

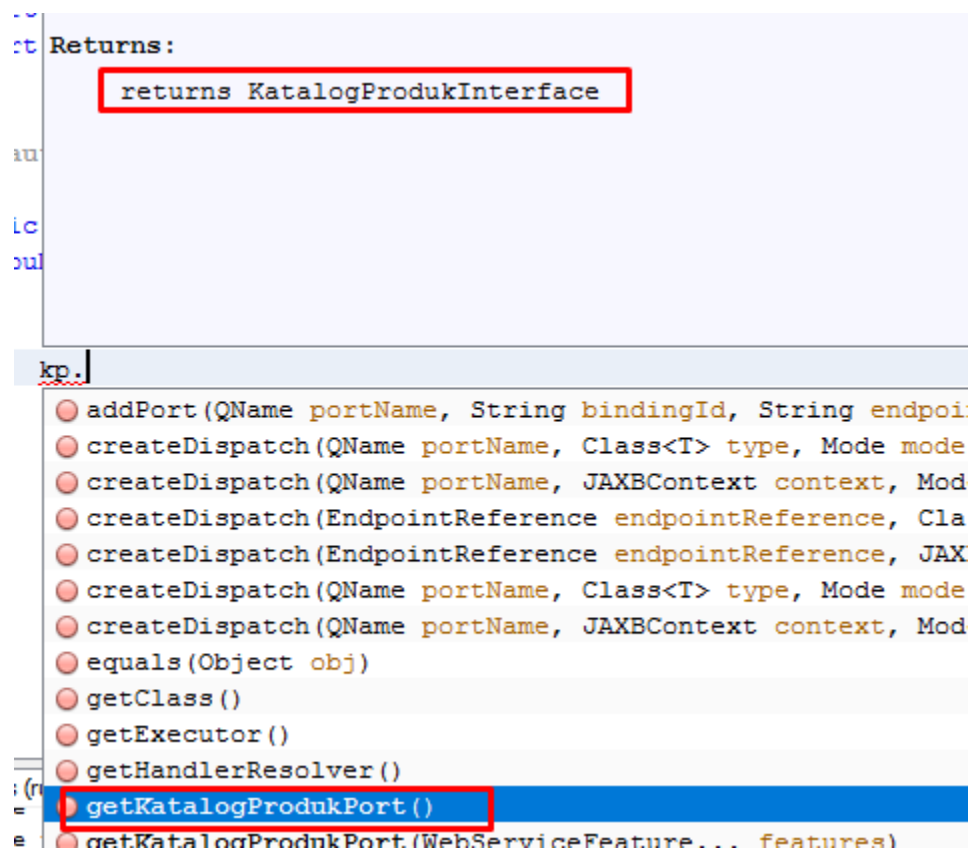
LAPORAN PRAKTIKUM PEMROGRAMAN LANJUT 2

Nama : Arif Hidayat

NIM : 16/400617/SV/11121

```
<binding>
- <service name="KatalogProdukService">
  - <port name="KatalogProdukPort" binding="tns:KatalogProdukPortBinding">
    <soap:address location="http://10.33.109.17:8080/KatalogProduk/KatalogProduk"/>
  </port>
</service>
</definitions>
```

Gambar 1



Gambar 2

```

-|      public static void main(String[] args){
|          KatalogProdukService kp = new KatalogProdukService();
|          KatalogProdukInterface kpi = kp.getKatalogProdukPort();
|          if(args.length!=1){
|              for(String katagori : kpi.getProductCategories()){
|                  System.out.println("Kategori "+katagori);
|              }
|          }else{
|              String katagori = args[0];
|              for (Produk produk : kpi.getProducts(katagori)){
|                  System.out.println("Name : "+produk.getName());
|                  System.out.println("SKU : "+produk.getSku());
|                  System.out.println("Price : "+produk.getPrice());
|              }
|          }
|      }
|  }
}

```

Gambar 3

Analisis : Pada **Gambar 1** terdapat **KatalogProdukService** untuk menghubungkan webservice dengan java client. Oleh karena itu pada Class main **menginstansiasi** kelas **KatalogProdukService** yang telah diwsimport, menjadi objek **kp**. Lalu untuk mengetahui bagaimana cara mengambil data melewati mana ke webservice maka pada objek **kp** melakukan pemanggilan method **getKatalogProdukPort**, karena pada **Gambar 1** di dalam **KatalogProdukService** terdapat method **getKatalogProdukPort**. Karena pada **Gambar 2** ternyata method **getKatalogProdukPort** melakukan return ke **KatalogProdukInterface** maka value dari method **getKatalogProdukPort** disimpan didalam objek **kpi** yang merupakan hasil dari **KatalogProdukInterface**. Pada webservice xml terdapat method **getProductCategories** berbentuk **List**. Oleh karena itu tidak memerlukan parameter sehingga **argumen tidak perlu diisi**. Untuk menampilkan listnya dilakukan dengan cara membuat variabel bertipe String untuk mendapat value dari method **getProductCategories** yang merupakan string juga. Lalu jika ingin menampilkan data berdasarkan kategori yang dipilih maka memerlukan argumen dengan cara meletakkan argumen untuk disimpan kedalam variabel **katagori**. Lalu karena ingin menampilkan semua data yang ingin dicari didalam method **getproducts** berdasarkan **katagori** maka memerlukan **perulangan** dengan mengulang sebanyak isi dari **katagori produk** tersebut. Karena method **getProducts** mereferensi kelas **produk** maka untuk menampilkan datanya memerlukan objek dari kelas **produk** dan kelas **produk** memiliki beberapa method untuk menampilkan value yang dicari berdasarkan **parameter**.

TUGAS

```
- <service name="CountryInfoService">
  - <documentation>
    This DataFlex Web Service opens up country information. 2 letter ISO codes are used for Country code. There
    are functions to retrieve the used Currency, Language, Capital City, Continent and Telephone code.
  </documentation>
  - <port name="CountryInfoServiceSoap" binding="tns:CountryInfoServiceSoapBinding">
    <soap:address location="http://webservices.oorsprong.org/websamples.countryinfo
    /CountryInfoService.wso"/>
  </port>
  - <port name="CountryInfoServiceSoap12" binding="tns:CountryInfoServiceSoapBinding12">
    <soap12:address location="http://webservices.oorsprong.org/websamples.countryinfo
    /CountryInfoService.wso"/>
  </port>
</service>
</definitions>
```

Gambar 1

1. Continent

```
import org.oorsprong.websamples.ArrayOfContinents;
import org.oorsprong.websamples.CountryInfoService;
import org.oorsprong.websamples.CountryInfoServiceSoapType;
import org.oorsprong.websamples.TContinent;

/**
 *
 * @author PC-10
 */
public class ContinentNameMain {
    public static void main(String[] args) {
        CountryInfoService ci = new CountryInfoService();
        CountryInfoServiceSoapType cit = ci.getCountryInfoServiceSoap();
        ArrayOfContinents ac = cit.listOfContinentsByName();
        for (TContinent TCon : ac.getTContinent()) {
            System.out.println("Benua " + TCon.getSCode() + "-" + TCon.getSName());
        }
    }
}
```

tput - Tugas (run) × Main.java ×

run:

Benua AF-Africa
Benua AN-Antarctica
Benua AS-Asia
Benua EU-Europe
Benua OC-Oceania
Benua AM-The Americas

Gambar 2

Analisis : Pada **Gambar 1** terdapat **CountryInfoService** untuk menghubungkan webservice dengan java client. Oleh karena itu pada Class main **menginstansiasi** kelas **CountryInfoService** yang telah diwsimport, menjadi objek **ci**. Lalu untuk mengetahui bagaimana cara mengambil data melewati mana ke webservice maka pada objek **ci** melakukan pemanggilan method **getCountryInfoServiceSoap**, karena pada **Gambar 1** di dalam **CountryInfoService** terdapat method **getCountryInfoServiceSoap**. Karena pada **Gambar 2** ternyata method **getCountryInfoServiceSoap** melakukan return ke **CountryInfoServiceSoapType** maka value dari method **getCountryInfoServiceSoap** disimpan didalam objek **cit** yang merupakan hasil dari **getCountryInfoServiceSoap**. Lalu pada objek **cit** untuk menampilkan method **listOfContinentByName** ternyata melakukan return ke **ArrayOfContinent**. Maka membuat objek **ac** dari kelas **ArrayOfContinent**. Karena value dalam objek **ac** merupakan list maka dilakukan perulangan dengan memanggil method **getTContinent** dimana method tersebut mereferensikan ke kelas **TContinent** untuk memanggil valuenya. Maka dilakukan output dengan memanggil objek dari kelas TContinent yaitu **getSName**.

2. Country

```
import org.oorsprong.websamples.ArrayOfCountryCodeAndName;
import org.oorsprong.websamples.CountryInfoService;
import org.oorsprong.websamples.CountryInfoServiceSoapType;
import org.oorsprong.websamples.TCountryCodeAndName;

/**
 *
 * @author PC-10
 */
public class CountryNameMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        CountryInfoService ci = new CountryInfoService();
        CountryInfoServiceSoapType cit = ci.getCountryInfoServiceSoap();
        ArrayOfCountryCodeAndName cna = cit.listOfCountryNamesByName();
        for (TCountryCodeAndName TC : cna.getTCountryCodeAndName()) {
            System.out.println("Negara " + TC.getSISOCODE() + "-" + TC.getSName());
        }
    }
}
```

tput - Tugas (run) x Main.java x

```
run:
Negara AX-Åland Islands
Negara AF-Afghanistan
Negara AL-Albania
Negara DZ-Algeria
Negara AS-American Samoa
Negara AD-Andorra
Negara AO-Angola
```

Gambar 3

Analisis : Pada **Gambar 1** terdapat **CountryInfoService** untuk menghubungkan webservice dengan java client. Oleh karena itu pada Class main **menginstansiasi** kelas **CountryInfoService** yang telah diwsimport, menjadi objek **ci**. Lalu untuk mengetahui bagaimana cara mengambil data melewati mana ke webservice maka pada objek **ci** melakukan pemanggilan method **getCountryInfoServiceSoap**, karena pada **Gambar 1** di dalam **CountryInfoService** terdapat method **getCountryInfoServiceSoap**. Karena pada **Gambar 3** ternyata method

getCountryInfoServiceSoap melakukan return ke **CountryInfoServiceSoapType** maka value dari method **getCountryInfoServiceSoap** disimpan didalam objek **cit** yang merupakan hasil dari **getCountryInfoServiceSoap**. Lalu pada objek **cit** untuk menampilkan method **listOfCountryCodeAndName** ternyata melakukan return ke **ArrayOfCountryCodeAndName**. Maka membuat objek **ac** dari kelas **ArrayOfCountryCodeAndName**. Karena value dalam objek **cna** merupakan list maka dilakukan perulangan dengan memanggil method **getTCountryCodeAndName** dimana method tersebut mereferensikan ke kelas **TCountryCodeAndName** untuk memanggil valuenya. Maka dilakukan output dengan memanggil objek dari kelas **TCountryCodeAndName** yaitu **getSName**.

3. Currency

```
import org.oorsprong.websamples.ArrayOfCurrency;
import org.oorsprong.websamples.CountryInfoService;
import org.oorsprong.websamples.CountryInfoServiceSoapType;
import org.oorsprong.websamples.TCurrency;

/**
 *
 * @author PC-10
 */
public class CurrencyNameMain {
    public static void main(String[] args){
        CountryInfoService ci = new CountryInfoService();
        CountryInfoServiceSoapType cit = ci.getCountryInfoServiceSoap();
        ArrayOfCurrency Ac = cit.listOfCurrenciesByName();
        for (TCurrency TC : Ac.getTCurrency()){
            System.out.println("Currency "+TC.getSISOCODE()+"-"+ TC.getSName());
        }
    }
}
```

```
put - Tugas (run) x Main.java x
Currency TRL-Liras [being phased out]
Currency MTL-Liri
Currency LTL-Litai
Currency SPL-Luigini
Currency LSL-Maloti
Currency AZM-Manats
Currency TMM-Manats
Currency MZM-Meticais
Currency MAD-Morocco Dirhams
Currency NGN-Nairas
```


Gambar 4

Analisis : Pada **Gambar 1** terdapat **CountryInfoService** untuk menghubungkan webservice dengan java client. Oleh karena itu pada Class main **menginstansiasi** kelas **CountryInfoService** yang telah diwsimport, menjadi objek **ci**. Lalu untuk mengetahui bagaimana cara mengambil data melewati mana ke webservice maka pada objek **ci** melakukan pemanggilan method **getCountryInfoServiceSoap**, karena pada **Gambar 1** di dalam **CountryInfoService** terdapat method **getCountryInfoServiceSoap**. Karena pada **Gambar 4** ternyata method **getCountryInfoServiceSoap** melakukan return ke **CountryInfoServiceSoapType** maka value dari method **getCountryInfoServiceSoap** disimpan didalam objek **cit** yang merupakan hasil dari **getCountryInfoServiceSoap**. Lalu pada objek **cit** untuk menampilkan method **listOfCurrenciesByName** ternyata melakukan return ke **ArrayOfCurrenciesByName**. Maka membuat objek **Ac** dari kelas **ArrayOfCurrency**. Karena value dalam objek **Ac** merupakan list maka dilakukan perulangan dengan memanggil method **getTCurrency** dimana method tersebut mereferensikan ke kelas **TCurrency** untuk memanggil valuenya. Maka dilakukan output dengan memanggil objek dari kelas **TCurrency** yaitu **getSName**.

4. Language

```
import org.oorsprong.websamples.ArrayOfLanguage;
import org.oorsprong.websamples.CountryInfoService;
import org.oorsprong.websamples.CountryInfoServiceSoapType;
import org.oorsprong.websamples.TLanguage;

/**
 *
 * @author PC-10
 */
public class LanguageNameMain {
    public static void main(String[] args){
        CountryInfoService ci = new CountryInfoService();
        CountryInfoServiceSoapType cit = ci.getCountryInfoServiceSoap();
        ArrayOfLanguage Al = cit.listOfLanguagesByName();
        for (TLanguage TL : Al.getTLanguage()){
            System.out.println("Language "+TL.getSISOCODE()+"-"+TL.getSName());
        }
    }
}
```



Gambar 5

Analisis : Pada **Gambar 1** terdapat **CountryInfoService** untuk menghubungkan webservice dengan java client. Oleh karena itu pada Class main **menginstansiasi** kelas **CountryInfoService** yang telah diwsimport, menjadi objek **ci**. Lalu untuk mengetahui bagaimana cara mengambil data melewati mana ke webservice maka pada objek **ci** melakukan pemanggilan method **getCountryInfoServiceSoap**, karena pada **Gambar 1** di dalam **CountryInfoService** terdapat method **getCountryInfoServiceSoap**. Karena pada **Gambar 5** ternyata method **getCountryInfoServiceSoap** melakukan return ke **CountryInfoServiceSoapType** maka value dari method **getCountryInfoServiceSoap** disimpan didalam objek **cis** yang merupakan hasil dari **getCountryInfoServiceSoap**. Lalu pada objek **cis** untuk menampilkan method **listOfLanguageByName** ternyata melakukan return ke **ArrayOfLanguage**. Maka membuat objek **Al** dari kelas **ArrayOfLanguage**. Karena value dalam objek **Al** merupakan list maka dilakukan perulangan dengan memanggil method **getTLanguage** dimana method tersebut mereferensikan ke kelas **TLanguage** untuk memanggil valuenya. Maka dilakukan output dengan memanggil objek dari kelas **Tlanguage** yaitu **getName**.

5. Find Country and Currency

```
public static void main(String[] args){
    CountryInfoService ci = new CountryInfoService();
    CountryInfoServiceSoapType cis = ci.getCountryInfoServiceSoap();
    Scanner scanner = new Scanner(System.in);
    System.out.print("Kode Negara : ");
    String code = scanner.nextLine();
    System.out.print("Kode Mata uang : ");
    String codeCurrency = scanner.nextLine();
    String code = args[0];
    cis.countryName(code);
    String codeCurrency = args[1];
    System.out.println("Country Name "+cis.countryName(code));
    System.out.println("Capital City "+cis.capitalCity(code));
    System.out.println("Currency "+cis.currencyName(codeCurrency));
}
```

Gambar 6

```
E:\Pertemuan 11\Tugas>java -jar "E:\Pertemuan 11\Tugas\dist\Tugas.jar" ID IDR
Country Name Indonesia
Capital City Jakarta
Currency City Rupiahs
```


Analisis : Analisis : Pada **Gambar 1** terdapat **CountryInfoService** untuk menghubungkan webservice dengan java client. Oleh karena itu pada Class main **menginstansiasi** kelas **CountryInfoService** yang telah diwsimport, menjadi objek **ci**. Lalu untuk mengetahui bagaimana cara mengambil data melewati mana ke webservice maka pada objek **ci** melakukan pemanggilan method **getCountryInfoServiceSoap**, karena pada **Gambar 1** di dalam **CountryInfoService** terdapat method **getCountryInfoServiceSoap**. Karena pada **Gambar 6** ternyata method **getCountryInfoServiceSoap** melakukan return ke **CountryInfoServiceSoapType** maka value dari method **getCountryInfoServiceSoap** disimpan didalam objek **cis** yang merupakan hasil dari **getCountryInfoServiceSoap**. Pada objek **cis** terdapat method untuk memanggil **CountryName**, **capitalCity** dan **currencyName** berdasarkan parameter **SISOCODE**. Nilai dari parameter tersebut didapat dari variabel **code** yang merupakan argumen **pertama** dan **codeCurrency** yang merupakan argumen **kedua**.