1. **DEFINISI MASALAH**
2. Buatlah class Employee yang d-*inherit* oleh class Manager, Pegawai tetap, Pegawai tidak tetap. Setelah itu program akan meminta user untuk menginputkan nama, idKerja, jabatan employee, mempunyai istri atau tidak, mempunyai anak atau tidak dan memasukkan tahun masuk kerja. Ketentuan untuk mengerjakan soal ini adalah sebagai berikut :
3. Berlaku untuk semua jabatan (kecuali non PNS)

* Jika masa kerja di bawah 5 tahun maka tidak mendapatkan bonus dan tidak mendapatkan tunjangan (sama dengan employee)
* Jika masa kerja di antara 6-10 tahun maka mendapatkan bonus 0.05 dari lama kerja dan tidak mendapatkan tunjangan
* Jika masa kerja di atas 10 tahun maka mendapatkan bonus 0.1 dari lama kerja dan mendapatkan tunjangan 0.1 dari lama kerja
* Jika mempunyai istri maka akan mendapatkan tunjangan 0.1 dari total gaji yang didapat (gaji+lama kerja+tunjangan) jika tidak maka tunjangan istri 0 rupiah
* Jika mempunyai anak maka mendapatkan tunjangan anak sebesar 0.15 dari jumlah gaji yang di dapatkan (gaji+lama kerja+tunjangan), tunjangan anak hanya diberikan sampai anak ketiga saja.

1. Berlaku untuk Manager saja

* Untuk manager selain mendapatkan bonus kerja dan tunjangan, pada manager mendapatkan tunjangan jabatan sebesar 0.1 dari jumlah gaji (gaji+lama kerja+tunjangan).

1. Berlaku untuk pegawai tidak tetap

* Gaji yang didapatkan adalah sama dengan gaji employee, namun terdapat gaji lembur. Untuk mendapatkan gaji lembur, pegawai non PNS harus bekerja lebih dari 10 jam dengan mendapatkan gaji lembur sebesar 10.000 perjamnya. Namun jika di bawah 10 jam maka tidak akan mendapatkan gaji lembur.

Untuk mengerjakan soal ini gunakan test case sebanyak 1 Manager, 10 Pegawai

tetapdan 5 pegawai tidak tetap yang mencakup seluruh ketentuan yang ada.

**B. SOURCE CODE**

Employee.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | package PrakSIH;  public class Employee {  String name, idKerja, jabatan, istri, anak;  int jmlanak, thnkerja, lamakerja, gaji;  double tunjangan, tunjangananak, tunjanganistri, tunjanganmasa, bonus;  public Employee(String name, String id, String jabatan, String istri,  String anak, int thnkerja, int gaji, int jmlanak) {  this.name = name;  this.idKerja = id;  this.jabatan = jabatan;  this.istri = istri;  this.anak = anak;  this.thnkerja = thnkerja;  this.gaji = gaji;  this.jmlanak = jmlanak;  }  public Employee() {  }  public void setLamaKerja (int lamakerja){  this.lamakerja = lamakerja;  }  public int lamaKerja() {  lamakerja = 2016 - thnkerja;  return lamakerja;  }  public void setID(String idKerja){  this.idKerja = idKerja;  }  public String getID() {  return idKerja;  }  public double tunjanganDanBonus() {  if (lamakerja <= 5) {  bonus = 0;  tunjanganmasa = 0;  if (istri.equalsIgnoreCase("Ya")) {  tunjanganistri = 0.1 \* gaji;  }  if (anak.equalsIgnoreCase("Ya")) {  tunjangananak = ((0.15 \* jmlanak) \* gaji);  }  tunjangan = tunjangananak + tunjanganistri + tunjanganmasa + bonus;  }  if (lamakerja >= 6 && lamakerja <= 10) {  bonus = (lamakerja \* 0.05 \* gaji);  tunjanganmasa = 0;  if (istri.equalsIgnoreCase("Ya")) {  tunjanganistri = 0.1 \* gaji;  }  if (anak.equalsIgnoreCase("Ya")) {  tunjangananak = ((0.15 \* jmlanak) \* gaji);  }  tunjangan = tunjangananak + tunjanganistri + tunjanganmasa + bonus;  }  if (lamakerja > 10) {  bonus = (lamakerja \* 0.01) \* gaji;  tunjanganmasa = (lamakerja \* 0.01) \* gaji;  if (istri.equalsIgnoreCase("Ya")) {  tunjanganistri = 0.1 \* gaji;  }  if (anak.equalsIgnoreCase("Ya")) {  tunjangananak = ((0.15 \* jmlanak) \* gaji);  }  tunjangan = tunjangananak + tunjanganistri + tunjanganmasa + bonus;  }  return tunjangan;  }  public String getName() {  return name;  }  public double getGaji() {  return gaji + tunjangan;  }  public String getJabatan(){  return jabatan;  }  } |

Manager.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 | package PrakSIH;  import java.util.Scanner;  public class Manager extends Employee {  private double tunjanganjabatan;  public Manager(String name, String id, String jabatan, String istri,  String anak, int thnkerja, int gaji, int jmlanak) {  super(name, id, jabatan, istri, anak, thnkerja, gaji, jmlanak);  }  public Manager() {  }  public double getTunjanganJabatan() {  return this.tunjanganjabatan = 0.1 \* (gaji + lamakerja + tunjangan);  }  public double getGaji() {  double gajiManager = super.getGaji();  return gajiManager + getTunjanganJabatan();  }  public void mene(){  Scanner input = new Scanner (System.in);  System.out.println("----------------DATA MANAGER----------------");  System.out.print("Masukkan Nama : ");  super.name = input.nextLine();  System.out.print("Masukkan ID : ");  super.idKerja = input.nextLine();  System.out.print("Masukkan Jabatan : ");  super.jabatan = input.nextLine();  System.out.print("Masukkan Tahun Bekerja : ");  super.thnkerja = input.nextInt();  System.out.print("Mempunyai Istri (Ya/Tidak) : ");  super.istri = input.next();  System.out.print("Mempunyai Anak (Ya/Tidak) : ");  super.anak = input.next();  if (anak.equalsIgnoreCase("Ya")) {  System.out.print("Masukkan Jumlah anak : ");  jmlanak = input.nextInt();  } else {  System.out.println();  }  }  public void cetakManager(){  mene();  super.gaji = 1000000; System.out.println("============================================");  System.out.println(" DATA GAJI MANAGER "); System.out.println("============================================");  System.out.println("Nama : " + getName());  System.out.println("ID : " + getID());  System.out.println("Jabatan : " + getJabatan());  System.out.println("Lama Kerja : " + lamaKerja() + " Tahun");  System.out.println("Gaji Awal : Rp." + gaji);  System.out.println("Tunjangan dan Bonus : Rp." + (super.tunjanganDanBonus()));  System.out.println("Tunjangan Jabatan : Rp." + getTunjanganJabatan());  System.out.println("--------------------------------------------");  System.out.println("Total Gaji : Rp." + (double)(super.getGaji()+getTunjanganJabatan()));  System.out.println();  }  } |

PegawaiTetap.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62 | package PrakSIH;  import java.util.Scanner;  public class PegawaiTetap extends Employee {  private int jamLembur, hariLembur;  public PegawaiTetap(String name, String id, String jabatan, String istri,  String anak, int thnkerja, int gaji, int jmlanak) {  super(name, id, jabatan, istri, anak, thnkerja, gaji, jmlanak);  }  public PegawaiTetap(){  }  public void pegTetap(){  Scanner input = new Scanner (System.in);  System.out.println("-------------DATA PEGAWAI TETAP -------------");  System.out.print("Masukkan Nama : ");  super.name = input.nextLine();  System.out.print("Masukkan ID : ");  super.idKerja = input.nextLine();  System.out.print("Masukkan Jabatan : ");  super.jabatan = input.nextLine();  System.out.print("Masukkan Tahun Bekerja : ");  super.thnkerja = input.nextInt();  System.out.print("Mempunyai Istri (Ya/Tidak) : ");  super.istri = input.next();  System.out.print("Mempunyai Anak (Ya/Tidak) : ");  super.anak = input.next();  if (anak.equalsIgnoreCase("Ya")) {  System.out.print("Masukkan Jumlah anak : ");  jmlanak = input.nextInt();  } else {  System.out.println();  }  }  public void cetakPegawaiTetap(){  pegTetap();  super.gaji = 500000; System.out.println("=============================================");  System.out.println(" DATA GAJI PEGAWAI TETAP "); System.out.println("=============================================");  System.out.println("Nama : " + getName());  System.out.println("ID : " + getID());  System.out.println("Jabatan : " + getJabatan());  System.out.println("Lama Kerja : " + lamaKerja() + " Tahun");  System.out.println("Gaji Awal : Rp." + gaji);  System.out.println("Tunjangan dan Bonus : Rp." + (super.tunjanganDanBonus()));  System.out.println("---------------------------------------------");  System.out.println("Total Gaji : Rp." + getGaji());  System.out.println();  }  } |

PegawaiTidakTetap.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  7677  7879  80 | package PrakSIH;  import java.util.Scanner;  public class PegawaiTidakTetap extends Employee {  private int gajilembur, jamKerja, hariLembur;  public PegawaiTidakTetap(String name, String id, String jabatan, String istri,  String anak, int thnkerja, int gaji, int jmlanak) {  super(name, id, jabatan, istri, anak, thnkerja, gaji, jmlanak);  }  public PegawaiTidakTetap(){  }  public void setJamKerja(int a) {  jamKerja = a;  }  public double getGajiLembur() {  if (jamKerja > 10) {  gajilembur = ((jamKerja - 10) \* 10000)\*hariLembur;  } else {  gajilembur = 0;  }  return gajilembur;  }  public double getGaji() {  return gaji + getGajiLembur();  }  public void pegTidak(){  Scanner input = new Scanner (System.in);  System.out.println("----------DATA PEGAWAI TIDAK TETAP----------");  System.out.print("Masukkan Nama : ");  super.name = input.nextLine();  System.out.print("ID : ");  super.idKerja = input.nextLine();  System.out.print("Masukkan Jabatan : ");  super.jabatan = input.nextLine();  System.out.print("Masukkan Tahun Bekerja : ");  super.thnkerja = input.nextInt();  System.out.print("Total Hari Lembur : ");  this.hariLembur = input.nextInt();  System.out.print("Total Jam Kerja : ");  this.jamKerja = input.nextInt();  System.out.print("Mempunyai Istri (Ya/Tidak) : ");  super.istri = input.next();  System.out.print("Mempunyai Anak (Ya/Tidak) : ");  super.anak = input.next();  if (anak.equalsIgnoreCase("Ya")) {  System.out.print("Masukkan Jumlah anak : ");  jmlanak = input.nextInt();  } else {  System.out.println();  }  }  public void cetakTidakTetap(){  pegTidak();  super.gaji = 250000; System.out.println("============================================");  System.out.println(" DATA GAJI PEGAWAI TIDAK TETAP "); System.out.println("============================================");  System.out.println("Nama Staff : " + getName());  System.out.println("ID : " + getID());  System.out.println("Jabatan : " + getJabatan());  System.out.println("Lama Kerja : " + lamaKerja() + " Tahun");  System.out.println("Gaji Awal : Rp." + gaji);  System.out.println("Gaji Lembur : Rp." + getGajiLembur());  System.out.println("--------------------------------------------");  System.out.println("Total Gaji : Rp." + (double)(super.getGaji()+getGajiLembur()));  System.out.println();  }  } |

MainEmployee.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | package PrakSIH;  public class MainEmployee {  public static void main(String[] args) {  Manager m1 = new Manager ();  m1.cetakManager();  PegawaiTetap pt1 = new PegawaiTetap();  pt1.cetakPegawaiTetap();  PegawaiTetap pt2 = new PegawaiTetap();  pt2.cetakPegawaiTetap();  PegawaiTetap pt3 = new PegawaiTetap();  pt3.cetakPegawaiTetap();  PegawaiTetap pt4 = new PegawaiTetap();  pt4.cetakPegawaiTetap();  PegawaiTetap pt5 = new PegawaiTetap();  pt5.cetakPegawaiTetap();  PegawaiTetap pt6 = new PegawaiTetap();  pt6.cetakPegawaiTetap();  PegawaiTetap pt7 = new PegawaiTetap();  pt7.cetakPegawaiTetap();  PegawaiTetap pt8 = new PegawaiTetap();  pt8.cetakPegawaiTetap();  PegawaiTetap pt9 = new PegawaiTetap();  pt9.cetakPegawaiTetap();  PegawaiTetap pt10 = new PegawaiTetap();  pt10.cetakPegawaiTetap();  PegawaiTidakTetap nonpns1 = new PegawaiTidakTetap();  nonpns1.cetakTidakTetap();  PegawaiTidakTetap nonpns2 = new PegawaiTidakTetap();  nonpns2.cetakTidakTetap();  PegawaiTidakTetap nonpns3 = new PegawaiTidakTetap();  nonpns3.cetakTidakTetap();  PegawaiTidakTetap nonpns4 = new PegawaiTidakTetap();  nonpns4.cetakTidakTetap();  PegawaiTidakTetap nonpns5 = new PegawaiTidakTetap();  nonpns5.cetakTidakTetap();  }  } |

**C. PEMBAHASAN**

Employee

|  |  |
| --- | --- |
| 1  2  3  4  5  7  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  46  47  48  49  50  51  52  53  54  55  56  58  59  60  61  62  63  64  65  66  67  68  70  71  72  73  74  75  76  77  78  79  80  81  82 | Deklarasi package PrakSIH  Deklarasi class bernama Employee  Deklarasi variable bernama name, idKerja, jabatan, istri, anak bertipe data String  Deklarasi variable bernama jmlanak, thnkerja, lamakerja, gaji bertipe data integer  Deklarasi variable bernama tunjangan, tunjangananak, tunjanganistri, tunjanganmasa, bonus bertipe data double  Membuat konstruktor Employee dengan parameter String name, String id, String jabatan, String istri, String anak, int thnkerja, int gaji, int jmlanak  Inisialisasi variable name sama dengan parameter name  Inisialisasi variable idKerja sama dengan parameter id  Inisialisasi variable jabatan sama dengan parameter jabatan  Inisialisasi variable istri sama dengan parameter istri  Inisialisasi variable anak sama dengan parameter anak  Inisialisasi variable thnkerja sama dengan parameter thnkerja  Inisialisasi variable gaji sama dengan parameter gaji  Inisialisasi variable jmlanak sama dengan parameter jmlanak  Menutup konstruktor Employee  Membuat konstruktor default yaitu konstruktor tanpa parameter  Menutup konstruktor default  Membuat method public bernama setLamaKerja dengan parameter int lamakerja  Inisialisasi variable lamakerja sama dengan parameter lamakerja  Menutup method setLamaKerja  Membuat method public bernama lamaKerja  Proses matematika lamakerja = 2016 – thnkerja  Mengembalikan nilai variable lamakerja  Menutup method lamaKerja  Membuat method public bernama setID dengan parameter String idKerja  Inisialisasi variable idKerja sama dengan parameter idKerja  Menutup method setID  Membuat method public bernama getID  Mengembalikan nilai variable getID  Menutup method getID  Membuat method public bernama tunjanganDanBonus  Proses percabangan if dengan syarat lamakerja <= 5  Inisialisasi variable bonus = 0  Inisialisasi variable tunjanganmasa = 0  Proses percabangan if dengan syarat nilai variable istri adalah “Ya”  Proses matematika tunjanganistri = 0.1 \* gaji  Menutup proses percabangan if  Proses percabangan if dengan syarat nilai variable anak adalah “Ya”  Proses matematika tunjangananak = ((0.15 \* jmlanak) \* gaji)  Menutup proses percabangan if  Proses matematika tunjangan = tunjangananak + tunjanganistri + tunjanganmasa + bonus  Menutup proses percabangan if  Proses percabangan if dengan syarat lamakerja >= 6 && lamakerja <=10  Inisialisasi variable bonus = (lamakerja \* 0.05 \* gaji)  Inisialisasi variable tunjanganmasa = 0  Proses percabangan if dengan syarat nilai variable istri adalah “Ya”  Proses matematika tunjanganistri = 0.1 \* gaji  Menutup proses percabangan if  Proses percabangan if dengan syarat nilai variable anak adalah “Ya”  Proses matematika tunjangananak = ((0.15 \* jmlanak) \* gaji)  Menutup proses percabangan if  Proses matematika tunjangan = tunjangananak + tunjanganistri + tunjanganmasa + bonus  Menutup proses percabangan if  Proses percabangan if dengan syarat lamakerja >10  Inisialisasi variable bonus = (lamakerja \* 0.01 \* gaji)  Inisialisasi variable tunjanganmasa = (lamakerja \* 0.01) \* gaji  Proses percabangan if dengan syarat nilai variable istri adalah “Ya”  Proses matematika tunjanganistri = 0.1 \* gaji  Menutup proses percabangan if  Proses percabangan if dengan syarat nilai variable anak adalah “Ya”  Proses matematika tunjangananak = ((0.15 \* jmlanak) \* gaji)  Menutup proses percabangan if  Proses matematika tunjangan = tunjangananak + tunjanganistri + tunjanganmasa + bonus  Menutup proses percabangan if  Mengembalikan nilai variable tunjangan  Menutup method tunjanganDanBonus  Membuat method public bernama getName  Mengembalikan nilai variable name  Menutup method getName  Membuat method public bernama getGaji  Mengembalikan nilai variable gaji + tunjangan  Menutup method getGaji  Membuat method public bernama getJabatan  Mengembalikan nilai variable jabatan  Menutup method getJabatan  Menutup public class |

Manager

|  |  |
| --- | --- |
| 1  2  3  4  5  8  10  11  12  13  14  16  17  18  19  20  21  22  23  25  26  27  28  29  30  31  32  33  34  35  36  37  38  40  41  42  43  44  45  46  47  48  50  53  54  56  57  59  61  62  64  66  68  70  71  72 | Deklarasi package PrakSIH  Mengimpor utilitas java yaitu Scanner  Deklarasi class bernama Manager turunan dari class Employee  Deklarasi variable bernama tunjanganjabatan bertipe data double dan akses modifier private  Membuat konstruktor Manager dengan parameter String name, String id, String jabatan, String istri, String anak, int thnkerja, int gaji, int jmlanak  Deklarasi super untuk memanggil superclass dengan parameter name, id, jabatan, istri, anak, thnkerja, gaji, jmlanak  Menutup konstruktor Manager  Membuat konstruktor default yaitu konstruktor tanpa parameter  Menutup konstruktor default  Membuat method public getTunjanganJabatan  Mengembalikan nilai variable tunjanganjabatan = 0.1 \* (gaji + lamakerja + tunjangan)  Menutup method getTunjanganJabatan  Membuat method public bernama getGaji  Deklarasi variable gajiManager bertipe data double = method getGaji di superclass  Mengembalikan nilai variable gajiManager + getTunjanganJabatan  Menutup method getGaji  Membuat method public bernama mene  Pendeklarasian new Scanner in  Proses mencetak “----------------DATA MANAGER----------------“  Proses mencetak “Masukkan Nama : “  User menginput nilai variable name  Proses mencetak “Masukkan ID : “  User menginput nilai variable idKerja  Proses mencetak “Masukkan Jabatan : “  User menginput nilai variable jabatan  Proses mencetak “Masukkan Tahun Bekerja: “  User menginput nilai variable thnkerja  Proses mencetak “Mempunyai Istri (Ya/Tidak) : “  User menginput nilai variable istri  Proses mencetak “Mempunyai Anak (Ya/Tidak) : “  User menginput nilai variable anak  Proses percabangan if dengan syarat nilai variable anak adalah “Ya”  Proses mencetak “Masukkan Jumlah Anak : “  User menginput nilai variable jmlanak  Menutup proses percabangan if, proses percabangan else  Proses mencetak baris kosong  Menutup proses percabangan else  Menutup method mene  Membuat method public bernama cetakManager  Memanggil method mene  Inisialisasi nilai variable gaji pada superclass adalah 1000000  Proses mencetak “============================================”  Proses mencetak “ DATA GAJI MANAGER “  Proses mencetak “============================================”  Proses mencetak “Nama : “ dan method getName  Proses mencetak “ID : “ dan method getID  Proses mencetak “Jabatan : “ dan method getJabatan  Proses mencetak “Lama Kerja : “ dan method lamaKerja dan “ Tahun”  Proses mencetak “Gaji Awal : Rp.“ dan nilai variable gaji  Proses mencetak “Tunjangan dan Bonus : Rp.“ dan method tunjanganDanBonus pada superclass  Proses mencetak “Tunjangan Jabatan : Rp.“ dan method getTunjanganJabatan  Proses mencetak “----------------------------------------------------------------------”  Proses mencetak “Total Gaji : Rp.“ dan method getGaji pada superclass dan method getTunjanganJabatan  Proses mencetak baris kosong  Menutup method cetakManager  Menutup public class |

PegawaiTetap

|  |  |
| --- | --- |
| 1  2  3  4  5  8  10  11  12  13  14  15  17  18  19  20  21  22  23  24  25  26  27  28  29  30  32  33  34  35  36  37  38  39  40  42  44  46  48  49  51  53  54  56  58  60  61  62 | Deklarasi package PrakSIH  Mengimpor utilitas java yaitu Scanner  Deklarasi class bernama PegawaiTetap turunan dari class Employee  Deklarasi variable bernama jamLembur, hariLembur bertipe data double dan akses modifier private  Membuat konstruktor PegawaiTetap dengan parameter String name, String id, String jabatan, String istri, String anak, int thnkerja, int gaji, int jmlanak  Deklarasi super untuk memanggil superclass dengan parameter name, id, jabatan, istri, anak, thnkerja, gaji, jmlanak  Menutup konstruktor PegawaiTetap  Membuat konstruktor default yaitu konstruktor tanpa parameter  Menutup konstruktor default  Membuat method public bernama pegTetap  Pendeklarasian new Scanner in  Proses mencetak “----------------DATA PEGAWAI TETAP----------------“  Proses mencetak “Masukkan Nama : “  User menginput nilai variable name  Proses mencetak “Masukkan ID : “  User menginput nilai variable idKerja  Proses mencetak “Masukkan Jabatan : “  User menginput nilai variable jabatan  Proses mencetak “Masukkan Tahun Bekerja: “  User menginput nilai variable thnkerja  Proses mencetak “Mempunyai Istri (Ya/Tidak) : “  User menginput nilai variable istri  Proses mencetak “Mempunyai Anak (Ya/Tidak) : “  User menginput nilai variable anak  Proses percabangan if dengan syarat nilai variable anak adalah “Ya”  Proses mencetak “Masukkan Jumlah Anak : “  User menginput nilai variable jmlanak  Menutup proses percabangan if, proses percabangan else  Proses mencetak baris kosong  Menutup proses percabangan else  Menutup method pegTetap  Membuat method public bernama cetakPegawaiTetap  Memanggil method pegTetap  Inisialisasi nilai variable gaji pada superclass adalah 500000  Proses mencetak “============================================”  Proses mencetak “ DATA GAJI PEGAWAI TETAP “  Proses mencetak “============================================”  Proses mencetak “Nama : “ dan method getName  Proses mencetak “ID : “ dan method getID  Proses mencetak “Jabatan : “ dan method getJabatan  Proses mencetak “Lama Kerja : “ dan method lamaKerja dan “ Tahun”  Proses mencetak “Gaji Awal : Rp.“ dan nilai variable gaji  Proses mencetak “Tunjangan dan Bonus : Rp.“ dan method tunjanganDanBonus pada superclass  Proses mencetak “----------------------------------------------------------------------”  Proses mencetak “Total Gaji : Rp.“ dan method getGaji  Proses mencetak baris kosong  Menutup method cetakPegawaiTetap  Menutup public class |

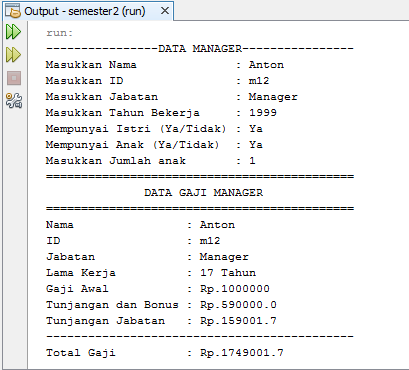
PegawaiTidakTetap

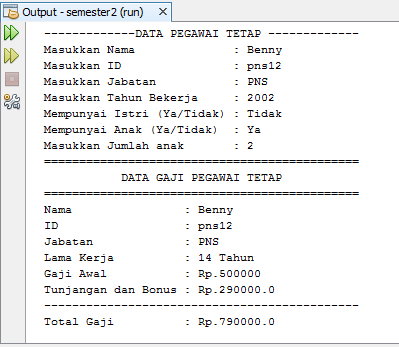
|  |  |
| --- | --- |
| 1  2  3  4  5  8  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  50  51  52  53  54  55  56  57  58  60  62  64  66  67  69  71  72  74  76  78  79  80 | Deklarasi package PrakSIH  Mengimpor utilitas java yaitu Scanner  Deklarasi class bernama PegawaiTidakTetap turunan dari class Employee  Deklarasi variable bernama gajiLembur, jamKerja, hariLembur bertipe data integer dan akses modifier private  Membuat konstruktor PegawaiTidakTetap dengan parameter String name, String id, String jabatan, String istri, String anak, int thnkerja, int gaji, int jmlanak  Deklarasi super untuk memanggil superclass dengan parameter name, id, jabatan, istri, anak, thnkerja, gaji, jmlanak  Menutup konstruktor PegawaiTidakTetap  Membuat konstruktor default yaitu konstruktor tanpa parameter  Menutup konstruktor default  Membuat method bernama setJamKerja dengan parameter int a  Inisialisasi variable jamKerja sama dengan parameter a  Menutup method setJamKerja  Membuat method public getGajiLembur  Proses percabangan if dengan syarat nilai variable jamKerja > 10  Proses matematika gajilembur = ((jamKerja - 10) \* 10000) \* hariLembur  Menutup percabangan if, proses percabangan else  Inisialisasi nilai variable gajilembur = 0  Menutup proses percabangan else  Mengembalikan nilai variable gajilembur  Menutup method getGajiLembur  Membuat method public getGaji  Mengembalikan nilai variable gaji + getGajiLembur  Menutup method getGaji  Membuat method public bernama pegTidak  Pendeklarasian new Scanner in  Proses mencetak “----------------DATA PEGAWAI TIDAK TETAP----------------“  Proses mencetak “Masukkan Nama : “  User menginput nilai variable name  Proses mencetak “Masukkan ID : “  User menginput nilai variable idKerja  Proses mencetak “Masukkan Jabatan : “  User menginput nilai variable jabatan  Proses mencetak “Masukkan Tahun Bekerja: “  User menginput nilai variable thnkerja  Proses mencetak “Total Hari Lembur : “  User menginput nilai variable hariLembur  Proses mencetak “Total Jam Kerja : “  User menginput nilai variable jamKerja  Proses mencetak “Mempunyai Istri (Ya/Tidak) : “  User menginput nilai variable istri  Proses mencetak “Mempunyai Anak (Ya/Tidak) : “  User menginput nilai variable anak  Proses percabangan if dengan syarat nilai variable anak adalah “Ya”  Proses mencetak “Masukkan Jumlah Anak : “  User menginput nilai variable jmlanak  Menutup proses percabangan if, proses percabangan else  Proses mencetak baris kosong  Menutup proses percabangan else  Menutup method pegTidak  Membuat method public bernama cetakTidakTetap  Memanggil method pegTidak  Inisialisasi nilai variable gaji pada superclass adalah 250000  Proses mencetak “============================================”  Proses mencetak “ DATA GAJI PEGAWAI TIDAK TETAP “  Proses mencetak “============================================”  Proses mencetak “Nama Staff : “ dan method getName  Proses mencetak “ID : “ dan method getID  Proses mencetak “Jabatan : “ dan method getJabatan  Proses mencetak “Lama Kerja : “ dan method lamaKerja dan “ Tahun”  Proses mencetak “Gaji Awal : Rp.“ dan nilai variable gaji  Proses mencetak “Gaji Lembur : Rp.“ dan method getGajiLembur  Proses mencetak “----------------------------------------------------------------------”  Proses mencetak “Total Gaji : Rp.“ dan method getGaji pada superclass dan method getGaji  Proses mencetak baris kosong  Menutup method cetakTidakTetap  Menutup public class |

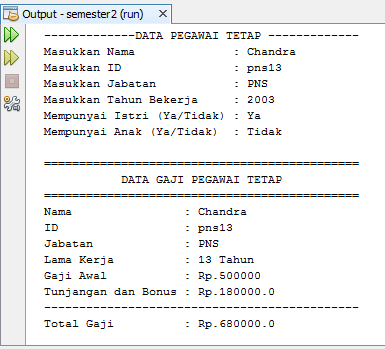
MainEmployee

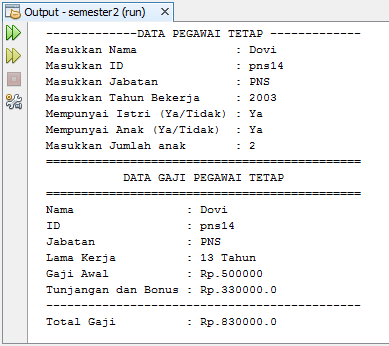
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | Deklarasi package PrakSIH  Deklarasi main class bernama MainEmployee  Deklarasi main method  Membuat object bernama m1 dari class Manager  Pemanggilan method cetakManager oleh object m1  Membuat object bernama pt1 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt1  Membuat object bernama pt2 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt2  Membuat object bernama pt3 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt3  Membuat object bernama pt4 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt4  Membuat object bernama pt5 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt5  Membuat object bernama pt6 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt6  Membuat object bernama pt7 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt7  Membuat object bernama pt8 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt8  Membuat object bernama pt9 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt9  Membuat object bernama pt10 dari class PegawaiTetap  Pemanggilan method cetakPegawaiTetap oleh object pt10  Membuat object bernama nonpns1 dari class PegawaiTidakTetap  Pemanggilan method cetakTidakTetap oleh object nonpns1  Membuat object bernama nonpns2 dari class PegawaiTidakTetap  Pemanggilan method cetakTidakTetap oleh object nonpns2  Membuat object bernama nonpns3 dari class PegawaiTidakTetap  Pemanggilan method cetakTidakTetap oleh object nonpns3  Membuat object bernama nonpns4 dari class PegawaiTidakTetap  Pemanggilan method cetakTidakTetap oleh object nonpns4  Membuat object bernama nonpns5 dari class PegawaiTidakTetap  Pemanggilan method cetakTidakTetap oleh object nonpns5  Menutup main method  Menutup main class |

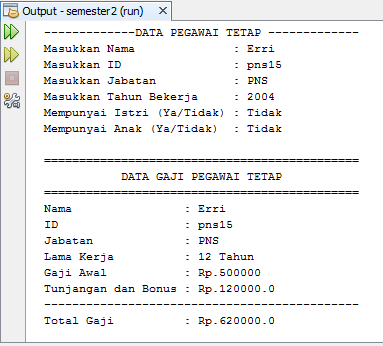
**D. SCREENSHOT PROGRAM**

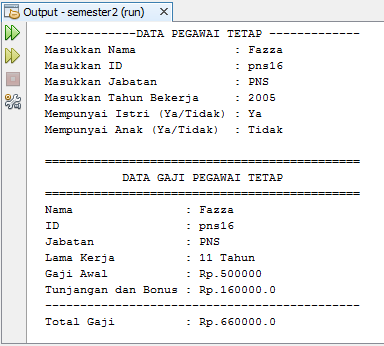


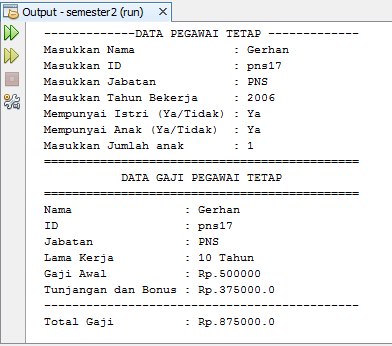


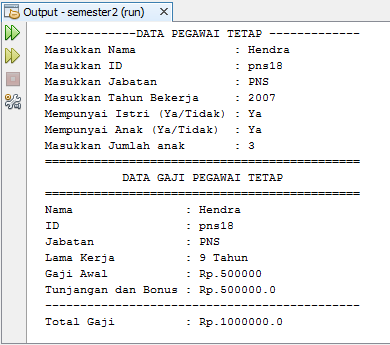


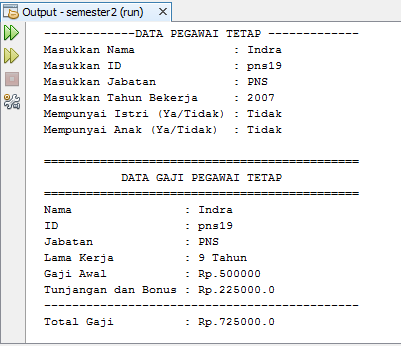


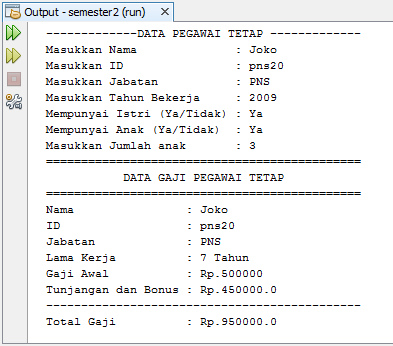


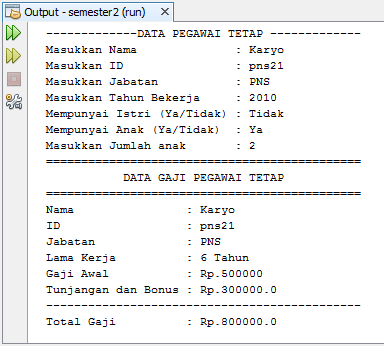


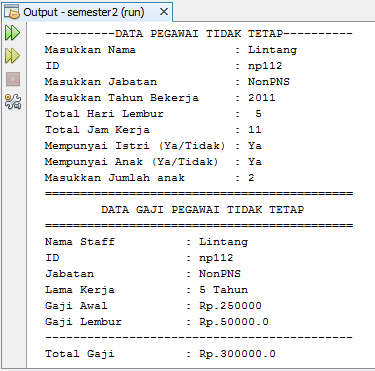


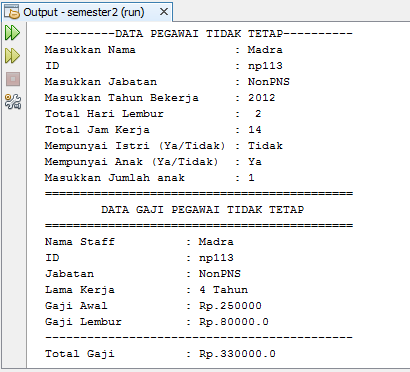


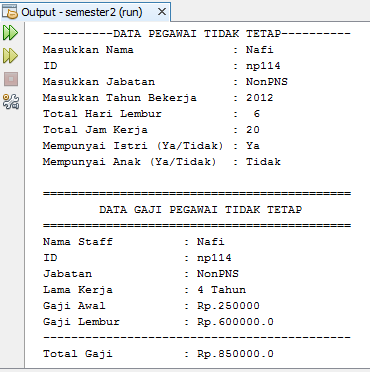


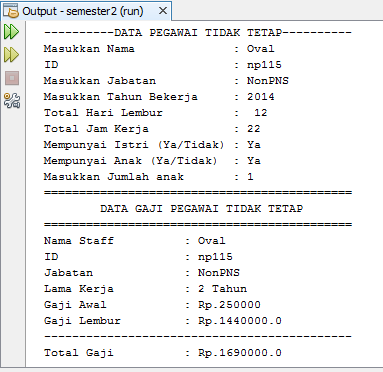


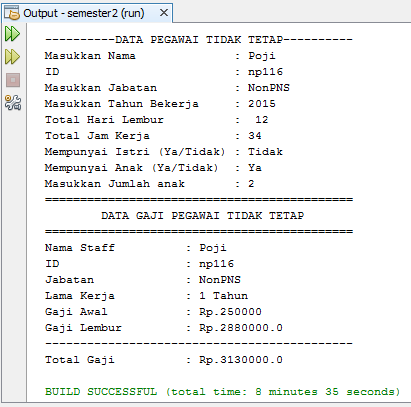








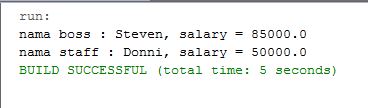




**E. PRAKTIKUM**

1. Jalankan code program diatas dan benahi jika menemukan kesalahan!

Jawab : Tidak ditemukan kesalahan pada code program di atas.



1. Bagaimana cara kontruktor pada subclass memanggil konstruktor di superclass nya? Apakah hal itu perlu dilakukan? Sertakan alasan anda !

Jawab :

KonstruktorSubClass (parameter,…){

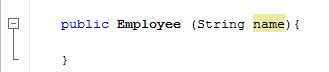
super (parameter,….);

}

Perlu dilakukan, karena dengan menggunakan syntax super() memasukkan parameter yang sama.

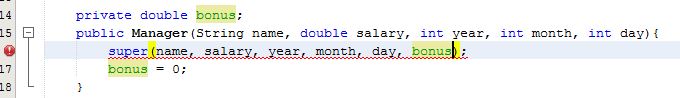
1. Tambahkan constructor pada class Employee dengan parameter *String name*! amati perubahan apa yang terjadi, jelaskan jawaban anda!

Jawab : Tidak terjadi perubahan, karena hanya menambah constructor baru saja.



1. Pada Class Manager baris ke 5, setelah variable day tambahkan variable bonus! Amati apa yang terjadi dan mengapa demikian?

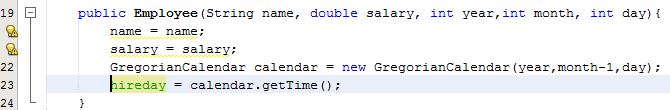
Jawab : Terjadi error. Karena seharusnya parameter dari super sama seperti konstruktor Manager yang diambil dari konstruktor Employee.



Terjadi eror. Karena, seharusnya parameter dari super sama seperti konstruktor Manager yang diambil dari konstruktor Employee.

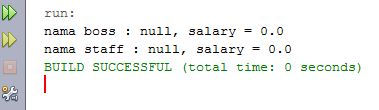
1. Untuk apa digunakan keyword this pada class manager dan employee? Hapus keyword this dan amati apa yang terjadi?

Jawab : Keyword this digunakan untuk menyatakan objek sekarang.



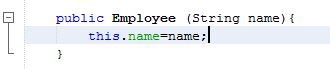
4.JPG

Tidak terjadi eror pada saat penghapusan keyword this, namun akan menghasilkan output :

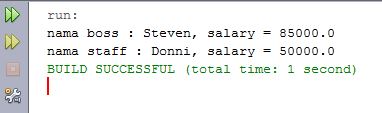


1. Tambahkan constructor pada class Employee dengan parameter Bertipe data string bernama name yang nantinya bila constructor ini akan dipanggil akan menginisialisasi variable name! Amati perubahannya pada class anak dan jelaskan! Benahi bila terjadi kesalahan!

Jawab :



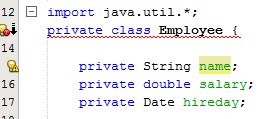
Output :



Tidak terjadi perubahan pada output.

1. Pada bab sebelumnya anda telah belajar mengenai konsep encapsulation, jelaskan mengapa pada super class menggunakan modifier protected? Apa yang terjadi jika modifier anda ubah menjadi private atau public? Jelaskan !

Jawab : Pada super class menggunakan akses modifier protected karena akses modifier ini mampu diakses dari method pada class turunan dalam satu package dan dapat diakses oleh objek dalam satu package. Jika akses modifier diubah akan terjadi error, karena private hanya dapat digunakan dalam 1 package.

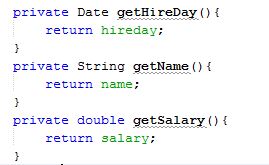


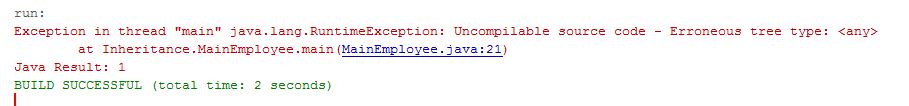
1. Ubahlah acces modifier method pada kelas employee menjadi :
2. Private
3. Protected

Amati perubahan apa yang terjadi? Jelaskan jawaban anda dengan detail!

Jawab :

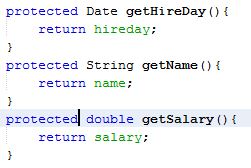
1. Private

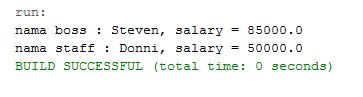




Akan terjadi error pada output. Ini dikarenakan private tidak dapat diakses dari method pada class turunan dalam 1 package.

1. Protected





Tidak terjadi error pada output. Method yang menggunakan protected dapat diakses dari method pada class turunan dalam 1 package.

**F. KESIMPULAN**

* Inheritance memiliki arti pewarisan. Dalam hal ini beberapa data dan method menurunkan suatu karakteristik yang dimiliki oleh data dan method tersebut kepada kelas atau object lain. Kelas yang mewariskan atau menurunkan suatu karakteristik itu disebut dengan super class atau induk class, sedangkan yang memiliki warisan disebut sub class.
* Menerapkan konsep inheritance di java dapat dilakukan dengan menggunakan syntax seperti di bawah ini :

public class [namasubclass] extends [namaclassparent] (berisi parameter){

super (parameter);

}

* Keyword super digunakan untuk memanggil konstruktor dari class parent atau menjadi variabel yang mengacu pada class parent. Cara penggunaannya yaitu seperti contoh berikut :

public class Manager extends Employee (String name){

super (name);

}

* Penggambaran diagram UML dengan konsep inheritance yaitu seperti contoh di bawah ini :

**Employee**

+nama

+idKerja

+gaji

**Programmer**

+bonus

**Manager**

+tunjangan