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

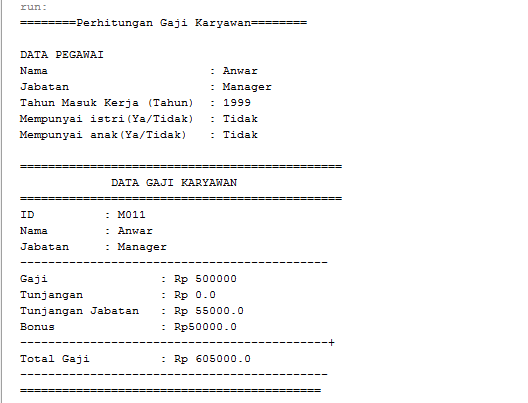
1. **SOURCE CODE**

|  |  |
| --- | --- |
| 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 | public class Employee {  private String nama;  private String idKerja;  private String jabatan;  private String istri;  private String anak;  int jmlanak, ThnKerja, LamaKerja;  double tunjangan, tunjanganAnak, tunjanganIstri, bonus, gaji;  private static int count=0;  public employee(String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak){  this.nama = name;  this.idKerja = id;  this.jabatan=Jabatan;  this.istri = istri;  this.anak = anak;  this.ThnKerja = ThnKerja;  this.gaji = gaji;  this.jmlanak = jmlhAnak;  } public String getNama() {  return nama;  } public double getTunjangan() {  return tunjangan=tunjanganAnak+tunjanganIstri;  }public double getBonus() {  return bonus;  }public double getGaji() {  return gaji + bonus + tunjangan;  }public String getID(){  return idKerja;  }public static int Counting(){  count++;  return count;  }public int LamaKerja(int ThnKerja){  LamaKerja=2015-ThnKerja;  return 2015-ThnKerja;  }public void tunjanganDanBonus(int ThnKerja) {  if (LamaKerja(ThnKerja) <= 5) {  bonus = 0;  tunjangan=0;  if (istri.equalsIgnoreCase("Ya")) {  tunjanganIstri = 0.1 \* gaji;  } else {  tunjanganIstri= 0;  }if (anak.equalsIgnoreCase("Ya")) {  if(jmlanak>3){  tunjanganAnak = ((0.15 \* 3) \* gaji);  }else{  tunjanganAnak = ((0.15 \* jmlanak) \* gaji);  }}}  if (LamaKerja(ThnKerja) >= 6 && LamaKerja(ThnKerja) <= 10) {  bonus = (gaji \* 0.05);  tunjangan=0;  if (istri.equalsIgnoreCase("Ya")) {  tunjanganIstri = ((0.1 \* gaji));  } else {  tunjanganIstri= 0;  } if (anak.equalsIgnoreCase("Ya")) {  if(jmlanak>3){  tunjanganAnak = ((0.15 \* 3) \* gaji);  }else{  tunjanganAnak = ((0.15 \* jmlanak) \* gaji);  }}}  if (LamaKerja(ThnKerja) > 10) {  bonus = (gaji \* 0.1);  tunjangan= (gaji \* 0.1);  if (istri.equalsIgnoreCase("Ya")) {  tunjanganIstri = (0.1 \* gaji) + ((0.1 \* LamaKerja(ThnKerja)) \* gaji);  } else {  tunjanganIstri= 0;  }if (anak.equalsIgnoreCase("Ya")) {  if(jmlanak>3){  tunjanganAnak = ((0.15 \* 3) \* gaji);  }else{  tunjanganAnak = ((0.15 \* jmlanak) \* gaji);  }}}}} |
| 1  2  3  4  5  6  7  8  9  10 | public class manager extends employee {  private double TunjanganJabatan;  public manager(String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak) {  super(name, id, Jabatan, istri, anak, ThnKerja, gaji, jmlhAnak);  }public double getTunjanganJabatan() {  return this.TunjanganJabatan = 0.1 \* (gaji + bonus + tunjangan);  @Override  }public double getGaji() {  double gajiManager = super.getGaji();  return gajiManager + getTunjanganJabatan();  }} |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | public class pegawaiTidakTetap extends employee {  private int GajiLembur, JamKerja;  public pegawaiTidakTetap(String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak) {  super(name, id, Jabatan, istri, anak, ThnKerja, gaji, jmlhAnak);  }public void setJamKerja(int a) {  JamKerja = a;  }public double getGajiLembur() {  if (JamKerja > 10) {  GajiLembur = (JamKerja - 10) \* 10000;  } else {  GajiLembur = 0;  }return GajiLembur;  @Override  }public double getGaji() {  return gaji + getGajiLembur();  }} |
| 1  2  3  4  5  6 | public class PegawaiTetap extends employee {  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);  @Override  }public double getGaji() {  return super.getGaji();  }} |
| 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  83  84  85  86  87  88  89  90  91  92 | import java.util.Scanner;  public class MainEmployee {  public static void main(String[] args) {  System.out.println("========Perhitungan Gaji Karyawan========");  System.out.println("");  System.out.println("DATA PEGAWAI");  Scanner input = new Scanner(System.in);  int jmlhAnak = 0;  boolean ulang = false;  do {  ulang = false;  System.out.print("Nama : ");  String nama = input.next();  System.out.print("Jabatan : ");  String jabatan = input.next();  System.out.print("Tahun Masuk Kerja (Tahun) : ");  int masakerja = input.nextInt();  System.out.print("Mempunyai istri(Ya/Tidak) : ");  String istri = input.next();  System.out.print("Mempunyai anak(Ya/Tidak) : ");  String anak = input.next();  if (anak.equalsIgnoreCase("Ya")) {  System.out.print("\t--Jumlah anak :");  jmlhAnak = input.nextInt();  } if (jabatan.equalsIgnoreCase("NonPNS")) {  System.out.println("Total jam kerja: ");  int jamkerja = input.nextInt();  pegawaiTidakTetap NonPNS = new pegawaiTidakTetap(nama, "037", jabatan, istri, anak, masakerja, 5000000, jmlhAnak);  NonPNS.setJamKerja(jamkerja);  System.out.println(""); System.out.println("==============================================");  System.out.println(" DATA GAJI KARYAWAN "); System.out.println("==============================================");  System.out.println("ID : " + NonPNS.getID() + "" +NonPNS.Counting());  System.out.println("Nama : " + NonPNS.getNama());  System.out.println("Jabatan : " + jabatan);  System.out.println("--------------------------------------------");  System.out.println("Gaji : Rp" + 5000000);  System.out.println("Gaji lembur :Rp" + NonPNS.getGajiLembur());  System.out.println("--------------------------------------------+");  System.out.println("Total Gaji :Rp " + NonPNS.getGaji());  System.out.println("--------------------------------------------");  } if (jabatan.equalsIgnoreCase("PNS")) {  PegawaiTetap PNS = new PegawaiTetap(nama, "P01", jabatan, istri, anak, masakerja, 5000000, jmlhAnak);  PNS.tunjanganDanBonus(masakerja);  System.out.println(""); System.out.println("==============================================");  System.out.println(" DATA GAJI KARYAWAN "); System.out.println("==============================================");  System.out.println("ID : " + PNS.getID() + "" + PNS.Counting());  System.out.println("Nama : " + PNS.getNama());  System.out.println("Jabatan : " + jabatan);  System.out.println("--------------------------------------------");  System.out.println("Gaji : Rp" + 5000000);  System.out.println("Tunjangan :Rp " + PNS.getTunjangan());  System.out.println("Bonus :Rp " + PNS.getBonus());  System.out.println("--------------------------------------------+");  System.out.println("Total Gaji : Rp" + PNS.getGaji());  System.out.println("--------------------------------------------");  } if (jabatan.equalsIgnoreCase("manajer")) {  manajer boss = new manajer(nama, "M01", jabatan, istri, anak, masakerja, 5000000, jmlhAnak);  boss.tunjanganDanBonus(masakerja);  System.out.println(""); System.out.println("==============================================");  System.out.println(" DATA GAJI KARYAWAN "); System.out.println("==============================================");  System.out.println("ID : " + boss.getID() + "" + boss.Counting());  System.out.println("Nama : " + boss.getNama());  System.out.println("Jabatan : " + jabatan);  System.out.println("--------------------------------------------"); System.out.println("Gaji : Rp " + 5000000);  System.out.println("Tunjangan :Rp " + boss.getTunjangan());  System.out.println("Tunjangan Jabatan:Rp "+boss.getTunjanganJabatan());  System.out.println("Bonus : Rp" + boss.getBonus());  System.out.println("--------------------------------------------+");  System.out.println("Total Gaji :Rp " + boss.getGaji());  System.out.println("--------------------------------------------"); }System.out.println("===========================================");  System.out.println("Apakah anda ingin menginputkan lagi? ");  System.out.println("1. Ya");  System.out.println("2. Tidak");  System.out.print("Masukkan pilihan: ");  int pil = input.nextInt();  if (pil == 1) {  ulang = true;  } else if (pil == 2) { System.out.println("==============TERIMAKASIH==============");  break;  } else {  System.out.println("Maaf pilihan yg diinputkan salah");  break;  }} while (ulang == true);  }} |

1. **PEMBAHASAN**

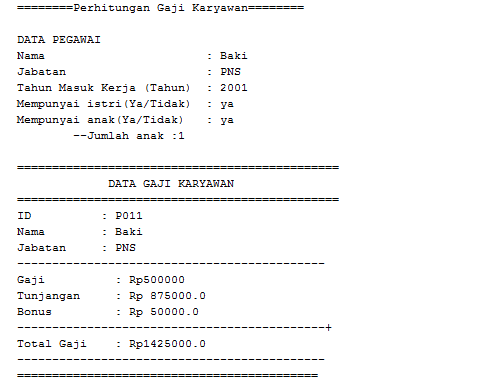
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71 | Deklarasi kelas dan method main, mengetikkan “public class employee {“  Mendeklarasikan variable nama dengan tipe data string  Mendeklarasikan variable idKerja dengan tipe data string  Mendeklarasikan variable jabatan dengan tipe data string  Mendeklarasikan variable istri dengan tipe data string  Mendeklarasikan varibale anak dengan tipe data string  Mendeklarasikan variable jmlanak, ThnKerja, dan LamaKerja dengan tipe data integer  Mendeklarasikan variable tunjangan, tunjanganAnak, tunjanganIstri, bonus dan gaji dengan tipe data double  Mendeklarasikan variable count sama dengan nol dengan tipe data static integer  Membuat konstruktor dengan parameter String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak  Nyatakan object dengan menggunakan kata “this”, seperti “this.nama=name;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.idKerja = id;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.jabatan=Jabatan;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.istri = istri;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.anak=anak;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.ThnKerja=ThnKerja;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.gaji=gaji;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.jmlanak=jmlhAnak;”  Membuat method getNama dengan tipe data string  Pengembalian nilai ke variable nama;  Membuat method getTunjangan dengan tipe data double  Pengembalian nilai ke variable tunjangan=tunjanganAnak+tunjanganIstri  Membuat method getBonus dengan tipe data double  Pengembalian nilai ke variable bonus  Membuat method getGaji dengan tipe data double  Pengembalian nilai ke variable gaji+bonus+tunjangan  Membuat method getId dengan tipe data String  Pengembalian nilai ke variable idKerja  Membuat method Counting dengan tipe data static integer  Nilai variabel count akan bertambah satu setiap method Count dipanggil  Pengembalian nilai ke variable count  Membuat method LamaKerja dengan tipe data integer dan parameter int ThnKerja  Penghitungan LamaKerja dengan cara 2015 dikurang tahun masuk pekerja  Pengembalian nilai 2015 dikurangi nilai variable tahun masuk pekerja  Membuat method tunjanganDanBonus bertipe void dan berparameter int ThnKerja  Menggunakan pernyataan if, jika lama kerja kurang dari sama dengan lima  Maka bonusnya samadengan nol  Maka tunjangan samadengan nol  Menggunakan pernyataan if untuk membandingkan objek istri dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Maka tunjanganIstri sama dengan 0.1 \* gaji  Dan jika tidak  Maka tunjanganIstri sama dengan 0  Menggunakan pernyataan if untuk membandingkan objek anak dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Jika jmlanak lebih dari tiga  Maka tunjanganAnak sama dengan (0.15 \* 3) \* gaji  Jika kurang dari tiga  Maka tunjanganAnak sama dengan (0.15 \* jmlanak) \* gaji  Menggunakan pernyataan if, jika lama kerja lebih dari sama dengan enam dan lama kerja kurang dari sama dengan sepuluh  Maka bonusnya samadengan gaji pokok\*0.05  Maka tunjangan samadengan nol  Menggunakan pernyataan if untuk membandingkan objek istri dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Maka tunjanganIstri sama dengan 0.1 \* gaji  Dan jika tidak  Maka tunjanganIstri sama dengan 0  Menggunakan pernyataan if untuk membandingkan objek anak dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Jika jmlanak lebih dari tiga  Maka tunjanganAnak sama dengan (0.15 \* 3) \* gaji  Jika kurang dari tiga  Maka tunjanganAnak sama dengan (0.15 \* jmlanak) \* gaji  Menggunakan pernyataan if, jika lama kerja lebih dari sepuluh  Maka bonus sama dengan gaji \* 0.1  Dan tunjangan sama dengan gaji \* 0.1  Menggunakan pernyataan if untuk membandingkan objek istri dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Maka tunjanganIstri sama dengan ((0.1 \* gaji) + (0.1\*LamaKerja) \*gaji)  Dan jika tidak  Maka tunjanganIstri sama dengan 0  Menggunakan pernyataan if untuk membandingkan objek anak dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Jika jmlanak lebih dari tiga  Maka tunjanganAnak sama dengan (0.15 \* 3) \* gaji  Jika kurang dari tiga  Maka tunjanganAnak sama dengan (0.15 \* jmlanak) \* gaji |
| 1  2  3  4  5  6  7  8  9  10 | Deklarasi subkelas manager turunan dari superclass employee  Mendeklarasikan variable TunjanganJabatan dengan tipe data double  Membuat konstruktor dengan parameter String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak  Memanggil constructor milik superclass employee  Membuat method getTunjanganJabatan dengan tipe data double  Pengembalian nilai ke variable TunjanganJabatan sama dengan 0.1 \* (gaji + bonus + tunjangan)  Meng-override method dari super class employee  Membuat method getGaji dengan tipe data double  Mendeklarasikan variable gajiManager sama dengan getGaji milik superclass  Pengembalian nilai ke variable gajiManager ditambah getTunjanganJabatan() |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Deklarasi subkelas pegawaiTidakTetap turunan dari superclass employee  Mendeklarasikan variable GajiLembur dan JamKerja dengan tipe data integer  Membuat konstruktor dengan parameter String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak  Memanggil constructor milik superclass employee  Membuat method setJamKerja bertipe void dengan parameter int a  Variable JamKerja sama dengan a  Membuat method getGajiLembur dengan tipe data double  Menggunakan pernyataan if, jika JamKerja lebih dari sepuluh  Maka GajiLembur sama dengan (JamKerja – 10) \* 10000  Jika kurang dari sepuluh  Maka GajiLembur samadengan nol  Pengembalian nilai ke variable GajiLembur  Meng-override method dari super class employee  Membuat method getGaji dengan tipe data double  Pengembalian nilai ke variable gaji + method getGajiLembur() |
| 1  2  3  4  5  6 | Deklarasi subkelas pegawaiTetap turunan dari superclass employee  Membuat konstruktor dengan parameter String name, String id, String Jabatan, String istri, String anak, int ThnKerja, double gaji, int jmlhAnak  Memanggil constructor milik superclass employee  Meng-override method dari superclass employee  Membuat method getGaji dengan tipe data double  Mengembalikan nilai method getGaji milik superclass employee |
| 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  83  84  85  86  87  88  89  90  91 | Melakukan import dari package prakproglan, ketikkan “import java.util.Scanner;”  Deklarasi class dan method main dengan mengetik public class MainEmployee {  Kemudian mengetikkan “public static void main(String[]args){” sebagai fungsi main  Menampilkan kalimat “==Perhitungan Gaji Karyawan==”  Enter  Menampilkan kalimat “DATA PEGAWAI”  Deklarasi variable Scanner dan nama variable input ”Scanner input = new Scanner(System.in);” tujuan nya yaitu meminta kepada pengguna untuk memasukkan data yang kemudian diproses oleh program.  Mendeklarasikan variable jmlhAnak sama dengan nol dengan tipe data integer  Mendeklarasikan variable ulang yang bernilai false dengan tipe data boolean  Menggunakan pernyataan do-while  Ulang sama dengan false  User diminta meng-input nama pekerja  Membaca inputan dari user  User diminta meng-input jabatan pekerja  Membaca inputan dari user  User diminta meng-input tahun masuk pekerja  Membaca inputan dari user  User diminta meng-input apakah pekerja mempunyai istri atau tidak  Membaca inputan dari user  User diminta meng-input apakah pekerja mempunyai anak atau tidak  Membaca inputan dari user  Menggunakan pernyataan if untuk membandingkan objek anak dengan kata “ya” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  User diminta meng-input jumlah anak  Membaca inputan dari user  Menggunakan pernyataan if, untuk membandingkan objek jabatan.dengan kata “NonPNS ” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  User diminta meng-input total jam kerjanya  Membaca inputan dari user  Membuat objek NonPNS dari kelas pegawaiTidakTetap dengan parameter nama, no id, jabatan, istri, anak, masakerja, gaji, dan jumlah anak  Memanggil method setJamKerja dari class pegawaiTidakTetap  Enter  Menampilkan variable “=” sebagai pembatas kata  Menampilkan kalimat “ DATA GAJI KARYAWAN=  Menampilkan variable “=” sebagai pembatas kata  Menampilkan id pegawai NonPNS yang nomor serinya akan bertambah seiring dengan pertambahan inputan data (untuk jabatan yang sama)  Menampilkan nama hasil inputan  Menampilkan jabatan hasil inputan  Menampilkan “-“ sebagai pembatas kata  Menampilkan gaji nonPNS adalah 5000000  Menampilkan gaji lembur non pns  Menampilkan “-“ sebagai pembatas kata  Menampilkan total gaji non pns  Menampilkan “-“ sebagai pembatas kata  Menggunakan pernyataan if, untuk membandingkan objek jabatan.dengan kata “PNS ” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Membuat objek PNS dari kelas PegawaiTetap dengan parameter nama, no id, jabatan, istri, anak, masakerja, gaji, dan jumlah anak  Memanggil method tunjanganDanBonus dengan parameter masaKerja dari class PegawaiTetap  Enter  Menampilkan variable “=” sebagai pembatas kata  Menampilkan kalimat “ DATA GAJI KARYAWAN=  Menampilkan variable “=” sebagai pembatas kata  Menampilkan id PNS yang nomor serinya akan bertambah seiring dengan pertambahan inputan data (untuk jabatan yang sama)  Menampilkan nama hasil inputan  Menampilkan jabatan hasil inputan  Menampilkan “-“ sebagai pembatas kata  Menampilkan gaji PNS adalah 5000000  Menampilkan tunjangan para pns  Menampilkan bonus para pns  Menampilkan “-“ sebagai pembatas kata  Menampilkan total gaji pns  Menampilkan “-“ sebagai pembatas kata  Menggunakan pernyataan if, untuk membandingkan objek jabatan.dengan kata “manajer ” pada parameter fungsi. Penggunaan fungsi ini tidak melihat perbedaan antara huruf besar dan huruf kecil  Membuat objek boss dari kelas manajer dengan parameter nama, no id, jabatan, istri, anak, masakerja, gaji, dan jumlah anak  Memanggil method tunjanganDanBonus dengan parameter masaKerja dari class manajer  Enter  Menampilkan variable “=” sebagai pembatas kata  Menampilkan kalimat “ DATA GAJI KARYAWAN=  Menampilkan variable “=” sebagai pembatas kata  Menampilkan id manajer yang nomor serinya akan bertambah seiring dengan pertambahan inputan data (untuk jabatan yang sama)  Menampilkan nama hasil inputan  Menampilkan jabatan hasil inputan  Menampilkan “-“ sebagai pembatas kata  Menampilkan gaji manajer adalah 5000000  Menampilkan tunjangan manajer  Menampilkan tunjangan jabatan manejer  Manampilkan bonus manajer  Menampilkan “-“ sebagai pembatas kata  Menampilkan total gaji manajer  Menampilkan “-“ sebagai pembatas kata  Menampilkan “=” sebagai pembatas kata  Menampilkan pertanyaan “apakah anda ingin menginputkan lagi ?”  Menampilkan jawaban “ya”  Menampilkan jawaban “tidak”  User diminta meng-input masukan pilihan  Membaca inputan dari user  Menggunakan pernyataan if, jika pil sama dengan satu  Maka ulang sama dengan true  Dan jika pil sama dengan dua  Maka akan menampilkan kata “=terimakasih=”  Menghentikan proses looping  Dan jika memasukan pilihan yang salah  Maka akan menampilkan kata “maaf pilihan yang diinputkan salah”  Menghentikan proses looping  Pernyataan while bahwa ulang sama dengan true |

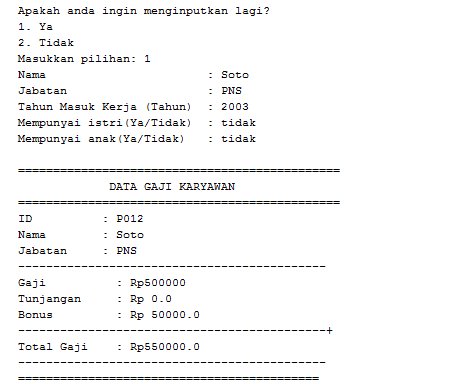
1. **SCREENSHOT PROGRAM**

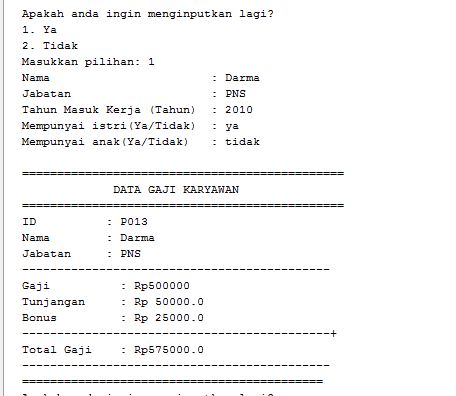
****

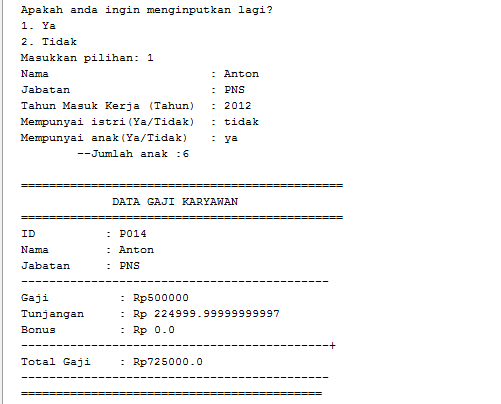
**PNS:**

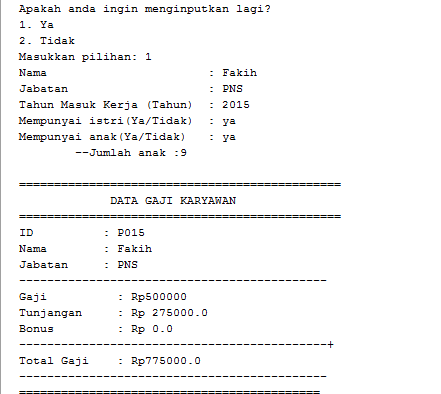
****

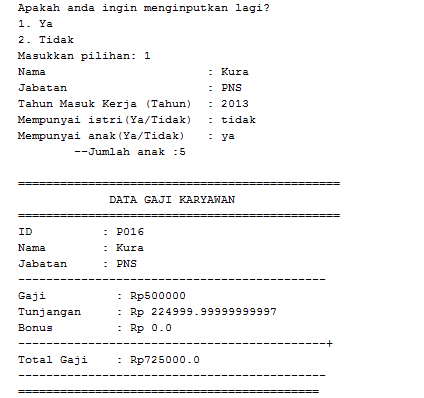
****

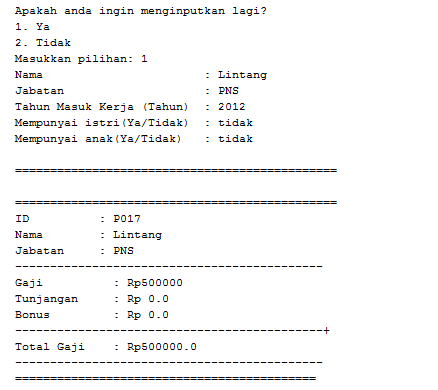
****

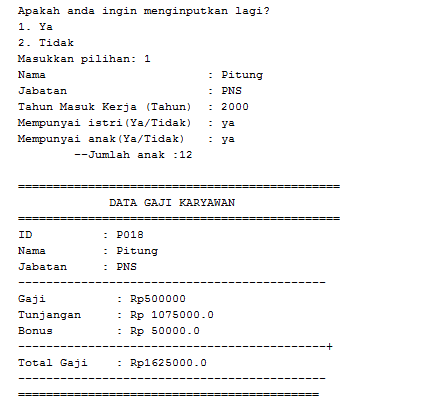
****

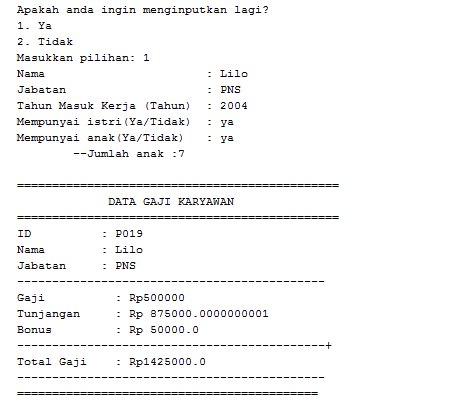
****

****

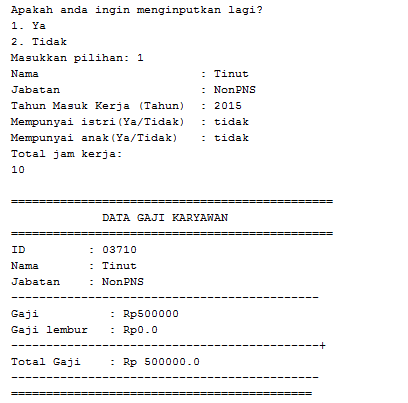
****

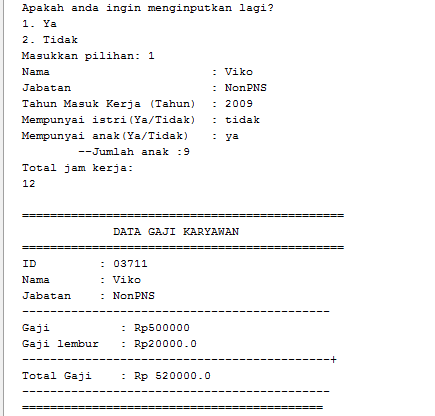
****

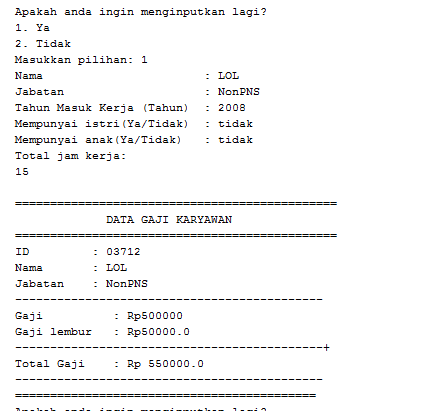
****

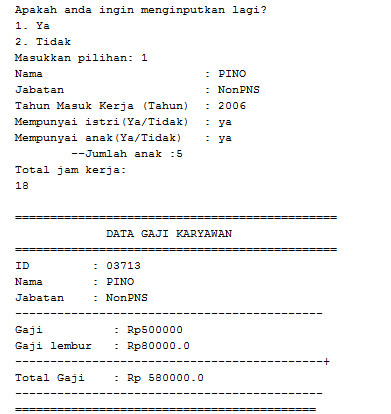
****

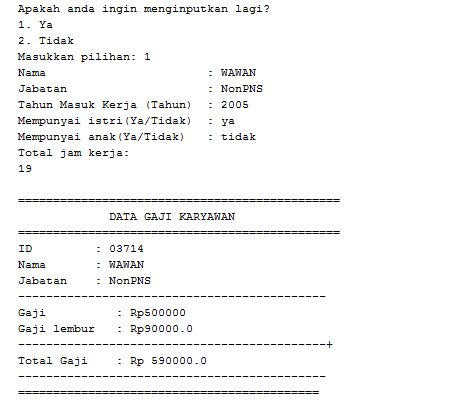
**NonPNS:**

****

****

****

****

****

1. **PRAKTIKUM**

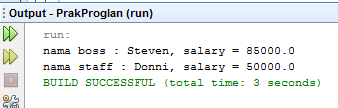
**Data dan Analisis hasil percobaan**

Pertanyaan

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

**Jawab :** saya sudah melakukan percobaan dan tidak menemukan kesalahan

**Output :**

****

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

**Jawab :**

KonstruktorSubClass ( parameter1, parameter2, ...){

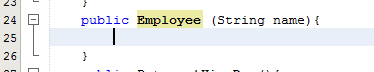
super ( parameter1, parameter2, ...) ;

}

Dengan cara menggunakan perintah super(parameter). Hal itu perlu dilakukan karena dengan menggunakan perintah super() kita dapat memanggil parameter di konstruktor yang berada di super class Employee

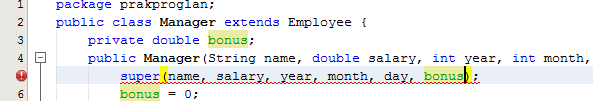
1. Tambahkan constructor pada class Employee dengan parameter *String name*! amati perubahan apa yang terjadi, jelaskan jawaban anda!

**Jawab :**

****

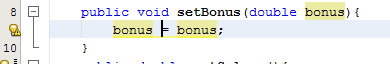
Tidak terjadi perubahan apa-apa, karena hanya menambahkan konstruktor baru

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

**Jawab **Akan terjadi error, karena class manager membaca construktor pada class employee, sedangkan pada class employee dalam pembuatan konstruktornya variable bonus tidak dimasukkan ke dalam parameter sehingga variable bonus tidak dapat dipanggil pada baris ke lima.

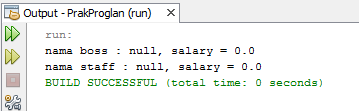
1. Untuk apa digunakan keyword this pada class manager dan employee? Hapus keyword this dan amati apa yang terjadi?

**Jawab :**

****

**D:\Sistem Informasi\Semester 2\PEMROGRAMAN LANJUT\prak inheritance\Capture3.PNG**

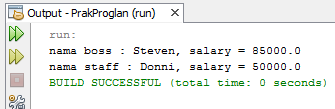
Tidak akan terjadi error, tetapi ouptunya tidak mengeluarkan hasil dan hanya bernilai null

****

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:**

D:\Sistem Informasi\Semester 2\PEMROGRAMAN LANJUT\prak inheritance\Capture7.PNG



Tidak terjadi perubahan apa-apa dalam program dan outputnya tetap sama, dikarenakan pada class Employee terdapat lebih dari satu konstruktor dan yang terpanggil di main class ialah konstruktor yang lainnya.

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 :**

Access modifier protected menspesifikasikan atau menentukan anggota kelas (class member) yang hanya bisa diakses oleh method-methodyang ada di dalam kelas yang sama dan juga dapat diakses oleh subclass (kelas turunan) dari kelas tersebut.Superclass tidak bisa menggunakan private karena hak akses private hanya diberikan pada kelasnya sendiri sementara hak akses public mencakup semua kelas walaupun yang bukan merupakan kelas turunan

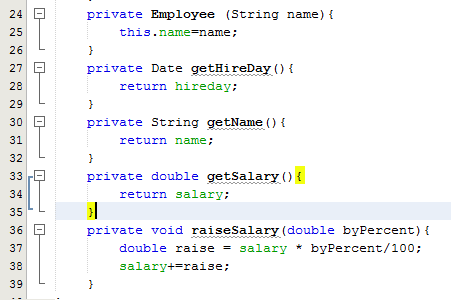
1. Ubahlah acces modifier method pada kelas Person menjadi :
2. Private
3. Protected

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

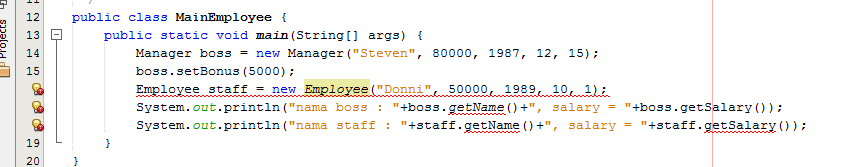
**Jawab :**

1. **Private**

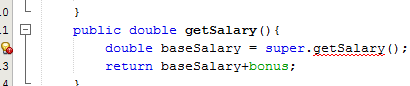
**Pada class Employee**

****

**Pada mainclass Employee**

****

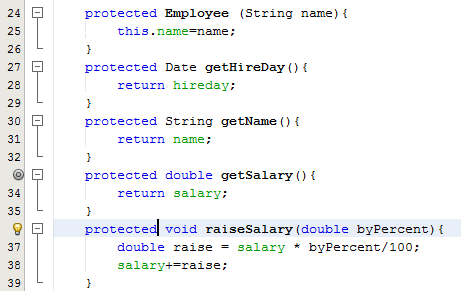
**Pada subclass Manager**

****

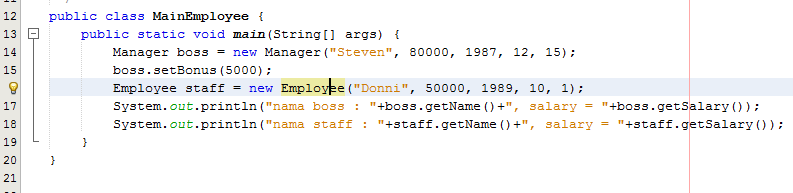
Ketika di class Employe hak akses method diubah menjadi private dari public, hanya akan bisa di akses dari kelas Employee itu saja dan akan terjadi error di main class Employee dan subclass Manager, karena mereka tidak mempunyai akses untuk mengakses method yang ada pada class Employee

1. **protected**

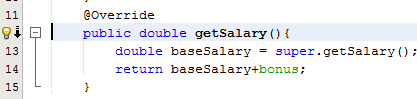
**pada class Employee**

****

**Pada mainclass Employee**

****

**Pada subclass Manager**

****

Ketika hak akses diubah menjadi protected dari public tidak akan terjadi error di mana-mana. Karena hak akses protected bisa di akses di kelas yang berbeda dan dapat di akses oleh subclass.

1. **KESIMPULAN**

Inheritance adalah turunan atau pewarisan suatu class ke class baru dengan mewakili semua function yang ada di class utama (parent class). Inheritance sering sekali digunakan di dalam promgraman berorientasi objek atau OOP. Dengan kata lain, class utama (parent class) dengan classes turunannya berhubungan satu sama lain.

Kita baru perlu menerapkan konsep inheritance pada saat kita jumpai ada suatu class yang dapat diperluas dari class lain.

Misal terdapat class Pegawai :

public class Pegawai {public String nama;public double gaji;}

Misal terdapat class Manager :

public class Manajer {public String nama;public double gaji;public String departemen;}

Dari 2 buah class diatas, kita lihat class Manajer mempunyai data member yang identik sama dengan class Pegawai, hanya saja ada tambahan data member departemen. Sebenarnya yang terjadi disana adalah class Manajer merupakan perluasan dari class Pegawai dengan tambahan data member departemen. Disini perlu memakai konsep inheritance, sehingga class Manajer dapat kita tuliskan seperti berikut :  
*public class Manajer extends Pegawai {public String departemen;}*

**Super ->** Keyword ini digunakan untuk memanggil konstruktor dari superclass atau menjadi variabel yang mengacu pada superclass. Kata kunci super dipakai untuk merujuk pada member dari parent class, sebagaimana kata kunci this yang dipakai untuk merujuk pada member dari class itu sendiri.

● **The Unified Modeling Language (UML)** adalah bahasa standart untuk melakukan spesifikasi, visualisasi, konstruksi, dan dokumentasi dari komponen-komponen perangkat lunak, dan digunakan untuk pemodelan bisnis.

• UML menggunakan notasi grafis untuk menyatakan suatu desain

• Pemodelan dengan UML berarti menggambarkan yang ada dalam dunia nyata ke dalam bentuk yang dapat dipahami dengan menggunakan notasi standart

● Masing-masing diagram UML didesain untuk menunjukkan satu sisi dari bermacam-macam sudut pandang (perspektif) dan terdiri dari tingkat abstraksi yang berbeda

● **Diagram UML** adalah sebuah Bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun dan mendokumentasikan dari sebuah system pengembangan software OOP (Object Oriented Programming).

