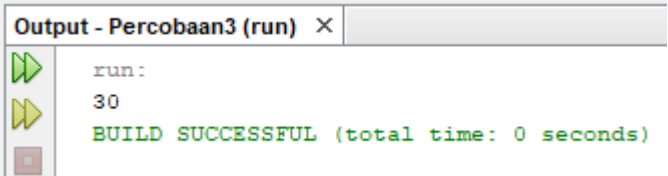
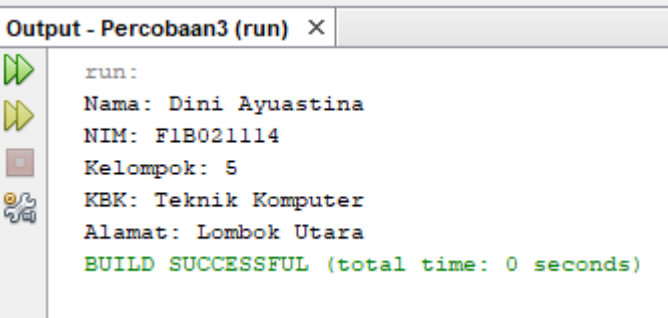
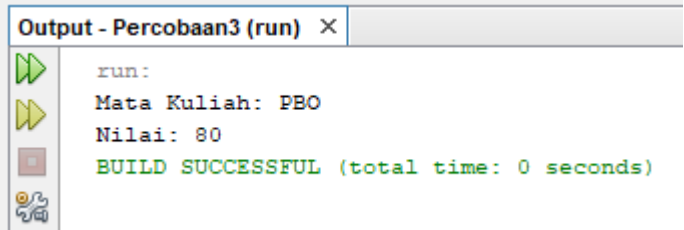
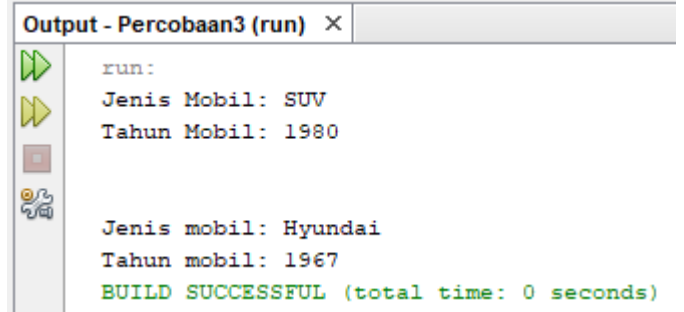
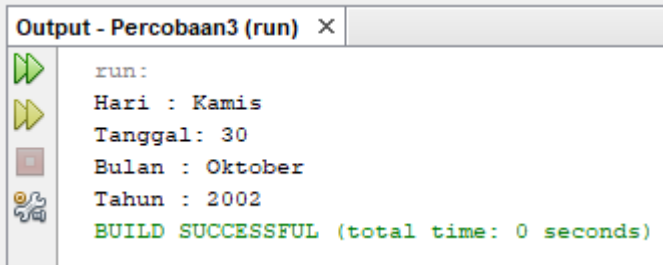


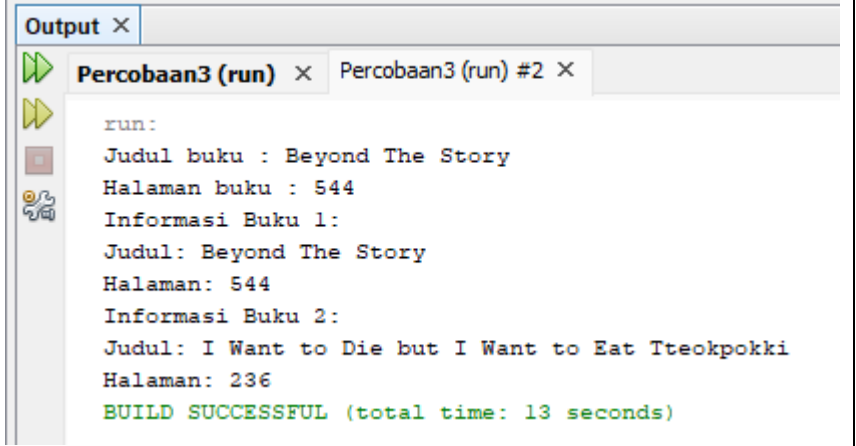
Nama : Dini Ayuastina  
 NIM; F1B021114  
 Kelompok : 5

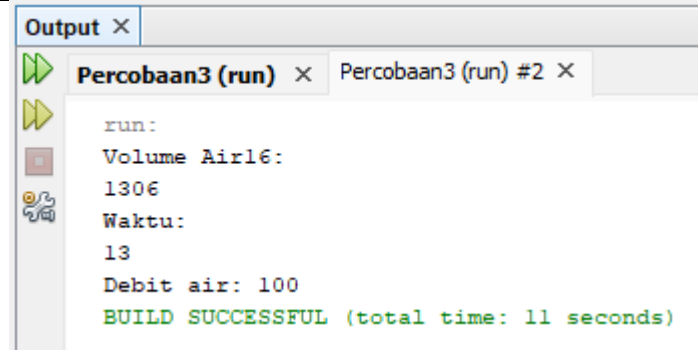
### Jobsheet Constructor

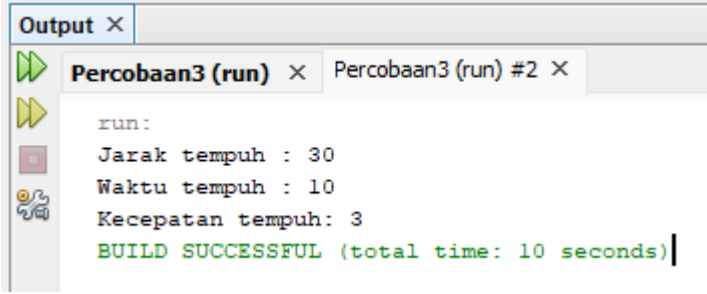
NO	Kegiatan	Script	Hasil Running
1	Method constructor – membuat constructor tanpa parameter	<pre>public class Mahasiswa {     int x = 30;      public static void main(String[] args){         Mahasiswa mhs = new Mahasiswa();         System.out.println(mhs.x);     } }</pre>	 <pre>Output - Percobaan3 (run) X run: 30 BUILD SUCCESSFUL (total time: 0 seconds)</pre>
2	Method constructor – membuat constructor dengan parameter	<pre>public class Identitas{     String nama;     String nim;     int kelompok;     String kbk;     String alamat;      public Identitas (String nama, String nim, int kelompok, String kbk, String alamat) {         this.nama = nama;         this.nim = nim;         this.kelompok = kelompok;         this.kbk = kbk;         this.alamat = alamat;     }      public static void main(String[] args) {         Identitas mahasiswa = new Identitas("Dini Ayuastina", "F1B021114", 5, "Teknik Komputer", "Lombok Utara");          System.out.println("Nama: " + mahasiswa.nama);         System.out.println("NIM: " +</pre>	 <pre>Output - Percobaan3 (run) X run: Nama: Dini Ayuastina NIM: F1B021114 Kelompok: 5 KBK: Teknik Komputer Alamat: Lombok Utara BUILD SUCCESSFUL (total time: 0 seconds)</pre>

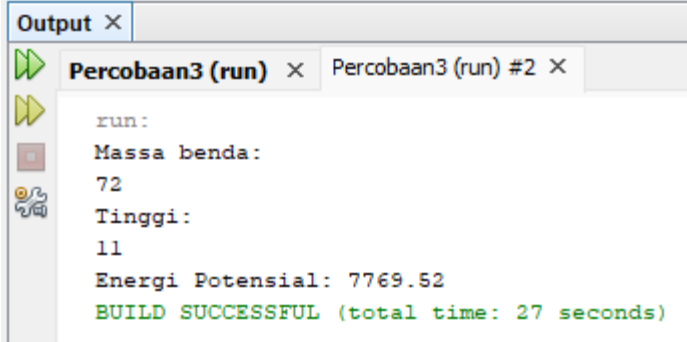
		<pre> mahasiswa.nim);         System.out.println("Kelompok: " + mahasiswa.kelompok);         System.out.println("KBK: " + mahasiswa.kbk);         System.out.println("Alamat: " + mahasiswa.alamat);     } } </pre>	
3	Method constructor – parameter yang berbeda	<pre> public class Nilai_Mahasiswa{     String mk;     int nilai;      public Nilai_Mahasiswa(String mk, int nilai) {         this.mk = mk;         this.nilai = nilai;     }      public static void main(String[] args) {         Nilai_Mahasiswa mahasiswa = new Nilai_Mahasiswa("PBO", 80);          System.out.println("Mata Kuliah: " + mahasiswa.mk);         System.out.println("Nilai: " + mahasiswa.nilai);     } } </pre>	 <pre> Output - Percobaan3 (run) X run: Mata Kuliah: PBO Nilai: 80 BUILD SUCCESSFUL (total time: 0 seconds) </pre>
4	Method Constructor – gabungan	<pre> public class Mobil{     private String nama;     private int tahun;      public Mobil() {         nama = "Hyundai";         tahun = 1967;     }     public Mobil(String nm, int tahun) {         this.tahun = tahun;         this.nama = nm;     }     public String getNama(){         return nama;     } } </pre>	 <pre> Output - Percobaan3 (run) X run: Jenis Mobil: SUV Tahun Mobil: 1980  Jenis mobil: Hyundai Tahun mobil: 1967 BUILD SUCCESSFUL (total time: 0 seconds) </pre>

		<pre> public int getTahun(){     return tahun; }  public static void main(String[] args){     Mobil mbl1 = new Mobil("SUV", 1980);     System.out.println("Jenis Mobil: " + mbl1.getNama());     System.out.println("Tahun Mobil: " + mbl1.getTahun());     System.out.println("");     System.out.println("");      Mobil mbl2 = new Mobil();     System.out.println("Jenis mobil: " + mbl2.nama);     System.out.println("Tahun mobil: " + mbl2.tahun); } } </pre>	
5	Method constructor dengan 2 constructor	<pre> /*  * To change this license header, choose License Headers in Project Properties.  * To change this template file, choose Tools   Templates  * and open the template in the editor.  */ package percobaan3;  /**  *  * @author DINI AYUASTINA  */ public class birthday {     String hari;     String bulan;     int tanggal;     int tahun;      public birthday(String hari, String bulan) {         this.hari = hari;         this.bulan = bulan; </pre>	 <pre> run: Hari : Kamis Tanggal: 30 Bulan : Oktober Tahun : 2002 BUILD SUCCESSFUL (total time: 0 seconds) </pre>

		<pre>         }         public birthday(int tanggal, int tahun) {             this.tanggal = tanggal;             this.tahun = tahun;         }         public static void main(String[] args) {             birthday lahir1 = new birthday("Kamis", "Oktober");             birthday lahir2 = new birthday(30, 2002);              System.out.println("Hari : " + lahir1.hari);             System.out.println("Tanggal: " + lahir2.tanggal);             System.out.println("Bulan : " + lahir1.bulan);             System.out.println("Tahun : " + lahir2.tahun);         }     } </pre>	
6	Method constructor dengan 2 constructor (input secara dinamis)	<pre> import java.util.Scanner; public class buku {     String judul;     int halaman;      public buku(String judul, int halaman) {         this.judul = judul;         this.halaman = halaman;     }     public buku() {         this.judul = "I Want to Die but I Want to Eat Tteokpokki";         this.halaman = 236;     }      public static void main(String[] args) {         Scanner input = new Scanner(System.in);         System.out.print("Judul buku : "); </pre>	

		<pre> String judul = input.nextLine(); System.out.print("Halaman buku : ");  int halaman = input.nextInt();  buku buku1 = new buku(judul, halaman);  System.out.println("Informasi Buku 1:"); System.out.println("Judul: " + buku1.judul); System.out.println("Halaman: " + buku1.halaman);  buku buku2 = new buku();  System.out.println("Informasi Buku 2:"); System.out.println("Judul: " + buku2.judul); System.out.println("Halaman: " + buku2.halaman); } } </pre>	
7	method constructor dengan perhitungan (menghitung debit air secara dinamis)	<pre> import java.util.Scanner;  public class debit {     int hitungDebit;     int volume;     int waktu;      public debit(int volume, int waktu)     {         hitungDebit = volume / waktu;     }      public int hitungDebit() {         return hitungDebit;     }      public static void main(String[] args) {         Scanner input = new Scanner(System.in);         System.out.println("Volume </pre>	 <pre> Output X Percobaan3 (run) x Percobaan3 (run) #2 x run: Volume Airl6: 1306 Waktu: 13 Debit air: 100 BUILD SUCCESSFUL (total time: 11 seconds) </pre>

		<pre> Air16: ");         int volume = input.nextInt();         System.out.println("Waktu: ");         int waktu = input.nextInt();          debit nilai = new debit(volume, waktu);         System.out.println("Debit air: " + nilai.hitungDebit);     } } </pre>	
8	<p>Methode constructor dengan perhitungan (menghitung kecepatan mobil secara dinamis)</p>	<pre> import java.util.Scanner; public class kecepatan {     int hitungKecepatan;     int jarak;     int waktu;      public kecepatan(int jarak, int waktu) {         hitungKecepatan = jarak / waktu;     }     public int hitungKecepatan() {         return hitungKecepatan;     }     public static void main(String[] args) {         Scanner input = new Scanner(System.in);         System.out.print("Jarak tempuh : ");          int jarak = input.nextInt();         System.out.print("Waktu tempuh : ");          int waktu = input.nextInt();         kecepatan hasil = new kecepatan(jarak, waktu);         System.out.println("Kecepatan tempuh: " + hasil.hitungKecepatan());     } } </pre>	

9	<p>Methode constructor dengan perhitungan (menghitung energi potensial secara dinamis)</p>	<pre>import java.util.Scanner; public class potensial {     double energiPotensial;     final double gravitasi = 9.81;      public potensial(double massa, double gravitasi, double tinggi) {         energiPotensial = massa * gravitasi * tinggi;     }     public double getEnergiPotensial() {     return energiPotensial; }     public static void main(String[] args) {         Scanner input = new Scanner(System.in);         System.out.println("Massa benda: ");         double massa = input.nextInt();         System.out.println("Tinggi: ");         double tinggi = input.nextInt();         potensial energi = new potensial(massa, 9.81, tinggi);         double hasilEnergiPotensial = energi.getEnergiPotensial();         System.out.println("Energi Potensial: " + hasilEnergiPotensial);     } }</pre>	 <pre>Output x Percobaan3 (run) x Percobaan3 (run) #2 x  run: Massa benda: 72 Tinggi: 11 Energi Potensial: 7769.52 BUILD SUCCESSFUL (total time: 27 seconds)</pre>
---	--	--	---