

MODUL IV

DHTML DAN JAVASCRIPT

A. TUJUAN

- Memahami konsep dasar *active web page*.
- Mampu menghasilkan halaman web yang interaktif.
- Mampu memanfaatkan bahasa *scripting* untuk operasi-operasi spesifik dan validasi data.

B. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

C. DASAR TEORI

1. Active Web Page

Sebagaimana diketahui, HTML merupakan model dokumen yang statis—begitu ditampilkan tidak akan berubah sampai ada aktivitas navigasi. Adapun untuk menjadikan HTML sebagai halaman dinamis atau aktif, kita bisa memanfaatkan bahasa scripting.

2. DHTML dan JavaScript

Scripting merupakan jenis lain dari pemrograman, yang umumnya lebih mudah dipahami. Script dapat disisipkan ke dalam dokumen HTML dengan menggunakan tag `<script>`. Tidak seperti style sheet, script bisa terlihat di dalam `<head>` ataupun `<body>`.

DHTML (Dynamic HTML) merupakan suatu teknik untuk menciptakan halaman web yang interaktif dengan cara mengombinasikan elemen-elemen seperti HTML, style sheet, dan scripting. Jadi, DHTML bukanlah suatu jenis bahasa pemrograman melainkan hanya sekadar istilah saja. Pada bagian ini, pembuatan DHTML akan dilakukan dengan memanfaatkan JavaScript.

D. LATIHAN

1. Elemen Dasar JavaScript

Sebagaimana bahasa pemrograman umumnya, JavaScript terbentuk atas elemen-elemen fundamental. Di sini kita tidak akan mengulas elemen-elemen tersebut, melainkan langsung membahas beberapa komponen dasar yang kerap digunakan saat bekerja dengan JavaScript.

- Kotak Dialog

JavaScript memungkinkan kita untuk menampilkan kotak dialog melalui fungsi `alert()`. Kotak dialog ini dapat dimanfaatkan untuk menyampaikan informasi atau konfirmasi.

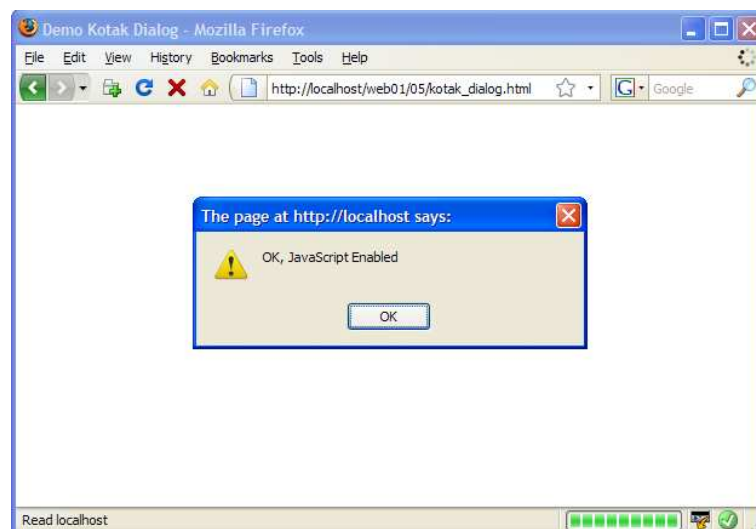
```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Kotak Dialog</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
alert('OK, JavaScript Enabled');
//-->
</script>

Refresh (F5) untuk melihat pesan JavaScript
</body>
</html>
```



Gambar 1. Kotak dialog konfirmasi JavaScript

▪ Dukungan Scripting

Melalui tag `<noscript>`, dengan mudah kita bisa mengidentifikasi apakah browser mendukung scripting atau tidak. Pada dasarnya, tag ini tidak dimaksudkan untuk menghentikan interpretasi terhadap kode program. Dengan kata lain, browser akan tetap menerjemahkan baris kode kecuali tag `<script>`.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Dukungan Scripting</title>
</head>

<body>

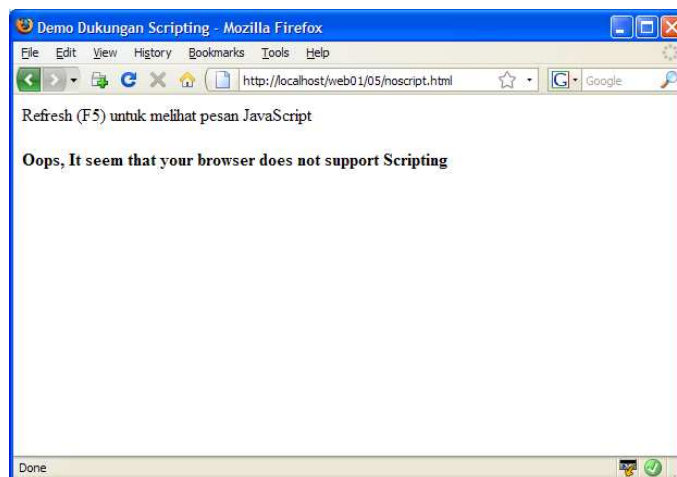
<script language="JavaScript" type="text/javascript">
<!--
alert('OK, JavaScript Enabled');
//-->
</script>

Refresh (F5) untuk melihat pesan JavaScript

<noscript>
  <h4>Oops, It seem that your browser does not support Scripting</h4>
</noscript>

</body>
</html>
```

Jika browser tidak mendukung scripting, maka yang dijalankan adalah tag `<noscript>`.



Gambar 2. Browser tidak support scripting

▪ Merujuk Elemen

Salah satu kemampuan utama JavaScript adalah mengontrol objek-objek di halaman web. Sehubungan dengan hal ini, ada beragam pendekatan yang bisa digunakan untuk mengacu atau mengakses elemen-elemen objek.

Sebagai contoh, misalkan kita memiliki elemen form sebagai berikut:

```
<form>
  <input type="text" name="myName" id="myID" />
</form>
```

Masing-masing pendekatan di bawah ini dapat digunakan dan mengacu pada elemen yang sama di atas.

```
// Menggunakan id numerik (urutan elemen)
document.forms[0].elements[0];

// Menggunakan nama elemen
document.forms[0].elements['myName'];

// Menggunakan id elemen
document.forms[0].elements['myID'];

// Menggunakan nama dan id secara langsung
document.forms[0].myName;
document.forms[0].myID;

// Menggunakan method getElementById
document.getElementById('myID');
```

▪ Event

Setiap objek HTML mendefinisikan properti-properti, method-method, dan event-event. Salah satu event yang sangat populer dan frekuensi penggunaannya cukup tinggi adalah onclick. Event ini akan dipicu manakala objek—misalnya button atau teks—diklik.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Event Onclick</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
// Fungsi sederhana untuk memunculkan dialog
function show_dialog() {
  alert('OK, JavaScript Enabled');
}
//-->
</script>

<a href="#" onclick="show_dialog()">Click Me</a>
```

```
<form action="#">
<input type="submit" value="Click Me" onclick="show_dialog()" />
</form>

<!-- Menulis alert langsung -->
<a href="#" onclick="alert('Simple Alert');">Click Me</a>

</body>
</html>
```

▪ Linked Scripting

Seperti halnya CSS, penulisan kode JavaScript juga bisa dilakukan di file terpisah. Langkah ini sebaiknya digunakan ketika melakukan implementasi nyata, dengan tujuan membuat kode yang bersih. Contoh berikut memperlihatkan pemisahan kode JavaScript dan HTML.

File **lat_script.js**:

```
// Fungsi sederhana untuk memunculkan dialog
function show_dialog() {
    alert('OK, JavaScript Enabled');
}
```

File **linked_script.html**:

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Demo Linked Script</title>

    <script type="text/javascript" src="lat_script.js"></script>
</head>

<body>

<a href="#" onclick="show_dialog()">Click Me</a>

</body>
</html>
```

Sekali lagi, penggunaan pendekatan embedded pada praktikum ini hanya dimaksudkan agar lebih memfokuskan scripting. Bagaimanapun, untuk implementasi nyata sebaiknya kode JavaScript diletakkan terpisah.

2. Me-retrieve Nilai

Saat bekerja dengan elemen-elemen masukan—seperti text field, select, dan check box—HTML, kita bisa memanfaatkan JavaScript untuk operasi validasi. Adapun langkah penting terkait dengan elemen-elemen ini adalah pengambilan nilai.

▪ Text Field

Elemen text field menyediakan properti value yang bisa digunakan untuk menetapkan atau mendapatkan nilai elemen.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Retrieve</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--

function getTextValue() {
  var myForm = document.form1;
  // Mendapatkan nilai text field
  alert("Nilai Teks: " + myForm.nama.value);
}
//-->
</script>

<form name=form1 action="#">
Nama <input type="text" name="nama" />
<input type="submit" value="Submit" onclick="getTextValue()" />
</form>

</body>
</html>
```

▪ Select (Combo Box)

Secara umum, properti value tersedia di semua elemen HTML. Dengan demikian, kita juga bisa memanfaatkannya untuk mendapatkan nilai select atau combo box.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Retrieve</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--

function getSelectedValue() {
  var myForm = document.form1;
  // Mendapatkan nilai selected field
  alert("Nilai : " + myForm.sex.value);
}
//-->
</script>
```

```
<form name=form1 action="#">

Jenis Kelamin:
<select name="sex">
  <option value="Pria">Pria</option>
  <option value="Wanita">Wanita</option>
</select>

<input type="button" value="Submit" onclick="getSelectedValue()" />
</form>

</body>
</html>
```

Untuk operasi cepat—tanpa perlu klik button—pada select, kita bisa menambahkan event onchange.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Retrieve</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
function getSelectedValue() {
  var myForm = document.form1;
  // Mendapatkan nilai selected field
  alert("Nilai : " + myForm.sex.value);
}
//-->
</script>

<form name=form1 action="#">

Jenis Kelamin:
<select name="sex" onchange="getSelectedValue();">
  <option value="Pria">Pria</option>
  <option value="Wanita">Wanita</option>
</select>

</form>

</body>
</html>
```

▪ Check Box

Khusus untuk pengambilan nilai check box, terlebih dahulu kita perlu mengidentifikasi item yang dipilih.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
```

```
<title>Demo Retrieve</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
function getCheckedVal() {
    var element;
    var len = document.form1.length;
    var str = "";

    for (i = 0; i < len; i++) {
        element = document.form1[i];
        if (element.type == "checkbox") {
            if (element.checked == true) {
                str = str + element.value + "\n";
            }
        }
    }

    if (str.length == 0) {
        alert("Anda tidak memilih");
    } else {
        alert(str);
    }
}
-->
</script>

<form name=form1 action="#">

<p> Aksesoris: <br />
<input type="checkbox" name="chkDVD" value="DVD-ROM" />DVD-ROM <br />
<input type="checkbox" name="chkMouse" value="Mouse" />Mouse <br />
<input type="checkbox" name="chkkey" value="Keyboard" />Keyboard<br />
<input type="button" value="OK" onclick="getCheckedVal()" />
</form>

</body>
</html>
```

3. Memilih Item Check

Agar pemilihan item check box bisa lebih efisien, kita dapat menggunakan pendekatan check/uncheck all.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Demo Check/Uncheck</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
/**
 * field Kontrol pemicu
 * list Elemen list
```

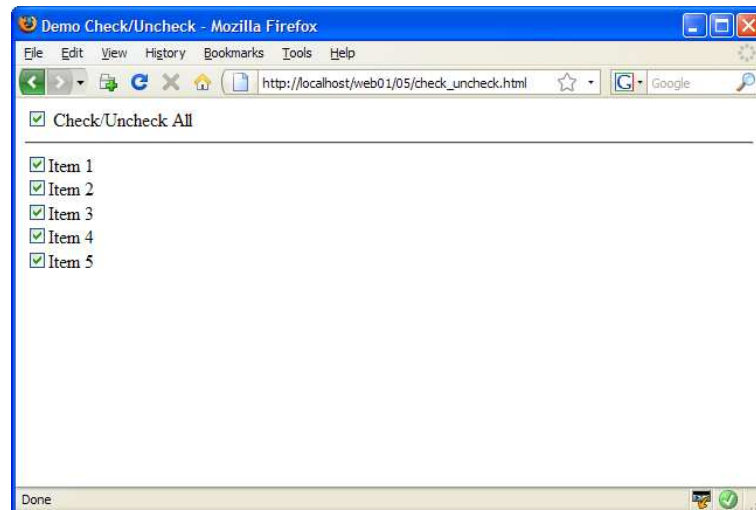


```
*/
function check(field, list) {
  if (field.checked) {
    checkUncheck(list, 1);
  } else {
    checkUncheck(list, 0);
  }
}

// Check/Uncheck field berdasar value
function checkUncheck(field, value) {
  var len = field.length;
  for (i = 0; i < len; i++) {
    field[i].checked = value;
  }
}
//-->
</script>

<form action="#" method="post">
<input type="checkbox" onClick="check(this, list)" />
Check/Uncheck All <hr />
<input type="checkbox" name="list" value="item_1" />Item 1<br />
<input type="checkbox" name="list" value="item_2" />Item 2<br />
<input type="checkbox" name="list" value="item_3" />Item 3<br />
<input type="checkbox" name="list" value="item_4" />Item 4<br />
<input type="checkbox" name="list" value="item_5" />Item 5<br />
</form>

</body>
</html>
```



Gambar 3. Check dan uncheck list

4. Window

- Menciptakan Window

Di dalam window aktif, kita juga bisa menciptakan window baru melalui method `window.open()`.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Open Window</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
var subWindow;

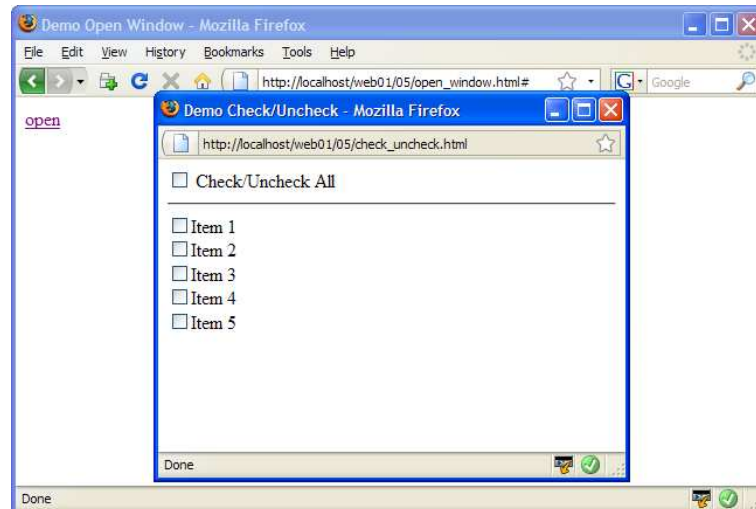
/**
 * url String alamat url
 */
function openWindow(url) {
  if (!subWindow || subWindow.closed) {
    var width  = 400;
    var height = 250;
    // Mendapatkan posisi kiri atas (dari layar)
    var left   = parseInt((screen.availWidth/2) - (width/2));
    var top    = parseInt((screen.availHeight/2) - (height/2));
    // Menetapkan fitur window
    var fitur  = "width=" + width + ", height=" +
      height + ", resizable=0, left=" +
      left + ",top=" + top + ",screenX=" +
      left + ",screenY=" + top;

    // Membuka window
    subWindow = window.open(url, "sub", fitur);
  } else {
    // Sub window sudah dibuka, sehingga tinggal
    // memberinya fokus
    subWindow.focus();
  }
}
<!-->
</script>

<a href="#" onclick="openWindow('check_uncheck.html')">open</a>

</body>

</html>
```



Gambar 4. Membuka window baru

- Komunikasi Window

Apabila diperlukan, window induk—yang menciptakan subwindow—juga dapat berkomunikasi dengan window buatannya.

Sebagai contoh, buat window induk terlebih dahulu, simpan dengan nama `main_window.html`.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Open Window</title>
</head>

<body>

<script language="JavaScript" type="text/javascript">
<!--
var subWindow;

/**
 * url String alamat url
 */
function openWindow(url) {
  if (!subWindow || subWindow.closed) {
    var width  = 400;
    var height = 250;
    // Mendapatkan posisi kiri atas (dari layar)
    var left   = parseInt((screen.availWidth/2) - (width/2));
    var top    = parseInt((screen.availHeight/2) - (height/2));
    // Menetapkan fitur window
    var fitur  = "width=" + width + ", height=" +
      height + ", resizable=0, left=" +
      left + ",top=" + top + ",screenX=" +
      left + ",screenY=" + top;

    // Membuka window
```

```
        subWindow = window.open(url, "sub", fitur);
    } else {
        // Sub window sudah dibuka, sehingga tinggal
        // memberinya fokus
        subWindow.focus();
    }
}
//-->
</script>

<form action="#">
<input type="text" name="txtsex" />
<a href="#" onclick="openWindow('sub_window.html')">Pilih</a>
</form>

</body>
</html>
```

Buat subwindow dengan nama sub_window.html.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Demo Open Window</title>
</head>

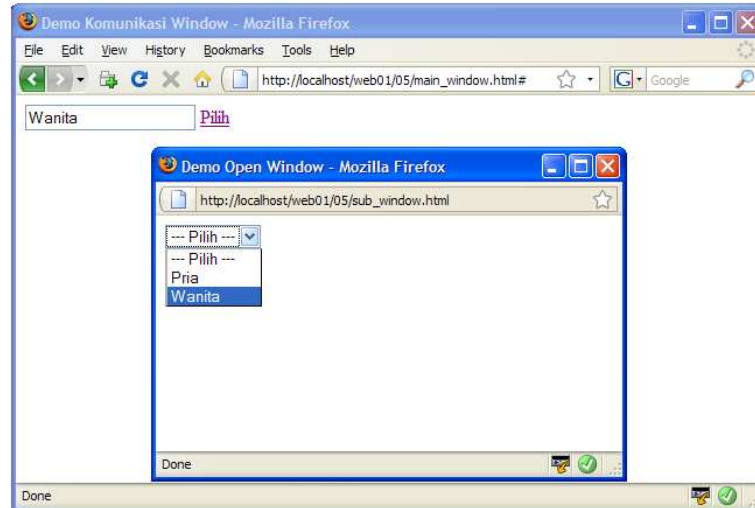
<body>

<script language="JavaScript" type="text/javascript">
<!--
// Set nilai elemen opener
function setSelected(fld) {
    // Men-set nilai field txtsex di parent
    opener.document.forms[0].txtsex.value = fld.value;
    // Tutup window ini (subwindow)
    window.close()
}
//-->
</script>

<select name="sex" onchange="setSelected(this);">
    <option>--- Pilih ---</option>
    <option value="Pria">Pria</option>
    <option value="Wanita">Wanita</option>
</select>

</body>
</html>
```

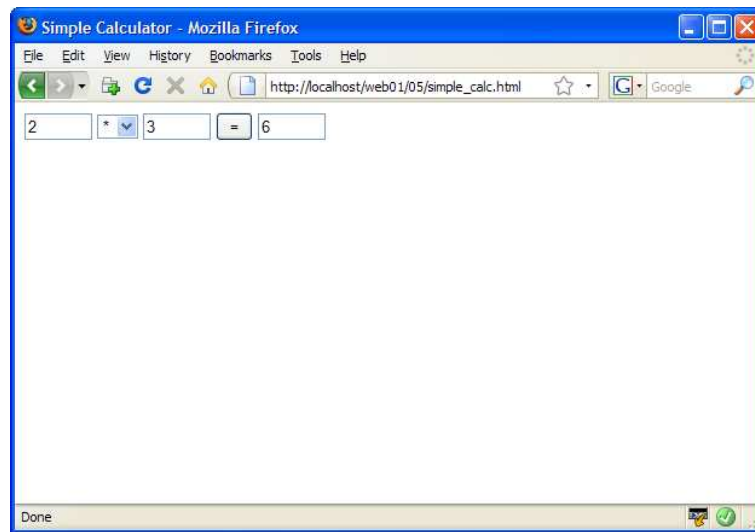
Jalankan halaman main_window, kemudian klik link Pilih dan isikan pilihannya.



Gambar 5. Komunikasi antar-window

E. STUDI KASUS

1. Buat aplikasi kalkulator sederhana :



Gambar 6. Kalkulator sederhana

F. TUGAS PRAKTIKUM

1. Buat aplikasi form pemesanan makanan/minuman seperti Gambar 7.

No	Makanan/Minuman	Harga	Pesan
1	Bakso Istimewa	@ 12000	2
2	Soto Spesial	@ 10000	1
3	Mie Ayam Super	@ 15000	0
4	Es Degan	@ 5000	1
5	Es Campur	@ 7000	2
Jumlah Total			53000
Diskon			10000
Jumlah Dibayar			43000

Batal

Gambar 7. Aplikasi pemesanan makanan/minuman

Spesifikasi:

- Field Harga, Jumlah Total, Diskon, dan Jumlah Dibayar bersifat read-only
- Jika pesan diisi, maka field jumlah total secara otomatis akan menghitung totalnya, termasuk juga field jumlah dibayar.

Rumus:

$$\text{Jumlah Total} = \sum (\text{harga} * \text{pesan})$$

$$\text{Jumlah Dibayar} = \text{Jumlah Total} - \text{Diskon}$$

- Jika pembelian lebih dari 50000, maka field diskon secara otomatis akan berisi nilai diskon sebesar 10000 (tidak berlaku kelipatannya) dan otomatis pula field jumlah dibayar akan berkurang
- Jika tombol Batal diklik, maka semua field Pesan akan dibersihkan