1. **DEFINISI MASALAH**
2. Setiap alat musik memiliki nama, dan cara bermain masing-masing, seperti misalnya gitar yaitu di petik, suling di tiup, biola di gesek, dan lain sebagainya. Selain itu alat musik juga memiliki nada-nada, yang dasar Do nya di tentukan berdasarkan tangga nadanya, sehingga terdapat juga fungsi untuk menentukan nada Do berada pada nada yang mana, misalnya Do = C. Setiap alat musik juga memiliki bahan sendiri untuk membuatnya, seperti gitar dari kayu, flute dari logam, dan lain sebagainya.

Buatlah sebuah interface alat musik yang di implementasi kan oleh kelas abstrak induk alat musik petik, alat musik tiup, alat musik tabuh, dan lain sebagainya (pilih

2) dimana setiap induk memiliki minimum 1 kelas anak (misal: suling extends alat musik tiup). Tentukan sendiri alat musik apa yang menurunkan kelas abstrak induknya. buatlah tampilan menu dan inputan sekreatif dan menarik.

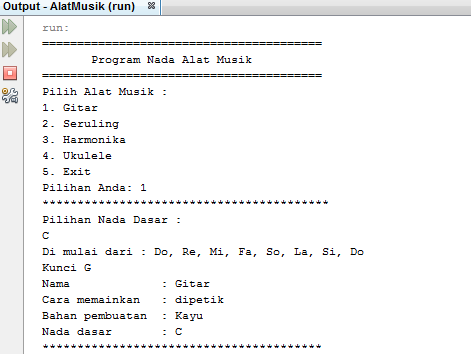
1. **SOURCE CODE**

|  |  |
| --- | --- |
| 1  2  3  4 | public interface AlatMusik {  public String CaraDimainkan();  public String BahanBuatan();  public String NadaDasar();  } |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Class MusikPetik  import java.util.Scanner;  public abstract class MusikPetik implements AlatMusik {  Scanner input = new Scanner (System.in);  private String namaAlat;  public MusikPetik(String namaAlat) {  this.namaAlat = namaAlat;  }  public String getNamaAlat() {  return namaAlat;  }  @Override  public String CaraDimainkan() {  return "dipetik";  }  @Override  public abstract String NadaDasar();  @Override  public abstract String BahanBuatan();  } |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class Gitar  public class Gitar extends MusikPetik implements AlatMusik {  public Gitar(String namaAlat) {  super(namaAlat);  }  @Override  public String NadaDasar() {  System.out.println("Pilihan Nada Dasar : ");  String Nada = input.nextLine();  if (Nada.equalsIgnoreCase("C")){  System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");  System.out.println("Kunci G");  }  if (Nada.equalsIgnoreCase("A")){  System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La");  System.out.println("Kunci F");  }  return Nada;  }  @Override  public String BahanBuatan() {  return "Kayu";  }} |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  15  16  17  18  19 | Class Ukulele  public class Ukulele extends MusikPetik implements AlatMusik {  public Ukulele(String namaAlat) {  super(namaAlat);  }  @Override  public String NadaDasar() {  System.out.println("Pilihan Nada Dasar : ");  String Nada = input.nextLine();  if (Nada.equalsIgnoreCase("C")){  System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");  System.out.println("Kunci G");  }  if (Nada.equalsIgnoreCase("A")){  System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La");  System.out.println("Kunci F");  }  return Nada;  }  @Override  public String BahanBuatan() {  return "Kayu";  }} |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Class MusikTiup  import java.util.Scanner;  public abstract class MusikTiup implements AlatMusik {  Scanner input = new Scanner (System.in);  private String namaAlat;  public MusikTiup(String namaAlat) {  this.namaAlat = namaAlat;  }  public String getNamaAlat() {  return namaAlat;  }  @Override  public String CaraDimainkan() {  return "ditiup";  }  @Override  public abstract String NadaDasar();  @Override  public abstract String BahanBuatan();  } |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class Suling  public class Suling extends MusikTiup implements AlatMusik {  public Suling(String namaAlat) {  super(namaAlat);  }  @Override  public String NadaDasar() {  System.out.println("Pilihan Nada Dasar : ");  String Nada = input.nextLine();  if (Nada.equalsIgnoreCase("C")){  System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");  System.out.println("Kunci G");  }  if (Nada.equalsIgnoreCase("A")){  System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La");  System.out.println("Kunci F");  }  return Nada;  }  @Override  public String BahanBuatan() {  return "Bambu";  }  } |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class Harmonika  public class Harmonika extends MusikTiup implements AlatMusik{  public Harmonika(String namaAlatMusik) {  super(namaAlatMusik);  }  @Override  public String NadaDasar() {  System.out.println("Pilihan Nada Dasar : ");  String Nada = input.nextLine();  if (Nada.equalsIgnoreCase("C")){  System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");  System.out.println("Kunci G");  }  if (Nada.equalsIgnoreCase("A")){  System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La");  System.out.println("Kunci F");  }  return Nada;  }  @Override  public String BahanBuatan() {  return "Alumunium";  } |
| 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 | Class MainAlatMusik  import java.util.Scanner;  public class MainAlatMusik {  public static void main(String[] args) {  int pil, i;  MusikPetik petik;  MusikTiup ditiup;  Scanner input = new Scanner(System.in);  do {  System.out.println("========================================");  System.out.println(" Program Nada Alat Musik "); System.out.println("========================================");  System.out.println("Pilih Alat Musik :"  + "\n1. Gitar"  + "\n2. Seruling"  + "\n3. Harmonika"  + "\n4. Ukulele"  + "\n5. Exit");  System.out.print("Pilihan Anda: ");  pil = input.nextInt(); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  switch (pil) {  case 1:  petik = new Gitar("Gitar");  System.out.println("Nama : " + petik.getNamaAlat()  + "\nCara memainkan : " + petik.CaraDimainkan()  + "\nBahan pembuatan : " + petik.BahanBuatan()  + "\nNada dasar : " + petik.NadaDasar());  break;  case 2:  ditiup = new Suling("Suling"); System.out.println("Nama : " + ditiup.getNamaAlat()  + "\nCara memainkan : " + ditiup.CaraDimainkan()  + "\nBahan pembuatan : " + ditiup.BahanBuatan()  + "\nNada dasar : " + ditiup.NadaDasar());  break;  case 3:  ditiup = new Harmonika("Harmonica");  System.out.println("Nama : " + ditiup.getNamaAlat()  + "\nCara memainkan : " + ditiup.CaraDimainkan()  + "\nBahan pembuatan : " + ditiup.BahanBuatan()  + "\nNada dasar : " + ditiup.NadaDasar());  break;  case 4 :  petik = new Ukulele("Ukulele");  System.out.println("Nama : " + petik.getNamaAlat()  + "\nCara memainkan : " + petik.CaraDimainkan()  + "\nBahan pembuatan : " + petik.BahanBuatan()  + "\nNada dasar : " + petik.NadaDasar());  break;  case 5:  System.exit(0);  break;  default:  System.out.println("Maaf pilihan tersebut tidak ada");  break;  }  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.print("Ulangi Kembali? (1 = Ya, 0=Tidak)");  System.out.println("\nPilihan (1/0):");  i = input.nextInt();  if (i==0){  System.out.println("ANDA TIDAK MEMILIH APAPUN");  }  } while (i != 0);  } } |

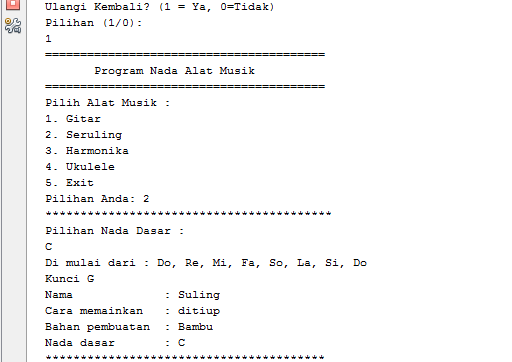
1. **PEMBAHASAN**

|  |  |
| --- | --- |
| 1  2  3  4 | Mendeklarasikan interface dengan nama AlatMusik  Membuat method CaraDimainkan() dengan tipe data string  Membuat method BahanBuatan() dengan tipe data string  Membuat method NadaDasar() dengan tipe data string |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Class MusikPetik  Melakukan import dari package proglanInterface, ketikkan “import java.util.Scanner;”  Mendeklarasikan kelas abstract MusikPetik yang di implementasikan dari interface AlatMusik  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 namaAlat dengan tipe data string  Membuat konstruktor MusikPetik dengan parameter namaAlat  this.namaAlat sama dengan parameter namaAlat, this untuk menunjuk komponen milik class itu sendiri.  Membuat method getNamaAlat dengan tipe data string  Mengembalikan nilai ke variable namaAlat  Meng-Ovverride method abstract CaraDimainkan dari method yang berada di interface AlatMusik  Membuat method CaraDimainkan dengan tipe data string  Mengembalikan nilai objek “di petik”  Meng-Ovverride method abstract NadaDasar dari method yang berada di interface AlatMusik  Membuat method abstract NadaDasar dengan tipe data string  Meng-Ovverride method BahanBuatan dari method yang berada di interface AlatMusik  Membuat method abstract Buatan dengan tipe data string |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class GItar  Mendeklarasikan sub kelas Gitar yang diturunkan dari kelas MusikPetik yang di implementasikan dari interface AlatMusik  Membuat konstruktor Gitar dengan parameter String namaAlat  Memanggil variable milik kelas abstract MusikPetik  Meng-Ovverride method NadaDasar dari method yang berada di kelas MusikPetik  Membuat method NadaDasar dengan tipe data string  User diminta memasukan pilihan nada dasar  Membaca inputan dari user  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata C  Maka menampilkan “System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");”  Dan menampilkan kata “Kunci G”  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata A  Maka menampilkan “System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La);"  Dan menampilkan kata “Kunci f”  Mengembalikan nilai ke variable Nada  Meng-Ovverride method BahanBuatan dari method yang berada kelas MusikPetik  Membuat method NadaDasar dengan tipe data string  Mengembalikan nilai objek “Kayu” |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class Ukulele  Mendeklarasikan sub kelas Ukulele yang diturunkan dari kelas MusikPetik yang di implementasikan dari interface AlatMusik  Membuat konstruktor Ukulele dengan parameter String namaAlat  Memanggil variable milik kelas abstract MusikPetik  Meng-Ovverride method NadaDasar dari method yang berada di kelas MusikPetik  Membuat method NadaDasar dengan tipe data string  User diminta memasukan pilihan nada dasar  Membaca inputan dari user  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata C  Maka menampilkan “System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");”  Dan menampilkan kata “Kunci G”  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata A  Maka menampilkan “System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La);"  Dan menampilkan kata “Kunci f”  Mengembalikan nilai ke variable Nada  Meng-Ovverride method BahanBuatan dari method yang berada kelas MusikPetik  Membuat method NadaDasar dengan tipe data string  Mengembalikan nilai objek “Kayu” |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Class MusikTiup  Melakukan import dari package proglanInterface, ketikkan “import java.util.Scanner;”  Mendeklarasikan kelas abstract MusikTiup yang di implementasikan dari interface AlatMusik  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 namaAlat dengan tipe data string  Membuat konstruktor MusikPetik dengan parameter namaAlat  this.namaAlat sama dengan parameter namaAlat, this untuk menunjuk komponen milik class itu sendiri.  Membuat method getNamaAlat dengan tipe data string  Mengembalikan nilai ke variable namaAlat  Meng-Ovverride method abstract CaraDimainkan dari method yang berada di interface AlatMusik  Membuat method CaraDimainkan dengan tipe data string  Mengembalikan nilai objek “di tiup”  Meng-Ovverride method abstract NadaDasar dari method yang berada di interface AlatMusik  Membuat method abstract NadaDasar dengan tipe data string  Meng-Ovverride method BahanBuatan dari method yang berada di interface AlatMusik  Membuat method abstract Buatan dengan tipe data string |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class Suling  Mendeklarasikan sub kelas Suling yang diturunkan dari kelas MusikTiup yang di implementasikan dari interface AlatMusik  Membuat konstruktor Suling dengan parameter String namaAlat  Memanggil variable milik kelas abstract MusikTiup  Meng-Ovverride method NadaDasar dari method yang berada di kelas MusikTiup  Membuat method NadaDasar dengan tipe data string  User diminta memasukan pilihan nada dasar  Membaca inputan dari user  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata C  Maka menampilkan “System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");”  Dan menampilkan kata “Kunci G”  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata A  Maka menampilkan “System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La);"  Dan menampilkan kata “Kunci f”  Mengembalikan nilai ke variable Nada  Meng-Ovverride method BahanBuatan dari method yang berada kelas MusikTiup  Membuat method NadaDasar dengan tipe data string  Mengembalikan nilai objek “Bambu” |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Class Harmonika  Mendeklarasikan sub kelas Harmonika yang diturunkan dari kelas MusikTiup yang di implementasikan dari interface AlatMusik  Membuat konstruktor Harmonika dengan parameter String namaAlat  Memanggil variable milik kelas abstract MusikTiup  Meng-Ovverride method NadaDasar dari method yang berada di kelas MusikTiup  Membuat method NadaDasar dengan tipe data string  User diminta memasukan pilihan nada dasar  Membaca inputan dari user  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata C  Maka menampilkan “System.out.println("Di mulai dari : Do, Re, Mi, Fa, So, La, Si, Do");”  Dan menampilkan kata “Kunci G”  Menggunakan pernyatan if jika kata nada dibandingkan dengan kata A  Maka menampilkan “System.out.println("Di mulai dari : La, Si, Do, Re, Mi, Fa, So, La);"  Dan menampilkan kata “Kunci f”  Mengembalikan nilai ke variable Nada  Meng-Ovverride method BahanBuatan dari method yang berada kelas MusikTiup  Membuat method NadaDasar dengan tipe data string  Mengembalikan nilai objek “Aluminium” |
| 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 | Class MainAlatMusik  Deklarasi import java.util.Scanner untuk memasukkan fungsi Scanner pada program  Mendeklarasikan kelas MainAlatMusik  Kemudian mengetikkan “public static void main(String[] args) {” sebagai fungsi main  Deklarasi variabel pil dan i bertipe data integer  Menginstansiasi objek petik dari class MusikPetik  Menginstansiasi objek ditiup dari class MusikTiup  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  Perintah perulangan menggunakan syntax do  Perintah untuk mencetak keluaran “=========================”  Perintah untuk mencetak keluaran “Program Nada Alat Musik”  Perintah untuk mencetak keluaran “==========================”  Perintah untuk mencetak keluaran “Pilih Alat Musik:”  Perintah untuk mencetak keluaran “1. Gitar”  Perintah untuk mencetak keluaran “2. Seruling”  Perintah untuk mencetak keluaran “3. Harmonika”  Perintah untuk mencetak keluaran “4. Exit”  Perintah untuk mencetak keluaran “Pilihan Anda: ”  Kode untuk meminta user memasukkan variabel pil dari output yang akan di simpan  Perintah untuk mencetak “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”  Perintah pengkondisian switch dimana memanggil variabel pil  Jika user memberikan nilai sebesar 1 pada variabel pil maka case 1 akan dijalankan.  Menginstansiasi objek dengan variabel petik dari class Gitar  Perintah untuk mencetak “Nama “ dengan objek petik yang memanggil method getNamaAlat  Perintah untuk mencetak “Cara Memainkan “dengan objek petik yang memanggil method CaraDimainkan  Perintah untuk mencetak “Bahan Pembuatan “ dengan objek petik yang memanggil method BahanBuatan  Perintah untuk mencetak “Nada Dasar “ dan ditempelkan dengan objek petik yang memanggil method NadaDasar  Perintah break untuk menghentikan proses  Jika user memberikan nilai sebesar 2 pada variabel pil maka case 2 akan dijalankan.  Menginstansiasi objek dengan variabel ditiup dari class Suling  Perintah untuk mencetak “Nama “ dengan objek ditiup yang memanggil method getNamaAlat  Perintah untuk mencetak “Cara Memainkan “dengan objek ditiup yang memanggil method CaraDimainkan  Perintah untuk mencetak “Bahan Pembuatan “dengan objek ditiup yang memanggil method BahanBuatan  Perintah untuk mencetak “Nada Dasar “dengan objek ditiup yang memanggil method NadaDasar  Perintah break untuk menghentikan proses  Jika user memberikan nilai sebesar 3 pada variabel pil maka case 3 akan dijalankan.  Menginstansiasi objek dengan variabel ditiup dari class Harmonica  Perintah untuk mencetak “Nama “dengan objek ditiup yang memanggil method getNamaAlat  Perintah untuk mencetak “Cara Memainkan “dengan objek ditiup yang memanggil method CaraDimainkan  Perintah untuk mencetak “Bahan Pembuatan“ dengan objek ditiup yang memanggil method BahanBuatan  Perintah untuk mencetak “Nada Dasar “ dengan objek ditiup yang memanggil method NadaDasar  Perintah break untuk menghentikan proses  Jika user memberikan nilai sebesar 4 pada variabel pil maka case 4 akan dijalankan.  Menginstansiasi objek dengan variabel petik dari class Ukulele  Perintah untuk mencetak “Nama “dengan objek petik yang memanggil method getNamaAlat  Perintah untuk mencetak “Cara Memainkan “dengan objek petik yang memanggil method CaraDimainkan  Perintah untuk mencetak “Bahan Pembuatan“ dengan objek petik yang memanggil method BahanBuatan  Perintah untuk mencetak “Nada Dasar “ dengan objek petik yang memanggil method NadaDasar  Perintah break untuk menghentikan proses  Jika user memberikan nilai sebesar 5 pada variabel pil maka case 5 akan dijalankan.  Memberhentikan program dengan deklarasi system.exit (0)  Perintah break untuk menghentikan proses  Case default apabila user menginputkan semua case selain case 1, 2 dan 3 maka akan menjalankan program yang berada dalam default.  Perintah untuk mencetak keluaran “ANDA TIDAK MEMILIH APAPUN” bila user menginputkan yang tidak sesuai dengan yang diminta  Perintah break untuk menghentikan proses  Pentintah untuk mencetak keluaran (“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*“)  Perintah untuk mencetak keluaran apakah user ingin ulangi memilih program kembali?  Perintah untuk mencetak keluaran “Pilihan (1 atau 0) ”  Kode untuk meminta user memasukkan variabel i dari output yang akan di simpan  Perintah percabangan if, jika i=0 atau jika user memilih option tidak  Perintah untuk mencetak keluaran “tidak ada alat music yang dipilih”  Perintah while, jika i tidak sama dengan 0 |

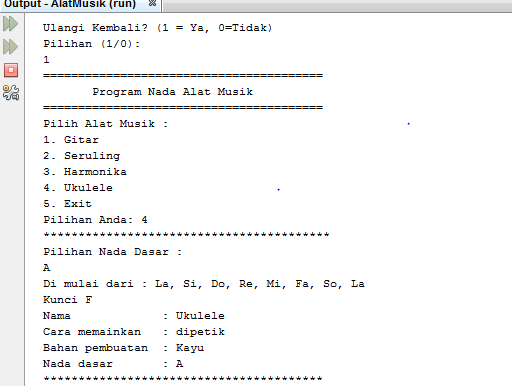
1. **SCREENSHOT PROGRAM**

****

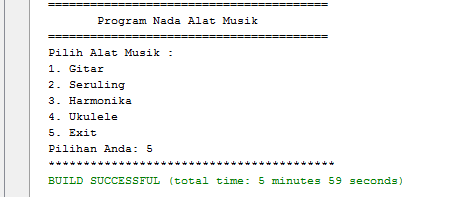
Gambar 1. Program menampilkan daftar alat musik, lalu user diperintahkan untuk memilih alat musik yang mana. Setelah itu user menentukan nada dasarnya, lalu muncul tangga nada alat musik tersebut. Kemudian program menampilkan nama alat musik, cara memainkannya, bahan pembuatannya dan nada dasarnya.



Gambar 2. User ditanya apakah ingin mengulang atau tidak, jika ya maka input 1 jika tidak maka input 0. Jika ya maka akan menampilkan daftar alat musik, lalu user diperintahkan untuk memilih alat musik yang mana. Setelah itu user menentukan nada dasarnya, lalu muncul tangga nada alat musik tersebut. Kemudian program menampilkan nama alat musik, cara memainkannya, bahan pembuatannya dan nada dasarnya.



Gambar 3. User ditanya apakah ingin mengulang atau tidak, jika ya maka input 1 jika tidak maka input 0. Jika ya maka akan menampilkan daftar alat musik, lalu user diperintahkan untuk memilih alat musik yang mana. Setelah itu user menentukan nada dasarnya, lalu muncul tangga nada alat musik tersebut. Kemudian program menampilkan nama alat musik, cara memainkannya, bahan pembuatannya dan nada dasarnya.



Gambar 4. User ditanya apakah ingin mengulang atau tidak, jika ya maka input 1 jika tidak maka input 0. Jika ya maka akan menampilkan daftar alat musik dan dipilihan terakhir ada pilihan untuk keluar program.

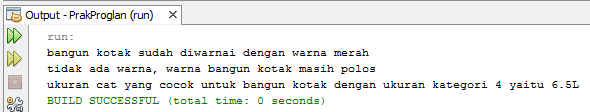
1. **PRAKTIKUM**

**Pertanyaan**

|  |  |
| --- | --- |
| 1  2  3 | public interface Cororable {  public void howToColor();  } |
| 1  2  3 | public interface Comparable {  public void compareTo(Object obj);  } |
| 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 | public class Rectangle implements Cororable, Comparable {  private String warna;  private int kategori;  public Rectangle() {  }  public Rectangle(String warna) {  this.warna = warna;  }  public void howToColor() {  if(this.warna == null){  System.out.println("tidak ada warna, warna bangun kotak masih polos");  }  else{  System.out.println("bangun kotak sudah diwarnai dengan warna "+this.warna);  }  }  public void compareTo(Object obj) {  this.kategori = (int) obj;  if(this.kategori == 0){  System.out.println("ukuran cat yang cocok untuk bangun kotak dengan ukuran kategori " +this.kategori+" yaitu 2.5L" );  }  else{  System.out.println("ukuran cat yang cocok untuk bangun kotak dengan ukuran kategori " +this.kategori+" yaitu 6.5L" );  }  }  } |
| 1  2  3  4  5  6  7  8  9  10 | public class MainClass {  public static void main(String[] args) {  Rectangle kotak1 = new Rectangle("merah");  Rectangle kotak2= new Rectangle();  Rectangle kotak3 = new Rectangle();  kotak1.howToColor();  kotak2.howToColor();  kotak3.compareTo(4);  }  } |

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan serta jelaskan!

**Jawab :**Setelah melakukan percobaan di atas saya tidak menemukan kesalahan apa-apa



1. Apakah Interface bisa diinstansiasi menjadi sebuah objek? Jelaskan alasannya!

**Jawab :**Tidak bisa, karena interface itu isinya method abstrack semua. karena method abstract tidak bisa dijadikan objek

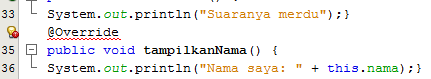
1. Apakah suatu class dapat mengimplementasi interface yang jumlahnya lebih dari satu?

Jelaskan alasannya!

**Jawab :**Iya bisa, karena pada interface method-methodnya harus didefinisikan kembali pada class yang mengimplementasikannya.

1. Pada interface Identitas.java hapus method tampilkan nama, amati apa yang terjadi dan mengapa demikian?

**Jawab :**



Akan terjadi error di sub class manusia. Dikarenakan terdapat method tampilkanNama() yang di override dari kelas interface.java

1. Jika pada class hewan kita hanya ingin mengimplements interface MakhlukHidup saja apa

yang terjadi? Jelaskan

D:\Sistem Informasi\Semester 2\PEMROGRAMAN LANJUT\interface\5.PNG

Tidak berpengaruh apa-apa karena method tampilkan nama dan tampilkan umur tidak menggunakan syntax ovveride, sehingga method tersebut dianggap hanya berada di kelas itu sendiri.

1. Buatlah konstruktor pada manusia dengan parameter umur dan nama kemudian panggil

pada Class Main dengan menginstan objek bernama nama anda!

**Jawab :**

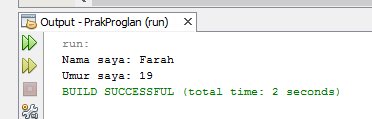
**Source Code :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public class Manusia implements MakhlukHidup, Identitas {  private String nama;  private int umur;  public Manusia (String N, int U){  this.nama=N;  this.umur=U;  } |
| 1  2  3  4  5 | public class MainIdentitas {  public static void main(String[] args) {  Manusia people = new Manusia ("Farah", 19);  people.tampilkanNama();  people.tampilkanUmur(); |

**Pembahasan :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | Mendeklarasikan sub kelas manusia implement dari Makhluk Hidup dan Identitas  Deklarasi variable nama dengan tipe data String  Deklarasi variable umur dengan tipe data integer  Membuat konstruktor Manusia dengan parameter String N dan int U  Nyatakan object dengan menggunakan kata “this”, seperti “this.nama=N;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.umur=U;” |
| 1  2  3  4  5  6 | Deklarasi kelas dan method main, mengetikkan “public class MainIdentitas {“  Kemudian mengetikkan “public static void main(String[]args){” sebagai fungsi main  Membuat objek dari kelas people dengan parameter Farah dan 19  Memanggil method tampilkanNama() pada kelas Manusia  Memanggil method tampilkanUmur() pada kelas Manusia |

**Output :**

****

1. Ubah source code diatas menjadi proses meminta masukan dari user dan buat menjadiinteraktif!

**Jawab :**

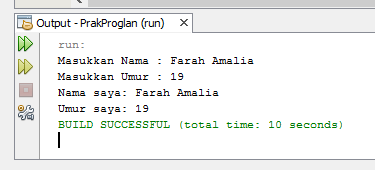
**Source Code :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public class Manusia implements MakhlukHidup, Identitas {  private String nama;  private int umur;  public Manusia (String N, int U){  this.nama=N;  this.umur=U;  } |
| 1  2  3  4  5  6  7  8  9  10  11 | import java.util.Scanner;  public class MainIdentitas {  public static void main(String[] args) {  Scanner input = new Scanner (System.in);  System.out.print("Masukkan Nama : ");  String nama = input.nextLine ();  System.out.print("Masukkan Umur : ");  int umur = input.nextInt();  Manusia people = new Manusia (nama, umur);  people.tampilkanNama();  people.tampilkanUmur(); |

**Pembahasan :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | Mendeklarasikan sub kelas manusia implement dari Makhluk Hidup dan Identitas  Deklarasi variable nama dengan tipe data String  Deklarasi variable umur dengan tipe data integer  Membuat konstruktor Manusia dengan parameter String N dan int U  Nyatakan object dengan menggunakan kata “this”, seperti “this.nama=N;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.umur=U;” |
| 1  2  3  4  5  6  7  8  9  10  11 | Deklarasi kelas dan method main, mengetikkan “public class MainIdentitas {“  Kemudian mengetikkan “public static void main(String[]args){” sebagai fungsi main  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.  User diminta memasukkan namanya  Membaca inputan dari user  User diminta memasukkan umurnya  Membaca inputan dari user  Membuat objek dari kelas people dengan parameter nama dan umur  Memanggil method tampilkanNama() pada kelas Manusia  Memanggil method tampilkanUmur() pada kelas Manusia |

**Output :**



1. Buat objek selain objek diatas dengan menggunakan method yang berbeda dengan yangdiatas! (min.1 contoh)

**Jawab :**

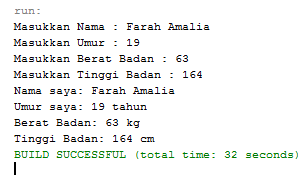
**Source Code:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public class Manusia implements MakhlukHidup, Identitas{  private String nama;  private int umur;  private int beratBadan;  private int tinggi;  public Manusia (String N, int U, int bb, int T){  this.nama=N;  this.umur=U;  this.beratBadan=bb;  this.tinggi=T;  } |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | import java.util.Scanner;  public class MainIdentitas {  public static void main(String[] args) {  Scanner input = new Scanner (System.in);  System.out.print("Masukkan Nama : ");  String nama = input.nextLine ();  System.out.print("Masukkan Umur : ");  int umur = input.nextInt();  System.out.print("Masukkan Berat Badan : ");  int BB = input.nextInt();  System.out.print("Masukkan Tinggi Badan : ");  int tinggi = input.nextInt();  Manusia people1 = new Manusia (nama, umur, BB, tinggi);  People1.tampilkanNama();  People1.tampilkanUmur();  People1.tampilkanBeratBadan();  People1.tampilkanTinggiBadan(); |

**Pembahasan :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Mendeklarasikan sub kelas manusia implement dari Makhluk Hidup dan Identitas  Deklarasi variable nama dengan tipe data String  Deklarasi variable umur dengan tipe data integer  Deklarasi variable beratBadan dengan tipe data integer  Deklarasi variable tinggi dengan tipe data integer  Membuat konstruktor Manusia dengan parameter String N, int U, int bb, int tinggi  Nyatakan object dengan menggunakan kata “this”, seperti “this.nama=N;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.umur=U;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.beratBadan=BB;”  Nyatakan object dengan menggunakan kata “this”, seperti “this.tinggi=tinggi;” |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Deklarasi kelas dan method main, mengetikkan “public class MainIdentitas {“  Kemudian mengetikkan “public static void main(String[]args){” sebagai fungsi main  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.  User diminta memasukkan namanya  Membaca inputan dari user  User diminta memasukkan umurnya  Membaca inputan dari user  User diminta memasukkan berat badannya  Membaca inputan dari user  User diminta memasukkan tinggi badannya  Membaca inputan dari user  Membuat objek dari kelas people dengan parameter nama dan umur  Memanggil method tampilkanNama() pada kelas Manusia  Memanggil method tampilkanUmur() pada kelas Manusia  Memanggil method tampilkanBeratBadan() pada kelas Manusia  Memanggil method tampilkanTinggiBadan() pada kelas Manusia |

**Output :**

****

1. **KESIMPULAN**

Interface adalah class yang hanya berisi deklarasi method tanpa memiliki implementasi dan semua property yang dimilikinya bersifat final. Interface mirip dengan class abstrak, tetapi interface tidak terikat dengan class hierarki. Interface mendefinisikan sebuah (signature) dari sebuah kumpulan method tanpa tubuh. Interface mendefenisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class

Sebuah Interface mendefinisikan protokol untuk berkomunikasi antar 2 object.

- Class yang meng-implements sebuah interface harus implement semua method yang dideclarasikan di dalam Interface.

- Nama Interface dapat digunakan di mana saja.

Interface ini bisa diturunkan atau diwariskan kepada class yang ingin memakai method yang ada dalam masing-masing interface tersebut dengan keyword **extends** [interface yang didefinisikan]. Dan juga sebuah class mengimplementasi 1 interface yang sudah dibuat dengan keyword **implements**

Secara default, interface hanya dapat diimplementasikan oleh class dipaket yang sama, tetapi bila memberi modifier public berarti kita memungkinkan class-class di luar paket dapat mengimplementasikan. Tubuh interface mengacu ke metode dan variable static final yang menyusun interface. Untuk mendeklarasikan sebuah interface, digunakan kata kunci interface sebelum nama interface.

Syntax Dasar

1.  public interface [InterfaceName]

2.  {

3.  //beberapa method tanpa isi

4.  }

Polimorfisme merupakan proses kebanyakrupaan, artinya kita dapat mengimplementasikan sesuatu hal yang berbeda melalui satu cara yang sama. Sedangkan Interface adalah jenis khusus dari blok yang hanya berisi method signature(atau constant). Interface mendefinisikan sebuah (signature) dari sebuah kumpulan method tanpa tubuh.

Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class.Perbedaan antara Interface dengan Polymorfisme adalah saat instansiasi objek di main class. Untuk polymorfisme objek dapat berubah bentuk dengan pendeklarasian <superclass><nama objek> = new <nama subclass> ( ), sedangkan Interface hanya dapat di instansiasi dengan menggunakan class turunannya saja dan interface tidak memiliki atribut atau description.

Contoh-contoh interface dalam bahasa pemrograman java!

* Comparable
* Serializable
* Runnable
* GUI event-listener Interfaces
* SwingConstans

• Kita menggunakan interface jika kita ingin class yangtidak berhubungan mengimplementasikan methodatau fungsi-fungsi yang sama.

• Melalui interface kita dapat menangkap kemiripandiantara class yang tidak berhubungan

• Sebuah Class dapat mengimplementasikan lebih darisatu interface