ADF Faces : View data from session EJB (Part1)

By: Dr. Ahmad Taufik Jamil, Pusat Teknologi Maklumat, HUKM

This tutorial will show you how to pass parameter value originated from a session EJB and view them in a JSF page. The tutorial will show how to get data from session EJB using JavaBean. The tutorials will show how to develop User Interface using ADF Faces.

Part 1

Using JavaBean (backing bean)

- 1. Creating new workspace and project folder.
 - a. Create a new workspace. Right click at Applications, click New Application... At *Create Application* dialog box, Enter **EJB-JB-ADF** for *Application Name:*, and **No Template [All Technologies]** for *Application Template:*, then click *OK*.

> Create Application	D
Enter the name and location for the new application and specify the application template to use.	4
Application Name:	
EJB-JB-ADF	
Directory Name:	
G:\jdev10.1.3.2\jdev\mywork\EJB-JB-ADF	Browse
Application Template:	
No Template [All Technologies]	Manage Templates
Template Description:	
This application template consists of a single project configured to JDeveloper technologies.	o allow easy access to all
	Cancel

b. Create a project folder. Enter model for *Project Name:*, then click OK.

oiect Name:	
nodel	
rectory Name:	



- 2. Creating a new session EJB
 - a. Right click at project folder *model* and click New...



b. At New Gallery dialog box, inside *Categories:*, under *Business Tier*, click *EJB*, at *Items* : click *Session EJB(EJB1.1/2.x/3.0)*, then click *OK*.



c. At *create session Bean* wizard, click *Next*, at step 1 of 4, select *Enterprise JavaBean 3.0 (J2EE 5.0)*, then click *Next*.

Create Session Bean	Step 1 of 4: Select EJB Version EJB Version An EJB module does not yet exist for this project. Please select the EJB version you wish to use: Enterprise JavaBeans 1.1 Enterprise JavaBeans 2.0 (JZEE 1.3) Enterprise JavaBeans 2.1 (JZEE 1.4) Enterprise JavaBeans 3.0 (JZEE 5.0) Code Generation Apply changes immediately to EJB interface and descriptor files Capture EJB information using %Doclet tags
	< Back Next > Einish Cancel

d. At step 2 of 4, enter Patient for EJB Name:, then click Next.

🔆 Create Session Bean	- Step 2 of 4: EJB N Select an EJB name and EJB Name: Datient	lame and Option	15 X
	Session EJB 3.0 Optio	ns	
	Session Type:	💿 Stateless	◯ Statef <u>u</u> l
	Transaction Type:	⊙ <u>C</u> ontainer	O Bean
	Implement javax.	ejb.SessionSynchron	nization Interface:
5	Generate Session	Facade Methods	More Options
<u> </u>	Entity implementation	; 💿 EJB 3.0 Entitie	es O IopLink POJOs
	Persistence Unit:		~
		< <u>B</u>	ack Next > Einish Cancel

}

- e. Accept all default value at step 3 of 4 and 4 of 4, and click Finish.
- f. Now, source editor for *PatientBean.java* is seen. Add source code below, below constructor *PatientBean() {* } after the curly bracket (}).

```
public ArrayList viewPatient() {
   ArrayList patientList= new ArrayList(10);
    String[] patient1;
    patient1 = new String[]{"Ahmad Zam zam","0122345678","700628045678","Selangor"};
    String[] patient2;
    patient2 = new String[] {"Arin Sujak","0177745678","890628045671","Perlis"};
    String[] patient3;
    patient3 = new String[]{"Baharin Ujang","0195675678","880628045672","Pahang"};
    String[] patient4;
    patient4 = new String[]{"Zulhairi Hasan","0136345671","560628045674","Trengganu"};
    String[] patient5;
    patient5 = new String[] {"Hasan Kamal","0159345673","450628045633","Johor"};
    String[] patient6={"Mohamad Kamal","0188345632","670628045687","Sarawak"};
    patientList.add(patient1);
   patientList.add(patient2);
    patientList.add(patient3);
    patientList.add(patient4);
    patientList.add(patient5);
   patientList.add(patient6);
    return patientList;
```

🔗 Pati	ent8e	Bean, java 🛛 🔊 PatientBean, java	
7	⊎St	Stateless(name="Patient")	
†i ⊟	pub	ublic class PatientBean implements Patient, PatientLocal {	
9 🖻	l.	<pre>public PatientBean() {</pre>	
10		3	
11	<u>k</u>		
12 🖻);	<pre>public ArrayList viewPatient() {</pre>	
13		<pre>ArrayList patientList= new ArrayList(10);</pre>	
14		String[] patientl;	
15		<pre>patient1 = new String[]{"Ahmad Zam zam","0122345678","700628045678","Selangor</pre>	;"};
16		<pre>String[] patient2;</pre>	
17		<pre>patient2 = new String[] {"Arin Sujak","0177745678","890628045671","Perlis"};</pre>	
18	8	String[] patient3;	
19		<pre>patient3 = new String[]{"Baharin Ujang","0195675678","880628045672","Pahang"}</pre>	;
20		String[] patient4;	
21		<pre>patient4 = new String[]{"Zulhairi Hasan","0136345671","560628045674","Trengga</pre>	mu"};
22		String[] patient5;	
23		<pre>patient5 = new String[] {"Hasan Kamal","0159345673","450628045633","Johor"};</pre>	
24		String[] patient6={"Mohamad Kamal","0188345632","670628045687","Sarawak"};	
25	8		
26	1	<pre>patientList.add(patientl);</pre>	
27		<pre>patientList.add(patient2);</pre>	
28		<pre>patientList.add(patient3);</pre>	
29		<pre>patientList.add(patient4);</pre>	
30		<pre>patientList.add(patient5);</pre>	
31		<pre>patientList.add(patient6);</pre>	
32	8		
33		return patientList;	
34)	
35	1		
36	ji –		

g. Expand project folder *model*, until you see *PatientBean.java* and click to mark it. Under structure below it, you will see *viewPatient()* under folder *method*.



h. Right click at viewPatient(), and click Properties...



i. At *Bean Method details* dialog box, check both *Expose through Local Interface* and *Expose through Remote Interface*, and click *OK*.

	viewPatient	
Re <u>t</u> urn Type:	java.util.ArrayList	Browse
Parameters:		
Thro <u>w</u> s:		
Method Spec:	ArrayList viewPatient() throws	RemoteException
🛃 Expose thr	rough Local interface	
Expose the	rough Remote interface	
Expose through Remote Interface		
— ·	Expose through Web Service Endpoint interface	
Expose thr	rough <u>W</u> eb Service Endpoint int	terface
Expose thr	rough <u>W</u> eb Service Endpoint int	Service Properties
Local Propert	rough <u>W</u> eb Service Endpoint int ies Remote Properties Web	terface Service Properties
Container Tra	rough Web Service Endpoint int ies Remote Properties Web insaction Attribute:	erface Service Properties
Expose thr	rough <u>W</u> eb Service Endpoint int ies Remote Properties Web	terface Service Properties

- j. Save all and compile (Make) file PatientBean.java
- 3. Creating new project folder for view layer
 - a. Right click at *EJB-JB-ADF* workspace, click *New Project*... At *New Gallery* dialog box, at column *Categories*:, under *General*, select *Projects*, then click *Empty Project* at column *Items*: then click *OK*.

Phications	Connections	
EJB3ADF EJB3ADF EJB3ADF2 EJBADF		
EJB-JB-ADE	New Project	
6-10 A	New Add to EJB-JB-ADF.jws	Ctrl+N
	🗙 Delete	Delete
⊕ िि HelloW ⊕ िि HR_EJB_A ⊕ िि HW	<u>M</u> ake Re <u>b</u> uild	Ctrl+Shift+F9 Alt+Shift+F9
⊕ - CE HW1 ⊕ - CE JSFEJB30	Re <u>f</u> ormat Organi <u>z</u> e Imports	Ctrl+Alt+L Ctrl+Alt+O
⊕	Import Module Compare With	٢
EJB-JB-ADF. jw	Replace <u>W</u> ith	

ategories:	Items:
General Applications Connections Connections External Applications Forects Business Tier ADF Business Components Web Services Database Tier Integration Tier BPEL BFEL BFEL BFEL BFEL BFEL BFEL BFEL BF	Empty Protect BPEL Process Project Business Components Project ESB Project Java Application Project Project from Existing Source Project from WAR File TopLink Project Web Project
	Description:
	Opens the Create Project dialog, in which you define a directory and project (.jpr) filename for a new empty project. Use projects to organize your files. To enable this option, you must select an application, a project, or a file within a project in the Application Navigator.

b. At Create Project dialog box, enter view for Project Name: and click OK. New view project folder is created.

oject Name:		
iew.		
ectory Name:		
	Assumed AF 1P 1P ADEView	Province



- 4. Creating new Managed Bean
 - a. Right click at view project and click New...



b. At *New Gallery* dialog box, at *Categories:*, expand *General*, find and click *JavaBeans*. Inside *Items*, click *Bean* and then click *OK*.

Categories:	Items:
∃General	😕 Bean
Applications Connections Deployment Descriptors Deployment Profiles Diagrams External Applications InvaBeans Projects Simple Files Web Tier JSF ZC	(1) BeanInfo
Servlets	Description:
	Opens the Create Bean dialog, in which you assign a name and a package to your bean and define the class it extends. To enable this option, you must select a project or a file within a project in the Application Navigator.

c. At *Create Bean* dialog box, enter **PatientJB** for *Name:*, **view** for *Package:*, and choose **java.lang.Object** for Extends:, then click *OK*. Source editor for *PatientJB.java* is now opened.

n	
ne details of your new	S
PatientJB	
view	Browse
java.lang.Object 🔽	Browse
	PatientJB View

d. Add the following lines inside the *PatientJB* class, immediately below *PatienJBt* constructor (public PatientJB() { }).

```
private String name;
private String telno;
private String icno;
private String state;
public PatientJB(String name, String telno, String icno, String state) {
     this.name=name;
     this.telno=telno;
     this.icno=icno;
     this.state=state;
 }
public String getName() {
    return name;
 }
public String getTelno() {
     return telno;
 }
public String getIcno() {
     return icno;
```

```
public String getState(){
    return state;
}
```

}

🔡 fa	же	s-config.xml 🛃 borang.jsp 🔠 Borang.java 🛃 PatientJB.java
1		package view;
2		
3	8	public class PatientJB {
4	9	<pre>public PatientJB() {</pre>
5		}
6		
7		private String name;
8		private String telno;
9		private String icno;
10		private String state;
11		
12	8	<pre>public PatientJB(String name, String telno, String icno, String state) {</pre>
13		
14		this.name=name;
15		this.telno=telno;
16		this.icno=icno;
17		this.state=state;
18		}
19		
20	E	public String getName(){
21		return name;
22		2
20		mublic String getTelno()(
25	-	partic scring generation(),
26		l
27		
28	a	public String getIcno(){
29		return icno:
30		
31		
32	9	<pre>public String getState() {</pre>
33		return state;
34		}
35		
Sour	ce	Design GWT Design History

- e. Save all and compile(Make).
- 5. Creating page navigation using face-config.xml.
 - a. Right click at *view* project and click *Project Properties*.... In *Project Properties* dialog box, at left column; click *Technology Scope*. Find and click *JSF* at *Available Technologies* and move to right column (*Selected Technologies*)-*Java, JSP and Servlets* will be moved as well.

Project Content	Technology Scope		
ADFm Settings ADF View Settings Ant Business Components	Use Custom Settings	Customize Settings.	•
 Compiler Dependencies EJB Module J2EE Application ⇒ Javadoc JSP Tag Libraries Libraries Offline Database Run/Debug Technology Scope 	Available Technologies: ADF Business Components ADF Swing Ant Content Repository Database EJB Google Web Toolkit HTML Integration JavaBeans JSP for Business Components Mobile Oracle JSP Tag Libraries Iechnology Description:	Selected Technologies: Java JSF JSP and Servlets	

b. At the same dialog box, click at *Dependencies* (at left column) and check *model.jpr* (at right column), then click *OK*.

🕹 Project Properties - G:\jds	ev10.1.3.2\jdev\mywork\EJB-JB-ADF\view\view.jpr		×
Project Content ADFm Settings ADF View Settings ADF View Settings Business Components Compiler Dependencies	Dependencies Use Custom Settings Use Project Settings Project Dependencies: Image: Comparison of the set of	Customize Settings	
E3B Module → J2EE Application (+) Javadoc → J5P Tag Libraries → Libraries → Offline Database → Run/Debug → Technology Scope		Qrdering	
Help		OK Cancel)

c. Right click at *view* project, and click *Open JSF navigation*.



d. File *faces-config.xml* is opened in Diagram view.



d. Create new page navigation. Click, drag & drop JSF Page from Component Palette, onto the diagram view of faces-config.xml, and then drop another JSF page beside it. Click JSF Navigation Case from Component Palette, then click at /untitled1.jsp and then click at /untitled2.jsp. An arrow is created and pointing to /untitled2.jsp. Change – success to – submit.



e. Create another JSF Navigation from */untitled2.jsp* to */untitled1.jsp*. Rename – success to – back.



- f. Click Save All
- 6. Creating JSF form and backing beans.
 - a. Go back to *faces-config.xml* Diagram view.



b. Double click icon */untitled1.jsp*. At *Create JSF JSP* wizard click *Next*. At Step 1 of 4, enter **borang.jsp** for *File name:*. Click *Next*. Choose *Automatically Expose UI Components in a New managed Beans*. Click *Finish*.

	File Name:
	borang.jsp
	Directory Name:
1	G:\jdev10.1.3.2\jdev\mywork\ADFJB\view\public_html Browse
	Туре:
	JSP Page (*, jsp)
	A JSP page is a text-based document that describes how to process a request to create a response.
	◯ JSP Document (*.jspx)
	A JSP document is an xml representation of a JSP page.
	Mobile:
0.000	Add Mobile Support

	If you need to progra automatically expose representing UI comp corresponding proper in code, you can exp Do Not Automatic	mmatically modify this page's UI of them in a managed bean. This cru- onents added to the page, and b ty. If you choose not to automat sse individual components directly ally Expose UI Components in a M	components you can choose to eates bean properties inds each UI component to its ically expose all UI component: from the page later. lanaged Bean
	Automatically Exp	ose UI Components in a New Mar	laged Bean
	Nam <u>e</u> :	backing_borang	
	Class:	Borang	
20	<u>P</u> ackage:	view.backing	Browse
	O Automatically Exp	ose UI Components in an Existing	Managed Bean
	<u>M</u> anaged Bean:		*

c. At the *borang.jsp* design area, click, drag and drop *PanelPage* from *Components Palette*, to *borang.jsp* design area.



d. Using *Property Inspector*, change Title to View Patient's Details. Now the title of the page has changed.

Patient.java 🔣 faces-config.xm	borang.jsp		Components
None Default	None ☑ 1		ADF Faces Core
branding brandingApp	menuGlobal menu 1		PanelGroup PanelHeader PanelHorizontal
View Patient's Details	appPriva	CV E	 PanelLabelAndMessage PanelList PanelPage
			PanelPageHeader PanelPage - View Patient's Details
			General Title AttributeCha ChromeType compact

e. Click, drag & drop *CommandButton* from *Component Palatte* to the *panelPage*, and change the *Text* value in *Property Inspector* from *CommandButton 1* to View Patient and press Enter.

Patient.java Races-config.xml	borang.jsp		Component Palette	🛃 Data Controls	
🚱 None 💽 Default	💟 None 💟 🌇 🗞 🥔 B 👖 🖳	E E E E E -	ADF Faces Core		~
			De Pointer		
1			XII Attribute		
			ChooseColor		
branding brandingApp		menuGlobal	ChooseDate		
		menu1			
View Patient's Details					
View Patient					
appCopyright		appPrivacy		·	
appAbout			CommandButton - Vi		
			1 2 / 60 (9 🗄 🗊 🚚	
			⊡General		
			Text	View Patient	
			Action		
			ActionListener		_
			LaunchListener	Colored States	_
			PartialSubmit	raise	_
			According		<u> </u>
) udy1> <h:form#form1> <af:panelpage#panelpage1> <</af:panelpage#panelpage1></h:form#form1>	af:commandbutton#commandbutton1>	Go to Page Definition		
Design Source History][>			-

- 7. Editing backing bean (*Borang.java*).
 - a. Double click button *View Patient*, *Bind Action Property* dialog box will appear, accept all the default value and click *OK*.

ind Action Proj	perty	
<u>M</u> anaged Bean:	backing_borang	<u>N</u> ew
Method:	commandButton1_action	
Help	ОК	Cancel

b. You will see backing bean *Borang.java* is opened in the editor. Add the following lines to *Borang.java*, after the variable declaration (after this line private CoreCommandButton commandButton1;).

```
//Declare new variable for patient
   String namex=null;
   String telnox=null;
   String icnox=null;
   String statex=null;
   ArrayList <PatientJB> patientjlist = new ArrayList <PatientJB>();
//Declare new constructor to connect to EJB using initialcontext and get values for patient
   public Borang() {
   try {
        final Context context = getInitialContext();
        Patient patient = (Patient)context.lookup("Patient");
       ArrayList patientList= new ArrayList(10);
       patientList = patient.viewPatient();
        for(int i=0;i<patientList.size();i++) {</pre>
           String[] s = (String[])patientList.get(i);
            namex=s[0];
            telnox=s[1];
            icnox=s[2];
            statex=s[3];
```

```
patientjlist.add(new PatientJB(namex,telnox,icnox,statex));
         }
    } catch (Exception ex) {
         ex.printStackTrace();
     }
//Method for initial context
     private static Context getInitialContext() throws NamingException {
          // Get InitialContext for Embedded OC4J
          //\ {\rm The}\ {\rm embedded}\ {\rm server}\ {\rm must}\ {\rm be}\ {\rm running}\ {\rm for}\ {\rm lookups}\ {\rm to}\ {\rm succeed}.
          return new InitialContext();
     }
   //method to return value to JSF page
     public ArrayList <PatientJB> getPatientjlist() {
          return patientjlist;
```

```
}
```



c. Press ALT+ENTER few times if necessary to insert line import below, or you can manually add the line below to file *Borang.java*.

```
import view.PatientJB;
import java.util.ArrayList;
import javax.faces.component.html.HtmlForm;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import model.Patient;
```

- d. Find method public String commandButton1_action(), in file *Borang.java*, and change the line return null; to return "submit";
- e. Save All and compile (Make).
- 8. Creating JSF page to view data.
 - a. Go back to diagram view of *faces-config.xml*. Double click icon /untitled2.jsp
 - b. On *Create JSF JSP wizard* click *Next*, at Step 1 of 4, change file name to **papar.jsp** and click *Next*. Choose *Do Not Automatically Expose UI*.....then click *Finish*.

🕹 Create JSF JSP - S	Step 1 of 4: JSP File	
	Enter the name, directory, and choose a type for the JSP. File Name:	
	papar .jsp	
	Directory Name:	
	G:\jdev10.1.3.2\jdev\mywork\EJB-JB-ADF\view\public_html Brow	NSe
	Type: JSP Page (*, jsp) A JSP page is a text-based document that describes how to process a requ create a response.	iest to
000	◯ JSP D <u>o</u> cument (*.jsp×)	
	A JSP document is an xml representation of a JSP page. Mohile:	
April 1	Add Mobile Support	
	Device Category PDA	
	< <u>Back</u> <u>M</u> ext > <u>Finish</u>	Cancel

	If you need to progra automatically expose representing UI comp corresponding proper in code, you can exp	mmatically modify this page's UI components you them in a managed bean. This creates bean prop onents added to the page, and binds each UI cor ty. If you choose not to automatically expose all ose individual components directly from the page I	can choose to ierties inponent to its UI components later.
11	Do Not Automatic	ally Expose UI Components in a Managed Bean	
	Automatically Exp	ose UI Components in a New Managed Bean	
	Nam <u>e</u> :	backing_paparl	
	Class:	Paparl	
	<u>P</u> ackage:	view.backing	Browse
1		ose UI Components in an Existing Managed Bean	
	Managed Bean:	*	

c. At Design view for file *papar.jsp*, click, drag & drop *PanelPage* from *Components Palette* tab to design area, and change the title to ADF Faces: View value from other page.

pranding brandingApp	menuGlobal
	menu1
ADF Faces: View value from other page.	
ippCopyright ippAbout	appPrivacy
h:form	

d. Find component Table from Component Palette, click, and drag & drop Table to papar.jsp (inside panelPage).



e. Click Next on the welcome page of Create Table wizard.

🔶 Insert Table - Weld	some 🔀
	Welcome to the Create ADF Faces Table Wizard This wizard helps you quickly create an ADF Faces table including setting column information as well as formatting. Click Next to continue.
	Skip this Page Next Time
	< <u>Back</u> <u>N</u> ext > Einish Cancel

f. At Step 1 of 4, choose **Bind** Now, and click button Bind...

Navigate Run Debug Refactor Versioning Tools Wi	🐣 Expression Builder	\mathbf{X}
Navigate Run Debug Refactor Versigning Tools Wir Insert Table - Step 1 of 4: Binding Choose to bind the table not Bing Now Value: Value: Value: Value: Data Type Hint: Bind Later Column Count:	Expression Builder Filter By: List, ResultSet, Result, Arrays, CollectionModels, Data Variables: JSF Managed Beans Booking_borang Datentilist	<pre>Models Expression: #(backing_borang.patientjlist) Logical Operators: && > < >= == = = ! Mathematical Operators: + - * / %</pre>
Home	Help	OK Cancel

- g. At *Expression Builder*, under *backing_borang* and *JSF managed Beans*, click *patientilist*, and click button > to move it to the right column. Then click *OK*.
- h. Enter tableData for field Var: , and click button Browse...

👌 Insert Table - Step	1 of 4: Binding		🐣 Class Browser	X
	Choose to bind the t	able now or specify a number of columns	Select Class To search for a class, enter the simple class name. Enter the package prefix to search by package. Use a question mark (?) to match any single character, or an asterisk (*) to match any number of characters. Click the Hierarchy tab to browse t	3
	V <u>a</u> lue:	#{backing_borang.patientjlist}		
	<u>V</u> ar:	tableData	Search Hierarchy	
	Da <u>t</u> a Type Hint:		avax	
	O Bind Later Column Count:	4		
		< <u>B</u> ack <u>N</u> ext >		incel

i. At Class Browser, at *Hierarchy* tab, expand *view* and click *PatientJB* and click *OK*.

Rind Now	cable now or specify a number of columns an	a bina che cable lacer.
Value:	#{backing_borang.patientjlist}	Bind
<u>V</u> ar:	tableData	
Da <u>t</u> a Type Hint:	view.PatientJB	Browse
O Bind Later Column Count:	4	

j. Click Next at Step 1 of 4, then click Next at Step 2 of 4.

🏷 Insert Table - Step	S of 4: Column De Enter the following in Columns:	itaills nformation fo	r each of the columns	that will appear in th	nis table.
	Telno	Top	He <u>a</u> der Text:	Telno	Bin <u>d</u>
	Icno State	Up	Component <u>V</u> alue:	ableData.telno}	Bind
	Name	Down	UIComponent:	af:outputText	
		Bottom	Format Type:	text	
			Is Sortable		
		Group	Sort Property:		
V	N <u>e</u> w Delete	Undroup			
Help			< Back	ext > Einish	Cancel

k. At Step 3 of 4, click *Name* inside *Columns:*, and click button *Top* to move name to the top column. Change *Header Text:* to **Name, Tel No, IC No** and **State** for *Name, Telno, Icno* and *State* respectively, then click *Next.*

🔶 Insert Table - Step 🗄	of 4: Column Det	ails			
Insert Table - Step 3	s of 4s Column Det	ails formation fo Up Down Bottom Group Ungroup	r each of the columns Hegder Text: Component <u>V</u> alue: UIComponent: Format Type: Is Sortable Sort Property:	that will appear in th Tel No ableData.telno} af:outputText text	iis table. Bind Bind
			< <u>B</u> ack	ext >	Cancel

- l. At Step 4 of 4, click *Finish*.
- m. Click Save All and compile(Make).
- n. Run file *borang.jsp*

00-	E ht	ttp://192.168	3.218.10	0:8988/	EJB-JB-A	DF-view-context-	root/faces/borang
File Edit	View	Favorites	Tools	Help	; Links	CHETS-Prod	CHETS-Dev
🚖 🎄 👔	Home						
View Pa	tient'	s Details	3				
View Patie	ent)						

o. Click button View Patient. You will see the result below.

l

🥑 👻 🔻 🖉 h	ttp://192.168.2	218.100:8988/EJB	JB-ADF-view-	:ontext-root/faces/borang.js	p;jsessio
File Edit View	Favorites	Tools H <mark>elp </mark>	inks 🙋 CHE	TS-Prod 🙋 CHETS-Dev	🛒 ewarg
👌 🏟 🏾 🏉 Home	Ú.				
ADF Faces: \	/iew value	from othe	r page		
Name	Tel No	IC No	State		
Ahmad Zam zam	0122345678	700628045678	Selangor		
Arin Sujak	0177745678	890628045671	Perlis		
Baharin Ujang	0195675678	880628045672	Pahang		
	0136345671	560628045674	Trengganu		
Zulhairi Hasan	0450045070	450628045633	Johor		
Zulhairi Hasan Hasan Kamal	01593456/3				