

MODUL 3: Web Service berbasis SOAP dan WSDL

3.1. Deskripsi Singkat

Web service adalah suatu metode komunikasi antar dua perangkat elektronik lewat world wide web (www). Web service membantu kita dalam mengkonversi aplikasi kita menjadi aplikasi berbasis web. Saat ini terdapat 2 macam web service yaitu web service yang menggunakan SOAP dan WSDL, dan web service yang menggunakan REST, atau lebih dikenal dengan RESTful web service. Pada praktikum kali ini kita akan mempraktekkan pembuatan webservice berbasis SOAP dan WSDL.

3.2. Tujuan Praktikum

Mahasiswa dapat mempraktekkan pembuatan web service berbasis SOAP dan WSDL menggunakan Netbeans

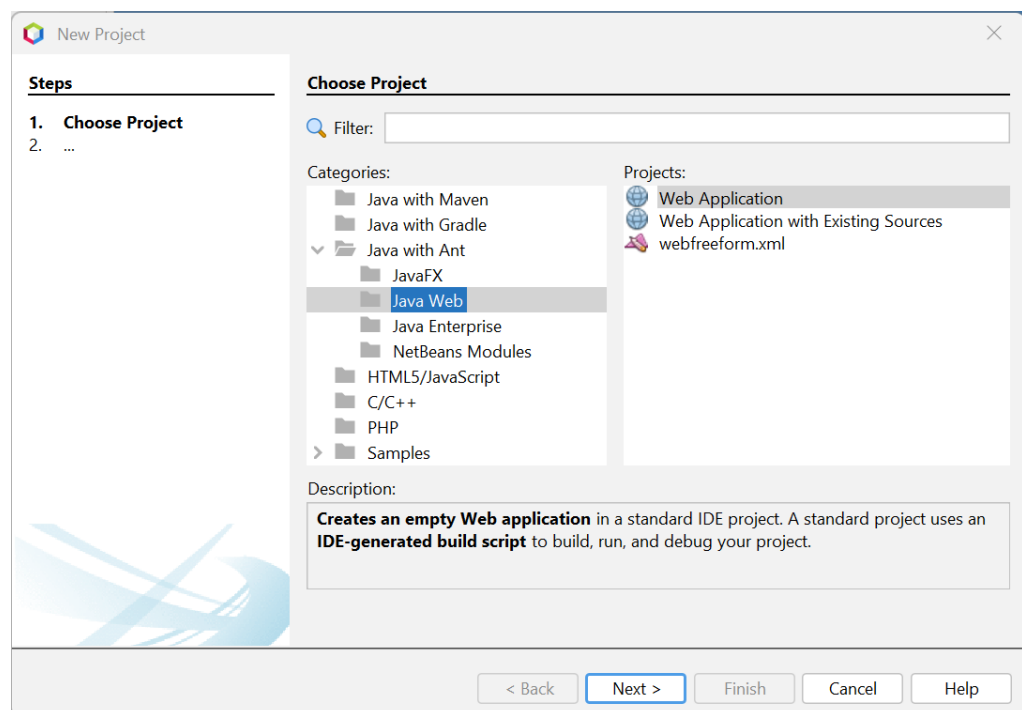
3.3. Material Praktikum

- Apache Netbeans 14
- Telah terinstall Tomcat Server (Dapat memanfaatkan XAMPP)

3.4. Kegiatan Praktikum

Merujuk pada <https://netbeans.apache.org/kb/docs/websvc/jax-ws.html>

Buka Netbeans-> Java web -> web application



The screenshot shows the 'New Web Application' dialog box with the 'Name and Location' step selected. The 'Steps' list on the left includes: 1. Choose Project, 2. **Name and Location**, 3. Server and Settings, and 4. Frameworks. The main area contains the following fields and options:

- Project Name:** CalculatorWSApplication
- Project Location:** C:\Users\User\Documents\NetBeansProjects (with a 'Browse...' button)
- Project Folder:** s\User\Documents\NetBeansProjects\CalculatorWSApplication
- ☐ Use Dedicated Folder for Storing Libraries
- Libraries Folder:** (with a 'Browse...' button)
- Text: Different users and projects can share the same compilation libraries (see Help for details).

At the bottom, there are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted.

Beri nama project CalculatorWSApplication

The screenshot shows the 'New Web Application' dialog box with the 'Server and Settings' step selected. The 'Steps' list on the left includes: 1. Choose Project, 2. Name and Location, 3. **Server and Settings**, and 4. Frameworks. The main area contains the following fields and options:

- Add to Enterprise Application:** <None> (dropdown menu)
- Server:** Apache Tomcat or TomEE (dropdown menu) with an 'Add...' button
- Java EE Version:** Java EE 7 Web (dropdown menu)
- Note: Source Level 7 will be set for Java EE 7 project.
- Context Path:** /CalculatorWSApplication

At the bottom, there are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted.

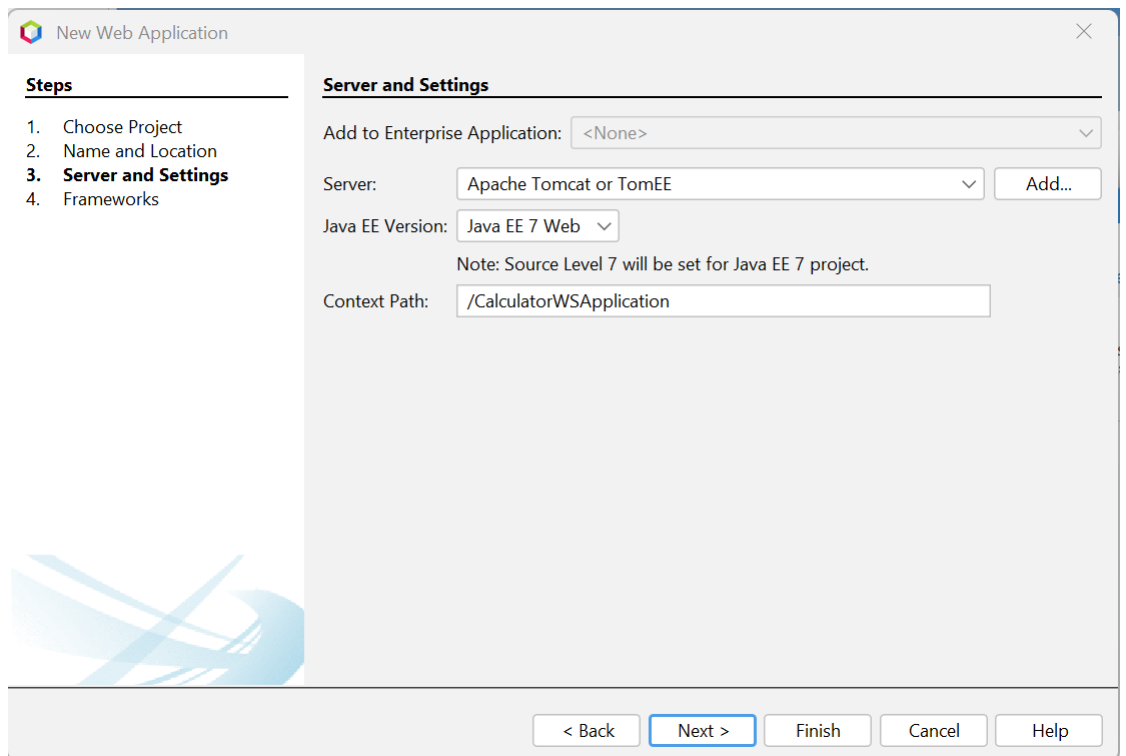
Jika belum terdapat server yang terdaftar pada netbeans, tambahkanlah dahulu dengan menekan tombol Add

The screenshot shows the 'Add Server Instance' dialog box with the 'Steps' panel on the left and the 'Choose Server' panel on the right. The 'Steps' panel lists '1. Choose Server' and '2. ...'. The 'Choose Server' panel has a 'Server:' label and a list box containing 'Amazon Beanstalk', 'Apache Tomcat or TomEE' (which is selected), 'GlassFish Server', 'Payara Server', and 'WildFly Application Server'. Below the list box is a 'Name:' label and a text field containing 'Apache Tomcat or TomEE (1)'. At the bottom of the dialog are five buttons: '< Back', 'Next >' (highlighted with a blue border), 'Finish', 'Cancel', and 'Help'.

Pilih Apache Tomcat or TomEE kemudian masukkan setting, masukkan username admin dan password admin dan check pilihan create user if does ot exist

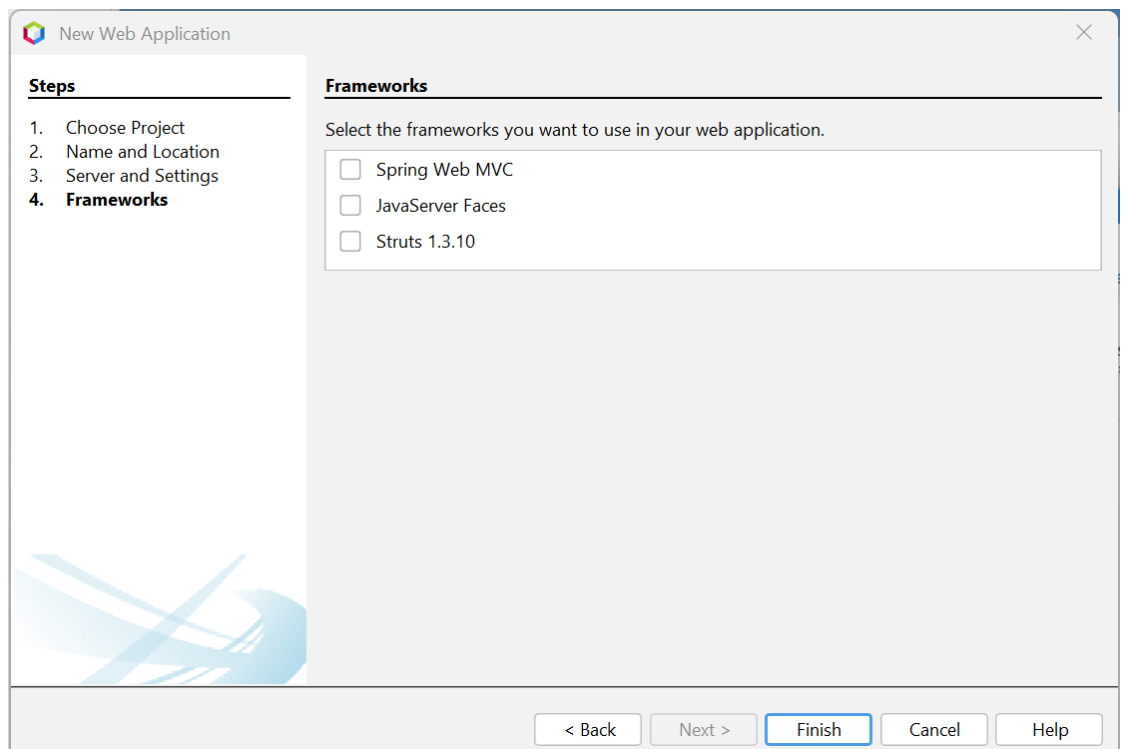
The screenshot shows the 'Add Server Instance' dialog box with the 'Steps' panel on the left and the 'Installation and Login Details' panel on the right. The 'Steps' panel lists '1. Choose Server' and '2. Installation and Login Details'. The 'Installation and Login Details' panel has a title bar 'Installation and Login Details' and a subtitle 'Specify the Server Location (Catalina Home) and login details'. It contains a 'Server Location:' label and a text field with 'C:\xampp\tomcat', followed by a 'Browse ...' button. Below this is a checkbox labeled 'Use Private Configuration Folder (Catalina Base)' which is unchecked. Underneath is a 'Catalina Base:' label and an empty text field, followed by another 'Browse ...' button. The next section is titled 'Enter the credentials of an existing user in the manager or manager-script role'. It has a 'Username:' label and a text field with 'admin', and a 'Password:' label and a text field with '.....'. At the bottom is a checked checkbox labeled 'Create user if it does not exist'.

Jika sudah maka kita klik next.



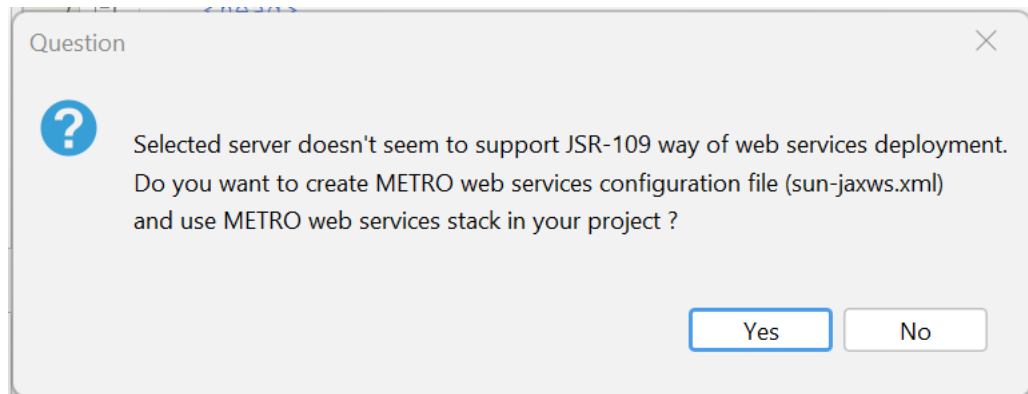
The screenshot shows the 'New Web Application' wizard at Step 3, 'Server and Settings'. The 'Steps' list on the left includes: 1. Choose Project, 2. Name and Location, 3. **Server and Settings**, and 4. Frameworks. The main area contains the following settings: 'Add to Enterprise Application' is set to '<None>'; 'Server' is set to 'Apache Tomcat or TomEE' with an 'Add...' button; 'Java EE Version' is set to 'Java EE 7 Web'; a note states 'Source Level 7 will be set for Java EE 7 project.'; and 'Context Path' is set to '/CalculatorWSApplication'. At the bottom, the 'Next >' button is highlighted in blue, with other buttons being '< Back', 'Finish', 'Cancel', and 'Help'.

Tidak perlu memilih framework karena kita akan membuat from scratch. Klik finish

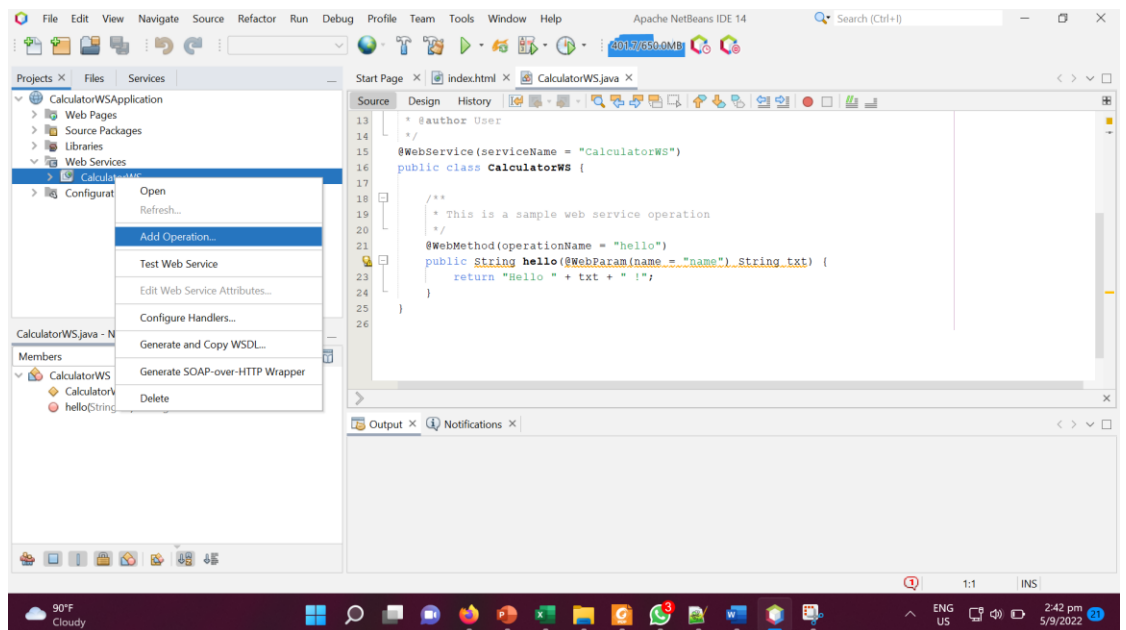


The screenshot shows the 'New Web Application' wizard at Step 4, 'Frameworks'. The 'Steps' list on the left includes: 1. Choose Project, 2. Name and Location, 3. Server and Settings, and 4. **Frameworks**. The main area is titled 'Frameworks' and contains the instruction 'Select the frameworks you want to use in your web application.' Below this, there is a list of frameworks with checkboxes: 'Spring Web MVC', 'JavaServer Faces', and 'Struts 1.3.10'. All checkboxes are currently unchecked. At the bottom, the 'Finish' button is highlighted in blue, with other buttons being '< Back', 'Next >', 'Cancel', and 'Help'.

Buatlah Web Service dengan cara klik kanan folder project dan tambahkan web service.



Klik yes jika muncul notifikasi seperti di atas



Klik kanan CalculatorWS, Add Operation

Add Operation

Name:

Return Type: Browse...

Parameters Exceptions

| Name | Type | Final |
|------|------|--------------------------|
| a | int | <input type="checkbox"/> |
| b | int | <input type="checkbox"/> |

Add Remove Up Down

OK Cancel

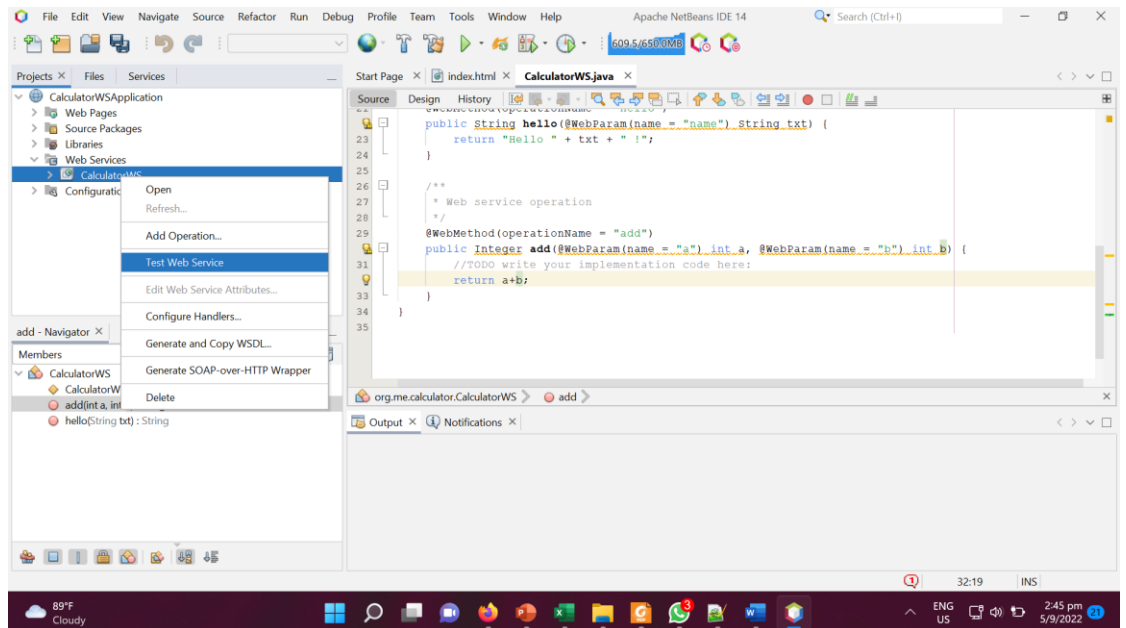
Tambahkan operasi baru dengan nama add bertipe Integer, parameter a dan b bertipe int

```

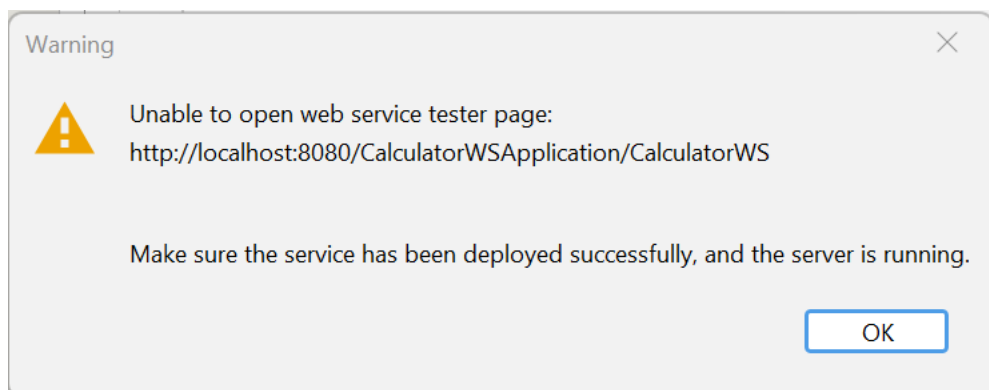
21  @WebMethod(operationName = "hello")
22  public String hello(@WebParam(name = "name") String txt) {
23      return "Hello " + txt + " !";
24  }
25
26  /**
27   * Web service operation
28   */
29  @WebMethod(operationName = "add")
30  public Integer add(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
31      //TODO write your implementation code here:
32      return a+b;
33  }
34
35

```

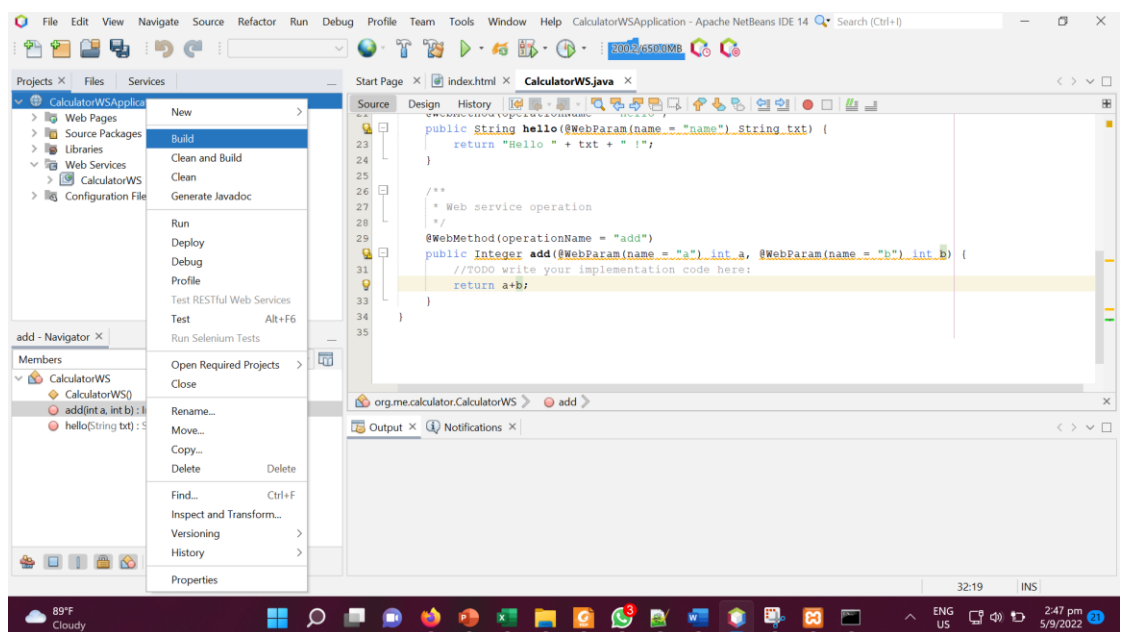
Ubah menjadi return a+b;



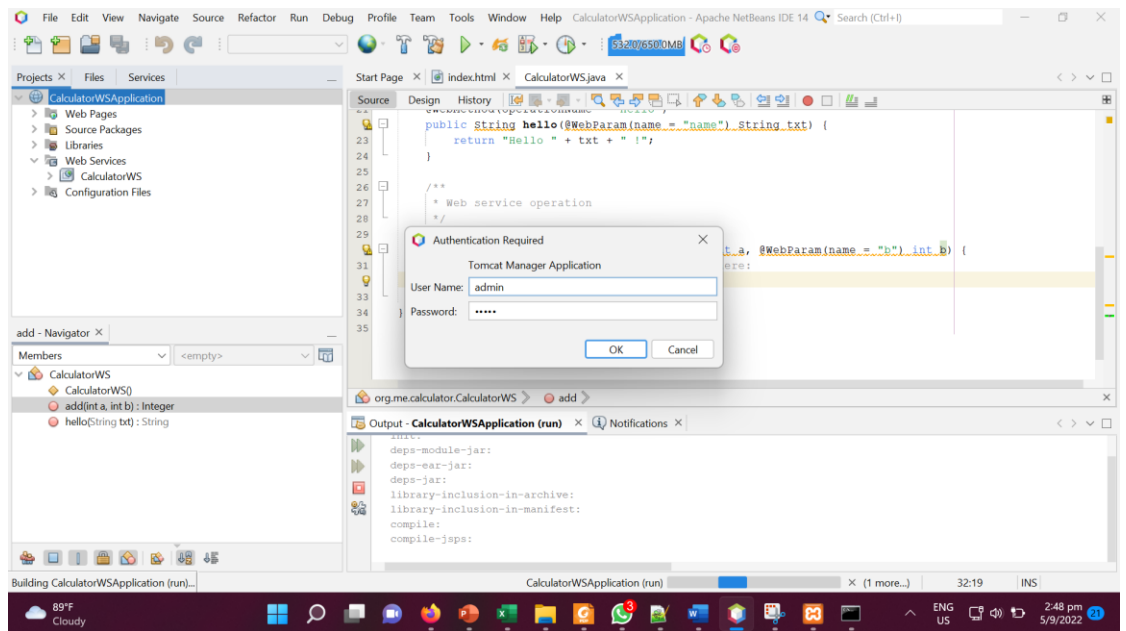
Test web service



Muncul error karena belum di-run



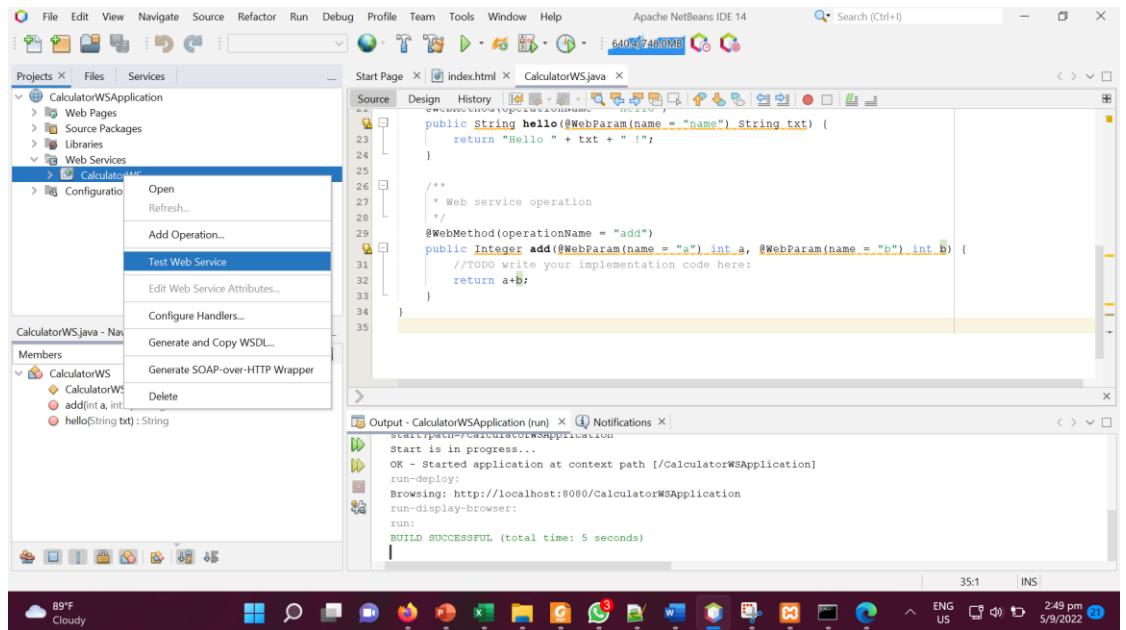
Build, kemudian Run



Masukkan username admin dan password admin, OK

Akan muncul halaman web





Test web service



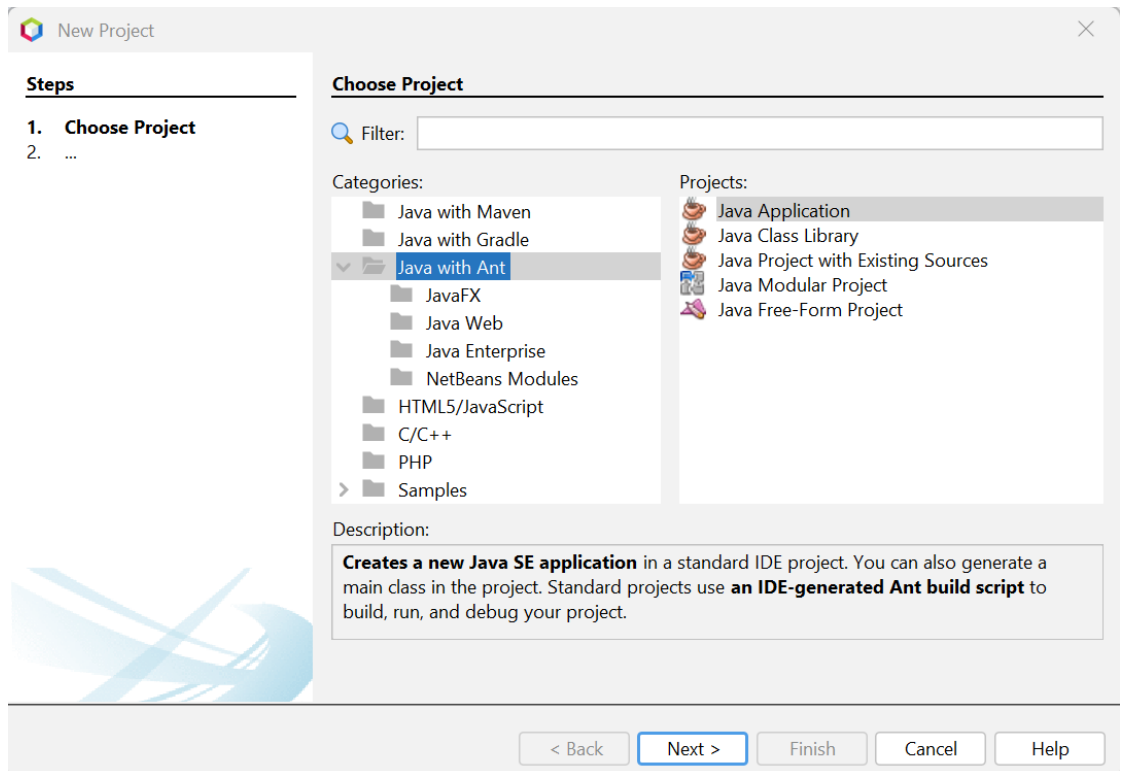
Web Services

| Endpoint | Information |
|--|---|
| Service Name: {http://calculator.me.org/}CalculatorWS | Address: http://localhost:8080/CalculatorWSApplication/CalculatorWS |
| Port Name: {http://calculator.me.org/}CalculatorWSPort | WSDL: http://localhost:8080/CalculatorWSApplication/CalculatorWS?wsdl |
| | Implementation class: org.me.calculator.CalculatorWS |

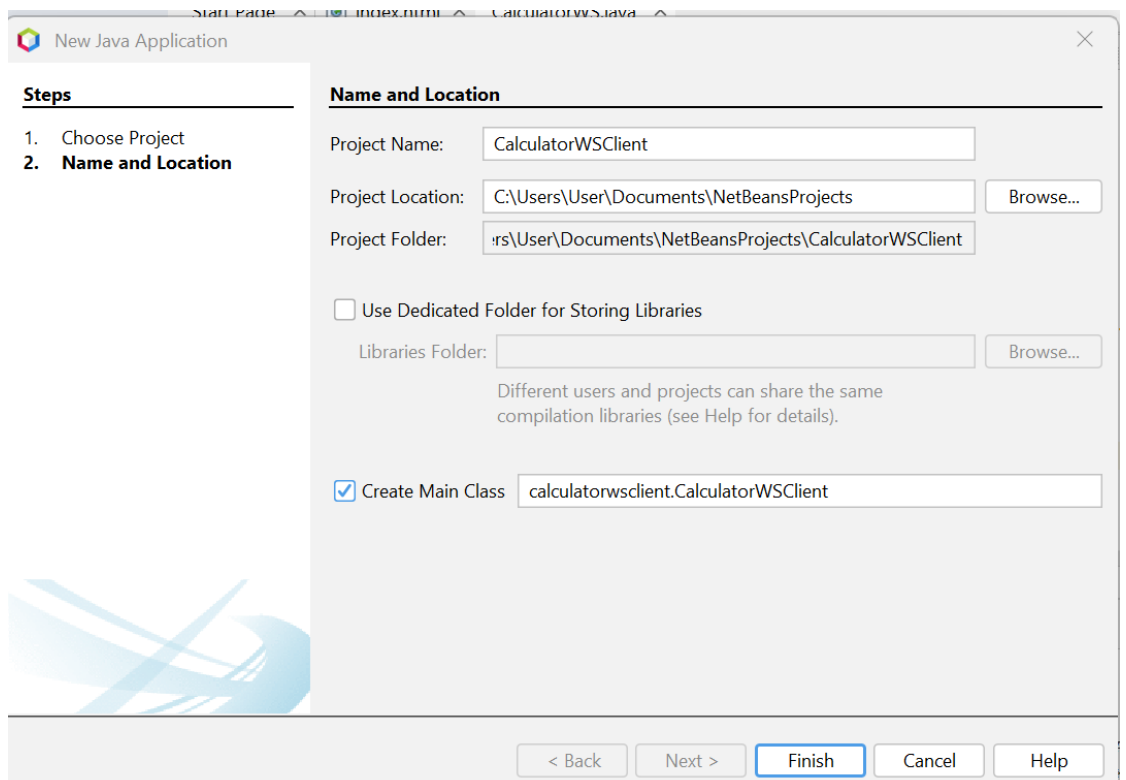
Muncul halaman WSDL yang nanti bisa dikonsumsi oleh aplikasi consumer

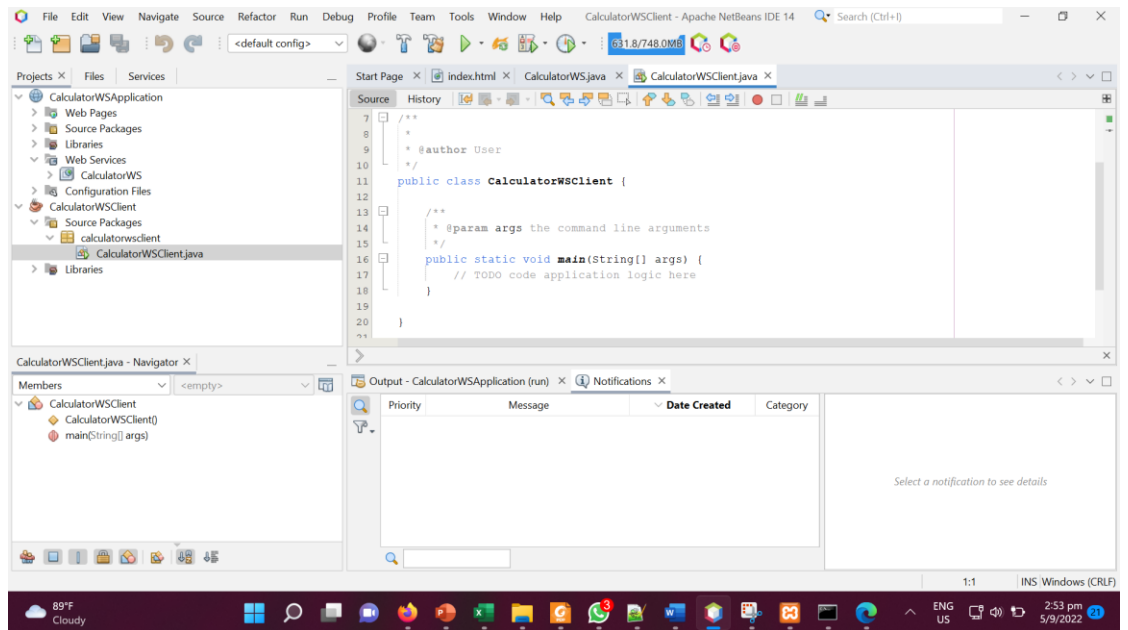
MENGKONSUMSI WEB SERVICE

Buatlah project baru untuk mengkonsumsi web service

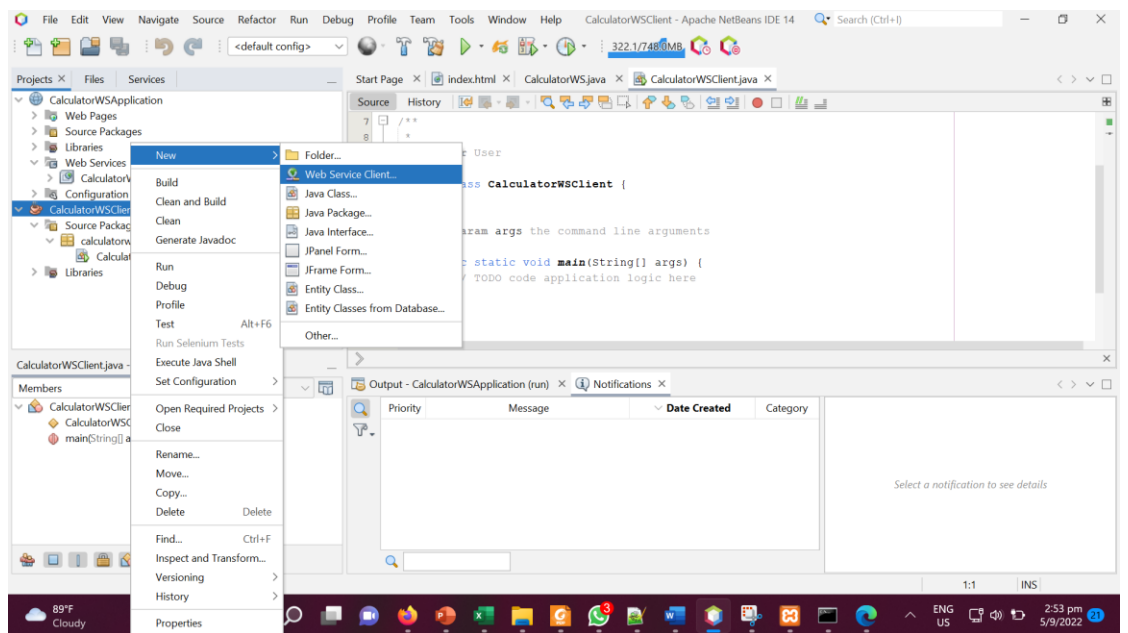


Beri nama CalculatorWSClient

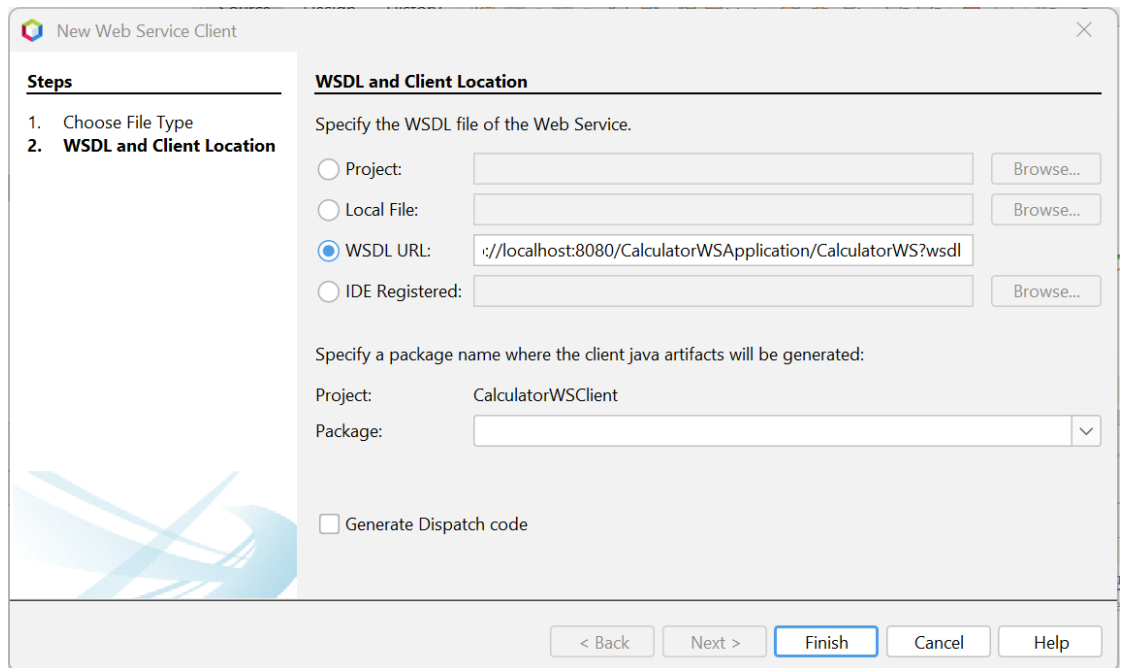




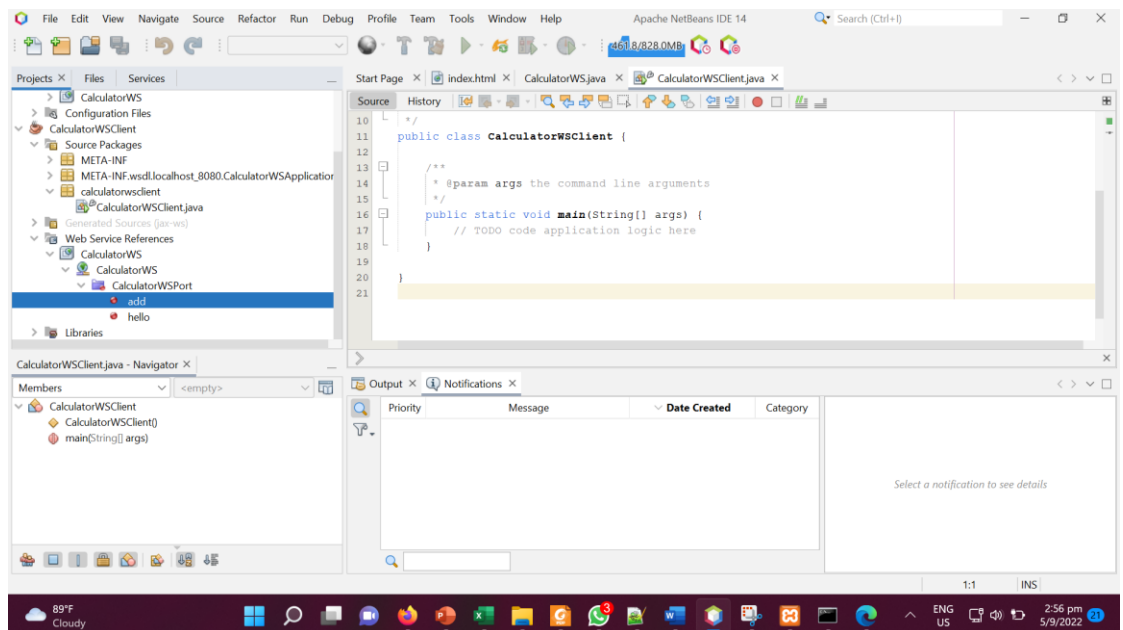
Pada project, klik kanan, new web service client



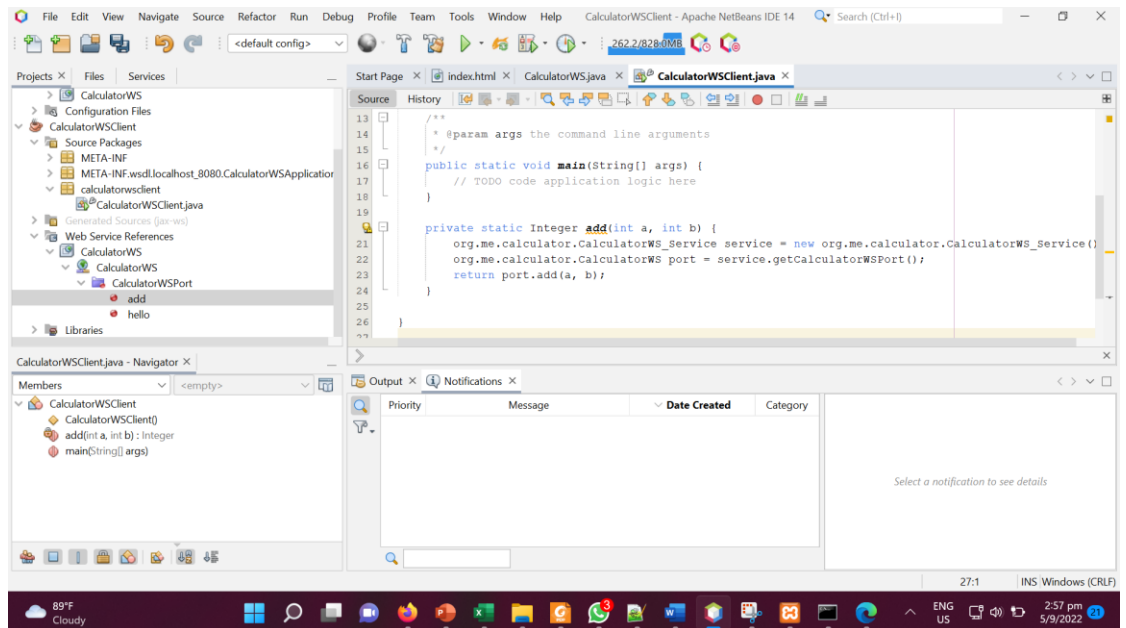
Bisa memilih project atau menambahkan wsdl URL sesuai di halaman sebelumnya



Klik Finish



Seret method add pada Web Service References ke kanan, sehingga muncul implementasi method add

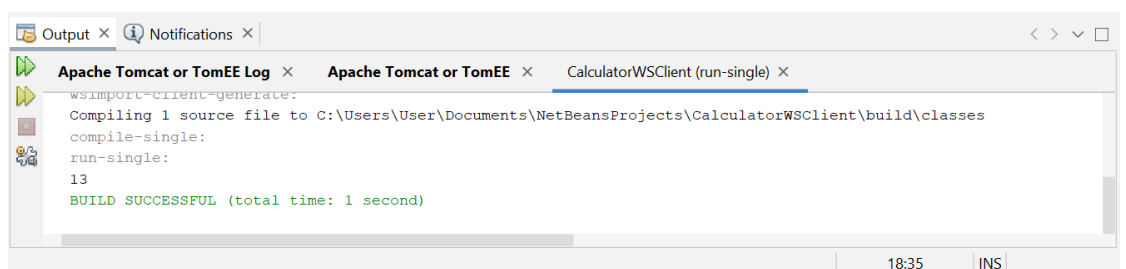


Ubah pada method main untuk memanggil add

```
public static void main(String[] args) {
    // TODO code application logic here
    System.out.println(add(5, 8));
}

private static Integer add(int a, int b) {
    org.me.calculator.CalculatorWS_Service se
    org.me.calculator.CalculatorWS port = ser
    return port.add(a, b);
}
```

Kemudian jalankan



Aplikasi Client CalculatorWSClient memanggil method add dengan parameter 5 dan 8. Web Service CalculatorWSApplication menerima input tersebut, menjumlahkan hasilnya, dan mengembalikan nilainya ke aplikasi client.

3.5. Penugasan

Buatlah laporan tangkapan layar pada kegiatan yang anda lakukan pada 3.4.