	void	l
getL	public	double	l
keliling	public	double	$2 * \text{phi} * r$
luas	public	double	$\text{phi} * r * r$

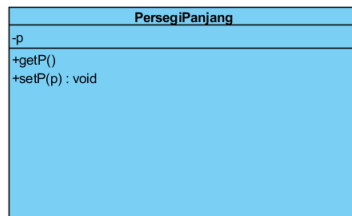
2. Langkah pertama double klik pada nama class kemudian ganti dengan nama PersegiPanjang.



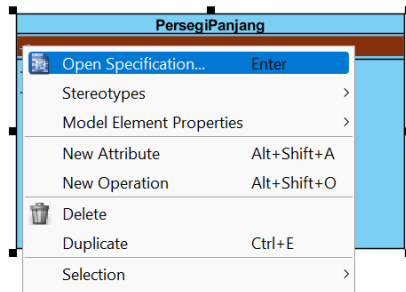
3. Klik kanan pada Class Diagram kemudian pilih add -> attribute with Getter and Setter.



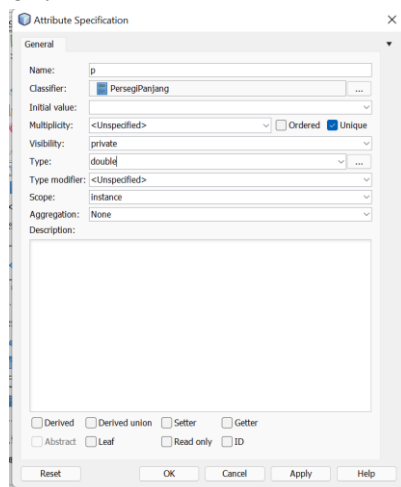
4. Tambahkan attribute sesuai dengan spesifikasi Class Diagram yang diinginkan (lihat pada table).



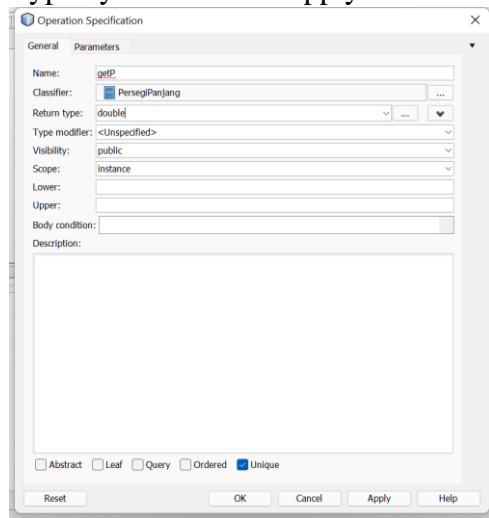
5. Pada attribute p klik kanan dan pilih Open Spesification.



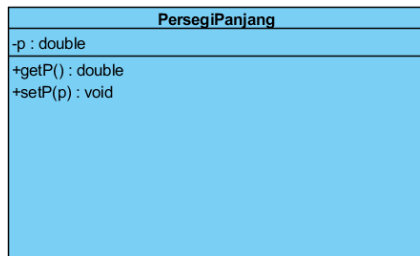
6. Pada attribute specification Type pilih type data double, setelah itu klik **apply** dan **ok**.



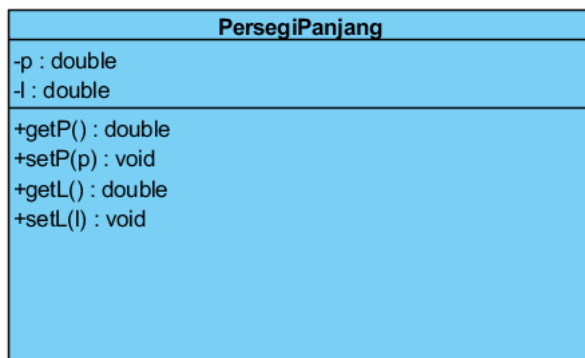
7. Ulangi cara yang sama pada method getP dan isikan type data double pada Return Type nya. Setelah itu apply dan ok.



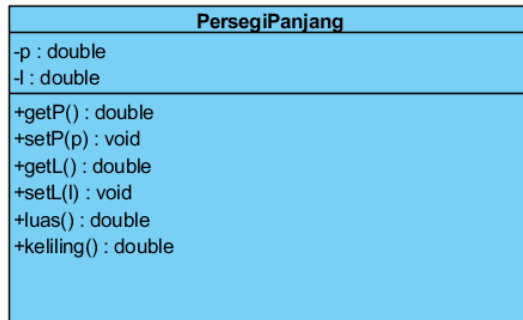
8. Hasilnya.



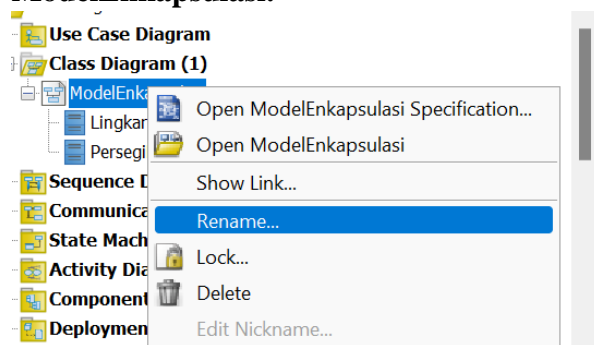
9. Kemudian tambahkan attribute berikutnya sesuai dengan yang ada di properties table.
10. Hasilnya setelah ditambahkan semua attribute pada Class Diagram.



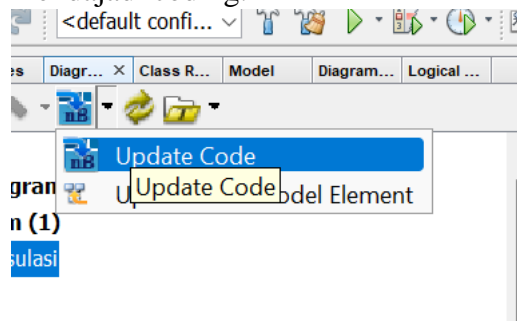
11. Tambahkan **operation** untuk menambahkan operation/method keliling dan luas, pastikan operation/ method mempunyai **return type** double.
12. Hasil setelah dimasukkan semua method dan attribute.



13. Setelah semua method/operasi dan attribute sudah dimasukkan, langkah berikutnya adalah kita ganti nama package yang ada di UML Diagram menjadi **ModelEnkapsulasi**.



14. Setelah itu pilih update code untuk melakukan generate dari visual paradigm mendjadi coding.



15. Setelah dilakukan update code, maka secara otomatis kita bisa melihat hasil generate code dari Class Diagram yang sudah kita buat.
16. Berikut adalah hasil dari generate Code yang secara otomatis akan ditampilkan di dalam project yang kita buat.

```

1 private double l;
2
3 public double getP() {
4     return this.p;
5 }
6
7 /**
8  * @param p
9  */
10 public void setP(int p) {
11     // TODO - implement PersegiPanjang.setP
12     throw new UnsupportedOperationException();
13 }
14
15 public double getL() {
16     return this.l;
17 }
18
19 /**
20  * @param l
21  */
22 public void setL(int l) {
23     // TODO - implement PersegiPanjang.setL
24     throw new UnsupportedOperationException();
25 }
26
27 public double luas() {
28     // TODO - implement PersegiPanjang.luas
29     throw new UnsupportedOperationException();
30 }
31
32 public double keliling() {
33     // TODO - implement PersegiPanjang.keliling
34     throw new UnsupportedOperationException();
35 }

```

PersegiPanjang

17. Perbaiki kode diatas menjadi seperti ini.

```

1 public class PersegiPanjang {
2
3     private double p;
4     private double l;
5
6     public double getP() {
7         return this.p;
8     }
9
10    public void setP(int p) {
11        this.p = p;
12    }
13
14    public double getL() {
15        return this.l;
16    }
17
18    public void setL(int l) {
19        this.l = l;
20    }
21
22    public double luas() {
23        double hasil;
24        hasil = p * l;
25        return hasil;
26    }
27
28    public double keliling() {
29        double hasil;
30        hasil = 2 * (p + l);
31        return hasil;
32    }
33 }

```