

USULAN PENELITIAN

YOK BISA YOK



Oleh:

Setia Pramana, S.Si., Ph.D.

Firdaus, MBA

PUSAT PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT

POLITEKNIK STATISTIKA STIS

JAKARTA 2022

HALAMAN PENGESAHAN

PROPOSAL PENELITIAN DOSEN

1. Judul Penelitian : Yok bisa yok
2. Bidang : SDG's
3. Ketua Tim Peneliti
 - a. Nama Lengkap : Setia Pramana, S.Si., Ph.D.
 - b. Jabatan Fungsional : Guru Besar
 - c. Program Studi : KS
 - d. Nomor HP : 081297893257
 - e. Email : setia.pramana@stis.ac.id
4. Jumlah Anggota : 1 orang
 - a. Nama Anggota : Firdaus, MBA
Program Studi : KS

Mengetahui

Kepala PPPM

Jakarta, 08 December 2022

Ketua Peneliti



Dr. Eng. Arie Wahyu Wijayanto, M.T.

NIDN :

Setia Pramana, S.Si., Ph.D.

NIDN : 197707222000031002

Menyetujui

Direktur Politeknik Statistika STIS

Dr. Erni Tri Astuti, M.Math.

NIDN :

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini

Nama : Setia Pramana, S.Si., Ph.D.

NIDN : 197707222000031002

Jabatan Fungsional : Guru Besar

Dengan ini menyatakan bahwa proposal penelitian saya yang berjudul: Yok bisa yok yang diusulkan dengan skema penelitian dosen tahun 2022 bersifat original dan belum pernah mendapatkan biaya/dibiayai oleh lembaga atau sumber biaya lainnya. Jika dikemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, saya bersedia dituntut dan diproses sesuai dengan ketentuan yang berlaku dan mengembalikan seluruh biaya yang sudah saya terima.

Jakarta, 08 December 2022

Ketua Peneliti



Setia Pramana, S.Si., Ph.D.

NIDN : 197707222000031002

TUGAS/PERAN TIM PENELITIAN

NO	NAMA PENELITIAN	BIDANG KEAHLIAN	TUGAS/PERAN DALAM PENELITIAN
1	Setia Pramana, S.Si., Ph.D.	Machine Learning	Ketua Penelitian
2	Firdaus, MBA	Sistem Informasi	Pengolahan Data

LUARAN DAN TARGET CAPAIAN

NO	JENIS LUARAN	TARGET CAPAIAN	JURNAL/KONFERENS YANG DITUJU*)
1	Jurnal Penelitian Nasional	Seminar Nasional	Jurnal Sinta S6

*) tentative

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

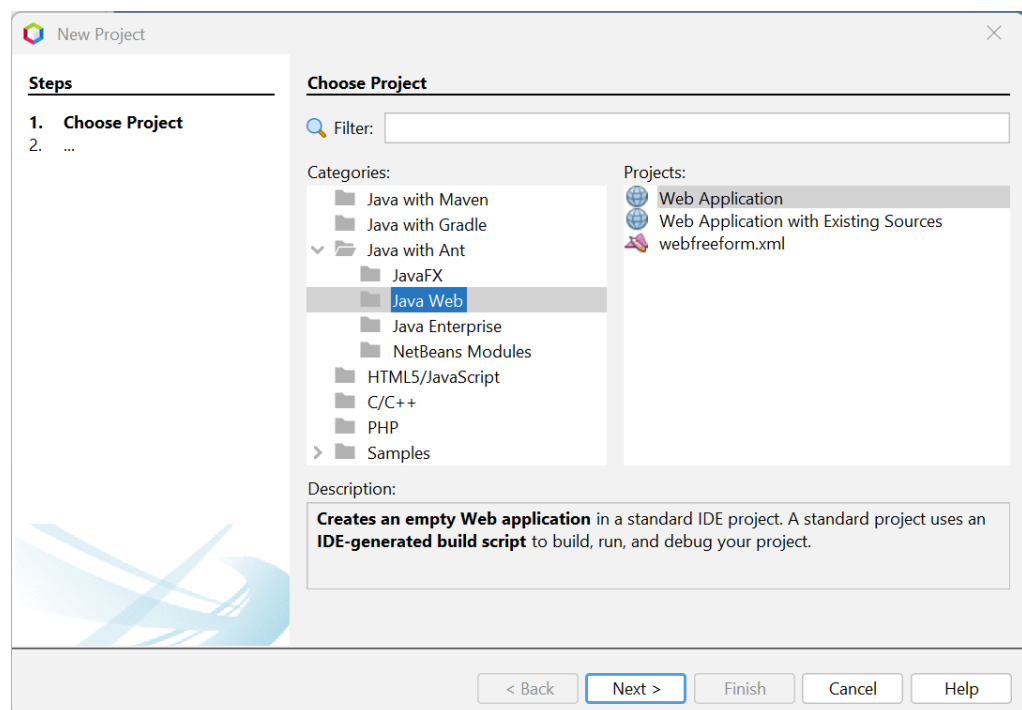
3.3. Material Praktikum

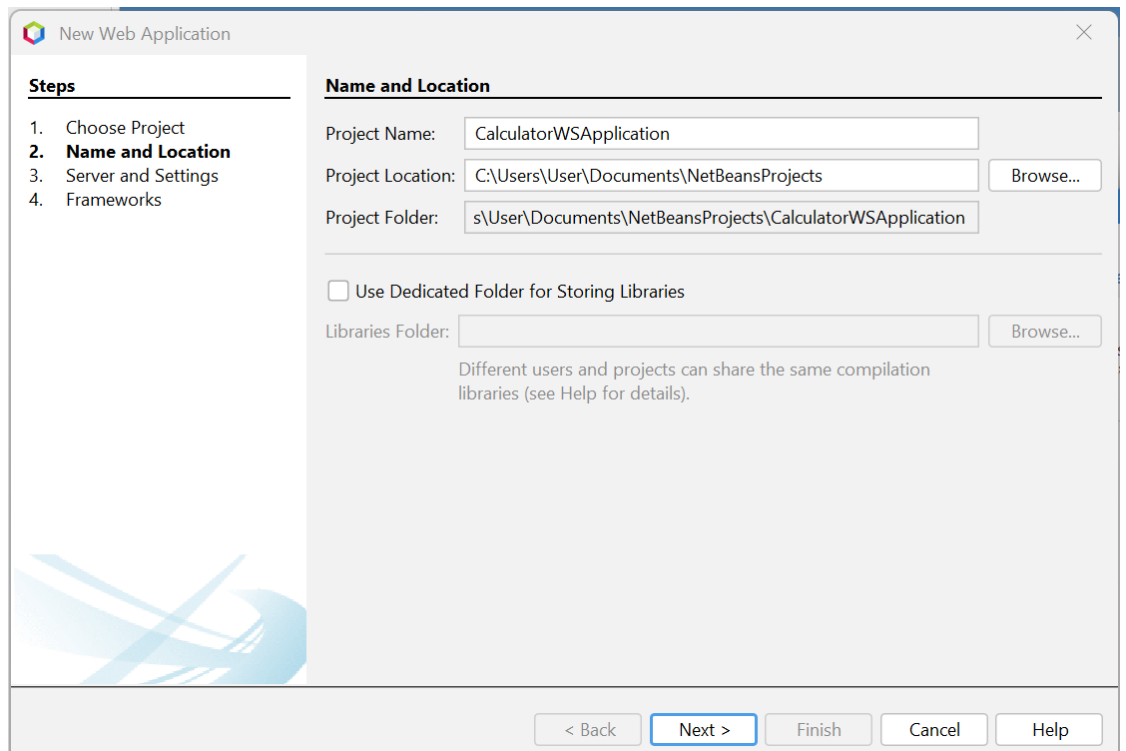
- Apache Netbeans 14
- Telah terinstall Tomcat Server (Dapat memanfaatkan XAMPP)

3.4. Kegiatan Praktikum

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

Buka Netbeans-> Java web -> web application





New Web Application

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

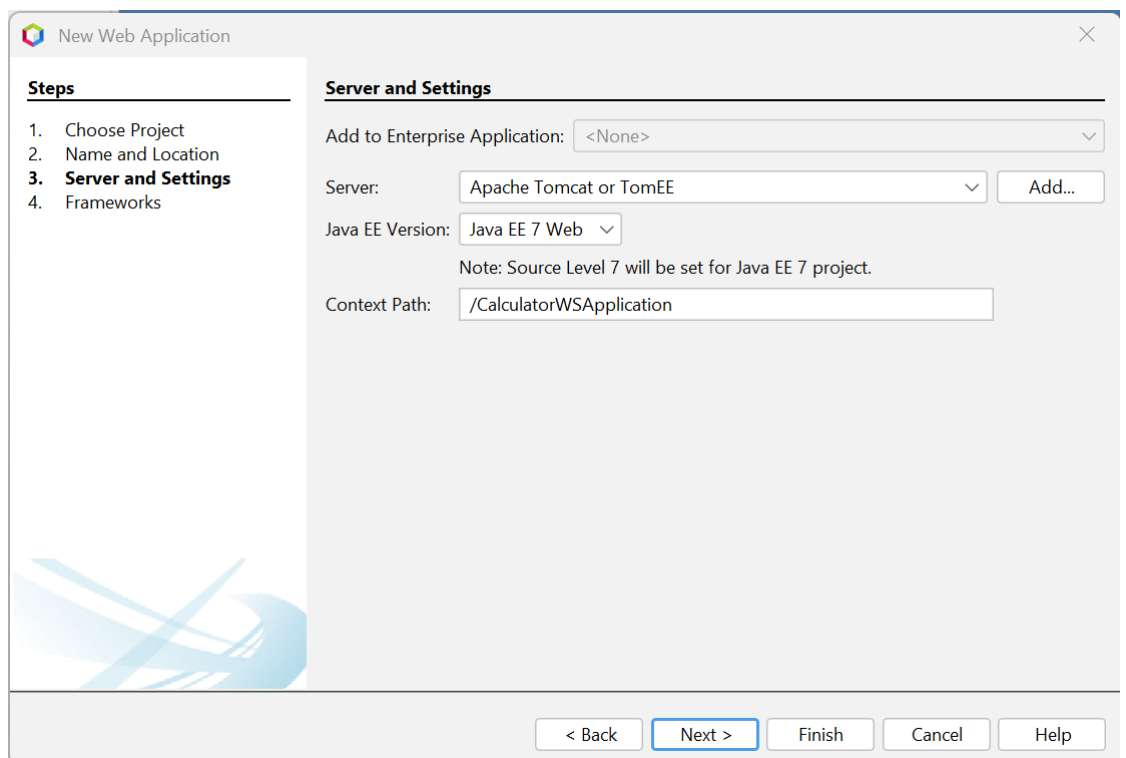
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

< Back **Next >** Finish Cancel Help

Beri nama project CalculatorWSApplication



New Web Application

Steps

1. Choose Project
2. Name and Location
- 3. Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application:

Server:

Java EE Version:

Note: Source Level 7 will be set for Java EE 7 project.

Context Path:

< Back **Next >** Finish Cancel Help

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 are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Add Server Instance

Steps

1. Choose Server
2. ...

Choose Server

Server: Amazon Beanstalk
Apache Tomcat or TomEE
GlassFish Server
Payara Server
WildFly Application Server

Name: Apache Tomcat or TomEE (1)

< Back Next > Finish Cancel 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 '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', a 'Browse ...' button, a checkbox for 'Use Private Configuration Folder (Catalina Base)', a 'Catalina Base:' label and a text field, another 'Browse ...' button, a section for 'Enter the credentials of an existing user in the manager or manager-script role' with 'Username:' and 'Password:' labels and text fields, and a checked checkbox for 'Create user if it does not exist'.

Add Server Instance

Steps

1. Choose Server
2. Installation and Login Details

Installation and Login Details

Specify the Server Location (Catalina Home) and login details

Server Location: C:\xampp\tomcat Browse ...

☐ Use Private Configuration Folder (Catalina Base)

Catalina Base: Browse ...

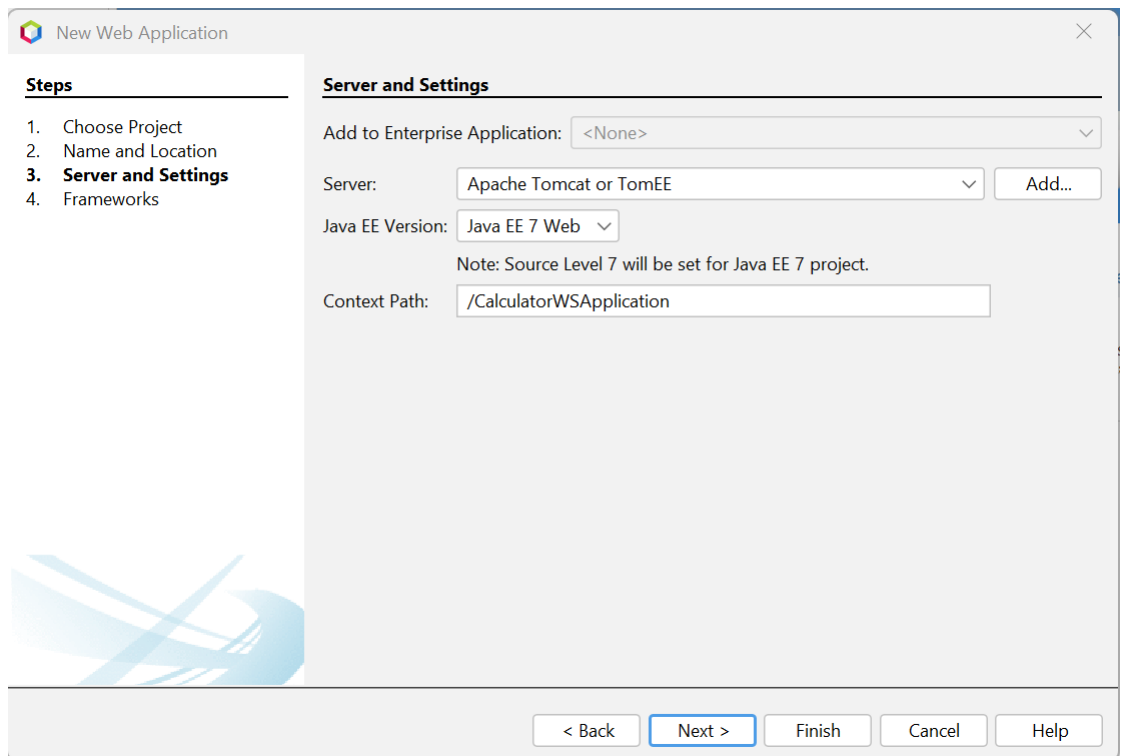
Enter the credentials of an existing user in the manager or manager-script role

Username: admin

Password:

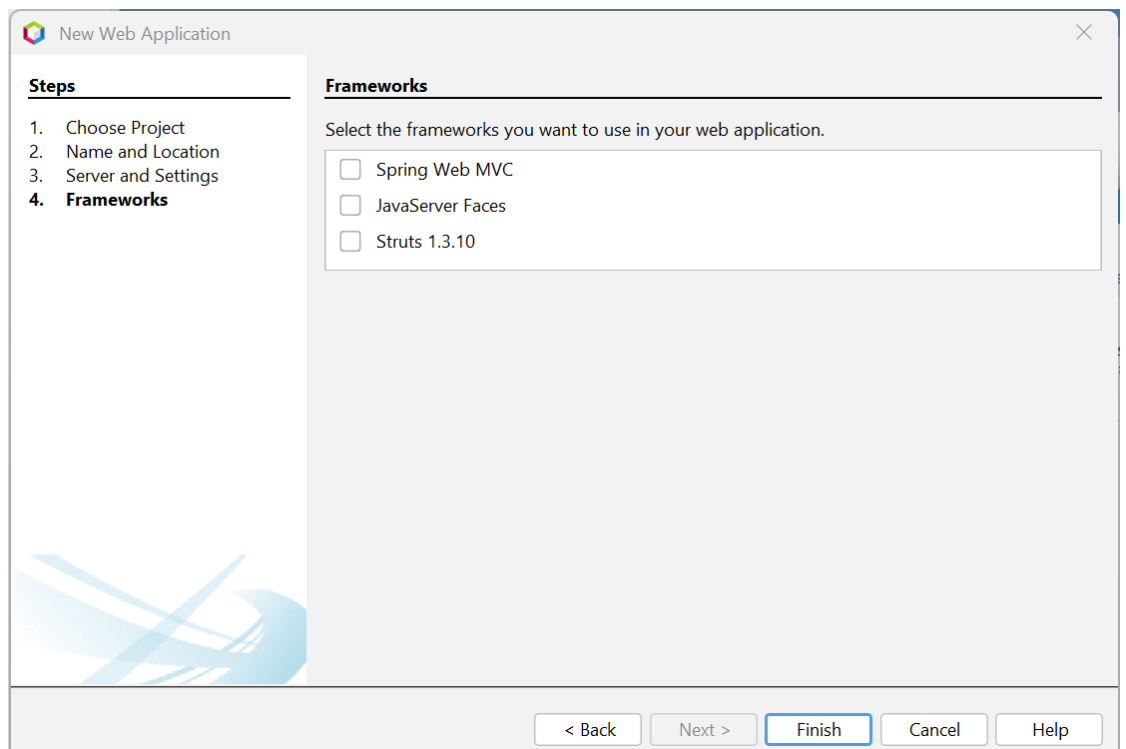
☒ Create user if it does not exist

Jika sudah maka kita klik next.



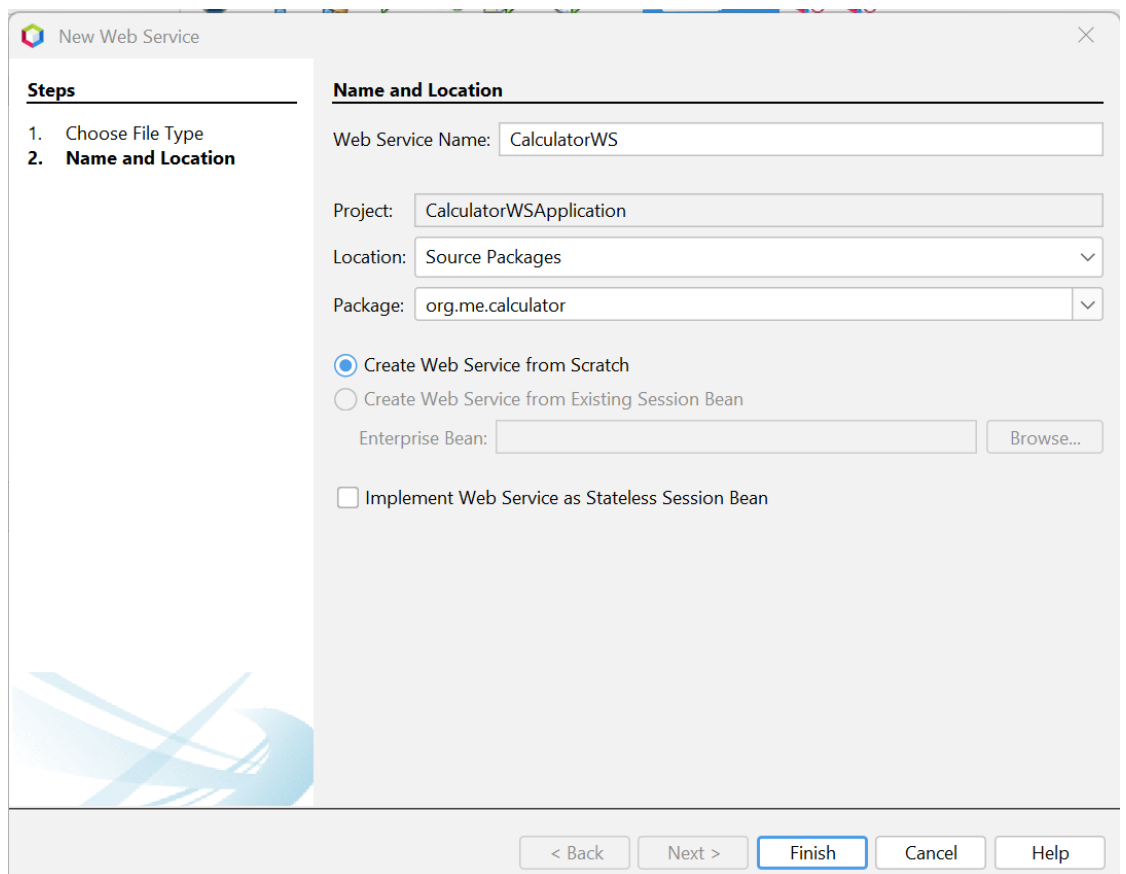
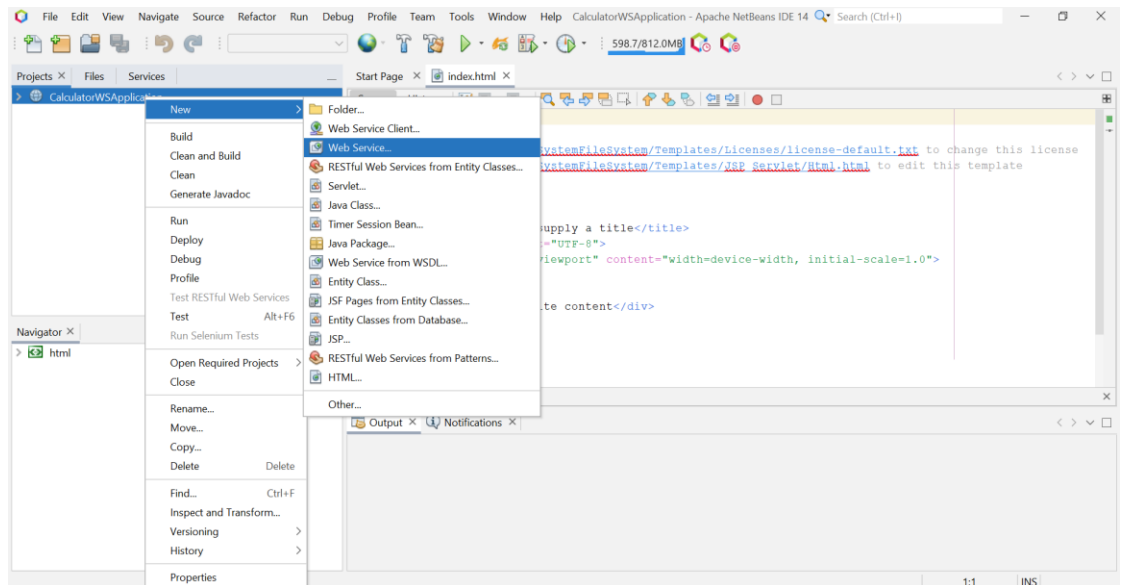
The screenshot shows the 'New Web Application' wizard at step 3, 'Server and Settings'. On the left, a 'Steps' list shows: 1. Choose Project, 2. Name and Location, 3. **Server and Settings**, and 4. Frameworks. The main area contains the following settings: 'Add to Enterprise Application' is set to '<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, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

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

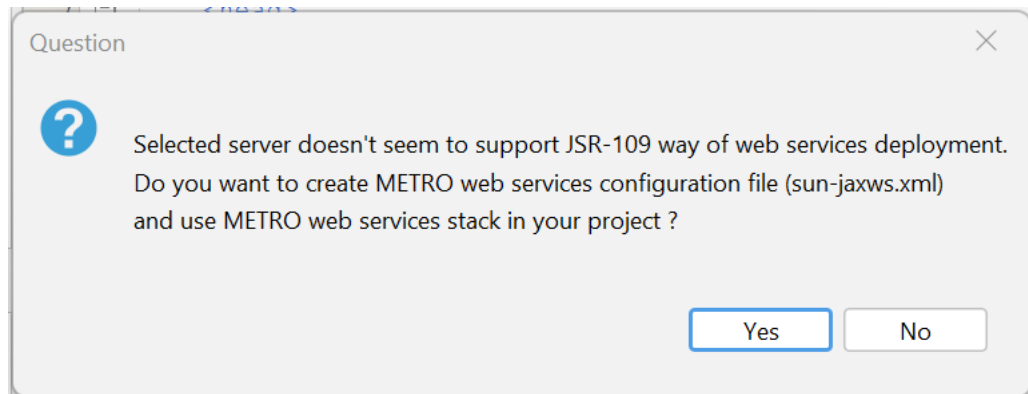


The screenshot shows the 'New Web Application' wizard at step 4, 'Frameworks'. On the left, the 'Steps' list shows: 1. Choose Project, 2. Name and Location, 3. Server and Settings, and 4. **Frameworks**. The main area has the title 'Frameworks' and the instruction 'Select the frameworks you want to use in your web application.' Below this is a list of three frameworks, each with an unchecked checkbox: 'Spring Web MVC', 'JavaServer Faces', and 'Struts 1.3.10'. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Finish' button is highlighted with a blue border.

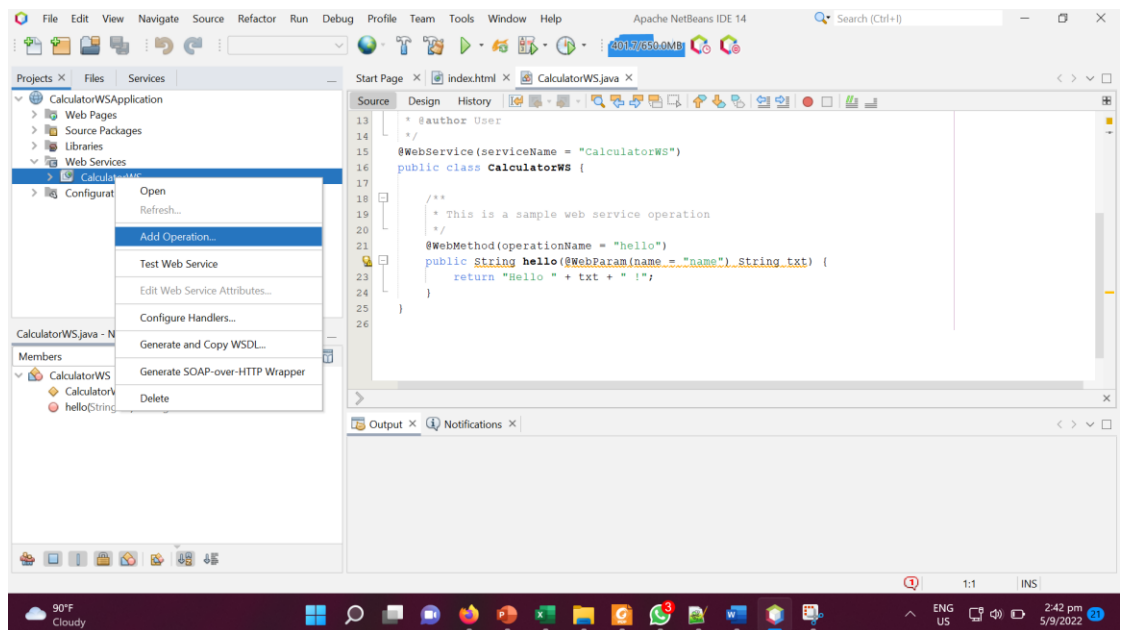
Buatlah Web Service dengan cara klik kanan folder project dan tambahkan web service.



Misalnya namakan web service tersebut sebagai CalculatorWS kemudian klik finish.



Klik yes jika muncul notifikasi seperti di atas



Klik kanan CalculatorWS, Add Operation

Add Operation

Name:

Return Type:

Parameters Exceptions

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

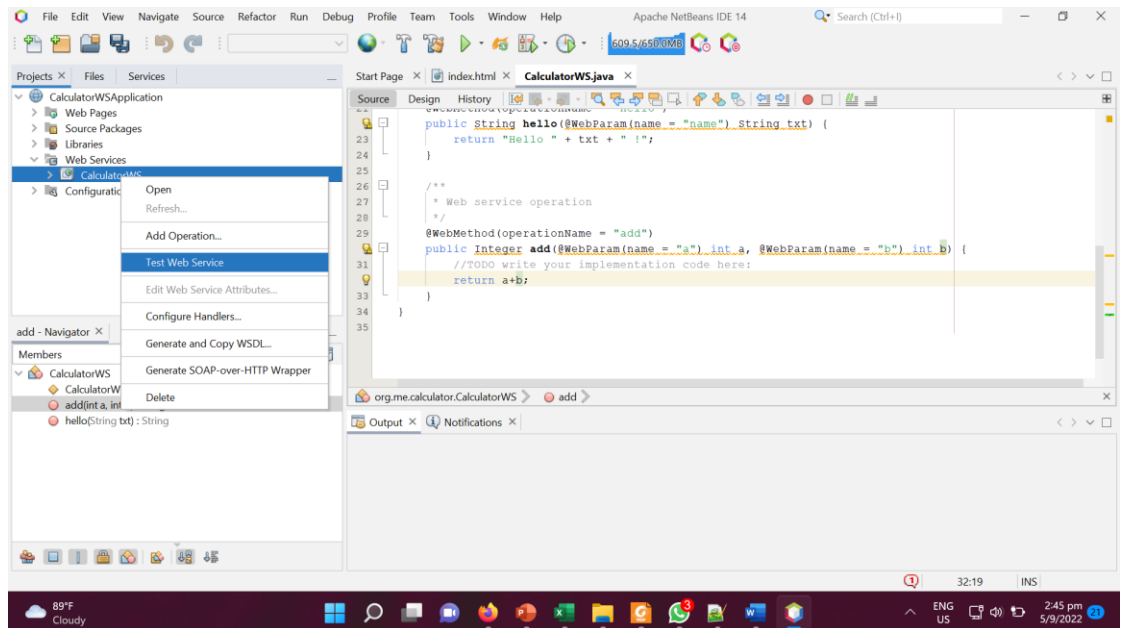
Tambahkan operasi baru dengan nama add bertipe Integer, parameter a dan b bertipe int

```

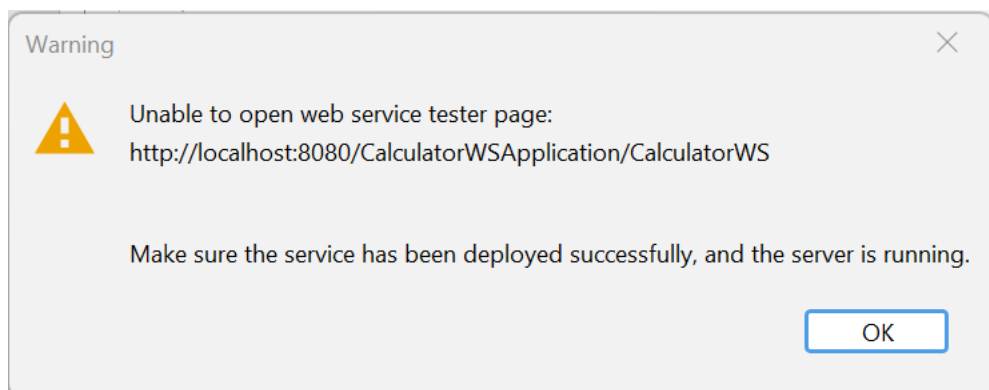
21  @WebMethod(operationName = "hello")
22  public String hello(@WebParam(name = "name") String txt) {
23      return "Hello " + txt + " !";
24  }
25
26  /**
27   * Web service operation
28   */
29  @WebMethod(operationName = "add")
30  public Integer add(@WebParam(name = "a") int a, @WebParam(name =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

```

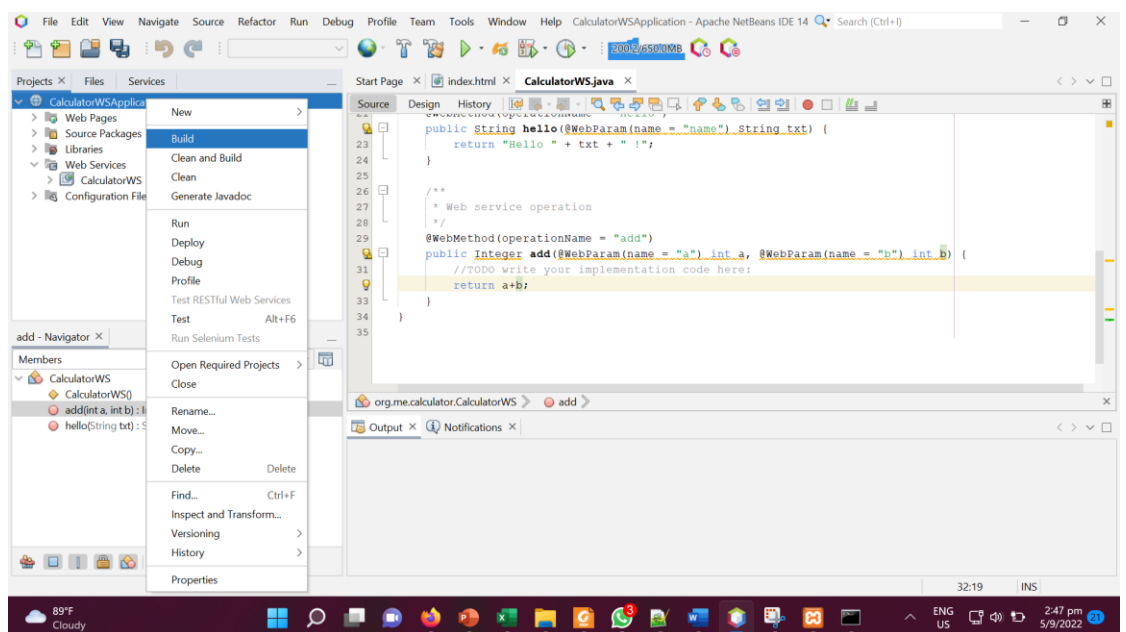
Ubah menjadi return a+b;



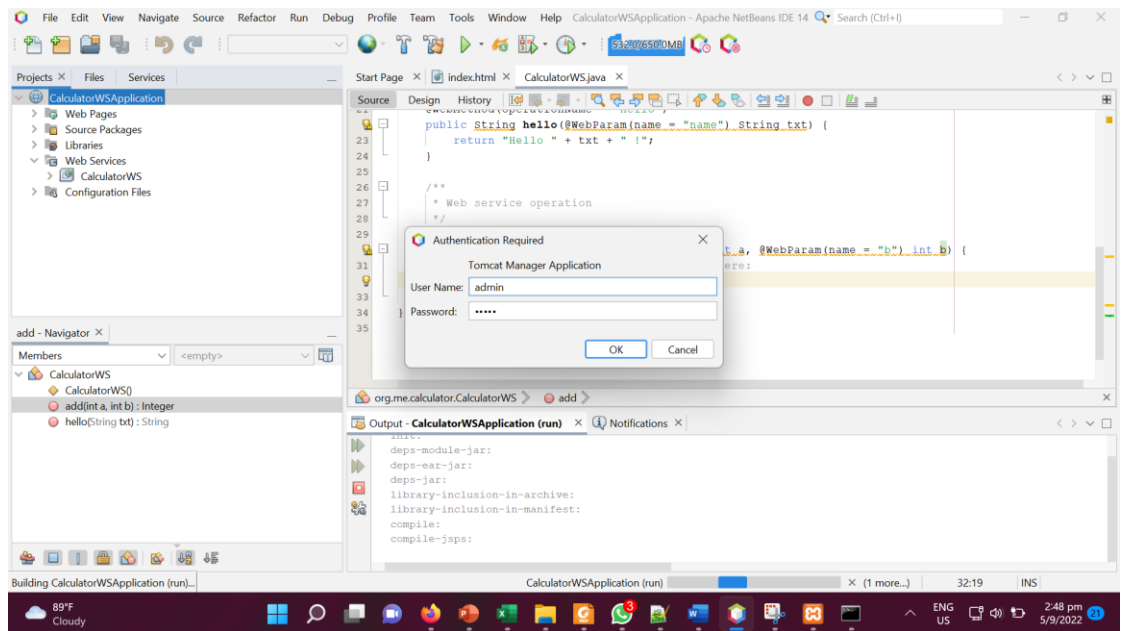
Test web service



Muncul error karena belum di-run



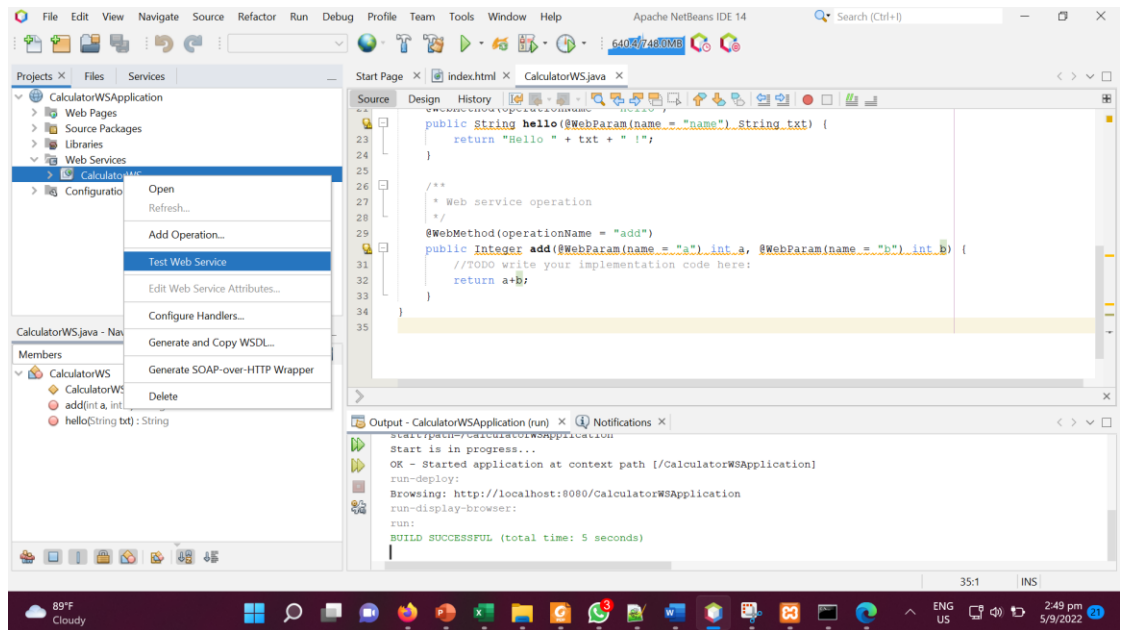
Build, kemudian Run



Masukkan username admin dan password admin, OK

Akan muncul halaman web





Test web service



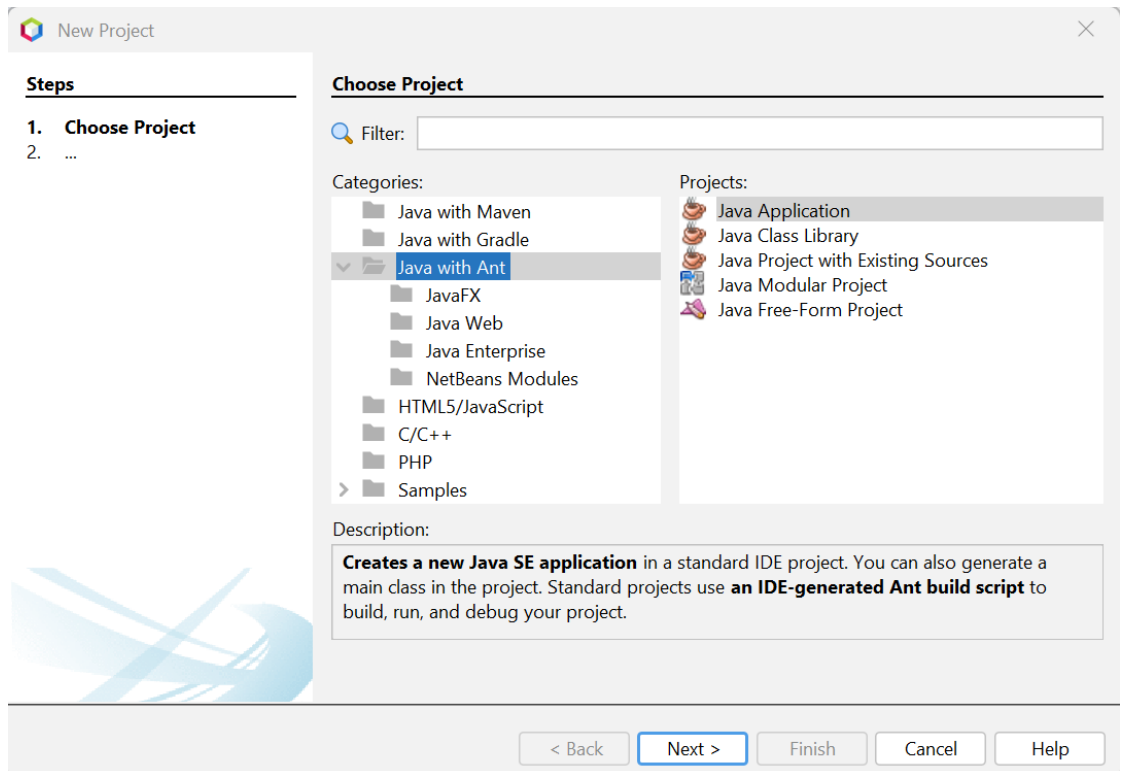
Web Services

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

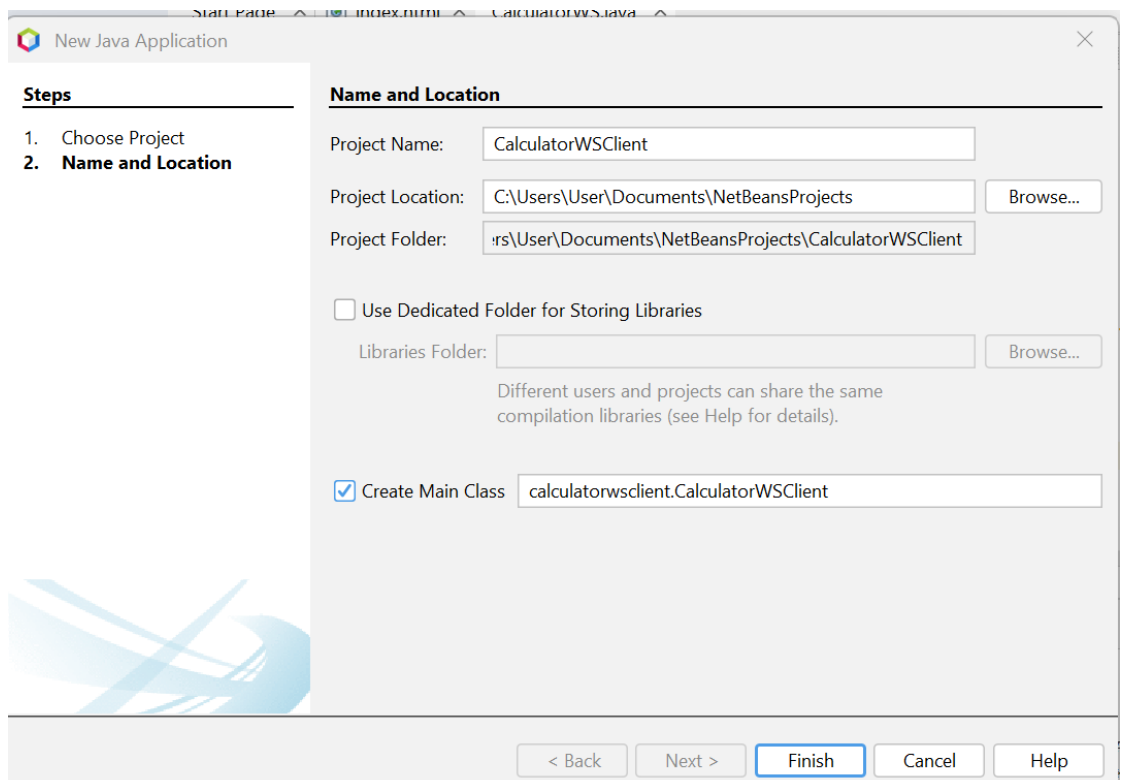
Muncul halaman WSDL yang nanti bisa dikonsumsi oleh aplikasi consumer

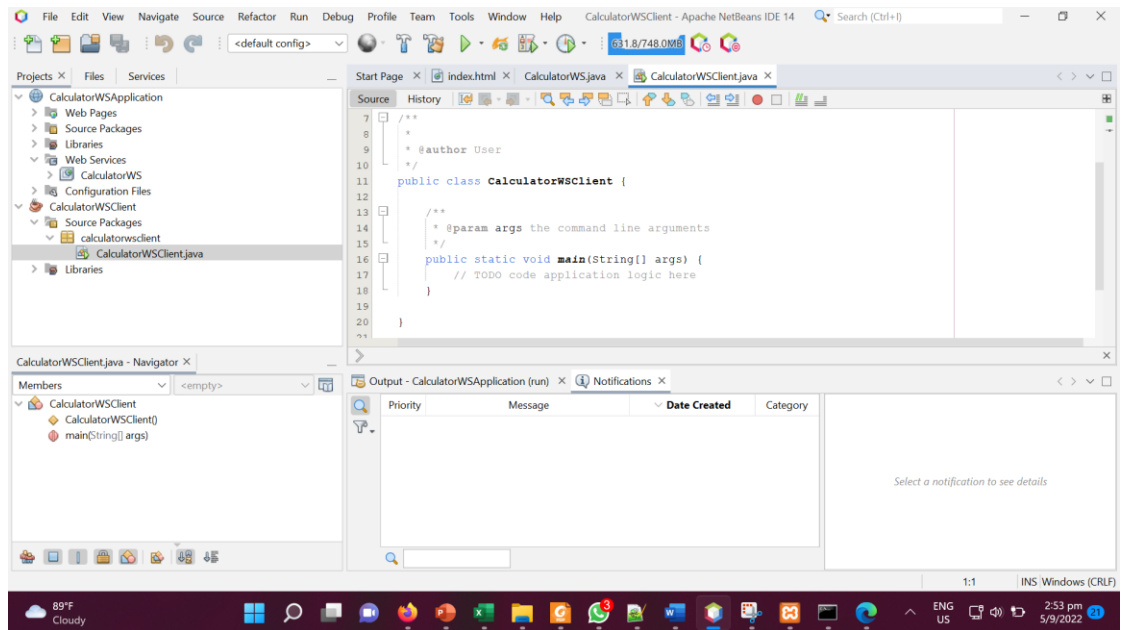
MENGKONSUMSI WEB SERVICE

Buatlah project baru untuk mengkonsumsi web service

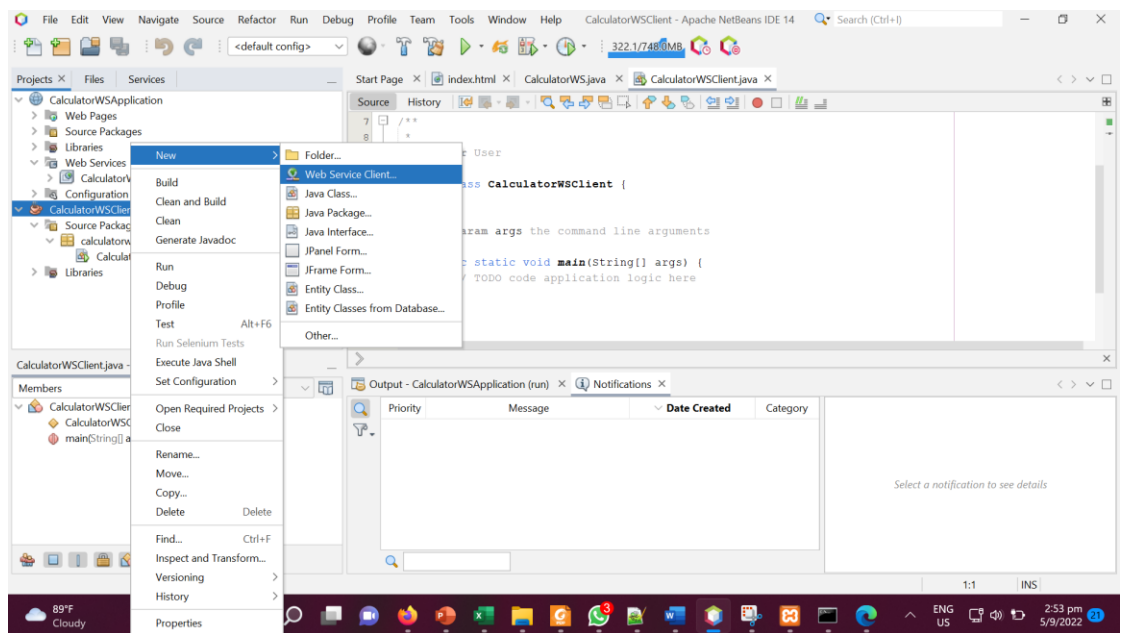


Beri nama CalculatorWSClient

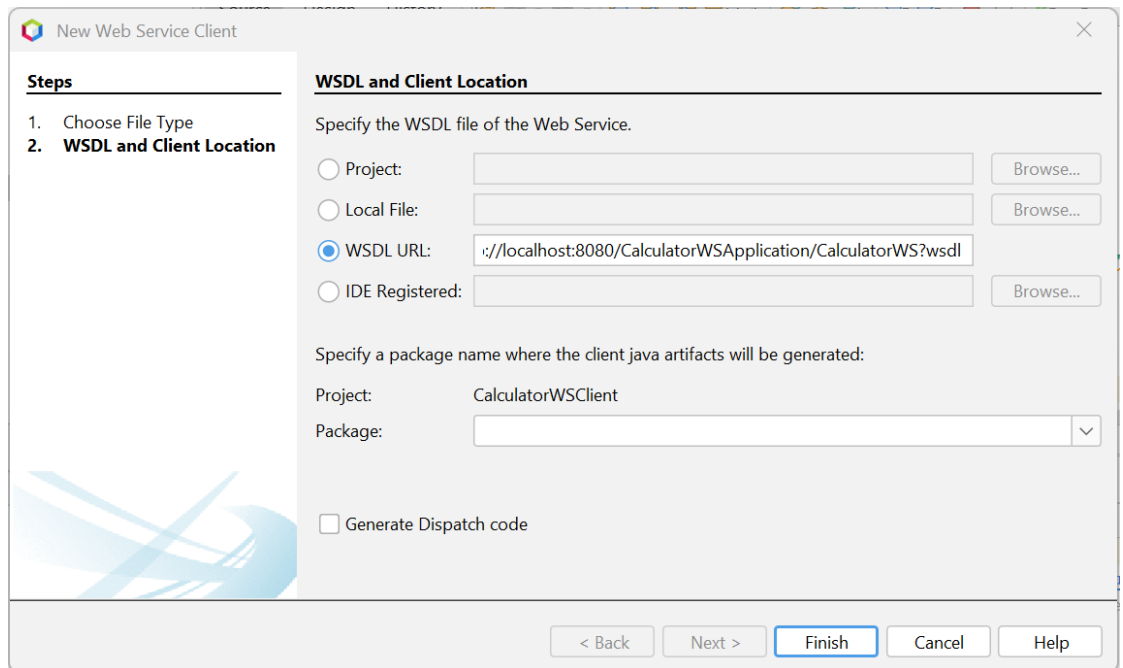




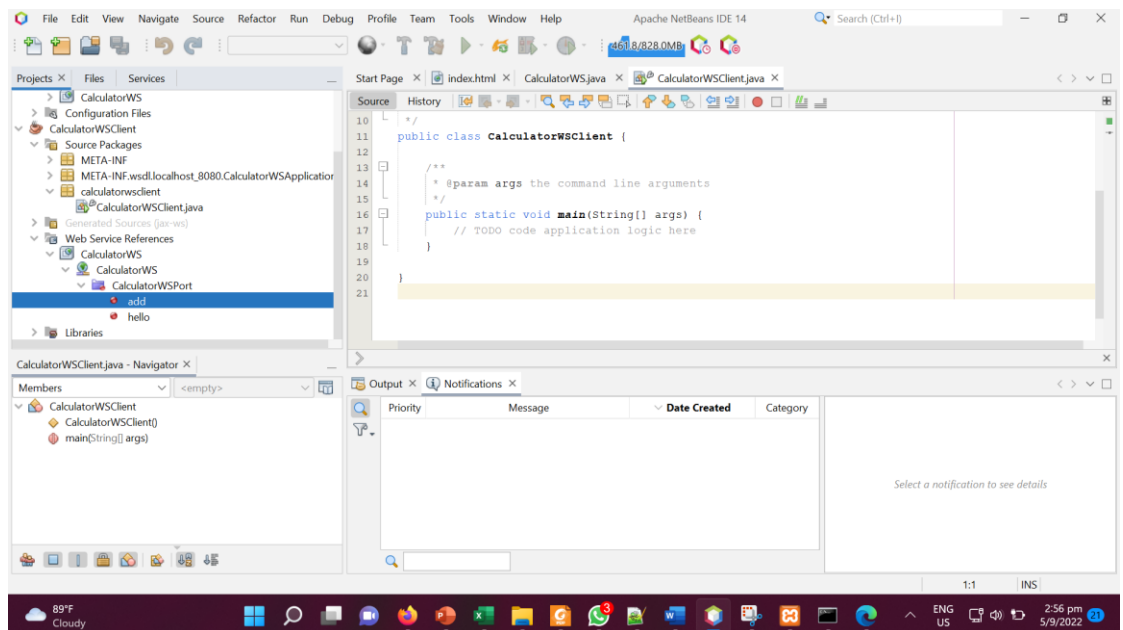
Pada project, klik kanan, new web service client



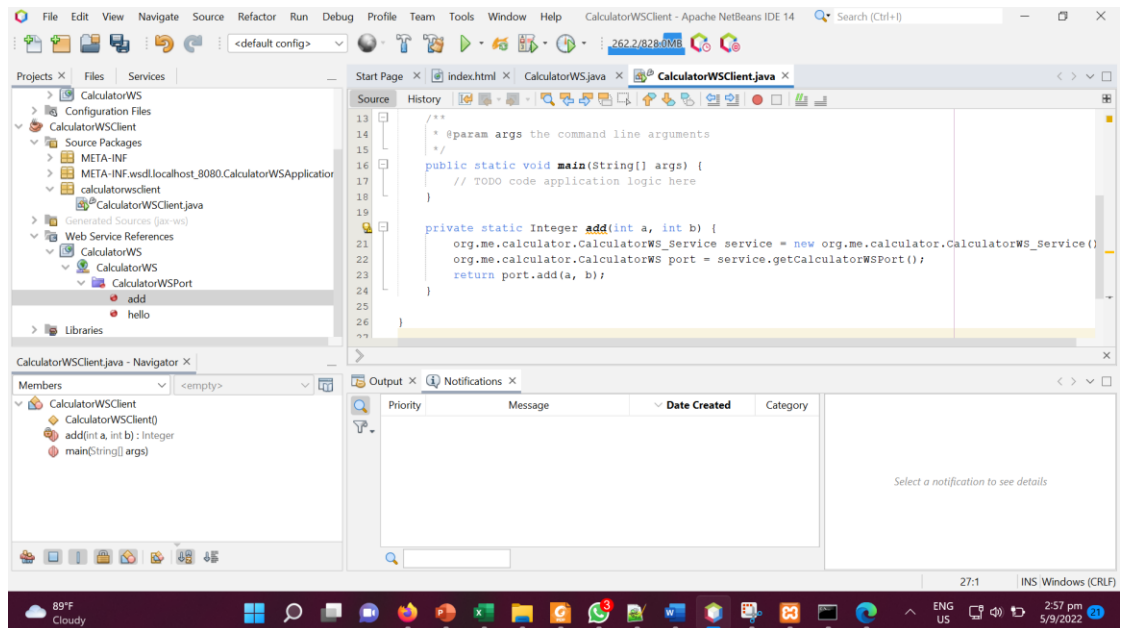
Bisa memilih project atau menambahkan wsdl URL sesuai di halaman sebelumnya



Klik Finish



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

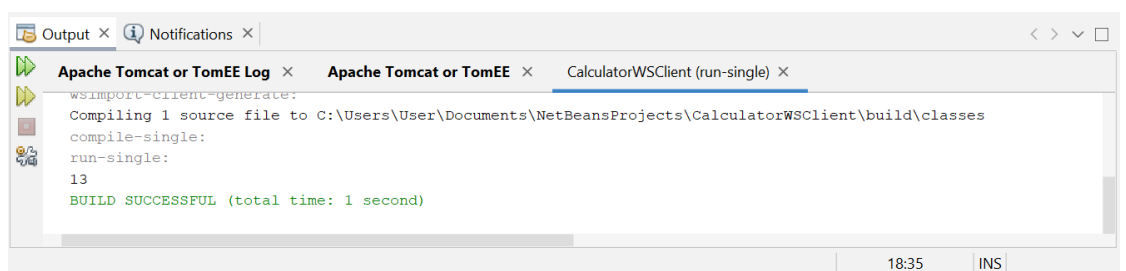


Ubah pada method main untuk memanggil add

```
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.