Percobaan

1. Percobaan 1

-Class A

package percobaan1;

/\*\*

\*

\* @author asus

\*/

public class ClassA {

public int x;

public int y;

public void getNilai(){

System.out.println("nilai x:"+ x);

System.out.println("nilai y:"+ y);

}

}

-Class B

package percobaan1;

/\*\*

\*

\* @author asus

\*/

public class ClassB {

public int z;

public void getNilaiZ(){

System.out.println("nilai Z:"+ z);

}

public void getJumlah(){

System.out.println("jumlah:"+ (x+y+z));

}

}

-Main Percobaan 1

package percobaan1;

/\*\*

\*

\* @author asus

\*/

public class Percobaan1 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

ClassB hitung = new ClassB();

hitung.x=20;

hitung.y=30;

hitung.z=5;

hitung.getNilai();

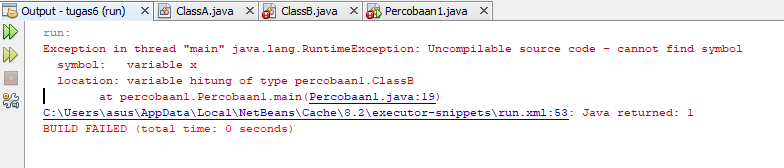
hitung.getNilaiZ();

hitung.getJumlah();

}

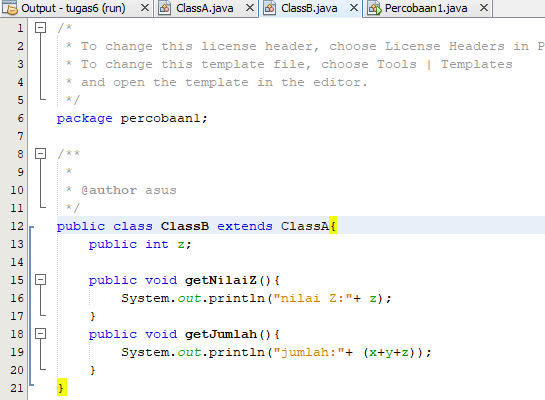
}

-Hasilnya:

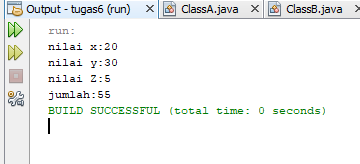


-Pertanyaan 1

1 cara agar tidak terjadi error adalah dengan menambahkan extend di Class B dengan Class A



Hasilnya setelah di run



2 Penyebab pada percobaan 1 adalah tidak adanya mengembalikan nilai x dan y di Class B sehingga perlu extend untuk dapat menghubungkan dengan Class A karena nilai x dan y berada didalam Class A

1. Percobaan 2

-Class A

package percobaan2;

/\*\*

\*

\* @author asus

\*/

public class ClassA {

public int x;

public int y;

public void setX(int x) {

this.x = x;

}

public void setY(int y) {

this.y = y;

}

public void getNilai(){

System.out.println("nilai x:"+ x);

System.out.println("nilai y:"+ y);

}

}

-Classs B

package percobaan2;

/\*\*

\*

\* @author asus

\*/

public class ClassB {

public int z;

public void setZ(int z) {

this.z = z;

}

public void getNilaiZ(){

System.out.println("nilai Z:"+ z);

}

public void getJumlah(){

System.out.println("jumlah:"+ (x+y+z));

}

}

-Percobaan 2

package percobaan2;

/\*\*

\*

\* @author asus

\*/

public class Percobaan2 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

ClassB hitung = new ClassB();

hitung.setX(20);

hitung.setY(30);

hitung.setZ(5);

hitung.getNilai();

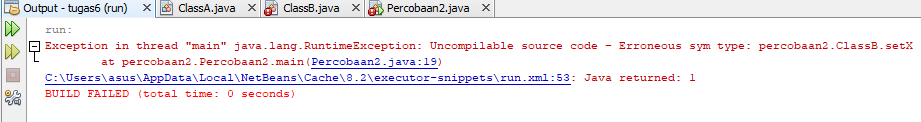
hitung.getNilaiZ();

hitung.getJumlah();

}

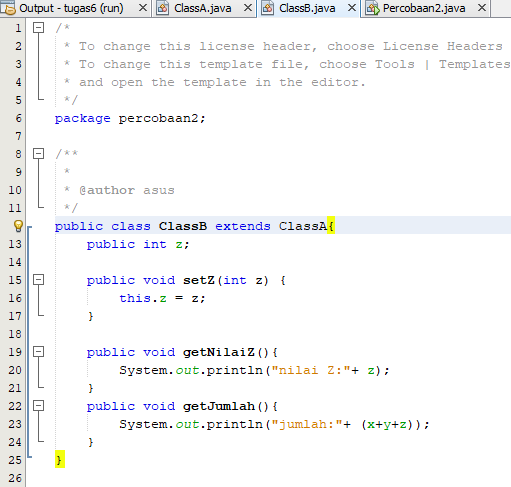
}

-Hasilnya

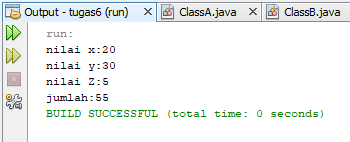


-Pertanyaan

1 cara agar tidak terjadi error adalah dengan menambahkan extend di Class B dengan Class A



Hasilnya



2 Penyebab pada percobaan 1 adalah tidak adanya mengembalikan nilai x dan y di Class B sehingga perlu extend untuk dapat menghubungkan dengan Class A karena nilai x dan y berada didalam Class A

1. Percobaan 3

-Class Bangun

package percobaan3;

/\*\*

\*

\* @author asus

\*/

public class Bangun {

protected double phi;

protected int r;

}

-Class Tabung

package percobaan3;

/\*\*

\*

\* @author asus

\*/

public class Tabung extends Bangun{

protected int t;

public void setSuperPhi(double phi){

super.phi=phi;

}

public void setSuperR(int r){

super.r=r;

}

public void setT(int t){

this.t=t;

}

public void volume(){

System.out.println("Volume Tabung adalah: "+(super.phi\*super.r\*super.r\*this.t));

}

}

-Main Percobaan 3

package percobaan3;

/\*\*

\*

\* @author asus

\*/

public class Percobaan3 {

public static void main(String[] args) {

Tabung tabung = new Tabung();

tabung.setSuperPhi(3.14);

tabung.setSuperR(10);

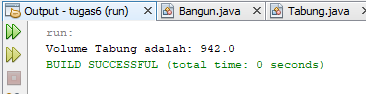
tabung.setT(3);

tabung.volume();

}

}

Hasilnya:



Pertanyaan:

1 fungsi super di class tabung, untuk super.phi berfungsi untuk memanggil element phi yang diinputkan di class Bangun atau juga class parentnya dan untuk super.r untuk memamnggil element r yang diinputkan di class Bangun atau juga class parentnya.

2 fungsi super dan this di class tabung pada method void volume, yaitu untuk super berfungsi memanggil element yang diinputkan di class Bangun atau juga class parentnya, dan untuk fungsi this berfungsi memanggil element yang berada di dalam class nya sendiri yaitu class Tabung atau juga class Childnya.

3 karena dalam class Tabung mempunyai hubungan dengan class Bangun yang merupakan class parentnya dengan kode program extend ke class Bangun.

1. Percobaan 4

-Class A

package percobaan4;

/\*\*

\*

\* @author asus

\*/

public class ClassA {

ClassA(){

System.out.println("konstruktor A dijalankan");

}

}

-Class B

package percobaan4;

/\*\*

\*

\* @author asus

\*/

public class ClassB extends ClassA{

ClassB() {

System.out.println("konstruktor B dijalankan");

}

}

-Class C

package percobaan4;

/\*\*

\*

\* @author asus

\*/

public class ClassC extends ClassB{

ClassC() {

System.out.println("konstruktor C dijalankan");

}

}

-Percobaan 4

package percobaan4;

/\*\*

\*

\* @author asus

\*/

public class Percobaan4 {

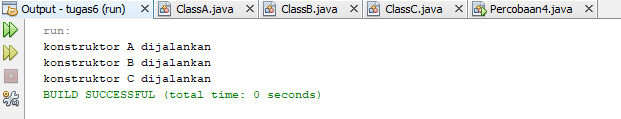
public static void main(String[] args) {

ClassC test = new ClassC();

}

}

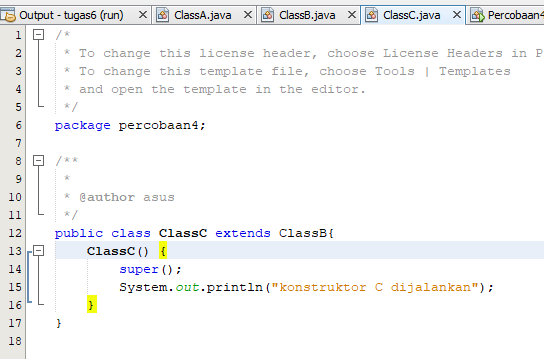
Hasilnya:



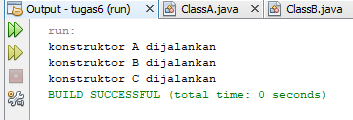
Pertanyaan

1 didalam percobaan 4 terdapat superclass dan subclass yaitu, class A merupakan superclass dari class B yang merupakan subclass nya class A karena dalam class B menghubungkan ke class A dengan mengguanakan extends dan class B merupakan superclass dari class C yang merupakan subclass nya class B karena dalam class C menghubungkan ke class B dengan menggunakan extends.

2 ubahlah kode program didalam class c

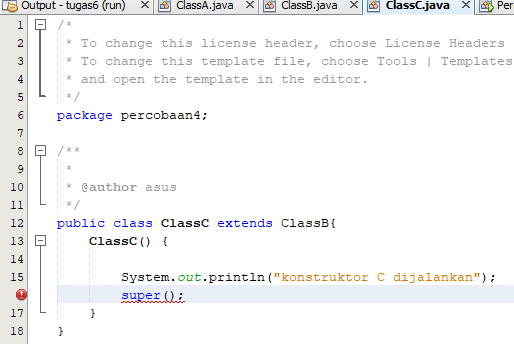


Hasilnya :



Sama saja dengan sebelumnya karena class super() tidak terhubung ke class manapun dan akan merujuk pada classnya sendiri

3 ubah kode program class c



Mengapa bisa begitu, karena untuk memanggil sebuah constructor super harus dalam keadaan posisi di awal kode program dalam class C

Urutan proses jalannya konstruktor saat objek test dibuat yaitu:

Hasil untuk percobaan urut dari parent class yang utama yaitu class A kemudian class B lalu class C yang terakhir hal ini karena yang dioutputkan dimulai dari class yang tidak punya extends lalu dilanjutkan lagi bawahnya dan seterusnya.

4 fungsi super() pada class C yaitu berfungsi untuk merujuk ke classnya sendiri karena tidak di inputkan akan merujuk ke class yang mana

1. Tugas

* Class Pegawai

package tugas;

/\*\*

\*

\* @author asus

\*/

public class Pegawai{

public String nip;

public String nama;

public String alamat;

public int gaji;

public Pegawai(String nip, String nama, String alamat) {

this.nip = nip;

this.nama = nama;

this.alamat = alamat;

}

public String getNama() {

return nama;

}

public int getGaji() {

return gaji;

}

}

* Class Dosen

package tugas;

/\*\*

\*

\* @author asus

\*/

public class Dosen extends Pegawai{

public int jumlahSKS;

public int TARIF\_SKS;

public Dosen(String nip, String nama, String alamat) {

super(nip, nama, alamat);

}

public void setSKS(int jumlahSKS) {

this.jumlahSKS = jumlahSKS;

}

@Override

public int getGaji() {

gaji= TARIF\_SKS\*jumlahSKS;

return gaji;

}

}

* Class DaftarGaji

package tugas;

/\*\*

\*

\* @author asus

\*/

public class DaftarGaji {

private final Pegawai[] listPegawai;

DaftarGaji(int pegawai){

this.listPegawai=new Pegawai[pegawai];

this.dtpegawai();

}

private void dtpegawai(){

for(int i=0;i<this.listPegawai.length;i++){

this.listPegawai[i]=new Pegawai(null, null, null);

}

}

public void addPegawai(Pegawai pegawai){

for(int i=0;i<this.listPegawai.length;i++){

if(this.listPegawai[i].getNama()==null){

this.listPegawai[i]=pegawai;

break;

}

}

}

public void printSemuaGaji(){

for (Pegawai gjpegawai : this.listPegawai)

if (gjpegawai.getNama() != null){

System.out.println("Nama Pegawai : "+ gjpegawai.getNama());

System.out.println("Besar Gaji Pegawai : "+ gjpegawai.getGaji());

}

}

public static void main(String[] args) {

Pegawai listpg1,listpg2,listpg3;

listpg1=new Pegawai("2101","Ilham","Malang");

listpg2=new Pegawai("2102","Lutfi","Malang");

listpg3=new Pegawai("2103","Fian","Malang");

listpg1.gaji=2000000;

listpg2.gaji=1500000;

listpg3.gaji=1000000;

Dosen dsn=new Dosen("2001", "Syah", "Malang");

dsn.TARIF\_SKS=200000;

dsn.setSKS(5);

DaftarGaji gj = new DaftarGaji(3);

gj.addPegawai(listpg1);

gj.addPegawai(listpg2);

gj.addPegawai(listpg3);

gj.printSemuaGaji();

System.out.println("Nama Dosen : "+dsn.nama);

System.out.println("Besar Gaji Dosen : "+dsn.getGaji());

}

}

Hasilnya:

