



**um**  
The Learning  
University

**PROBIS**  
PUSAT BISNIS  
Universitas Negeri Malang

# MODUL

## PRAKTIKUM PERKULIAHAN

**PROGRAM STUDI  
TEKNIK INFORMATIKA**

<http://probis.um.ac.id>

**Modul Praktikum  
Teknik Informatika (IFT)  
Tahun ajaran (2017-2018)**



Oleh :

Adhif Ali Musyafa, S.Kom  
Joko Teguh, S.T  
M. Makinudin, S.Kom  
Novan Satya Pratama, S.Kom  
Rudy Kurniawan, S.Pd  
Vivin Ayu Lestari, S.Pd, M.Kom  
Ahmad Nurul Latif A.Md

**PROBIS PUSAT BISNIS  
UNIVERSITAS NEGERI MALANG  
2018**

# DAFTAR ISI

HALAMAN MODUL .....	1
DAFTAR ISI .....	2
<b>BAB 1 PROGRAMAN WEB 1</b>	
BAB 1 Dasar HTML .....	4
BAB 2 Link dan Gambar .....	15
BAB 3 Form .....	21
BAB 4 Layout Web Page .....	31
BAB 5 Cascading Style Sheet .....	38
<b>BAB 2 SISTEM BASIS DATA</b>	
MODUL 1 Dasar MyQSL .....	51
MODUL 2 DML .....	58
MODUL 3 Data Retrieval Lenguage (DRL) .....	70
MODUL 4 JOIN .....	82
<b>BAB 3 PEMOGRAMAN VISUAL</b>	
BAB 1 Form,Kontrol,Properti dan Event .....	92
BAB 2 Seleksi Kondisi .....	100
BAB 3 Perulangan .....	111
BAB 4 Timer dan Menu .....	115
BAB 5 Pemograman Database 1 .....	120
<b>BAB 4 ALGORITMA PEMOGRAMAN</b>	
PRAKTIKUM 1 Algoritma dan Flowchart .....	140
PRAKTIKUM 2 Label Button .....	149
PRAKTIKUM 3 Label,ComboBox, dan Button .....	152
PRAKTIKUM 4 Komentar .....	156
PRAKTIKUM 5 Penyatuan terstruktur .....	162
PRAKTIKUM 6 Perulangan .....	166
PRAKTIKUM 7 Membuat Menu,Pop up menu, dan page control .....	170

# 1

## PEMROGRAMAN WEB I

## BAB 1 Dasar HTML

### 1.1 Format Penulisan HTML

Untuk membuat suatu halaman HTML standar digunakan tag <html>, <head>,<body> beserta tag penutupnya masing-masing. Susunan tag – tag tersebut membentuk dokumen HTML sebagai berikut :

```
1 <html>
2   <head>
3     ... informasi tentang dokumen HTML
4   </head>
5
6   <body>
7     ... informasi yang akan ditampilkan di web browser
8   </body>
9 </html>
10
```

Gambar 1.1 Format Penulisan HTML

### 1.2 Memulai Membuat Dokumen dengan Format HTML

Langkah – langkah untuk memulai membuat dokumen HTML sebagai berikut

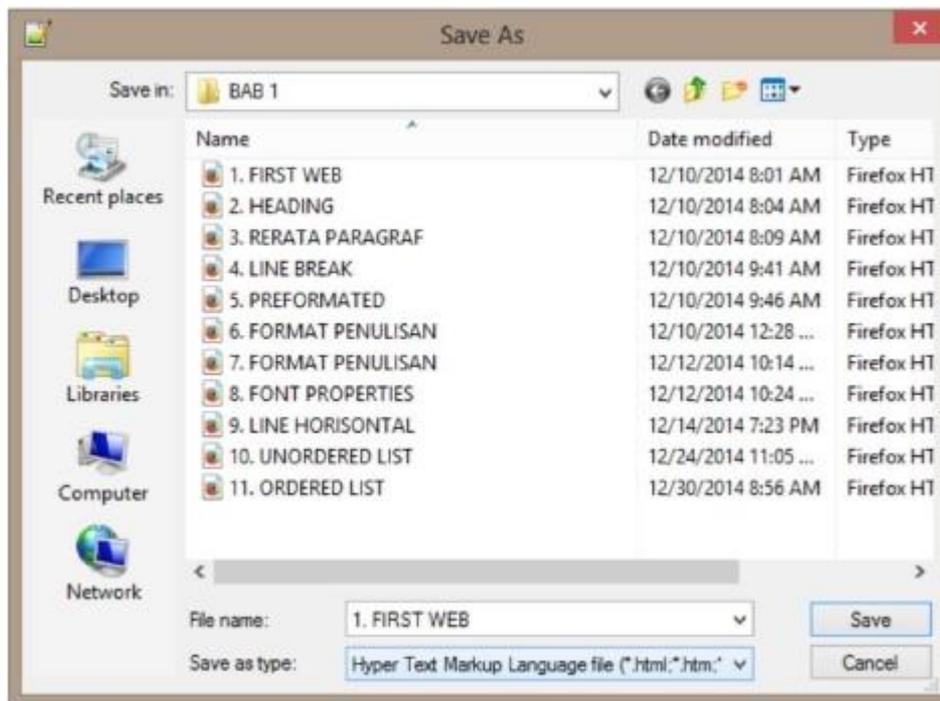
1. Buka notepad++ sebagai editor dokumen HTML.
2. Buat skrip sederhana seperti di bawah ini di notepad++ :

```
1 <html>
2 <head>
3   <title> web pertama mantap </title>
4 </head>
5 <body>
6
7   ini adalah web pertama saya dan sukses bisa!!!
8
9 </body>
10 </html>
```

Gambar 1.2 Format Penulisan HTML

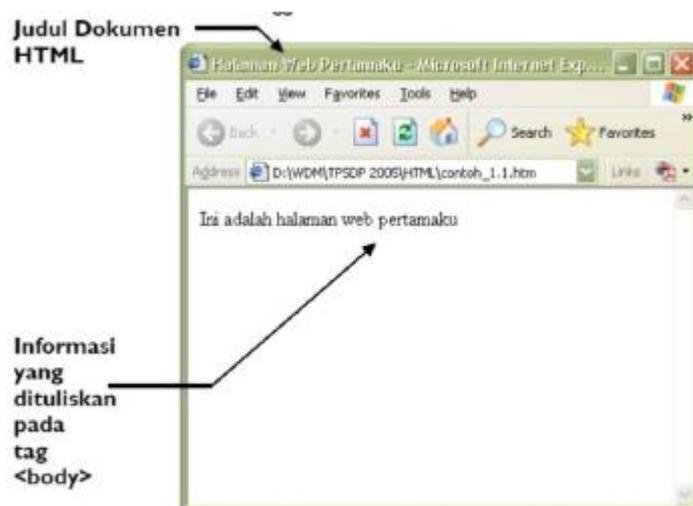
3. Simpan dokumen yang telah dibuat dengan menggunakan ekstensi .htm atau .html

Path directory : ....\nama\_folder\xxx.html



**Gambar 1.3** Format Penyimpanan HTML

4. Untuk melihat hasil dari dokumen HTML yang telah dibuat, buka dokumen tersebut menggunakan web browser. Dengan klik Run – Launch in IE.



**Gambar 1. 4.** Tampilan Hasil Eksekusi Contoh

### 1.3. Tag Dasar dalam HTML

#### 1.3.1 Heading

Heading merupakan sekumpulan kata yang menjadi judul atau sub judul dalam dokumen HTML, dengan berbagai ukuran yang berbeda.

Pada dasarnya untuk menyatakan suatu Heading, digunakan tag <Hx> dimana x adalah level 1

sampai 6. Tag Heading atau <Hx> adalah tag berpasangan, yaitu tag yang mempunyai tag awal <Hx> dan tag penutup </Hx>

Format : <H1>.....</H1> sampai <H6>.....</H6>

```
<HTML>
<BODY>
<H1>Heading 1 adalah 24 Point</H1>
<br>
<H2>Heading 2 adalah 18 Point</H2>
<br>
<H3>Heading 3 adalah 14 Point</H3>
<br>
<H4>Heading 4 adalah 12 Point</H4>
<br>
<H5>Heading 5 adalah 10 Point</H5>
<br>
<H6>Heading 6 adalah 7 Point</H6>
<br>
```

### 1.3.2 Paragraf

#### a. Melakukan Formating Paragraf

Jika kita akan membuat sebuah website, dapat dipastikan kita juga kan menyajikan informasi-informasi yang diperlukan. HTML juga mendefinisikan elemen khusus, untuk mendefinisikan teks dengan makna khusus.

HTML menggunakan unsur-unsur seperti <b> dan <i> untuk memformat output, seperti teks tebal atau miring.

Elemen format yang dirancang untuk menampilkan jenis khusus dari teks:

- ☐teks tebal : menggunakan tag <b>
- ☐teks penting : menggunakan tag <strong>
- ☐teks miring : menggunakan tag <i>
- ☐teks menekankan : menggunakan tag <em>
- ☐teks menandai : menggunakan tag <mark>
- ☐teks kecil : menggunakan tag <small>
- ☐teks menghapus : menggunakan tag <del>
- ☐teks memasukkan : menggunakan tag <ins>
- ☐subscript / pangkat dibawah : menggunakan tag <sub>
- ☐superscripts / pangkat di atas : menggunakan tag <sup>

#### b. Menggunakan Line Break

Jika kita biasa menggunakan tombol Enter untuk ganti baris, maka hal itu tidak berlaku dalam penulisan dokumen HTML. Seberapa banyak kita memencet tombol Enter, maka tidak kan dianggap 1 pun.

Dalam HTML tag Line Break dapat digunakan untuk membuat baris baru layaknya fungsi tombol Enter pada keyboard. Line break menggunakan tag **<br>**

### c. Preformatted Text

Jarak antar kata suatu teks atau kalimat di dalam dokumen web adalah sebesar satu spasi. Agar browser web menampilkan sesuai dengan dokumen web tersebut di dalam editor teks, maka teks yang bersangkutan harus di beri tanda tag **<pre>** di awalnya dan **</pre>** di akhir teks. Dengan tag **<pre>**, yang mempunyai kepanjangan "**preformatted**" teks akan di tampilkan dalam browser dengan

ukuran font dengan lebar - fix. Tag ini juga akan menjaga spasi, baris baru, dan tap sesuai dengan aslinya pada saat di tampilkan dalam browser.

### d. Mengolah Properti Dari Font

Saat proses koding, ada kalanya kita harus melakukan pewarnaan, atau bahkan menentukan jenis huruf dari font yang kita pakai dengan alasan tertentu, mungkin karena info penting agar lebih bisa diperhatikan dll. Kita dapat menggunakan tag `style="property:value"` untuk mengubah style property dari font. Gunakan atribut style untuk elemen styling HTML

- **background-color** untuk warna latar belakang
- **color** untuk warna teks
- **font-family** untuk font teks
- **font-size** untuk ukuran teks
- **text-align** untuk perataan teks

### e. Membuat garis Horizontal

Garis horizontal (horizontal line) digunakan untuk memisahkan halaman web secara visual. Tag yang digunakan adalah **<hr>**. Beberapa atributnya adalah ALIGN, SIZE, dan WIDTH. Tag hr seperti halnya tag br (**<br>**) adalah tag yang berdiri sendiri, jadi tidak perlu ditutup dengan tag penutup.

### 1.3.3 List

List merupakan bentuk umum yang biasa kita gunakan untuk menampilkan daftar dengan urutan tertentu. Dalam HTML, dikenal 3 (tiga) istilah list : ordered list, unordered list dan definition list.

### a. List Tanpa Urutan (Unordered Lists)

Unordered list adalah metode mengurutkan daftar dengan menggunakan simbol atau special character pada HTML. nama lain dari unordered list sendiri adalah bullet list. Bullet atau onordered list ini umumnya digunakan apabila kita ingin membuat suatu daftar yang terlepas dari suatu urutan-urutan tertentu. Untuk list jenis ini, pembuatannya sangat mirip dengan Ordered list, hanya saja ada perbedaan pada jenis synthax yang digunakan dimana unordered list akan menggunakan `<ul>`.

### b. List Berurut (Ordered Lists)

Ordered list adalah suatu metode mengurutkan daftar dengan menggunakan angka, abjad dan angka romawi pada HTML. Dalam ordered list ini akan ada dua jenis elemen yang dibutuhkan yaitu `<ol>` yang merupakan singkatan ordered list dan `<li>` yang merupakan singkatan list item

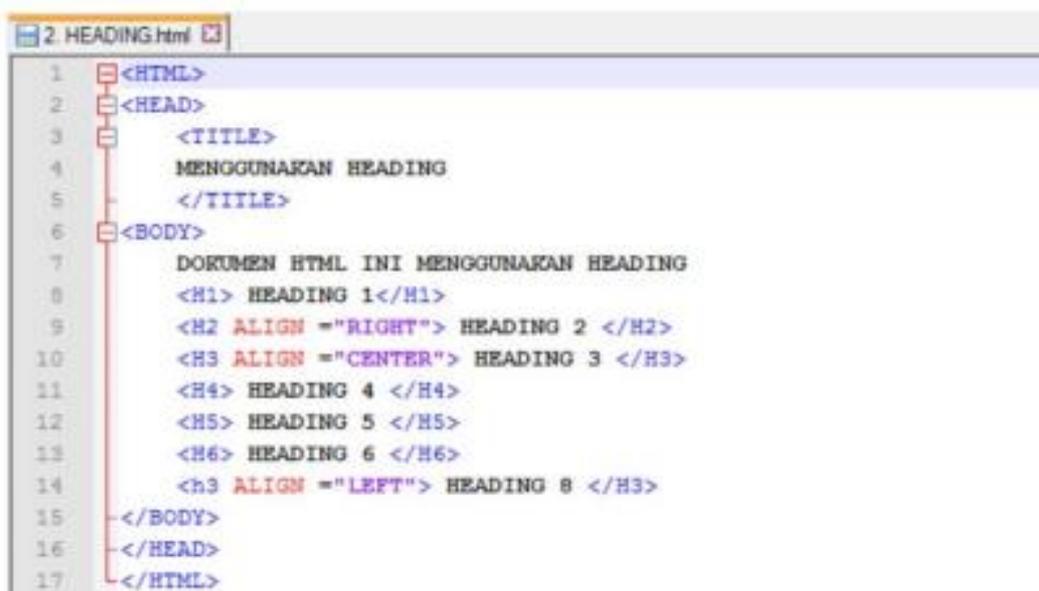
### c. Definition List

Definition list merupakan jenis daftar yang umumnya digunakan untuk mendefinisikan sesuatu. Definition listsendiri sebenarnya sudah terlepas dari ketiga daftar diatas, dalam artian mempunyai jenis opening dan closing tag yang benarbenar berbeda dari tiga jenis daftar diatas. Definition list akan menggunakan tiga jenis tag : Diawali dengan `<dl>` yang merupakan singkatan definition list; `<dt>` yang merupakan singkatan definition term (digunakan untuk kata yang akan didefinisikan); `<dd>` definition description yang merupakan penjelasan dari suatu istilah (`<dt>`).

## LATIHAN

1. Menggunakan Heading.

Simpan dengan nama **latihan1.html**



```
1 <HTML>
2 <HEAD>
3   <TITLE>
4     MENGGUNAKAN HEADING
5   </TITLE>
6 <BODY>
7   DORUMEN HTML INI MENGGUNAKAN HEADING
8   <H1> HEADING 1</H1>
9   <H2 ALIGN = "RIGHT"> HEADING 2 </H2>
10  <H3 ALIGN = "CENTER"> HEADING 3 </H3>
11  <H4> HEADING 4 </H4>
12  <H5> HEADING 5 </H5>
13  <H6> HEADING 6 </H6>
14  <h3 ALIGN = "LEFT"> HEADING 8 </H3>
15 </BODY>
16 </HEAD>
17 </HTML>
```

Gambar 1.5. Menggunakan Heading

## 2. Memformat Paragraf.

Simpan dengan nama **Latihan2.html**

```
1 <html>
2 <head>
3   <title> Melakukan Formating Paragraf </title>
4 </head>
5 <body>
6   <p> Paragraf pertama </p>
7   <p> Paragraf pertama ini menggunakan rata kiri </p>
8   <p></p>
9   <p align="center"> paragraf kedua </p>
10  <p align="center"> paragraf kedua ini menggunakan rata tengah </p>
11  <p></p>
12  <p align="right"> paragraf ketiga </p>
13  <p align="right"> paragraf ketiga ini menggunakan rata kanan </p>
14 </body>
15 </html>
```

**Gambar 1.6.** Skrip contoh format paragraph

## 3. Penggunaan Line Break dalam HTML.

Simpan dengan nama **Latihan3.html**

```
1 <html>
2 <head>
3   <title> Penggunaan Line Break </title>
4 </head>
5 <body>
6   Menggunakan Line Break <br><br>
7   Dengan menggunakan line break, kita dapat memindahkan kalimat<br>
8   setelah elemen tersebut satu bari dibawahnya.<br>
9   seperti menggunakan ENTER pada word processing.
10 </body>
11 </html>
```

**Gambar 1.7.** Contoh penggunaan Line Break

## 4. Penggunaan Pre dalam HTML.

Simpan dengan nama **Latihan4.html**

```
1 <html>
2 <head>
3   <title> Penggunaan Preformatted </title>
4 </head>
5 <body>
6   <pre>
7     Pengetikan ini dicoba menggunakan Preformatted Text
8
9     Dimana format penulisan ditampilkan dalam web sesua dengan format yang dilakukan
10    Seperti contoh melakukan satu tab seperti ini.
11  </pre>
12 </body>
13 </html>
```

**Gambar 1.8.** Contoh skrip penggunaan preformatted

5. Memformat Bentuk Tulisan dalam HTML ( penebalan, penulisan miring dan garis bawah pada font).

Simpan dengan nama **Latihan5.html**

```
1 <html>
2   <head>
3     <title> Menggunakan Teknik Format Penulisan </title>
4   </head>
5   <body>
6     <!-- membuat tulisan menjadi Tebal -->
7     <b> Tulisan ini dicetak TEBAL </b>
8   <p>
9     <!-- membuat tulisan menjadi Miring -->
10    <i> Tulisan ini dicetak MIRING </i>
11  <p>
12    <!-- membuat tulisan menjadi bergaris bawah -->
13    <u> Tulisan ini dicetak BERGARIS BAWAH</u>
14  </body>
15 </html>
```

**Gambar 1.9.** Contoh skrip format penulisan

6. Penggunaan Superscript dan Subscript dalam HTML.

Simpan dengan nama **Latihan6.html**

```
1 <html>
2   <head>
3     <title> Melakukan Format Penulisan </title>
4   </head>
5   <body>
6     <!-- Membuat Tulisan menjadi TEBAL -->
7     <b> Tulisan ini telah dicetak TEBAL </b>
8   <p>
9     <!-- Membuat Tulisan tertulis MIRING -->
10    <i> Tulisan ini telah dicetak MIRING </i>
11  <p>
12    <!-- Membuat Tulisan bergaris bawah -->
13    <u> Tulisan ini dicetak BERGARIS BAWAH </u>
14  <p>
15    <!-- Membuat Format Superscript -->
16    Dalam matematika, x pangkat 2 ditulisa dengan X <sup>2</sup>
17  <p>
18    <!-- Membuat Format Subscript -->
19    Rumus kimia untuk Oksigen adalah O <sub>2</sub>
20  </body>
21 </html>
```

**Gambar 1.10.** Contoh skrip format penulisan

7. Mengolah property text.

Simpan dengan nama **Latihan7.html**

```
1 <html>
2   <head>
3     <title>
4       Pengolahan FONT dengan Properties
5     </title>
6   </head>
7   <body>
8     <font size = "1" face="Tahoma" color = "#FF0000"> tulisan ini bertuliskan dengan ukuran 1, dengan font Tahoma dan berwarna HIJAU
9     </font>
10    <br>
11    <font size = "4" face="arial" color="green" > tulisan ini bertuliskan dengan ukuran 4, dengan font Arial dan berwarna HIJAU
12    </font>
13  </body>
14 </html>
```

**Gambar 1.11.** Contoh skrip property text

## 8. Menambahkan Garis Horisontal pada HTML.

Simpan dengan nama **Latihan8.html**

```
1 <html>
2   <head>
3     <title>
4       Menambahkan Horizontal Rules Pada Website
5     </title>
6   </head>
7   <body>
8     Menambahkan garis mendatar pada halaman html
9     <hr size = "15">
10    <hr width="100" align = "left" >
11    <hr color="blue">
12
13    <hr size = "20" width = "300" noshade>
14  </body>
15 </html>
```

**Gambar 1.12.** Contoh skrip untuk penambahan garis horizontal

## 9. Penggunaan Unordered List pada HTML

Simpan dengan nama **Latihan9.html**

```
1 <html>
2   <head>
3     <title> Penggunaan Listing </title>
4   </head>
5   <body>
6     <h3> Menggunakan Unordered List </h3>
7     <hr>
8     <h4> Ordered List Dengan Type Default </h4>
9     <ul>
10      <li> Agung Nugroho </li>
11      <li> Mulya Sulityono </li>
12      <li> Bernadhed </li>
13    </ul>
14
15    <h4> Unordered List Dengan Type Circle </h4>
16    <ul type = "circle">
17      <li> Agung Nugroho </li>
18      <li> Mulya Sulityono </li>
19      <li> Bernadhed </li>
20    </ul>
21
22    <h4> Unordered List Dengan Type Disc </h4>
23    <ul type ="disc">
24      <li> Agung Nugroho </li>
25      <li> Mulya Sulityono </li>
26      <li> Bernadhed </li>
27    </ul>
28
29    <h4> Unordered List Dengan Type Square </h4>
30    <ul type ="square">
31      <li> Agung Nugroho </li>
32      <li> Mulya Sulityono </li>
33      <li> Bernadhed </li>
34    </ul>
35  </body>
36 </html>
```

Gambar 1.13. Contoh skrip untuk penggunaan Unordered List

## 10. Penggunaan Ordered List pada HTML

Simpan dengan nama **Latihan10.html**

```
1 <html>
2 <head>
3   <title> Menggunakan Ordered List </title>
4 </head>
5 <body>
6   <h3> Menggunakan Ordered List </h3>
7   <hr>
8   <h4> Ordered List Menggunakan Type Default </h4>
9   <ol>
10    <li> Agung Nugroho </li>
11    <li> Bernadhed </li>
12    <li> Malya Sulistryono </li>
13  </ol>
14
15  <h4> Ordered List Menggunakan Type a </h4>
16  <ol type = "a">
17    <li> Agung Nugroho </li>
18    <li> Bernadhed </li>
19    <li> Malya Sulistryono </li>
20  </ol>
21
22  <h4> Ordered List Menggunakan Type I </h4>
23  <ol type = "I">
24    <li> Agung Nugroho </li>
25    <li> Bernadhed </li>
26    <li> Malya Sulistryono </li>
27  </ol>
28
29  <h4> Ordered List Menggunakan Type Atribut Start </h4>
30  <ol start = "14">
31    <li> Agung Nugroho </li>
32    <li> Bernadhed </li>
33    <li> Malya Sulistryono </li>
34  </ol>
35 </body>
36 </html>
```

**Gambar 1.14.** Contoh skrip untuk penggunaan Ordered List

### Studi Kasus

1. Buatlah Skrip HTML yang mampu menampilkan format tulisan seperti dibawah ini

$H_2O$

$X^2 + 4X + 4$

**Linux** adalah sistem operasi yang bersifat *open source*.

2. Buatlah Skrip HTML yang mampu menampilkan informasi seperti dibawah ini



## Bab 2. Link dan Gambar

### Tujuan

1. Mahasiswa mampu membuat link antar dokumen HTML
2. Mahasiswa dapat membuat link ke bagian tertentu dari dokumen yang sama
3. Mahasiswa mampu membuat dokumen HTML dan memasukkan gambar ke halaman HTML
4. Mahasiswa mampu membuat link dari gambar

**Overview** Kemudahan dalam aplikasi berbasis web salah satunya adalah bisa menghubungkan satu dokumen dengan dokumen lainnya, baik dalam satu server aplikasi web maupun dengan server aplikasi web yang berbeda di seluruh dunia maya. HTML menyediakan hypertext link yang merupakan daerah teks (ataupun gambar) yang bisa link (menyambungkan) ke dokumen HTML yang lain.

Seiring berkembangnya teknologi internet, bukan hanya teks dan gambar yang bisa link ke dokumen tertentu, file multimedia pun sudah dapat dibuat link ke dokumen yang diinginkan.

**2.1. Link** Kekuatan utama dokumen HTML terletak pada **hypertext link** atau **hyperlink** atau lebih singkat lagi disebut **link**. Dengan hyperlink ini kita bisa membuka dokumen HTML lain atau langsung menuju ke bagian tertentu sebuah dokumen HTML. Hyperlink ini dapat diletakkan pada teks tertentu ataupun pada sebuah image, bila diletakkan pada teks maka teks tersebut (secara default) akan digaris-bawahi dan warnanya menjadi berbeda. Sintaksis penulisan :

```
<A href=url_tujuan> nama_link </A>
```

Beberapa atribut yang sering ditambahkan atau sering digunakan dalam penggunaan sintak diatas adalah seperti dibawah ini:

**Tabel. 2.1.** Atribut dalam Link

Atribut	Kegunaan
Href	Menunjukkan url yang dituju
Name	Memberikan nama pada bagian tertentu pada dokumen
Target	Menunjukkan target ditampilkannya link
Title	Menunjukkan title dari link ketika cursor digerakkan di sekitar area hyperlink

### 2.1.2 Membuat Link Ke Dalam Dokumen

Seringkali halaman web adalah halaman yang panjang dan selalu membukanya dari posisi awal akan sangat melelahkan bagi pengguna. Untuk itu akan lebih memudahkan bila kita langsung merujuk pada bagian tertentu sebuah halaman web. Caranya dengan menyisipkan Tag Anchor <a> dengan atribut "name" di bagian dokumen yang akan dituju. Untuk perintah standartnya :

```
<a name="nama_section"> </a>
```

Untuk menggunakan link yang akan merujuk ke bagian dokumen itu, ditambahkan '#nama\_section' menjadi :

```
<a href="sebuah_halaman.html#nama_section">langsung ke  
nama section</a>
```

## 2.2. Gambar

Untuk menyisipkan gambar kita dapat memanfaatkan tag <IMG>. Format file gambar yang bisa ditampilkan bisa bermacam-macam, misalnya jp, jpeg, pcx, gif, psd, dan sebagainya. Pada umumnya, perancang web hanya menggunakan kombinasi dari tiga format file gambar saja yaitu jpeg, gif dan psd. Sintaksis yang digunakan secara standar adalah

```
<IMG src="url_file" width="img_width" height="img_height"  
vspace="10" hspace="10" alt="alt_teks">
```

### 2.2.1 Membuat obyek gambar menjadi sebuah hyperlink

Apabila kita mengakses sebuah website, seperti facebook atau search engine google terdapat logo dari dua website tersebut. Logo tersebut merupakan gambar dimana kita sering dapat melakukan klick terhadap gambar tersebut dan merujuk ke sebuah link tertentu. Berikut contoh sintaks penggunaan obyek gambar menjadi hyperlink :

```
<img src='url gambar' alt='nama alternatif gambar' width='tinggi' height='lebar'>
```

- Tag <img> adalah tag HTML untuk mendefinisikan sebuah gambar.
- Atribut src adalah sumber/url dari gambar.
- Atribut alt adalah untuk memberikan nama alternatif gambar. Apabila gambar tidak bisa ditampilkan, maka yang muncul adalah nama alternatifnya.
- Atribut height dan width adalah untuk mengatur ukuran gambar.

contoh: 

## LATIHAN

1. Membuat Link Antar Dokumen Dalam Contoh ini, dapat dilihat dari script di bawah ini, sebelumnya siapkan halaman html dengan nama **Magister Teknik Informatika.html**, **Sistem Informasi.html**, **Teknik Informasi.html** (disimpan di direktori anda masing-masing) dengan script masing-masing seperti dibawah ini:

```

1 <html>
2   <head>
3     <title> Magister Teknik Informatika </title>
4   </head>
5   <body>
6     <h4> Magister Teknik Informatika </h4>
7   </body>
8 </html>

```

```

1 <html>
2   <head>
3     <title> Sistem Informatika </title>
4   </head>
5   <body>
6     <h4> Sistem Informatika </h4>
7   </body>
8 </html>

```

```

1 <html>
2   <head>
3     <title> Teknik Informatika </title>
4   </head>
5   <body>
6     <h4> Teknik Informatika </h4>
7   </body>
8 </html>

```

**Gambar 2.1.** Skrip untuk direksional link dari HTML utama

Setelah kita mengetikkan skrip diatas dan melakukan penyimpanan, selanjutnya kita akan membuat halaman HTML baru yang nanti ada link untuk merujuk ke tiap halaman HTML diatas. Simpan dengan nama **latihan11.html**.

```

1 <html>
2   <head>
3     <title> STMIK Amikom Yogyakarta </title>
4   </head>
5   <body>
6     <a href = "http://www.amikom.ac.id">
7       </a>
8
9     <center>
10      <h3> STMIK Amikom </h3>
11      <h5> Jln Ringroad Utara Depok Sleman Yogyakarta </h5>
12      <br>
13      [ <a href="Teknik Informatika .html"> Teknik Informatika </a> ]
14      [ <a href="Sistem Informatika .html" target="blank"> Sistem Informatika </a> ]
15      [ <a href="E:/pembrograman web/PW/SAS 2/LINK DOMINEN/Magister Teknik Informatika.html"> Magister Teknik Informatika </a> ]
16      <br>
17      <br>
18      [ <a href="http://www.barnadlaggr.com"> BAAK </a> ]
19      [ <a href="http://www.budayamenanamhoaya.com"> Dikti </a> ]
20    </center>
21  </body>
22 </html>

```

**Gambar 2.2.** Skrip untuk Halaman utama

## 2. Membuat Link ke dalam Dokumen

Simpan dengan nama **latihan12.html**.

```
1 <html>
2   <head>
3     <title> Membuat Link Ke Bagian Dalam Dokumen </title>
4   </head>
5   <body>
6     <h4> Membuat Link Ke Bagian Dalam Dokumen </h4>
7     <a name="Bab1"> </a>
8     <b> Bab 1 </b>
9     <br>
10    Pada bab ini akan dijelaskan tutorial membuat dokumen HTML. <br>
11    Mulai dari pemahaman, pembuatan dokumen HTML sederhana sampai membuat website. <br>
12    Sebagai pendahuluan , coba klik link ini untuk
13    <a href="#sejarah"> menuju ke bagian lain </a> pada dokumen ini.
14    <p>&nbsp;</p>
15    <p>&nbsp;</p>
16    <p>&nbsp;</p>
17    <p>&nbsp;</p>
18    <p>&nbsp;</p>
19    <p>&nbsp;</p>
20    <p>&nbsp;</p>
21    <p>&nbsp;</p>
22    <p>&nbsp;</p>
23    <p>&nbsp;</p>
24    <p>&nbsp;</p>
25    <p>&nbsp;</p>
26    <p>&nbsp;</p>
27    <p>&nbsp;</p>
28    <p>&nbsp;</p>
29    <p>&nbsp;</p>
30    <p>&nbsp;</p>
31    <a name="sejarah"> </a>
32    <b> 1.1 Sekilas Sejarah Internet </b> <br>
33    Sub Bab ini merupakan bagian dari bagian bab yang dituju ketika link pada Bab1 di klik. <br>
34    Untuk kembali ke Bab1, klik <a href="#Bab1"> disini </a>
35  </body>
36 </html>
```

### 3. Menyisipkan gambar pada HTML.

Simpan dengan nama **latihan13.html**.

```
1 <html>
2   <head>
3     <title> Menambahkan Image </title>
4   </head>
5   <body>
6     <h4> Menambahkan Image Ke Dalam Dokumen HTML </h4>
7     
8     <br>
9     <h4> Mengatur Border Pada Image </h4>
10    
11    <br>
12    <h4> Alternating Text </h4>
13    Pada Image di bawah ini terdapat Alternating Text.
14    Silahkan gerakan mouse melewati Image <br>
15    
16  </body>
17 </html>
```

**Gambar 2.5.** Skrip untuk menyisipkan gambar didalam HTML

### 4. Membuat gambar menjadi hyperlink.

```
1 <html>
2   <head>
3     <title> Halaman Awal </title>
4   </head>
5   <body>
6     <h4> Selamat Datang di Personal Web Saya. </h4>
7     Untuk Melanjutkan silahkan klik gambar dibawah ini
8     <br>
9     <a href="4. a Gambar Sebagai Link Sub .html">
10    
11  </a>
12  <br>
13  <br>
14  <br>
15  <br>
16  Selamat Menjelajah
17 </body>
18 </html>
```

**Gambar 2.7.** Skrip untuk menyisipkan sebuah link dalam gambar

Jika diperhatikan, skrip diatas terdapat sebuah sintak yang mengarah sebuah halaman html tertentu. Oleh karenanya, buatlah sesuai nama halaman html diatas agar bisa dirujuk oleh halaman tersebut. Simpan dengan nama **gambar sebagai link sub.html**.

```
1 <html>
2   <head>
3     <title> Bios </title>
4   </head>
5   <body>
6     <a href= "http://www.bernadtagger.com">
7       
8     </a>
9     Saya adalah seorang tenaga pendidik di sebuah Kampus Swasta Jogja
10  </body>
11 </html>
12
```

**Gambar 2.8.** Skrip untuk menyisipkan sebuah link tautan

Jika pengetikan skriptidak terjadi kesalahan, maka obyek gambar diatas jika diklik maka akan merujuk ke halaman html yang telah kita buat setelahnya.

## Bab 3. Form

### Tujuan

1. Mahasiswa mampu membuat form pada dokumen HTML
2. Mahasiswa memahami cara penggunaan dari fasilitas Form pada halaman dokumen HTML
3. Mahasiswa mampu menginputkan field sesuai dengan kegunaan

### Overview

Salah satu fasilitas yang disediakan oleh dokumen HTML memungkinkan kita untuk melakukan "interaksi lebih" terhadap halaman dokumen tersebut. Semisal, kemampuan dokumen HTML mampu menyediakan fasilitas yang dapat menerima masukan atau isian data dari user. Data isian user ini nantinya akan dapat diproses lebih lanjut menjadi informasi yang dibutuhkan oleh user maupun oleh pemilik situs tersebut. Teknik pengisian ini dilakukan menggunakan elemen Form. Form dalam HTML adalah suatu bagian yang berfungsi sebagai input atau masukan dari pengguna yang kemudian akan diproses atau diolah untuk dapat digunakan sesuai dengan kebutuhan.

Contohnya proses pengiriman data, browse, hapus, penyuntingan data dan lain sebagainya.

### 3.1 Menu Form

Form dalam HTML adalah suatu bagian yang berfungsi sebagai input atau masukan dari pengguna yang kemudian akan diproses atau diolah untuk dapat digunakan sesuai dengan kebutuhan. Contohnya proses pengiriman data, browse, hapus, penyuntingan data dan lain sebagainya. Cara penulisan form dalam html adalah.

```
<form action="_url_" method="get|post" enctype="">  
    ... elemen - elemen yang ditambahkan dalam form ...  
</form>
```

```
form name="nama_form" method="post/get" action="url">...isi form...</form>
```

Pada penulisan form "nama form" adalah nama form yang sedang digunakan, method adalah metode atau cara yang digunakan untuk menyimpan data ke server sedangkan action adalah alamat atau bisa juga file lain yang digunakan untuk memproses input dari form kedalam server.

### 3.2 Jenis-jenis media input dalam Form

Pada bagian form tersedia kontrol-kontrol input yang digunakan untuk masing-masing keperluan dalam memasukan data input. Dan kontrol form terdiri dari :

### 3.2.1 Kontrol Berupa Text

Pengisian informasi dalam bentuk teks pada form HTML dapat dilakukan dengan menggunakan dua buah elemen: `textarea` dan `input`. `textarea` digunakan untuk masukan teks yang terdiri dari beberapa baris, sementara `input` digunakan untuk masukan teks yang hanya satu baris.

Penggunaan elemen `textarea` dapat dilakukan dengan sangat sederhana, hanya langsung memasukkan tag-nya saja:

```
<textarea>
</textarea>
```

selain memberikan tag kosong seperti di atas, kita juga dapat mengisikan tag untuk mendapatkan nilai masukan standar:

```
<textarea>
  Contoh isi textarea
</textarea>
```



**Gambar 3.1** Contoh Penggunaan Elemen TextArea

Pengaturan panjang dan lebar dari `textarea` dapat dilakukan melalui CSS, dengan menggunakan properti `height` dan `width`.

Walaupun dapat mengisikan teks dengan banyak sekaligus, `textarea` tentunya tidak dapat digunakan untuk seluruh kasus pengisian data. Seringkali kita menginginkan pengguna hanya mengisikan data singkat, tanpa isi teks yang banyak. Untuk jenis masukan seperti itu, kita dapat menggunakan elemen `input`:

```
<input type="text">
```

yang akan menghasilkan elemen masukan seperti berikut:



**Gambar 3.2** Contoh Penggunaan Elemen Input

Perhatikan juga bahwa kita menggunakan atribut `type` pada elemen ini untuk menentukan data-data yang akan diisikan. Terdapat banyak jenis tipe data yang dapat kita isikan, dan browser akan menyesuaikan jenis masukan teks yang ada, sesuai dengan atribut `type` yang kita tentukan. Nilai-nilai yang dapat diisikan pada atribut `type` yaitu :

1. color
2. datetime-local
3. number
4. tel
5. week
6. date
7. email
8. range
9. time
10. datetime
11. month
12. search
13. url
14. password
15. text
16. file

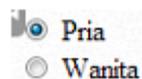
### 3.2.2 Kontrol Berupa Radio Button

Radio button di gunakan untuk menyatakan pilihan yang bersifat tunggal, umumnya pada web di gunakan untuk kelengkapan pertanyaan seperti pada pendaftaran online, biasanya pertanyaan itu adalah pilihan untuk menyetakan jenis kelamin, agama, ataupun sebagainya. Radio Button ini akan sering kita temui saat kita berselancar di dunia maya yang terkait dengan proses input data seperti pendaftaran dll.

Radio button dibuat dalam HTML dengan menggunakan elemen input, dengan atribut type bernilai : code`radio`:

```
<input type="radio" name="sex" value="pria"> Pria <br>  
<input type="radio" name="sex" value="wanita"> Wanita
```

dalam pembuatan elemen radio button, kita juga wajib menambahkan dua atribut lainnya, yaitu name dan value. Atribut name digunakan untuk memberitahukan browser bahwa radio button dengan atribut name yang sama adalah merupakan kumpulan radio button yang sama, sehingga pengguna tidak boleh memilih dua buah pilihan pada radio button tersebut. Atribut value digunakan untuk menyimpan nilai yang ingin kita kirimkan ke server.



**Gambar 3.3** Contoh Penggunaan Elemen Radio Button

### 3.2.3 Kontrol Berupa Checkbox

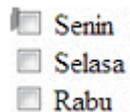
Berbeda dengan radio button yang hanya memungkinkan user memilih satu pilihan, pada input type checked box Anda dapat memilih satu atau beberapa pilihan, atau tidak memilih sama sekali. Pilihan ini biasanya dipakai untuk memasukkan data yang sifatnya opsional. Anda dapat menggunakan “Checkbox” pada atribut. Input checkbox adalah sebuah masukan untuk

memilih sesuatu opsi/ pilihan. Jika dipilih maka pengguna akan men-check (mencentang) dan jika tidak memilih akan dibiarkan saja.

Elemen ini sama seperti radio button, dibuat dengan elemen input yang nilai atribut type-nya berisikancheckbox.

```
<input type="checkbox" name="day" value="senin"> Senin <br>  
<input type="checkbox" name="day" value="selasa"> Selasa <br>  
<input type="checkbox" name="day" value="rabu"> Rabu
```

Seperti yang dapat dilihat, atribut name dan value juga wajib dimiliki oleh checkbox, dengan alasan yang sama untuk radio button.



**Gambar 3.4** Contoh Penggunaan Elemen Checkbox

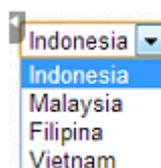
Meskipun dapat memberikan pilihan nilai kepada pengguna, radio button maupun checkbox sangat tidak ideal untuk digunakan pada pilihan yang memiliki nilai yang sangat banyak. Misalnya, pilihan negara yang ditinggali oleh pengguna dari ratusan negara yang ada di dunia akan jika ditampilkan pada radio button akan menyebabkan daftar yang dipaparkan sangat banyak dan sulit dibaca oleh pengguna. Dalam kasus seperti ini lebih baik kita menggunakan **dropdown list**.

### 3.2.4 Dropdown list

Pembuatan elemen dropdown dilakukan dengan menggabungkan dua elemen, yaitu select dan option. Elemeselect membungkus seluruh elemen option yang ada, untuk membentuk sebuah dropdown. Elemen option sendiri merupakan nilai dari dropdown yang diinginkan. Perhatikan kode di bawah:

```
<select name="country">  
  <option value="indonesia">Indonesia</option>  
  <option value="malaysia">Malaysia</option>  
  <option value="filipina">Filipina</option>  
  <option value="vietnam">Vietnam</option>  
</select>
```

yang akan menghasilkan:



**Gambar 3.5** Contoh Penggunaan Elemen Dropdown

dan jika ingin memungkinkan pengguna memilih beberapa pilihan kita dapat menambahkan atribut multiple pada elemen select:

```

<select name="country" multiple>
  <option value="indonesia">Indonesia</option>
  <option value="malaysia">Malaysia</option>
  <option value="filipina">Filipina</option>
  <option value="vietnam">Vietnam</option>
</select>

```



**Gambar 3.6** Contoh Penggunaan Elemen Dropdown dengan Pilihan Banyak

### 3.2.5 Button

Pada saat kita melakukan browsing di dunia maya, dan mengakses sebuah website. Sering kita menemui tombol-tombol fungsi seperti reset, submit, send dan lain sebagainya. Dan sering pula kita sering menemui gambar yang dibuat fungsi seperti tombol.

#### 3.2.5.1 Submit

Tombol ini berfungsi untuk memanggil url yang sudah didefinisikan di atribut action. Pembuatan tombol submit juga dilakukan dengan menggunakan elemen input, yang atribut type-nya diisikan dengan nilai submit, seperti berikut:

```

<input type="submit" name="submit" value="Masukkan Form">

```

Perhatikan bahwa berbeda dengan elemen-elemen input sebelumnya, nilai dari atribut value diproses menjadi teks dari tombol yang ditampilkan ke pengguna. Hal ini menyebabkan kita tidak lagi perlu menambahkan teks setelah elemen form, seperti pada radio button ataupun checkbox.



**Gambar 3.7** Contoh Penggunaan Tombol Submit

#### 3.2.5.2 Reset

Tombol ini berfungsi untuk mengembalikan form ke kondisi awal (mengosongkan nilai semua elemen yang ada pada form). Pembuatan tombol dilakukan sama persis seperti pada tombol submit, dengan perbedaan nilai pada atribut type, yang diisikan dengan reset pada tombol penghapusan ini.

```

<input type="reset" name="reset" value="Kosongkan Form">

```

### 3.2.6 Organisasi Elemen Form

Mengetahui bagaimana membuat elemen-elemen masukan / input pada form masih merupakan langkah awal dalam pembuatan form HTML. Elemen-elemen masukan

form jika ditampilkan tanpa informasi tambahan tidaklah berguna, karena pengguna tidak dapat mengetahui data apa yang harus diisikan ke dalam elemen-elemen tersebut.

Untuk memberikan informasi tambahan kepada pengguna, HTML juga menyediakan elemen-elemen yang dapat digunakan sebagai penanda dari elemen masukan form. Adapun elemen-elemen yang dapat digunakan untuk tujuan tersebut yaitu label, fieldset, dan legend.

### 3.2.6.1 Label

Elemen label memberikan fasilitas untuk menambahkan teks pada elemen masukan form. Elemen ini diisikan dengan deskripsi dari elemen masukan yang ingin ditambahkan, dan harus memiliki atribut for. Atribut for pada elemen label berisikan nilai yang sama dengan atribut id pada elemen masukan form. Pengisian nilai yang sama akan mengikatkan elemen label dengan elemen masukan, sehingga pengguna dapat langsung mengisikan nilai dengan melakukan klik pada teks yang dihasilkan oleh elemen label.

Perhatikan kode berikut di mana isi dari atribut for pada label adalah sama dengan isi atribut id pada elemen input.

```
<label for="username">Username</label>
<input type="text" name="username" id="username">
```

### 3.2.6.2 Fieldset

Elemen fieldset merupakan elemen yang digunakan untuk membungkus beberapa elemen masukan form, untuk menandakan bahwa elemen-elemen tersebut merupakan elemen masukan yang berada pada satu grup yang sama, atau saling berhubungan.

Secara standar, elemen fieldset akan memberikan border di sekitar grup elemen-elemen di dalamnya, yang tentunya dapat diubah dengan menggunakan CSS.

Berikut adalah contoh penggunaan fieldset :

```
<fieldset>
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="text" name="password" id="password">
</fieldset>
```

yang hasil eksekusinya adalah:



Username  Password

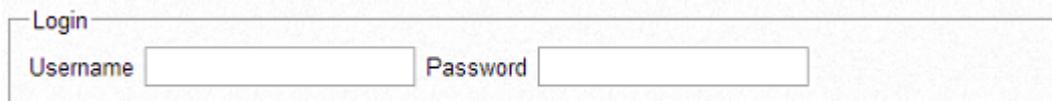
**gambar 3.8** Contoh Penggunaan Fieldset

### 3.2.6.2 Legend

Elemen legend digunakan untuk memberikan judul pada sebuah fieldset. Penggunaan elemen legend sangat sederhana, hanya dengan menambahkan elemen tersebut sebagai child pertama dari fieldset, seperti berikut:

```
<fieldset>
  <legend>Login</legend>
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="text" name="password" id="password">
</fieldset>
```

yang akan menghasilkan tampilan seperti berikut:



**Gambar 3.9** Contoh Penggunaan Legend

dan tentunya tampilan dari legend dapat diubah dengan menggunakan CSS.

### 3.2.7 Validasi Masukan pada Form

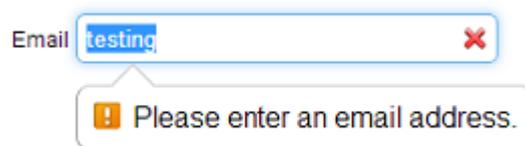
Seringkali dalam pengembangan form pada aplikasi, terdapat elemen-elemen yang tidak wajib diisi oleh pengguna dan elemen-elemen yang wajib diisi oleh pengguna. Aplikasi kemudian akan melakukan pengecekan terhadap nilai masukan dari pengguna, apakah nilai yang wajib ada sudah diisi atau belum. Jika nilai belum diisi maka aplikasi akan menolak form pengguna, dengan pesan kesalahan yang menjelaskan bahwa terdapat isian yang wajib diisi oleh pengguna. Proses pengecekan kewajiban isi atau berbagai batasan lain ini dikenal dengan nama validasi.

Validasi pada elemen-elemen form HTML dilakukan dengan beberapa cara, yaitu:

1. Atribut required pada elemen-elemen masukan yang wajib diisi. Jika pengguna tidak mengisi elemen maka browser secara otomatis akan memberikan peringatan kepada pengguna dan membatalkan pengiriman form. Contoh penggunaan atribut ini yaitu:

```
<input type="text" name="username" required>
```

2. Penggunaan atribut type yang benar, untuk memastikan browser dapat melakukan pengecekan nilai yang diisi pengguna dengan format sesuai dengan yang dispesifikasikan pada standar HTML. Misalnya, jika menggunakan `type="email"`, maka pengguna tidak dapat mengisi bukan email sebagai masukan form.



**Gambar 3.10** Contoh Pesan Kesalahan Pengisian Email pada HTML

## LATIHAN

1. Penggunaan input text dalam sebuah form.

Simpan dengan nama **latihan14.html**

```
1 <html>
2   <head>
3     <title> Menyisipkan Elemen Berupa Text </title>
4   </head>
5   <body>
6     <font type=arial black font size="6" > Data Diri </font>
7     <br> </br>
8     <font ty=arial size ="5" >
9     nama : <input type = "text" name = "nama">
10    <br>
11    <br>
12    nim : <input type = "text" name = "nim" value="" maxlength="9" size="9">
13    </font>
14  </body>
15 </html>
```

**Gambar 3.11.** Contoh skrip untuk penggunaan input text dalam sebuah form

2. Penggunaan Input Text berupa Password.

Simpan dengan nama **latihan15.html**

```
1 <html>
2   <head>
3     <title> Input Password </title>
4   </head>
5   <body>
6
7     Password : <input type ="password" name="nama" size="11" maxlength="9">
8
9   </body>
10 </html>
```

**Gambar 3.12.** Contoh skrip untuk penggunaan input text dalam sebuah password menu

3. Penggunaan Input text berupa radio button

Simpan dengan nama **latihan16.html**

```
1 <html>
2   <head>
3     <title> Penggunaan Radio Button </title>
4   </head>
5   <body>
6     Jenis Kelamin :
7     <br>
8     <input type="radio" name="jenis_kelamin" value="L"> Laki-laki
9     <br>
10    <input type="radio" name="jenis_kelamin" value="P"> Perempuan
11    <br>
12
13  </body>
14 </html>
```

**Gambar 3.13.** Contoh skrip untuk penggunaan input text berupa radio button

4. Penggunaan checkbox dalam HTML.

Simpan dengan nama **latihan17.html**

```
1 <html>
2   <head>
3     <title> Penggunaan Check Box </title>
4   </head>
5   <body>
6     Sebutkan Hobi Anda, boleh lebih dari satu :
7     <br>
8     <input type="checkbox" name="hobi_1" value="Menonton"> Menonton
9     <br>
10    <input type="checkbox" name="hobi_2" value="Game"> Bermain Game
11    <br>
12    <input type="checkbox" name="hobi_3" value="Shopping"> Belanja
13    <br>
14    <input type="checkbox" name="hobi_4" value="Mancing"> Memancing
15    <br>
16    <input type="checkbox" name="hobi_5" value="Olahraga"> Olahraga
17    <br>
18  </body>
19 </html>
```

**Gambar 3.14.** contoh skrip untuk penggunaan input text berupa checkbox

5. Penggunaan submit, reset, dan image sebagai button.

Simpan dengan nama **latihan18.html**

```
1 <html>
2   <head>
3     <title> Konfigurasi Button </title>
4   </head>
5   <body>
6     <form>
7
8     <input type="submit" value="Kirim" name="Submit">
9     <input type="reset" value="Ulangi" name="Reset">
10    <input type="button" value="Batal" name="Cancel">
11    <input type="image" name="Gambar" img src="research.png" width="194" height="90">
12
13    </form>
14  </body>
15 </html>
```

**Gambar 3.15.** Contoh skrip penggunaan input berupa button

6. Penggunaan Text area dalam Form.

Simpan dengan nama **latihan19.html**

```
1 <html>
2   <head>
3     <title> Membuat Text Area </title>
4   </head>
5   <body>
6     Deskripsi Diri:
7     <br>
8     <textarea name="deskripsi" cols="45" rows="5" align="center"> Tuliskan deskripsi singkat tentang anda.
9     </textarea>
10  </body>
11 </html>
```

**Gambar 3.16.** Contoh skrip penggunaan input berupa text area

## 7. Penggunaan Select Dropdown dalam Form.

Simpan dengan nama **latihan20.html**

```
1 <html>
2 <head>
3   <title> Menggunakan Menu Dropdown </title>
4 </head>
5 <body>
6   
7 </img>
8 <br>
9   Silahkan pilih orientasi minat :
10 <br>
11 <select name="Orientasi Minat">
12   <option value="dg"> Design Grafis </option>
13   <option value="pj"> Programmer Java </option>
14   <option value="sa"> Sistem Analis </option>
15   <option value="ds"> Dagelan Saja </option>
16 </select>
17 <br>
18 <br>
19 <select name="Orientasi Kemampuan" Multiple>
20   <option value="dg"> Design Grafis </option>
21   <option value="pj"> Programmer Java </option>
22   <option value="sa"> Sistem Analis </option>
23   <option value="aj"> Ahli Jaringan </option>
24   <option value="it"> Trainer IT </option>
25   <option value="ds"> Dagelan Saja </option>
26 </select>
27 </body>
28 </html>
```

**Gambar 3.17.** Contoh skrip penggunaan Select Dropdown

## **Bab 4. Layout Web Page**

### **Tujuan**

1. Mahasiswa mampu membuat dan memformat tabel pada halaman dokumen HTML
2. Mahasiswa dapat memahami dalam pembuatan frame pada dokumen HTML
3. Mahasiswa mampu memahami penggunaan frame tersebut
4. Mahasiswa mampu memahami pembagian tampilan layar ke dalam beberapa pembagian, yang setiap bagian dihubungkan dengan dokumen HTML yang terpisah
5. Mahasiswa mampu mengatur properti frame border
6. Mahasiswa mampu menentukan target frame cells dari hypertext links

### **Overview**

Saat kita membuat sebuah halaman website, dapat dibuat menggunakan tabel dan frame. Hal ini memang tidak ada perbedaan didalamnya yang begitu signifikan, tapi saat pada proses pengaksesan halaman web mungkin akan sangat berpengaruh terhadap respons time yang diperlukan.

Tabel merupakan cara untuk menampilkan informasi dalam halaman web dengan bentuk kolom dan baris. Hampir semua web yang mempunyai kualitas dan profesional, dirancang dengan menggunakan tabel. Layaknya sebuah spreadsheet yang memiliki sel dan berisi angka-angka, tabel dalam web juga mempunyai sel yang berisi link, gambar dan text. Frame akan dapat membagi sebuah HTML menjadi beberapa dokumen HTML. Dimana masing-masing bagian terhubung dengan dokumen HTML yang terpisah.

### **4.1 Tabel**

#### **4.1.1 Membuat Sebuah Tabel dan Judul Tabel**

Tabel digunakan untuk menyajikan data dalam bentuk kolom dan baris. Umumnya setiap kolom menunjukkan data yang sejenis, dan setiap baris yang terdiri atas kolom-kolom menunjukkan kelompok data dalam satu kesatuan.

Sebuah tabel mempunyai judul, tempat anda menjelaskan kolom/baris yang dilibatkan, baris untuk informasi dan sel untuk setiap itemnya. Pada tabel berikut, kolom pertama berisi informasi header, setiap baris menjelaskan sebuah tag tabel HTML dan setiap sel berisi sebuah pasangan tag atau penjelasan dari fungsi tag.

Berikut ini disajikan elemen-elemen yang sering digunakan dalam pembuatan tabel dalam HTML.

**Tabel 4.1** Elemen-elemen dalam pembuatan tabel HTML

Elemen	Penjelasan
<code>&lt;TABLE&gt;...&lt;/TABLE&gt;</code>	Mendefinisikan sebuah tabel dalam HTML jika atribut border dituliskan, maka browser akan menampilkan tabel dengan border.
<code>&lt;CAPTION&gt;...&lt;/CAPTION&gt;</code>	Mendefinisikan tulisan untuk judul tabel. Posisi default dari judul adalah ditengah pada bagian paling atas tabel. Atribut <code>align="bottom"</code> dapat digunakan untuk menempatkan judul pada bagian bawah tabel. Catatan: judul dapat diberi tag apa saja
<code>&lt;TR&gt;...&lt;/TR&gt;</code>	Menspesifikasikan sebuah baris tabel dalam tabel. Anda dapat mendefinisikan atribut untuk seluruh baris: <code>align(left, center, right)</code> dan/atau <code>valign (top,middle,bottom)</code> .
<code>&lt;TH&gt;...&lt;/TH&gt;</code>	Mendefinisikan sel header tabel. Secara default teks dalam sel ini ditebalkan dan ditampilkan di tengah.
<code>&lt;TD&gt;...&lt;/TD&gt;</code>	Mendefinisikan sebuah sel data tabel. Secara default teks dalam sel ini ditampilkan rata kiri, dan ditengah secara vertikal. Sel data table dapat berisi atribut untuk mendefinisikan karakteristik dari sel dan isinya.

#### 4.1.2 Melakukan format Baris dan Kolom

Setiap tabel pasti akan terdiri dari atas baris-baris yang diwakili dengan sebuah tag `<tr>` dan juga mempunyai kolom-kolom yang menggunakan sebuah tag `<td>`. Setiap baris dan kolom pada tabel memiliki atribut nya masing-masing.

#### 4.1.3 Pembahasan Tabel Lebih Lanjut

Pada pembahasan selanjutnya, kita akan belajar menyisipkan sebuah gambar kedalam sebuah susunan tabel. Tabel di dalam sebuah website dapat berisi hyperlink, gambar, gambar yang menunjuk ke sebuah hyperlink dan teks dengan huruf yang berwarna.

### 4.2 Frame

Framing dalam HTML dapat kita pergunakan untuk membuat tampilan halaman HTML yang terbagi-bagi menjadi beberapa dokumen HTML, dimana didalam setiap bagian halaman HTML yang salah satu atau beberapa bagian berganti-ganti isinya sedangkan bagian lain tetap sehingga dapat melakukan efisiensi terhadap bandwidth koneksi internet dan mempercepat proses download page secara keseluruhan.

Berikut adalah sintaks umum yang sering digunakan dalam membuat frame

```
<HTML>
<HEAD>
</HEAD>
<FRAMESET BORDER="#" { [ROWS | COLS] } = { #,[#,[...]] }>
  <FRAME SRC="url" NAME="FrameName">
</FRAMESET>
</HTML>
```

Tabel 4.1 Atribut yang digunakan dalam frame

Atribut	Fungsi
<b>Frameset cols</b>	Membuat frame vertikal dengan lebar kolom tertentu
<b>Frameset rows</b>	Membuat frame horisontal dengan tinggi baris tertentu
<b>Frame src</b>	Memasukkan dokumen HTML ke dalam frame
<b>Noframe</b>	Memasukkan body text untuk browser yang tidak menampilkan frame

**LATIHAN**

1. Penggunaan table dalam HTML.

Simpan dengan nama **latihan21.html**

```

1 <html>
2 <head>
3 <title> Pembuatan Tabel </title>
4 </head>
5 <body>
6 <font face="Tahoma" size="2" color="blue">
7 <table border="0.8">
8 <caption align="top"> <b> <u> Daftar Alamat </u> </b> </caption>
9 <tr bgcolor="blue" align="center">
10 <td width="40"> No. </td>
11 <td width="150"> Nama </td>
12 <td width="200"> Alamat </td>
13 </tr>
14 <tr bgcolor="skyblue">
15 <td align="center"> 1. </td>
16 <td> Muiya Sulistyono </td>
17 <td> Sleman </td>
18 </tr>
19 <tr bgcolor="orange">
20 <td align="center"> 2. </td>
21 <td> Hendra Kurniawan </td>
22 <td> Solo </td>
23 </tr>
24 <tr bgcolor="skyblue">
25 <td align="center"> 3. </td>
26 <td> Agung Nugroho </td>
27 <td> Borobudur </td>
28 </tr>
29 <tr bgcolor="orange">
30 <td align="center"> 4. </td>
31 <td> Agus Fatkhurahman </td>
32 <td> Magelang </td>
33 </tr>
34 </table>
35 </body>
36 </html>

```

**Gambar 4.1.** Skrip Penggunaan table dalam HTML

## 2. Penggunaan formatting baris dan kolom dalam skrip HTML.

Simpan dengan nama **latihan22.html**

```
1 <html>
2 <head>
3 <title> Mengatur Baris dan Kolom </title>
4 </head>
5 <body>
6 <table border="3" cellpadding="3" cellspacing="5">
7 <caption> Daftar Order Barang </caption>
8 <thead>
9 <th> No </th>
10 <th> Barang </th>
11 <th> Jenis </th>
12 <th> Harga </th>
13 <th> Jumlah </th>
14 <th> Total Barang </th>
15 </thead>
16 <tbody>
17 <tr>
18 <td align="center"> 1 </td>
19 <td> Pociari Sweat </td>
20 <td rowspan="2"> Makanan </td>
21 <td align="right"> 5 </td>
22 <td align="right"> Rp. 22.500 </td>
23 </tr>
24 <tr>
25 <td align="center"> 2 </td>
26 <td> Nutrisari Jeruk </td>
27 <td align="right"> 3 </td>
28 <td align="right"> Rp. 7.350 </td>
29 </tr>
30 <tr>
31 <td align="center"> 3 </td>
32 <td> Filus Garuda </td>
33 <td rowspan="3"> Cemilan </td>
34 <td align="right"> 2 </td>
35 <td align="right"> Rp. 9.800 </td>
36 </tr>
37 <tr>
38 <td align="center"> 4 </td>
39 <td> Tango Coklat </td>
40 <td align="right"> 1 </td>
41 <td align="right"> Rp. 6.450 </td>
42 </tr>
43 <tr>
44 <td align="center"> 5 </td>
45 <td> Makanan Ringan </td>
46 <td align="right"> 2 </td>
47 <td align="right"> Rp. 12.900 </td>
48 </tr>
49 <tr>
50 <td colspan="5" style="border: 3px solid red; background-color: gray; height: 30px; text-align: center; vertical-align: middle;">
51 </td>
52 </tr>
53 </tbody>
54 </table>
55 </body>
56 </html>
```

Gambar 4.2. Skrip contoh untuk melakukan formatting baris dan kolom dalam table

## 3. Penyisipan gambar dan text dalam Tabel. Simpan dengan nama

**latihan23.html**

```
1 <html>
2 <head>
3 <title> Sisipan Gambar Dalam Tabel </title>
4 </head>
5 <body>
6 <table border="2">
7 <caption> Dokument Pemotretan </caption>
8 <thead>
9 <tr>
10 <th> 1. </th>
11 <th> 2. </th>
12 <th> 3. </th>
13 </tr>
14 <tr align="center">
15 <td> Bike-Mount </td>
16 <td> Bike-Modelling </td>
17 <td> Bike-Concept </td>
18 </tr>
19 <tr align="center">
20 <td align="center">
21 
22 <td align="center">
23 
24 <td align="center">
25 <a href="2. Format Baris dan Kolom.html">
26 
27 </td>
28 </tr>
29 </tbody>
30 </table>
31 </body>
32 </html>
```

Gambar 4.3. Skrip contoh untuk penyisipan gambar dan text

#### 4. Contoh Penggunaan menu menggunakan Frame HTML

Sebelum membuat sebuah frame, kita buat beberapa dokumen web yang nanti akan ditampilkan ke dalam tiap frame. Misal kita akan membuat tiga halaman yang akan ditampilkan yang terdiri dari halaman **header.html**, **pascasarjana.html**, **menuutama.html**, **home.html** dan **menu.html**. sebuah tips jika akan membuat sebuah website dengan sub subnya, akan lebih efektif lagi jika kita membuat subsub nya terlebih dahulu. Misal membuat header, menu-menu baru dibuat frame utama berisi sub-sub diatas.

##### a. Header

Simpan dengan nama **header.html**

```
1 <html>
2   <head>
3     <title> STMIK Amikom Yogyakarta </title>
4   </head>
5   <body>
6     <center>
7       <font face="Tahoma">
8         <h3> STMIK Amikom Yogyakarta </h3>
9         <h5> Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta
10        Telp: (0274) 884201 - 207 Fax: (0274) 884208 Kodepos: 55283 </h5>
11        
12      </font>
13    </center>
14  </body>
15 </html>
```

Gambar 4.4. Skrip contoh untuk pembuatan header

##### b. Menu Utama

Selanjutnya adalah kita membuat menu utama. Dimana menu utama ini adalah menu paling depan sebagai "menu penyambut" ketika user mengakses halaman website yang kita buat. Berikut ini adalah skrip yang dibuat. Simpan dengan nama **menuutama.html**

```
1 <html>
2   <head>
3     <title> Sejarah Amikom </title>
4   </head>
5   <body>
6     <h3> STMIK Amikom </h3>
7     <p>
8       <font face="Tahoma">
9         Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta (selanjutnya disebut STMIK AMIKOM YOGYAKARTA)
10        merupakan salah satu perguruan tinggi swasta yang berkedudukan di Provinsi DIY Kabupaten Sleman di bawah naungan
11        Yayasan AMIKOM Yogyakarta. Beberapa jurusan yang terdapat di STMIK Amikom adalah sebagai berikut:
12      </font>
13    <ol>
14      <li> S2. Magister Teknik Informatika </li>
15      <li> S1. Teknik Informatika </li>
16      <li> S1. Sistem Informatika </li>
17      <li> D3. Teknik Informatika </li>
18      <li> D3. Manajemen Informatika </li>
19    </ol>
20  </body>
21 </html>
```

Gambar 4.5. Skrip contoh untuk pembuatan menu utama

c. Sub Menu Pasca Sarjana

Pada langkah selanjutnya adalah membuat sub menu **PascaSarjana.html**. Mari kita ketikkan skrip dibawah ini kedalam aplikasi editor.

```
1 <html>
2   <head>
3     <title> Magister Teknik Informatika Amikom </title>
4   </head>
5   <body>
6     <img src = "mti logo.jpg" width="168" height="131" alt="STMIK Amikom" align="left">
7     <h3> Magister Teknik Informatika Amikom </h3>
8     <hr>
9   <pre>
10
11
12
13
14   Program Magister Teknik Informatika Program Pascasarjana STMIK AMIKOM Yogyakarta diarahkan pada
15   hasil lulusan yang berbudi luhur, berjiwa entrepreneur dan memiliki kualifikasi sebagai berikut:
16   1. Kemampuan merancang dan mengelola proyek-proyek perangkat lunak
17   2. Kemampuan menguasai aspek teknis dan manajemen dari pemanfaatan teknologi informasi
18   3. Kemampuan mengevaluasi dan menghadapi dampak dari teknologi informasi
19   4. Kemampuan mempunyai daya inovatif dalam menghasilkan produk/jasa teknologi informasi
20   5. Kemampuan merancang, mengembangkan, dan mengimplementasikan proyek dalam bidang media digital
21
22   <table width="100%" border="1" cellpadding="1" cellspacing="1">
23     <tr bgcolor="#e2e2e2">
24       <td>Tahapan
25       <td>Mata Kuliah Reguler</td>
26     </tr>
27     <tr>
28       <td rowspan="7">Semester 1</td>
29       <td>Analaysis and System Design</td>
30     </tr>
31     <tr><td>Design and Networking Management</td></tr>
32     <tr><td>Software Engineering</td></tr>
33     <tr><td>Information System</td></tr>
34     <tr><td>Database Management System</td></tr>
35     <tr><td>Praktikum Aplikasi Internet</td></tr>
36     <tr><td>Praktikum Dasar Jaringan Komputer</td></tr>
37   </table>
38 </body>
39 </html>
```

Gambar 4.6. Skrip contoh untuk pembuatan sub menu “pasca sarjana”

d. Menu Pilihan Margin Kiri

Langkah selanjutnya adalah membuat bagian dimana untuk memberikan akses kepada pengguna dengan menyajikan kumpulan link-link dari sub menu yang ada. Contoh skrinnya adalah sebagai berikut. Simpan dengan nama **menu.html**

```
1 <html>
2   <head>
3     <title> STMIK Amikom Yogyakarta </title>
4   </head>
5   <body>
6     <center>
7       [ <a href="menu utama .html" target="main"> Home </a> ]<br>
8       [ <a href="Pasca .html" target="main"> MTI </a> ]<br>
9       [ <a href="#"> S1 Teknik Informatika </a> ]<br>
10      [ <a href="#"> S1 Sistem Informasi </a> ]
11    </center>
12  </body>
13 </html>
```

e. Frame

Setelah kita telah selesai membuat sub menu dari website tersebut, selanjutnya kita akan membuat frame. Agar dapat memilah satu halaman website menjadi beberapa bagian yang berisi tiap halaman tersebut. Berikut ini adalah contoh skrip yang akan kita buat. Simpan dengan nama **home.html**

```
1 <html>
2 <head>
3 <title>Menggunakan Frame</title>
4 </head>
5 <frameset rows=200,*>
6 <frame src="Header .html" scrolling="no" name="atas">
7 <frameset cols=250,*>
8 <frame src="menu .html" name="kiri">
9 <frame src="menu" name="main">
10 </frameset>
11
12 </frameset>
13 <body>
14 </body>
15 </noframes>
16 </html>
```

**Gambar 4.8.** Skrip contoh untuk pembuatan frame

## Bab 5. Cascading Style Sheet (CSS)

### Tujuan

1. Mahasiswa Memahami penggunaan CSS dalam HTML
2. Mahasiswa Mampu mengaplikasikan dan membuat CSS

### Overview

**Cascading Style Sheets** (CSS) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan sebuah dokumen yang ditulis dalam bahasa markup. CSS diperkenalkan untuk pengembangan website kurang lebih pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap style.

Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C). CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca braille. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

Setelah CSS distandarisasikan, Internet Explorer dan Netscape melepas browser terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

CSS = Cascading Style Sheets ( Bahasa lembar Gaya ). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language. Jika kita berbicara dalam konteks web, bisa di artikan secara bebas sebagai : CSS merupakan bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML.

### Pengertian CSS

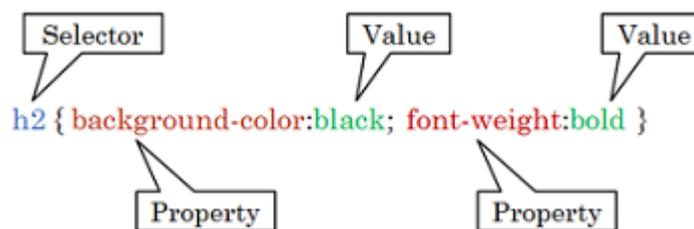
- CSS adalah singkatan dari Cascading Style Sheets. Berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web.
- Perancangan desain text dapat dilakukan dengan mendefinisikan fonts (huruf) , colors (warna), margins (ukuran), latar belakang (background), ukuran font (font sizes) dan lain-lain. Elemen-elemen seperti colors (warna) , fonts (huruf), sizes (ukuran) dan spacing (jarak) disebut juga “styles”.
- Cascading Style Sheets juga bisa berarti meletakkan styles yang berbeda pada layers (lapisan) yang berbeda.
- CSS terdiri dari style sheet yang memberitahukan browser bagaimana suatu dokumen akan disajikan.
- Fitur-fitur baru pada halaman web lama dapat ditambahkan dengan bantuan style sheet.

- Saat menggunakan CSS, Anda tidak perlu menulis font, color atau size pada setiap paragraf, atau pada setiap dokumen. Setelah Anda membuat sebuah style sheet, Anda dapat menyimpan kode tersebut sekali saja dan dapat kembali menggunakannya bila diperlukan.

### Keuntungan Menggunakan CSS

- CSS memberikan keseragaman pada halaman web.
- Dengan CSS dapat menghemat banyak waktu dan pekerjaan berulang. Saat menggunakan CSS, perubahan tidak perlu dilakukan dalam setiap halaman web. Anda hanya perlu membuat perubahan dalam style sheet.
- CSS memungkinkan Anda untuk memuat halaman web Anda dengan mudah.
- Layers (Lapisan), seperti item pop-up, dapat digunakan dalam dokumen.
- CSS membantu Anda memelihara halaman web Anda dengan mudah dan efektif.

### 5.1 Pengertian Selector, Property dan Value pada CSS



Gambar 5.1 Selector, Property dan Value pada CSS

#### 5.1.1 Selector

Karena kode CSS digunakan untuk mengubah/memanipulasi tampilan dari tag HTML, CSS membutuhkan suatu cara untuk „mengaitkan“ atau **menghubungkan** kode CSS dengan tag HTML yang sesuai. Hal inilah yang dimaksud dengan **Selector** dalam CSS. Sesuai dengan namanya, selector digunakan untuk mencari bagian web yang ingin dimanipulasi atau yang ingin di-style. Misalnya : “cari seluruh tag <p>”, atau “cari seluruh tag HTML yang memiliki atribut class=“warning”” atau “cari seluruh link yang ada di dalam tag <p>”.

Selector paling dasar dari CSS adalah tag dari HTML itu sendiri, misalnya: tag p, i, h1, li,dll. Selector didalam CSS dapat menjadi kompleks tergantung kebutuhannya.

#### 5.1.2 Property

**Property** CSS adalah jenis style, atau elemen apa yang akan diubah dari sebuah tag HTML. CSS memiliki puluhan **property** yang dapat digunakan agar menampilkan hasil akhir yang kita inginkan. Hampir semua **property** dalam CSS dapat dipakai untuk seluruh **selector**.

Jika selector digunakan misalnya untuk “mencari seluruh tag <p>”, maka property adalah “efek apa yang ingin dimanipulasi dari tag p tersebut”, seperti ukuran text, warna text, jenis fontnya, dll.

### 5.1.3 Value

**Value** CSS adalah nilai dari **property**. Misalkan untuk property **backgroundcolor** yang digunakan untuk mengubah warna latar belakang dari sebuah selector, **value** atau nilainya dapat berupa **red, blue, black**, atau **white**.

## 5.2 Penempatan Sebuah CSS

Ada tiga cara untuk menempatkan sebuah CSS ke dalam sebuah halaman web, ketiga cara ini dapat digunakan untuk memformat halaman web dengan style yang diinginkan.

### 5.2.1 Inline Style Sheet

CSS dalam posisi inline style sheet dituliskan menjadi satu dengan halaman web yang akan diatur style-nya dan menjadi bagian dari body. Penulisan styledilakukan dengan cara menambahkan atribut title pada elemen (tag) HTML yang akan diatur style-nya. Oleh karena itu, untuk mengimplementasikan CSS pada halaman web, semua tag harus diformat secara independen.

Dengan menggunakan model penempatan inline style sheet, jika ada sebuah tag HTML yang digunakan berulang kali dalam sebuah halaman web, pembuat website dapat mengimplementasikan style yang berbeda pada tag tersebut. Ketika menggunakan inline style sheet untuk memanipulasi halaman web, pembuat web hanya dapat menggunakan satu property saja pada tag HTML yang akan dimanipulasi.

### 5.2.2 Embedded Style Sheet

Sama halnya dengan CSS dalam posisi inline style sheet, penulisan CSS dalam posisi embedded style sheet juga menjadi satu dengan halaman web yang akan diatur style-nya, hanya saja, posisi CSS menjadi bagian dari header (berada diantara tag <head>) dengan menambahkan tag <style type="text/css">.

Dengan menggunakan model penempatan CSS sebagai embedded style sheet, pembuat web cukup satu kali mendefinisikan style yang akan dikenakan pada tag – tag yang berada dalam halaman web. Jika ada sebuah tag yang digunakan secara berulang, secara otomatis akan mempunyai style yang sama, berbeda dengan model inline style sheet yang mengharuskan pembuat web menentukan style pada tag – tag yang digunakan berulang kali dan memungkinkan untuk menentukan style yang berbeda pada tag tersebut.

### 5.2.3 Linked Style Sheet

Berbeda dengan 2 (dua) model penempatan CSS sebelumnya, menggunakan linked style sheet berarti harus menyediakan sebuah file CSS khusus berisi berbagai format style yang terpisah dari halaman web. File ini nantinya akan dipanggil oleh halaman web yang membutuhkan pengaturan style. Dengan menggunakan model ini, style akan terpusat pada sebuah file, sehingga jika ada perubahan style, pembuat website tidak perlu merubah di semua halaman web yang dibuat, tetapi cukup dengan merubah style yang didefinisikan pada file CSS.

Dengan menggunakan model ini, style akan terpusat pada sebuah file, sehingga jika ada perubahan style, pembuat website tidak perlu merubah di semua halaman web yang dibuat, tetapi cukup dengan merubah style yang didefinisikan pada file CSS.

### 5.3 Penggunaan CSS Pada Halaman Web

Kita dapat menggunakan salah satu dari ketiga penempatan CSS yang ada, namun tidak menutup kemungkinan juga kita dapat menggunakan dua atau ketiganya dalam sebuah website jika memang diperlukan.

#### 5.3.1 Pengolahan Font

Salah satu tag HTML yang biasa digunakan untuk memanipulasi font adalah paragraf (<p>). Beberapa tag yang lain juga memungkinkan untuk dimanipulasi atau ditambahkan CSS untuk memanipulasi font jika pada tag HTML tersebut akan berisi tulisan / text, seperti : <th><td>, <a>, <li>, <h1>...<h6>, dan tag – tag lain yang memungkinkan berisi text. Bentuk manipulasi font yang dimungkinkan dapat dilihat pada tabel di bawah ini :

Tabel 5.1 Font Properties

Nama Property	Value	Contoh Penggunaan
Font-family	Nama font	P { font-family: arial, Helvetica }
Font-size	Nilai integer dengan pilihan satuan : em pt px %	P {font-size:24em}
Font-style	Oblique italic normal	P {font-style:italic}
Font-variant	Normal small-caps	P {font-variant:small-caps}
Font-weight	Normal bold bolder 100-900	P {font-weight:bold}

#### 5.3.2 Memanipulasi Color dan Background

Bentuk lain dari memanipulasi halaman web adalah dengan menentukan warna pada tulisan, menambahkan warna background, atau dengan menambahkan gambar sebagai background. Beberapa property yang bisa digunakan untuk memanipulasi beberapa hal di atas, ditampilkan pada tabel di bawah ini.

**Tabel 5.1** Properties Warna dan Gambar latar

Nama Property	Value	Contoh Penggunaan
Background-color	Rangkaian 6 digit heksadesimal atau nama warna	Body { Background-color : green; }
Background-Image	url dimana image disimpan	Body { Background-image : url (gambar1.jpg); }
Background-attachment	Fixed scroll inherit	Body { Background-attachment : scroll; }
Background-Repeat	Inherit repeat no-repeat repeat-x repeat-y	Body { Background-repeat : repeat; Background-image : url (image/gambar1.jpg); }
Background-position	Bottom center left right top inherit	Body { Background-repeat : no-repeat; Background-image : url (image/gambar1.jpg); Background-position : bottom center }
Color	Rangkaian 6 digit	H1 { color : red }

## 5.4 Mengenal Jenis-jenis Selector Dasar CSS

**Selector** adalah sebuah pola (**pattern**) yang digunakan untuk „mencari“ suatu tag di dalam HTML. Analogi untuk selector, misalnya: mencari semua tag p, atau mencari seluruh tag h1 yang memiliki atribut class=judul.

### 5.4.1 Universal Selector

Universal selector hanya ada 1 di dalam CSS, yaitu tanda bintang “\*”. Selector ini bertujuan untuk „mencari“ semua tag yang ada.

Contoh Universal Selector CSS:

```
* {
  color: blue;
  background-color: white;}
}
```

Kode CSS diatas bermaksud untuk membuat seluruh tag HTML berwarna biru, dan background berwarna putih.

### 5.4.2 Element Type Selector

Element Type Selector atau Tag Selector adalah istilah untuk selector yang nilainya merupakan tag HTML itu sendiri. Setiap tag HTML bisa digunakan sebagai selector, dan seluruh tag tersebut akan ditangkap oleh selector ini.

Contoh Element Type Selector CSS:

```

h1 {
    text-decoration: underline;
}
p {
    font-size:14px;
}

```

Contoh kode CSS diatas akan membuat semua tag <h1> akan bergaris bawah, dan seluruh tag <p> akan berukuran 14 pixel.

Efek dari element type selector adalah dari awal tag, sampai akhir tag. Jika didalam tag <p> terdapat tag <i>, maka tag tersebut juga akan berukuran 14 pixel, sampai ditemui tag penutup </p>.

### 5.4.3 Class Selector

Class Selector merupakan salah satu selector yang paling umum dan paling sering digunakan. Class Selector akan „mencari“ seluruh tag yang memiliki atribut class dengan nilai yang sesuai. Untuk penggunaan Class Selector, kita harus memiliki tag HTML yang mempunyai atribut class. Contohnya:

```

<p class="paragraf_pertama"> Ini adalah sebuah paragraf pertama</p>
<h1 class="judul">Judul Artikel</h1>
<h2 class="judul penting berwarna">Sub Judul Artikel</h2>

```

Perhatikan bahwa untuk semua tag diatas, kita menambahkan atribut class dengan nilainya adalah nama dari kelas itu sendiri. Sebuah nama class dapat dimiliki oleh lebih dari 1 tag, dan dalam sebuah tag dapat memiliki lebih dari 1 class.

Contohnya dalam baris terakhir pada contoh diatas, tag h2 memiliki atribut class="judul penting berwarna". Tag ini terdiri dari 3 class, yaitu judul, penting, dan class berwarna. Sedangkan untuk kode CSS Class Selector adalah sebagai berikut:

```

.paragraf_pertama {
    color: red;
}

.judul {
    font-size:20px;
}

.penting {
    color:red;
    font-size: 1em;
}

```

Untuk menggunakan class selector, di dalam CSS kita menggunakan tanda titik sebelum nama dari class.

Untuk contoh kita, seluruh class yang memiliki nilai "paragraf\_pertama", warna text akan menjadi merah. Dan seluruh class judul akan memiliki font 20 pixel.

#### 5.4.4 ID Selector

ID Selector bersama-sama dengan class selector merupakan selector paling umum dan juga sering dipakai (walau tidak sesering class selector). Penggunaan ID selector hampir sama dengan class selector, dengan perbedaan jika pada Class Selector kita menggunakan atribut class untuk tag HTML, untuk ID selector, kita menggunakan atribut id.

Contoh penggunaan atribut id pada tag HTML

```
<p id="paragraf_pembuka"> Ini adalah sebuah paragraf pembuka</p>
<h1 id="judul_utama">Judul Artikel</h1>
<h2 id="sub_judul">Sub Judul Artikel</h2>
```

Atribut id selain untuk selector CSS, juga berperan sebagai kode unik untuk masing-masing tag (terutama dipakai untuk kode JavaScript). Karena hal tersebut, id yang digunakan harus unik dan tidak boleh sama. Dengan kata lain, id hanya bisa digunakan satu kali dalam sebuah halaman web dan tidak boleh sama.

Contoh penggunaan id selector kode CSS Class Selector adalah sebagai berikut:

```
#paragraf_pembuka {
    color: red;
}

#judul_utama {
    font-size:20px
}
```

Di dalam kode CSS, kita menggunakan tanda pagar “#” sebagai penanda bahwa kita mencari tag yang memiliki id tersebut.

#### 5.4.5 Attribute Selector

Selector ini sedikit lebih advanced dibandingkan dengan selector-selector sebelumnya. Atribut Selector ini digunakan untuk mencari seluruh tag yang memiliki atribut yang dituliskan.

Contoh penggunaan Attribute Selector kode CSS adalah sebagai berikut:

```
[href] {
    font-size:20px ;
}

[type="submit"] {
    width:30px;
}
```

Seperti yang dapat dilihat dari contoh diatas, setiap atribut selector harus berada diantara tanda kurung siku “[” dan “]”. [href] akan cocok dengan seluruh tag yang

memiliki atribut href, apapun nilai dari href (href biasanya terdapat pada tag <a>). Untuk contoh [type="submit"] akan cocok dengan tag yang memiliki atribut typedengan nilai submit, yang dalam hal ini adalah tombol submitdalam form. Walaupun memiliki kemampuan mencari tag yang sangat spesifik, namun atribut selector ini tidak terlalu sering digunakan.

## LATIHAN

### 1. Penggunaan Inline Style Sheet

Simpan dengan nama **latihan24.html**

```
1 <html>
2 <head>
3   <title>CSS untuk memanipulasi font</title>
4 </head>
5 <body>
6   <h3 style="font-family:'tahoma', Courier ">Sejarah AMIKEM</h3>
7   <p style="font-family:Verdana, Arial, Helvetica, sans-serif">
8     Sekolah Tinggi Manajemen Informatika dan Komputer AMIKEM Yogyakarta,
9     selanjutnya disebut STMIK AMIKEM YOGYAKARTA) merupakan salah satu
10    perguruan tinggi swasta yang berkedudukan di Provinsi DIY
11    Kabupaten Sleman di bawah naungan Yayasan AMIKEM Yogyakarta.
12  </p>
13
14  <p style="font-size:16px">
15    AMIKEM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik
16    Informatika. Program studi ini masing-masing dikelola oleh seorang
17    Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen,
18    dan Staff Administrasi.
19
20
21  <p style="font-style:italic">
22    Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
23    tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan
24    Program-Program Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer
25    STMIK AMIKEM Yogyakarta di DIY (Perubahan bentuk dari AMIKEM) yang diselenggarakan
26    oleh Yayasan AMIKEM Yogyakarta di DIY.
27  </p>
28 </body>
29
```

**Gambar 5.2** Contoh Skrip CSS Inline Style Sheet

## 2. Penggunaan Embedded Style Sheet. Simpan dengan nama latihan25.html

```
1 <html>
2 <head>
3 <title>CSS untuk memanipulasi font</title>
4 <style type="text/css">
5 p{font-family:Verdana, Arial, Helvetica, sans-serif}
6 td{font-weight:bold font-family: calibri}
7 a{font-size:50px}
8 </head>
9 <body>
10 <h2>Sejarah Amikom</h2>
11 <p>Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.
12 selanjutnya disebut STMIK AMIKOM YOGYAKARTA) merupakan salah satu
13 perguruan tinggi swasta yang berkedudukan di Provinsi DIY
14 Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta.</p>
15 <table>
16 <tr>
17 <td>
18 AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik
19 Informatika. Program studi ini masing-masing dikelola oleh seorang
20 Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen,
21 dan Staff Administrasi.
22 </td>
23 </tr>
24 <tr>
25 <td>
26 Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
27 tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan
28 Program-Program Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer
29 STMIK) AMIKOM Yogyakarta di DIY (Perubahan bentuk dari AMIKOM) yang diselenggarakan
30 oleh Yayasan AMIKOM Yogyakarta di DIY.
31 </td>
32 </tr>
33 </table>
34 <br>
35 Untuk mencoba menggunakan aplikasi Online Test, silakan klik link di bawah ini :<br>
36 <a href="http://www.bernadtagger.com">Online Test</a>
37 </body>
38 </html>
```

Gambar 5.3 Contoh Skrip CSS Embedded Style Sheet

3. Penggunaan Linked Style Sheet. Simpan File Pertama dengan nama **style definition.css**:

```
1  /* Dokumen CSS */
2
3  tulisan_miring{
4      font-family:Arial, Helvetica, sans-serif;
5      font-size:12px;
6      font-style:italic;
7  }
8  .cetak_tebal {
9      font-family:Verdana, Arial, Helvetica, sans-serif;
10     font-size:15px;
11     font-weight:bold;
12     color:skyblue;
13 }
14
15 #14pt {
16     font-size:14pt;
17 }
18
19 p {
20     font-family:Verdana, Arial, Helvetica, sans-serif
21 }
22
23 td {
24     font-weight:bold
25 }
26
27 a {
28     font-size:24px
29 }
```

**Gambar 5.4** Contoh Skrip Linked Style Sheet

Kemudian buat file kedua dengan nama **latihan26.html**

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CSS untuk memanipulasi font</title>
5     <link href="Style Definition.css" rel="stylesheet" type="text/css">
6   </head>
7   <body>
8     <h2>Jarak Antar</h2>
9     <table>
10      <tr>
11        <td>
12          <div class="margin">
13            Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.
14            selanjutnya adalah STMIK ARMAN YOGYAKARTA merupakan salah satu
15            perguruan tinggi swasta yang berkedudukan di Provinsi DIY
16            Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta.
17          </div>
18        </td>
19      </tr>
20      <tr>
21        <td>
22          AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik
23          Informatika. Program studi ini masing-masing dikelola oleh seorang
24          Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Dosen Tetap,
25          dan Staff Administrasi. <br>
26        </td>
27      </tr>
28      <tr>
29        <td>
30          Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
31          tanggal 24 April 2002, Nomor 78/D/0/2002 tentang Pembinaan Liris Penyelenggaraan
32          Program-Program Studi dan Pendidikan Sekolah Tinggi Manajemen Informatika dan Komputer
33          AMIKOM YOGYAKARTA di DIY (Perubahan bentuk dari AMIKOM yang diselenggarakan
34          oleh Yayasan AMIKOM Yogyakarta di DIY. <br>
35        </td>
36      </tr>
37    </table>
38   </body>
39 </html>
```

**Gambar 5.5** Contoh Skrip Linked Style Sheet

4. Pemberian property multi kepada selector. Simpan dengan nama **latihan27.html**

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CSS untuk memanipulasi font</title>
5     <style type="text/css">
6       @font-family Arial, Helvetica, sans-serif;font-size:12px; font-style:italic;
7       @font-family Verdana, Arial, Helvetica, sans-serif;font-size:14px;font-weight:bold;
8     </style>
9   </head>
10  <h2>Jarak Antar</h2>
11  <table>
12    <tr>
13      <td>
14        Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta selanjutnya adalah STMIK ARMAN YOGYAKARTA merupakan salah satu
15        perguruan tinggi swasta yang berkedudukan di Provinsi DIY Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta. <br>
16      </td>
17      <td>
18        AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik Informatika. Program studi ini masing-masing dikelola oleh seorang
19        Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Dosen Tetap, dan Staff Administrasi. <br>
20      </td>
21      <td>
22        Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
23        tanggal 24 April 2002, Nomor 78/D/0/2002 tentang Pembinaan Liris Penyelenggaraan
24        Program-Program Studi dan Pendidikan Sekolah Tinggi Manajemen Informatika dan Komputer
25        AMIKOM YOGYAKARTA di DIY (Perubahan bentuk dari AMIKOM yang diselenggarakan
26        oleh Yayasan AMIKOM Yogyakarta di DIY. <br>
27      </td>
28    </tr>
29  </table>
30  <table>
31    <tr>
32      <td>
33        Untuk kembali ke berita sebelumnya
34        <a href="2. CSS Embedded Style Sheet .html">
35          silahkan klik disini << berarti ini telah mendapat style CSS pada "a" dan multiple dari "td"
36        </a>
37      </td>
38    </tr>
39  </table>
40 </html>
```

**Gambar 5.6** Contoh Skrip dalam pemberian property multi kepada selector

5. Manupulasi font dengan menggunakan selector bebas, class, dan ID. Simpan dengan nama **latihan28.html**

```
1 <html>
2   <head>
3     <title>
4       CSS untuk memanipulasi font
5     </title>
6
7     <style type="text/css">
8       tulisan_miring {
9         font-family:Arial, Helvetica, sans-serif;
10        font-size:12px;
11        font-style:italic;
12      }
13      .cetak_tebal {
14        font-family:Verdana, Arial, Helvetica, sans-serif;
15        font-size:15px;
16        font-weight:bold;
17        color:skyblue;
18      }
19      #14pt {
20        font-size:14pt;
21      }
22    </style>
23
24  </head>
25  <body>
26    <h3>Sejarah Amikom </h3>
27    <table>
28      <tr>
29        <td>
30          <tulisan_miring>
31            Sekolah Tinggi Manajemen Informatika dan Komputer AMIRIN Yogyakarta,
32            selanjutnya disebut STMIK AMIRIN YOGYAKARTA) merupakan salah satu
33            perguruan tinggi swasta yang berkedudukan di Provinsi DIY
34            Kabupaten Sleman di bawah naungan Yayasan AMIRIN Yogyakarta.
35          </tulisan_miring><br>&nbsp;
36
37        <td class="cetak_tebal">
38          AMIRIN Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik
39          Informatika. Program studi ini masing-masing dikelola oleh seorang
40          Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perekat Dosen,
41          dan Staff Administrasi. <br>
42
43        <td id="14pt">
44          Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
45          tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan
46          Program-Program Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer
47          STMIK AMIRIN Yogyakarta di DIY (Pembaharuan bentuk dari AMIRIN) yang diselenggarakan
48          oleh Yayasan AMIRIN Yogyakarta di DIY.
49
50      </td>
51    </tr>
52  </table>
53 </body>
54 </html>
```

**Gambar 5.7** Contoh Hasil Skrip penggunaan multi kepada selector bebas, class dan ID

# 2

## **SISTEM BASIS DATA**

## MODUL 1 DASAR MySQL

### A. TUJUAN

1. Mahasiswa dapat memahami dasar-dasar MySQL
2. Mahasiswa dapat membuat database dan tabel dengan menerapkan data definition language (DDL)
3. Mahasiswa dapat memahami penggunaan perintah-perintah untuk mendefinisikan atribut-atribut database, tabel, field, maupun batasan-batasan terhadap suatu atribut dan hubungan antar tabel

### B. TEORI

**Basis Data** : Himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.

#### Bahasa Basis Data

DBMS merupakan perantara antara user dengan database. Cara komunikasi diatur dalam suatu bahasa khusus yang ditetapkan oleh DBMS. Misalnya SQL,dBase,QUEL, dsb.

Ada beberapa bentuk bahasa sql, namun yang biasa digunakan adalah 2 bentuk bahasa sql yaitu DDL dan DML

Data Definition Language (DDL) >> digunakan dalam membuat tabel baru, indeks, mengubah tabel, menentukan struktur tabel, dsb

Data Manipulation Language (DML) >> digunakan dalam memanipulasi dan pengambilan data pada database. Manipulasi data dapat mencakup :

- Pemanggilan data yang tersimpan dalam database (query)
- Penyisipan/penambahan data baru ke database
- Penghapusan data dari database
- Pengubahan data pada database

#### Dasar-Dasar MySQL

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (row atau record) dan kolom (column atau field). Sedangkan dalam sebuah database dapat terdiri dari beberapa table. Beberapa tipe data dalam MySQL yang sering dipakai:

Tipe data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LOB	Teks dengan panjang maksimum 4294967295 karakter

Data Definition Language (DDL) merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut-atribut database, tabel, atribut kolom (field), maupun batasan-batasan terhadap suatu atribut dan relasi/ hubungan antar tabel. Yang termasuk dalam kelompok perintah DDL adalah :

- Create
- Alter
- Drop

**CREATE** merupakan perintah DDL yang digunakan untuk membuat database maupun tabel. Nama database maupun tabel tidak boleh mengandung spasi (space). Nama database tidak boleh sama antar database.

**ALTER** merupakan perintah DDL yang digunakan untuk mengubah nama/struktur tabel

**DROP** merupakan perintah DDL yang digunakan untuk menghapus database ataupun tabel

a. Database

Perintah untuk membuat database sebagai berikut :

- **CREATE DATABASE namadatabase;**

Contoh :

**CREATE DATABASE db\_polinema;**

Untuk menampilkan daftar nama database yang terdapat dalam database server pada MySQL menggunakan perintah :

- **SHOW DATABASES;**

Sebelum membuat suatu tabel yang akan digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan salah satu database sebagai database aktif yang akan digunakan untuk menyimpan beberapa tabel yang akan dibuat. Untuk memilih/mengaktifkan salah satu database menggunakan perintah berikut :

- **USE namadatabase;**

Sebagai contoh, database db\_polinema yang akan digunakan / diaktifkan maka perintahnya sebagai berikut :

**USE db\_polinema;**

Perintah yang berfungsi untuk menghapus database maupun tabel adalah perintah DROP. Untuk menggunakan perintah tersebut, berikut penulisan perintahnya :

- **DROP DATABASE namadatabase;**

Misalnya database db\_polinema yang akan dihapus maka perintahnya sebagai berikut :

**DROP DATABASE db\_polinema;**

b. Tabel

- Membuat Tabel

Nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel, ada beberapa yang harus dideklarasikan dalam pembuatannya yaitu antara lain meliputi : nama tabel, nama kolom (field), tipe data dari field dan panjang data. Perintah yang digunakan untuk membuat tabel secara umum adalah sebagai berikut :

- **CREATE TABLE namatabel (field1 typedata1, field2 typedata2);**

Contoh berikut ini adalah syntax untuk membuat tabel mahasiswa :

```
CREATE TABLE mahasiswa (nim CHAR (20),nama_mhs CHAR(50),umur INT, PRIMARY KEY(nim));
```

- Menampilkan tabel

Untuk menampilkan daftar nama tabel yang terdapat dalam database yang sedang aktif/digunakan menggunakan perintah :

- **SHOW TABLES;**

- Menampilkan deskripsi atribut tabel

Untuk menampilkan deskripsi atribut-atribut yang terdapat pada suatu tabel dengan menggunakan perintah :

- **DESC namatabel;**

Contoh :

```
DESC mahasiswa;
```

- Menghapus Tabel

Untuk menghapus Tabel perintahnya sama dengan untuk menghapus database yaitu dengan menggunakan perintah DROP. Perintah yang digunakan adalah :

- **DROP TABLE namatabel;**

Misalnya yang akan dihapus adalah tabel mahasiswa :

```
DROP TABLE mahasiswa;
```

- Mendefinisikan PRIMARY KEY pada Tabel

Suatu keharusan dalam suatu tabel adalah harus memiliki satu kolom yang dijadikan sebagai perwakilan dari tabel tersebut. Pembuatan perwakilan tabel ini berfungsi untuk melakukan hubungan / relasional dengan tabel lain. Bentuk perwakilan ini dalam database disebut sebagai PRIMARY KEY yang aturan pembuatannya adalah sebagai berikut :

- Satu tabel bisa memiliki 2 primary key
- Nama kolom kunci tidak digunakan pada kolom lain satu tabel
- Nama kolom kunci tidak boleh sama dengan kolom kunci yang ada pada tabel lain
- Bentuk kolom kunci harus diset NOT NULL

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini perintah yang digunakan :

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), umur INT, PRIMARY KEY (nim));
```

Atau

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL PRIMARY KEY, nama_mhs  
CHAR (50), umur INT);
```

Atau

```
ALTER TABLE mahasiswa ADD CONSTRAINT namaconstraint PRIMARY  
KEY(namakolom);
```

- Menghapus PRIMARY KEY pada tabel

Cara 1: Jika primary key dibuat menggunakan alter table:

```
ALTER TABLE namatabel DROP CONSRTRAIINT namaconstraint;
```

Cara 2: Jika primary key dibuat melalui create table:

```
ALTER TABLE namatable DROP PRIMARY KEY;
```

- Menambah kolom baru pada tabel

Untuk menambah kolom baru pada suatu tabel bisa menggunakan perintah sebagai berikut :

```
ALTER TABLE namatabel ADD fieldbaru typedata(lebar);
```

Namatabel merupakan nama tabel yang akan ditambahkan kolomnya. Filedbaru merupakan nama kolom yang akan ditambahkan, typedata(lebar) merupakan type data dan lebar data yang akan ditambahkan. Contohnya menambahkan kolom telepon pada tabel mahasiswa setelah kolom umur:

```
ALTER TABLE mahasiswa ADD COLUMN telepon CHAR (15) AFTER umur;
```

- Mengubah Tipe Data atau Lebar Kolom pada Tabel

Perintah yang digunakan adalah :

```
ALTER TABLE namatabel MODIFY COLUMN field type(lebar);
```

Contoh :

```
ALTER TABLE mahasiswa MODIFY COLUMN telepon(12);
```

- Mengubah Nama Kolom(Field)

Perintah yang digunakan adalah :

```
ALTER TABLE namatabel CHANGE COLUMN namakolomlama namakolombaru  
typedatabaru(lebarbaru);
```

Contoh :

```
ALTER TABLE mahasiswa CHANGE COLUMN telepon phone CHAR(25);
```

- Menghapus Kolom pada Tabel

Perintah yang digunakan adalah :

```
ALTER TABLE namatabel DROP COLUMN namakolom;
```

Contoh :

```
ALTER TABLE mahasiswa DROP COLUMN phone;
```

### C. LANGKAH PRAKTIKUM

Langkah	Keterangan																
1.	Buka prompt jalankan perintah berikut ini : C:\>Program Files\xampp\mysql\bin>mysql -u root -p (enter)																
	<pre> Yushintias-MacBook-Pro:~ YushintiaPramitarini\$ mysql -u root -p Enter password: Welcome to the MySQL monitor.  Commands end with ; or \g. Your MySQL connection id is 17 Server version: 5.7.15 Homebrew  Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.  Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  mysql&gt;                     </pre>																
2.	Buatlah sebuah database dengan nama db_polinema																
	<pre> mysql&gt; create database db_polinema; Query OK, 1 row affected (0.01 sec)  mysql&gt; show databases; +-----+   Database   +-----+   information_schema     db_polinema     mysql     performance_schema     sys   +-----+ 5 rows in set (0.00 sec)                     </pre>																
3.	Buatlah beberapa tabel dalam database tersebut sesuai dengan kriteria berikut : <p>a. Tabel <b>Mahasiswa</b></p> <table border="1" data-bbox="395 1666 1353 2018"> <thead> <tr> <th data-bbox="395 1666 638 1711">Field</th> <th data-bbox="638 1666 1353 1711">Type Data</th> </tr> </thead> <tbody> <tr> <td data-bbox="395 1711 638 1756">Nim</td> <td data-bbox="638 1711 1353 1756">Int (8) Primary Key</td> </tr> <tr> <td data-bbox="395 1756 638 1800">nama_mhs</td> <td data-bbox="638 1756 1353 1800">Char (50)</td> </tr> <tr> <td data-bbox="395 1800 638 1845">jenis_kelamin</td> <td data-bbox="638 1800 1353 1845">Enum ('L','P') DEFAULT 'L'</td> </tr> <tr> <td data-bbox="395 1845 638 1890">Alamat</td> <td data-bbox="638 1845 1353 1890">Varchar (50)</td> </tr> <tr> <td data-bbox="395 1890 638 1935">Kota</td> <td data-bbox="638 1890 1353 1935">Varchar (20) DEFAULT 'MALANG'</td> </tr> <tr> <td data-bbox="395 1935 638 1980">asal_sma</td> <td data-bbox="638 1935 1353 1980">Char (30)</td> </tr> <tr> <td data-bbox="395 1980 638 2018">no_hp</td> <td data-bbox="638 1980 1353 2018">Varchar (12)</td> </tr> </tbody> </table>	Field	Type Data	Nim	Int (8) Primary Key	nama_mhs	Char (50)	jenis_kelamin	Enum ('L','P') DEFAULT 'L'	Alamat	Varchar (50)	Kota	Varchar (20) DEFAULT 'MALANG'	asal_sma	Char (30)	no_hp	Varchar (12)
Field	Type Data																
Nim	Int (8) Primary Key																
nama_mhs	Char (50)																
jenis_kelamin	Enum ('L','P') DEFAULT 'L'																
Alamat	Varchar (50)																
Kota	Varchar (20) DEFAULT 'MALANG'																
asal_sma	Char (30)																
no_hp	Varchar (12)																

	<table border="1"> <tr> <td>Umur</td> <td>INT</td> </tr> <tr> <td>kode_prodi</td> <td>Char (6) foreign key fk0 (kode_prodi) references prodi (kode_prodi)</td> </tr> </table>	Umur	INT	kode_prodi	Char (6) foreign key fk0 (kode_prodi) references prodi (kode_prodi)						
Umur	INT										
kode_prodi	Char (6) foreign key fk0 (kode_prodi) references prodi (kode_prodi)										
	<pre>mysql&gt; create table mahasiswa (nim int(8) primary key, nama_mhs char(50), jenis_kelamin enum('L','P') default 'L', alamat varchar(50), kota varchar(20), asal_sma char(30), no_hp varchar(12), umur integer, kode_prodi char(6) foreign key fk0 (kode_prodi) references prodi(kode_prodi)); Query OK, 0 rows affected (0.01 sec)</pre>										
4.	<p>b. Tabel Prodi</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Type Data</th> </tr> </thead> <tbody> <tr> <td>kode_prodi</td> <td>Char (6) Primary Key</td> </tr> <tr> <td>nama_prodi</td> <td>Char (30)</td> </tr> </tbody> </table>	Field	Type Data	kode_prodi	Char (6) Primary Key	nama_prodi	Char (30)				
Field	Type Data										
kode_prodi	Char (6) Primary Key										
nama_prodi	Char (30)										
	<pre>mysql&gt; create table prodi (kode_prodi char(6) primary key, nama_prodi varchar(30)); Query OK, 0 rows affected (0.02 sec)</pre>										
5.	<p>c. Tabel mata_kuliah</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Type Data</th> </tr> </thead> <tbody> <tr> <td>mk_id</td> <td>Char (10) Primary Key</td> </tr> <tr> <td>nama_mk</td> <td>Char (50)</td> </tr> <tr> <td>jumlah_jam</td> <td>Float (4,2)</td> </tr> <tr> <td>Sks</td> <td>Integer</td> </tr> </tbody> </table>	Field	Type Data	mk_id	Char (10) Primary Key	nama_mk	Char (50)	jumlah_jam	Float (4,2)	Sks	Integer
Field	Type Data										
mk_id	Char (10) Primary Key										
nama_mk	Char (50)										
jumlah_jam	Float (4,2)										
Sks	Integer										
	<pre>mysql&gt; create table mata_kuliah (mk_id char(10) primary key, nama_mk char(50), jumlah_jam float(4,2), sks integer); Query OK, 0 rows affected (0.02 sec)</pre>										
6.	<p>d. Tabel ruang</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Type Data</th> </tr> </thead> <tbody> <tr> <td>ruang_id</td> <td>Char (3) Primary Key</td> </tr> <tr> <td>nama_ruang</td> <td>Char (20)</td> </tr> <tr> <td>Kapasitas</td> <td>Integer</td> </tr> </tbody> </table>	Field	Type Data	ruang_id	Char (3) Primary Key	nama_ruang	Char (20)	Kapasitas	Integer		
Field	Type Data										
ruang_id	Char (3) Primary Key										
nama_ruang	Char (20)										
Kapasitas	Integer										
	<pre>mysql&gt; create table ruang (ruang_id char(3) primary key, nama_ruang char(20), kapasitas integer); Query OK, 0 rows affected (0.02 sec)</pre>										
7.	<p>e. Tabel dosen</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Type Data</th> </tr> </thead> <tbody> <tr> <td>Nidn</td> <td>integer (20) Primary Key</td> </tr> <tr> <td>nama_dosen</td> <td>Char (50)</td> </tr> <tr> <td>Status</td> <td>Enum ('PNS','KONTRAK') Default 'PNS'</td> </tr> </tbody> </table>	Field	Type Data	Nidn	integer (20) Primary Key	nama_dosen	Char (50)	Status	Enum ('PNS','KONTRAK') Default 'PNS'		
Field	Type Data										
Nidn	integer (20) Primary Key										
nama_dosen	Char (50)										
Status	Enum ('PNS','KONTRAK') Default 'PNS'										

	<table border="1"> <tr> <td>jenis_kelamin</td> <td>Enum ('L','P') Default 'L'</td> </tr> <tr> <td>no_hp</td> <td>Varchar (15)</td> </tr> </table>	jenis_kelamin	Enum ('L','P') Default 'L'	no_hp	Varchar (15)
jenis_kelamin	Enum ('L','P') Default 'L'				
no_hp	Varchar (15)				
	<pre>mysql&gt; create table dosen (nidn integer(20) primary key, nama_dosen char(50), status enum ('PNS','KONTRAK') default 'PNS', jenis_kelamin enum ('L','P') default 'L', no_hp varchar(15)); Query OK, 0 rows affected (0.02 sec)</pre>				
8.	Tambahkan sebuah kolom agama (varchar(10)) pada tabel mahasiswa sebagai kolom terakhir				
9.	Tambahkan kolom alamat(varchar(50)) pada tabel dosen sebagai kolom terakhir				
10.	Tampilkan struktur(metadata) tabel mahasiswa				

#### D. TUGAS

1. Buatlah basis data Matakuliah\_NIM (Matakuliah\_10234567)
2. Buatlah 3 tabel sesuai dengan format berikut ini:

**Tabel mata\_kuliah {kode\_mk, nama\_mk}**

**Tabel dosen {kd\_dosen,nama\_dosen}**

**Tabel jadwal {kode\_mk,hari,jam\_ke,kelas,ruang}**

3. tambahkan kolom alamat\_dosen pada tabel dosen di kolom terakhir
4. tambahkan kolom sks pada tabel mata\_kuliah
5. tambahkan kolom kd\_dosen dari tabel dosen kedalam tabel\_mata\_kuliah serta berikanlah kunci foreign key
6. tampilkan semua data yang ada pada tiap tabel

## MODUL 2 DML

### A. TUJUAN

1. Mahasiswa dapat Memahami penggunaan SQL statement INSERT.
2. Mahasiswa dapat Memahami penggunaan SQL statement UPDATE.
3. Mahasiswa dapat Memahami penggunaan SQL statement DELETE.

### B. TEORI

DML merupakan istilah untuk beberapa sintaksis (syntax) dari SQL yang digunakan untuk melakukan perubahan pada data (isi tabel-tabel) dalam suatu database. DML terdiri dari 3 klausa utama yaitu:

1. **INSERT** : Menambah baris baru pada sebuah tabel
2. **UPDATE** : Mengubah nilai suatu baris pada sebuah tabel.
3. **DELETE** : Menghapus suatu baris dari sebuah tabel.

#### Operasi INSERT

Klausa pembentuk:

1. **INSERT**
2. **INTO**
3. **VALUES**

Format:

1. **INSERT INTO** nama\_tabel (kolom1, kolom2, ...dst.) **VALUES** (nilai\_kolom1, nilai\_kolom2, ...dst.);
2. **INSERT INTO** nama\_tabel **VALUES** (nilai\_kolom1, nilai\_kolom2, ...dst.);
3. [Salah satu dari kedua format sebelumnya], (nilai\_kolom\_kolom\_baris1), (nilai\_kolom\_kolom\_baris2), ...dst.

Ada dua pilihan tindakan yang dapat dilakukan jika ada satu atau lebih constraint yang dilanggar akibat operasi insert, yaitu :

1. Menolak (reject) operasi insertion. Biasanya DBMS memberikan penjelasan mengapa proses insertion ditolak.
2. Berusaha memperbaiki alasan penolakan proses insertion. Dimana insertion akan diterima jika user melakukan perubahan nilai-nilai attribute sehingga insertion diterima.

#### Operasi DELETE

Operasi delete bertujuan untuk menghapus satu atau beberapa tuple di dalam suatu relasi R. Operasi ini hanya dapat melanggar **referential integrity**, jika tuple yang dihapus diacu oleh kunci-kunci tamu dari tuple yang lain dalam basis data.

Klausa pembentuk:

1. **DELETE**
2. **FROM**
3. **WHERE** [opsional]

Format:

1. **DELETE FROM** nama\_tabel **WHERE** nama\_kolom\_patokan [**operator\_perbandingan**] nilai\_patokan;
2. **DELETE \* FROM** nama\_tabel; atau **DELETE FROM** nama\_tabel;

\*Operator perbandingan/comparison operator dapat berupa: =, <, >, <=, >=, <>

Ada empat pilihan tindakan yang dapat dilakukan jika suatu deletion melanggar constraint yang telah ditentukan, yaitu :

1. Menolak(reject) proses deletion.
2. Berusaha untuk melakukan "cascade deletion", yaitu dengan menghapus sejumlah tuple yang mengacu pada tuple yang akan dihapus.
3. Melakukan modifikasi nilai attribute yang mengacu pada tuple yang dihapus, yaitu setiap nilai diset NULL atau diganti dengan nilai dari tuple lain yang valid sebagai acuan baru. Akan tetapi, bila attribute yang mengacu yang menyebabkan pelanggaran adalah bagian dari primary key, maka ia tidak dapat diset NULL (karena melanggar entity integrity).
4. Kombinasi 2 dan 3.

## Operasi UPDATE

Operasi update digunakan untuk merubah nilai-nilai satu atau lebih attribute dalam satu atau lebih tuple dalam sejumlah relasi R.

Klausa pembentuk:

1. **UPDATE**
2. **SET**
3. **WHERE** [opsional]

Format:

1. **UPDATE** nama\_tabel **SET** nama\_kolom = nilai\_baru **WHERE** nama\_kolom\_patokan [**operator\_perbandingan**] nilai\_patokan;

2. **UPDATE** nama\_tabel **SET** nama\_kolom1 = nilai\_baru1, nama\_kolom2 = nilai\_baru2, ...dst. **WHERE** nama\_kolom\_patokan [**operator\_perbandingan**] nilai\_patokan;

\*Operator perbandingan/comparison operator dapat berupa: =, <, >, <=, >=, <>

Operasi UPDATE bisa dilakukan pada tiga jenis attribut, dengan permasalahan yang berbeda sebagaimana berikut ini :

1. Modifikasi nilai suatu foreign key, maka DBMS harus melakukan pengecekan bahwa nilai-nilai baru yang diberikan mengacu pada tuple yang ada dalam relasi-relasi yang dijadikan acuan.
2. modifikasi nilai suatu primary key serupa dengan proses deletion satu tuple dan inserting yang lain pada tempat yang sama. Akibatnya, pilihan- pilihan seperti yang dilakukan pada operasi INSERT dan DELETE dapat dipakai agar modifikasi tidak melanggar constraint.
3. modifikasi suatu attribute yang bukan primary key atau bukan foreign key biasanya tidak akan menimbulkan masalah. DBMS hanya perlu untuk mengecek apakah nilai-nilai baru yang diberikan mempunyai tipe data dan domain yang valid.

### C. LANGKAH PRAKTIKUM

Langkah	Keterangan
1	Buatlah Sebuah basis data
2	BUatlah tabel Departemen dan Karyawan
	<pre> -- Create schema ----- DROP DATABASE IF EXISTS software_house; CREATE DATABASE software_house; USE software_house;  ----- -- Table departemen ----- CREATE TABLE departemen (   kode VARCHAR(3) NOT NULL,   nama VARCHAR(45) UNIQUE NOT NULL,   nik_manajer VARCHAR(10) NULL,   PRIMARY KEY (kode) ); </pre>

	<pre> ----- -- Table karyawan ----- CREATE TABLE karyawan (   nik VARCHAR(10) NOT NULL,   nama VARCHAR(100) NOT NULL,   tanggal_lahir DATE NULL,   jenis_kelamin ENUM('PRIA', 'WANITA') NULL,   alamat VARCHAR(255) NULL,   nomor_telepon VARCHAR(20) NOT NULL,   email VARCHAR(100) UNIQUE NOT NULL,   gaji DECIMAL(10,2) UNIQUE NULL,   departemen_kode VARCHAR(3) NOT NULL,   PRIMARY KEY (nik),   FOREIGN KEY fk_karyawan_departemen_kode (departemen_kode)     REFERENCES departemen (kode)     ON DELETE CASCADE     ON UPDATE CASCADE ); </pre>
3	Cek database Anda dengan perintah 'SHOW TABLES' untuk memastikan bahwa semua tabel sudah dibuat.
4	Buat 1 tabel backup untuk tabel departemen bernama <b>departemen_backup</b> . Tabel ini akan bermanfaat ketika bagian INSERT dan DELETE kita kerjakan.
	<pre> CREATE TABLE departemen_backup (   kode VARCHAR(3) PRIMARY KEY,   nama VARCHAR (45),   nik_manajer VARCHAR(10) ); </pre>
5	Cek strukturnya.
	<pre> [mysql&gt; SHOW COLUMNS FROM departemen_backup; +-----+-----+-----+-----+-----+-----+   Field        Type            Null   Key   Default   Extra   +-----+-----+-----+-----+-----+-----+   kode         varchar(3)      NO     PRI   NULL               nama         varchar(45)     YES          NULL               nik_manajer   varchar(10)     YES          NULL             +-----+-----+-----+-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
6	Setelah selesai membuat database diatas, lanjutkan ke <b>Praktikum – Bagian 2</b> .

**Praktikum – Bagian 2: Percobaan Statement INSERT**

Langkah	Keterangan
1	Untuk menambahkan data (mengisi) suatu tabel, digunakan statement (pernyataan) <b>INSERT</b> . Eksekusi SQL berikut untuk menambahkan 1 baris (record) baru pada tabel <b>departemen</b> .
	<pre>INSERT INTO departemen (kode, nama, nik_manajer) VALUES ('MKK', 'Manajemen dan Kontrol Kualitas', NULL);</pre>
2	Statement SQL tersebut menambahkan 1 baris baru ke tabel departemen pada kolom yang dinyatakan di dalam tanda kurung () pertama. Untuk melihat hasil SQL yang kita eksekusi tersebut, gunakan statement <b>SELECT</b> seperti berikut.
	<pre>SELECT * FROM departemen;</pre>
3	Pembahasan lebih lengkap mengenai SELECT dijadwalkan untuk disampaikan pada pertemuan berikutnya, namun secara umum, statement SELECT digunakan untuk menyajikan record-record yang ada pada suatu tabel. Karakter * akan menampilkan isi dari semua kolom yang ada pada tabel. Statement SELECT tersebut menghasilkan:
	<pre>[mysql&gt; SELECT * FROM departemen; +-----+-----+-----+   kode   nama                  nik_manajer   +-----+-----+-----+   MKK    Manajemen dan Kontrol Kualitas   NULL          +-----+-----+-----+ 1 row in set (0.00 sec)</pre>
4	Apabila data di-insert-kan pada semua kolom tabel, maka kita dapat langsung menggunakan klausa VALUES tanpa harus menuliskan nama-nama kolom dahulu.
	<pre>INSERT INTO departemen VALUES ('RND', 'Riset dan Pengembangan', NULL);</pre>
	<pre>[mysql&gt; SELECT * FROM departemen; +-----+-----+-----+   kode   nama                  nik_manajer   +-----+-----+-----+   MKK    Manajemen dan Kontrol Kualitas   NULL            RND    Riset dan Pengembangan            NULL          +-----+-----+-----+ 2 rows in set (0.00 sec)</pre>
5	Untuk menambahkan beberapa kolom sekaligus dalam 1 statement digunakan statement dengan format seperti berikut.

	<pre>INSERT INTO departemen VALUES ('DUX', 'Desain dan User Experience', NULL), ('MKT', 'Marketing', NULL), ('CSR', 'Customer Service and Relation', NULL);</pre>
6	Dan seperti berikut, jika hanya kolom tertentu saja yang akan diberi nilai.
	<pre>INSERT INTO departemen (kode, nama) VALUES ('PRD', 'Produksi'), ('HRD', 'Human Resource Development'), ('MGT', 'Top Level Management')</pre>
	<pre>[mysql&gt; SELECT * FROM departemen; +-----+-----+-----+   kode   nama                                 nik_manajer   +-----+-----+-----+   CSR    Customer Service and Relation        NULL            DUX    Desain dan User Experience           NULL            HRD    Human Resource Development           NULL            MGT    Top Level Management                 NULL            MKK    Manajemen dan Kontrol Kualitas       NULL            MKT    Marketing                             NULL            PRD    Produksi                              NULL            RND    Riset dan Pengembangan                NULL          +-----+-----+-----+ 8 rows in set (0.00 sec)</pre>
7	Statement INSERT juga dapat dieksekusi dengan menggunakan klausa SET alih-alih VALUES.
	<pre>INSERT INTO departemen SET kode = 'PKU', nama = 'Pekerjaan Umum';</pre>

```
[mysql> SELECT * FROM departemen;
+-----+-----+-----+
| kode | nama                               | nik_manajer |
+-----+-----+-----+
| CSR  | Customer Service and Relation     | NULL        |
| DUX  | Desain dan User Experience         | NULL        |
| HRD  | Human Resource Development         | NULL        |
| MGT  | Top Level Management              | NULL        |
| MKK  | Manajemen dan Kontrol Kualitas    | NULL        |
| MKT  | Marketing                          | NULL        |
| PKU  | Pekerjaan Umum                    | NULL        |
| PRD  | Produksi                          | NULL        |
| RND  | Riset dan Pengembangan             | NULL        |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

8 Pada statement INSERT juga dapat digunakan klausa SELECT. Misalnya kita ingin menyalin semua baris pada tabel departemen ke tabel **departemen\_backup**, maka kita SQL berikut dapat digunakan.

```
INSERT INTO departemen_backup
SELECT * FROM departemen;
```

```
[mysql> SELECT * FROM departemen_backup;
+-----+-----+-----+
| kode | nama                               | nik_manajer |
+-----+-----+-----+
| CSR  | Customer Service and Relation     | NULL        |
| DUX  | Desain dan User Experience         | NULL        |
| HRD  | Human Resource Development         | NULL        |
| MGT  | Top Level Management              | NULL        |
| MKK  | Manajemen dan Kontrol Kualitas    | NULL        |
| MKT  | Marketing                          | NULL        |
| PKU  | Pekerjaan Umum                    | NULL        |
| PRD  | Produksi                          | NULL        |
| RND  | Riset dan Pengembangan             | NULL        |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

9 Sekarang penambahan data pada tabel **karyawan** dapat dilakukan karena data departemen sudah siap. Eksekusi SQL berikut ini untuk menambahkan beberapa data pegawai.

	<pre>INSERT INTO karyawan VALUES ('741104013', 'Ahmad Husain', '1988-07-12', 'PRIA', 'Jl. Ikan Tombro No. 8', '0858992878', 'a.husain88@gmail.com', 100000, 'DUX'), ('741104019', 'Brian Anggriawan', '1987-12-9', 'PRIA', 'Jl. Bandung No. 2', '0857646577', 'brian.anggriawan@live.com', 89000, 'PKU');</pre>
	<pre>[mysql&gt; SELECT * FROM karyawan; +-----+-----+-----+-----+-----+   nik        nama            tanggal_lahir   jenis_kelamin   alamat             nomor_telepon   email            gaji            departemen_kode +-----+-----+-----+-----+-----+   741104013   Ahmad Husain    1988-07-12      PRIA            Jl. Ikan Tombro No. 8   0858992878     a.husain88@gmail.com   100000.00      DUX     741104019   Brian Anggriawan   1987-12-09      PRIA            Jl. Bandung No. 2   0857646577     brian.anggriawan@live.com   89000.00       PKU +-----+-----+-----+-----+-----+ 2 rows in set (0.00 sec)</pre>
10	Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke <b>Praktikum - Bagian 3</b> .

### Praktikum - Bagian 3: Percobaan Statement UPDATE

Langkah	Keterangan
1	UPDATE digunakan untuk mengubah nilai suatu baris pada sebuah tabel. Format dasar statement ini adalah sebagai berikut:
	<b>UPDATE departemen SET nik_manajer = '741104013';</b>
2	Statement tersebut mengubah nilai SEMUA baris dari tabel <b>departemen</b> pada kolom <b>nik_manajer</b> dengan nilai <b>741104013</b> . Apabila kita tampilkan isi tabel, maka sekarang semua manajer akan dikepalai oleh karyawan dengan nik tersebut.

```
[mysql> SELECT * FROM departemen;
+-----+-----+-----+
| kode | nama |          | nik_manajer |
+-----+-----+-----+
| CSR  | Customer Service and Relation | 741104013 |
| DUX  | Desain dan User Experience    | 741104013 |
| HRD  | Human Resource Development    | 741104013 |
| MGT  | Top Level Management          | 741104013 |
| MKK  | Manajemen dan Kontrol Kualitas | 741104013 |
| MKT  | Marketing                     | 741104013 |
| PKU  | Pekerjaan Umum                | 741104013 |
| PRD  | Produksi                      | 741104013 |
| RND  | Riset dan Pengembangan        | 741104013 |
+-----+-----+-----+
9 rows in set (0.01 sec)
```

3 Untuk mengubah nilai pada baris tertentu saja, kita tambahkan klausa WHERE pada statement UPDATE. Misalkan kita akan menjadikan karyawan dengan nip 7411040719 sebagai manajer pada departemen PKU, maka dapat digunakan SQL sebagai berikut:

```
UPDATE departemen
SET nik_manajer = '741104019'
WHERE kode = 'PKU';
```

```
[mysql> SELECT * FROM departemen;
+-----+-----+-----+
| kode | nama |          | nik_manajer |
+-----+-----+-----+
| CSR  | Customer Service and Relation | 741104013 |
| DUX  | Desain dan User Experience    | 741104013 |
| HRD  | Human Resource Development    | 741104013 |
| MGT  | Top Level Management          | 741104013 |
| MKK  | Manajemen dan Kontrol Kualitas | 741104013 |
| MKT  | Marketing                     | 741104013 |
| PKU  | Pekerjaan Umum                | 741104019 |
| PRD  | Produksi                      | 741104013 |
| RND  | Riset dan Pengembangan        | 741104013 |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

4 Klausa WHERE tidak selalu hanya membatasi UPDATE pada 1 baris saja, ia juga bisa memberlakukan UPDATE pada banyak baris sekaligus. Semuanya tergantung pada kondisi yang kita tentukan. Statement berikut ini akan mengosongkan **nik\_manajer** untuk semua **departemen** yang dikepalai oleh karyawan dengan **nik** 741104013.

	<pre>UPDATE departemen SET nik_manajer = NULL WHERE nik_manajer = '741104013';</pre>
	<pre>[mysql&gt; SELECT * FROM departemen; +-----+-----+-----+   kode   nama   nik_manajer   +-----+-----+-----+   CSR   Customer Service and Relation   NULL     DUX   Desain dan User Experience   NULL     HRD   Human Resource Development   NULL     MGT   Top Level Management   NULL     MKK   Manajemen dan Kontrol Kualitas   NULL     MKT   Marketing   NULL     PKU   Pekerjaan Umum   741104019     PRD   Produksi   NULL     RND   Riset dan Pengembangan   NULL   +-----+-----+-----+ 9 rows in set (0.00 sec)</pre>
5	<p>Untuk mengubah beberapa kolom sekaligus dalam satu kali eksekusi statement UPDATE, dapat digunakan format berikut.</p>
	<pre>UPDATE departemen SET nama = 'Desain dan Pengalaman Pengguna',     nik_manajer = '741104013' WHERE kode = 'DUX';</pre>
	<pre>[mysql&gt; SELECT * FROM departemen; +-----+-----+-----+   kode   nama   nik_manajer   +-----+-----+-----+   CSR   Customer Service and Relation   NULL     DUX   Desain dan Pengalaman Pengguna   741104013     HRD   Human Resource Development   NULL     MGT   Top Level Management   NULL     MKK   Manajemen dan Kontrol Kualitas   NULL     MKT   Marketing   NULL     PKU   Pekerjaan Umum   741104019     PRD   Produksi   NULL     RND   Riset dan Pengembangan   NULL   +-----+-----+-----+ 9 rows in set (0.00 sec)</pre>
6	<p>Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke <b>Praktikum - Bagian 4</b>.</p>

**Praktikum - Bagian 4: Percobaan Statement DELETE**

Langkah	Keterangan																					
1	DELETE digunakan untuk menghapus satu atau lebih baris dari sebuah tabel. Misalkan kita ingin menghapus departemen yang memiliki nilai pada kolom nik_manajernya, maka format dasar statement seperti berikut dapat kita gunakan:																					
	<b>DELETE FROM departemen WHERE nik_manajer IS NOT NULL;</b>																					
	<pre>[mysql&gt; SELECT * FROM departemen;</pre> <table border="1"> <thead> <tr> <th>kode</th> <th>nama</th> <th>nik_manajer</th> </tr> </thead> <tbody> <tr> <td>CSR</td> <td>Customer Service and Relation</td> <td>NULL</td> </tr> <tr> <td>HRD</td> <td>Human Resource Development</td> <td>NULL</td> </tr> <tr> <td>MGT</td> <td>Top Level Management</td> <td>NULL</td> </tr> <tr> <td>MKK</td> <td>Manajemen dan Kontrol Kualitas</td> <td>NULL</td> </tr> <tr> <td>MKT</td> <td>Marketing</td> <td>NULL</td> </tr> <tr> <td>PRD</td> <td>Produksi</td> <td>NULL</td> </tr> </tbody> </table> <pre>6 rows in set (0.00 sec)</pre>	kode	nama	nik_manajer	CSR	Customer Service and Relation	NULL	HRD	Human Resource Development	NULL	MGT	Top Level Management	NULL	MKK	Manajemen dan Kontrol Kualitas	NULL	MKT	Marketing	NULL	PRD	Produksi	NULL
kode	nama	nik_manajer																				
CSR	Customer Service and Relation	NULL																				
HRD	Human Resource Development	NULL																				
MGT	Top Level Management	NULL																				
MKK	Manajemen dan Kontrol Kualitas	NULL																				
MKT	Marketing	NULL																				
PRD	Produksi	NULL																				
2	HATI-HATI apabila kita menggunakan statement DELETE <b>tanpa</b> WHERE! Cobalah eksekusi syntax SQL berikut:																					
	<b>DELETE FROM departemen;</b>																					
3	Semua data dalam satu tabel departemen akan hilang!																					
	<pre>[mysql&gt; SELECT * FROM departemen;</pre> <p><b>Empty set (0.00 sec)</b></p>																					

**D. TUGAS**

1. Kembalikan lagi data yang hilang pada tabel **departemen** dengan cara mengisi data yang sudah ada pada tabel **departemen\_backup!**
2. Tambahkan sedikitnya 10 (sepuluh) data karyawan dengan ketentuan:
  - a. 4 orang berjenis kelamin PRIA.
  - b. 6 orang lainnya WANITA.
  - c. Masing-masing departemen ditempati oleh TEPAT 1 orang Karyawan **KECUALI:**
    - i. Departemen dengan kode CSR dimana pada departemen ini harus ada TEPAT 3 orang karyawan berjenis kelamin WANITA
    - ii. Departemen HRD dimana pada departemen ini anggotanya TEPAT 2 orang berjenis kelamin 1 PRIA dan 1 WANITA.

3. Buatlah SEMUA departemen sedemikian sehingga masing-masing memiliki manajer dimana manajernya haruslah karyawan yang bekerja pada departemen tersebut!

## MODUL 3 Data Retrieval Language (DRL)

### A. TUJUAN

Mahasiswa diharapkan dapat memahami penggunaan dan dapat membuat SQL statement SELECT dengan berbagai klausa.

### B. TEORI

DRL atau DSL (Data Selection Language) adalah set perintah yang digunakan untuk mengambil data dari server basis data. DRL terdiri dari klausa yang dapat dikembangkan lebih lanjut, antara lain:

1. **FROM** : Memilih nama sebuah tabel
2. **WHERE** : Menunjukkan secara spesifik suatu row/baris yang akan dicari
3. **GROUP BY** : Menyusung data ke dalam grup
4. **HAVING** : Memilih diantara kelompok-kelompok data yang didefinisikan klausa GROUP BY
5. **ORDER BY** : Secara spesifik digunakan untuk menentukan suatu baris pada kolom tertentu
6. **AS** : Memberikan nama alias sementara untuk tabel atau kolom

DRL lebih diterapkan dalam beberapa hal dalam beberapa hal seperti :

1. **QUERY**
2. **SUB-QUERY**
3. **FUNGSI AGREGASI**

### QUERY

Query merupakan operasi yang melibatkan satu atau lebih tabel untuk melakukan retrieval data. Pengambilan data dilakukan dengan beberapa klausa berikut :

1. SELECT \* untuk memilih semua kolom  
Format

```
SELECT * FROM table_name;
```

2. SELECT dengan WHERE untuk menampilkan baris dengan suatu kondisi  
Format

```
SELECT column1, column2,...  
FROM table_name  
WHERE condition;
```

3. SELECT dengan DISTINCT untuk menampilkan data dengan eliminasi data yang sama (duplicate)  
Format

```
SELECT DISTINCT column FROM table_name;
```

4. SELECT dengan IN untuk menampilkan data yang spesifik  
Format

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

5. SELECT dengan BETWEEN untuk menampilkan data pada jarak (range) tertentu  
Format

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

6. SELECT dengan LIKE untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan

Format

```
SELECT column1,column2,...
FROM table_name
WHERE columnN LIKE pattern;
```

7. SELECT dengan GROUP BY untuk menampilkan susunan data dalam bentuk grup

Format

```
SELECT column1, column2,...
condition
FROM table_name
GROUP BY column1, column2, ...;
```

8. SELECT dengan ORDER BY untuk menampilkan baris secara spesifik dan terurut maju atau mundur

Format

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

9. SELECT dengan AND, OR and NOT untuk menampilkan data dengan kondisi dan atau atau tidak

Format

AND

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

10. SELECT dengan UNION, INTERSECT dan EXCEPT untuk menampilkan data dengan operasi himpunan yang melibatkan lebih dari satu tabel

Format

UNION

*Tanpa duplikasi*

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

*Dengan duplikasi*

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```

INTERSECT

*Tanpa duplikasi*

```
SELECT column_name(s) FROM table1  
INTERSECT  
SELECT column_name(s) FROM table2;
```

*Dengan duplikasi*

```
SELECT column_name(s) FROM table1  
INTERSECT ALL  
SELECT column_name(s) FROM table2;
```

EXCEPT

*Tanpa duplikasi*

```
SELECT column_name(s) FROM table1  
EXCEPT  
SELECT column_name(s) FROM table2;
```

*Dengan duplikasi*

```
SELECT column_name(s) FROM table1
EXCEPT ALL
SELECT column_name(s) FROM table2;
```

## FUNGSI AGREGASI

Fungsi agregasi akan melakukan perhitungan kepada sekumpulan nilai dan menghasilkan suatu nilai tunggal.

Jenis:

1. **AVG** untuk menghitung rata-rata

Format

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

2. **MAX** untuk mencari nilai tertinggi

Format

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

3. **MIN** untuk mencari nilai terendah

Format

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

4. **SUM** untuk menjumlahkan sekumpulan nilai

Format

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

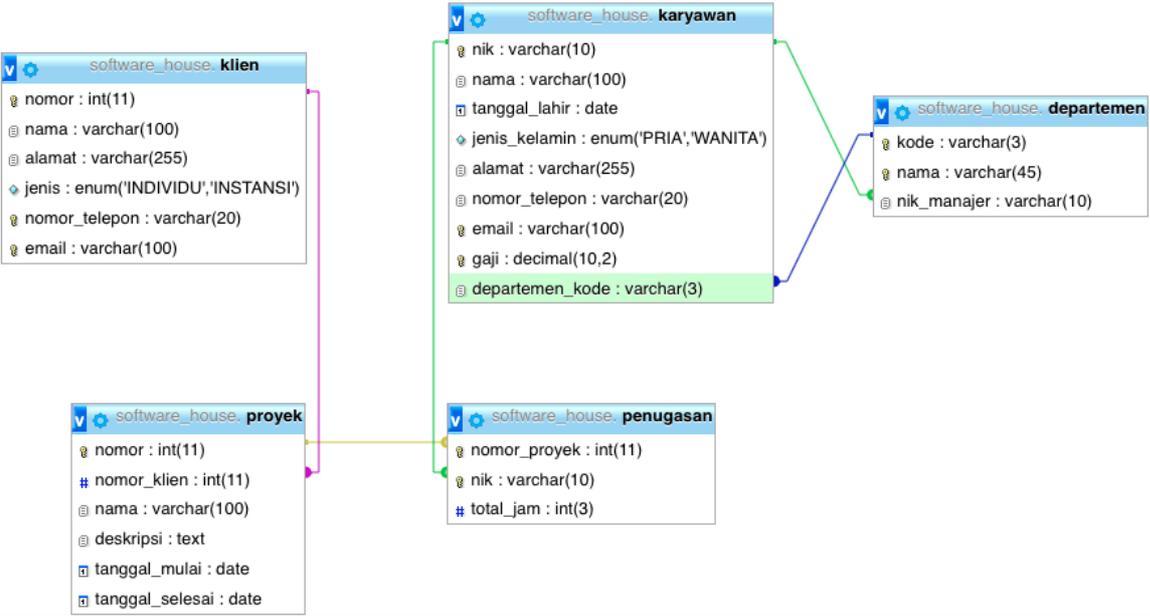
5. **COUNT()** untuk menunjukkan jumlah baris pada kolom yang diinginkan

Format

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

### C. LANGKAH PRAKTIKUM

#### Praktikum – Bagian 1: Membuat Database untuk Percobaan

Langkah	Keterangan																														
1	Berdasarkan Database yang dibuat pada praktikum sebelumnya, maka gunakan kembali database tersebut pada praktikum pertemuan ini																														
2	Selanjutnya tambahkan beberapa tabel lagi yaitu klien, penugasan dan proyek seperti sekma pada gambar dibawah ini																														
	 <p>The diagram shows four tables in a database named 'software_house':</p> <ul style="list-style-type: none"> <li><b>klien</b>: nomor (int(11)), nama (varchar(100)), alamat (varchar(255)), jenis (enum('INDIVIDU','INSTANSI')), nomor_telepon (varchar(20)), email (varchar(100)).</li> <li><b>proyek</b>: nomor (int(11)), nomor_klien (int(11)), nama (varchar(100)), deskripsi (text), tanggal_mulai (date), tanggal_selesai (date).</li> <li><b>karyawan</b>: nik (varchar(10)), nama (varchar(100)), tanggal_lahir (date), jenis_kelamin (enum('PRIA','WANITA')), alamat (varchar(255)), nomor_telepon (varchar(20)), email (varchar(100)), gaji (decimal(10,2)), departemen_kode (varchar(3)).</li> <li><b>departemen</b>: kode (varchar(3)), nama (varchar(45)), nik_manajer (varchar(10)).</li> </ul> <p>Relationships are indicated by lines: a pink line connects 'klien' to 'proyek'; a yellow line connects 'proyek' to 'karyawan'; a green line connects 'karyawan' to 'departemen'; and a blue line connects 'departemen' to 'karyawan'.</p>																														
3	Tambahkan data pada seluruh tabel sehingga jika menggunakan query untuk menampilkan data maka akan tampil data pada masing-masing tabel sebagai berikut:																														
	<pre>SELECT * FROM departemen;</pre> <pre>MariaDB [software_house]&gt; select * from departemen;</pre> <table border="1" data-bbox="352 1473 1126 1742"> <thead> <tr> <th>kode</th> <th>nama</th> <th>nik_manajer</th> </tr> </thead> <tbody> <tr><td>CSR</td><td>Customer Service and Relation</td><td>991567</td></tr> <tr><td>DUX</td><td>Desain dan User Experience</td><td>991566</td></tr> <tr><td>HRD</td><td>Human Resource Development</td><td>991565</td></tr> <tr><td>MGT</td><td>Top Level Management</td><td>991563</td></tr> <tr><td>MKK</td><td>Manajemen dan Kontrol Kualitas</td><td>991568</td></tr> <tr><td>MKT</td><td>Marketing</td><td>991561</td></tr> <tr><td>PKU</td><td>Pekerjaan Umum</td><td>991564</td></tr> <tr><td>PRD</td><td>Poduksi</td><td>991571</td></tr> <tr><td>RND</td><td>Riset dan Pengembangan</td><td>991562</td></tr> </tbody> </table>	kode	nama	nik_manajer	CSR	Customer Service and Relation	991567	DUX	Desain dan User Experience	991566	HRD	Human Resource Development	991565	MGT	Top Level Management	991563	MKK	Manajemen dan Kontrol Kualitas	991568	MKT	Marketing	991561	PKU	Pekerjaan Umum	991564	PRD	Poduksi	991571	RND	Riset dan Pengembangan	991562
kode	nama	nik_manajer																													
CSR	Customer Service and Relation	991567																													
DUX	Desain dan User Experience	991566																													
HRD	Human Resource Development	991565																													
MGT	Top Level Management	991563																													
MKK	Manajemen dan Kontrol Kualitas	991568																													
MKT	Marketing	991561																													
PKU	Pekerjaan Umum	991564																													
PRD	Poduksi	991571																													
RND	Riset dan Pengembangan	991562																													
	<pre>SELECT * FROM karyawan;</pre>																														

```
MariaDB [software_house]> select * from karyawan;
```

nik kode	nama nomor_telepon	tanggal_lahir email	jenis_kelamin gaji	alamat departemen
991561 I No. 10	Miftahul Huda 081555777111	1987-11-02 mifta.huda@gmail.com	PRIA 5000000.00	Jl. Mawar Gang MKT
991562 XI No. 3	Samsul Arif 081245789245	1970-03-03 s_arif@gmail.com	PRIA 6000000.00	Jl. Gatut Kaca RND
991563 no.6	Rosyid Akbar 085624819287	1977-01-10 r.akbar@yahoo.co.id	PRIA 6900000.00	Jl.Selorejo III MGT
991564 okwaru I No.1	Bandi Hidayat 087123456789	1984-12-12 hidayat_b@yahoo.com	PRIA 3100000.00	Jl. Simpang Low PKU
991565 III No. 2	Sandy Aulia 081245781115	1983-03-03 m_aulia@yahoo.com	WANITA 3600000.00	Jl. Gatut Kaca HRD
991566 o.1	Myta Alia 081124819287	1991-01-02 m_aulia@yahoo.co.com	WANITA 5200000.00	Jl.Selorejo I n DUX
991567 okwaru XI No.1	Rima Pratiwi 081555456781	1984-01-05 r_pratiwi@gmail.com	WANITA 3500000.00	Jl. Simpang Low CSR
991568 No. 10	Marina Prameswari 085645781115	1981-08-03 marina_p@gmail.com	WANITA 2900000.00	Jl. Malaka III MKK
991569 4	Winda Savitri 085624819282	1981-03-10 w_winda@yahoo.com	WANITA 3250000.00	Jl.Bromo XI no. CSR
991571 No.7	Fitri Indah Permata 085755456782	1977-01-01 f_permata@gmail.com	WANITA 3850000.00	Jl. Jombang II PRD
991572 no. 3	Rini Setyo Rini 081333257899	1988-04-08 rini_set@gmail.com	WANITA 3450000.00	Jl. Pungging IU CSR
991573 no. 3	Mario Saputra 98582459182	1982-05-05 mar_putra@gmail.com	PRIA 3150000.00	Jl. Puri Mas IU HRD

```
SELECT * FROM klien;
```

```
MariaDB [software_house]> select * from klien;
```

nomor	nama	alamat	jenis	nomot_telepon
1741	Ade Febriyanto	Jl. Merauke no 18	INSTANSI	081456744891
1742	Agung Firdaus	Jl. Lama no 10	INSTANSI	081456777896
1743	Andy Antono	Jl. Baru no 190	INSTANSI	081456777890
1744	Angelia Andriana	Jl. Baru no 10	INSTANSI	081456177890
1745	Ariesta Arif	Jl. Bromo no. 10	INDIVIDU	081456717390
1746	Bayu Chandra	Jl. Bandung U no.7	INSTANSI	081456777811
1747	Choirunisa Halim	Jl. Bandung IU no 2	INDIVIDU	081456777888
1748	Hidayat Achmad	Jl. Bandung IU no.1	INDIVIDU	08145677891
1749	Adria Saputra	Jl. Palmerah IU no.23	INSTANSI	081466777890
1750	Agung Adji	Jl. Palmerah IU no.44	INSTANSI	081453777890
1751	Agus Salim	Jl. Palmerah IU no.61	INDIVIDU	0221456777890
1752	Agustiana Rianti	Jl. Palmerah IU no.99	INDIVIDU	081456847890
1753	Rohid Mori	Jl. Palmerah IU no.19	INDIVIDU	081566777890
1754	Almira Alif	Jl. Payung no. 9	INSTANSI	081456577890
1755	Alvin Dani	Jl. Hamid Rusdi no.19	INSTANSI	081456777764
1756	Amelia Sari	Jl. Hamid Rusdi no.19	INSTANSI	081456777123
1757	Yulia Rahmawati	Jl. Hamid Rusdi no.10	INSTANSI	081456777321
1758	Maria Mirna	Jl. Permata IX no.90	INSTANSI	081456777853
1759	Zakiah Safitri	Jl. Permata II no.10	INSTANSI	081456712631
1760	Yunus Saputra	Jl. Permata XI no.4	INDIVIDU	0814567410891
1761	Yudha Hermadi	Jl. Permata III no.7	INSTANSI	085656777890
1762	Rani Savira	Jl. Permata I no.1	INSTANSI	085656777891
1763	Rini Hayati	Jl. Saudagar I no.1	INSTANSI	085656777822
1764	Binar Candra	Jl. Saudagar II no.1	INSTANSI	085656111111
1765	Sita Ayunda	Jl. Saudagar U no.4	INSTANSI	085656222222

```
SELECT * FROM proyek;
```

	<pre> MariaDB [software_house]&gt; select * from proyek; +-----+-----+-----+-----+-----+-----+   nomor   nomor_klien   nama        deskripsi   tanggal_mulai   tanggal_selesai   +-----+-----+-----+-----+-----+-----+   1       1741          Evergreen   NULL        2016-03-12       2016-06-12         2       1747          Amber Wae   NULL        2016-03-13       2016-04-13         3       1750          Einstein    NULL        2016-07-13       2016-12-13         4       1753          Phoenix     NULL        2016-01-14       2016-03-14         5       1760          Brigh Side   NULL        2016-04-16       2016-06-16       +-----+-----+-----+-----+-----+-----+ </pre>
	<pre> SELECT * FROM penugasan;  MariaDB [software_house]&gt; select * from penugasan; +-----+-----+-----+   nomor_proyek   nik        total_jam   +-----+-----+-----+   1               991561    10            1               991566    10            2               991567    10            3               991568    7             4               991571    15            5               991573    5           +-----+-----+-----+ </pre>
4	Setelah selesai membuat database diatas, lanjutkan ke <b><u>Praktikum – Bagian 2.</u></b>

**Praktikum – Bagian 2: Percobaan SELECT QUERY**

Langkah	Keterangan
1	<p>Untuk menampilkan baris dengan suatu kondisi tertentu, digunakan statement <b>SELECT</b> dengan <b>WHERE</b>. Eksekusi SQL berikut untuk menampilkan (record) pada tabel <b>karyawan</b> yang memiliki kolom departemen_kode='MKT'.</p> <pre> SELECT nama FROM karyawan WHERE departemen_kode = 'MKT'; </pre> <pre> +-----+   nama        +-----+   Miftahul Huda   +-----+ </pre>
2	<p>Untuk menampilkan data dengan eliminasi data yang sama (duplicate), digunakan statement <b>SELECT</b> dengan <b>DISTINCT</b>. Eksekusi SQL berikut untuk menampilkan kolom total_jam dari tabel <b>penugasan</b> yang bernilai tidak sama.</p> <pre> SELECT DISTINCT total_jam FROM penugasan; </pre>

	<pre> +-----+   total_jam   +-----+            10               7              15               5   +-----+ </pre>
3	<p>Untuk menampilkan data yang spesifik digunakan statement <b>SELECT</b> dengan <b>IN</b>. Eksekusi SQL berikut untuk menampilkan kolom nim dan nik dari tabel <b>karyawan</b> yang memiliki gaji 3100000 atau 6900000.</p>
	<pre> SELECT nik, nama FROM karyawan WHERE gaji IN (3100000,6900000); </pre>
	<pre> +-----+   nik        nama            +-----+   991564     Bandi Hidayat     991563     Rosyid Akbar    +-----+ </pre>
4	<p>Untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan digunakan <b>SELECT</b> dengan <b>LIKE</b>. Eksekusi SQL berikut untuk menampilkan kolom nama, alamat, dan jenis pada tabel <b>klien</b> yang memiliki nama dengan huruf awal 'a'.</p>
	<pre> SELECT nama, alamat, jenis FROM klien WHERE nama like 'a%'; </pre>
	<pre> +-----+   nama            alamat            jenis            +-----+   Ade Febriyanto   Jl. Merauke no 18   INSTANSI          Agung Firdaus    Jl. Lama no 10     INSTANSI          Andy Antono      Jl. Baru no 190    INSTANSI          Angelia Andriana   Jl. Baru no 10    INSTANSI          Ariesta Arif     Jl. Bromo no. 10   INDIVIDU          Adria Saputra    Jl. Palmerah IU no.23   INSTANSI          Agung Adji       Jl. Palmerah IU no.44   INSTANSI          Agus Salim       Jl. Palmerah IU no.61   INDIVIDU          Agustiana Rianti   Jl. Palmerah IU no.99   INDIVIDU          Almira Alif      Jl. Payung no. 9    INSTANSI          Alvin Dani       Jl. Hamid Rusdi no.19   INSTANSI          Amelia Sari      Jl. Hamid Rusdi no.19   INSTANSI        +-----+ </pre>
5	<p>Untuk menampilkan susunan data dalam bentuk grup, digunakan <b>SELECT</b> dengan <b>GROUP BY</b>. Eksekusi SQL berikut untuk menampilkan kolom nik, nama, gaji, dan departemen_kode pada tabel <b>karyawan</b> yang dikelompokkan berdasarkan gaji.</p>
	<pre> SELECT nik, nama, gaji, departemen_kode FROM karyawan GROUP BY gaji; </pre>

	<pre> +-----+-----+-----+-----+   nik        nama                  gaji                  departemen_kode      +-----+-----+-----+-----+   991568     Marina Prameswari    2900000.00           MKK                     991564     Bandi Hidayat        3100000.00           PKU                     991573     Mario Saputra        3150000.00           HRD                     991569     Winda Savitri       3250000.00           CSR                     991572     Rini Setyo Rini     3450000.00           CSR                     991567     Rima Pratiwi        3500000.00           CSR                     991565     Sandy Aulia          3600000.00           HRD                     991571     Fitri Indah Permata   3850000.00           PRD                     991561     Miftahul Huda       5000000.00           MKT                     991566     Myta Alia            5200000.00           DUX                     991562     Samsul Arif         6000000.00           RND                     991563     Rosyid Akbar        6900000.00           MGT                   +-----+-----+-----+-----+ </pre>
6	<p>Untuk menampilkan baris secara spesifik dan terurut maju atau mundur, digunakan <b>SELECT</b> dengan <b>ORDER BY</b>. Eksekusi SQL berikut untuk menampilkan kolom nama dan tanggal_mulai pada tabel <b>proyek</b> dari tanggal yang paling lama hingga baru.</p>
	<pre> SELECT nama, tanggal_mulai FROM proyek ORDER BY tanggal_mulai ASC; </pre>
	<pre> +-----+-----+   nama        tanggal_mulai   +-----+-----+   Phoenix     2016-01-14         Evergreen   2016-03-12         Amber Wae   2016-03-13         Brigh Side  2016-04-16         Einstein    2016-07-13       +-----+-----+ </pre>
7	<p>Untuk menampilkan data dengan kondisi dan atau atau tidak, digunakan <b>SELECT</b> dengan <b>AND</b>, <b>OR</b> and <b>NOT</b>. Eksekusi SQL berikut untuk menampilkan semua kolom pada tabel <b>karyawan</b> dengan jenis kelamin 'WANITA' dan gaji &lt; 5000000.</p>
	<pre> SELECT * FROM karyawan WHERE jenis_kelamin = 'WANITA' AND gaji &lt; 5000000; </pre>
	<pre> MariaDB [software_house]&gt; select * from karyawan where jenis_kelamin='WANITA' and gaji&gt;5000000; +-----+-----+-----+-----+-----+-----+   nik        nama                  tanggal_lahir        jenis_kelamin        alamat                  nomor_telepon         ----- ----- ----- ----- ----- -----    991566     Myta Alia            1991-01-02           WANITA               Jl.Selorejo I no.1     081124819287            m_aulia@yahoo.co.com   5200000.00           DUX                  +-----+-----+-----+-----+-----+-----+ </pre>
8	<p>Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan <b>SELECT</b> dengan <b>UNION</b>. <b>UNION</b> secara otomatis akan menghilangkan duplikasi. Eksekusi SQL berikut untuk menampilkan kolom nik yang ada di tabel <b>penugasan</b> atau <b>departemen</b>.</p>
	<pre> SELECT nik </pre>

	<pre>FROM penugasan UNION ALL SELECT nik_manajer FROM departemen;</pre>
	<pre>+-----+    nik    +-----+   991561     991566     991567     991568     991571     991573     991562     991563     991564     991565   +-----+</pre>
9	<p>Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan <b>SELECT</b> dengan <b>UNION ALL</b>. <b>UNION ALL</b> juga akan menampilkan duplikasi data. Eksekusi SQL berikut untuk menampilkan kolom nik yang ada di tabel <b>penugasan</b> atau <b>departemen</b>.</p>
	<pre>SELECT nik FROM penugasan UNION ALL SELECT nik_manajer FROM departemen;</pre>
	<pre>+-----+    nik    +-----+   991561     991566     991561     991567     991568     991571     991573     991561     991562     991563     991564     991565     991566     991567     991568     991571   +-----+</pre>
10	<p>Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke <b>Praktikum - Bagian 3</b>.</p>

**Praktikum - Bagian 3: Percobaan FUNGSI AGREGASI**

Langkah	Keterangan
1	Untuk menghitung rata-rata, digunakan tambahan statement <b>AVG</b> . Eksekusi SQL berikut untuk menampilkan rata-rata gaji sesuai kolom gaji dari tabel <b>karyawan</b> .
	<code>SELECT AVG(gaji) FROM karyawan;</code>
	<pre> +-----+    AVG(gaji)    +-----+   4158333.333333   +-----+ </pre>
2	Untuk menghitung nilai tertinggi, digunakan tambahan statement <b>MAX</b> . Eksekusi SQL berikut untuk menampilkan gaji tertinggi sesuai kolom gaji dari tabel <b>karyawan</b> .
	<code>SELECT MAX(gaji) FROM karyawan;</code>
	<pre> +-----+    MAX(gaji)    +-----+   6900000.00   +-----+ </pre>
3	Untuk menghitung nilai terendah, digunakan tambahan statement <b>MIN</b> . Eksekusi SQL berikut untuk menampilkan gaji terendah sesuai kolom gaji dari tabel <b>karyawan</b> .
	<code>SELECT MIN(gaji) FROM karyawan;</code>
	<pre> +-----+    MIN(gaji)    +-----+   2900000.00   +-----+ </pre>
4	Untuk menghitung total gaji keseluruhan digunakan tambahan statement <b>SUM</b> . Eksekusi SQL berikut untuk menampilkan total gaji sesuai kolom gaji dari tabel <b>karyawan</b> .
	<code>SELECT SUM(gaji) FROM karyawan</code>
	<pre> +-----+    sum(gaji)    +-----+   49900000.00   +-----+ </pre>
5	Untuk menunjukkan jumlah baris pada kolom yang diinginkan digunakan <b>COUNT()</b> . Eksekusi SQL berikut untuk menampilkan jumlah baris pada kolom dari tabel <b>karyawan</b> .
	<code>SELECT COUNT(gaji) FROM karyawan;</code>

```
+-----+
| count(gaji) |
+-----+
|           12 |
+-----+
```

#### D. TUGAS

1. Lakukan pencarian jumlah karyawan yang bekerja pada kode\_departemen 'CSR' kemudian tunjukkan sintaksis dan outputnya!
2. Carilah data yang menunjukkan semua nik dari seorang manajer yang berperan dalam penugasan. Gunakan operasi intersect! Tunjukkan sintaksis dan output pencarian tersebut!
3. Tunjukkan total nik, nomor\_proyek dan total\_jam dari tabel penugasan yang dikelompokkan berdasarkan nomor proyek yang sama berjumlah kurang dari dua. Kelompok tersebut diurutkan sesuai nomor\_proyek secara terurut dari atas ke bawah. Sertakan sintaksis dan output dari sintaksis SELECT tersebut.

## MODUL 4 JOIN

### A. TUJUAN

1. Mahasiswa mampu Memahami keterhubungan entitas di dalam basis data
2. Mahasiswa mampu Memahami jenis-jenis operasi pengambilan data di beberapa entitas
3. Mahasiswa mampu Mampu menyelesaikan kasus-kasus *retrieval* yang melibatkan lebih dari satu entitas

### B. TEORI

#### ✓ Relationship

Relationship adalah suatu hubungan antara beberapa entitas. Konsep ini sangat penting sekali di dalam basis data, di mana memungkinkan entitas-entitas untuk saling berhubungan satu sama lain. Di dalam sebuah relationship, primary key memiliki peran penting untuk mengaitkan entitas. Selain itu, primary key juga digunakan untuk mendefinisikan batasan keterhubungan.

#### ✓ Join

Join merupakan salah satu konstruksi dasar dari SQL dan basis data. Join dapat didefinisikan sebagai kombinasi record dari dua atau lebih tabel di dalam basis data relasional dan menghasilkan sebuah tabel (temporary) baru yang disebut sebagai joined table. Join dapat diklasifikasikan ke dalam dua jenis: inner dan outer.

##### a. Inner Join

Inner join pada dasarnya adalah menemukan persimpangan (intersection) antara dua buah tabel.

Sintaks *inner join* diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1
INNER JOIN r2
ON r1.join_key = r2.join_key
```

##### Cara 1

```
SELECT A1, A2, ..., An
FROM r1
INNER JOIN r2
ON r1.join_key = r2.join_key
```

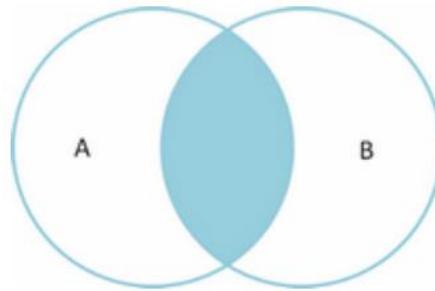
*Inner join* juga dapat direpresentasikan dalam bentuk implisit.

```
SELECT A1, A2, ..., An
FROM r1, r2
WHERE r1.key = r2.key
```

##### Cara 2

```
SELECT A1, A2, ..., An
FROM r1, r2
WHERE r1.key = r2.key
```

Misalkan terdapat tabel A dan B, maka hasil *inner join* dapat diperlihatkan—sebagai bidang terarsir—dalam diagram Venn seperti Gambar 1.



Gambar 1. Inner Join

b. Outer Join

- **Left Outer Join**

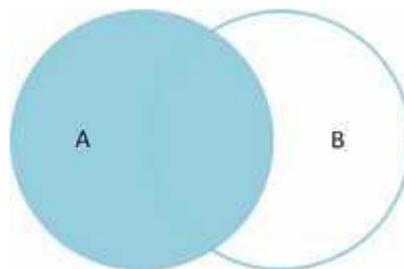
Left outer join (atau left join) mengembalikan semua nilai dari tabel kiri ditambah dengan nilai dari tabel kanan yang sesuai (atau NULL jika tidak ada nilai yang sesuai).

```
SELECT A1, A2, ..., An
FROM r1
LEFT OUTER JOIN r2
ON r1.join_key = r2.join_key
```

**Syntax**

```
SELECT A1, A2, ..., An
FROM r1
LEFT OUTER JOIN r2
ON r1.join_key = r2.join_key
```

*Left outer join* antara tabel A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar 2.



Gambar 2. Left Outer Join

- **Right Outer Join**

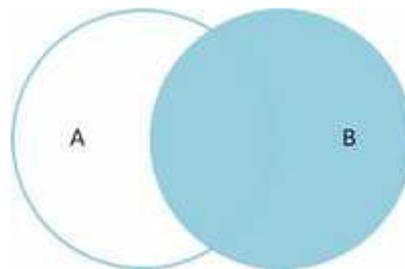
Right outer join (atau right join) pada dasarnya sama seperti left join, namun dalam bentuk terbalik—kanan dan kiri. Sintaks *right outer join* diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1
  RIGHT OUTER JOIN r2
  ON r1.join_key = r2.join_key
```

**Syntax**

```
SELECT A1, A2, ..., An
FROM r1
RIGHT OUTER JOIN r2
ON r1.join_key = r2.join_key
```

*Right outer join* antara tabel A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar 3.



Gambar 3. Right Outer Join

- **Full Outer Join**

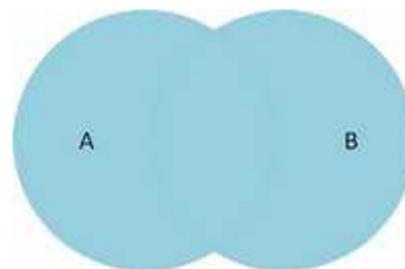
Full outer join (atau full join) pada hakekatnya merupakan kombinasi dari left dan right join. Sintaks *full outer join* diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1
  FULL OUTER JOIN r2
  ON r1.join_key = r2.join_key
```

**Syntax**

```
SELECT A1, A2, ..., An
FROM r1
FULL OUTER JOIN r2
ON r1.join_key = r2.join_key
```

Bentuk visual dari *full outer join* dapat diperlihatkan menggunakan diagram Venn seperti Gambar 4.



Gambar 4. Full Outer Join

Selain empat jenis join yang utama di atas, masih ada beberapa variasi join lainnya, seperti CROSS JOIN (cartesian product), NATURAL JOIN, dan sebagainya. Perlu juga diperhatikan, join bisa diimplementasikan dalam bentuk bersarang (nested join). Jadi, di dalam sebuah operasi join bisa terdapat operasi join lainnya.

### C. LANGKAH PRAKTIKUM

Langkah	Keterangan																		
1	Buat database nama_kantor (isi dengan nama anda)																		
2	Buat tabel karyawan dan tabel departemen dengan struktur sebagai berikut:																		
	<pre>mysql&gt; create table karyawan( -&gt; nama varchar (30) NOT NULL, -&gt; id_dep int (5) NOT NULL) -&gt; ENGINE = myISAM; Query OK, 0 rows affected (0.11 sec)  mysql&gt; create table departemen( -&gt; id_dep int (5) NOT NULL, -&gt; nama_dep varchar (30) NOT NULL, -&gt; PRIMARY KEY(id_dep)) -&gt; ENGINE = myISAM; Query OK, 0 rows affected (0.34 sec)</pre>																		
3	<p>Lakukan pengisian data pada masing-masing tabel. Data yang digunakan adalah sebagai berikut:</p> <ul style="list-style-type: none"> <li>• Tabel Karyawan <table border="1" data-bbox="525 1229 979 1657"> <thead> <tr> <th>Nama</th> <th>id_dep</th> </tr> </thead> <tbody> <tr> <td>Agus</td> <td>10</td> </tr> <tr> <td>Budi</td> <td>16</td> </tr> <tr> <td>Citra</td> <td>12</td> </tr> <tr> <td>Dani</td> <td>17</td> </tr> </tbody> </table> </li> <li>• Tabel Departmen <table border="1" data-bbox="525 1702 979 2031"> <thead> <tr> <th>id_dep</th> <th>nama_dep</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>Penelitian</td> </tr> <tr> <td>11</td> <td>Pemasaran</td> </tr> <tr> <td>12</td> <td>SDM</td> </tr> </tbody> </table> </li> </ul>	Nama	id_dep	Agus	10	Budi	16	Citra	12	Dani	17	id_dep	nama_dep	10	Penelitian	11	Pemasaran	12	SDM
Nama	id_dep																		
Agus	10																		
Budi	16																		
Citra	12																		
Dani	17																		
id_dep	nama_dep																		
10	Penelitian																		
11	Pemasaran																		
12	SDM																		

		13	Keuangan																									
<b>Inner Join</b>																												
4	<p>Sebagaimana dijelaskan, inner join akan mengembalikan data di tabel A dan B yang sesuai. Sebagai contoh, kita bisa mendapatkan data karyawan yang memiliki departemen.</p>																											
	<p>Cara 1</p> <pre>mysql&gt; select * from karyawan       -&gt; INNER JOIN departemen       -&gt; ON karyawan.id_dep = departemen.id_dep;</pre> <table border="1"> <thead> <tr> <th>nama</th> <th>id_dep</th> <th>id_dep</th> <th>nama_dep</th> </tr> </thead> <tbody> <tr> <td>Agus</td> <td>10</td> <td>10</td> <td>Penelitian</td> </tr> <tr> <td>Citra</td> <td>12</td> <td>12</td> <td>SDM</td> </tr> </tbody> </table> <p>2 rows in set (0.03 sec)</p> <p>Cara 2</p> <pre>mysql&gt; select * from karyawan, departemen       -&gt; where karyawan.id_dep = departemen.id_dep;</pre> <table border="1"> <thead> <tr> <th>nama</th> <th>id_dep</th> <th>id_dep</th> <th>nama_dep</th> </tr> </thead> <tbody> <tr> <td>Agus</td> <td>10</td> <td>10</td> <td>Penelitian</td> </tr> <tr> <td>Citra</td> <td>12</td> <td>12</td> <td>SDM</td> </tr> </tbody> </table> <p>2 rows in set (0.00 sec)</p>				nama	id_dep	id_dep	nama_dep	Agus	10	10	Penelitian	Citra	12	12	SDM	nama	id_dep	id_dep	nama_dep	Agus	10	10	Penelitian	Citra	12	12	SDM
nama	id_dep	id_dep	nama_dep																									
Agus	10	10	Penelitian																									
Citra	12	12	SDM																									
nama	id_dep	id_dep	nama_dep																									
Agus	10	10	Penelitian																									
Citra	12	12	SDM																									
<b>Outer Join</b>																												
<b>Left Outer Join</b>																												
5	<p>Data seluruh karyawan (yang memiliki departemen maupun tidak)</p> <pre>mysql&gt; select * from karyawan k left outer join departemen d       -&gt; on k.id_dep = d.id_dep;</pre> <table border="1"> <thead> <tr> <th>nama</th> <th>id_dep</th> <th>id_dep</th> <th>nama_dep</th> </tr> </thead> <tbody> <tr> <td>Agus</td> <td>10</td> <td>10</td> <td>Penelitian</td> </tr> <tr> <td>Budi</td> <td>16</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>Citra</td> <td>12</td> <td>12</td> <td>SDM</td> </tr> <tr> <td>Dani</td> <td>17</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table> <p>4 rows in set (0.03 sec)</p>				nama	id_dep	id_dep	nama_dep	Agus	10	10	Penelitian	Budi	16	NULL	NULL	Citra	12	12	SDM	Dani	17	NULL	NULL				
nama	id_dep	id_dep	nama_dep																									
Agus	10	10	Penelitian																									
Budi	16	NULL	NULL																									
Citra	12	12	SDM																									
Dani	17	NULL	NULL																									
6	<p>Data karyawan yang tidak memiliki departemen</p> <pre>mysql&gt; select * from karyawan k left outer join departemen d       -&gt; on k.id_dep = d.id_dep       -&gt; where d.id_dep is null;</pre> <table border="1"> <thead> <tr> <th>nama</th> <th>id_dep</th> <th>id_dep</th> <th>nama_dep</th> </tr> </thead> <tbody> <tr> <td>Budi</td> <td>16</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>Dani</td> <td>17</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table> <p>2 rows in set (0.02 sec)</p>				nama	id_dep	id_dep	nama_dep	Budi	16	NULL	NULL	Dani	17	NULL	NULL												
nama	id_dep	id_dep	nama_dep																									
Budi	16	NULL	NULL																									
Dani	17	NULL	NULL																									
<b>Right Outer Join</b>																												
7	Lakukan Rigt Outer Join																											

```
mysql> select * from karyawan k right outer join departemen d
-> on k.id_dep = d.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
NULL	NULL	11	Pemasaran
Citra	12	12	SDM
NULL	NULL	13	Keuangan

4 rows in set (0.00 sec)

**Full Outer Join**

8

Lakukan Full outer Join

```
mysql> select * from karyawan k left outer join departemen d
-> on k.id_dep = d.id_dep
-> union
-> select * from karyawan k right outer join departemen d
-> on k.id_dep = d.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Budi	16	NULL	NULL
Citra	12	12	SDM
Dani	17	NULL	NULL
NULL	NULL	11	Pemasaran
NULL	NULL	13	Keuangan

6 rows in set (0.00 sec)

**Cross Join**

9

Lakukan Cross Join

Cara 1

```
mysql> select * from karyawan cross join departemen;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Budi	16	10	Penelitian
Citra	12	10	Penelitian
Dani	17	10	Penelitian
Agus	10	11	Pemasaran
Budi	16	11	Pemasaran
Citra	12	11	Pemasaran
Dani	17	11	Pemasaran
Agus	10	12	SDM
Budi	16	12	SDM
Citra	12	12	SDM
Dani	17	12	SDM
Agus	10	13	Keuangan
Budi	16	13	Keuangan
Citra	12	13	Keuangan
Dani	17	13	Keuangan

16 rows in set (0.00 sec)

Cara 2

```
mysql> select * from karyawan, departemen;
+-----+-----+-----+-----+
| nama | id_dep | id_dep | nama_dep |
+-----+-----+-----+-----+
| Agus | 10 | 10 | Penelitian |
| Budi | 16 | 10 | Penelitian |
| Citra | 12 | 10 | Penelitian |
| Dani | 17 | 10 | Penelitian |
| Agus | 10 | 11 | Pemasaran |
| Budi | 16 | 11 | Pemasaran |
| Citra | 12 | 11 | Pemasaran |
| Dani | 17 | 11 | Pemasaran |
| Agus | 10 | 12 | SDM |
| Budi | 16 | 12 | SDM |
| Citra | 12 | 12 | SDM |
| Dani | 17 | 12 | SDM |
| Agus | 10 | 13 | Keuangan |
| Budi | 16 | 13 | Keuangan |
| Citra | 12 | 13 | Keuangan |
| Dani | 17 | 13 | Keuangan |
+-----+-----+-----+-----+
16 rows in set (0.00 sec)
```

**D. TUGAS**

Perhatikan, dalam mengerjakan tugas praktikum ini, sebaiknya pernyataan SQL disimpan di file untuk kemudian dieksekusi. Berikut adalah data-data tabel yang akan digunakan (sesuaikan nilainya agar sama persis).

- Tabel mahasiswa

nim	Nama	jenis_kelamin	alamat
101	Arif	L	Jl. Kenangan
102	Budi	L	Jl. Jombang
103	Wati	P	Jl. Surabaya
104	Ika	P	Jl. Jombang
105	Tono	L	Jl. Jakarta
106	Iwan	L	Jl. Bandung
107	Sari	P	Jl. Malang

- Tabel ambil\_mk

nim	kode_mk
101	PTI447

103	TIK333
104	PTI333
104	PTI777
111	PTI123
123	PTI999

- Tabel matakuliah

kode_mk	nama_mk	sks	semester
PTI447	Praktikum Basis Data	1	3
TIK342	Praktikum Basis Data	1	3
PTI333	Basis Data Terdistribusi	3	5
TIK123	Jaringan Komputer	2	5
TIK333	Sistem Operasi	3	5
PTI123	Grafika Multimedia	3	5
PTI777	Sistem Informasi	2	3

### Soal

1. Dapatkan data mahasiswa yang mengambil matakuliah. Selesaikan dengan pendekatan join eksplisit dan implisit. Hindari duplikasi data.

```

+-----+
| nim | nama |
+-----+
| 101 | Arif |
| 103 | Wati |
| 104 | Ika  |
+-----+
3 rows in set (0.00 sec)

```

2. Kelompokkan data mahasiswa yang tidak mengambil matakuliah berdasarkan jenis kelaminnya, kemudian hitung banyaknya.

```

+-----+-----+
| jenis_kelamin | jumlah |
+-----+-----+
| L              |      3 |
| P              |      1 |
+-----+-----+
2 rows in set (0.00 sec)

```

3. Dapatkan nim dan nama mahasiswa yang mengambil matakuliah beserta kode\_mk dan nama\_mk yang diambilnya. Selesaikan dengan pendekatan join eksplisit dan implisit.

```

+-----+-----+-----+-----+
| nim | nama | kode_mk | nama_mk |
+-----+-----+-----+-----+
| 101 | Arif | PTI447  | Praktikum Basis Data |
| 103 | Wati | TIK333  | Sistem Operasi        |
| 104 | Ika  | PTI333  | Basis Data Terdistribusi |
| 104 | Ika  | PTI777  | Sistem Informasi      |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

4. Dapatkan nim, nama, dan total sks yang diambil oleh mahasiswa, di mana total sksnya lebih dari 4 dan kurang dari 10.

```

+-----+-----+-----+
| nim | nama | total_sks |
+-----+-----+-----+
| 104 | Ika  |          5 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

# 3

## PEMROGRAMAN VISUAL

# BAB 1

## FORM, KONTROL, PROPERTI DAN EVENT

### I. TUJUAN:

- Mengetahui lingkungan kerja Visual Basic 6.0
- Mengetahui Form dan Toolbox
- Mengetahui Property dan Event dari suatu kontrol
- Mengetahui deklarasi variabel

### II. DASAR TEORI

#### 2.1. Sejarah Visual Basic 6.0

Visual Basic 6.0 adalah bahasa pemrograman berbasis Windows. Saat ini, Visual Basic adalah salah satu bahasa pemrograman yang terbaik. Visual Basic merupakan pengembangan dari Basic. Basic (*Beginner's All-purpose Symbolic Instruction Code*) adalah sebuah bahasa pemrograman "kuno" yang merupakan awal dari bahasa-bahasa pemrograman tingkat tinggi lainnya. Basic dirancang pada tahun 1950-an dan ditujukan untuk dapat digunakan oleh para pemula. Biasanya Basic diajarkan untuk para pelajar sekolah menengah yang baru mengenal komputer, serta digunakan untuk pengembang program "cepat saji" yang ringan dan menyenangkan. Walaupun begitu, peran basic lebih dari sekedar itu saja. Banyak programmer andal saat ini memulai karirnya dengan mempelajari Basic. Sebagai bahasa pemrograman yang mutakhir, Visual Basic 6.0 didesain untuk dapat memanfaatkan fasilitas Windows, khususnya Windows 95/97/98 dan Windows NT. Visual Basic 6.0 juga merupakan bahasa pemrograman *Object Oriented Programming (OOP)*, yaitu pemrograman yang berorientasi objek. Visual Basic 6.0 menyediakan objek-objek yang sangat kuat, berguna, dan mudah dipakai. Dengan fasilitas tersebut, membuat Visual Basic 6.0 menjadi begitu diinginkan oleh *programmer*.

Microsoft Visual Basic 6.0 adalah sebuah bahasa pemrograman untuk *Windows* dan *Internet*. Sama seperti bahasa pemrograman *Basic*, *pascal*, *C* dan lain-lain. Tetapi *Basic*, *Pascal* dan *C* ditujukan untuk system operasi *Ms-DOS*, sedangkan *Visual Basic* ditujukan untuk system operasi *Windows* dan *Internet*. Jika anda sudah memahami bahasa pemrograman *Basic*, maka *Visual Basic* sudah lebih mudah anda pelajari, sebab sebagian besar perintah-perintah *Visual Basic* serupa dengan bahasa Basic.

Karena visual basic merupakan salah satu bahasa pemrograman sehingga sangat penting untuk kita pelajari, maka kami menyusun makalah ini. Dimana dalam makalah ini akan dijelaskan tentang visual basic, dan salah satu cara membuat program sederhana.

#### 2.2 Perkembangan Visual Basic

- 20 Mei 1991: Microsoft merilis Visual Basic versi 1.0 untuk Windows di Windows World 1991, Atlanta, Amerika Serikat. Programmer dapat membuat antar muka pengguna dengan mudah. Pembuatan aplikasi bisa lebih cepat dibandingkan sebelumnya. Kode

ditulis oleh programmer untuk memberikan aksi terhadap Event dari pengguna (bagaimana pengguna merespon kepada object yang terdapat di antar muka.

- 2. 1 September 1992: Microsoft mengumumkan Microsoft Visual Basic for MS-DOS® dalam edisi Standard dan Professional. Seperti Visual Basic untuk Windows, versi ini mengkombinasikan kemudahan pendisainan secara grafis dengan kekuatan dan keunggulan dalam banyak hal di pemrograman
- 2 November 1992: Microsoft mengumumkan kemampuan dari Visual Basic versi 2.0 untuk Windows pada edisi Professional dan Standard. Versi ini menyertakan lebih dari 300 fitur baru dan peningkatan untuk pengembangan aplikasi yang lebih cepat, pengaksesan ke fitur-fitur tingkat lanjut di Windows, dan produktivitas pengembang yang lebih besar lagi. Di antaranya: MDI Forms, ODBC, dan variabel
- 14 Mei 1993: Microsoft mengumumkan Visual Basic versi 3.0 untuk edisi Standard dan Professional. Versi 3.0 menyediakan kemudahan akses ke berbagai sumber data yang banyak dengan mengintegrasikan mesin database Microsoft Access Database for Windows 1.1 dan kemampuan di bidang aplikasi melalui Object Linking and Embedding (OLE) 2.0. Juga ditambahkan tools baru berupa controls baru, penggunaan yang lebih mudah, dan sebuah peningkatan standarisasi control bagi
- 29 Juni 1993: Microsoft mengumumkan bahwa Microsoft Visual Basic for Applications (VBA) akan diintegrasikan ke dalam Microsoft Excel 5.0 dan Microsoft Project 4.0. Visual Basic akan melayani bahasa Macro yang umum di versi yang akan datang dari aplikasi-aplikasi untuk Windows dan Macintosh. Fitur Macro yang umum untuk aplikasi dan OLE 2.0, keduanya merupakan elemen penting di visi dan strategi Microsoft untuk aplikasi yang dapat diprogram —yang akan memberikan kemampuan bagi pengguna untuk mengintegrasikan aplikasi-aplikasi, tugas-tugas otomatis, dan pembuatan solusi yang bersifat kostumais. VBA memiliki mesin pengembangan tangguh dan fleksibel yang sama yang membuat Visual Basic menjadi terkenal, teknologi pemrograman yang memperoleh
- 14 November 1994: Visual Basic versi 4.0 untuk Windows didemonstrasikan di Fall/COMDEX '94 di Las Vegas. Demonstrasi ini berfokus pada perannya sebagai aplikasi berbasis Windows 32-bit pertama yang menggunakan kontrol kostumais OLE (OCXs) —komponen perangkat lunak yang dapat digunakan kembali yang ditetapkan dengan spesifikasi OLE. Pada versi ini diperkenalkan: Class, OXC's, dan programmer dapat membuat add-ins sendiri dengan
- 12 September 1995: Visual Basic versi 4.0 untuk MS-DOS, Microsoft Windows NT®, dan Windows 95 diumumkan. Untuk meningkatkan edisi Standard dan Professional, maka Edisi Enterprise yang baru diperkenalkan untuk memenuhi kebutuhan perusahaan dan team pengembang. Semua versi menyertakan dukungan teknologi OLE yang lebih dikembangkan, perbaikan akses data, sebuah lingkungan pengembangan yang bersifat visual, dan migrasi ke Windows 95 dan Windows NT untuk aplikasi-aplikasi yang sudah
- 7 Desember 1995: Microsoft mengumumkan Microsoft Visual Basic Scripting Edition (VBScript), sebuah bahasa skrip Internet berbasiskan pada Visual Basic. VBScript merupakan sebuah skrip dengan performansi yang tinggi yang didisain untuk membuat isi yang bersifat aktif pada suatu situs. VBScript membolehkan pengembang untuk

mengaitkan dan mengotomatiskan berbagai jenis object di halaman situs, termasuk object OLE. VBScript menjadi sebuah spesifikasi yang diusulkan berlisensi gratis yang tersedia bagi komunitas

- 3 Februari 1997: Microsoft membuka sistem pemrograman Visual Basic versi 5.0, Edisi Professional — tool versi terakhir yang paling terkenal di dunia untuk membangun aplikasi yang tangguh (Rapid Application Development RAD) tool. Fitur-fitur Visual Basic versi 5.0 terdiri dari sejumlah performansi yang signifikan dan peningkatan produktivitas yang akan membuat para pengembang lebih efisien dan fleksibel serta cepat dalam memberikan solusi bagi pengguna akhir. Fitur-fitur tingkat lanjut — seperti kompilasi berjenis native code, akses database berkecepatan tinggi, dan sebuah lingkungan pengembangan yang ditingkatkan — membuat Visual Basic versi 5.0 menjadi versi yang paling hebat. Sebagai tambahan, kemampuan teknologi pembuatan komponen Microsoft ActiveX® di Visual Basic versi 5.0 akan membolehkan lebih dari 3 juta pengembang menggunakan Visual Basic dalam membuat aplikasi yang berbasis pada komponen yang bertujuan untuk digunakan di Internet, intranets, dan lingkungan tradisional client/
- 10 Maret 1997: Versi terakhir yang dirilis dari Visual Basic 5.0, Edisi Pembuatan Kontrol tersedia. Edisi Pembuatan Kontrol adalah cara mudah untuk membuat ActiveX Controls bagi Internet, intranets, dan aplikasi client/
- 15 Juni 1998: Microsoft mengumumkan Visual Basic versi 6.0, dan dimasukkan ke dalam Microsoft Visual Studio® versi 6.0. Fitur-fitur Visual Basic versi 6.0 menyediakan pengaksesan data secara terintegrasi dan bersifat grafis ke sumber data (data source) ODBC atau OLE DB manapun, dan perangkat tambahan database yang didisain untuk database Oracle dan Microsoft SQL Server™. Fitur unggulan di versi ini adalah: ActiveX Data Objects (ADO) untuk memanipulasi dan membuat database. Fitur Pengembangan Situs membawa kemudahan dalam penggunaan, model pemrograman berbasis komponen dari Visual Basic untuk membuat HTML – dan Dynamic HTML (DHTML) – berbasis aplikasi. Fitur-fitur baru ini — dikombinasikan dengan optimisasi performansi, pengembangan aplikasi yang disederhanakan dan debugging, dan dukungan untuk Microsoft teknologi server — membuat Visual Basic versi 6.0 sebuah pilihan yang ideal untuk membangun aplikasi berskala
- 2 September 1998: Visual Studio 6.0, solusi tool pengembangan lengkap berskala perusahaan tersedia. Visual Studio 6.0 menyertakan Visual Basic versi 6.0, Microsoft Visual C++®, Microsoft Visual FoxPro®, Microsoft Visual InterDev® Web, dan Microsoft Visual J++®. Bersamaan dengan diluncurkannya Developer Days '98, lebih dari 115 perusahaan papan atas mengumumkan dukungan terhadap Visual Studio 6.
- 28 April 1999: Lingkungan pengembangan VBA versi 6.0 dan Software Development Kit (SDK) VBA (SDK) versi 6.0 tersedia bagi vendor perangkat lunak pihak ketiga melalui program lisensi VBA. VBA 6.0 merupakan sebuah teknologi pengembangan yang tangguh untuk aplikasi yang dikemas secara kostumais dan merupakan sebuah komponen utama di Microsoft Office
- 15 Februari 2000: Steve Ballmer, presiden yang baru saja ditunjuk dan CEO dari Microsoft Corporation, berbicara kepada lebih dari 2.000 pengembang dalam pidatonya pada Visual

Basic Insiders Technical Summit (VBITS), menyatakan visi Microsoft terhadap situs yang dapat diprogram dan memaksa kembali komitmen perusahaan yang dalam terhadap para pengembang Visual

- 11 Juli 2000: Microsoft menyampaikan Tool Platform .NET untuk XML Web Services. Dalam pidatonya yang ditujukan pada acara Microsoft Professional Developers Conference (PDC) 2000 yang kedelapan, Paul Maritz, wakil presiden kelompok dari Platforms Group di Microsoft, memperlihatkan Microsoft .NET Framework dan Microsoft Visual Studio .NET. Ini merupakan versi terakhir dari perangkat pengembangan yang digunakan di seluruh dunia, Visual Studio .NET menyediakan dukungan untuk pengembangan drag-and-drop terhadap XML Web Services. Secara bersama, kedua produk tersebut menyediakan produktivitas yang tinggi kepada para pengembang, suatu lingkungan yang terdiri dari banyak bahasa untuk pembangunan, pengantaran, dan pengintegrasian XML Web Services pada platform Microsoft .
- 13 November 2000: Microsoft mengumumkan kemampuan Visual Studio .NET versi Beta 1. Dalam pidato acara COMDEX/Fall 2000, Bill Gates mengumumkan kemampuan versi beta pertama Visual Studio.NET dan .NET Framework, dua teknologi kunci untuk memungkinkan para pengembang membangun XML Web services pada platform .NET. XML Web services merupakan aplikasi dan komponen yang dibuat tersedia melewati situs dengan menggunakan XML dan Simple Object Access Protocol (SOAP), dan kunci yang dapat diprogram untuk membangun generasi masa depan Internet. Microsoft membuat Visual Studio .NET Beta 1 dan .NET Framework tersedia bagi jutaan pelanggan dan rekanan
- 13 Februari 2002: Microsoft mengumumkan kemampuan dari Visual Studio .NET versi akhir.

### 2.3 Fasilitas Visual Basic

Di dalam lingkungan Visual Basic, terdapat berbagai macam komponen, yaitu:

#### 1. Control Menu

Control Menu adalah menu yang digunakan terutama untuk memanipulasi jendela Visual Basic. Dari menu ini anda dapat mengubah ukuran, memindahkannya, atau menutup jendela.

#### 2. Menu

Menu Visual Basic berisi semua perintah Visual Basic yang dapat dipilih untuk melakukan tugas tertentu. Isi dari menu ini sebagian hampir sama dengan program-program Windows pada umumnya.

#### 3. Toolbar

Toolbar adalah tombol-tombol (*shortcut*) yang mewakili suatu perintah tertentu dari Visual Basic.

#### 4. Form Window

Form Window atau jendela form adalah daerah kerja utama tempat membuat program-program aplikasi Visual Basic.

## 5. Toolbox

Toolbox adalah sebuah “kotak piranti” yang mengandung semua objek atau “kontrol” yang dibutuhkan untuk membentuk suatu program aplikasi. Kontrol adalah suatu objek yang akan menjadi penghubung antara program aplikasi dan user-nya, dan yang kesemuanya harus diletakkan di dalam jendela form. Contoh komponen yang ada dalam toolbox :

- A. Label, digunakan untuk menampilkan teks yang tidak dapat diedit oleh user.
- B. Text Box, digunakan untuk memasukan dan mengedit teks.
- C. Option button, merupakan bagian dari Optiongroup, digunakan untuk menampilkan beberapa pilhan dimana user hanya dapat memilih satu.
- D. Image, digunakan untuk menampilkan bitmap, Ikon, file JPG atau GIF.
- E. Chek box, digunakan untuk memilih satu pilhan atau lebih secara bersamaan yang disediakan pada program.
- F. ComboBox, digunakan untuk memilih satu pilhan yang disediakan pada program yang menampilkan seluruh pilihan yang tersedia saat objek tersebut diklik.
- G. Commond Button, digunakan untuk membuat suatu aksi saat objek tersebut diklik.
- H. List box, digunakan untuk menampilkan daftar item dimana user dapat memilih dirinya.
- I. Frame, digunakan sebagai wadah Control atau Toolbox yang lain.
- J. Timer, digunakan untuk mengeksekusi event timer dalam interval waktu tertentu.
- K. Data, digunakan untuk menghubungkan dengan sebuah database dan menampilkan informasinya pada form.

## 6. Project Explorer

Jendela Project Explorer adalah jendela yang mengandung semua file di dalam aplikasi Visual Basic. Setiap aplikasi dalam Visual Basic disebut dengan istilah *project* (proyek), dan setiap proyek bisa mengandung lebih dari satu file. Pada Project Explorer ditampilkan semua file yang terdapat pada aplikasi (proyek), misalnya form, modul, class, dan sebagainya.

## 7. Jendela Properties

Jendela Properties adalah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi Visual Basic. Properti adalah sifat dari sebuah objek, misalnya seperti nama, warna, ukuran, posisi, dan sebagainya.

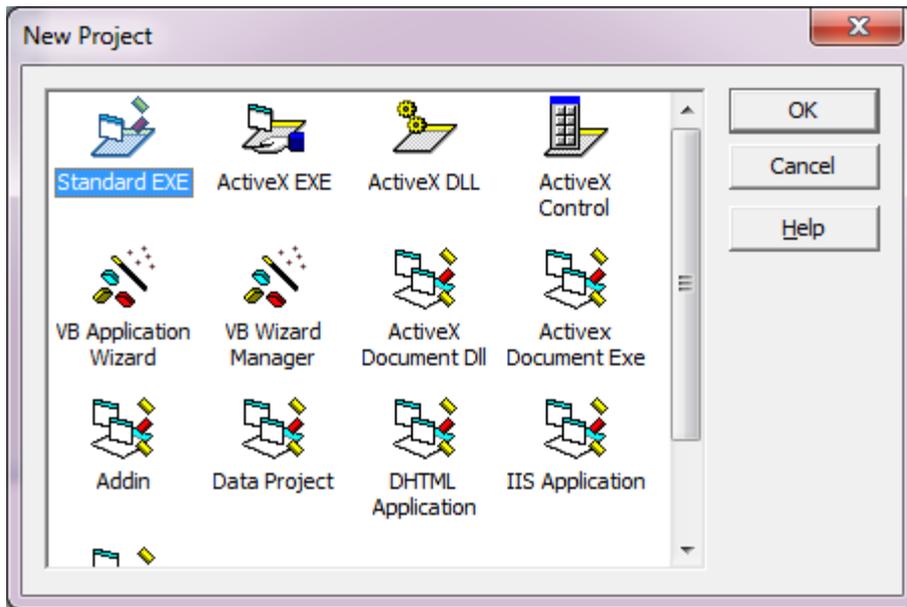
## 8. Form Layout Window

Form Layout Window adalah jendela yang menggambarkan posisi dari form yang ditampilkan pada layer monitor. Posisi *form* pada Form Layout Window inilah yang merupakan petunjuk tempat aplikasi akan ditampilkan pada layar monitor saat dijalankan.

## 9. Jendela Code

Jendela *Code* adalah salah satu jendela yang penting di dalam Visual Basic. Jendela ini berisi kode-kode program yang merupakan instruksi-instruksi untuk aplikasi Visual Basic yang dibuat

## 2.4 Jenis Project Yang Ada Pada Visual Basic 6



Jenis tipe/aplikasi Project yang terdapat pada Visual Basic 6.0

- **Standard EXE:** Tipe aplikasi ini merupakan tipe aplikasi yang biasa dipakai untuk keperluan bisnis yang umum, tanpa harus berhubungan dengan aplikasi yang lain. Project standar dalam Visual Basic dengan komponen-komponen standar. Jenis project ini sangat sederhana, tetapi memiliki keunggulan bahwa semua komponennya dapat diakui oleh semua unit komputer dan semua user meskipun bukan administrator.
- **ActiveX EXE:** Tipe aplikasi yang digunakan untuk aplikasi lain sebagai bagian dari suatu service. Service ini dapat berjalan didalam komputer yang sama ataupun pada komputer lain melalui "remote automation" atau DCOM. Biasanya service ini tidak menyediakan interface visual. Project ini adalah project ActiveX berisi komponen-komponen kemampuan untuk berinteraksi dengan semua aplikasi di sistem operasi windows.
- **ActiveX DLL:** Tipe aplikasi ini biasanya dibuat untuk digunakan oleh aplikasi lain sebagai bagian dari suatu service. DLL terdapat pada komputer yang sama dengan aplikasi yang menjalankannya. Aplikasi ini tidak menyediakan Interface Visual. Tetapi ia dapat juga dijalankan pada komputer yang lain yang menjalankan Microsoft Transaction Server. Project ini menghasilkan sebuah aplikasi library yang selanjutnya dapat digunakan oleh semua aplikasi di sistem operasi windows.
- **ActiveX Control:** Adalah komponen software yang bisa digunakan untuk perangkat-perangkat yang mendukung teknologi ActiveX. Control dari ActiveX ini, dapat ditulis dalam berbagai jenis bahasa pemrograman, termasuk C, C++, Visual Basic dan Java. Project ini menghasilkan komponen-komponen baru untuk aplikasi Visual Basic yang lain
- **VB Application Wizard:** Project ini memandu pengguna untuk membuat aplikasi secara mudah tanpa harus pusing-pusing dengan perintah-perintah pemrograman.
- **Addin:** Project seperti Standard EXE tetapi dengan berbagai macam komponen tambahan yang memungkinkan kebebasan kreasi dari pengguna.

- Data project: Project ini melengkapi komponennya dengan komponen-komponen database. Sehingga bisa dikatakan project ini memang disediakan untuk keperluan pembuatan aplikasi database.
- DHTML Application: Project ini digunakan untuk membuat aplikasi internet pada sisi client (client side) dengan fungsi-fungsi DHTML.
- IIS Application: Project ini menghasilkan aplikasi internet pada sisi server (server side) dengan komponen-komponen CGI (Common Gateway Interface).
- VB Wizard Manager biasanya digunakan untuk membangun sebuah wizard yaitu kumpulan informasi dari pengguna yang akan digunakan untuk membangun sebuah aplikasi.
- ActiveX Document Dll : Project jenis ini dapat membangun sebuah Document Active dengan ekstensi file berupa DLL.
- ActiveX Document Exe : ActiveX Document Exe hampir sama dengan ActiveX Document DLL, namun format file yang dihasilkan berupa ekstensi Exe.
- VB Enterprise Edition Control : VB Enterprise Edition Control pada dasarnya sejenis dengan project Standard EXE, bedanya kalau VB Enterprise Edition Control semua tools VB Enterprise Edition akan di aktifkan

Beberapa contoh ActiveX Control dan kegunaannya :

- DateTimePicker Control. Untuk menampilkan tanggal pada aplikasi dengan tampilan kalender yang mudah dinavigasikan. yang diinput melalui Date TimePicker Control telah memenuhi format penulisan tanggal yang benar.
- Progress Bar Control. Dengan penunjuk progress, aplikasi dapat terhindar dari tampilan yang monoton, yang bisa jadi pengguna akan mengira aplikasi tersebut mengalami hank jika tidak memiliki progress bar.
- DataGrid Control. dapat menampilkan database pada sebuah grid.
- TreeView Control. Berguna untuk menampilkan data dalam bentuk hirarki. Terutama jika data kita memiliki banyak level dan jenjang, maka dengan menggunakan TreView Control, ilustrasi data akan menjadi jelas dan user friendly.
- MsChart Control. menambahkan grafik pada aplikasi dengan berbagai pilihan hanya perlu menyediakan data berupa angka pada aplikasi untuk membentuk sebuah grafik yang diinginkan.

## **2.5 Variabel, Tipe Data dan Operator**

### **1. Variabel**

Variabel adalah nama atau simbol yang digunakan untuk mewakili suatu nilai.

Suatu variabel mempunyai nama dan menyimpan tipe data yang merupakan jenis data variabel.

Aturan penamaan variabel adalah sebagai berikut:

- Harus dimulai dengan sebuah huruf
- Tidak lebih dari 255 karakter
- Tidak boleh sama dengan nama statement, fungsi, metode, objek, dan sebagainya yang merupakan bahasa dari Visual BASIC.
- Tidak boleh ada spasi, tanda titik(.), tanda seru(!), atau karakter @, &, \$, dan #.

Deklarasi variabel dapat dituliskan dengan urutan sebagai berikut:

Public <nama\_variabel> As <Tipe\_Data>

Atau

Dim <nama\_variabel> As <Tipe\_Data>

**Contoh :**

Public Angka1 As Integer

Dim Nama As String

## 2. Tipe Data

Tipe data adalah jenis data yang disimpan dalam variabel. Tipe data untuk Visual BASIC adalah sebagai berikut:

- **Tipe Data Numerik:** digunakan untuk menyimpan data numerik, terdiri dari:
  - Tipe Data Ukuran Range
  - Byte 1 byte 0 sampai 255
  - Integer 2 byte -32.768 sampai 32.767
  - Long 4 byte -2.147.483.648 sampai 2.147.483.647
  - Single 4 byte -3,402823E38 sampai -1,401298E-45;  
1,401298E-45 sampai 3,402823E38
  - Double 8 byte -1.79769313486232E308 sampai -4,94065645841247E-324;  
4,94065645841247E-324 sampai 1.79769313486232E308
  - Currency 8 byte -922.337.203.685.477,5808 sampai 922.337.203.685.477,5807
  - Date : tipe data waktu/tanggal
  - Object : tipe data untuk sebuah objek misalnya gambar
  - Variant : tipe data variant
- **Tipe Data String :** digunakan untuk menyimpan data berbentuk karakter.
  - Panjang maksimal karakter yang dapat disimpan adalah 65.400 karakter.
  - Penulisan data dengan tipe ini diawali dan diakhiri dengan tanda petik dua (“”).

**Contoh:**

Dim Nama As String

Nama = “Habib Maulan”

- **Tipe Data Logika (Boolean)** : melakukan pengetesan logika. Data dengan tipe data ini hanya dapat bernilai benar(*True*) atau salah(*False*).

**Contoh:**

Dim Baru As Boolean

Baru = True

**3. Konstanta**

Konstanta adalah suatu nilai konstan yang tidak berubah. Seperti halnya variabel, konstanta dapat diberi nama dimana aturan penamaannya sama dengan variabel.

**Contoh:**

Const A = 10

**4. Operator**

**Operator Pemberi Nilai**

Deklarasi pemberian nilai pada Visual BASIC = Bahasa BASIC yaitu menggunakan operator sama dengan ( = ).

**Contoh :**

a = 24

nama = "aku"

**Operator Arimatika**

+ Penjumlahan

- Pengurangan

\* Perkalian

/ Pembagian

\ Pembagian dengan hasil bilangan bulat

Mod Sisa Pembagian (Modulus)

**Operator Boolean**

Not Negasi

And Logika and

Or Logika or

Xor Logika xor

**Operator Pembanding**

= Sama dengan

<> Tidak sama dengan

< Kurang dari

> Lebih dari

<= Kurang dari atau sama dengan

>= Lebih dari atau sama dengan

### **Derajat Operator**

#### **Not Tertinggi**

\* / \ mod and

+ - or xor

= <> <= >= **Terendah**

### **PRAKTIKUM 1.1**

Pada praktikum ini anda akan membuat program sederhana untuk menampilkan tiga warna form, menonaktifkan button, menghilangkan button dan membuat button berjalan.



Berikut ini codingnya:

```
Private Sub btaktif_Click()
```

```
BTHILANG.Enabled = True
```

```
BTMUNCUL.Enabled = True
```

```
End Sub
```

```
Private Sub BTHIJAU_Click()
```

```
Form1.BackColor = vbGreen
```

```
End Sub
```

```
Private Sub BTHILANG_Click()
```

```
BTMERAH.Visible = False
```

```
BTHIJAU.Visible = False
```

```
BTKUNING.Visible = False
```

```
End Sub
```

```
Private Sub BTKELUAR_Click()
```

```
End
```

```
End Sub
```

```
Private Sub BTKUNING_Click()
```

```
Form1.BackColor = vbYellow
```

```
End Sub
```

```
Private Sub BTMERAH_Click()
```

```
Form1.BackColor = vbRed
```

```
End Sub
```

```
Private Sub BTMERAH_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
BTMERAH.Left = BTMERAH.Left + 10
```

```
BTMERAH.Top = BTMERAH.Top + 10
```

```
End Sub
```

```
Private Sub BTMUNCUL_Click()
```

```
BTMERAH.Visible = True
```

```
BTHIJAU.Visible = True
```

```
BTKUNING.Visible = True
```

```
End Sub
```

```
Private Sub BTTIDAKAKTIF_Click()
```

```
BTHILANG.Enabled = False
```

```
BTMUNCUL.Enabled = False
```

```
End Sub
```

## **PRAKTIKUM 1.2**

Pada praktikum ini anda akan membuat program sederhana untuk pemakaian variable, type data dan penggunaan Val (untuk mengubah string ke angka/numeric)



**Berikut ini codingnya:**

```
Private Sub BTBAGI_Click()
```

```
Dim A, B, HASIL As Single
```

```
A = Val(Text1.Text)
```

```
B = Val(Text2.Text)
```

```
HASIL = A / B
```

```
LBLHASIL.Caption = HASIL
```

```
End Sub
```

```
Private Sub BTKALI_Click()
```

```
Dim A, B, HASIL As Single
```

```
A = Val(Text1.Text)
```

```
B = Val(Text2.Text)
```

```
HASIL = A * B
```

```
LBLHASIL.Caption = HASIL
```

```
End Sub
```

```
Private Sub BTKURANG_Click()
```

```
Dim A, B, HASIL As Single
```

```
A = Val(Text1.Text)
```

```
B = Val(Text2.Text)
```

```
HASIL = A - B
```

```
LBLHASIL.Caption = HASIL
```

```
End Sub
```

```
Private Sub BTTAMBAH_Click()
```

```
Dim A, B, HASIL As Single
```

```
A = Val(Text1.Text)
```

```
B = Val(Text2.Text)
```

```
HASIL = A + B
```

```
LBLHASIL.Caption = HASIL
```

```
End Sub
```

## **TUGAS**

- 1. Buat program kalkulator sederhana**
- 2. ....**

## **BAB 2**

### **SELEKSI KONDISI**

#### **I. TUJUAN:**

- Melakukan seleksi kondisi dengan statement if...then....else dan select case

#### **II. DASAR TEORI**

##### **Struktur Seleksi / percabangan**

Penyeleksian kondisi atau pernyataan kondisi merupakan suatu pernyataan yang menganalisa suatu keadaan dan mengambil keputusan berdasarkan hasil analisa itu. Hasil dari penyeleksian, Jika benar akan dikerjakan instruksi tertentu, jika salah akan dikerjakan instruksi yang lain.

1. Struktur seleksi sederhana ( if – Then )
2. Struktur seleksi umum ( if – then – else )
3. Struktur seleksi bersarang.
4. Struktur select Case

##### **A. Struktur seleksi sederhana ( if – Then )**

Struktur seleksi ini jika kondisinya benar maka aksi akan dilakukan jika kondisi salah maka proses langsung menuju endif.

- Bentuk penulisannya sebagai berikut:

**If kondisi then**

**Aksi.**

**Endif**

##### **B. Struktur seleksi umum ( if – then – else )**

Struktur seleksi ini jika kondisinya benar maka aksi 1 akan dilakukan jika kondisi salah maka aksi 2 akan dilakukan selanjutnya menuju endif

- Bentuk penulisannya sebagai berikut:

**If kondisi then**

**Aksi 1**

**Else**

**Akse 2**

**Endif**

##### **C. Struktur seleksi Bersarang**

Bentuk struktur ini bentuk yang tidak memiliki struktur khusus. Tergantung dari kasusnya.

#### D. Bentuk case

Bentuk ini dipilih berdasarkan pilihan yang sudah ditetapkan.

Select case pilihan

1: aksi 1

2: aksi 2

3: aksi 3

.....

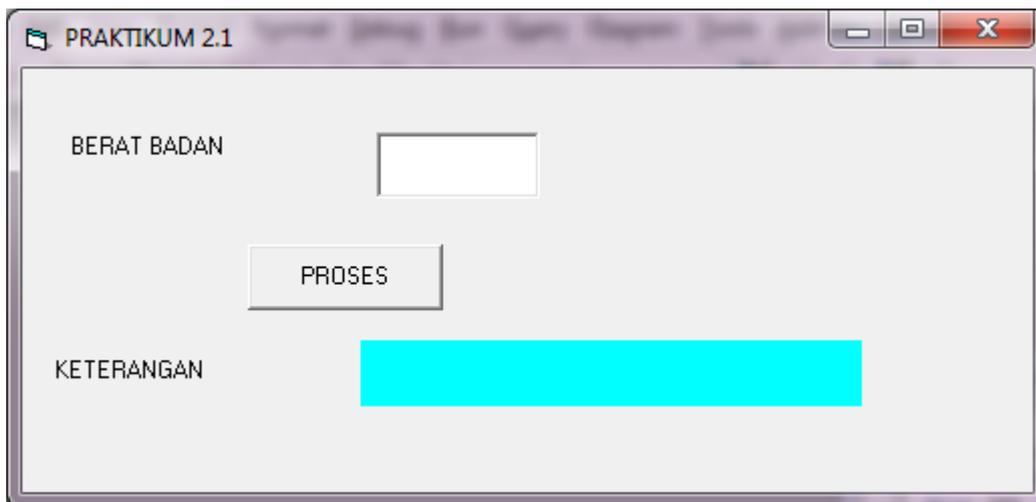
else

aksi 4

end case

#### PRAKTIKUM 2.1

Membuat program sederhana untuk menentukan kriteria berat badan:



```
Private Sub BTPROSES_Click()
```

```
Dim BB As Single
```

```
Dim HASIL As String
```

```
BB = Val(Text1.Text)
```

```
If BB < 7 Then
```

```
HASIL = "ANAK KECIL"
```

```
Else
```

```
HASIL = "REMAJA"
```

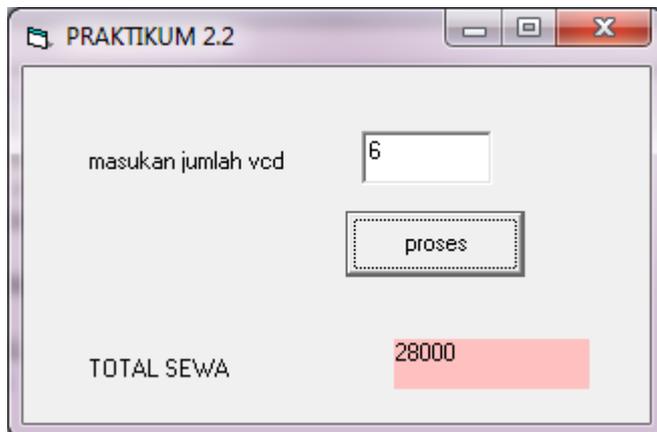
```
End If
```

```
LBLHASIL.Caption = HASIL
```

```
End Sub
```

## PRAKTIKUM 2.2

Program sederhana untuk menghitung besarnya sewa VCD



Berikut codingnya

```
Private Sub Command1_Click()
```

```
Dim JUM, TOTALSEWA As Single
```

```
JUM = Val(Text1.Text)
```

```
If JUM <= 5 Then
```

```
TOTALSEWA = JUM * 5000
```

```
Else
```

```
TOTALSEWA = (25000) + ((JUM - 5) * 3000)
```

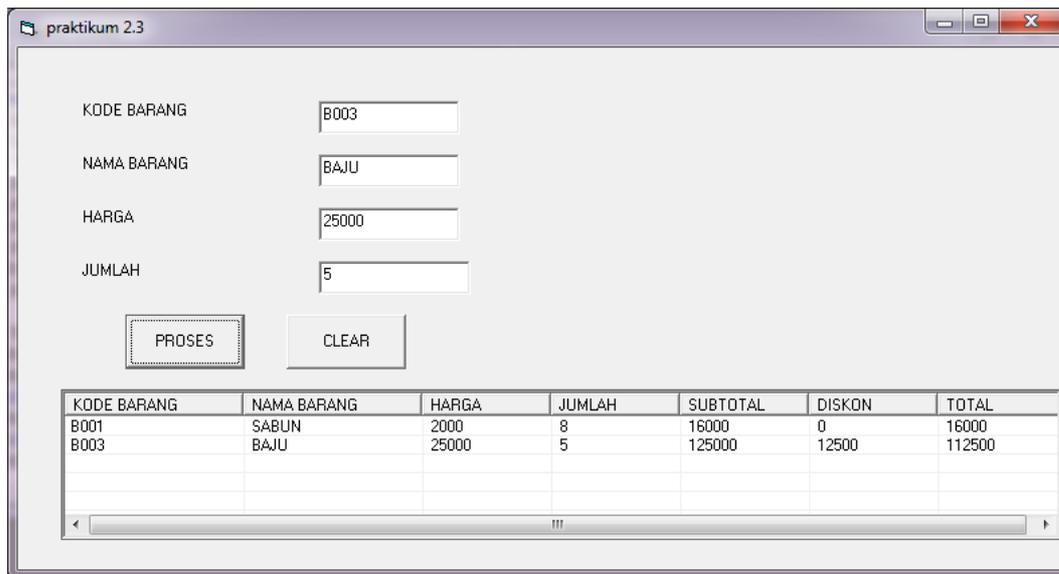
```
End If
```

```
LBLTOTAL.Caption = TOTALSEWA
```

```
End Sub
```

## PRAKTIKUM 2.3

Membuat program sederhana dengan memakai listview



Berikut codingnya:

```
Private Sub BTBERSIH_Click()
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
Text4.Text = ""
```

```
Text1.SetFocus
```

```
End Sub
```

```
Private Sub BTPROSES_Click()
```

```
Dim isi As ListItem
```

```
Set isi = ListView1.ListItems.Add()
```

```
isi.Text = Text1.Text
```

```
isi.SubItems(1) = Text2.Text
```

```
isi.SubItems(2) = Text3.Text
```

```
isi.SubItems(3) = Text4.Text
```

```
Dim SUBTOTAL As Single
```

```
SUBTOTAL = Val(Text3.Text) * Val(Text4.Text)
```

```
isi.SubItems(4) = SUBTOTAL
```

```
Dim TOTAL, DISKON As Single
```

```
If SUBTOTAL < 50000 Then
```

```

DISKON = 0
Elseif SUBTOTAL < 100000 Then
DISKON = 0.05 * SUBTOTAL
Elseif SUBTOTAL < 200000 Then
DISKON = 0.1 * SUBTOTAL
Else
DISKON = 0.2 * SUBTOTAL
End If
TOTAL = SUBTOTAL - DISKON
isi.SubItems(5) = DISKON
isi.SubItems(6) = TOTAL
Set isi = Nothing
End Sub

```

## TUGAS INDIVIDU

1. Buat program yang meminta masukan koordinat suatu titik dan menentukan pada kuadran mana titik tersebut berada. Buat dengan tampilan seperti berikut ini

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two text boxes for input: "SUMBU X" containing the number "5" and "SUMBU Y" containing "-6". Below these is a button labeled "PROSES". Underneath the button, the text "TERLETAK DI" is followed by "KUADRAN 4" which is highlighted in cyan. At the bottom, "JARAK KETITIK PUSAT" is followed by "7,81025" which is highlighted in red.

2. Buat program untuk menentukan kriteria kegemukan digunakan Indeks Massa Tubuh (IMT) dengan rumus

$$IMT = b / T^2$$

b= berat badan (kg)

t=tinggi badan (m)

Kreteria IMT sebagai berikut:

IMT $\leq$ 18.5	>>> kurus
18.5 < IMT $\leq$ 25	>>> normal
25 < IMT $\leq$ 30	>>> gemuk
IMT > 30	>>> kegemukan

## **BAB 3**

### **PERULANGAN**

#### **I. TUJUAN:**

- Mengetahui perulangan yang ada dalam Visual Basic 6.0

#### **II. DASAR TEORI**

Struktur Perulangan pada Visual Basic 6

Pada dasarnya struktur beberapa perulangan pada VB 6, yaitu:

- A. Perulangan Do While – Loop
- B. Perulangan Do Until – Loop
- C. Perulangan While - Wend
- D. Perulangan For – Next
- E. Perulangan For Each - Next

#### **A. Do While – Loop atau Do – Loop While**

Perulangan Do While – Loop adalah perulangan yang akan dijalankan jika kondisi bernilai TRUE (memenuhi syarat) dan akan berhenti pada saat kondisi bernilai FALSE (tidak memenuhi syarat). Struktur kode 1:

```
Do While {kondisi}
{pernyataan}
Loop
```

Struktur kode 2:

```
Do
{pernyataan}
Loop While {kondisi}
```

B. Do Until – Loop atau Do – Loop Until yaitu perulangan yang akan dijalankan jika kondisi bernilai FALSE (tidak memenuhi syarat) dan akan berhenti pada saat kondisi bernilai TRUE (memenuhi syarat).

Struktur kode 1:

```
Do Until {kondisi}
{pernyataan}
Loop
```

Struktur kode 2:

```
Do
{pernyataan}
```

Loop Until {kondisi}

### C. Perulangan While - Wend

Jenis perulangan While-Wend hampir sama dengan Do-While – Loop, hanya beda struktur dalam penulisan kode-nya saja. Struktur kode: While {kondisi} {aksi} Wend Pada struktur perulangan While-Wend, jika kondisi bernilai True, maka aksi akan dijalankan.

### D. Perulangan For - Next

Perulangan For-Next adalah jenis perulangan dengan jumlah perulangan (looping) sudah ditentukan terlebih dahulu (dengan variabel ber-type integer).

Struktur kode:

```
For {penghitung} = {awal} to {akhir}
```

```
{pernyataan}
```

```
Next {penghitung}
```

### E. Perulangan For Each- Next

Perulangan ini adalah perulangan untuk operasi kontrol pada suatu object atau container.

Struktur kode :

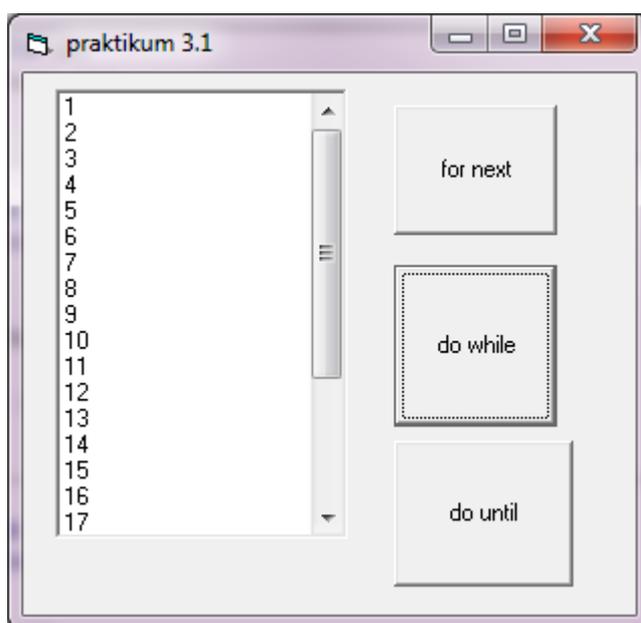
```
For Each {element} In {group}
```

```
{pernyataan}
```

```
Next
```

## PRAKTIKUM 3.1

Pada praktikum ini akan menampilkan bilangan 1 sampai 25 pada



Berikut codingnya

```
Private Sub bbtok_Click()  
Dim i As Integer  
For i = 1 To 25  
List1.AddItem (i)  
Next  
End Sub
```

```
Private Sub btuntil_Click()  
Dim i As Integer  
i = 1  
Do  
List1.AddItem (i)  
i = i + 1  
Loop Until i > 25  
  
End Sub
```

```
Private Sub btwhile_Click()  
Dim i As Integer  
i = 1  
Do While i <= 25  
List1.AddItem (i)  
i = i + 1  
Loop  
End Sub
```

### **PRAKTIKUM 3.2**

Modifikasi praktikum 3.1 sehingga bisa menampilkan bilangan ganjil dan bilangan genap

### **PRAKTIKUM 3.3**

Modifikasi praktikum 3.2 sehingga bisa menampilkan kata **SELAMAT DATANG**. Dengan tampilan sebagai berikut

**S**

**SE**

**SEL**

**SELA**

**SELAM**

**SELAMA**

**SELAMAT..... dan seterusnya**

**TUGAS INDIVIDU**

1. Buat program untuk menampilkan bilangan prima kurang dari 100
2. Buat program bebas sesuai aplikasi perulangan

## BAB 4 TIMER DAN MENU

### I. TUJUAN:

- Mengenal timer dalam Visual Basic 6.0
- Mengenal menu dalam VB 6.0

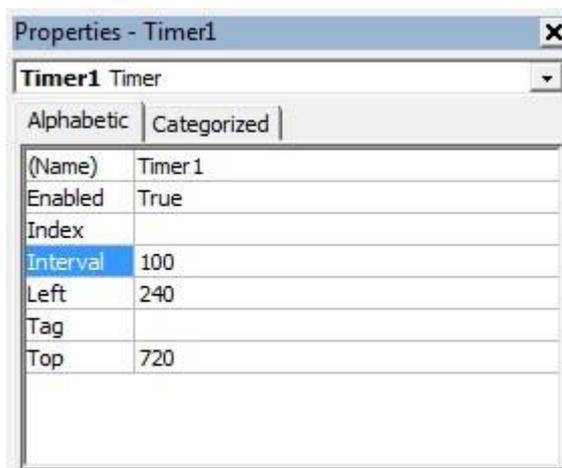
### II. DASAR TEORI

**Timer** pada visual basic 6.0 biasanya digunakan untuk menggerakkan data atau menghitung data secara otomatis atau membuat jam yang berjalan sesuai dengan jam yang ada dikomputer atau laptop, intinya timer digunakan sebagai pengatur waktu.

Dalam menggunakan komponen timer berkaitan dengan waktu,timer juga dapat digunakan untuk menjadikan tampilan kita lebih dinamis,dan sangat membantu dalam program yang akan dibuat.

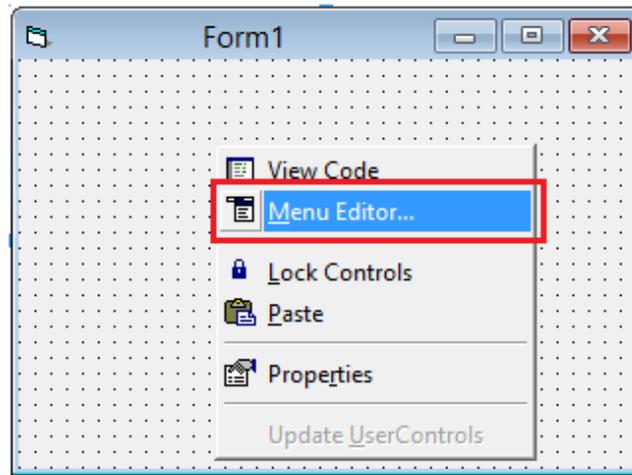
#### **Berikut Adalah Properti Penting Dari Kontrol Timer**

1. **Enable** Berfungsi Apabila True Maka Kontrol Timer Mulai Bekerja Dan apabila False Maka Kontrol Timer Berhenti. Dan Nilai defaultnya adalah True
2. **Interval** Adalah Menunjukkan Angka Milidetik (1/1000 Detik) Untuk selang waktu antar event kontrol.



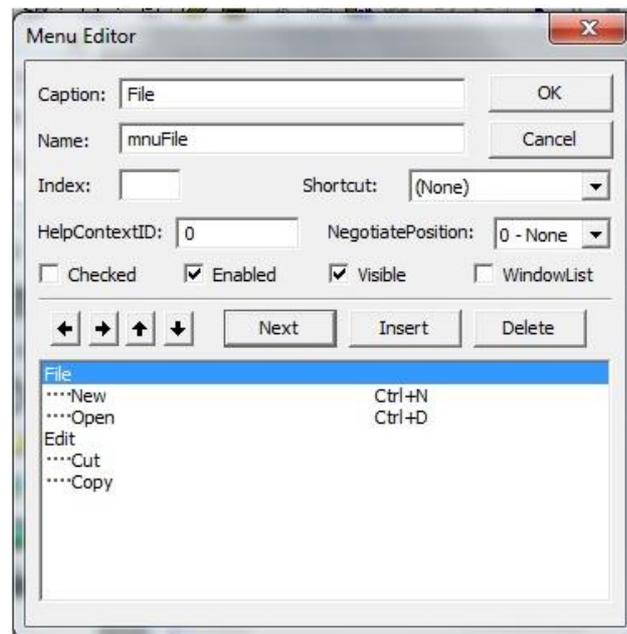
Menu adalah salah satu fitur yang penting dalam sebuah produk software. Setiap software standard memiliki menu. Secara umum kita biasa melihat menu di bagian atas tampilan software. Menu merupakan control tapi terpisah dari control di toolbox dan tidak bekerja seperti control biasa. Kita menambahkan menu ke form dengan menggunakan window Menu Editor. Pada form aktif tekan tombol Ctrl+E untuk menampilkan menu editor, atau

klik  pada toolbar, atau klik kanan -> Menu Editor.



### 1. Membuat Menu

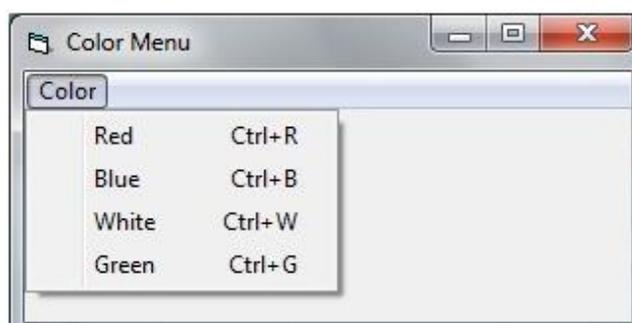
Kita dapat membuat menu dengan cara yang sangat mudah, cukup isi caption dan nama pada window Menu Editor kemudian klik OK.



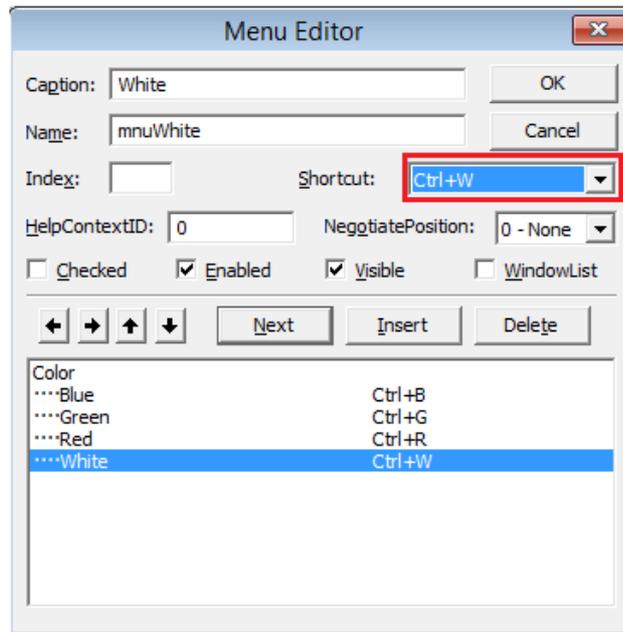
Cukup tekan panah kanan pada keyboard untuk membuat sub menu item. Klik tombol Next untuk membuat menu berikutnya. Lalu klik OK setelah selesai.

### 2. Menu pada Form

Kita akan coba membuat menu dengan UI design sebagai berikut:



Ctrl+R, Ctrl+B, Ctrl+W, dan Ctrl+G merupakan shortcut yang diatur pada bagian Shortcut di Menu Editor.



#### PRAKTIKUM 4.1

Pada praktikum ini akan menampilkan tanggal dan jam. Serta aplikasi bola berjalan.



Berikut codingnya

```
Dim ketemu As Boolean
```

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```

```

Private Sub Timer1_Timer()
LBJAM.Caption = Format(Now, "hh:mm:ss")
LBLTGL.Caption = Format(Now, "dd/MM/yyyy")
End Sub

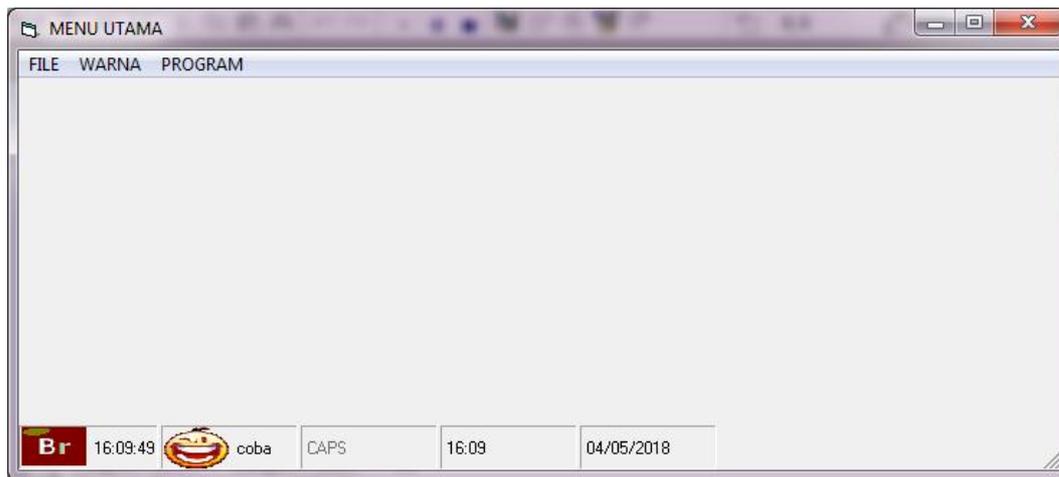
Private Sub Timer2_Timer()
If ketemu Then
If Shape1.Left < 5000 Then
Shape1.Left = Shape1.Left + 50
Shape1.BackColor = vbBlue
Else
ketemu = False
End If
End If

If Not ketemu Then
If Shape1.Left > 0 Then
Shape1.Left = Shape1.Left - 50
Shape1.BackColor = vbYellow
Else
ketemu = True
End If
End If
End Sub

```

## **PRAKTIKUM 4.2**

Pada praktikum ini akan kita buat menubar dan statusbar



Tugas individu

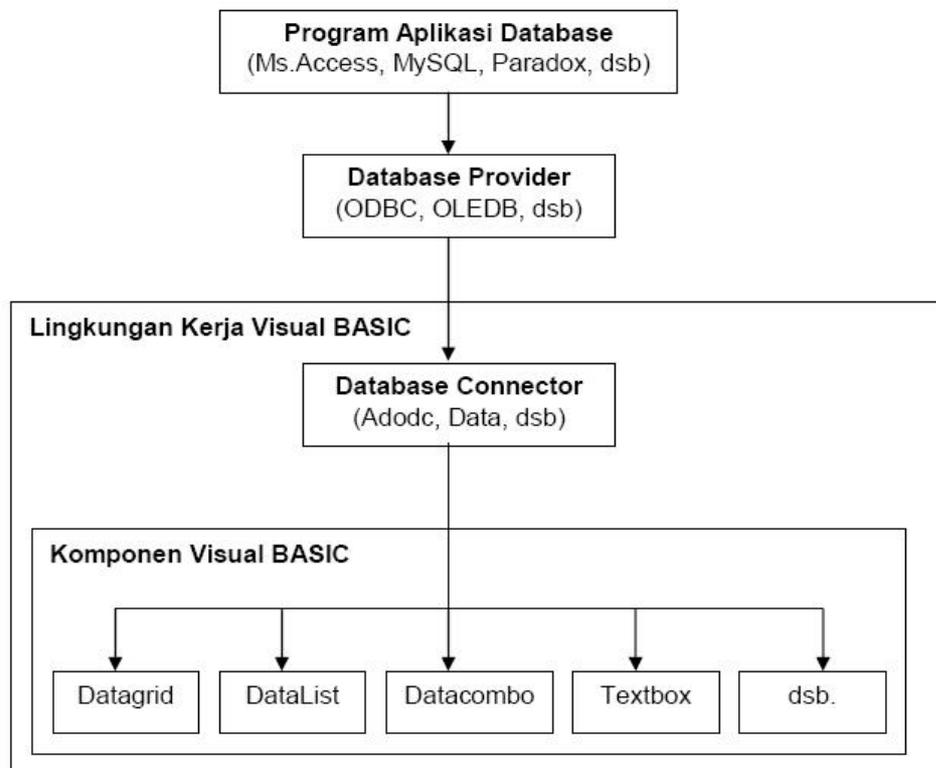
1. buat program aplikasi bola berputar
2. Buat program aplikasi bola bekel yang di jatuhkan dari atas

## BAB 5 PEMROGRAMAN DATABASE 1

### A. KONSEP DASAR

Pemrograman Database (Database Programming) merupakan suatu bentuk pemrograman alternatif untuk pengolahan database. Dengan pemrograman database kita dapat secara leluasa mengatur tampilan dan alur kerja sebuah database dengan lebih baik. Visual BASIC merupakan salah satu bahasa pemrograman yang telah mendukung pemrograman database. Visual BASIC dapat dihubungkan dengan program aplikasi pengolah data lain seperti Access, MySQL, SQL Server dan sebagainya.

Alur kerja pemrograman database dalam Visual BASIC dapat dijelaskan berikut.



### B. Koneksi Visual BASIC dengan Database Access

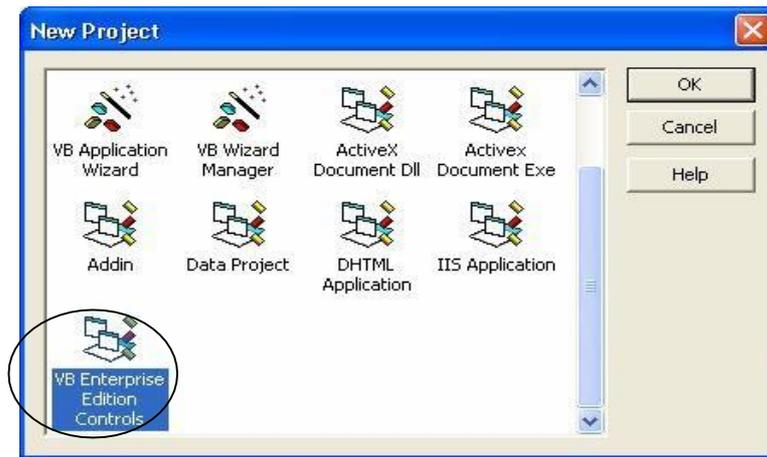
Untuk dapat menghubungkan Visual BASIC dengan database Access, kita akan menggunakan komponen ADO Data Control (ADODC). Komponen ini dapat dihubungkan dengan beberapa komponen yang digunakan untuk mengakses data seperti textbox, datagrid, dsb. Untuk dapat menggunakan ADODC, ada beberapa properti yang harus kita lakukan pengaturan antara lain:

Properti	Fungsi
<i>ConnectionString</i>	Untuk mendefinisikan database yang akan digunakan Adodc.
<i>RecordSource</i>	Untuk mendefinisikan tabel/query yang akan digunakan Adodc.

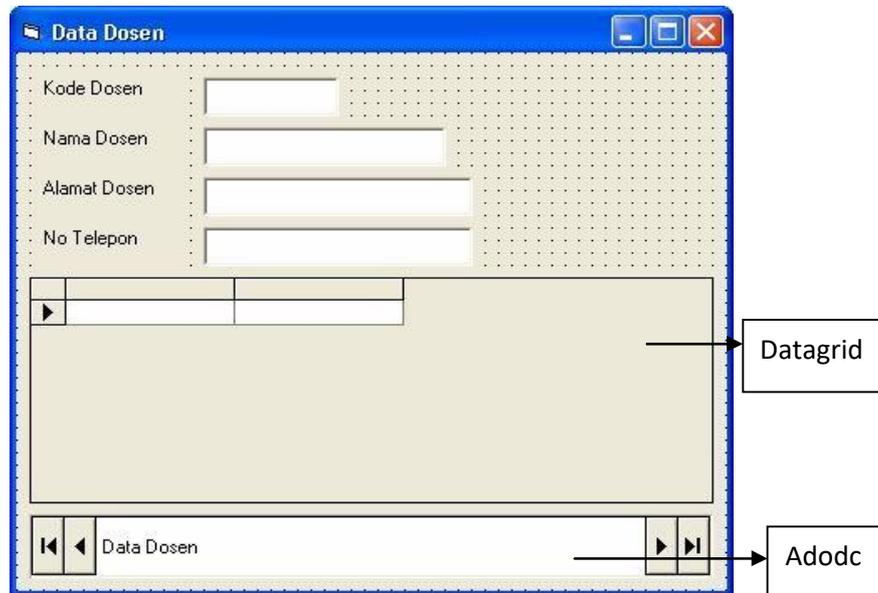
<i>CommandType</i>	Untuk mendefinisikan perintah yang digunakan Adodc.
--------------------	---

Untuk lebih jelasnya mencoba membuat sebuah program untuk memprektekkan cara kerja ADODC. Pada program ini akan coba dihubungkan database pengajaran (tabel dosen) yang telah dibuat.

Buatlah sebuah project baru dengan tipe project VB Enterprise Edition Controls, seperti gambar.



Desain tampilan form sehingga akan dihasilkan tampilan seperti pada gambar



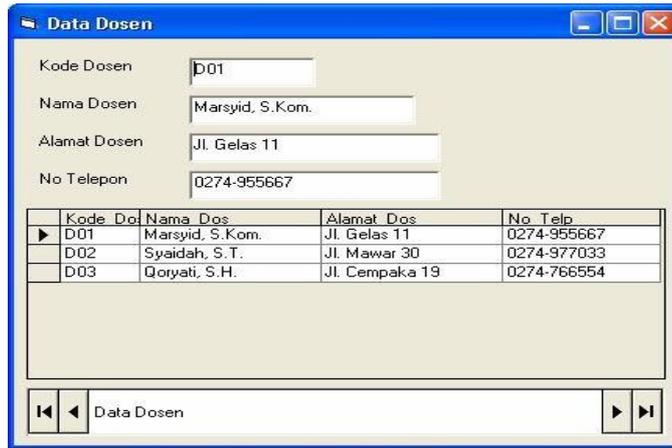
**Pengaturan Properti :**

Komponen	Properti	Nilai
Form1	(Name) Caption	frmdosen Data Dosen
Adodc1	(Name) Caption ConnectionString RecordSource	Adodosen Data Dosen Pengajaran.mdb Dosen
Label1	Caption	Kode Dosen
Label2	Caption	Nama Dosen
Label3	Caption	Alamat Dosen
Label4	Caption	No Telepon
Textbox1	(Name) Text DataSource Datafield	TxtKode (kosongkan) Adodosen (pilihan) Kode_Dos (pilihan)
Textbox2	(Name) Text DataSource Datafield	TxtNama (kosongkan) Adodosen (pilihan) Nama_Dos (pilihan)
Textbox3	(Name) Text DataSource Datafield	TxtAlamat (kosongkan) Adodosen (pilihan) Alamat_Dos (pilihan)
Textbox4	(Name) Text DataSource Datafield	TxtTelpon (kosongkan) Adodosen (pilihan) No_Telp (pilihan)
Datagrid1	DataSource	Adodosen

**Pengaturan ConnectionString pada Adodc :**

- Klik komponen Adodc pada form, pada bagian properties aktifkan pada pilihan property **ConnectionString**, kemudian klik tanda . Pada Jendela *Property Page* yang muncul klik tombol **Build...**
- Kemudian akan ditampilkan jendela *Data Link Properties*, Pilih **Microsoft JET 4.0 OLE DB Provider**, Kemudian klik tombol **Next >>**
- Pada bagian *Select or enter a database name*, Klik tombol, pilih lokasi penyimpanan database pengajaran, pilih database pengajaran, kemudian klik Open, klik tombol **Test Connection**, untuk mengetes koneksi, jika telah sukses klik tombol **OK**.
- Kemudian akan ditampilkan kembali jendela *Property Page*, klik tombol Apply, kemudian klik OK

Jalankan program sehingga dihasilkan tampilan seperti pada gambar



Untuk berpindah antar record, dapat dilakukan dengan menekan tombol navigasi pada Adodc.

Simpan Project dengan nama **AksesData.vbp**.

**Latihan :**

Tambahkan 2 buah form baru pada project diatas untuk mengakses tabel matakuliah (nama form : **frmMK**) dan tabel kuliah (nama form : **frmKuliah**) menggunakan cara diatas.

## PEMROGRAMAN DATABASE II

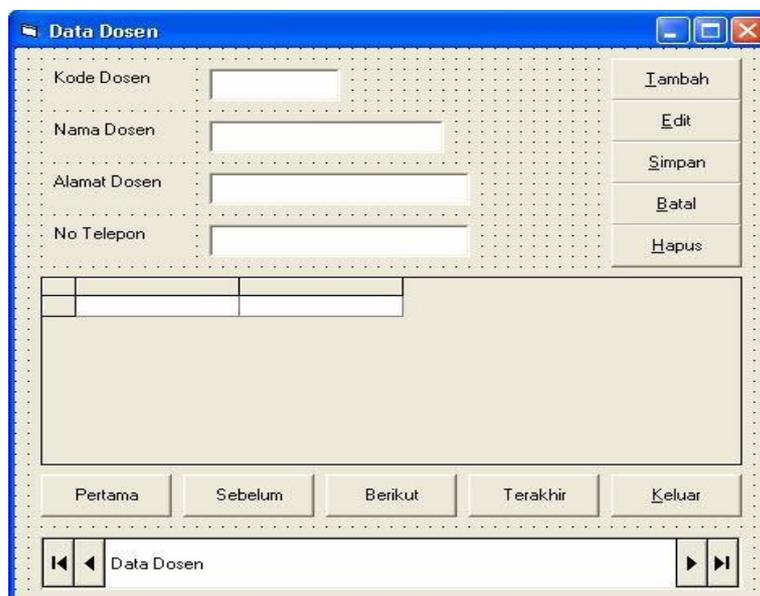
Pada pemrograman database, terdapat beberapa perintah khusus (*method*) yang dimiliki beberapa komponen untuk melakukan pemrosesan data dalam database. Sebagai contoh, jika kita menggunakan komponen **Adodc** sebagai komponen untuk koneksi ke database, terdapat beberapa method yang sering digunakan dalam pengaksesan database antara lain:

Perintah	Fungsi
<Adodc>.Recordset.MoveFirst	Berpindah ke record pertama
<Adodc>.Recordset.MovePrevious	Berpindah ke record sebelumnya
<Adodc>.Recordset.MoveNext	Berpindah ke record berikutnya
<Adodc>.Recordset.MoveLast	Berpindah ke record terakhir
<Adodc>.Recordset.AddNew	Menambahkan record baru
<Adodc>.Recordset.Update	Menyimpan perubahan/penambahan data
<Adodc>.Recordset.Cancel	Membatalkan perubahan/penambahan data
<Adodc>.Recordset.Delete	Menghapus data
<Adodc>.Recordset.Field("NamaField")	Mengakses field tertentu dalam tabel
<Adodc>.Recordset.Field!NamaField	Mengakses field tertentu dalam tabel
<Adodc>.Refresh	Menampilkan ulang data dalam tabel.

Untuk lebih memahami pemrograman database, kita akan membuat sebuah program database sederhana dengan menggunakan database pengajaran (tabel : dosen).

## Program Data Dosen

Buatlah sebuah project dan form baru pada Visual BASIC, desain tampilan form sehingga didapat tampilan seperti pada gambar.



### Pengaturan Properti :

Komponen	Properti	Nilai
Form1	(Name) Caption	frmdosen Data Dosen
Adodc1	(Name) Caption	Adodosen Data Dosen
Label1	Caption	Kode Dosen
Label2	Caption	Nama Dosen
Label3	Caption	Alamat Dosen
Label4	Caption	No Telepon
Textbox1	(Name) Text	TxtKode (kosongkan)
Textbox2	(Name) Text	TxtNama (kosongkan)
Textbox3	(Name) Text	TxtAlamat (kosongkan)
Textbox4	(Name) Text	TxtTelpon (kosongkan)
Datagrid1	DataSource	Adodosen
Command1	(Name) Caption	Cmdpertama Pertama
Command2	(Name) Caption	Cmdsebelum Sebelum
Command3	(Name) Caption	Cmdberikut Berikut
Command4	(Name) Caption	Cmdterakhir Terakhir
Command5	(Name) Caption	Cmdtambah &Tambah

Command6	(Name) Caption	Cmdedit &Edit
Command7	(Name) Caption	Cmdsimpan &Simpan
Command8	(Name) Caption	Cmdbatal &Batal

**Listing Program :**

```
Public Baru As Boolean
```

```
Private Sub cmdbatal_Click()
```

```
    Tombol True, True, False, False, True
    Adodosen.Recordset.Cancel
    Kosong
```

```
End Sub
```

```
Private Sub cmdberikut_Click()
```

```
    'Menuju ke record berikutnya
    Adodosen.Recordset.MoveNext
    'Jika berada di record terakhir menuju ke record terakhir
    If Adodosen.Recordset.EOF Then
        Adodosen.Recordset.MoveLast
    End If
```

```
End Sub
```

```
Private Sub cmdedit_Click()
```

```
    Tombol False, False, True, True, False
    With Adodosen.Recordset
        txtkode.Text = !Kode_Dos
        txtnama.Text = !Nama_Dos
        txtalamat.Text = !Alamat_Dos
        txttelpon.Text = !No_Telp
    End With
    txtkode.SetFocus
    Baru = False
```

```
End Sub
```

```
Private Sub cmdhapus_Click()
```

```
    Dim hapus
    Hapus = MsgBox("Anda yakin data ini vbYesNo, "Hapus Data")
    If hapus = vbYes Then
        Adodosen.Recordset.Delete
        Adodosen.Recordset.MoveLast
    Else
        MsgBox "Data tidak jadi dihapus!", vbOKOnly + vbInformation,
            "Batal Menghapus"
    End If
```

```
End Sub
```

```
Private Sub cmdkeluar_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdpertama_Click()
```

```
    'Menuju ke record pertama
    Adodosen.Recordset.MoveFirst
```

```
End Sub
```

```

Private Sub cmdsebelum_Click()
    'Menuju ke record sebelumnya
    Adodosen.Recordset.MovePrevious
    'Jika berada di record pertama menuju ke record pertama
    If Adodosen.Recordset.BOF Then
        Adodosen.Recordset.MoveFirst
    End If
End Sub

Private Sub cmdsimpan_Click()
    Tombol True, True, False, False, True
    With Adodosen.Recordset
    If Baru Then .AddNew
        !Kode_Dos = txtkode.Text
        !Nama_Dos = txtnama.Text
        !Alamat_Dos = txtalamat.Text
        !No_Telp = txttelpon.Text
        .Update
        .Sort = "Kode_Dos"
    End With
    Kosong
End Sub

Private Sub cmdtambah_Click()
    Tombol False, False, True, True, False
    Baru = True
    Kosong
    txtkode.SetFocus
End Sub

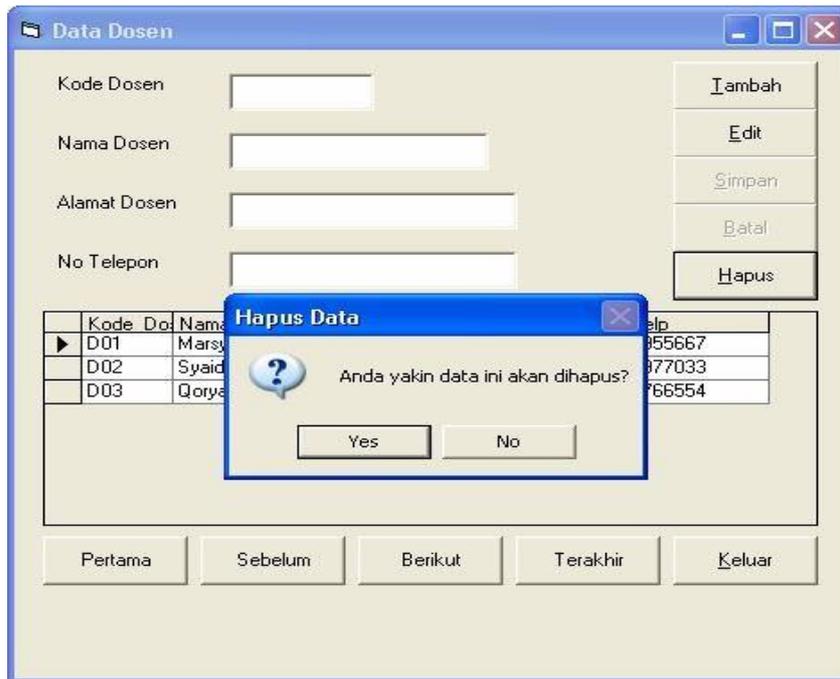
Private Sub cmdterakhir_Click()
    Adodosen.Recordset.MoveLast
End Sub

Public Sub Tombol(tambah, edit, simpan, batal, hapus As Boolean)
    cmdtambah.Enabled = tambah
    cmdedit.Enabled = edit
    cmdsimpan.Enabled = simpan
    cmdbatal.Enabled = batal
    cmdhapus.Enabled = hapus
End Sub

Private Sub Form_Load()
    Adodosen.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
    Source=" & App.Path & "\Pengajaran.mdb"
    Adodosen.RecordSource = "Dosen"
    Adodosen.Refresh
    Adodosen.Recordset.Sort = "Kode_Dos"
    Baru = False
End Sub

Public Sub Kosong()
    txtkode.Text = ""
    txtnama.Text = ""
    txtalamat.Text = ""
    txttelpon.Text = ""
End Sub

```



## PEMROGRAMAN DATABASE DENGAN SQL

Structured Query Language (SQL) adalah bahasa standard untuk melakukan berbagai operasi data pada database, diantaranya mendefinisikan tabel, menampilkan data dengan kriteria tertentu, menambahkan data hingga menghapus data tertentu. Penggunaan SQL pada beberapa bahasa pemrograman secara umum relatif sama. Berikut akan dijelaskan beberapa perintah SQL yang sering digunakan dalam operasi pemrograman database.

### 1. SELECT

Perintah SELECT digunakan untuk menampilkan data.

#### Bentuk Umum:

```
SELECT *|<nama_field> FROM <nama_tabel> [WHERE
    <syarat>] [ORDER BY <nama_field> DESC|ASC]
```

Klausula WHERE digunakan untuk membatasi data yang akan ditampilkan dengan menambahkan syarat tertentu pada data yang ingin ditampilkan. Klausula ORDER BY digunakan untuk menampilkan data dengan urutan menaik (ASC) atau urutan menurun (DESC).

#### Contoh :

‘menampilkan seluruh data dosen

```
SELECT * FROM Dosen
```

'menampilkan kode dosen dan nama dosen dari data dosen

```
SELECT Kode_Dos, Nama_Dos FROM Dosen
```

'menampilkan data dosen diurutkan berdasarkan nama secara menaik

```
SELECT * FROM Dosen ORDER BY Nama_Dos ASC
```

'menampilkan data dosen yang namanya diawali huruf M

```
SELECT * FROM Dosen Where Nama_Dos Like 'M%'
```

## 2. INSERT

Perintah INSERT digunakan untuk menambahkan data kedalam tabel.

**Bentuk Umum:**

```
INSERT INTO <nama_tabel> VALUES (<data_field1>,<data_field2>,...)
```

**Contoh :**

'menambahkan data dosen baru

```
INSERT INTO Dosen VALUES ('D05','Fatah Adi','Jl.Solo 100','799811')
```

## UPDATE

Perintah UPDATE digunakan untuk mengubah data pada suatu tabel dengan kriteria tertentu

```
UPDATE <nama_tabel> SET <field1>=<data1>,  
<field2>=<data2>,... [WHERE <syarat>]
```

**Bentuk Umum:**

**Contoh:**

' mengubah data nama dosen dengan Kode Dosen 'D05' menjadi 'Fatah Ali'

```
UPDATE Dosen SET Nama_Dos='Fatah Ali' WHERE Kode_Dos='D05'
```

## 3. DELETE

Perintah DELETE digunakan untuk menghapus data yang memenuhi kriteria tertentu.

```
DELETE FROM <nama_tabel> [WHERE <syarat>]
```

Bentuk Umum:

Contoh :

'menghapus data dosen dengan nama 'Fatah Ali'

```
DELETE FROM Dosen WHERE Nama_Dos='Fatah Ali'
```

'menghapus seluruh data dalam tabel

```
DELETE FROM Dosen
```

### Program Data Dosen dengan SQL

Buatlah sebuah project dan form baru pada Visual BASIC, desain tampilan form sehingga didapat tampilan seperti pada gambar berikut.

Pengaturan Properti :

Komponen	Properti	Nilai
Form1	(Name) Caption	frmdosen Data Dosen
Adodc1	(Name) Caption	Adodosen Data Dosen
Label1	Caption	Kode Dosen
Label2	Caption	Nama Dosen
Label3	Caption	Alamat Dosen

Label4	Caption	No Telepon
Textbox1	(Name) Text	TxtKode (kosongkan)
Textbox2	(Name) Text	TxtNama (kosongkan)
Textbox3	(Name) Text	TxtAlamat (kosongkan)
Textbox4	(Name) Text	TxtTelpon (kosongkan)
Datagrid1	DataSource	Adodosen
Command1	(Name) Caption	Cmdpertama Pertama
Command2	(Name) Caption	Cmdsebelum Sebelum
Command3	(Name) Caption	Cmdberikut Berikut
Command4	(Name) Caption	Cmdterakhir Terakhir
Command5	(Name) Caption	Cmdtambah &Tambah
Command6	(Name) Caption	Cmdedit &Edit
Command7	(Name) Caption	Cmdsimpan &Simpan
Command8	(Name) Caption	Cmdbatal &Batal
Command9	(Name) Caption	Cmdhapus &Hapus
Command10	(Name) Caption	Cmdkeluar &Keluar
Command11	(Name) Caption	CmdCari &Cari Data
Command12	(Name) Caption	CmdTampil T&ampilkan Semua Data

#### Listing Program :

```
Public Baru As Boolean
```

```
Private Sub cmdbatal_Click()
```

```
    Tombol True, True, False, False, True
    Adodosen.Recordset.Cancel
    Kosong
```

```
End Sub
```

```
Private Sub cmdberikut_Click()
```

```
    'Menuju ke record berikutnya
    Adodosen.Recordset.MoveNext
    'Jika berada di record terakhir menuju ke record terakhir
    If Adodosen.Recordset.EOF Then
        Adodosen.Recordset.MoveLast
    End If
```

```
End Sub
```

```
Private Sub cmdcari_Click()
```

```
    Kriteria = InputBox("Masukkan Nama Dosen yang dicari", "Mencari Data")
    Adodosen.RecordSource = "SELECT * FROM Dosen Where Nama_Dos Like'" &
    "%" & Kriteria & "%" & "'"
    Adodosen.Refresh
```

```

    If Adodosen.Recordset.EOF Then
        MsgBox "Data Tidak Ditemukan!", vbCritical, "Data Tidak Ada"
    End If
End Sub

Private Sub cmdedit_Click()
    Tombol False, False, True, False
    With Adodosen.Recordset
        txtkode.Text = .Fields("Kode_Dos")
        txtnama.Text = .Fields("Nama_Dos")
        txtalamat.Text = .Fields("Alamat_Dos")
        txttelpon.Text = .Fields("No_Telp")
    End With
    txtkode.SetFocus
    Baru = False
End Sub

Private Sub cmdhapus_Click()
    Dim hapus
    hapus = MsgBox("Anda yakin data ini akan dihapus?", vbQuestion +
vbYesNo, "Hapus Data")
    If hapus = vbYes Then
        Adodosen.Recordset.Delete
        Adodosen.Recordset.MoveLast
    Else
        MsgBox "Data tidak jadi dihapus!", vbOKOnly + vbInformation,
        "Batal Menghapus"
    End If
End Sub

Private Sub cmdkeluar_Click()
    Unload Me
End Sub

Private Sub cmdpertama_Click()
    'Menuju ke record pertama
    Adodosen.Recordset.MoveFirst
End Sub

Private Sub cmdsebelum_Click()
    'Menuju ke record sebelumnya
    Adodosen.Recordset.MovePrevious
    'Jika berada di record pertama menuju ke record pertama
    If Adodosen.Recordset.BOF Then
        Adodosen.Recordset.MoveFirst
    End If
End Sub

Private Sub cmdsimpan_Click()
    Tombol True, True, False, False, True
    With Adodosen.Recordset
        If Baru Then .AddNew
            .Fields("Kode_Dos") = txtkode.Text
            .Fields("Nama_Dos") = txtnama.Text
            .Fields("Alamat_Dos") = txtalamat.Text
            .Fields("No_Telp") = txttelpon.Text
            .Update
            .Sort = "Kode_Dos"
        End If
    End With
    Kosong
End Sub

```

```

Private Sub cmdtambah_Click()
    Tombol False, False, True, True, False
    Baru = True
    Kosong
    txtkode.SetFocus
End Sub

Private Sub cmdtampil_Click()
    Adodosen.RecordSource = "SELECT * FROM Dosen Order By Kode_Dos"
    Adodosen.Refresh
End Sub

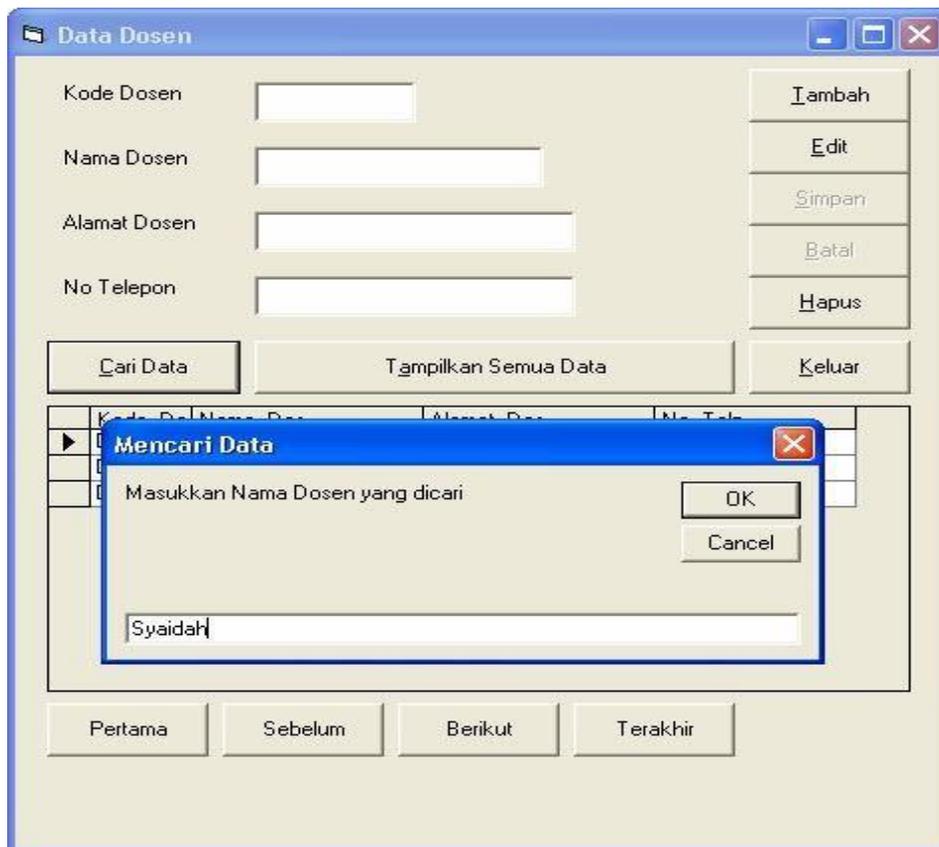
Private Sub cmdterakhir_Click()
    Adodosen.Recordset.MoveLast
End Sub

Public Sub Tombol(tambah, edit, simpan, batal, hapus As Boolean)
    cmdtambah.Enabled = tambah
    cmdedit.Enabled = edit
    cmdsimpan.Enabled = simpan
    cmdbatal.Enabled = batal
    cmdhapus.Enabled = hapus
End Sub

Private Sub Form_Load()
    Adodosen.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
    Source=" & App.Path & "\Pengajaran.mdb"
    Adodosen.CommandType = adCmdText
    Adodosen.RecordSource = "SELECT * FROM DOSEN ORDER BY Kode_Dos"
    Adodosen.Refresh
    Baru = False
End Sub

Public Sub Kosong()
    txtkode.Text = ""
    txtnama.Text = ""
    txtalamat.Text = ""
    txttelpon.Text = ""
End Sub

```

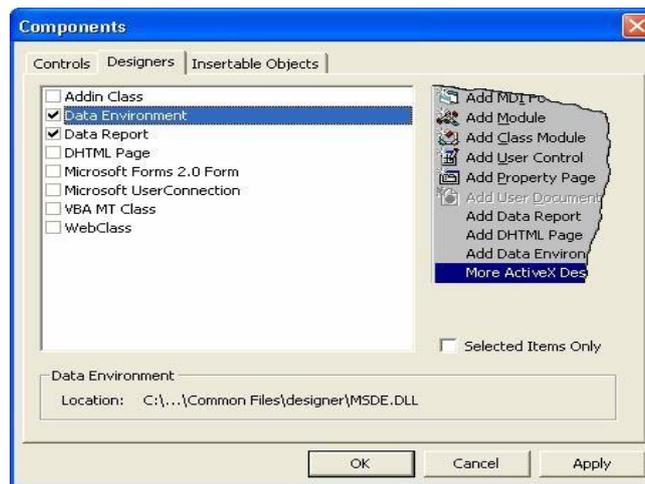


## LAPORAN

Salah satu komponen penting dalam pengolahan database adalah membuat laporan/*report*. Sebuah laporan tidak hanya dapat ditampilkan dalam bentuk tampilan layar(*soft copy*) tetapi juga harus dapat dicetak kedalam media output lain, seperti kertas (*hard copy*). Ada banyak program aplikasi yang dapat membantu kita dalam pembuatan laporan seperti Crystal Report, Quick Report, dsb. Namun sebenarnya dalam Visual BASIC telah disediakan komponen untuk pembuatan laporan pada program, yaitu komponen *Data Report*. Komponen *Data Report* digunakan untuk mendesain tampilan laporan, sedangkan untuk sumber datanya data report membutuhkan komponen pengakses database, yaitu *Data Environment*.

### 1. Menampilkan Pilihan Data Report dan Data Environment.

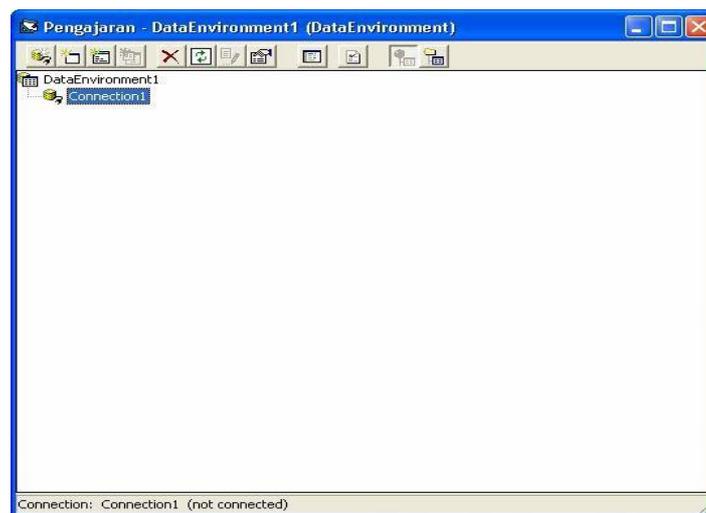
Walaupun pada awal instalasi Visual BASIC, komponen Data Report dan Data Environment telah diikutsertakan, namun ada kalanya kedua komponen ini belum ditampilkan secara otomatis pada menu project. Untuk menampilkannya dapat dilakukan dengan klik Menu Project – Components.



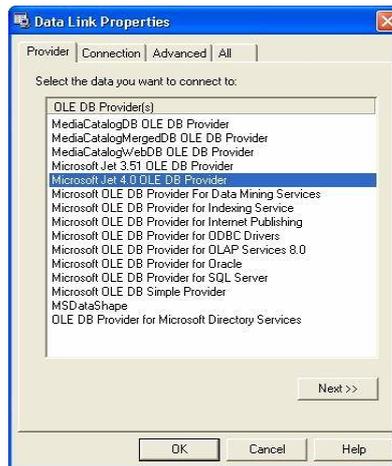
Pada jendela Components Gambar di atas, pilih Tab Designer. Berikan checklist pada Data Environment dan Data Report. Kemudian klik Apply, OK.

## 2. Menambahkan Data Environment

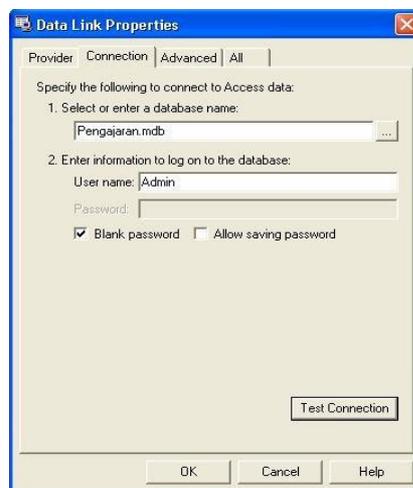
Untuk dapat membuat sebuah laporan, pertama kali kita harus menambahkan sebuah Data Environment yang berfungsi mendefinisikan database dan tabel yang akan ditampilkan pada laporan. Data Environment dapat ditambahkan dengan cara klik menu **Project – Add Data Environment**. Selanjutnya akan tampil jendela seperti pada Gambar berikut.



Untuk menghubungkan Data Environment dengan database pengajaran. Lakukan dengan cara klik kanan **Connection1 – Pilih Properties**. Selanjutnya akan ditampilkan jendela **Data Link Properties** seperti gambar berikut

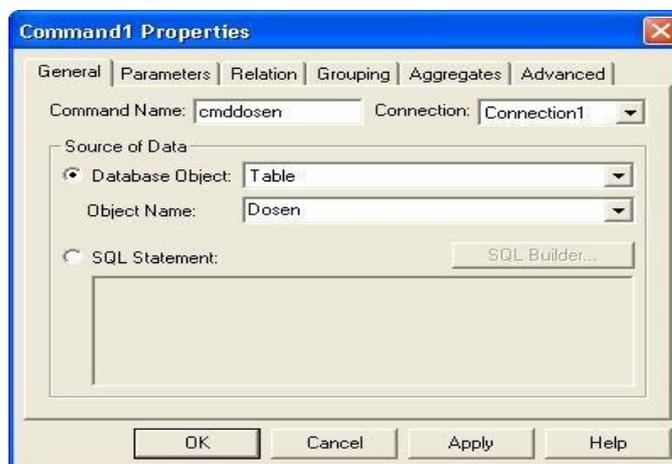


Pilih **Microsoft Jet 4.0 OLE DB Provider**, kemudian Klik **Next >>**, sehingga akan ditampilkan Tab **Connection** seperti pada gambar berikut :

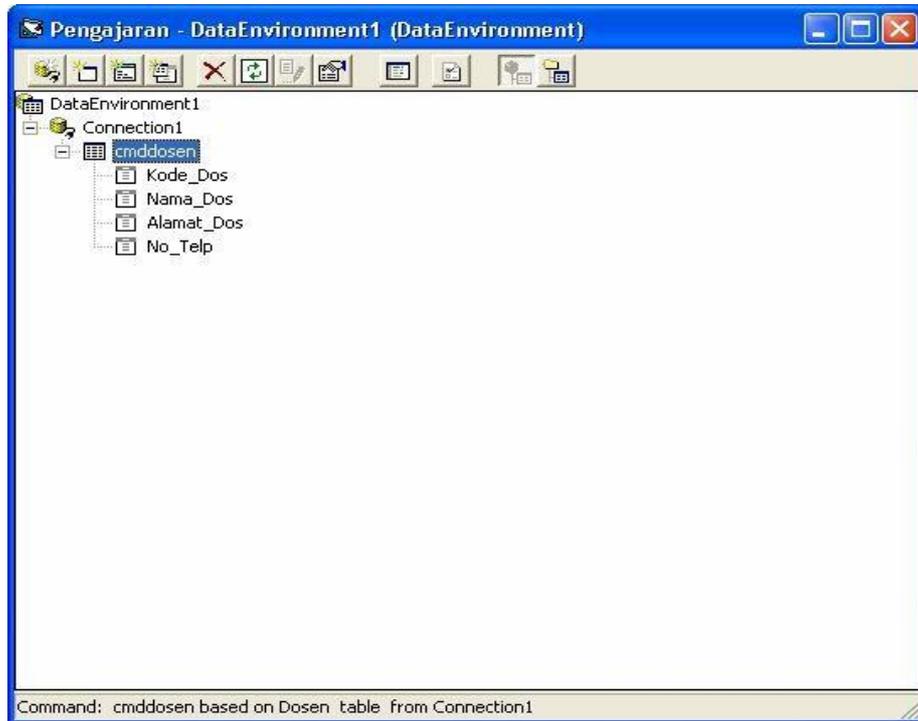


Pada bagian **Select or enter a database name** isikan dengan nama database (Pengajaran.mdb). Kemudian klik OK.

Setelah Data Environment terhubung dengan database, langkah berikutnya adalah menambahkan command sebagai pendefinisian tabel (1 command = 1 tabel/query) dengan cara klik kanan **Connection1** – Pilih **Add Command**. Kemudian **klik kanan Command1** – Pilih **Properties** sehingga akan ditampilkan jendela seperti pada gambar berikut :

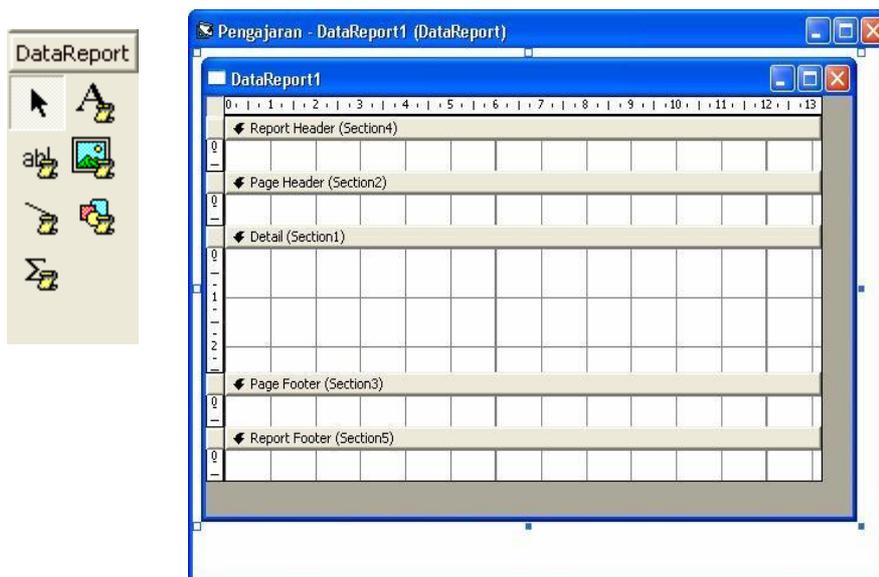


Langkah berikutnya ubah nama **Command1** menjadi **cmddotsen**, Pada bagian **Database Object** pilih pilihan **Table**. Pada **Object Name** pilih nama tabel yang akan dipakai (**Dosen**). Kemudian Klik **Apply – OK**, sehingga pada tampilan akhir Data Environment akan didapat hasil seperti pada gambar 10.6:



### 3. Menambahkan Data Report

**Data Report** merupakan komponen utama untuk mendesain tampilan laporan. Untuk menambahkan Data Report dapat dilakukan dengan cara memilih Menu **Project – Add Data Report**. Sehingga akan ditampilkan **Lembar Desain Data Report** (Gambar kiri) dan **Toolbox Data Report** (Gambar kanan).



Untuk mendesain laporan kita dapat menggunakan komponen-komponen yang terdapat pada data report.

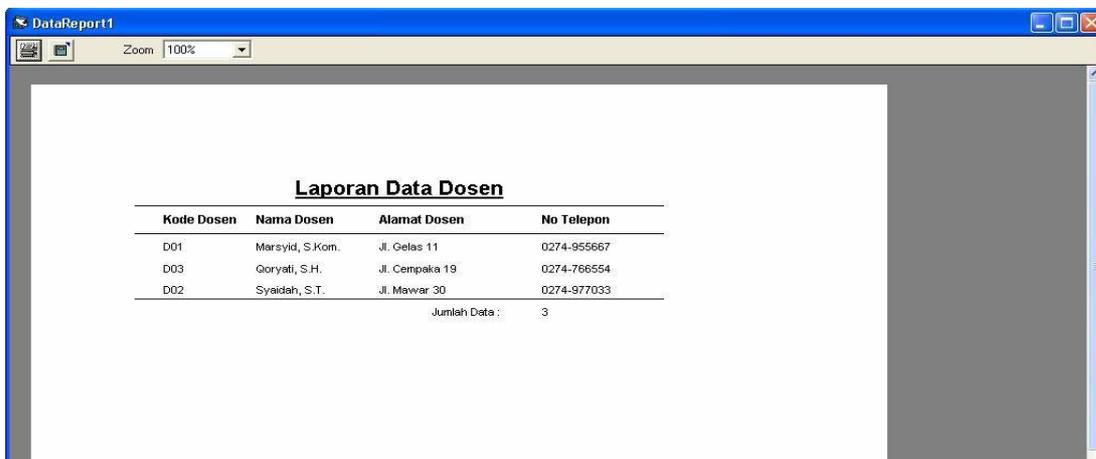
Sebagai contoh, kita akan mendesain laporan data dosen dari database pengajaran diatas. Tambahkan komponen-komponen data report kedalam desain laporan dengan pengaturan properti sebagai berikut:

Komponen	Properti	Nilai
DataReport1	(Name) DataSource DataMember	LapDosen DataEnvironment1 (pilihan) Cmddosen (pilihan)
<b>Report Header (Section4)</b>		
RptLabel1	Caption	Laporan Data Dosen
<b>Page Header (Section2)</b>		
RptLabel2	Caption	Kode Dosen
RptLabel3	Caption	Nama Dosen
RptLabel4	Caption	Alamat Dosen
RptLabel5	Caption	No Telepon
RptLine1	Top	0
RptLine2	Top	440
<b>Detail (Section1)</b>		
RptTextBox1	DataMember DataField	Cmddosen Kode_Dos
RptTextBox2	DataMember DataField	Cmddosen Nama_Dos
RptTextBox3	DataMember DataField	Cmddosen Alamat_Dos
RptTextBox4	DataMember DataField	Cmddosen No_Telp
<b>Page Footer (Section3)</b>		
RptLine3	Top	0
<b>Report Footer (Section5)</b>		
RptLabel6	Caption	Jumlah Data
RptFunction	FunctionType	4-rptFuncRCnt
RptLine4	Top	0

Sehingga akan didapatkan hasil seperti pada gambar berikut.



Kemudian jalankan program sehingga didapat hasil seperti berikut.



# 4

## ALGORITMA PEMROGRAMAN (DELPHI)

# PRAKTIKUM 1

## ALGORITMA DAN FLOWCHART

### A. TUJUAN

1. Mampu memahami persoalan
2. Mendesain penyelesaian persoalan ke dalam algoritma
3. Menotasikan algoritma yang sudah dibuat menggunakan notasi flowchart

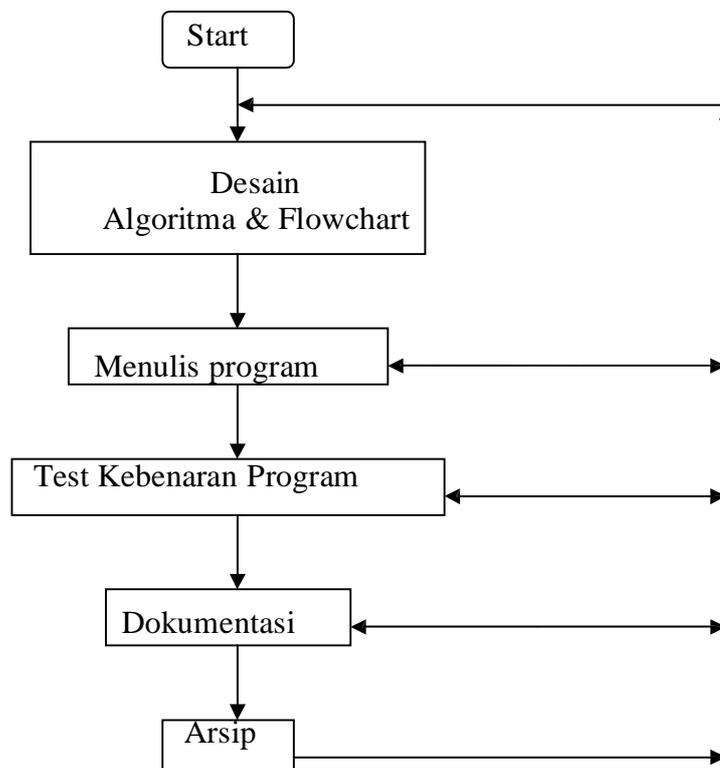
### B. DASAR TEORI

Beberapa langkah dalam proses pembuatan suatu program atau software :

#### 1. Mendefinisikan masalah dan menganalisisnya

Langkah awal yang harus dilakukan adalah mengidentifikasi masalah antara lain tujuan dari pembuatan program, parameter-parameter yang digunakan, fasilitas apa saja yang akan disediakan oleh program. Kemudian menentukan metode atau algoritma apa yang akan diterapkan untuk menyelesaikan masalah tersebut dan terakhir menentukan bahasa program yang digunakan untuk pembuatan program.

#### 2. Merealisasikan dengan langkah-langkah berikut :



### Algoritma

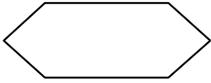
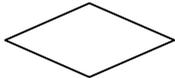
Algoritma adalah urutan langkah-langkah logika yang menyatakan suatu tugas dalam menyelesaikan suatu masalah atau problem.

Contoh : Buat algoritma untuk menentukan apakah suatu bilangan merupakan bilangan ganjil atau bilangan genap.

Algoritmanya :

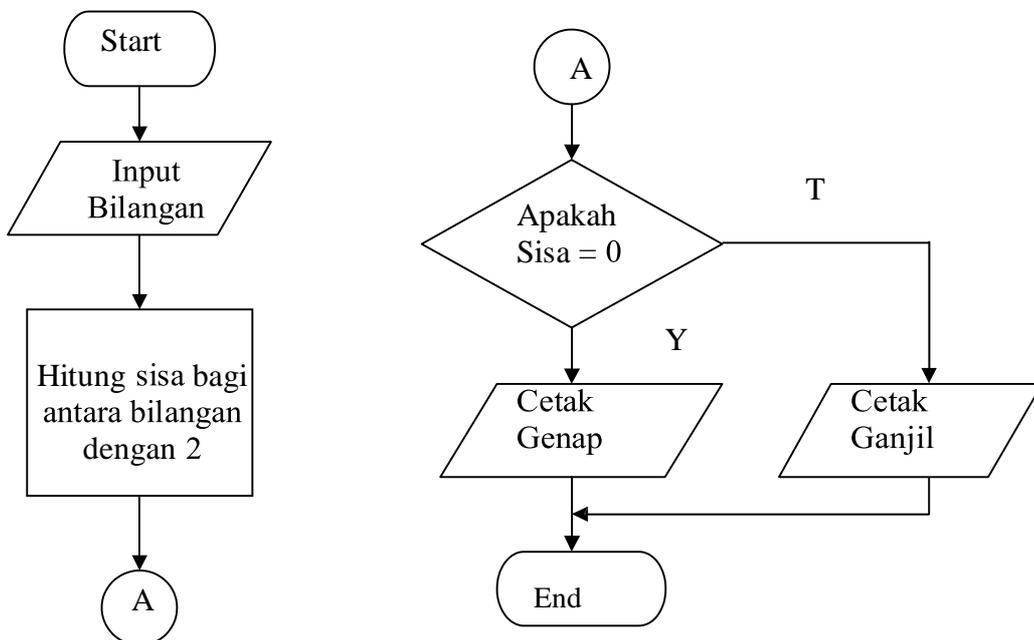
1. Masukkan sebuah bilangan sembarang

2. Bagi bilangan tersebut dengan bilangan 2
3. Hitung sisa hasil bagi pada langkah 2.
4. Bila sisa hasil bagi sama dengan 0 maka bilangan itu adalah bilangan genap tetapi bila sisa hasil bagi sama dengan 1 maka bilangan itu adalah bilangan ganjil.

<b>SIMBOL</b>	<b>NAMA</b>	<b>FUNGSI</b>
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inisialisasi/ pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/ proses pengolahan data
	<b>INPUT/OUTPUT DATA</b>	Proses input/output data, parameter, informasi
	<b>REDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/ proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

SIMBOL	NAMA	FUNGSI
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Dari contoh algoritma di atas tentang menentukan apakah suatu bilangan adalah bilangan ganjil atau bilangan genap, flowchart dari program adalah sebagai berikut :



### C. TUGAS

Untuk semua persoalan di bawah ini, desainlah algoritma dan flowchartnya :

- Menghitung luas lingkaran dan mencetak hasilnya
- Proses pengiriman email, jika email tidak terkirim maka akan dilakukan proses pengiriman email lagi.
- Proses melakukan panggilan telepon, jika tidak diangkat maka dilakukan panggilan telepon lagi.
- Perusahaan daerah air minum menentukan tarif per  $m^3$  air berdasarkan ketentuan sebagai berikut:

Air( $m^3$ )	Tarif per $m^3$
0-15	300
>15	650

Buatlah program yang membaca jumlah pemakaian air, untuk menghitung biaya yang harus dibayar dan menampilkan hasilnya di layar.

5. Buatlah program untuk menjumlahkan bilangan 1 sampai 10, tapi yang dijumlahkan bilangan ganjil saja.

## SEKILAS TENTANG DELPHI

### Maksud dan Tujuan :

1. Mengetahui tentang pemrograman Delphi sebagai visual programming
2. Mengetahui tipe file utama dalam project Delphi
3. Mengetahui tipe data dalam Delphi.
4. Mengetahui lingkungan kerja delphi

Delphi adalah sebuah perangkat lunak (bahasa pemrograman) untuk membuat program / aplikasi komputer berbasis windows. Delphi merupakan bahasa pemrograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek. Ciri sebuah objek adalah memiliki nama, properti dan method/procedure. Delphi disebut juga *visual programming* artinya komponen- komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul berupa gambar-gambar. Dengan demikian delphi lebih mudah digunakan dibandingkan dengan pascal, karena anda hanya tinggal mengatur properti atau sifat dari objek dan menambahkan sedikit pengkodean.

Didalam delphi ada beberapa pengertian yang memerlukan penjelasan untuk memahami bagaimana dan apa saja komponen yang ada dalam delphi, adapun diantaranya adalah:

#### a. Project ( \*.dpr )

Project delphi adalah sekumpulan file yang akan membangun aplikasi atau library yang dapat disebarkan kepada pemrogram lain. Pada saat memulai delphi, delphi akan membuat sebuah project baru. Setelah itu terserah pengguna apakah ingin memodifikasi projek tersebut, membuat projek baru lagi atau mungkin membuka projek yang sudah ada.

Project aplikasi terdiri atas :

1. File project : file ini tersimpan dengan ekstensi .dpr, hanya ada satu file projek untuk satu projek.
  2. File form : file form tersimpan dengan ekstensi .dfm, adalah file biner yang berisi gambaran grafis dari form. Setiap file .dfm selalu paralel dengan satu unit file yang berekstensi .pas.
  3. Unit (source) : file unit tersimpan dengan ekstensi .pas, dimana setiap unit berupa file source code dalam bahasa pascal. Tidak setiap file unit memiliki atau terhubung file .dfm
- File projek menggabungkan semua file diatas yang digunakan oleh delphi untuk melakukan kompilasi dan membuat file target (.exe atau .dll).

#### b. Form ( \*.dfm )

Form merupakan tempat kita merancang program. Jika seorang pemrogram diibaratkan sebagai seorang pelukis maka form inilah yang menjadi kanvasnya. Form yang dirancang dalam delphi bisa digunakan kembali oleh projek delphi yang lain. Form dapat juga disimpan dalam bentuk dll, sehingga dapat digunakan oleh aplikasi yang lain seperti c++, paradox, atau dbase. Delphi menawarkan kemudahan pembuatan form melalui model template.

#### c. Unit ( \*.pas )

Unit merupakan modul yang terpisah yang dapat dikompilasi tersendiri, yang berisi deklarasi bagian publik (antarmuka) dan bagian privat (implementasi). Setiap form dalam delphi mempunyai satu unit yang bersesuaian. Source code

unit tersimpan dengan ekstensi pas, sedangkan yang sudah terkompilasi tersimpan dengan ekstensi .dcu. proses link akan menggabung file .dcu menjadi satu file exe atau dll.

### Tipe data

Setiap data yang disimpan di memori komputer harus memiliki tipe data.

#### a. Tipe Integer

Type	Rentang Nilai	Byte
Byte Word	0...255	1
LongWord	0...55535	2
ShortInt	0...424967295	4
SmallInt	-128...127	1
LongInt	-32768...32767	2
Integer	-2147483648...2147483647	4
Cardinal	-2147483648...214783647	4
Int64	0...4294967295	4
	$-2^{63}$ ... $2^{63}-1$	8

#### b. Tipe Real

Type	Rentang Nilai	Ketelitian (digit)	Byte
Real48	$2.9 \times 10^{-39}$ ... $1.7 \times 10^{38}$	11 – 12	6
Single	$1.5 \times 10^{-45}$ ... $3.4 \times 10^{38}$	7 – 8	4
Double	$5.0 \times 10^{-324}$ ... $1.7 \times 10^{308}$	15 – 16	8
Real	$5.0 \times 10^{-324}$ ... $1.7 \times 10^{308}$	15 – 16	8
Extended	$5.0 \times 10^{-324}$ ... $1.7 \times 10^{308}$	19 – 20	10
Comp	$3.6 \times 10^{-4951}$ ... $1.1 \times 10^{4932}$	19 – 20	8
Currency	$-2^{63}+1$ ... $2^{63}-1$	19 - 20	8

#### c. Tipe Boolean

Tipe	Byte
Boolean	1
ByteBool	1
WordBool	2
LongBool	4

#### d. Tipe Character

Tipe	Byte
Char	1
AnsiChar	1
WideChar	2

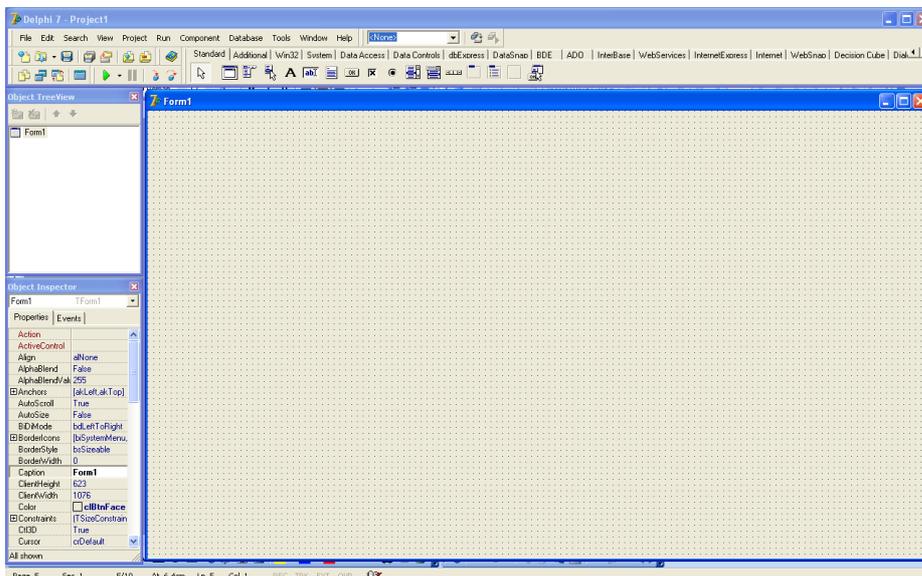
#### e. Tipe String

Tipe	Byte	Isi Maksimum
ShortString	2 sd. 256	256 karakter
AnsiString	4 sd. 2 GB	$2^{31}$ karakter
WideString	4 sd. 2 GB	$2^{30}$ karakter

### MENGENAL LINGKUNGAN KERJA DELPHI

Sebelum mempelajari ketiga struktur pemrograman ada baiknya kenali dahulu tampilan IDE, yang merupakan editor dan tools untuk membuat program Delphi. Pada IDE akan ditampilkan Form baru yang merupakan aplikasi/program Window yang akan dibuat.

Tampilan interface delphi adalah sebagai berikut :



Gambar 1.0 .Tampilan IDE delphi.

Untuk menjalankan:

1. Klik tombol **Start**, pilih menu **Programs – Borland Dephi 7 – Delphi 7**
2. Tunggu hingga program delphi aktif
3. Perhatikan komponen yang ada, diantaranya component pallete , berisi sekumpulan objek yang akan kita gunakan dalam pembuatan program.

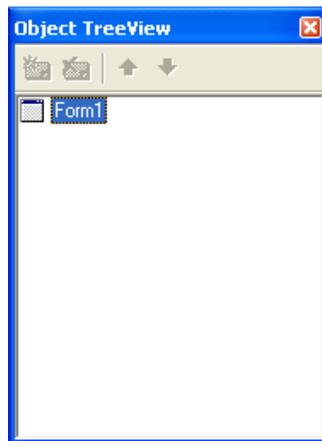
Kita juga dapat menambah komponen tambahan yang tidak disertakan dalam paket Delphi atau yang lebih sering dikenal dengan *plugin vcl (visual component library)*

eksternal.



Gambar 1.1. Toolbar Component Pallete.

4. Berikut ini adalah object treeview, digunakan untuk memonitoring suatu object yang terdapat pada project yang sedang kita buat. Semua komponen yang diletakan pada lembaran form akan terlihat di dalam object treeview.



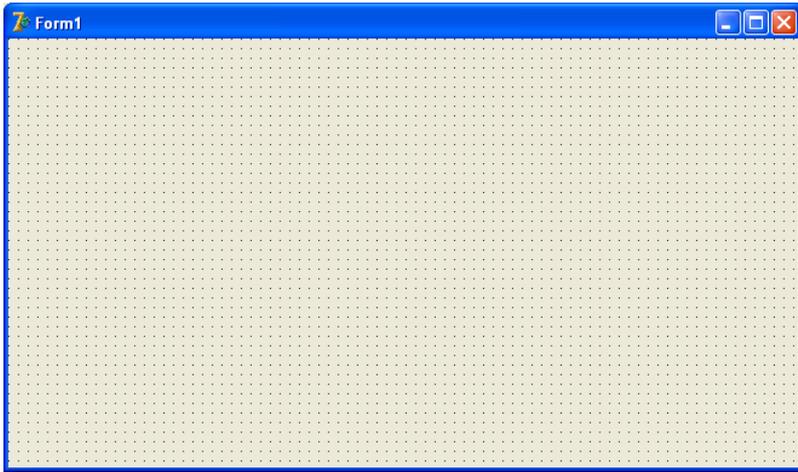
Gambar 1.2. Object treeview.

5. Object inspector (berisi daftar property atau sifat dari objek yang aktif yang dapat kita ubah sesuai dengan kebutuhan dalam program kita). Kedua objek ini merupakan komponen yang paling sering dipakai dalam pembuatan program visual.



Gambar 1.3. Object Inspector

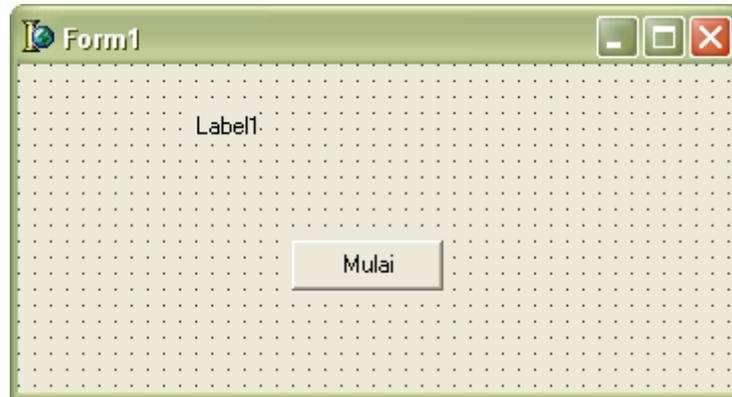
6. Pada gambar dibawah ini adalah form atau lembar kerja yang digunakan untuk meletakkan komponen.



*Gambar 1.4. Lembar kerja form*

## PRAKTIKUM 2

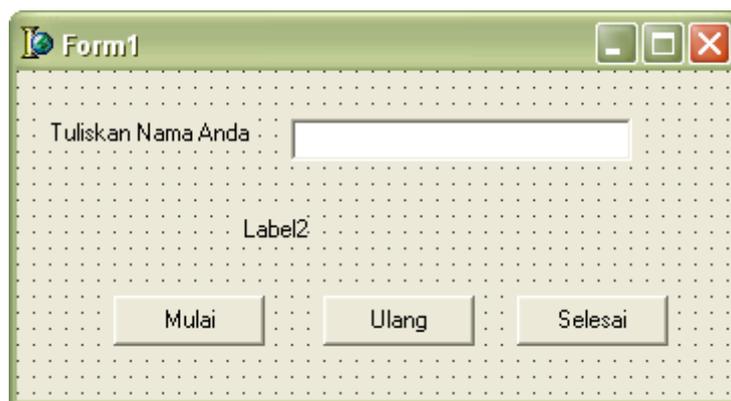
1. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**label dan Button.**



Double Klik Button **Mulai** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.MulaiClick(Sender: TObject);  
  
begin  
Label1.Caption:= ' Selamat Datang di Borland Delphi ' ;  
end;  
end.
```

2. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**2buah label, 3 buah Button dan 1 Edit.**



Double Klik Button **Mulai** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
Label2.Caption := Edit1.Text ;  
end;
```

Double Klik Button **Ulang** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;
Label2.Caption:="";
Edit1.SetFocus; end;
```

Double Klik Button **Selesai** dan tuliskan Sintax Program dibawah ini

```
procedure TForm1.Button3Click(Sender: TObject);
begin
close;
end;
```

3. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**label, Button, Edit, Groupbox dan Radiobutton**



Double Klik Button **Mulai** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Label2.Caption := Edit1.Text ;
end;
```

Double Klik Button **Ulang** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;
Label2.Caption:="";
Edit1.SetFocus; end;
```

Double Klik Button **Ulang** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.Button3Click(Sender: TObject);
begin
Application.terminate;
end;
```

Double Klik Radiobutton **Biru** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
Label2.Font.Color:= clBlue;
end;
```

Selanjutnya untuk RadioButton yang lain gunakan syntax program yang sama dengan tinggal merubah warnanya.

4. Buatlah tampilan dibawah ini dan Tuliskan syntax programnya ;

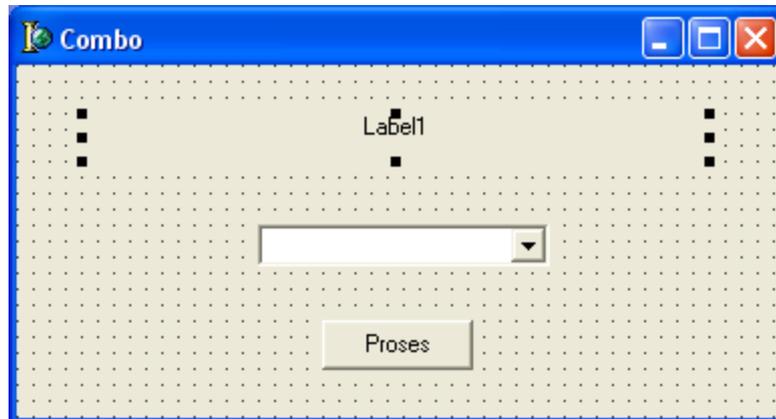


The screenshot shows a Windows form titled "Form1" with a blue title bar and standard window controls (minimize, maximize, close). The form contains the following elements:

- A text label "Masukkan Tullisan" followed by a text input field.
- A label "Label2" positioned below the input field.
- Three buttons labeled "MULAI", "ULANG", and "SELESAI" arranged horizontally.
- A group box titled "Pilihan Warna" containing five radio buttons: "Biru", "Merah", "Kuning", "Hijau", and "Coklat".
- A group box titled "Ukuran Huruf" containing five radio buttons: "12", "14", "16", "18", and "20".

### PRAKTIKUM 3

1. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**label, ComboBox dan Button.**



Double Klik Button **Proses** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm2.Button1Click(Sender: TObject);  
begin  
Label1.Caption := ComboBox1.Text;  
end;
```

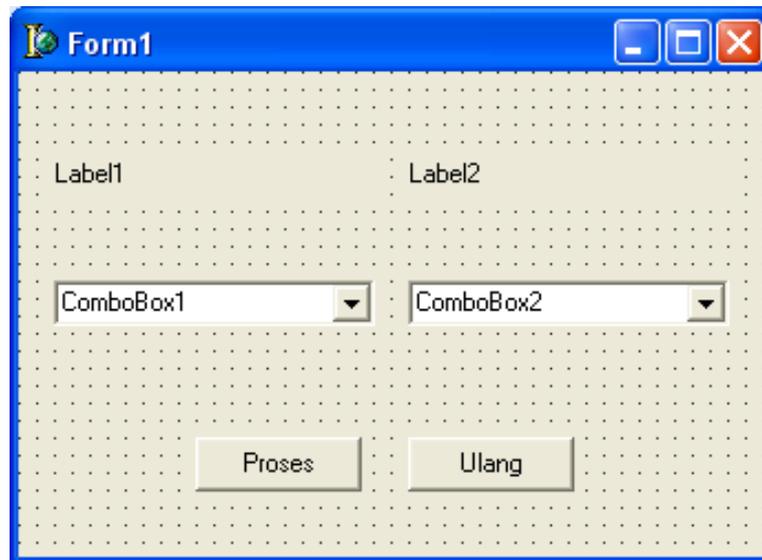
Klik **Form** masuk ke event **OnActive** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm2.FormActivate(Sender: TObject);  
begin  
ComboBox1.Items.Add('Bahasa');  
ComboBox1.Items.Add('Pemrograman');  
ComboBox1.Items.Add('Borland');  
ComboBox1.Items.Add('Delphi');  
end;
```

Double Klik **ComboBox** dan tuliskan Sintax Program dibawah ini :

```
procedure TForm2.ComboBox1Change(Sender: TObject);  
begin  
Label1.Caption := ComboBox1.Text;
```

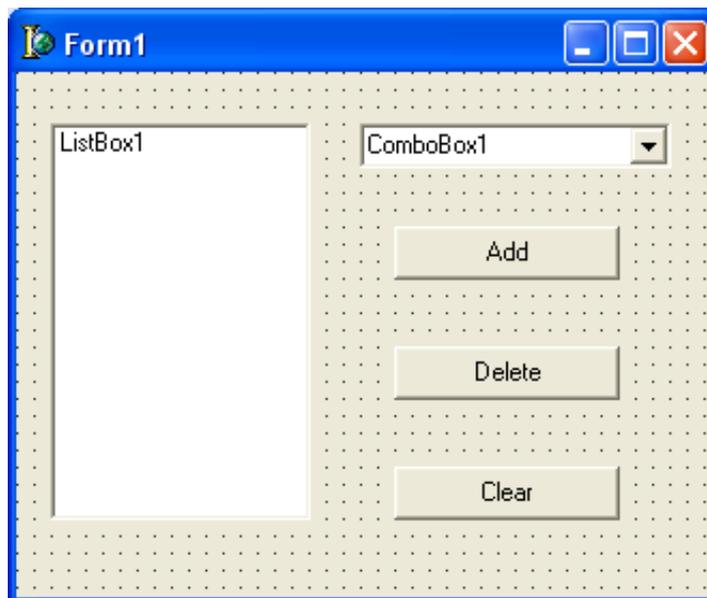
2. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**label, ComboBox dan Button.**



**Keterangan :**

Combo1 berisi Beberapa Field ini : <ul style="list-style-type: none"> <li>- Microsoft Windows XP</li> <li>- Linux</li> <li>- UNIX</li> <li>- Macintosh</li> <li>- Free BSD</li> </ul>	Combo2 berisi Field : <ul style="list-style-type: none"> <li>- Microsoft Word</li> <li>- Microsoft Excell</li> <li>- Microsoft Access</li> <li>- Microsoft Power Point</li> <li>- Microsoft Publisher</li> </ul>
---	--

- Button **Proses** untuk mengaktifkan data yang ada di label supaya muncul sesuai yang ada pada ComboBox
  - Button **Ulang** berfungsi untuk menghilangkan kembali tulisan yang ada pada label.
- 3.
  4. Buatlah tampilan dibawah ini dengan menggunakan **listbox** dan **Combo dan Button.**



- Isi combo dengan beberapa field nama seseorang :

**Michael Jhonson Peter**  
**Smith Albert Einstein**  
**Linus Torvalds Bill**  
**Gates**

Perintah **Add** untuk menampilkan data yang ada pada combo Box1 muncul pada listbox. Sintax Programnya ;

```
procedure TForm2.Button1Click(Sender: TObject);  
begin  
    ListBox1.Items.Add(ComboBox1.Text);  
end;
```

Perintah **Delete** untuk menghapus field yang sudah masuk di listbox. Dengan menghapus satu-satu pada listbox. Sintax programnya ;

```
procedure TForm2.Button2Click(Sender: TObject);  
begin  
    ListBox1.Items.Delete(ListBox1.ItemIndex);  
end;
```

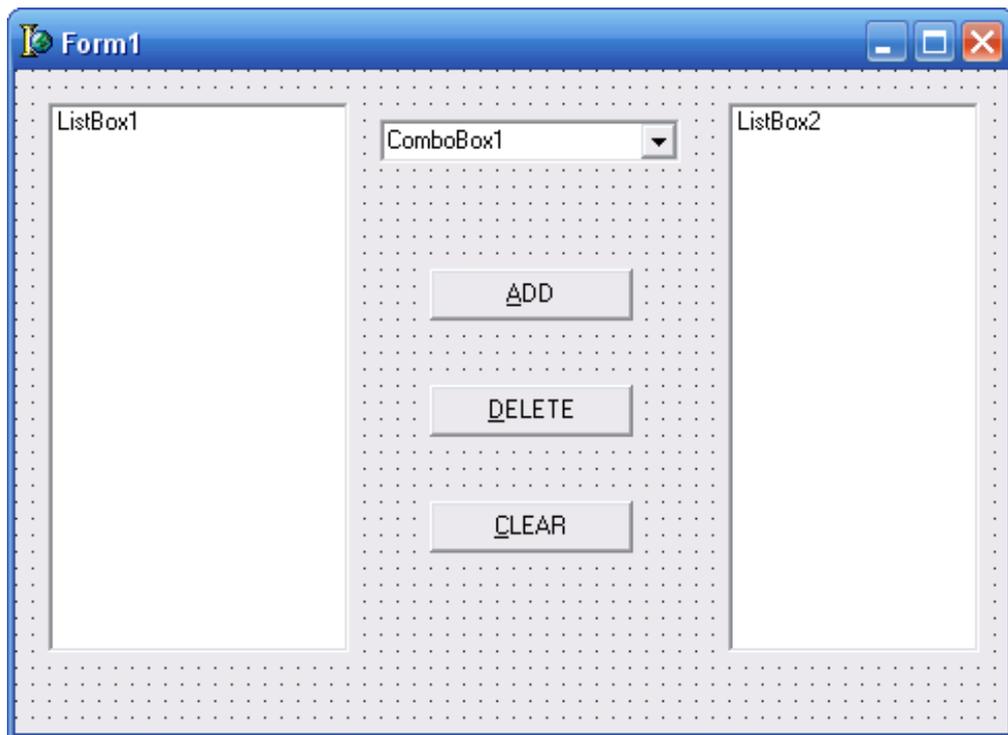
Perintah **Clear** untuk mengosongkan kembali tampilan pada ListBox.

```
procedure TForm2.Button3Click(Sender: TObject);  
begin  
    ListBox1.Clear;  
end;
```

Perintah dibawah ini untuk mengaktifkan kursor supaya kursor berada pada ComboBox1. Sintax programnya ;

```
procedure TForm2.FormActivate(Sender: TObject);  
begin  
    ComboBox1.SetFocus;  
end;
```

5. Buatlah program tampilan dibawah ini :



## PRAKTIKUM 4

### 4.1 Komentar

Komentar adalah bagian dari program yang tidak dikompilasi oleh kompiler. Komentar berguna bagi programmer itu sendiri atau orang lain yang membaca program. Kita bias membuat komentar dengan menuliskannya di dalam kurung kurawal buka ( { ) dan kurung kurawal tutup ( } ). Bisa juga di antara pasangan tanda asterisk di dalam sepasang kurung ( \* \* ). Komentar juga bisa kita hasilkan dari pemakaian dua tanda garis miring kanan atau slash ganda ( // ). Namun tanda ini tidak memiliki penutup, alhasil pemberhentian komentar ditentukan oleh akhir baris kode. Contoh :

1. { Menulis pada Edit1 }  
edit1.text := 'Borland Delphi';
2. (\* Menulis pada Edit1 \*)  
edit1.text := 'Borland Delphi';
3. edit1.text := 'Borland Delphi'; // Menulis pada Edit1

### 4.2 Konstanta dan Variabel

#### 4.2.1 Konstanta

Konstanta adalah nama yang diberikan pada suatu nilai yang besarnya tetap (konstan). Konstanta sering digunakan sebagai titik-titik terminasi dari suatu proses perhitungan program. Penulisan konstanta diawali dengan kata **Const**. Aturan penamaan konstanta :  
Panjang maksimum 63 karakter.  
Harus diawali sebuah huruf atau garis bawah ( \_ ).  
Karakter berikutnya bisa huruf, garis bawah atau angka 0 sampai 9. Tidak boleh mengandung simbol-simbol seperti \$, %, \*.  
Tidak boleh menggunakan reserved word, misalnya close, print. Hindari pemakaian nama yang telah dipakai Delphi.

#### Contoh:

```
procedure TForm1.Button1Click(Sender: TObject); Const  
    Namaku = 'Jenderal Kancil';  
    Umurku = '12 Tahun';  
begin  
    showmessage(namaku + ' "umurnya baru"' + umurku);  
end;  
end.
```

#### 4.2.2 Variabel

Variabel adalah sebuah nama yang mewakili suatu nilai yang dapat berubah-ubah. Lingkup suatu variabel adalah didalam blok tempat deklarasinya. Jika di blok program global, artinya berlaku di seluruh bagian program. Sedangkan apabila berada pada blok kecil seperti fungsi atau prosedur maka akan bersifat lokal. Adapun aturan penulisan variabel sama dengan aturan penulisan konstanta.

Penulisan deklarasi variabel diawali dengan kata **Var**, contohnya :

**Var**

**Akhir** : Integer;  
**Garis** : String;

### 4.3 Tipe Data

Secara garis besar, tipe data yang dikenal Delphi bisa digolongkan menjadi enam kelompok, yaitu :

1. Tipe data Dasar adalah himpunan nilai yang telah tertentu. Terdiri atas dua grup yaitu Ordinal dan Real.
  - A. Tipe Ordinal dibagi menjadi dua, tertentu dan ditentukan (ditentukan oleh pemakai).
    1. Ordinal tertentu adalah Integer, Character dan Boolean.
      - a) Integer terbagi menjadi Integer, ShortInt, SmallInt, LongInt, Byte, Word dan Cardinal.
      - b) Character terbagi menjadi AnsiChar dan WideChar.
      - c) Boolean terbagi menjadi Boolean, ByteBool, WordBool dan LongBool.
    2. Ordinal ditentukan adalah enumeration dan subrange.
  - B. Tipe real dibagi menjadi enam yaitu real, single, double, extended, comp dan currency.
2. Tipe data String adalah suatu urutan karakter dengan sebuah atribut dinamis dan sebuah atribut ukuran konstan. Tipe data string terbagi menjadi 3 tipe yakni ShortString, LongString, dan WideString.
3. Tipe data Terstruktur adalah susunan yang mampu menangani lebih dari sebuah nilai. Tipe data terstruktur terbagi atas enam tipe, yakni record, Array, Set, File, Class dan Class reference.
  - A. Record dibagi menjadi Record, Fixed Record, dan Variant Record.
  - B. Array dibagi menjadi Array Zero-Based, Array Multi Dimensi dan Array Character.
4. Tipe data Pointer adalah sebuah himpunan nilai yang menunjuk ke variabel dengan tipe data khusus.
5. Tipe data Procedural adalah sebuah tipe data yang bisa menangani prosedur dan fungsi sebagai objek.
6. Tipe data Variant bisa menangani variabel agar menggunakan nilai dari tipe data lainnya.

### Tipe Data yang sering digunakan

Tipe Data	Jangkauan	Panjang Bit
Integer	Seluruh bilangan antara -2147483648..2147483647	32 bit
Byte	Seluruh bilangan antara 0 .. 255. Tidak bisa menampung angka negatif.	8 bit

Word	Seluruh bilangan antara 0 .. 65535. Tidak bisa menampung angka negatif.	16 bit
Real	Bilangan desimal antara $2.9 \times 10^{-39}$ .. $1.7 \times 10^{38}$	6 byte
Single	Bilangan desimal antara $1.59 \times 10^{-45}$ .. $3.4 \times 10^{38}$	4 byte
Currency	(Dipakai untuk menyatakan nilai mata uang). Bilangan desimal antara – 922337203685477.5808 .. 922337203685477.5807	
Boolean	Nilai Boolean berupa True atau False.	
String	Data yang berbentuk karakter.	

### 3.4 Rutin Konversi Data

Tipe data yang telah dideklarasikan, masih bisa kita ubah tanpa harus mengubah deklarasi atau membuat deklarasi lokal, yaitu dengan proses casting atau konversi data. Ada beberapa sarana yang bisa kita gunakan untuk maksud tersebut, antara lain :

Casting langsung. Pemakaian rutin konversi. Pemakaian operator **as** dan **is**.

#### Rutin Konversi Tipe

Rutin	Gunanya
Chr	Mengubah sebuah angka tipe ordinal menjadi sebuah karakter ASCII.
Ord	Mengubah sebuah nilai tipe real menjadi sebuah nilai tipe integer.
Round	Mengubah sebuah nilai tipe real menjadi sebuah angka integer, dengan membulatkan sisanya.
Trunc	Mengubah nilai tipe real menjadi integer, dengan memotong sisanya.
Int	Menghasilkan bagian-bagian integer dari nilai argumen floating point.
IntToStr	Mengubah sebuah angka (integer) menjadi sebuah string.

StrToInt	Mengubah sebuah string menjadi sebuah angka dengan menampilkan eksepsi jika stringnya salah.
Val	Mengubah sebuah string menjadi sebuah angka
Str	Mengubah sebuah angka menjadi sebuah string dengan menggunakan paameter format.
FloatToStr	Mengubah nilai floating point (real) menjadi string.
StrToFloat	Mengubah nilai string Pascal menjadi sebuah nilai floating point(real).

#### 4.5 Tipe Subrange

Tipe data ini memberlakukan suatu ruang lingkup yang dapat kita tentukan batasnya dan berlaku untuk data-data integer, Boolean, char atau urutan. Tipe data ini sangat membantu kita apabila bekerja dengan data- data yang memiliki range (jangkauan).

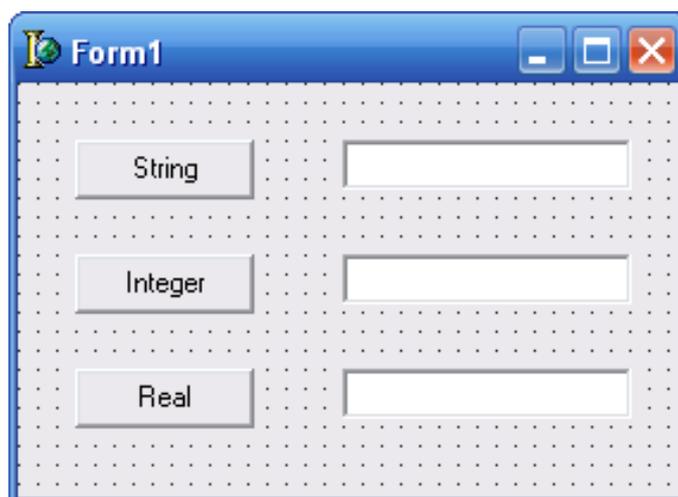
**Contoh :** type

```
Twaktu = 0..23;
Tumurjagung = 3..4;
```

Var

```
Waktu : Twaktu;
Umurjg : Tumurjagung;
```

1. Buat kan tampilan dibawah ini dengan memasukkan komponen ;  
**Button dan Edit**



Code Program untuk Button **String**

```
procedure TForm1.Button1Click(Sender: TObject);
var
begin
    test1: string;
    test1:= 'Borland Delphi'; Edit1.Text:= test1;
end;
```

Code Program untuk Button **Integer**

```
procedure TForm1.Button2Click(Sender: TObject);
var
begin
    test2: Integer;
    test2:= 20000;
    Edit2.Text:= IntToStr(test2);
end;
```

Code Program untuk Button **Real**

```
procedure TForm1.Button3Click(Sender: TObject);
var
begin
    test3: real;
    test3:= 3.14;
    Edit3.Text:= FloatToStr (test3);
end;
```

2. Buat tampilan dibawah ini dengan memasukkan komponen ;  
**Button , Edit dan Label**



Code Program di **Form-Event-OnActive** :

```
procedure TForm2.FormActivate(Sender: TObject);
begin
    Edit1.SetFocus ;
end;
```

Code Program di Button **Proses** :

```
procedure TForm2.Button1Click(Sender: TObject);
var X,Y : Integer;
begin
    X := StrToInt(Edit1.Text); Y :=
    StrToInt(Edit2.Text); Edit3.Text:= IntToStr(X*Y);
end;
```

Code Program di Button **Ulang** :

```
procedure TForm2.Button2Click(Sender: TObject);
begin
    Edit1.Text:= ' '; Edit2.Text:= ' ';
    Edit3.Text:= ' '; Edit1.SetFocus ;
end;
```

3. Buat tampilan dibawah ini dengan memasukkan komponen ;  
**Button , Edit dan Label**



## PRAKTIKUM 5 PERNYATAAN TERSTRUKTUR

### Macam-macam Pernyataan Terstruktur

Pernyataan terstruktur mencakup pernyataan-pernyataan yang dijalankan secara berurutan, didasarkan kondisi, maupun yang berulang. Macam pernyataan terstruktur antara lain :

1. Pernyataan majemuk, yang akan menyebabkan pernyataan- pernyataan di eksekusi secara berurutan.
2. Pernyataan berkondisi, yang akan dieksekusi berdasarkan suatu kondisi.
3. Pernyataan kalang (loop) yang akan menyebabkan pernyataan- pernyataan dieksekusi secara berulang.

#### 1. Pernyataan Majemuk

Pernyataan majemuk adalah sederetan pernyataan yang akan dijalankan secara sekuensial (berurutan). Pernyataan ini ditandai dengan awalan kata tercadang **begin** dan diakhiri dengan **end**. Format umumnya adalah sebagai berikut :

```
Begin Pernyataan_1;  
    Pernyataan_2;  
    ... Pernyataan_n;  
End
```

#### 2. Pernyataan Berkondisi

Ada dua macam pernyataan berkondisi :

Pernyataan **if**, dan  
Pernyataan **case**.

Pernyataan ini berguna untuk mengambil keputusan yang melibatkan dua alternatif atau lebih.

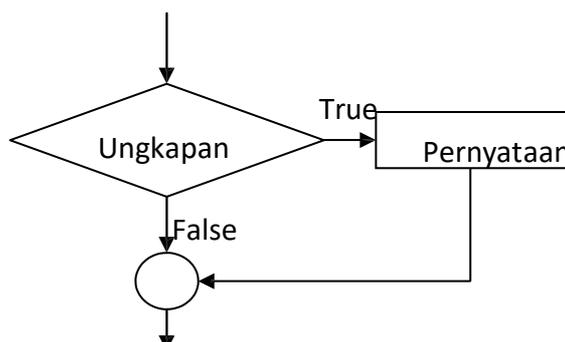
##### 2.1 Pernyataan if

Ada dua macam pernyataan If :

**If...then** dan  
**If...then...Else**.

**Bentuk pernyataan if..then adalah sebagai berikut : If ungkapan then pernyataan**

Dalam hal ini ungkapan berupa suatu ekspresi boolean (bernilai True atau False). Bila ungkapan bernilai **True**, pernyataan akan dijalankan, tetapi jika bernilai **False** pernyataan tidak dijalankan sama sekali.

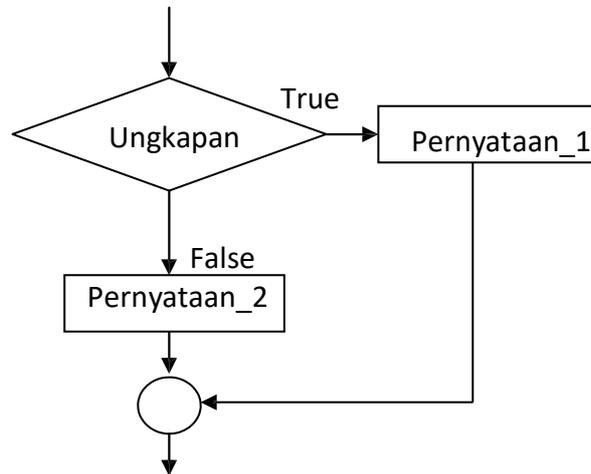


**Bentuk pernyataan if...then...else :**

**If** ungkapan **then** pernyataan\_1

**Else** pernyataan\_2

Pada bentuk ini pernyataan\_1 hanya dijalankan kalau ungkapan bernilai **true**. Apabila ungkapan bernilai **false**, pernyataan\_2 yang akan dijalankan.



**Contoh Latihan :**



**Code Program untuk Button Proses**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    If edit1.text = '7' then
    Begin
        Label3.caption:= 'Tepat, anda benar !';
        Form1.color := clyellow;
    End
    Else
    Begin
        Label3.caption := 'Salah, coba lagi !'; Form1.color := cldred;
    End;
end;
```

1. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**Button, Label dan Combobox**



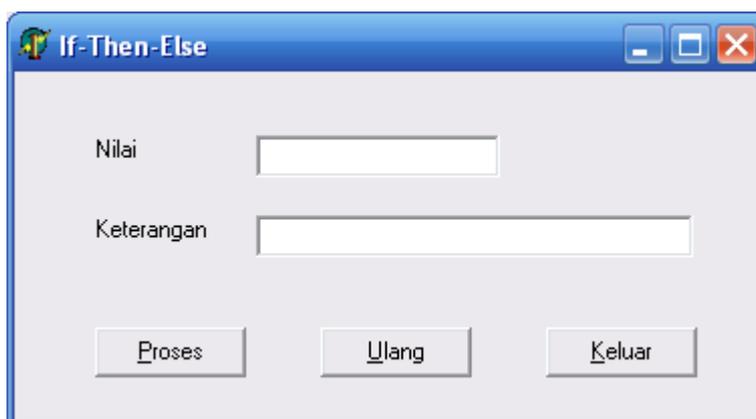
Code Program untuk **ComboBox.Change**

```

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
  if ComboBox1.Text = 'A' then Label3.Caption:= 'Sangat Memuaskan' else
  if ComboBox1.Text= 'B' then Label3.Caption:= 'Memuaskan' else
  if ComboBox1.Text = 'C' then Label3.Caption:= 'Cukup Memuaskan' else
  if ComboBox1.Text = 'D' then Label3.Caption:= 'Kurang Memuaskan' else
  Label3.Caption:= 'Tidak Dikenal' ;
end;

```

2. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**Button, Label dan Edit**



**Keterangan:**

$\geq 90$  = Sangat Memuaskan  
 $\geq 80 < 90$  = Memuaskan  
 $\geq 70 < 80$  = Baik  
 $\geq 60 < 70$  = Cukup  
 $< 60$  = Kurang

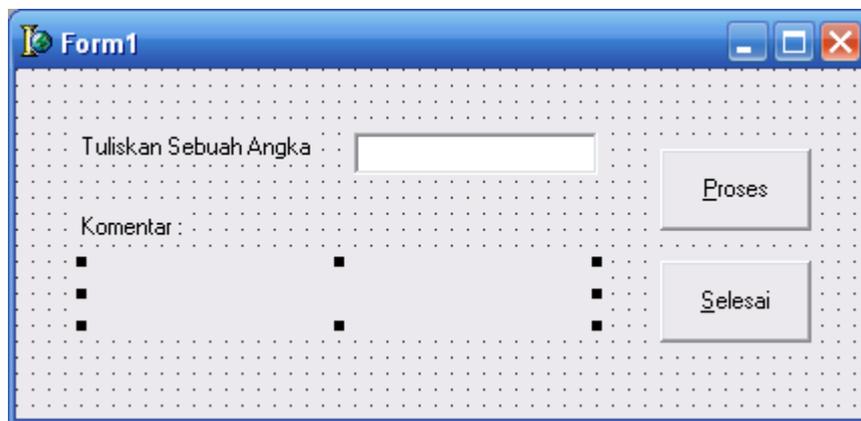
## Pernyataan kondisional Case...of

Jika pilihan kemungkinan hanya sedikit, kita bisa menggunakan **If**. Namun jika pilihannya banyak, kita harus menggunakan **Case**. Case pada dasarnya juga digunakan untuk percabangan yang banyak, hanya saja ia lebih disukai karena penulisannya lebih ringkas dan lebih mudah dimengerti serta prosesnya pun lebih cepat. Adapun struktur **Case...of** seperti berikut :

### Case variabel of

```
Pilihan 1 : pernyataan 1;  
Pilihan 2 : pernyataan 2;  
Pilihan 3 : pernyataan 3;  
Else  
    Pernyataan lain;
```

### Contoh Latihan :



### Code program **Button Proses** :

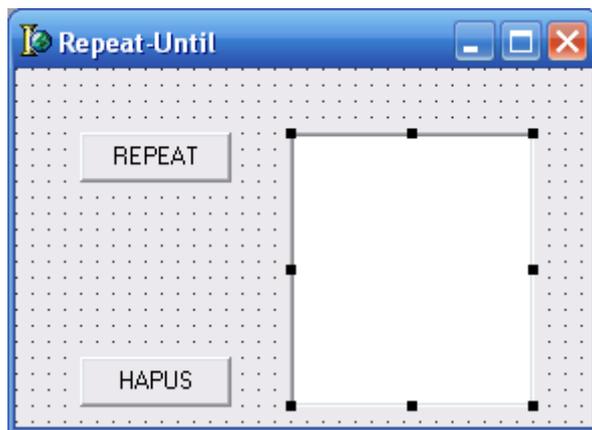
```
procedure TForm1.Button1Click(Sender: TObject);  
Var  
angka : Integer;  
Begin  
    Angka := strtoint(edit1.text);  
    Case angka of  
        1,3,5,7,9           : label3.caption := 'Angka Ganjil';  
        2,4,6,8,10        : label3.caption := 'Angka Genap';  
    else  
        label3.caption := 'Saya tidak kenal angka itu !';  
    end;  
end;  
end;
```

## PRAKTIKUM 6 PERULANGAN

### *Perulangan dengan Repeat*

Tidak seperti pemilihan, dalam perulangan perlu ditetapkan suatu nilai baru, yaitu akhir pengulangan. Hal ini bisa dinyatakan dalam banyaknya pengulangan atau kriteria keadaan sebagai titik berhenti. Pernyataan **Repeat** akan melaksanakan perulangan proses terhadap ekspresi sampai suatu keadaan dinyatakan **True** (bernilai benar). Penulisan pernyataan repeat diawali dengan kata **Repeat** dan diakhiri dengan kata **Until**.

### **Contoh Program :**



### Code program **Button Repeat** :

```
procedure TForm1.Button1Click(Sender: TObject); Var I: integer;
begin
    I:= 0; Repeat
        I:= I + 1;
        Listbox1.items.Add('Perulangan ' + Inttostr(I)); Until I = 100;
end;
```

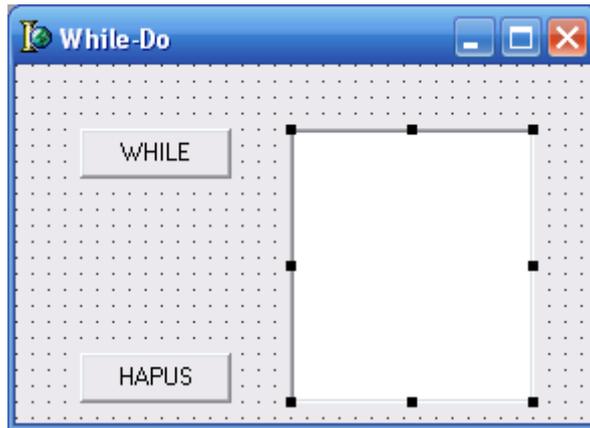
### Code program **Button Hapus** :

```
procedure TForm1.Button2Click(Sender: TObject);
begin
    ListBox1.Clear;
end;
```

### *Perulangan dengan While*

Pernyataan while akan mengevaluasi suatu kondisi ekspresi pada awal proses, merupakan kebalikan dari repeat. Penulisan ini diawali dengan pasangan kata **While...do** lalu diikuti dengan pasangan kata **begin...end**.

### Contoh Program :



### Code program **Button While** :

```
procedure TForm1.Button1Click(Sender: TObject); Var J: integer;
begin
    J:=0;
    While J<100 do
    Begin
        J:=J+1;
        Listbox1.Items.Add('While ' + Inttostr(J));
    end;
End;
```

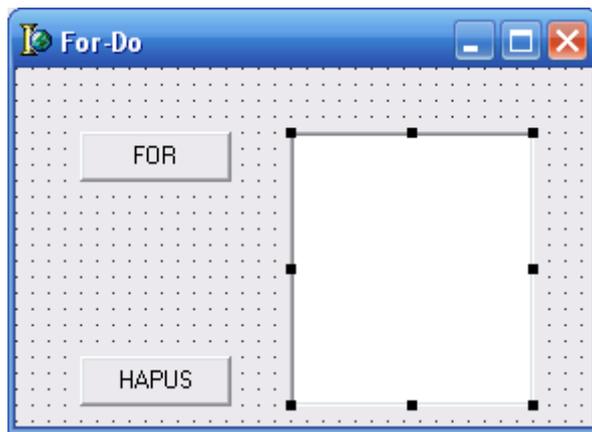
### Code program **Button Hapus** :

```
procedure TForm1.Button2Click(Sender: TObject);
begin
    ListBox1.Clear;
end;
end
```

### ***Perulangan dengan For***

Perulangan dengan for dilakukan dalam jumlah perulangan yang sudah ditentukan. Penulisan perulangan ini diawali dengan pasangan kata **for...do** di dalam pasangan kata **begin...end**.

### Contoh Program :



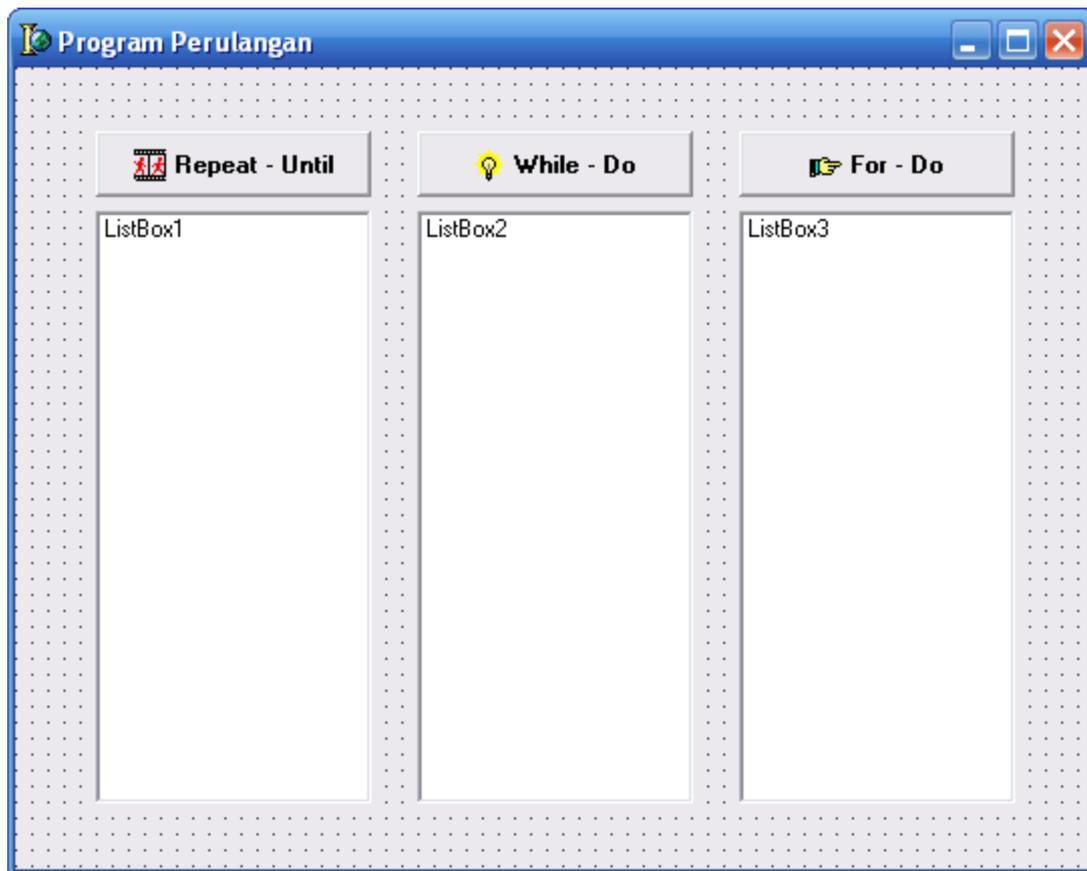
Code program **Button For** :

```
procedure TForm1.Button1Click(Sender: TObject); Var K: integer;  
begin  
    For K:= 1 to 100 do  
        Listbox1.Items.Add('For ' + IntToStr(K));  
  
end;
```

Code program **Button Hapus** :

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    ListBox1.Clear;  
end;  
end.
```

1. Buatlah tampilan dibawah ini dengan memasukkan komponen ;  
**Button, Label** dan Edit



**Keterangan :**

1. Ketika Anda Menekan Tombol **Repeat – Until** maka akan keluar tulisan **“Pemrograman”** sebanyak 100 kali. Dan gunakan perulangan type Repeat Until
2. Ketika Anda Menekan Tombol **While - Do** maka akan keluar tulisan **“Borland”** sebanyak 100 kali. Dan gunakan perulangan type While- Do.
3. Ketika Anda Menekan Tombol For - Do maka akan keluar tulisan **“Delphi”** sebanyak 100 kali. Dan gunakan perulangan type For Do.

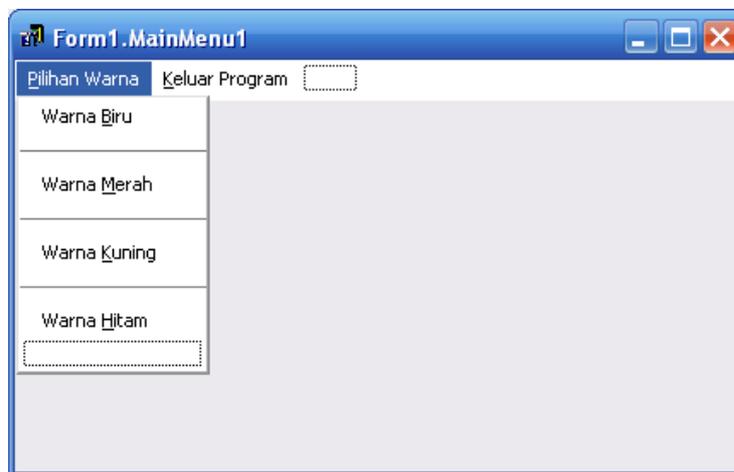
## PRAKTIKUM 7

### MEMBUAT MENU, POPUP MENU DAN PAGE CONTROL

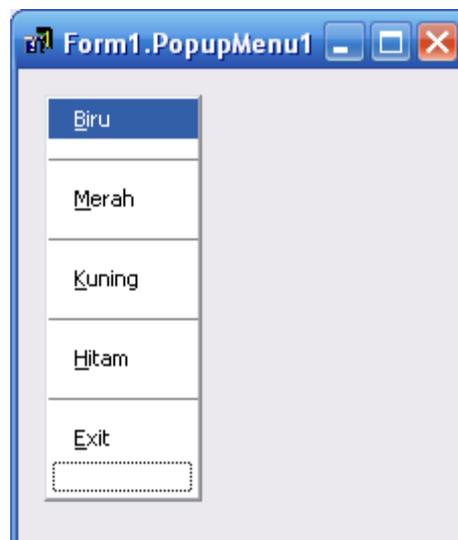
Menu memberikan cara yang mudah untuk menjalankan kelompok- kelompok perintah. Menu designer memungkinkan anda untuk menambahkan sebuah menu ke dalam form. Delphi mengenal dua buah system menu yang ditambahkan bagi program yang sedang dibuat, yaitu :

1. **Main menu**, yaitu berisi sekelompok judul menu yang berisi sejumlah item baru dan subsistem menu
2. **Popup menu**, istilah ini diberikan kepada menu yang disajikan pada saat anda menekan tombol kanan mouse. Menu ini sekarang sudah umum dipakai oleh aplikasi yang berbasis Windows 95 keatas.

Untuk menggunakan TmainMenu, pertama-tama buat aplikasi baru. Kemudian letakkan komponen TmainMenu pada Form anda. Klik ganda komponen tersebut sehingga delphi akan menampilkan Menu Designer.

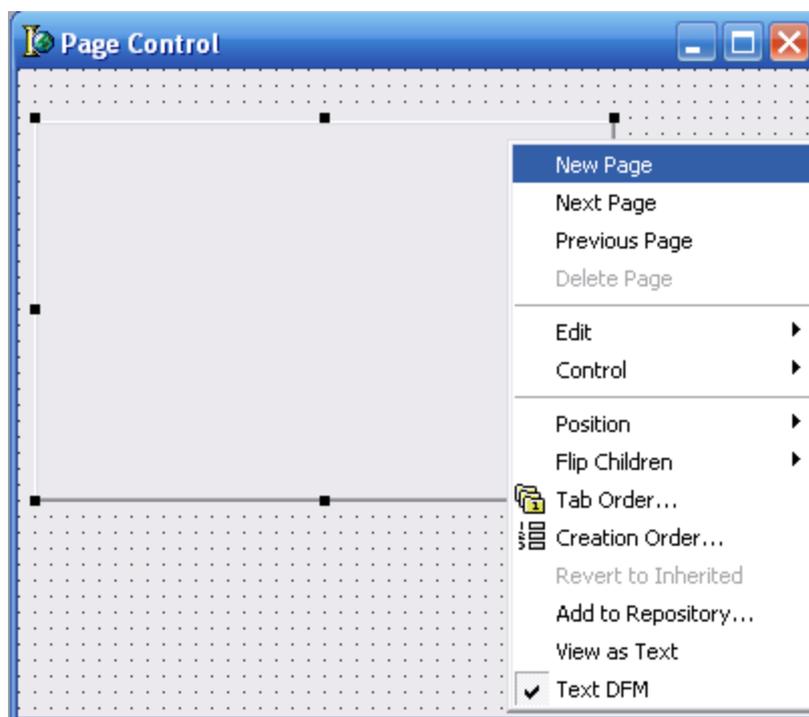
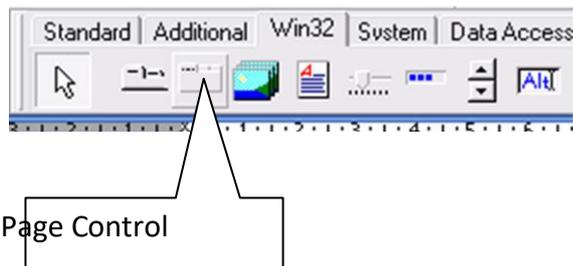


Perintah untuk menggunakan PopUpmenu juga hampir sama seperti desain pada TmainMenu, Cuma untuk bisa menjalankan PopupMenu kita harus mengaktifkan Properties PopupMenu pada Properties Formnya.



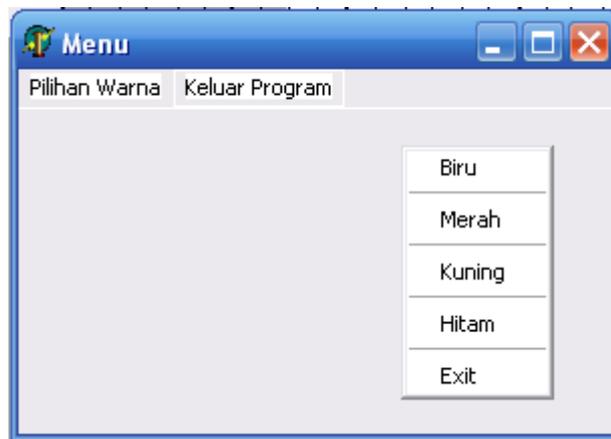
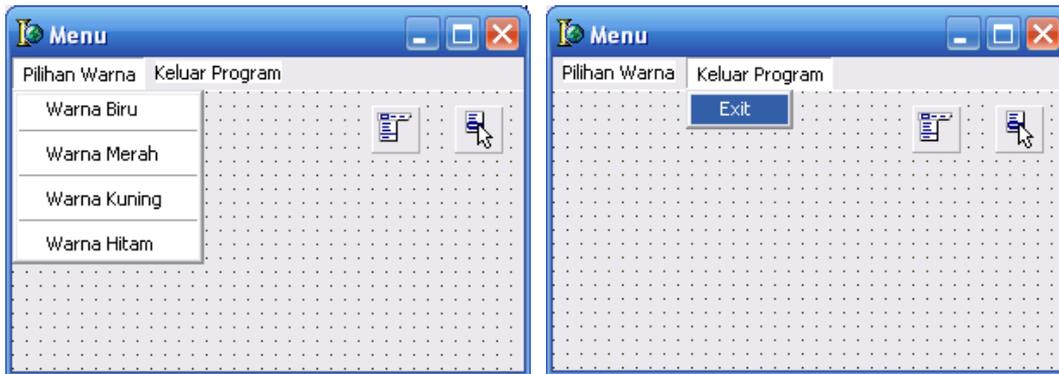
## Menggunakan Page Control

Page Control merupakan salah satu komponen yang ada pada Borland Delphi untuk membuat tampilan suatu Form bisa berganti dalam satu Fasilitas Form.



Untuk bisa menjalankan Page Control, masukkan komponen tersebut dan kemudian klik kanan pada komponen **Page Control – New Page**. Maka akan muncul tabsheet-tabsheet yang disesuaikan sesuai keinginan kita untuk memasukkan tabsheet-tabsheet tersebut.

1. Buatlah tampilan komponen ; **TmainMenu** dan **PopupMenu**.



**Keterangan :**

1. Menu Warna Biru saat anda Klik maka Form Berubah warna menjadi Biru
2. Menu Warna Merah saat anda Klik maka Form Berubah warna menjadi Merah
3. Menu Warna Kuning saat anda Klik maka Form Berubah warna menjadi Kuning
4. Menu Warna Hitam saat anda Klik maka Form Berubah warna menjadi Hitam
5. Menu Keluar Program akan muncul menu Exit yang berfungsi untuk keluar dari program.
6. Menu Popup akan merubah warna lebih cepat dengan cara klik kanan.

2. Buatlah tampilan komponen ; **Page Control**.

