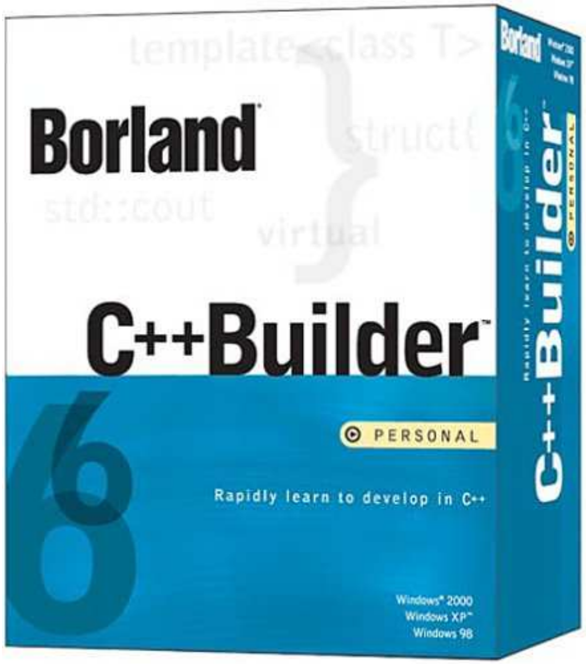
Algoritma Pemrograman

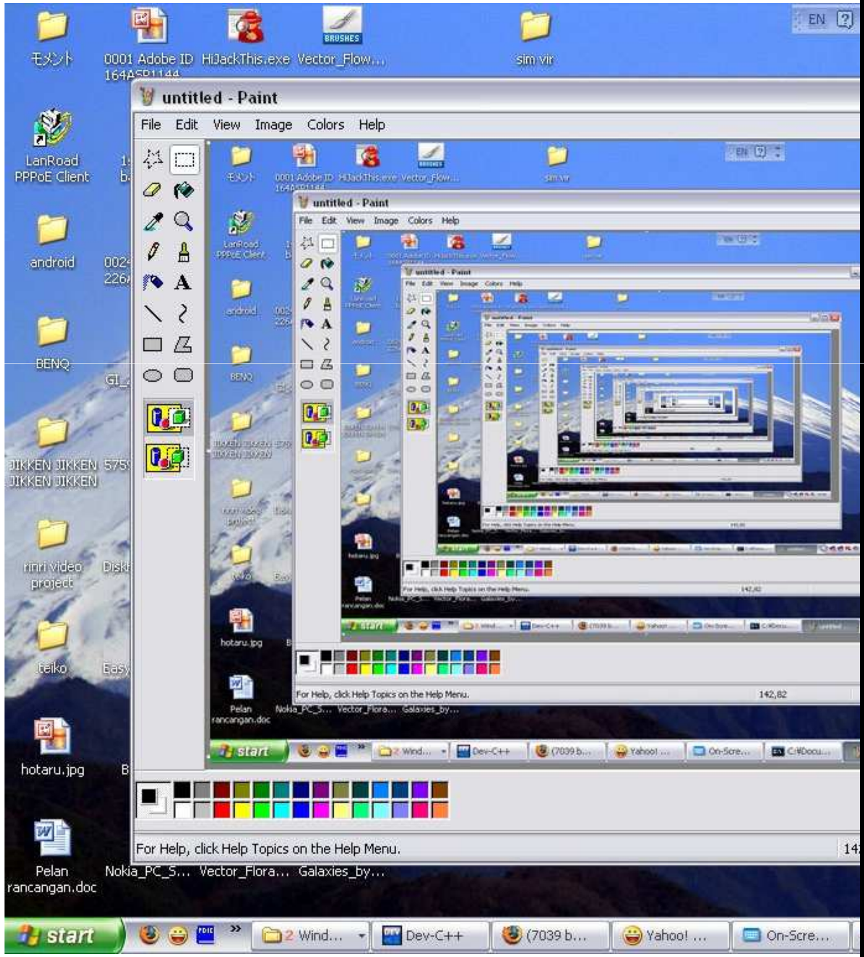
Rekursif & GUI Programming I

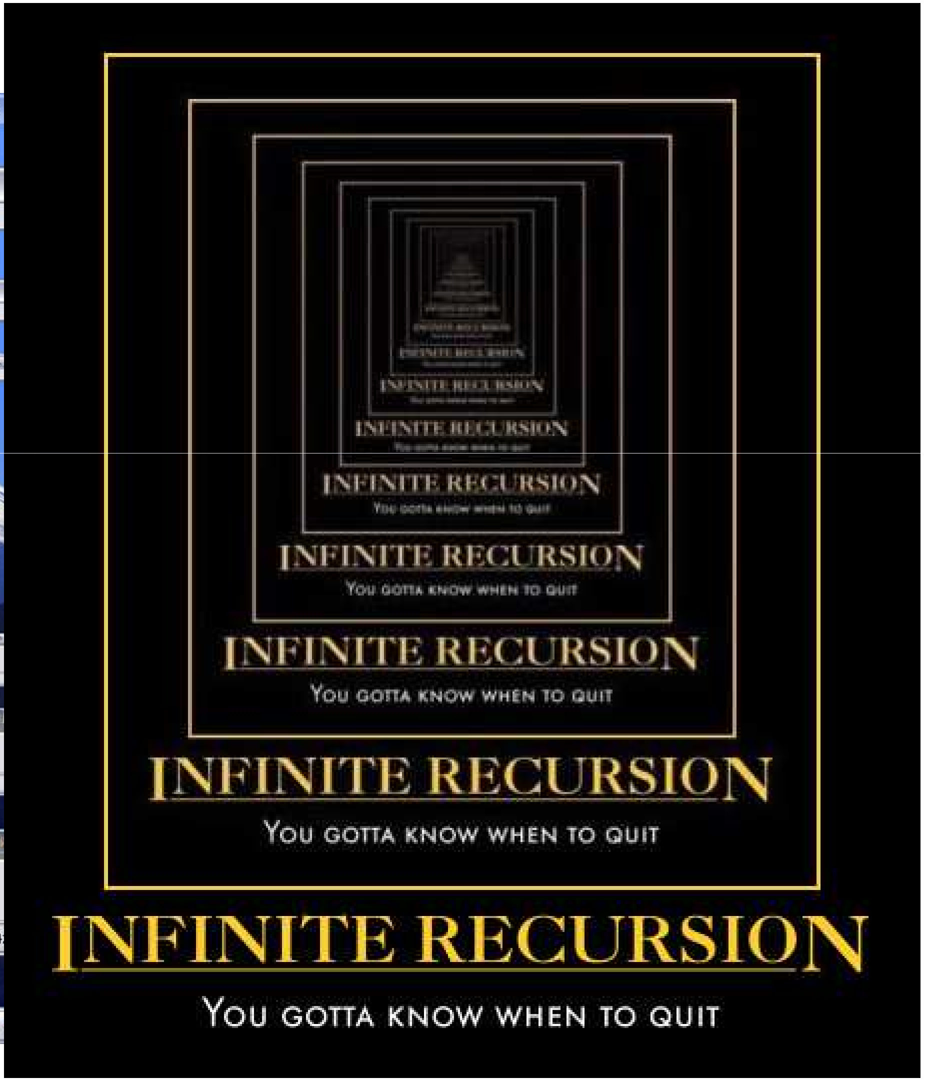


Tambahan: **Fungsi Rekursif**

- Fungsi yang berisi dirinya sendiri
- Fungsi yang mendefinisikan dirinya sendiri
- Fungsi yang memanggil dirinya sendiri
- Yang perlu diperhatikan adalah “aturan untuk berhenti dari perulangan tersebut”

Recursive


モメント
0001 Adobe ID 164ASD1144
HiJackThis.exe
Vector_Flow...
sim vir
EN
untitled - Paint
File Edit View Image Colors Help
LanRoad PPPoE Client
android
BENQ
JIKKEN JIKKEN JIKKEN JIKKEN
rinri video project
teiko
hotaru.jpg
Pelan rancangan.doc
Nokia_PC_S...
Vector_Flora...
Galaxies_by...
For Help, click Help Topics on the Help Menu.
start
2 Wind...
Dev-C++
(7039 b...
Yahoo! ...
On-Scre...
C:\Docu...

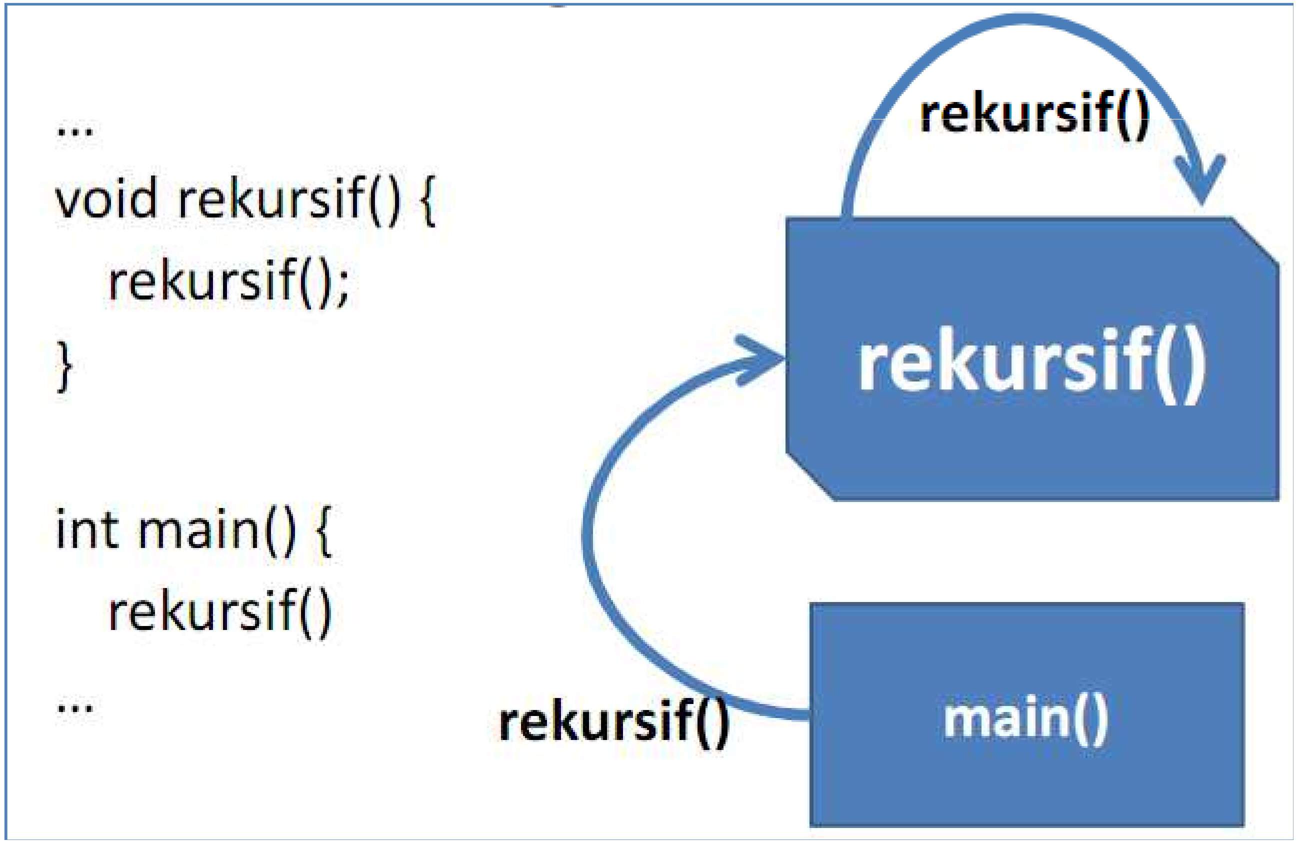

INFINITE RECURSION
You gotta know when to quit

Plus - Minus

- Karena program **lebih singkat** dan ada beberapa kasus yang lebih mudah menggunakan fungsi yang rekursif
- Memakan **memori yang lebih besar**, karena setiap kali bagian dirinya dipanggil, dibutuhkan sejumlah ruang memori tambahan.
- **Mengorbankan** efisiensi dan kecepatan
- **Problem**: rekursi seringkali tidak bisa “berhenti” sehingga memori akan habis dan komputer hang.
- **Saran**: jika memang bisa diselesaikan dengan **iteratif**, gunakanlah iteratif
- Jumlah maksimal tingkat rekursif bergantung pada compiler / sistem operasi

Bentuk Umum Fungsi Rekursif

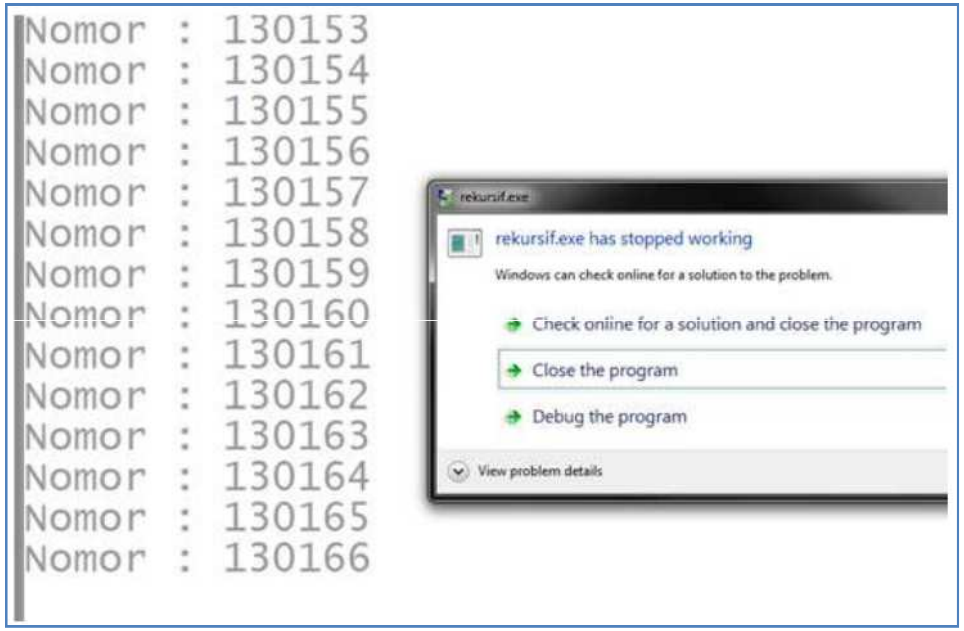
```
return_data_type function_name(parameter_list){
    ...
    function_name(...);
    ...
}
```



```
...
void rekursif() {
   rekursif();
}

int main() {
   rekursif()
...
```

Contoh Fungsi Rekursif

```
#include <stdio.h>
#include <conio.h>

void rekursif(int nomor);

int main() {
   rekursif(1);
   return 0;
}

void rekursif(int nomor) {
    printf("Nomor : %d\n", nomor);
    rekursif(nomor+1);
}
```



```
Nomor  :  130153
Nomor  :  130154
Nomor  :  130155
Nomor  :  130156
Nomor  :  130157
Nomor  :  130158
Nomor  :  130159
Nomor  :  130160
Nomor  :  130161
Nomor  :  130162
Nomor  :  130163
Nomor  :  130164
Nomor  :  130165
Nomor  :  130166
```

rekursif.exe has stopped working

Windows can check online for a solution to the problem.

Faktorial

- Faktorial

  5! = 5 x 4 x 3 x 2 x 1

  4! = 4 x 3 x 2 x 1

  Berarti 5! = 5 x 4!

Metode Iteratif

Salah satu cara untuk menghitung adalah dengan menggunakan loop, yang mengalikan masing-masing bilangan dengan hasil sebelumnya.

- Penyelesaian dengan cara ini dinamakan iteratif, yang mana secara umum dapat didefinisikan sebagai berikut:
- *n! = (n)(n-1)(n-2) …*

Program Iteratif

```
#include <stdio.h>
int fact_it (int n)
{
   int i,fak;

   for (i=1; i<=n; i++)
     fak = fak * i;
   return (fak);
}
int main()
{
   int fac;
   printf("Masukkan berapa faktorial : ");
   scanf("%d",&fac);
   printf("Hasil faktorial dari adalah : %d\n", fact_it(fac));
    return 0;
}
```

Faktorial Rekursif (2)

- n! = n*(n-1)!
- 0! = 1
- 1! = 1*(1-1)!

  = 1

  = 1

  = 1
- 2! = 2

  = 2

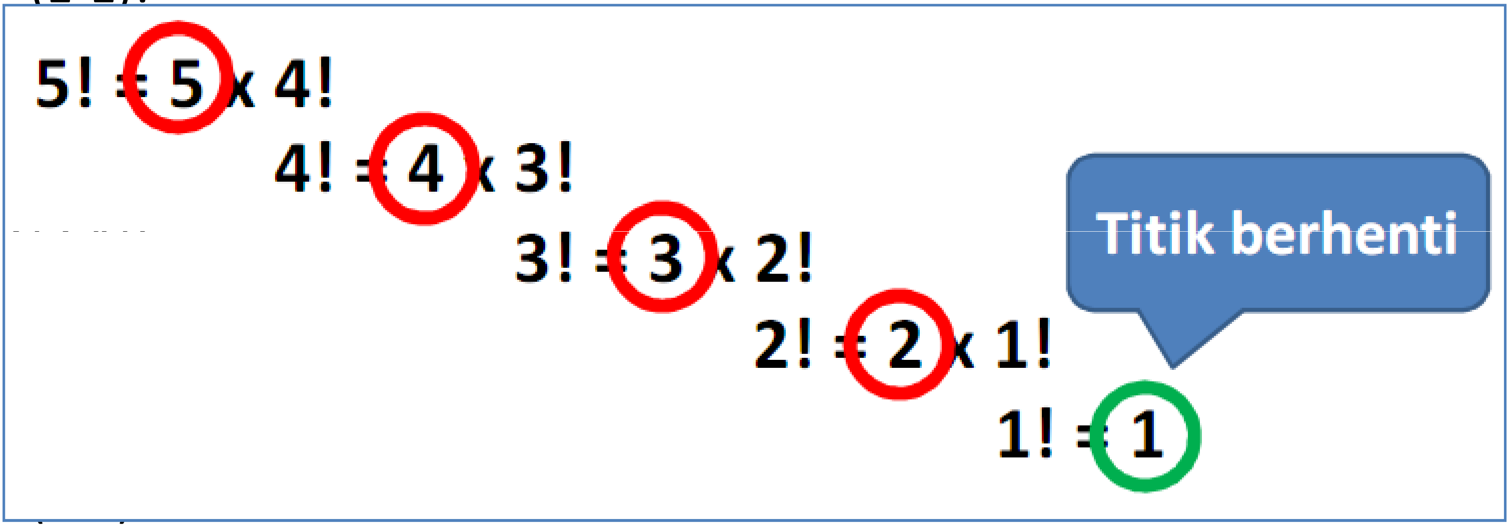
  = 2

  = 2
- 3! = 3

  = 3*2!

  = 3*2

  = 6



Program Rekursif

```
#include <stdio.h>
int fact_rec(int n)
{
    if (n < 0)
        return 0;
    else if (n == 0)
        return 1;
    else if (n == 1)
        return 1;
    else
        return n * fact_rec(n-1);
}
int main()
{
    int fac;
    printf("Masukkan berapa faktorial : ");
    scanf("%d",&fac);
    printf("Hasil faktorial dari adalah : %d\n",
    fact_rec(fac));
    return 0;
}
```

Fibonacci

- Deret Fibonacci adalah suatu deret matematika yang berasal dari penjumlahan dua bilangan sebelumnya.
- 1, 1, 2, 3, 5, 8, 13, 21, …

Fibo Iteratif

- **Secara iteratif**

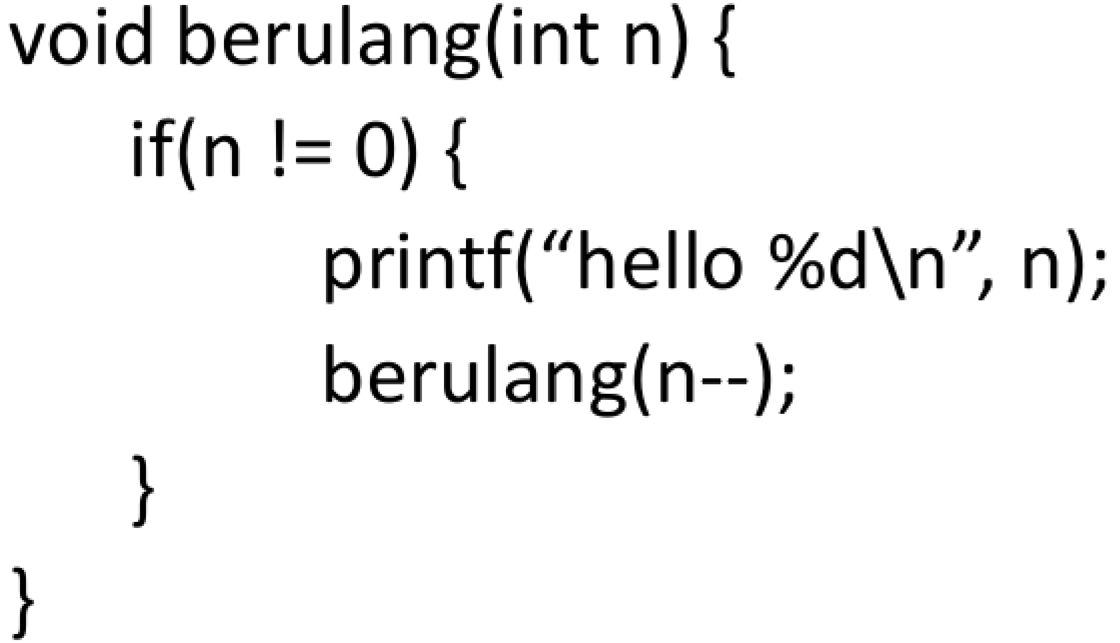
```
int fibonacci(int n){
    int f1=1, f2=1, fibo;
    if(n==1 || n==2) fibo=1;
    else{
      for(int i=2;i<=n;i++){
            fibo = f1 + f2;
            f1 = f2;
            f2 = fibo;
       }
    }
   return fibo;
}
```

Fibo Rekursif

```
int fibo_r (int n){
  if(n==1) return 1;
  else if(n==2) return 1;
  else return fibo_r(n-1) + fibo_r(n-
  2);
}
```

Latihan

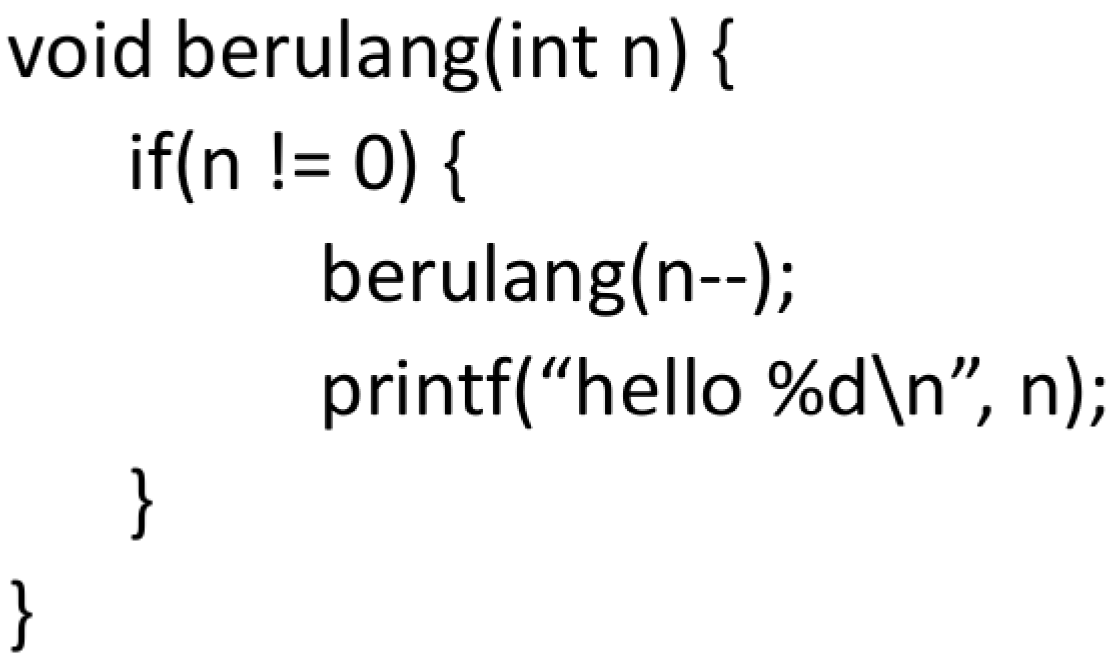
- Apa keluaran dari fungsi rekursif berikut bila dipanggil dengan berulang(4);



```
void berulang(int n) {
      if(n != 0) {
            printf(“hello %d\n”, n);
            berulang(n--);
      }
}
```

Latihan

- Apa keluaran dari fungsi rekursif berikut bila dipanggil dengan berulang(4);



```
void berulang(int n) {
    if(n != 0) {
        berulang(n--);
        printf(“hello %d\n”, n);
    }
}
```

GUI Programming

- Pemrograman berbasis **user interface**
  - Pemrograman dilakukan diatas **FORM**
  - Kadang ada yang menyebut pemrograman Visual
    - Itu SALAH!
- Menggunakan **GUI Editor dan IDE**!
  - Menyediakan tool terintegrasi:
    - Compile dan Run, Debugging, koneksi dengan database
    - Penggunaan komponen visual n non visual

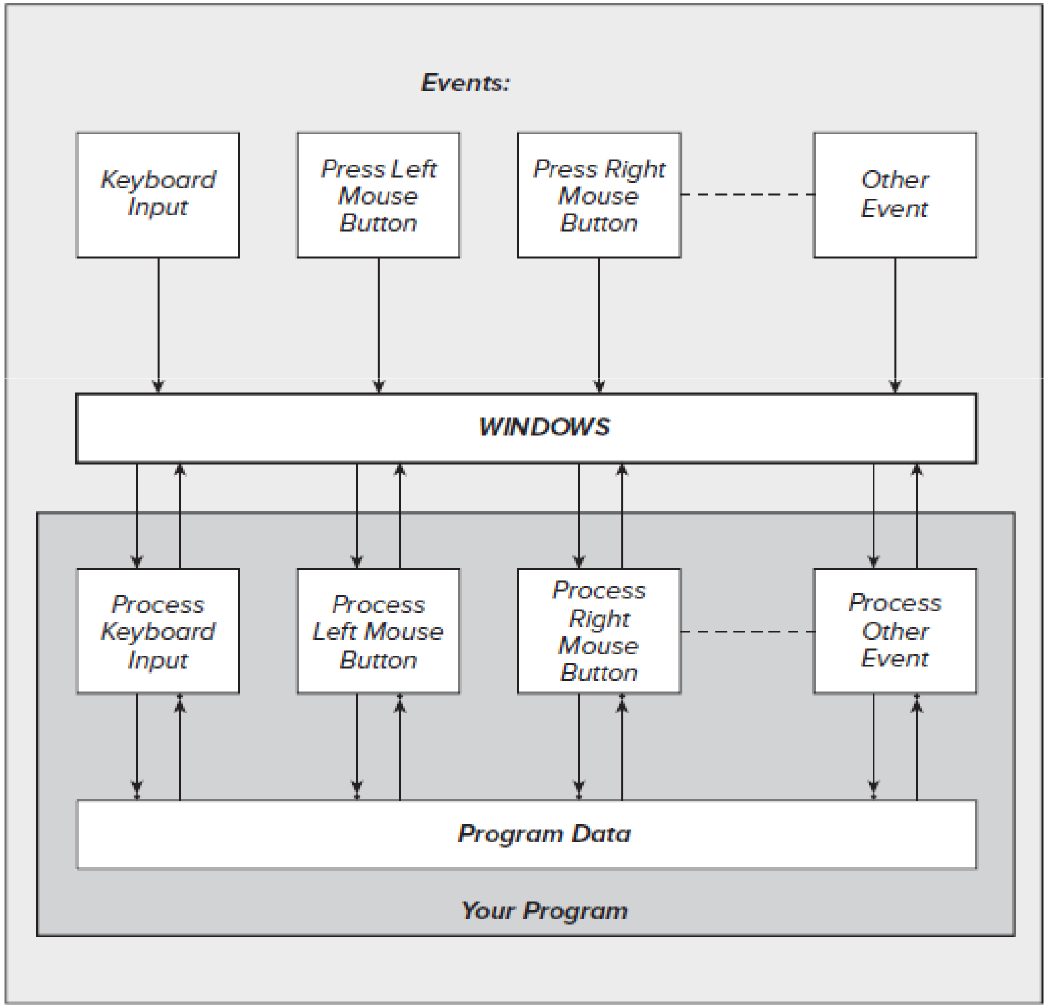
Event oriented programming

- **conventional** (request-response) programming:
  - sequence of operations is determined by the **program**
  - what you want to happen, happens when you want it
- **event-oriented** programming:
  - sequence of operations is determined by the **user's interaction** with the application's interface
  - anything that can happen, happens at **any time**

Event driven programming

- **Normal (control flow-based) programming Approach**
  - Start at **main**()
  - Continue until end of program or **exit**()
- **Event-driven programming**
  - **Unable to predict time & occurrence of event**
  - **Approach**
    - Start with **main**()
    - Build GUI
    - Await **events** (& perform associated computation)

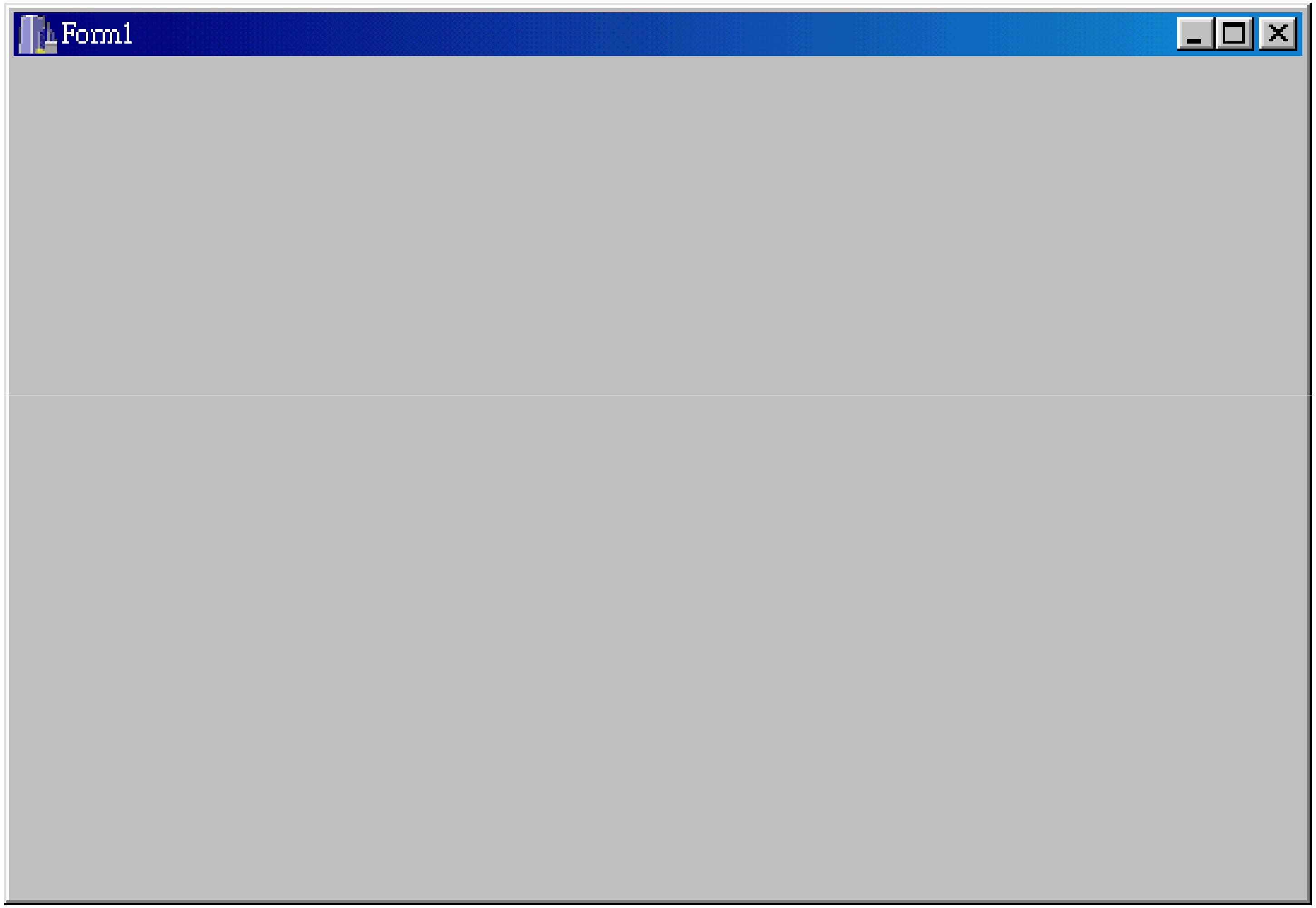
GUI Programming Event Driven



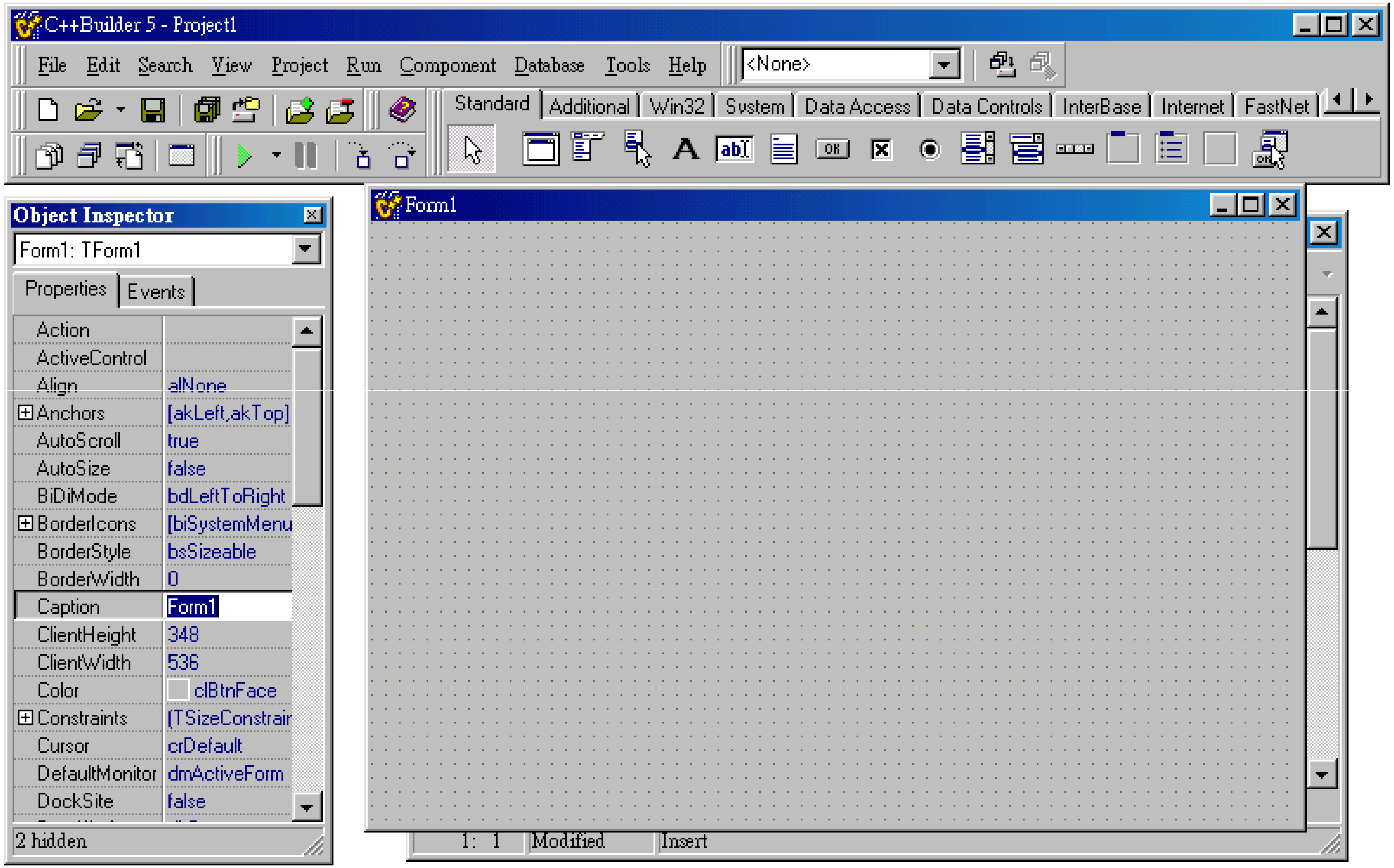
Borland C++ Builder

- **C++Builder** is a rapid application development (RAD) environment, developed by Borland and as of 2009, owned by the **CodeGear** subsidiary of **Embarcadero Technologies**, for writing programs in the C++ programming language
- In 2003 Borland released ***C++BuilderX*** (CBX), which was written using the same framework as **JBuilder** and bore little resemblance to either C++Builder or **Delphi**

First BCB GUI Program



BC++ Builder Environment

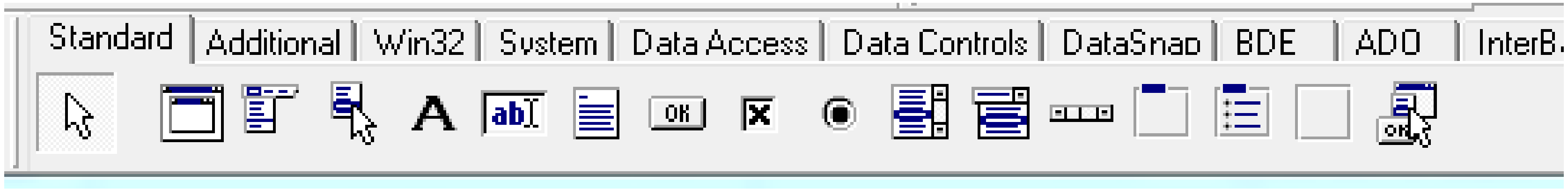


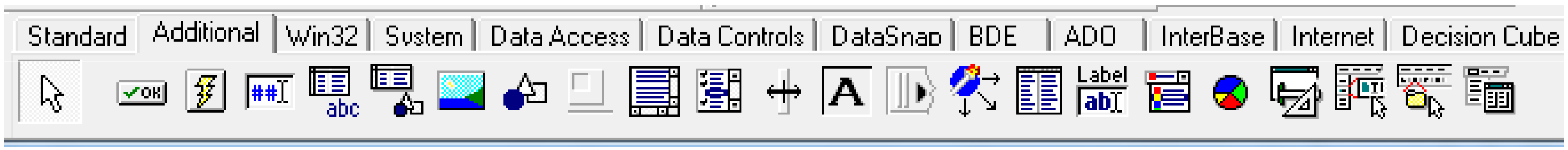
Form1: TForm1

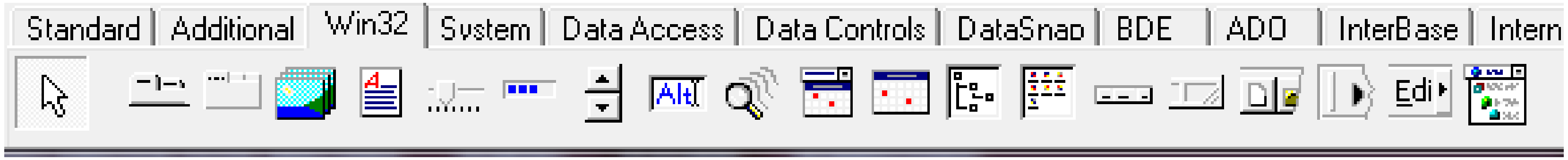
Action	
ActiveControl	
Align	alNone
⊞Anchors	[akLeft,akTop]
AutoScroll	true
AutoSize	false
BiDiMode	bdLeftToRight
⊞BorderIcons	[biSystemMenu
BorderStyle	bsSizeable
BorderWidth	0
Caption	Form1
ClientHeight	348
ClientWidth	536
Color	□ clBtnFace
⊞Constraints	(TSizeConstrair
Cursor	crDefault
DefaultMonitor	dmActiveForm
DockSite	false

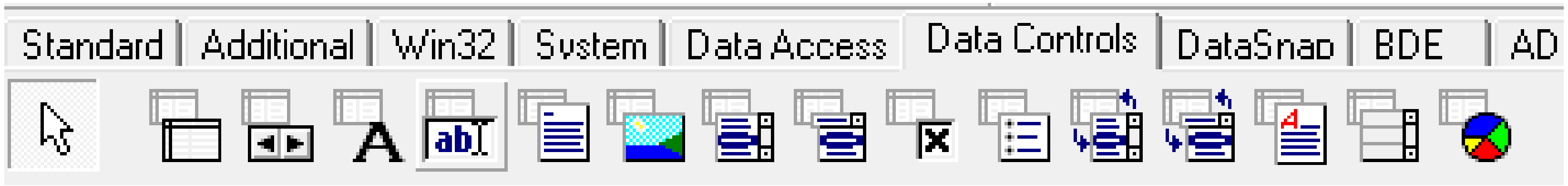
2 hidden

Component Pallete

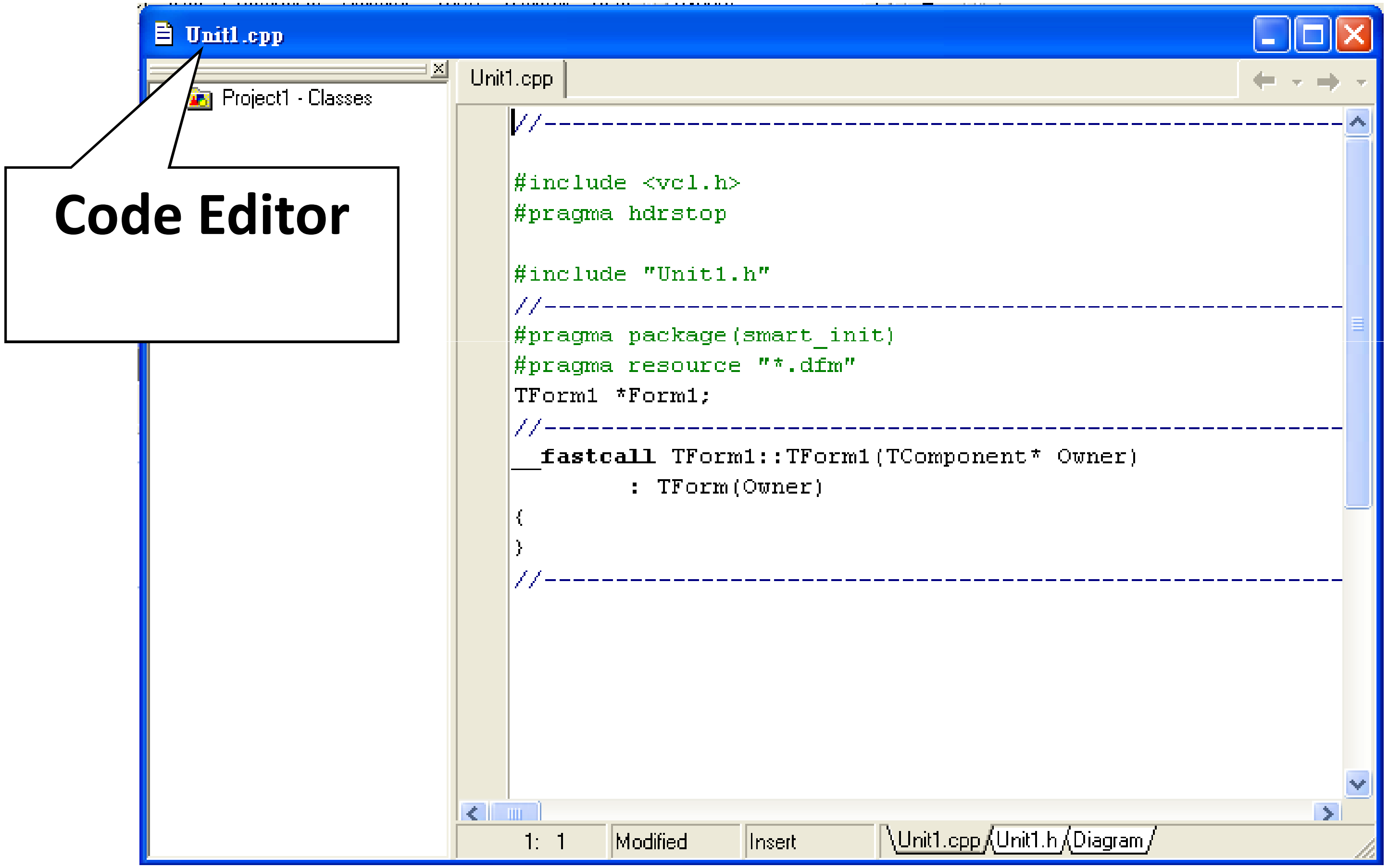








Code Editor

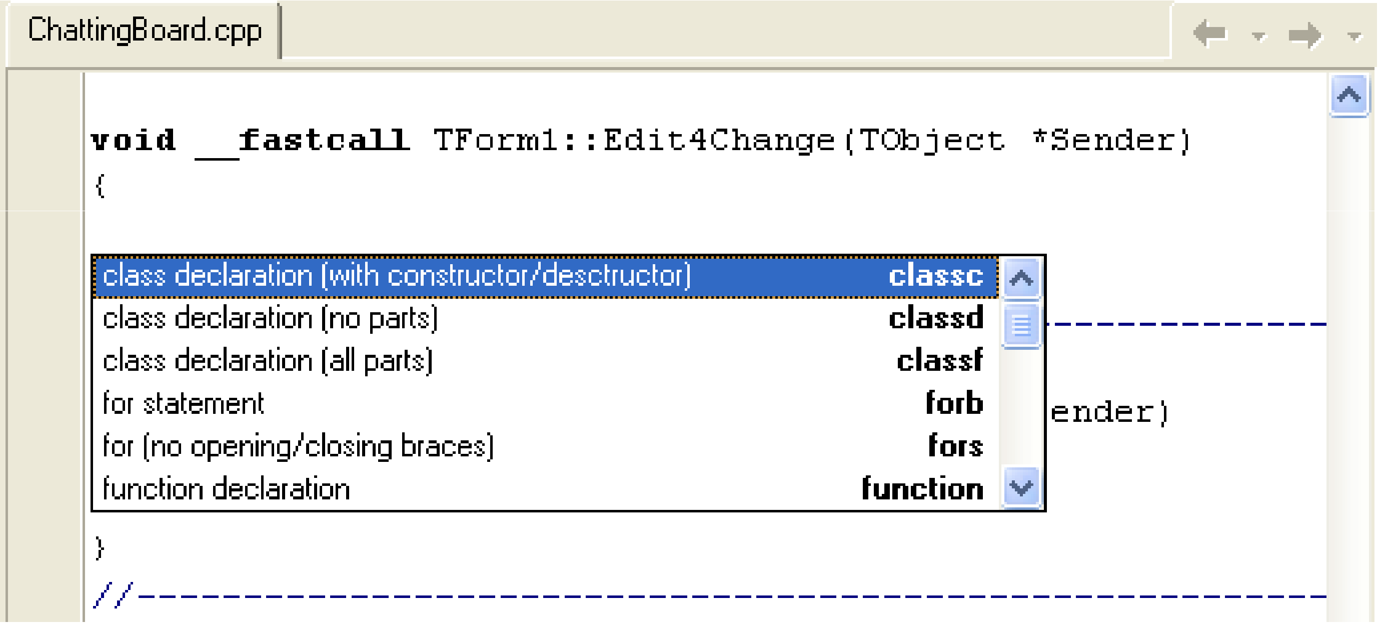


Code Editor

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
```

Code Editor Tip

- Code Templates [Ctrl+J]

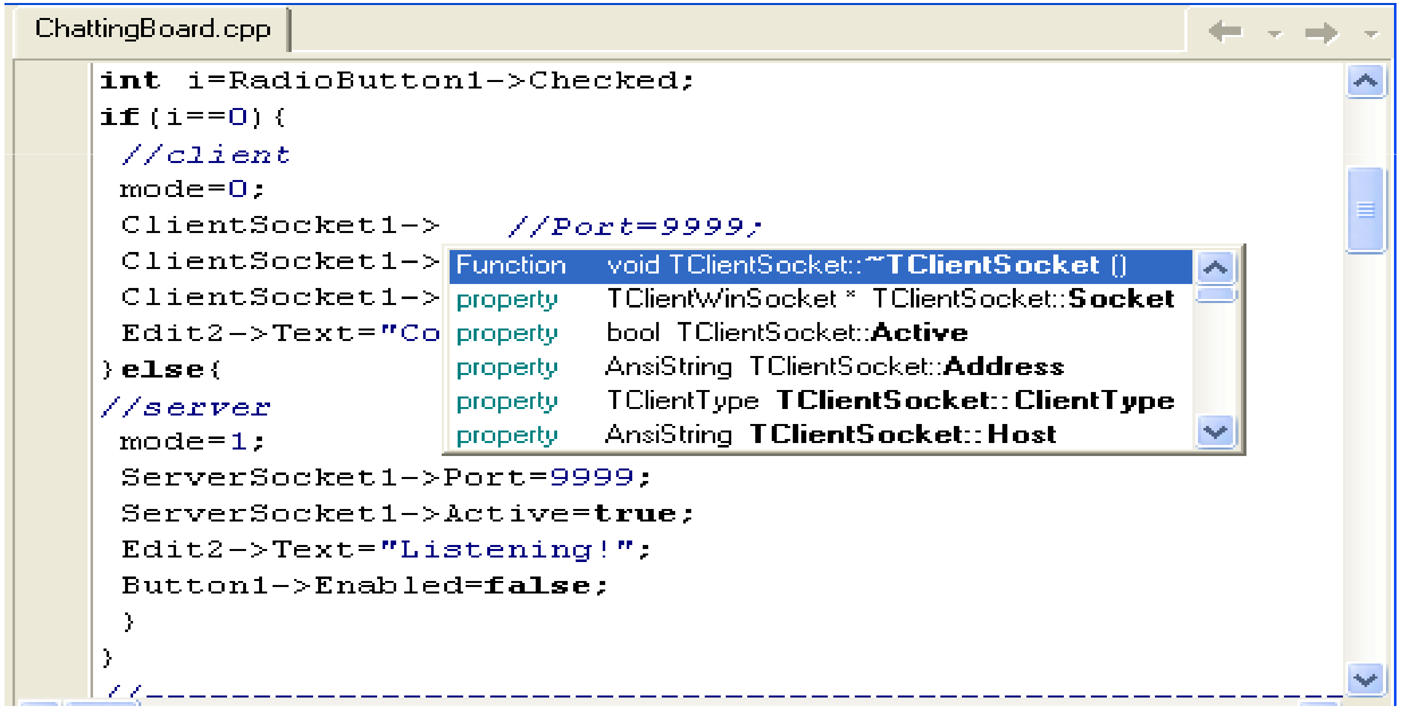


```
void __fastcall TForm1::Edit4Change(TObject *Sender)
{

}
//-------------------------------------------------------
```

class declaration (with constructor/desctructor)	classc
class declaration (no parts)	classd
class declaration (all parts)	classf
for statement	forb
for (no opening/closing braces)	fors
function declaration	function

Code Editor Tip

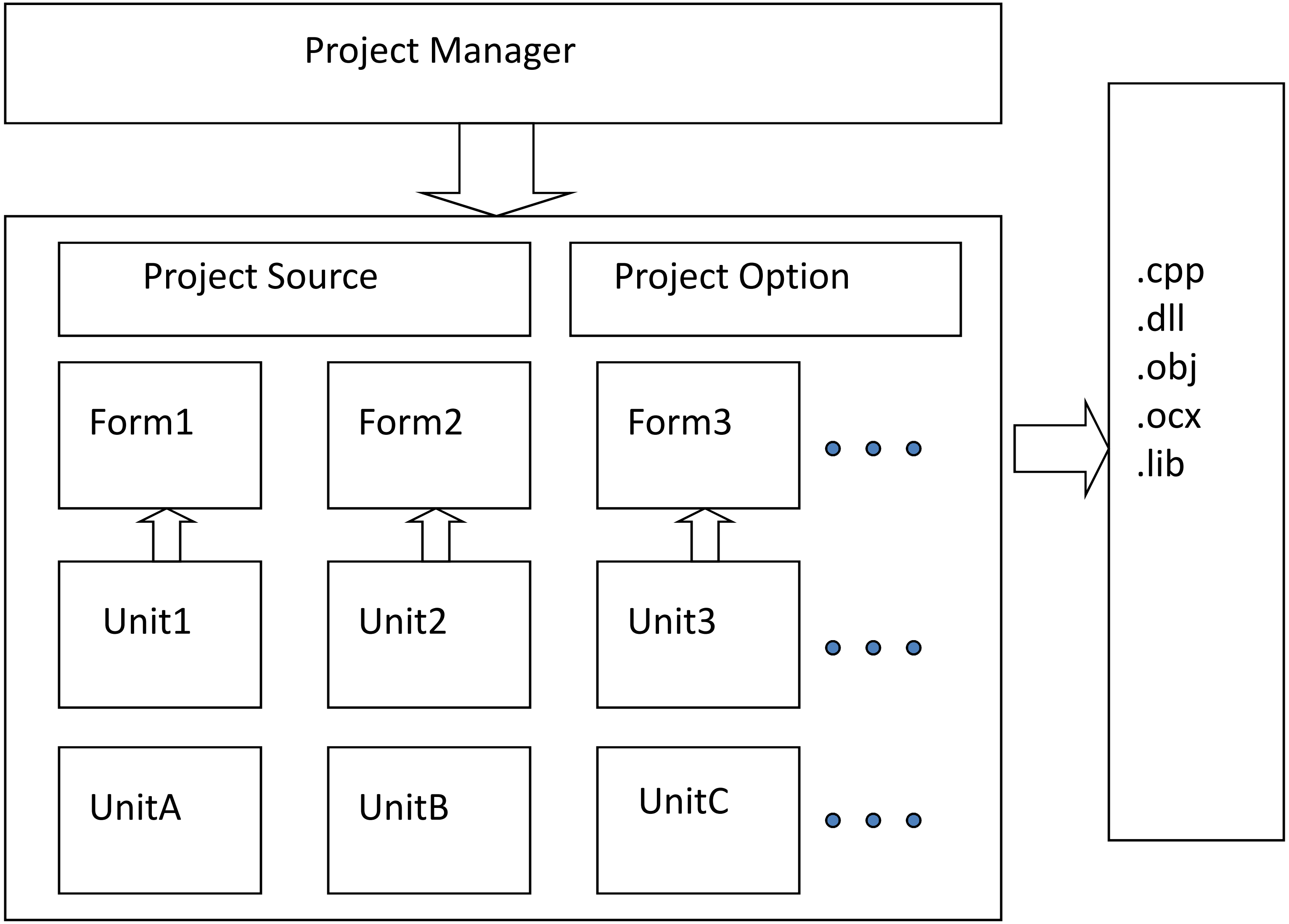
- Function Parameters
- Code Completion



```
int  i=RadioButton1->Checked;
if(i==0){
 //client
 mode=0;
 ClientSocket1->    //Port=9999;
 ClientSocket1->
 ClientSocket1->
 Edit2->Text="Co
}else{
//server
 mode=1;
 ServerSocket1->Port=9999;
 ServerSocket1->Active=true;
 Edit2->Text="Listening!";
 Button1->Enabled=false;
 }
}
//-----------------------------------------------
```

```
Function    void TClientSocket::~TClientSocket ()
property    TClientWinSocket * TClientSocket::Socket
property    bool TClientSocket::Active
property    AnsiString TClientSocket::Address
property    TClientType TClientSocket::ClientType
property    AnsiString TClientSocket::Host
```

BC++ Project Structure



Project Manager

Project Source

Project Option

Form1

Form2

Form3

Unit1

Unit2

Unit3

UnitA

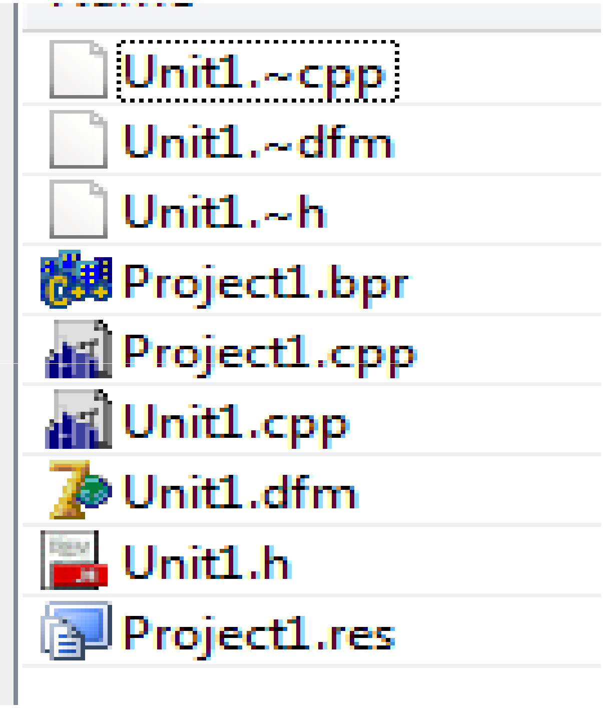
UnitB

UnitC

.cpp
.dll
.obj
.ocx
.lib

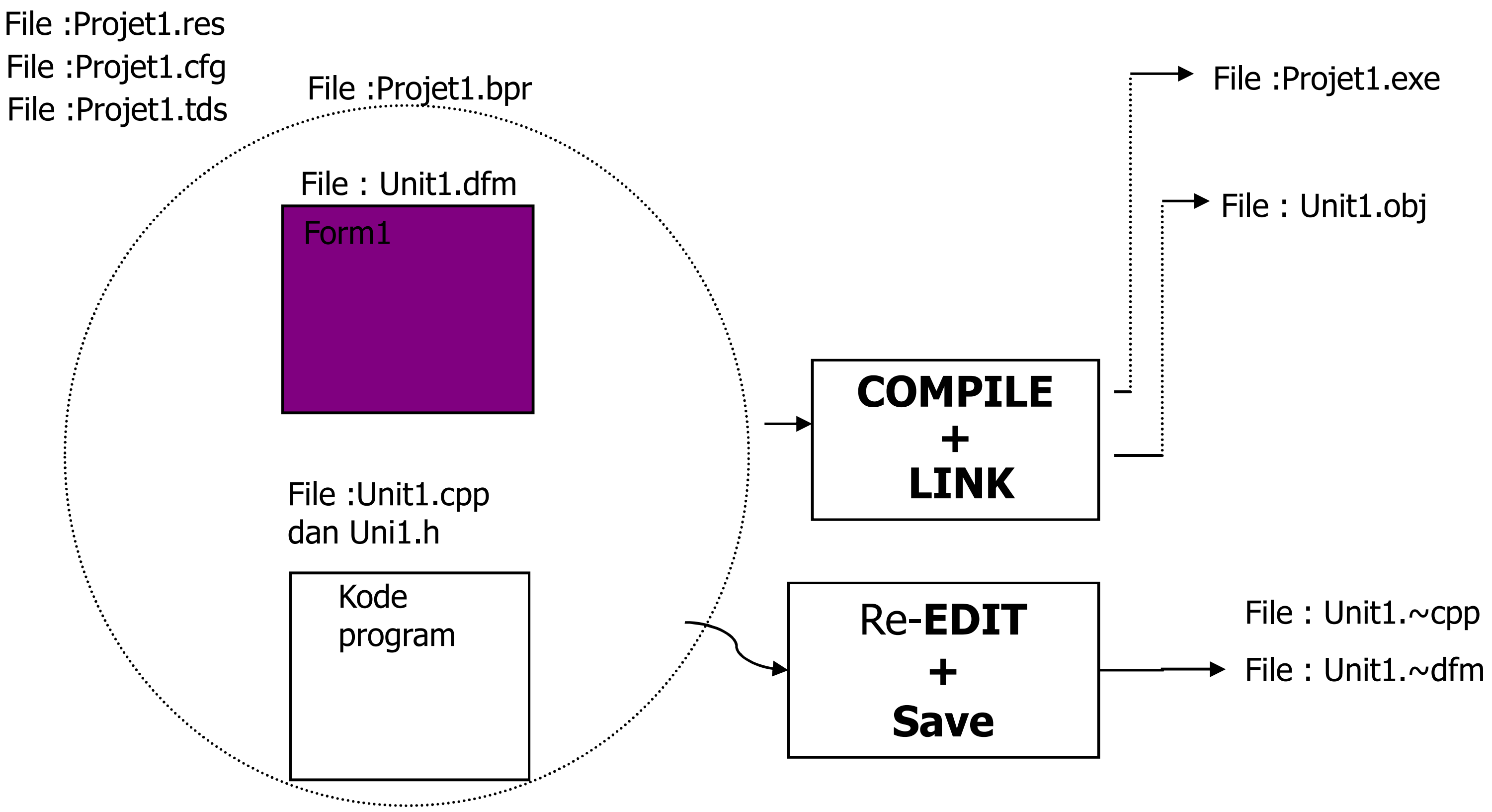
First BCB Project

- Create a file folder **FirstBCBProject**
- Save Project As ... | Project1
- Generated files:
  - Project1.bpr
  - Project1.cpp
  - Project1.res
  - Project1.tds
  - Unit1.dfm
  - Unit1.cpp
  - Unit1.h
  - File-file backup: .~cpp, .~dfm, .~h
  - File hasil kompilasi: .obj, .ddp



Unit1.~cpp
Unit1.~dfm
Unit1.~h
Project1.bpr
Project1.cpp
Unit1.cpp
Unit1.dfm
Unit1.h
Project1.res

File-file aplikasi BCB satu Form



File :Projet1.res

File :Projet1.cfg

File :Projet1.tds

File :Projet1.bpr

File : Unit1.dfm

Form1

File :Unit1.cpp
dan Uni1.h

Kode
program

COMPILE
+
LINK

File :Projet1.exe

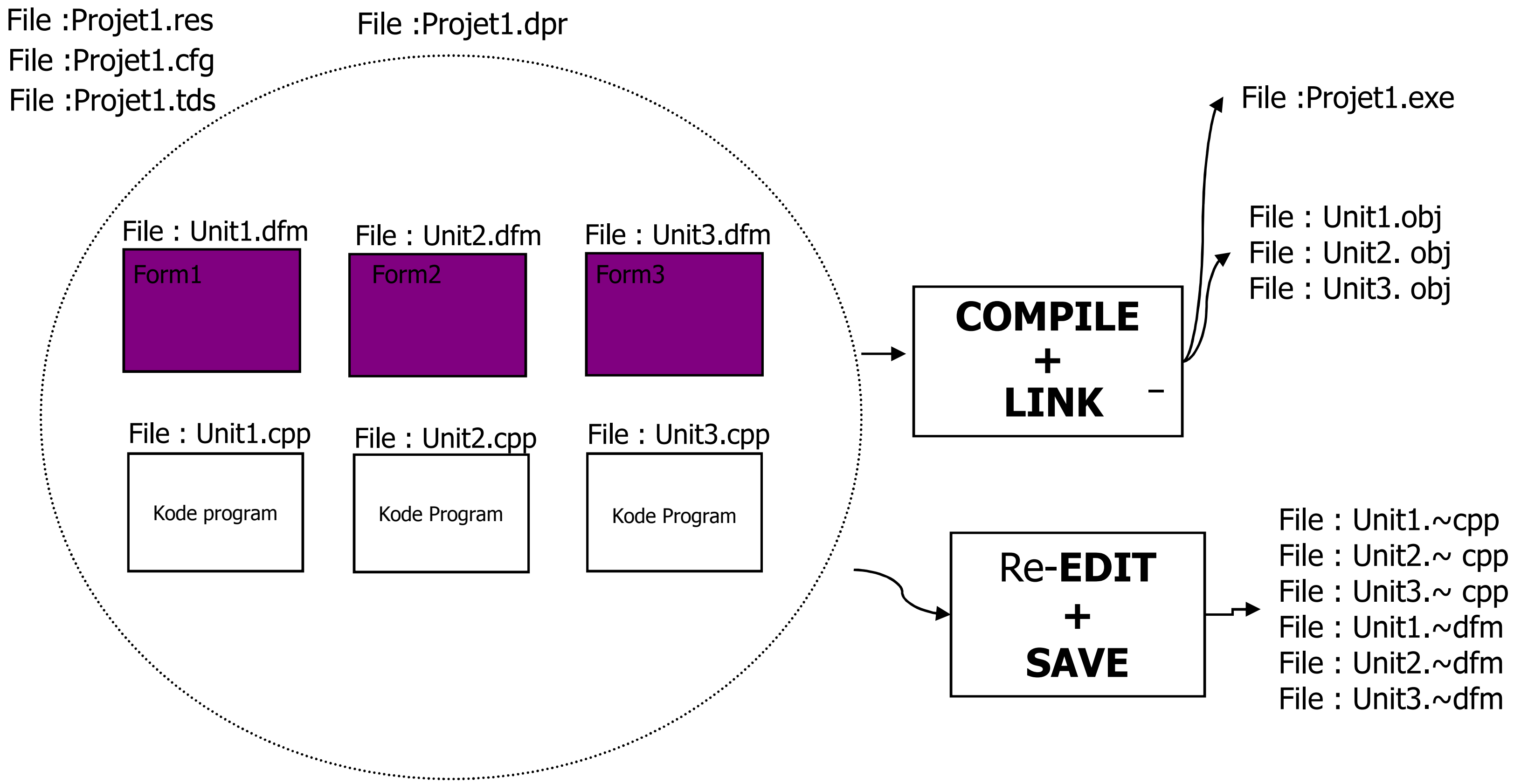
File : Unit1.obj

Re-**EDIT**
+
Save

File : Unit1.~cpp

File : Unit1.~dfm

File-file aplikasi BCB MultiForm



File :Projet1.res
File :Projet1.cfg
File :Projet1.tds

File :Projet1.dpr

File : Unit1.dfm

Form1

File : Unit2.dfm

Form2

File : Unit3.dfm

Form3

File : Unit1.cpp

Kode program

File : Unit2.cpp

Kode Program

File : Unit3.cpp

Kode Program

COMPILE
+
LINK

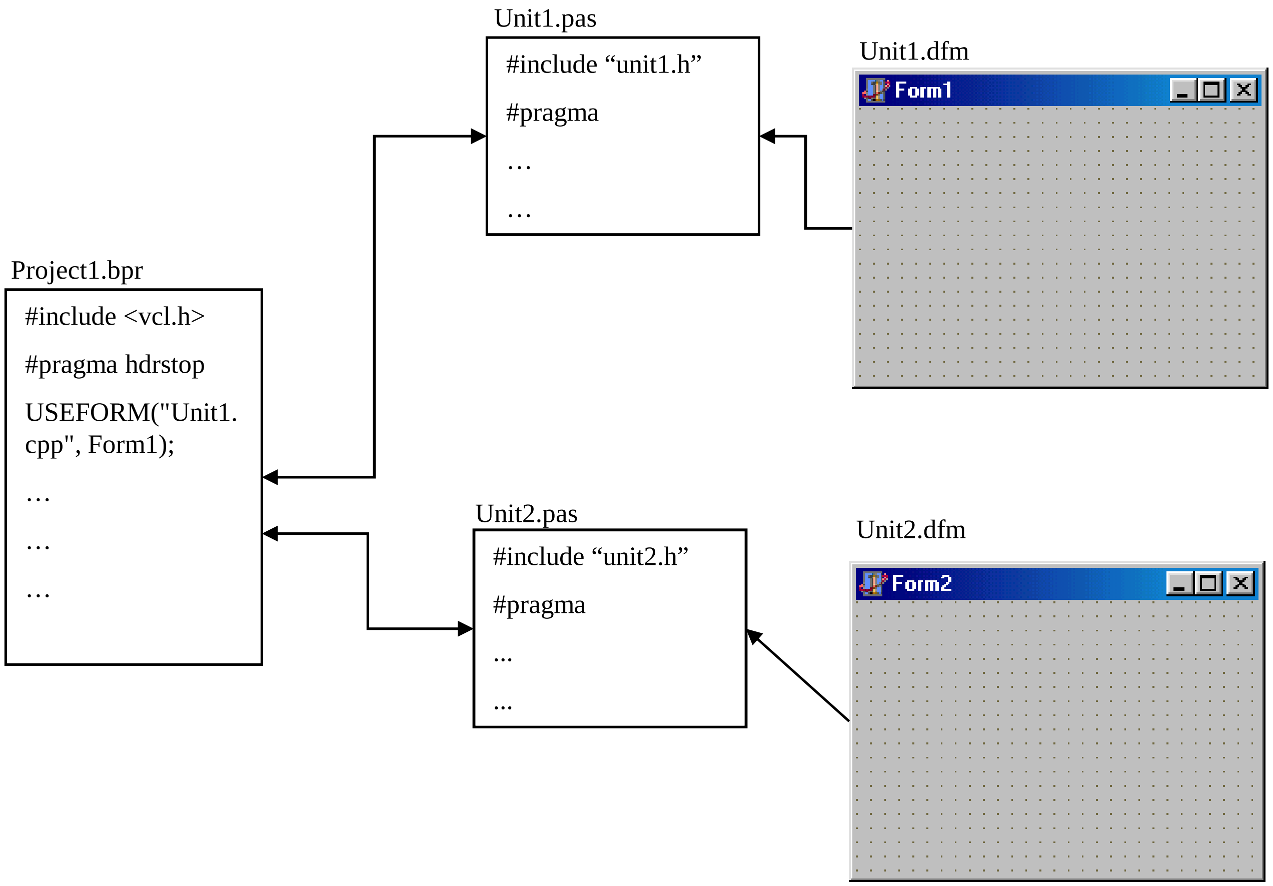
File :Projet1.exe

File : Unit1.obj
File : Unit2. obj
File : Unit3. obj

Re-**EDIT**
+
SAVE

File : Unit1.~cpp
File : Unit2.~ cpp
File : Unit3.~ cpp
File : Unit1.~dfm
File : Unit2.~dfm
File : Unit3.~dfm

Bagaimana kode program BCB dijalankan



Project1.bpr

```
#include <vcl.h>
#pragma hdrstop
USEFORM("Unit1.
cpp", Form1);
…
…
…
```

Unit1.pas

```
#include “unit1.h”
#pragma
…
…
```

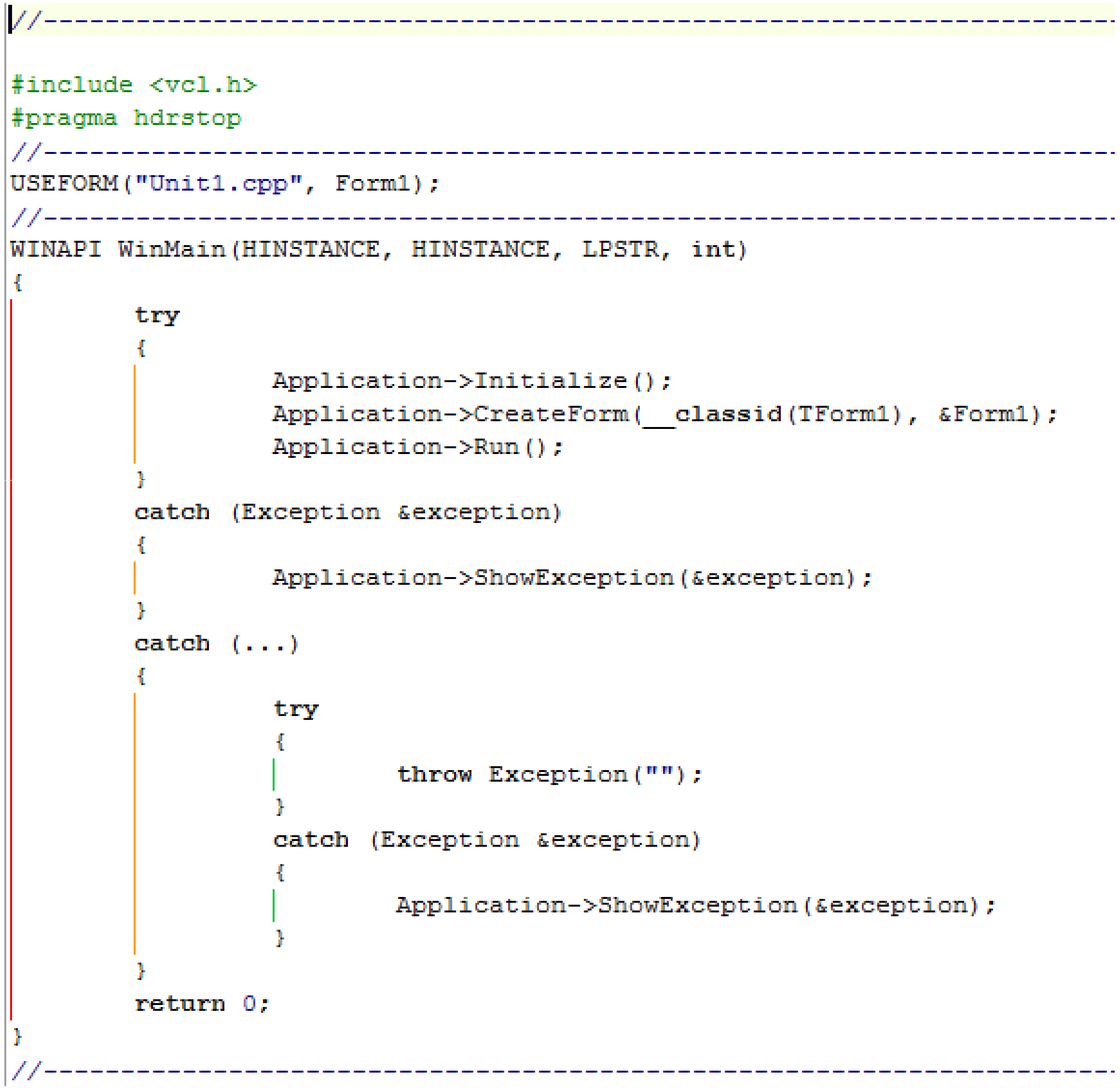
Unit1.dfm

Unit2.pas

```
#include “unit2.h”
#pragma
...
...
```

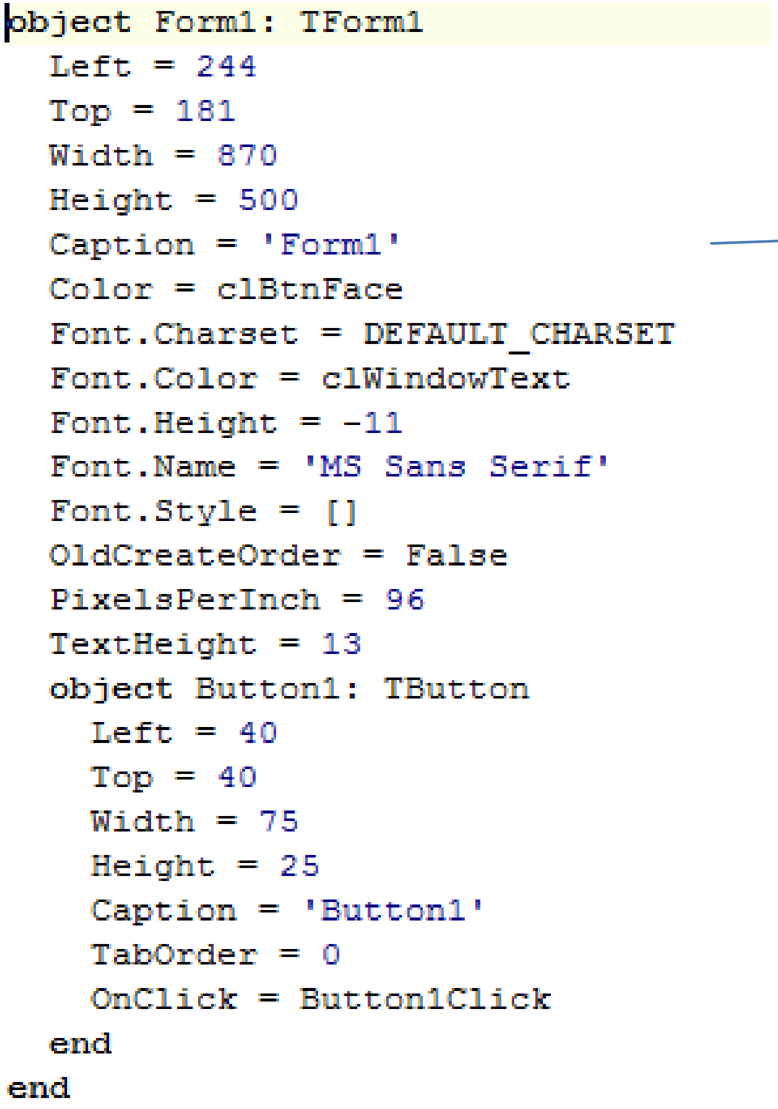
Unit2.dfm

Project Source – File Project1.cpp



```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
//---------------------------------------------------------------------------
USEFORM("Unit1.cpp", Form1);
//---------------------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                 Application->Initialize();
                 Application->CreateForm(__classid(TForm1), &Form1);
                 Application->Run();
        }
        catch (Exception &exception)
        {
                 Application->ShowException(&exception);
        }
        catch (...)
        {
                 try
                 {
                         throw Exception("");
                 }
                 catch (Exception &exception)
                 {
                         Application->ShowException(&exception);
                 }
        }
        return 0;
}
//---------------------------------------------------------------------------
```

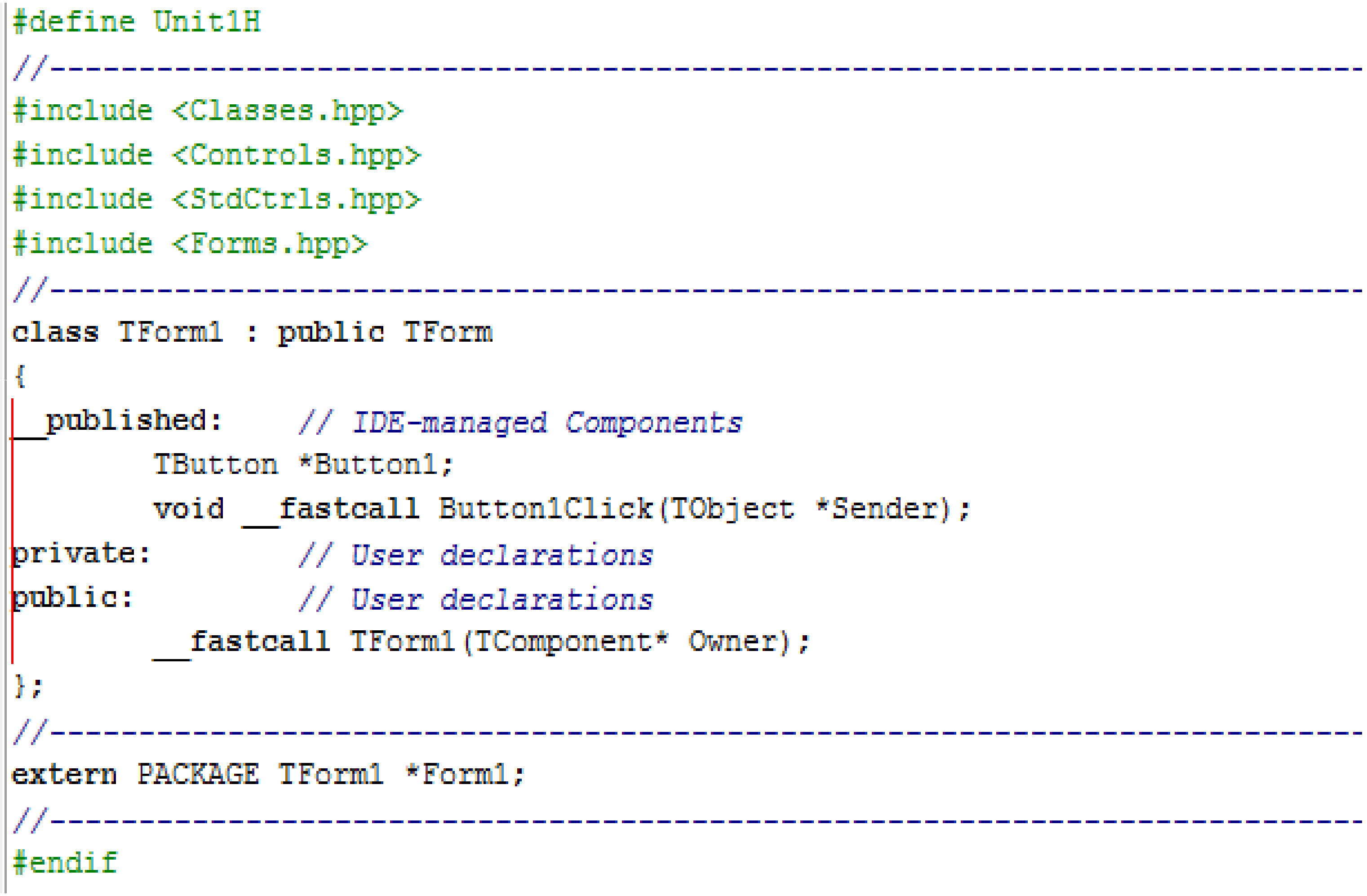
File Unit1.dfm



```
object Form1: TForm1
  Left = 244
  Top = 181
  Width = 870
  Height = 500
  Caption = 'Form1'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object Button1: TButton
    Left = 40
    Top = 40
    Width = 75
    Height = 25
    Caption = 'Button1'
    TabOrder = 0
    OnClick = Button1Click
  end
end
```

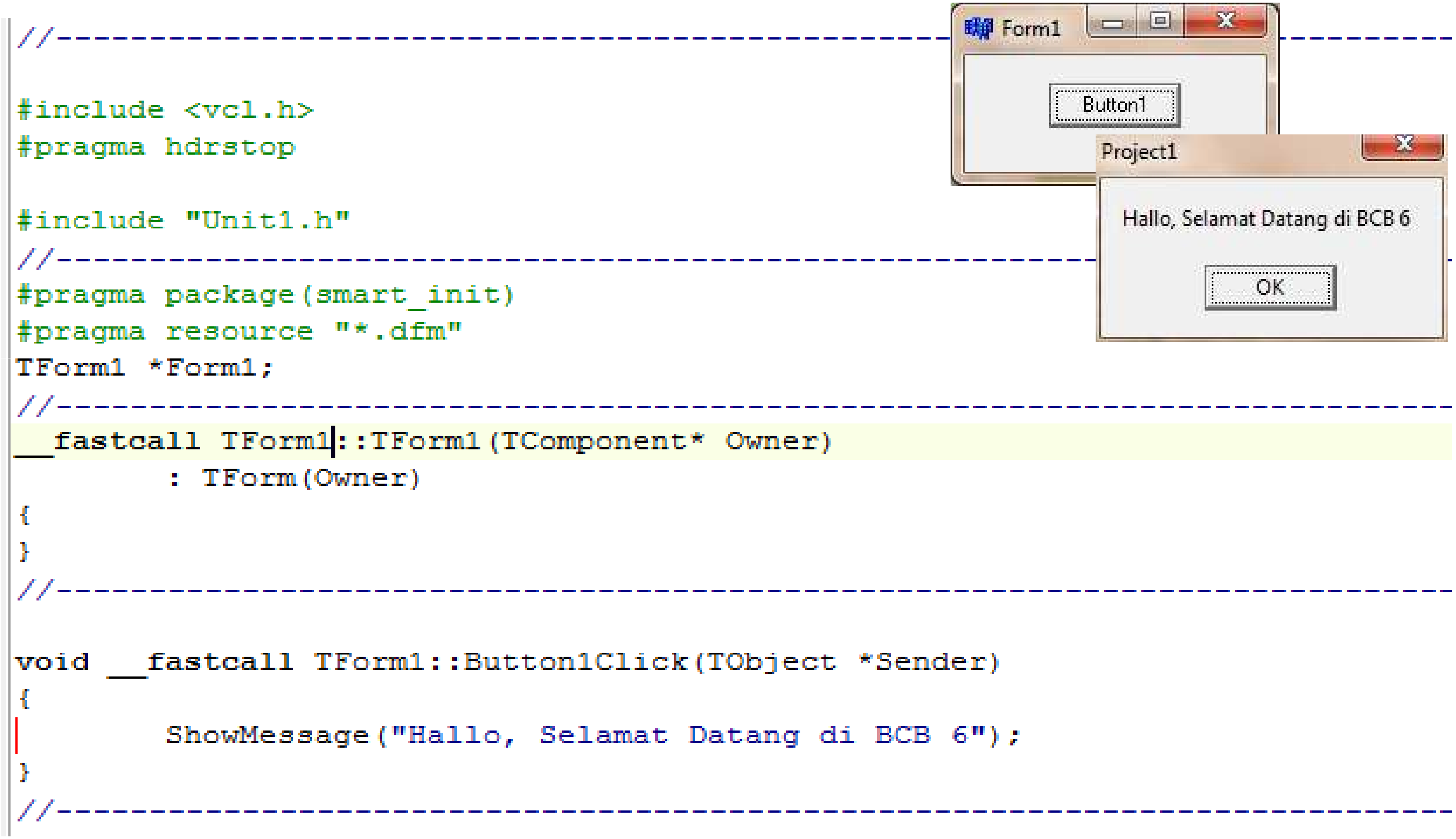
View as Text

File Unit1.h



```
#define Unit1H
//--------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//--------------------------------------------------------------------------
class TForm1 : public TForm
{
__published:    // IDE-managed Components
        TButton *Button1;
        void __fastcall Button1Click(TObject *Sender);
private:        // User declarations
public:         // User declarations
        __fastcall TForm1(TComponent* Owner);
};
//--------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;
//--------------------------------------------------------------------------
#endif
```

File Unit1.cpp : “Hello World”



```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button1Click(TObject *Sender)
{
        ShowMessage("Hallo, Selamat Datang di BCB 6");
}
//---------------------------------------------------------------------------
```

Hallo, Selamat Datang di BCB 6

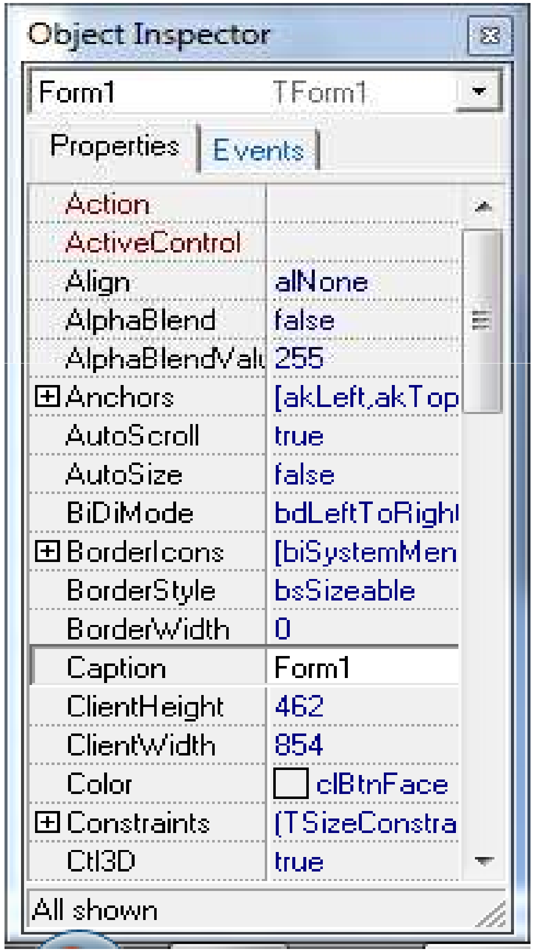
Visual Component Library

- Based on the **properties, methods, and events** (PME) model.
- The PME model defines the data members (**properties**), the functions that operate on the data (**methods**), and a way to interact with users of the class (**events**).
- A hierarchy of objects, written in Object Pascal and tied to the C++Builder IDE, that allows you to develop applications quickly.
- Using C++Builder Component palette and Object Inspector, you can place VCL components on forms and specify their properties **without writing code.**
- Visual / Non Visual

Properties & Method

- **Properties**: apa yang “melekat” pada suatu komponen baik visual / non visual
  - Misal: name, caption, width, height
- **Method**: sering disebut Events
  - Merupakan kejadian-kejadian yang dilakukan / dikenakan pada suatu komponen baik visual / non visual
  - Misal: onClick, onDoubleClick, onMouseDown

Properties dan Events



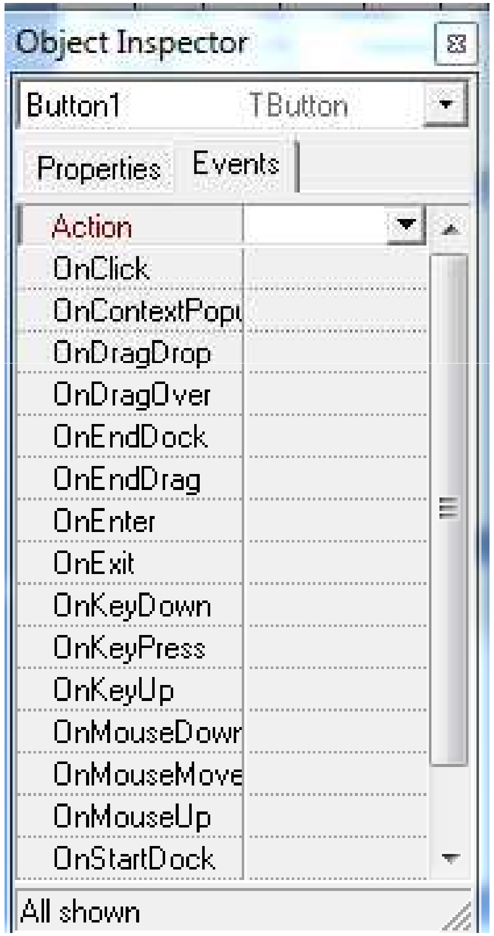
Object Inspector

Form1 TForm1

Properties | Events

Action	
ActiveControl	
Align	alNone
AlphaBlend	false
AlphaBlendVal	255
⊞ Anchors	[akLeft,akTop
AutoScroll	true
AutoSize	false
BiDiMode	bdLeftToRigh
⊞ BorderIcons	[biSystemMen
BorderStyle	bsSizeable
BorderWidth	0
Caption	Form1
ClientHeight	462
ClientWidth	854
Color	□ clBtnFace
⊞ Constraints	(TSizeConstra
Ctl3D	true

All shown



Object Inspector

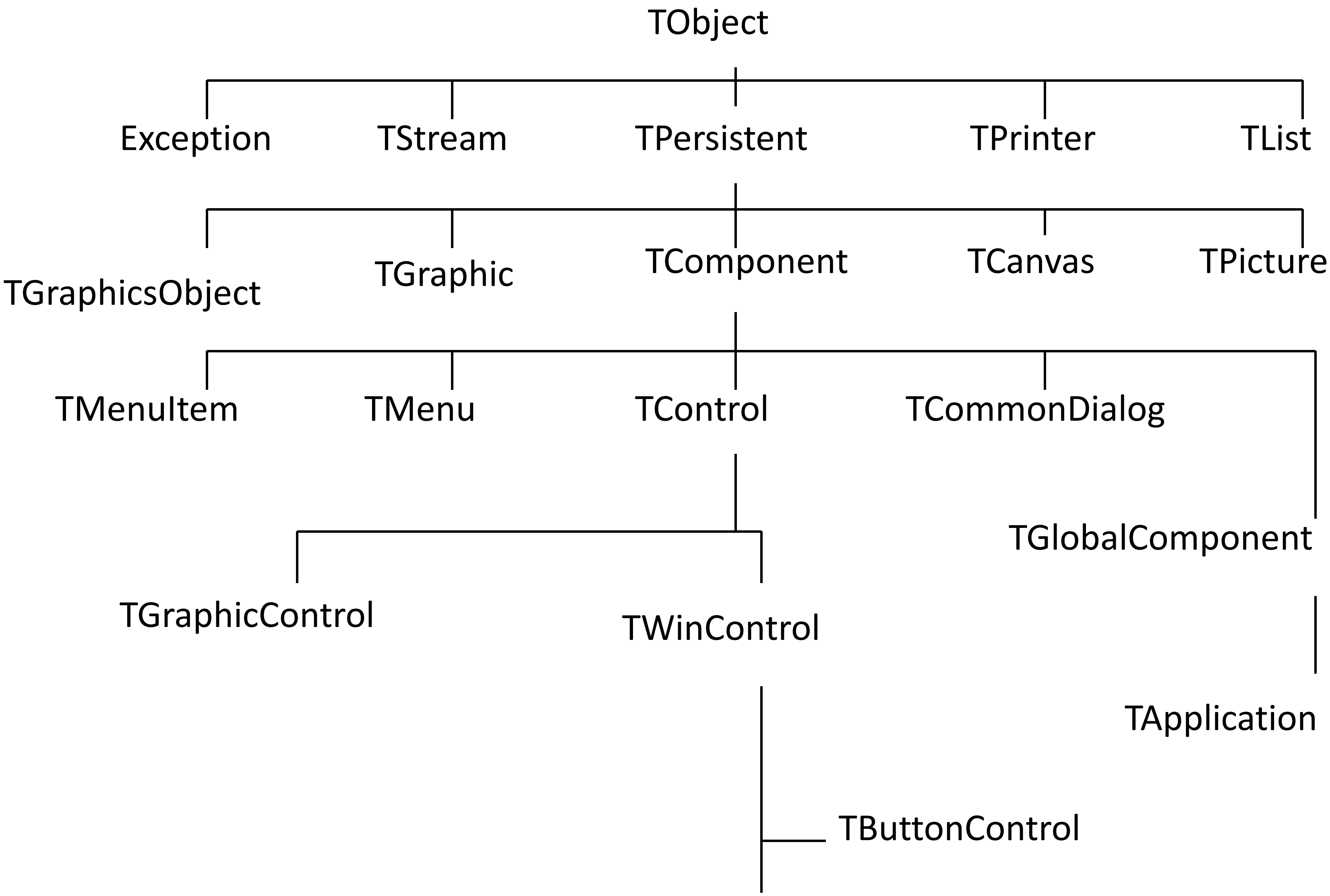
Button1 TButton

Properties | Events

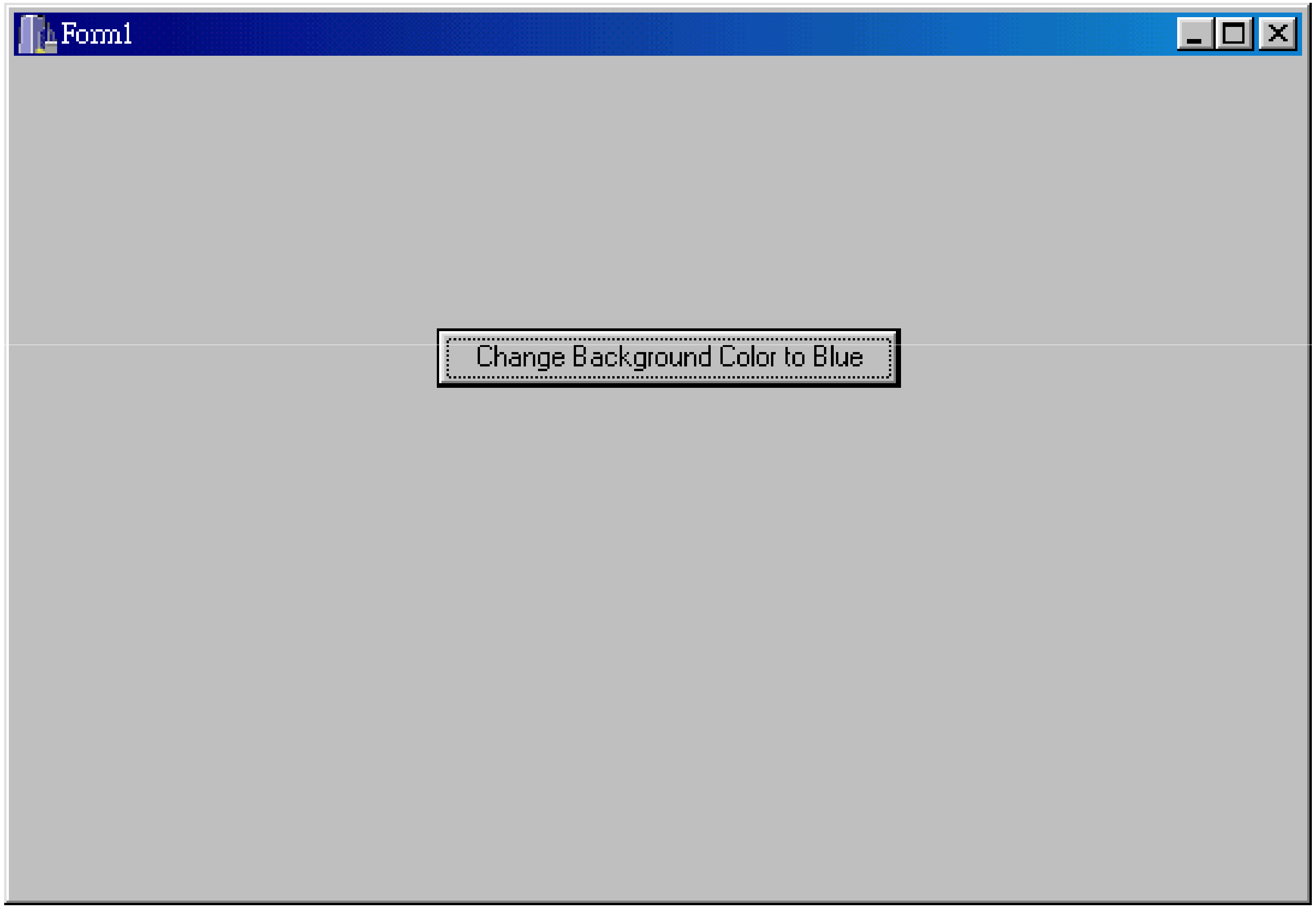
Action	
OnClick	
OnContextPopu	
OnDragDrop	
OnDragOver	
OnEndDock	
OnEndDrag	
OnEnter	
OnExit	
OnKeyDown	
OnKeyPress	
OnKeyUp	
OnMouseDown	
OnMouseMove	
OnMouseUp	
OnStartDock	

All shown

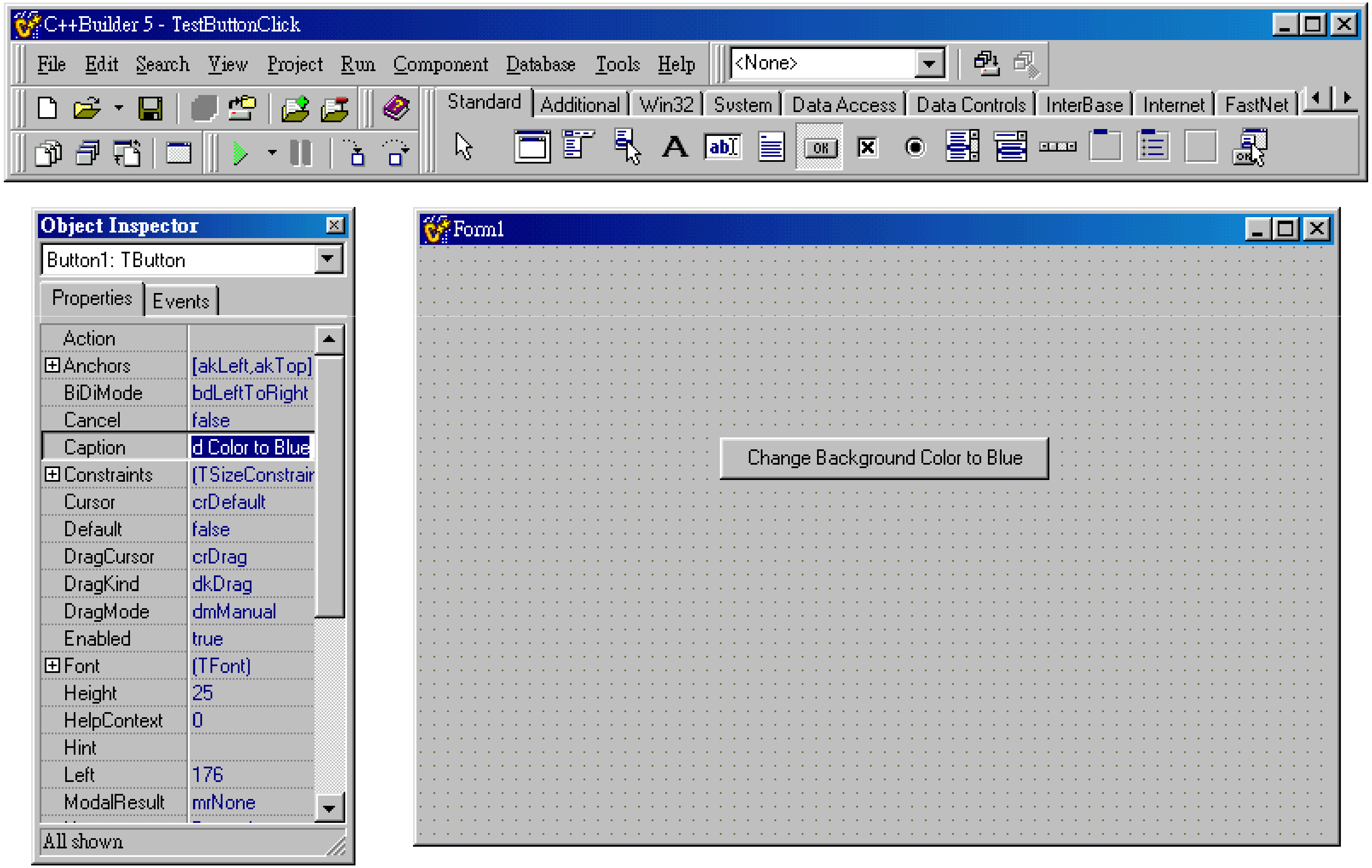
A Partial VCL Tree



A Simple Window Program – TestButtonClick



Inserting a Button – Property and Event


C++Builder 5 - TestButtonClick
File Edit Search View Project Run Component Database Tools Help
<None>
Standard Additional Win32 System Data Access Data Controls InterBase Internet FastNet
Object Inspector
Properties Events
All shown
Form1
Change Background Color to Blue

Button1: TButton

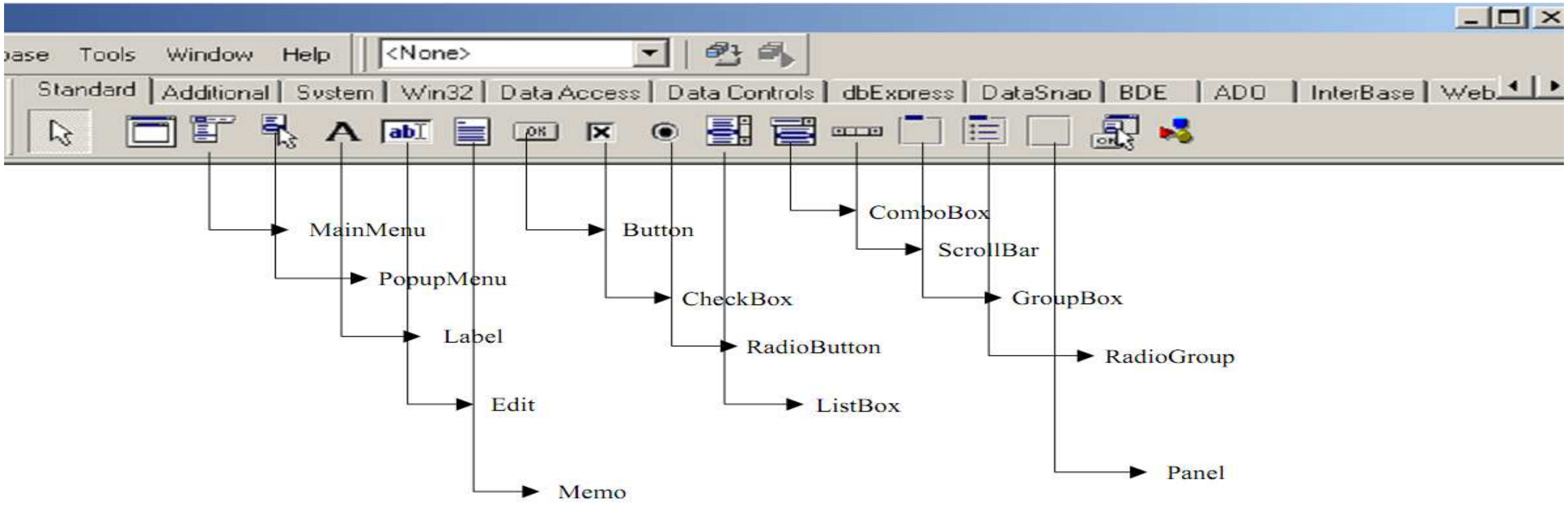
Action	
⊞Anchors	[akLeft,akTop]
BiDiMode	bdLeftToRight
Cancel	false
Caption	d Color to Blue
⊞Constraints	(TSizeConstrair
Cursor	crDefault
Default	false
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	true
⊞Font	(TFont)
Height	25
HelpContext	0
Hint	
Left	176
ModalResult	mrNone

Inserting a Button – Method (in Unit1.cpp)

```
//-----------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//---------------------------------------------
……
//----------------------------------------------
void __fastcall TForm1::ChColorButtonClick(TObject*
  Sender)
{
    //********************************
    Form1->Color = clBlue;
    //********************************
}
//-----------------------------------------------
```

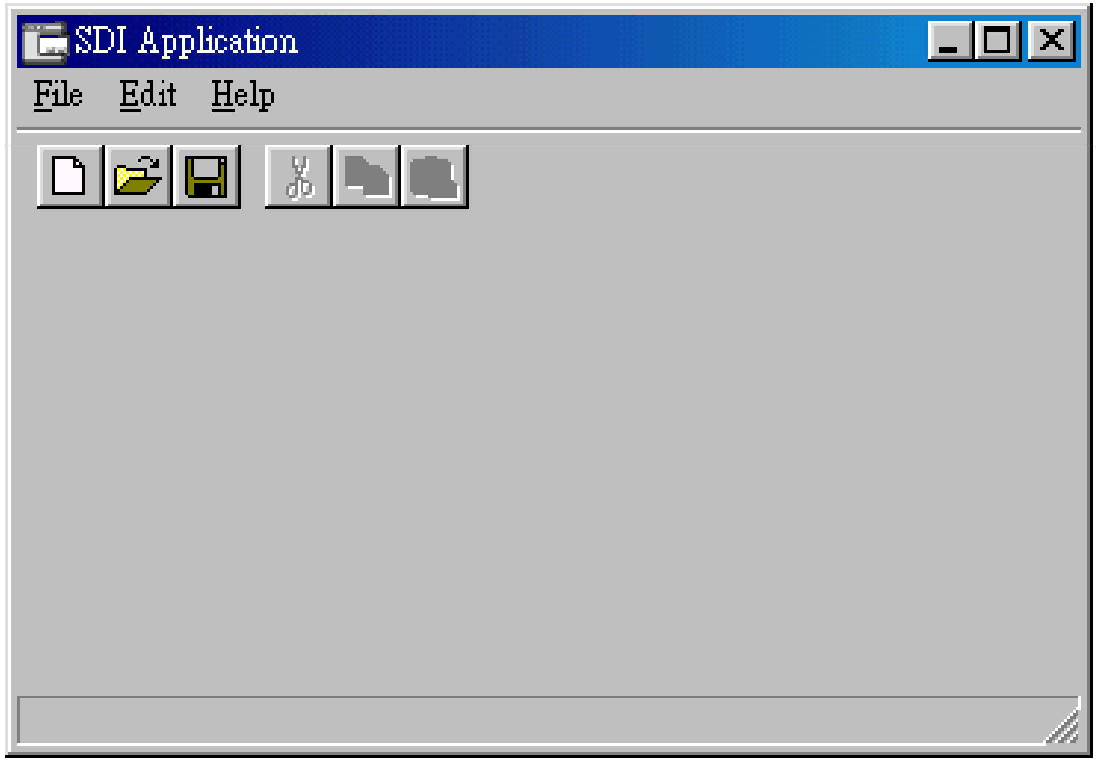
Common Controls

- List Box, Combo Box, Memo
- Radio Box, Check Box
- Panel, Group Box, Radio Group



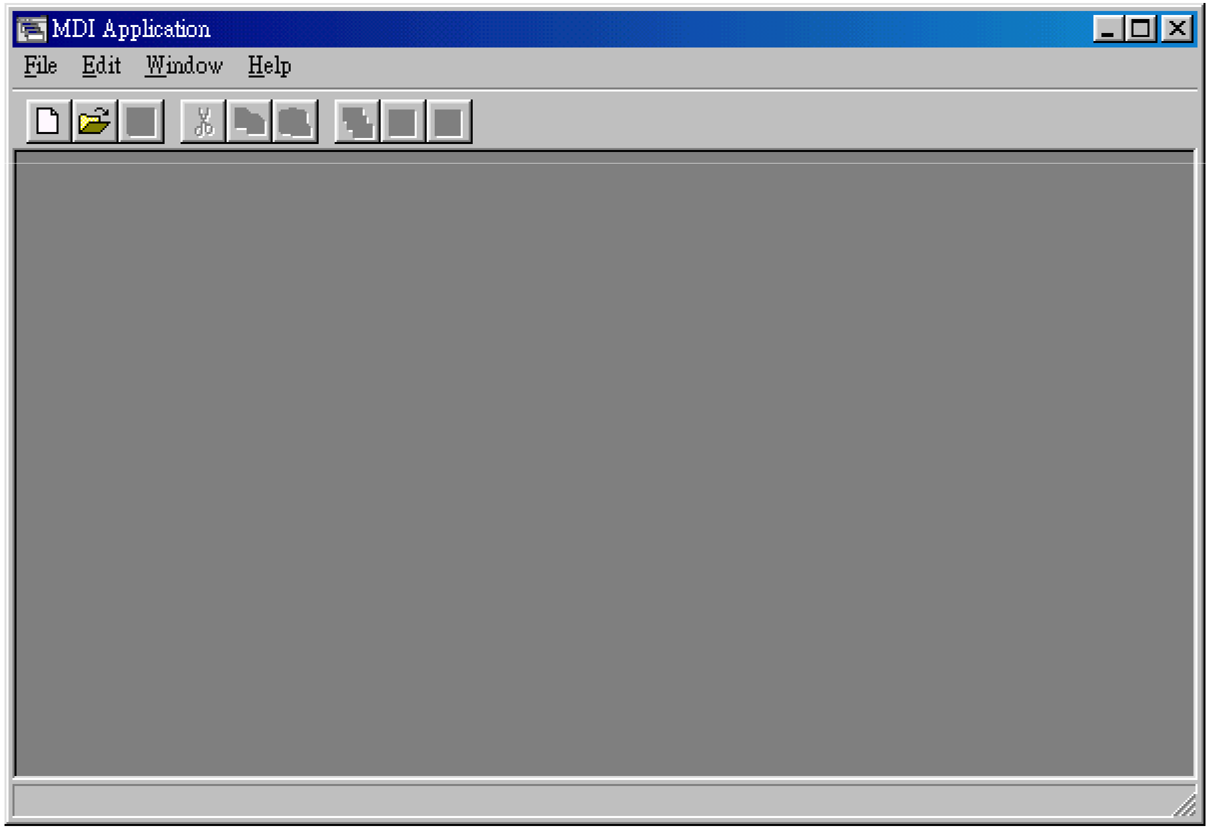
SDI Applications

- Create a file folder
- File | New... | Projects | SDI Applications



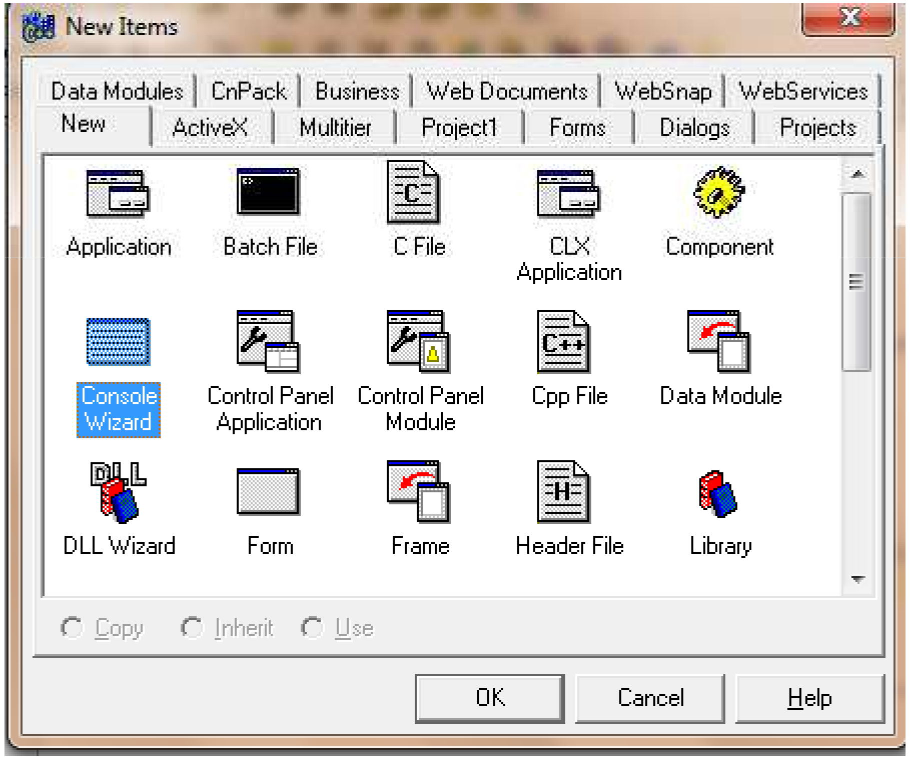
MDI Applications

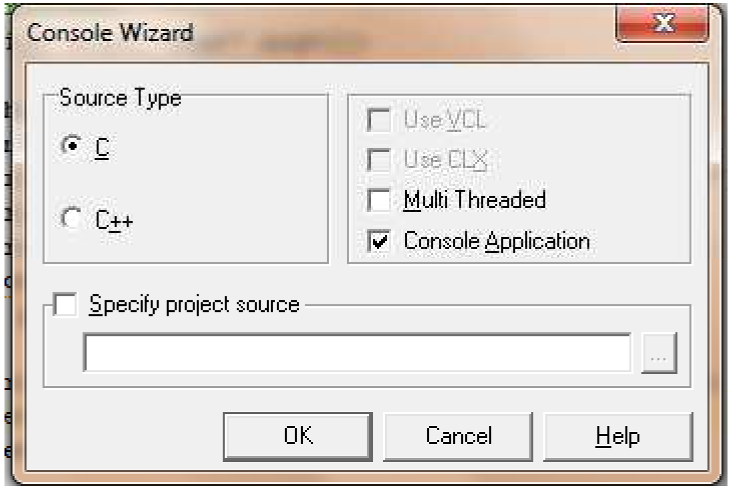
- Create a file folder
- File | New… | Projects | MDI Applications



Aplikasi Console di BCB

- File > New > Others





Source Type

● C

○ C++

□ Use VCL

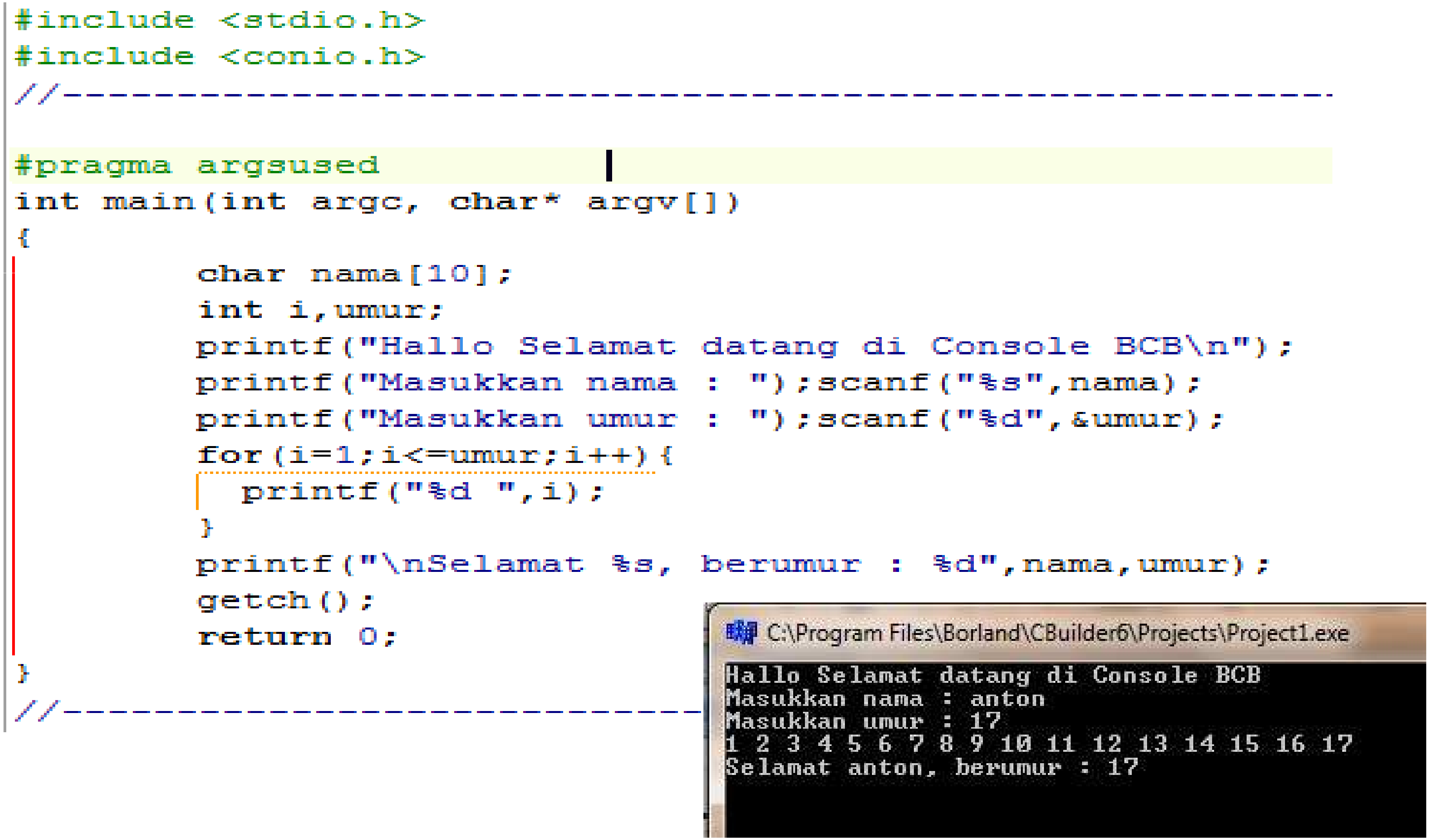
□ Use CLX

□ Multi Threaded

☑ Console Application

□ Specify project source

Tulis kode, RUN!



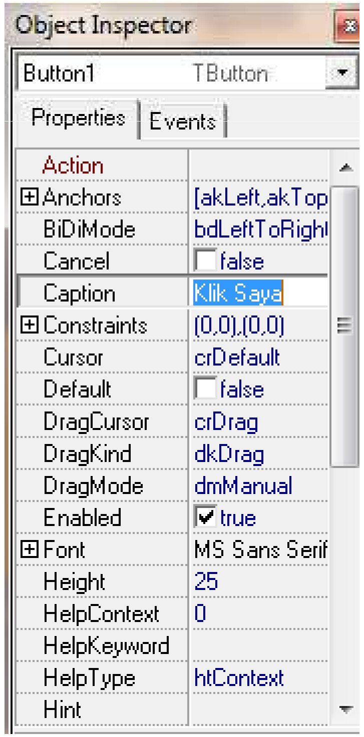
```
#include <stdio.h>
#include <conio.h>
//---------------------------------------------------------

#pragma argsused
int main(int argc, char* argv[])
{
        char nama[10];
        int i,umur;
        printf("Hallo Selamat datang di Console BCB\n");
        printf("Masukkan nama  :  ");scanf("%s",nama);
        printf("Masukkan umur  :  ");scanf("%d",&umur);
        for(i=1;i<=umur;i++){
          printf("%d ",i);
        }
        printf("\nSelamat %s, berumur : %d",nama,umur);
        getch();
        return 0;
}
//-----------------------------
```

C:\Program Files\Borland\CBuilder6\Projects\Project1.exe

```
Hallo Selamat datang di Console BCB
Masukkan nama : anton
Masukkan umur : 17
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Selamat anton, berumur : 17
```

Contoh Visual: Button

- Tombol yang dapat diklik
- Perhatikan properties dan Events pada Button yg terdapat pada Object Inspector

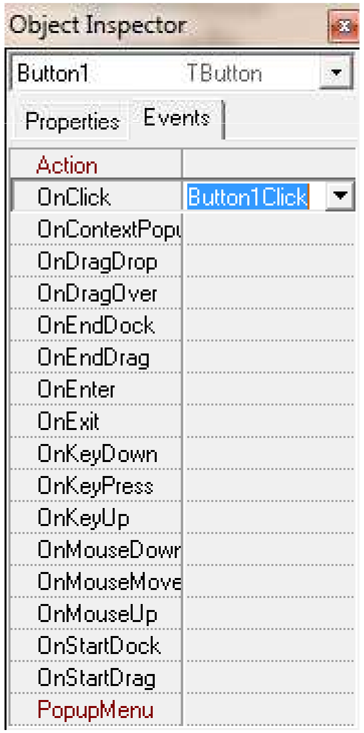


Object Inspector

Button1 TButton

Properties | Events

Action	
⊞Anchors	[akLeft,akTop
BiDiMode	bdLeftToRight
Cancel	□ false
Caption	Klik Saya
⊞Constraints	(0,0),(0,0)
Cursor	crDefault
Default	□ false
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	☑ true
⊞Font	MS Sans Serif
Height	25
HelpContext	0
HelpKeyword	
HelpType	htContext
Hint	



Object Inspector

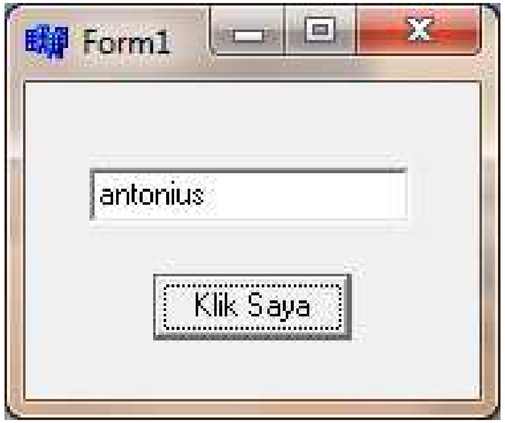
Button1 TButton

Properties | Events

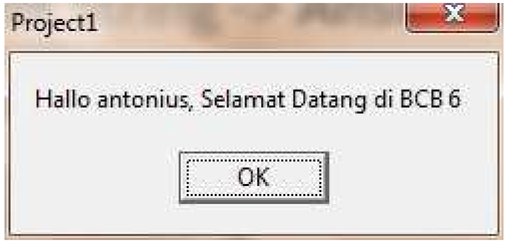
Action	
OnClick	Button1Click
OnContextPopu	
OnDragDrop	
OnDragOver	
OnEndDock	
OnEndDrag	
OnEnter	
OnExit	
OnKeyDown	
OnKeyPress	
OnKeyUp	
OnMouseDowr	
OnMouseMove	
OnMouseUp	
OnStartDock	
OnStartDrag	
PopupMenu	

Contoh Visual: EditBox

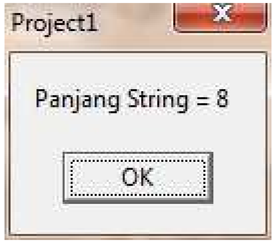
- Dapat menerima inputan oleh user dalam bentuk String -> **AnsiString**
- Contoh Aplikasi



antonius



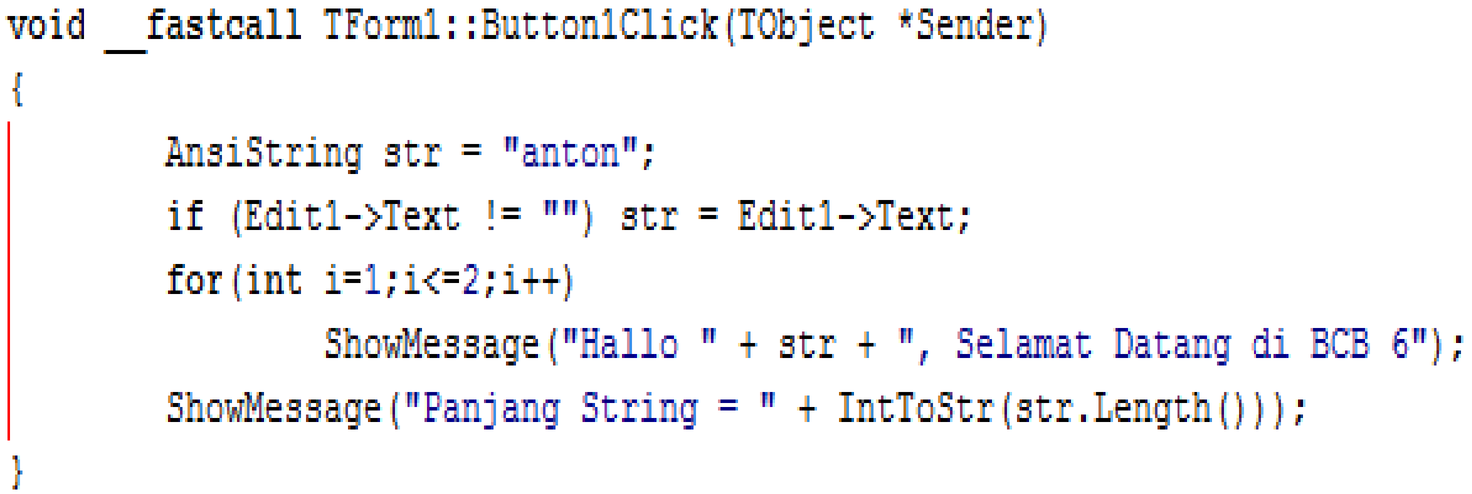
Hallo antonius, Selamat Datang di BCB 6



Panjang String = 8

Kode Program

- Double Click pada Button1, ketikkan program berikut:



```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
        AnsiString str = "anton";
        if (Edit1->Text != "") str = Edit1->Text;
        for(int i=1;i<=2;i++)
                ShowMessage("Hallo " + str + ", Selamat Datang di BCB 6");
        ShowMessage("Panjang String = " + IntToStr(str.Length()));
}
```

DEMO

- MultiForm Application
  - About Box
  - Menu Usage
- Cek username dan password

NEXT

- GUI Programming II