

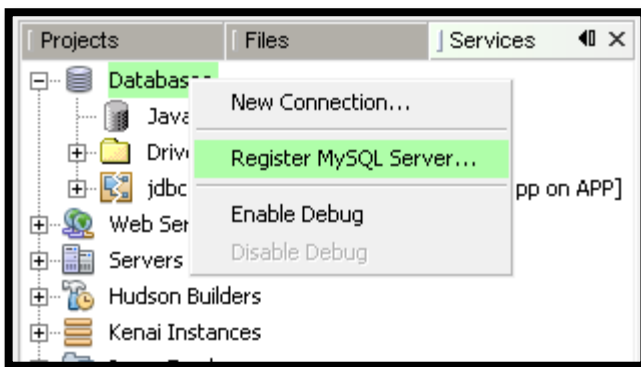
TUTORIAL JSF, JPA, & MYSQL

Waktu: 100 menit

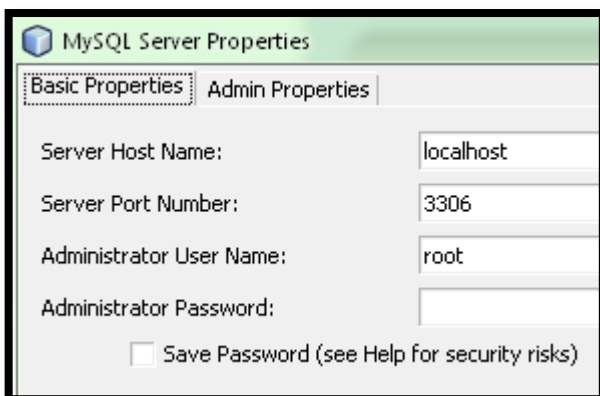
A. LANGKAH 1: Register MYSQL Server

Info Lengkap: <http://netbeans.org/kb/docs/ide/mysql.html>

1. Bukalah tab Services, dan klik kanan icon "Databases", pilih "Register MySQL Server"



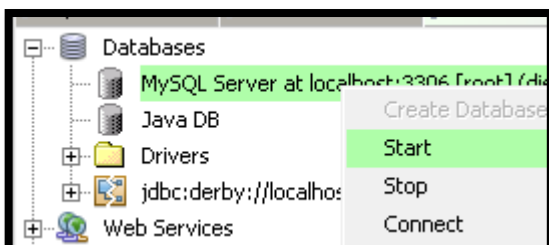
2. Isikan Basic Properties sbb:



3. Klik kanan MySQL Server, pilih Start

Note: langkah 3 dan 4 tidak perlu dilakukan apabila MYSQL sudah RUNNING via XAMPP

Tujuan langkah 3 & 4 ini adalah supaya kita bisa start/stop MySQL lewat NetBeans (sebagai alternative dari XAMPP Control panel).

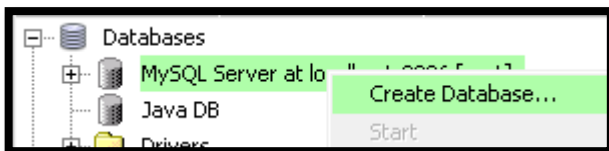


4. Isikan "Admin Properties" sbb:

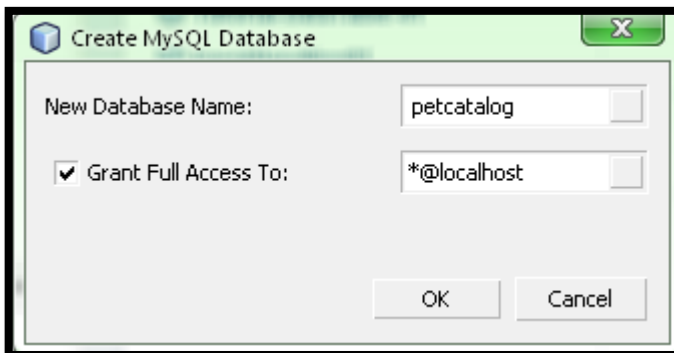


Note: Lokasi "c:\webserver" harus diubah tergantung lokasi dari database mysql yang diinstal di komputer Anda (misal: di lab FIT, kita biasanya menggunakan **c:\xampp**)

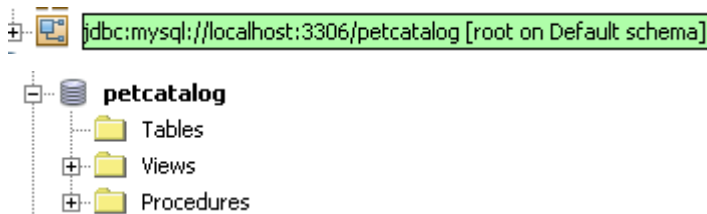
5. Klik kanan MySQL Server, pilih "Create Database..."



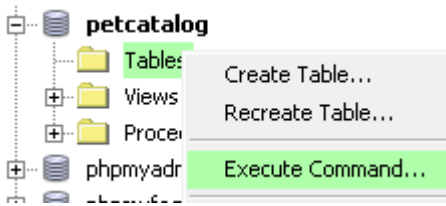
6. Database Name: "petcatalog"



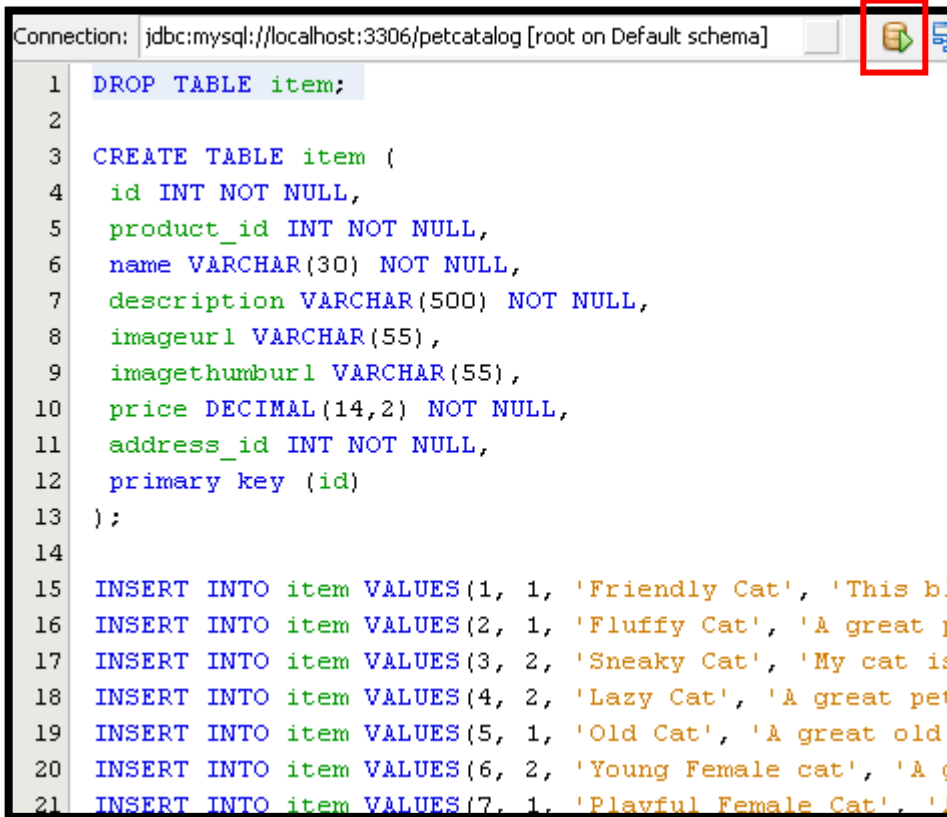
7. NetBeans akan membuat connection sbb:



8. Klik kanan “Tables”, lalu pilih “Execute Command...”



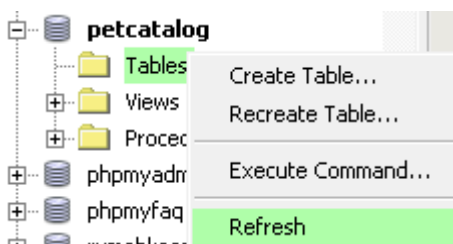
9. Copy & Paste isi dari **catalog.sql** yang telah disediakan, lalu klik icon execute

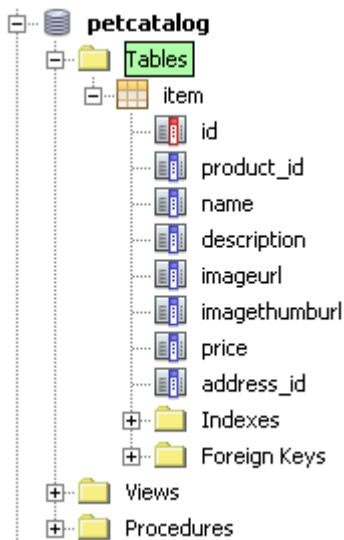


Apabila setelah dieksekusi muncul sebuah error sbb, maka abaikan saja karena memang tabel ‘item’ belum ada sebelumnya sehingga tidak bisa di-drop.

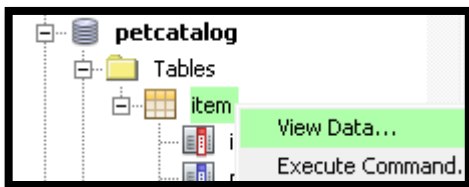
```
Error code 1051, SQL state 42S02: Unknown table 'item'
Line 1, column 1
```

10. Pilih “Refresh” untuk melihat tabel yang telah dibuat tersebut





11. Klik kanan tabel “item” dan pilih “View Data...” untuk melihat isi dari tabel tersebut:



Start Page x SQL Command 1 x

Connection: jdbc:mysql://localhost:3306/petcatalog [root on Default schema]

```
1 select * from item
```

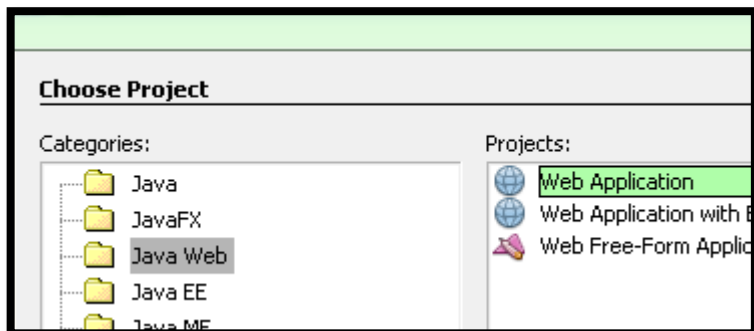
select * from item x

Page Size: 20 | Total Rows: 29 | Page: 1 of 2

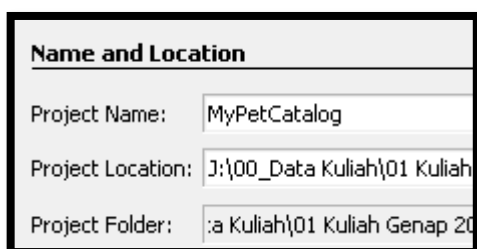
#	id	product_id	name	description	
1	1	1	Friendly Cat	This black and white colored cat is super fri...	anthony.jpg
2	2	1	Fluffy Cat	A great pet for a hair stylist! Have fun com...	bailey.jpg
3	3	2	Sneaky Cat	My cat is so sneaky. He is so curious that h...	bob.jpg
4	4	2	Lazy Cat	A great pet to lounge on the sofa with. If y...	chantelle.jpg
5	5	1	Old Cat	A great old pet retired from duty in the circ...	charlie.jpg
6	6	2	Young Female cat	A great young pet to chase around. Loves ...	elkie.jpg

B. LANGKAH 2: MEMBUAT APLIKASI WEB (JSF 2)

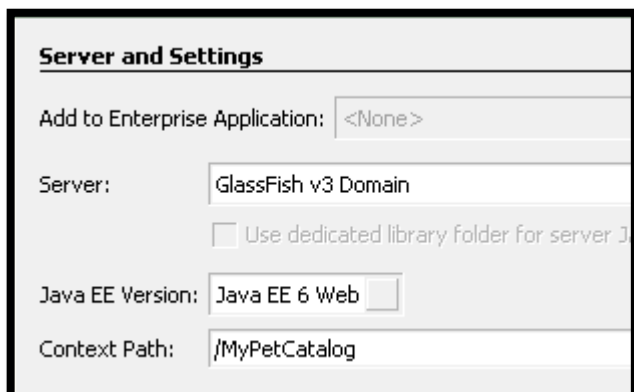
12. Selanjutnya seperti biasa kita akan membuat aplikasi web yang baru:



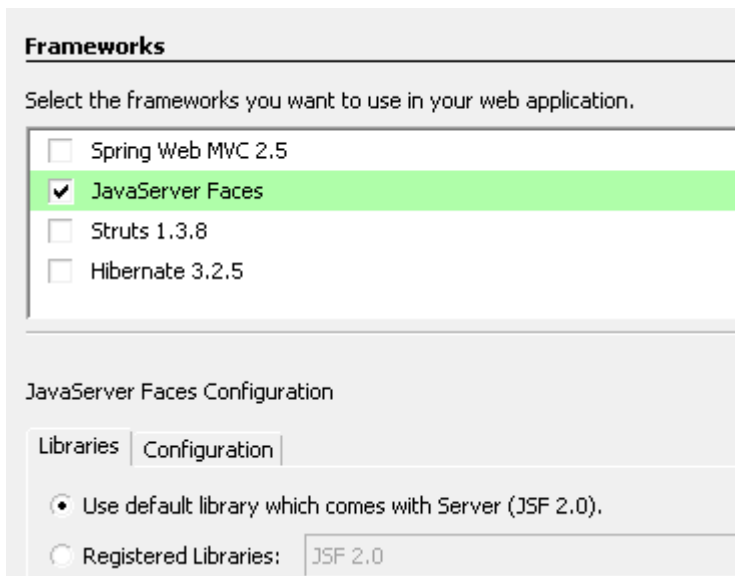
13. Isikan nama project "MyPetCatalog"



14. Berikut setting dan server yang digunakan:



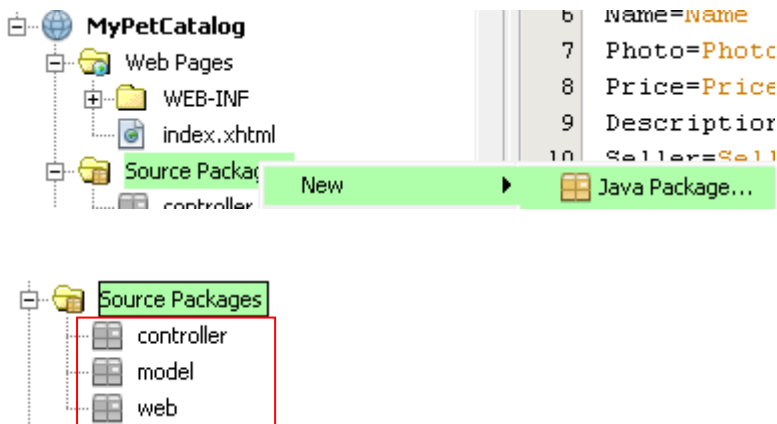
15. Untuk Frameworks, kita gunakan “JavaServer Faces” dan library “JSF 2.0”



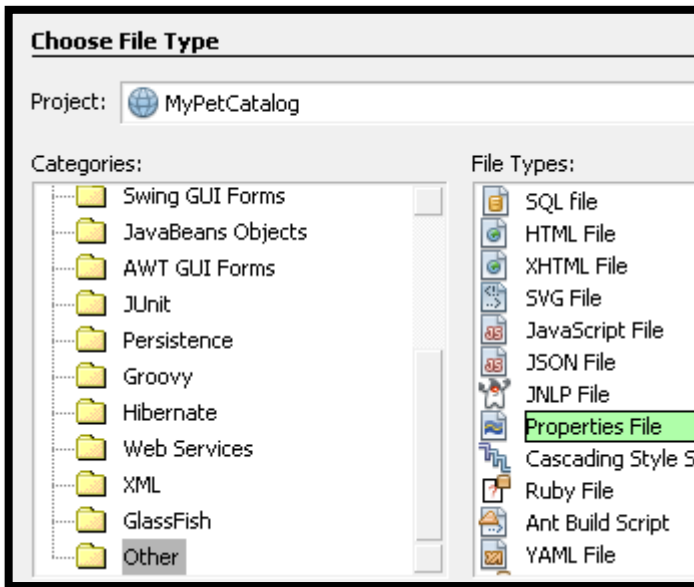
16. Cobalah Run project untuk memastikan bahwa semua settingan sudah ok.

C. Membuat Package dan File Properties

17. Buatlah 3 buah package baru pada source sbb:



18. Pada package “web”, buatlah sebuah “properties file” yang baru:



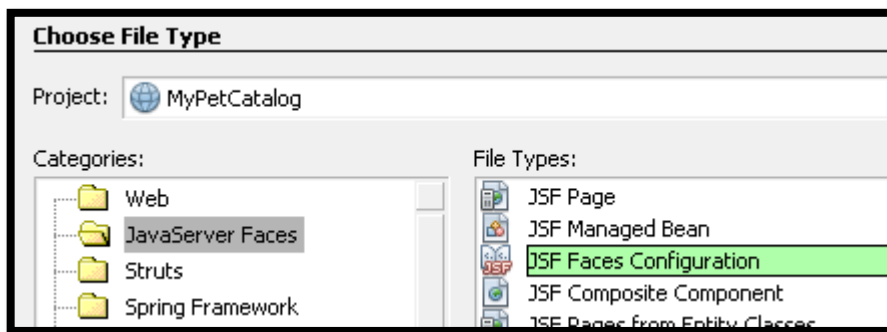
19. Berikut nama dan lokasi dari “messages”

Name and Location	
File Name:	messages
Project:	MyPetCatalog
Folder:	src\java\web

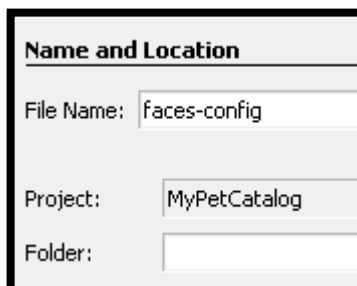
20. Isilah “messages.properties” sebagai berikut:

```
Title=Katalog Binatang Kesayangan
Next=Selanjutnya
Previous=Sebelumnya
Name>Nama
Photo=Foto
Price=Harga
Description=Keterangan
Seller=Lokasi Penjual
```

21. Di dalam folder "WEB-INF" buatlah sebuah file konfigurasi baru:



22. Kita beri nama "faces-config" seperti biasanya



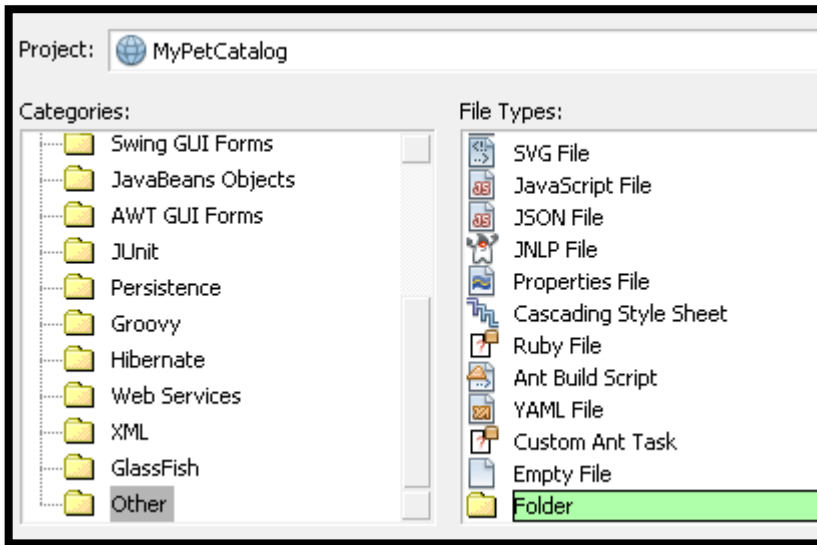
23. Tambahkan resource bundle agar "messages.properties" bisa dikenali di halaman JSF nantinya

```
<faces-config version="2.0"
  xmlns="http://java.sun.com/xml/ns/
  xmlns:xsi="http://www.w3.org/2001/
  xsi:schemaLocation="http://java.su

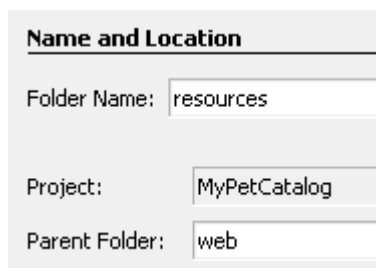
  <application>
    <resource-bundle>
      <base-name>web.messages</base-name>
      <var>msgs</var>
    </resource-bundle>
  </application>
```

D. Menambahkan Folder Gambar dan CSS

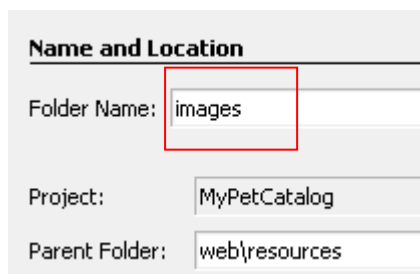
24. Di dalam folder “web pages”, buat subfolder yang baru:



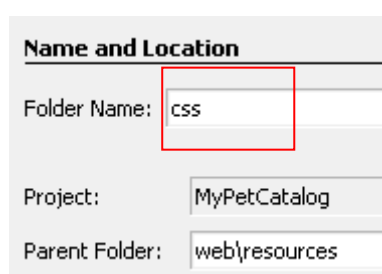
25. Beri nama “resources”



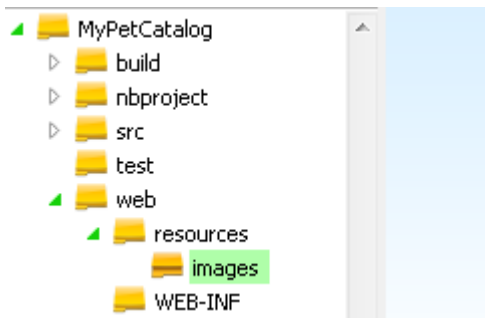
26. Buat 2 buah subfolder



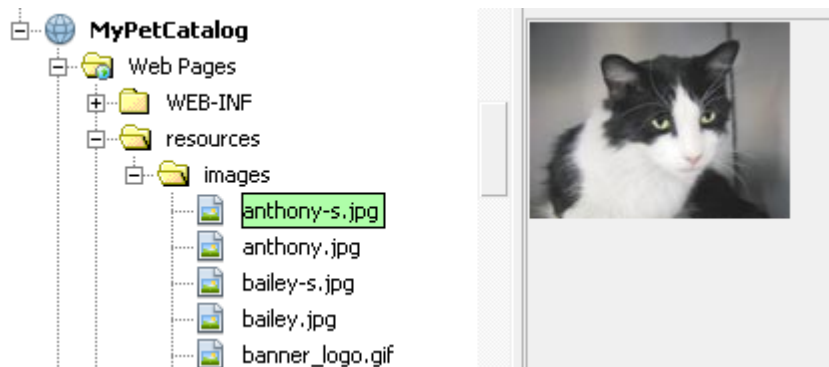
dan



27. Dengan Window explorer, copy-kan **semua gambar** ke folder “images” dan copykan file **style.css** ke dalam folder “css”



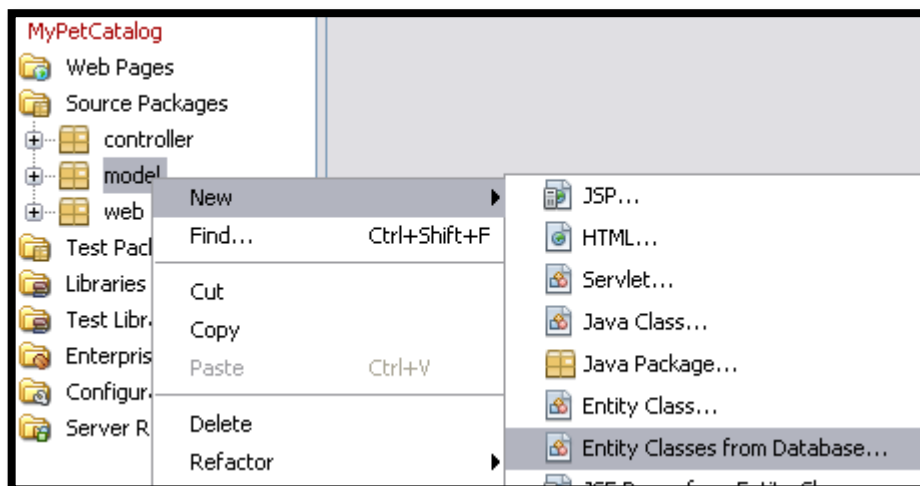
Hasil:



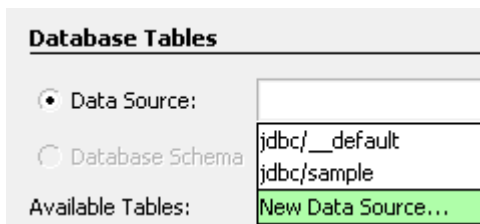
E. Membuat EntityBean, SessionBean dan PagingInfo Class

28. Buatlah Entity Class dari Database:

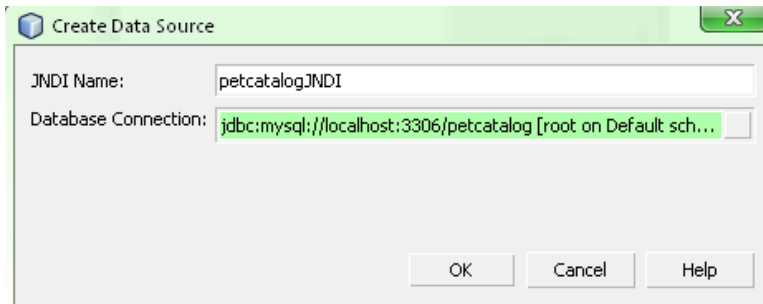
29.



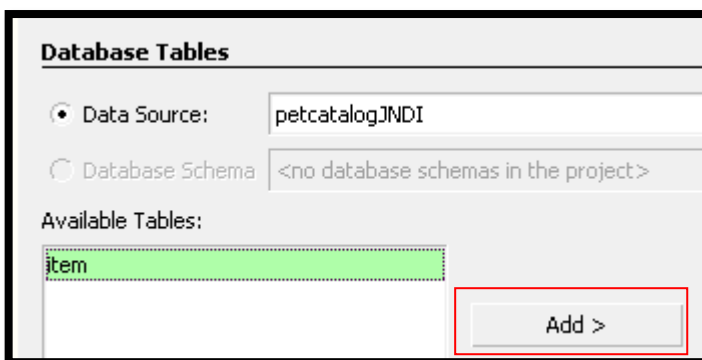
30. Untuk Data Source pilihlah “New Data Source...”



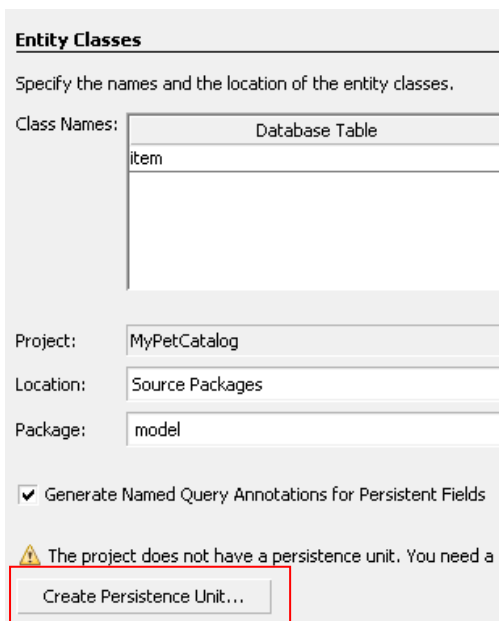
31. Isikan data sbb: (JNDI Name: “petcatalogJNDI”)



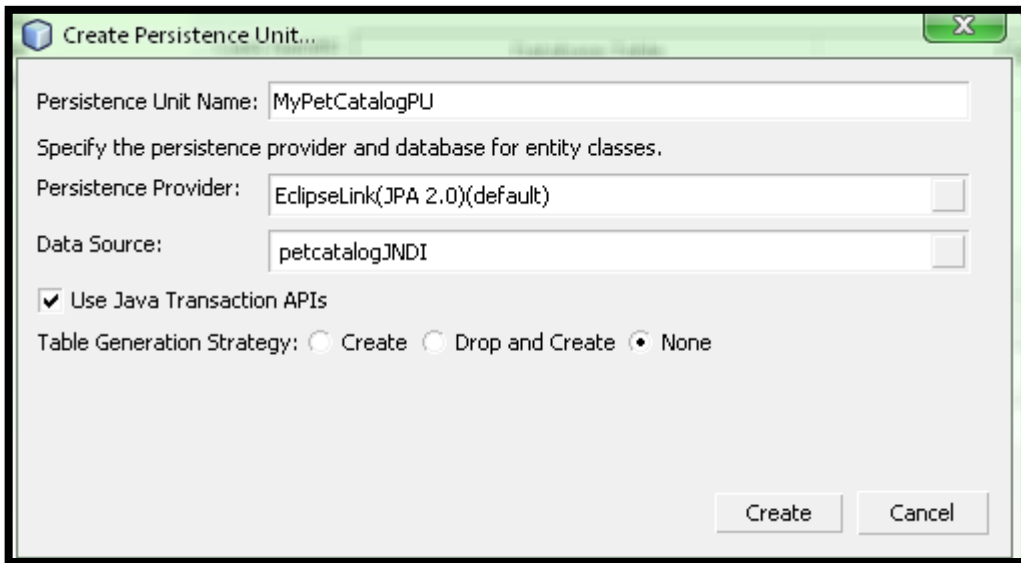
32. Pilih “Item” dan klik tombol “Add”



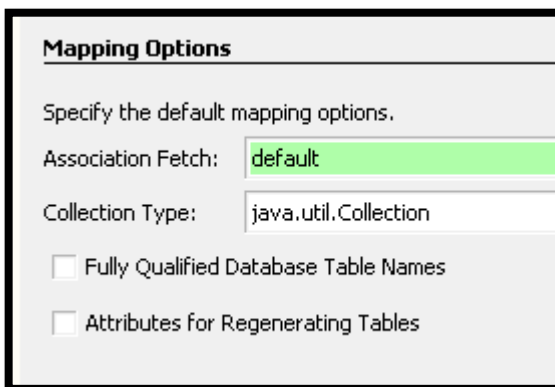
33. Klik tombol “Create Persistence Unit”:



34. Isikan Data sbb:



35. Selanjutnya isilah Mapping Options default sbb:



36. Setelah selesai, maka akan dibuat Entity Bean bernama Item.java sbb. Silahkan dibuka untuk melihat isi dari Entity Bean tersebut.

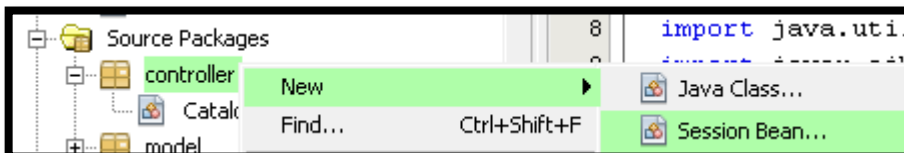


```

22  @Entity
23  @Table(name = "item")
24  @NamedQueries({
25      @NamedQuery(name = "Item.findAll", query = "SELECT i FROM Item i"),
26      @NamedQuery(name = "Item.findById", query = "SELECT i FROM Item i WHERE i.id = :id"),
27      @NamedQuery(name = "Item.findByProductId", query = "SELECT i FROM Item i WHERE i.productId = :productId"),
28      @NamedQuery(name = "Item.findByName", query = "SELECT i FROM Item i WHERE i.name = :name"),
29      @NamedQuery(name = "Item.findByDescription", query = "SELECT i FROM Item i WHERE i.description = :description"),
30      @NamedQuery(name = "Item.findByImageUrl", query = "SELECT i FROM Item i WHERE i.imageUrl = :imageUrl"),
31      @NamedQuery(name = "Item.findByImageThumburl", query = "SELECT i FROM Item i WHERE i.imageThumburl = :imageThumburl"),
32      @NamedQuery(name = "Item.findByPrice", query = "SELECT i FROM Item i WHERE i.price = :price"),
33      @NamedQuery(name = "Item.findByAddressId", query = "SELECT i FROM Item i WHERE i.addressId = :addressId")
34  })
35  public class Item implements Serializable {
36      private static final long serialVersionUID = 1L;
37      @Id
38      @Basic(optional = false)
39      @Column(name = "id")
40      private Integer id;
41      @Basic(optional = false)
42      @Column(name = "product_id")
43      private int productId;
44      @Basic(optional = false)
45      @Column(name = "name")
46      private String name;

```

37. Buatlah sebuah java class baru yang berfungsi sebagai penghubung antara Entity Bean dengan Managed Bean. Penghubung ini biasa kita sebut dengan nama "session bean".



38. Beri nama "ItemFacade"

Name and Location

EJB Name:

Project:

Location:

Package:

Session Type:

Stateless

Stateful

Singleton

Create Interface:

Remote

Local

Session bean akan berisikan semua business methods (fungsi-fungsi, logic, database access) yang nantinya akan diakses/panggil oleh Managed Bean.

39. Tambahkan "implements Serializable" pada nama class tersebut:

```
1 package controller;
2
3
4 import javax.ejb.Stateless;
5
6 @Stateless
7 public class ItemFacade implements Serializable{
8
9     // Add business logic below. (Right-click in editor
10    // "Insert Code > Add Business Method")
11
12 }
```

40. Import class Serializable tersebut:

```
6 public class ItemFacade implements
8 Add import for java.io.Serializable
```

41. Definiskan sebuah entity manager pada class tersebut dan beri referensi kepada persistence unit yang sudah kita buat sebelumnya (yaitu "MyPetCatalogPU"), sbb:

```
@PersistenceContext(unitName = "MyPetCatalogPU")
private EntityManager em;
```

Lakukan import class seperlunya. Isi dari session bean sekarang bertambah menjadi sbb:

```
1 package controller;
2
3 import java.io.Serializable;
4 import javax.ejb.Stateless;
5 import javax.persistence.EntityManager;
6 import javax.persistence.PersistenceContext;
7
8 @Stateless
9 public class ItemFacade implements Serializable{
10
11     @PersistenceContext(unitName = "MyPetCatalogPU")
12     private EntityManager em;
13
14 }
```

42. Sekarang kita akan membuat tiga buah business methods pada session bean tersebut:

- findAll
- findRange
- getItemCount

```
public List<Item> findAll() {
    return em.createQuery("select object(o) from Item as o").getResultList();
}

public List<Item> findRange(int maxResults, int firstResult) {
    Query q = em.createQuery("select object(o) from Item as o");
    q.setMaxResults(maxResults);
    q.setFirstResult(firstResult);
    return q.getResultList();
}

public int getItemCount() {
    return ((Long) em.createQuery("select count(o) from Item as o").getSingleResult()).intValue();
}
```

Berikut tampilan dari ketiga business method tsb:

```
@Stateless
public class ItemFacade implements Serializable {

    @PersistenceContext(unitName = "MyPetCatalogPU")
    private EntityManager em;

    public List<Item> findAll() {
        return em.createQuery("select object(o) from Item as o").getResultList();
    }

    public List<Item> findRange(int maxResults, int firstResult) {
        Query q = em.createQuery("select object(o) from Item as o");
        q.setMaxResults(maxResults);
        q.setFirstResult(firstResult);
        return q.getResultList();
    }

    public int getItemCount() {
        return ((Long) em.createQuery("select count(o) from Item as o").getSingleResult()).intValue();
    }
}
```

Import dua class yang dibutuhkan berikut ini:

- import javax.persistence.Query;
- import model.Item;

43. Selanjutnya kita akan membuat **Java Class** baru di dalam **package controller** yang berfungsi untuk “paging” data di halaman Web dengan nama “**PagingInfo.java**”:

```
package controller;

import java.io.Serializable;

public class PagingInfo implements Serializable {

    private int batchSize = 5;
    private int firstItem = 0;
    private int itemCount = -1;

    public int getBatchSize() {
        return batchSize;
    }

    public int getItemCount() {
        return itemCount;
    }

    public void setItemCount(int itemCount) {
        this.itemCount = itemCount;
    }

    public int getFirstItem() {
        if (itemCount == -1) {
            throw new IllegalStateException("itemCount must be set before invoking getFirstItem");
        }
        if (firstItem >= itemCount) {
            if (itemCount == 0) {
                firstItem = 0;
            } else {
                int zeroBasedItemCount = itemCount - 1;
                double pageDouble = zeroBasedItemCount / batchSize;
                int page = (int) Math.floor(pageDouble);
                firstItem = page * batchSize;
            }
        }
        return firstItem;
    }

    public void setFirstItem(int firstItem) {
        this.firstItem = firstItem;
    }

    public int getLastItem() {
        getFirstItem();
        return firstItem + batchSize > itemCount ? itemCount : firstItem + batchSize;
    }
}
```

lanjutan...

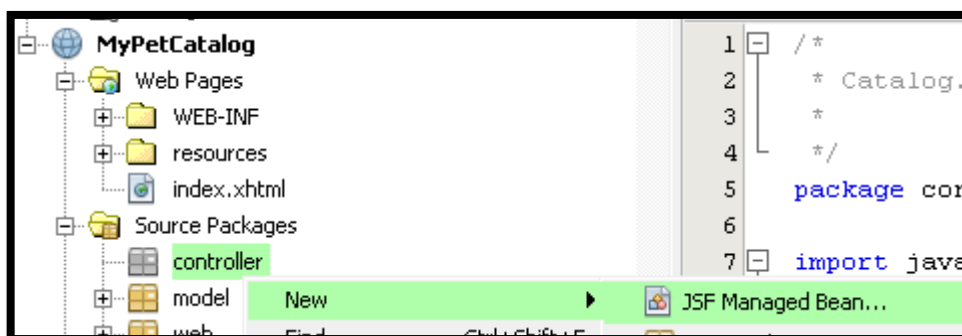
```
public void nextPage() {
    getFirstItem();
    if (firstItem + batchSize < itemCount) {
        firstItem += batchSize;
    }
}

public void previousPage() {
    getFirstItem();
    firstItem -= batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
}

public boolean getIsNextItems() {
    if (getLastItem() < getItemCount()) {
        return true;
    } else {
        return false;
    }
}

public boolean getIsPrevItems() {
    if (firstItem >= batchSize) {
        return true;
    } else {
        return false;
    }
}
}
```

44. Selanjutnya, buatlah sebuah **Managed Bean** pada package controller, sbb:



Name and Location	
Class Name:	Catalog
Project:	MyPetCatalog
Location:	Source Packages
Package:	controller
Created File:	Jliah Genap 2009-2010\01 PWL\Week 08\MyPet
<input type="checkbox"/> Add data to configuration file	
Configuration File:	WEB-INF/faces-config.xml
Name:	Catalog
Scope:	session

Berikut ini adalah content dari ManagedBean "Catalog.java"

```
@ManagedBean
@SessionScoped
public class Catalog implements Serializable {

    @EJB
    private ItemFacade itemFacade;
    private Item item = null;
    private List<Item> items = null;
    private PagingInfo pagingInfo = null;

    /**
     * Creates a new instance of Catalog
     */
    public Catalog() {
        pagingInfo = new PagingInfo();
    }

    public PagingInfo getPagingInfo() {
        if (pagingInfo.getItemCount() == -1) {
            pagingInfo.setItemCount(getItemCount());
        }
        return pagingInfo;
    }

    public List<Item> getNextItems(int maxResults, int firstResult) {
        return itemFacade.findRange(maxResults, firstResult);
    }

    public List<Item> getItems() {
        if (items == null) {
            getPagingInfo();
            items = getNextItems(pagingInfo.getBatchSize(), pagingInfo.getFirstItem());
        }
        return items;
    }

    public Item getItem() {
        return item;
    }

    public String next() {
        reset(false);
        getPagingInfo().nextPage();
        return "index";
    }

    public String prev() {
        reset(false);
        getPagingInfo().previousPage();
        return "index";
    }
}
```

```
public String showDetail(Item item) {
    this.item = item;
    return "detail";
}

public int getItemCount() {
    return itemFacade.getItemCount();
}

private void reset(boolean resetFirstItem) {
    item = null;
    items = null;
    pagingInfo.setItemCount(-1);
    if (resetFirstItem) {
        pagingInfo.setFirstItem(0);
    }
}
}
```

45. Import class yang dibutuhkan (Jangan salah pilih):

```
5   package controller;
6
7   import java.io.Serializable;
8   import java.util.List;
9   import javax.ejb.EJB;
10  import javax.faces.bean.ManagedBean;
11  import javax.faces.bean.SessionScoped;
12  import model.Item;
```

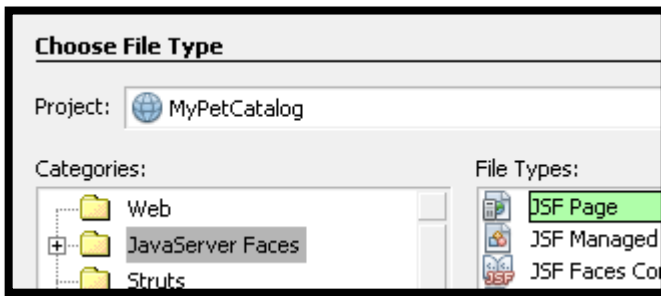
F. Memodifikasi tampilan halaman JSF

46. Selanjutnya kita akan memodifikasi file "index.xhtml" sbb:

- Tambahkan pendefinisian tag "f" pada <html> sbb: `xmlns:f="http://java.sun.com/jsf/core"`
- Isilah content `index.xhtml` sbb:

```
5 <h:head>
6     <title>Pet Catalog</title>
7 </h:head>
8 <h:body>
9     <h:panelGrid columns="2">
10         <h2>#{msgs.Title}</h2>
11         <h:graphicImage library="images" name="banner_logo.gif"
12             style="width: 65px; height: 55px;"/>
13     </h:panelGrid>
14     <h:form styleClass="form">
15         <h:outputStylesheet name="css/styles.css"/>
16         <h:outputText value="Item #{catalog.pagingInfo.firstItem + 1} ..
17             #{catalog.pagingInfo.lastItem} of
18             #{catalog.pagingInfo.itemCount}    "/>
19         <h:commandButton action="#{catalog.prev}" value="#{msgs.Previous}"
20             rendered="#{catalog.pagingInfo.isPrevItems}" />
21         <h:commandButton action="#{catalog.next}" value="#{msgs.Next}"
22             rendered="#{catalog.pagingInfo.isNextItems}" />
23         <h:dataTable var="row" value="#{catalog.items}" border="0"
24             cellpadding="2" cellspacing="0"
25             rowClasses="odd_row,even_row" rules="all"
26             style="border:solid 1px" >
27             <h:column>
28                 <f:facet name="header">
29                     <h:outputText value="#{msgs.Name}" />
30                 </f:facet>
31                 <h:commandLink action="#{catalog.showDetail(row)}"
32                     value="#{row.name}" />
33             </h:column>
34             <h:column>
35                 <f:facet name="header">
36                     <h:outputText value="#{msgs.Photo}" />
37                 </f:facet>
38                 <h:graphicImage library="images"
39                     name="#{row.imagethumburl}" />
40             </h:column>
41             <h:column>
42                 <f:facet name="header">
43                     <h:outputText value="#{msgs.Price}" />
44                 </f:facet>
45                 <h:outputText value="#{row.price}" />
46             </h:column>
47         </h:dataTable>
48     </h:form>
49 </h:body>
50 </html>
```

47. Buatlah sebuah halaman xhtml baru, sbb:



Pastikan anda membuatnya dalam format "Facelets" (bukan JSP):



48. Isilah content dari "detail.xhtml" sbb:

```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://java.sun.com/jsf/html">
5 <h:head>
6     <title>Show Detail</title>
7
8 </h:head>
9 <h:body>
10 <h:panelGrid columns="2">
11     <h2>#{msgs.Title}</h2>
12     <h:graphicImage library="images" name="banner_logo.gif"
13         style="width: 65px; height: 55px;"/>
14 </h:panelGrid>
15 <h:form>
16     <h:outputStylesheet name="css/styles.css"/>
17     <h:panelGrid columns="2" border="0" >
18         <h:outputText value="#{msgs.Name}:"/>
19         <h:outputText value="#{catalog.item.name}" title="Name" />
20         <h:outputText value="#{msgs.Description}:"/>
21         <h:outputText value="#{catalog.item.description}"
22             title="Description" />
23         <h:outputText value="#{msgs.Photo}:"/>
24         <h:graphicImage library="images"
25             name="#{catalog.item.imageurl}"
26             title="Imageurl" />
27         <h:outputText value="#{msgs.Price}:"/>
28         <h:outputText value="#{catalog.item.price}" title="Price" />
29
30     </h:panelGrid>
31     <h:commandButton id="back" value="Back" action="index"/>
32 </h:form>
33 </h:body>
34 </html>
```

49. Run Project!

index.xhtml

Katalog Binatang Kesayangan

Item 1 .. 5 of 29

Nama	Foto	Harga
Friendly Cat		307.10
Fluffy Cat		307.00
Sneaky Cat		307.20
Lazy Cat		307.30
Old Cat		307.00

Katalog Binatang Kesayangan



Nama: Princess cat

Keterangan: Just beauty and elegance. She will charm you from the moment she enters the room.

Foto:



Harga: 307.00

[Back](#)

© 2010 – Niko Ibrahim

Selamat Mengerjakan