

BAB IX

GUI PROGRAMMING

9.1 Tujuan

1. Praktikan mampu mengimplementasikan materi – materi sebelumnya dalam bentuk GUI pada bahasa pemrograman C++, Python, Java, dan PHP.
2. Praktikan mampu memahami penggunaan GUI dari empat bahasa pemrograman yang berbeda (C++, Python, Java, dan PHP).
3. Praktikan mampu membuat beberapa fungsi pada GUI Programming dalam pemrograman C++, Python, Java, dan PHP.
4. Praktikan dapat membedakan GUI Programming dengan CLI Programming.

9.2 Materi

9.2.1. Pengertian GUI

GUI (*Graphical User Interface*) merupakan antarmuka grafis yang memfasilitasi interaksi antara pengguna (*user*) dengan program aplikasi.

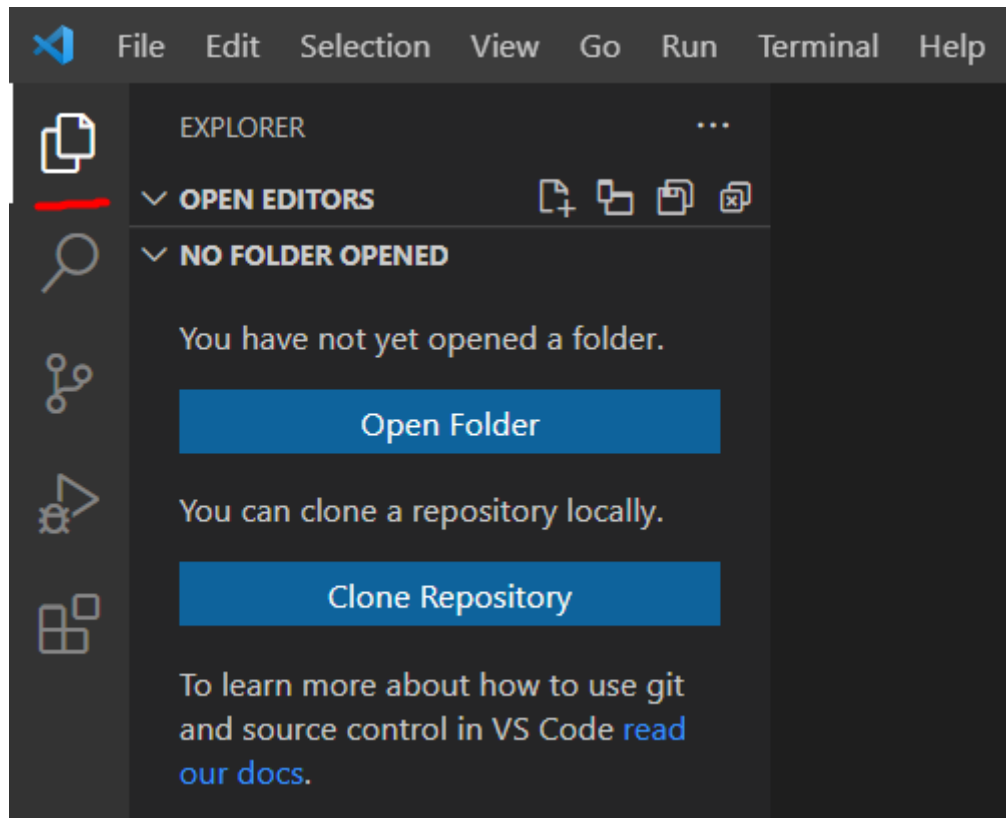
Berbeda dengan CLI (*Command Line Interface*) dimana dalam pemrogramannya menggunakan barisan perintah dan bukan berupa objek. Sedangkan GUI menggunakan objek sebagai antarmuka tampilan. Dengan GUI, *user* atau pengguna menjadi lebih mudah dalam menggunakan sebuah aplikasi.

9.2.2. GUI Programming pada Python

Pada bahasa pemrograman Python untuk membuat aplikasi berbasis GUI, kita menggunakan aplikasi Visual Studio Code yang sudah ter-*install* Python.

Untuk mencoba membuat aplikasi berbasis GUI pada bahasa pemrograman Python, ikuti langkah berikut ini:

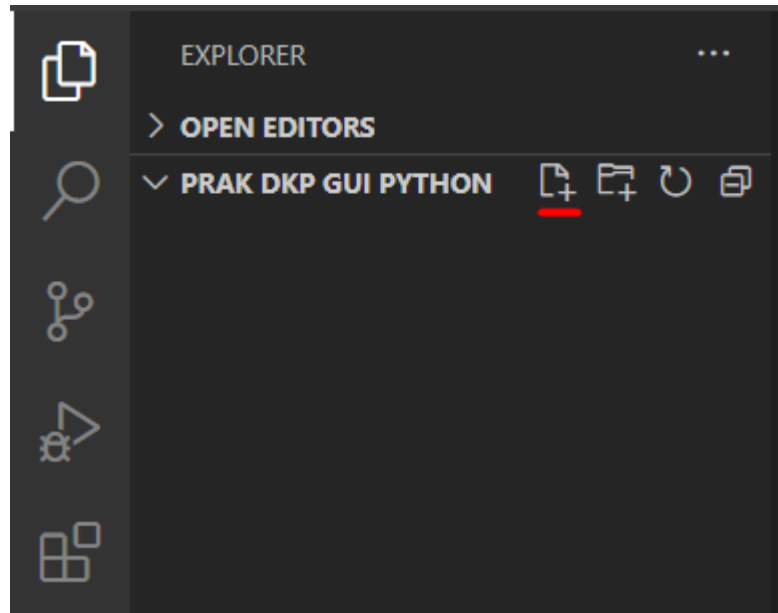
1. Buka aplikasi Visual Studio Code.
2. Pilih *icon Explorer*, seperti pada gambar di bawah. Lalu, klik open folder.



Gambar 9.1 *Icon Explorer*

3. Pilih folder yang ingin dijadikan tempat menyimpan file program untuk GUI Python.

4. Kemudian, pilih *icon new file* seperti pada gambar di bawah.



Gambar 9.2 *Icon New File*

5. Beri nama file “modul8dkp_kelxx.py”. (xx diisi dengan nomor kelompok praktikan).
6. **Paste full source code berikut.**

```
from ctypes import resize
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
from tkinter.messagebox import *

def on_click():
    print(askquestion("DKP", "Apakah anda senang mengikuti praktikum DKP?"))

window = Tk()
window.geometry("320x220")
window.title("GUI DKP")
window.resizable(width = 0, height = 0)

menu = Menu(window)
itemfile = Menu(menu, tearoff=0)
itemhelp = Menu(menu, tearoff=0)
itemfile.add_command(label='New')
itemfile.add_command(label='Save')
itemfile.add_command(label='Open')
itemhelp.add_command(label = 'About')
menu.add_cascade(label='File', menu=itemfile)
menu.add_cascade(label='Edit')
menu.add_cascade(label='View')
menu.add_cascade(label='Help', menu=itemhelp)
window.config(menu=menu)
```

```

buttonsubmit = Button(window,
                        text = "Submit",
                        command = on_click,
                        font = ("Times New Roman", 13),
                        fg = "black",
                        state = ACTIVE
                        ).place(x = 125, y = 170)

labelnama = Label(window,
                   text = "Name\t:",
                   font = ("times new roman", 10)).place(x=30,
y=20)
labeljk = Label(window,
                 text = "Gender\t:",
                 font = ("times new roman", 10)).place(x=30,
y=50)
labelhobby = Label(window,
                   text = "Hobby\t:",
                   font = ("times new roman",
10)).place(x=30, y=120)

strnama = StringVar()
entrynama = Entry(window,
                   textvariable = strnama,
                   font = ("times new roman",
10)).place(x=100, y=20)

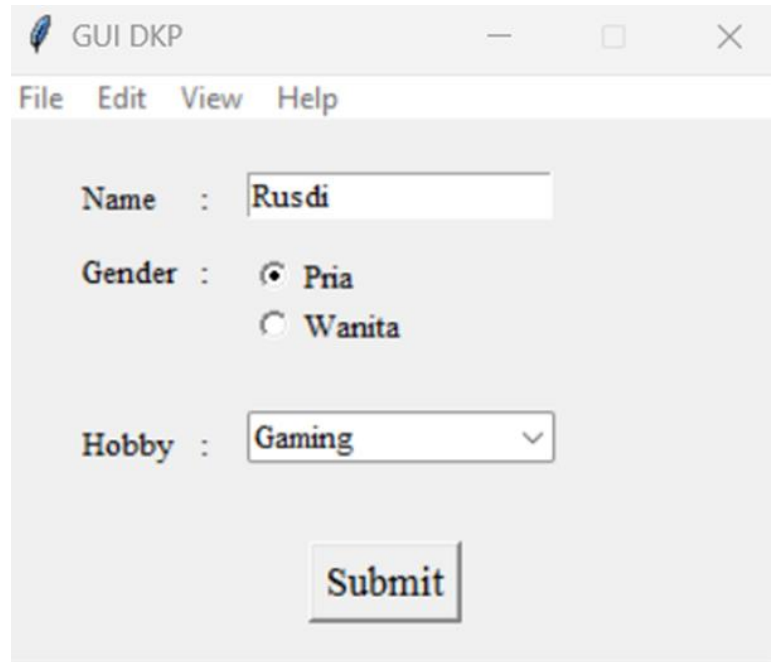
radio = IntVar()
R1 = Radiobutton(window,
                  text = "Pria",
                  font = ("times new roman", 10),
                  variable = radio,
                  value = 1).place(x = 100, y = 50)
R2 = Radiobutton(window,
                  text = "Wanita",
                  font = ("times new roman", 10),
                  variable = radio,
                  value = 2).place(x = 100, y = 70)

strhobby = StringVar(value='Pilih di sini')
combobox1 = ttk.Combobox(window,
                           width = 17,
                           font = ("times new roman", 10),
                           textvariable = strhobby,
                           state="readonly")
combobox1.place(x=100, y=117)
combobox1['values'] = ('Gaming', 'Music', 'Movie')

window.mainloop()

```

7. Jalankan aplikasi dan lihat hasilnya.

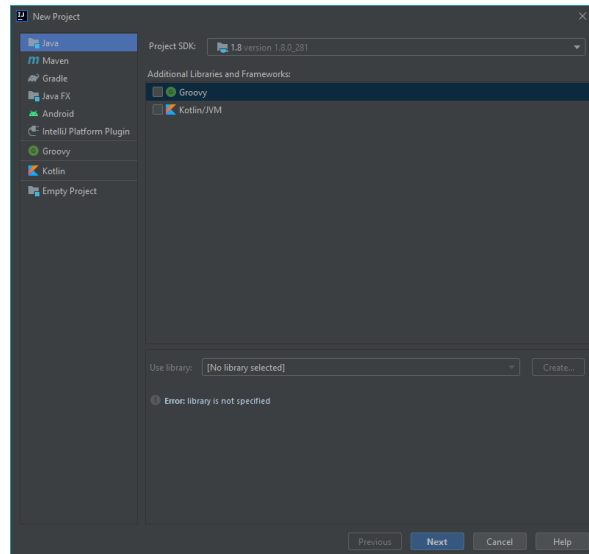


Gambar 9.3 Output

9.2.3. GUI Programming pada Java

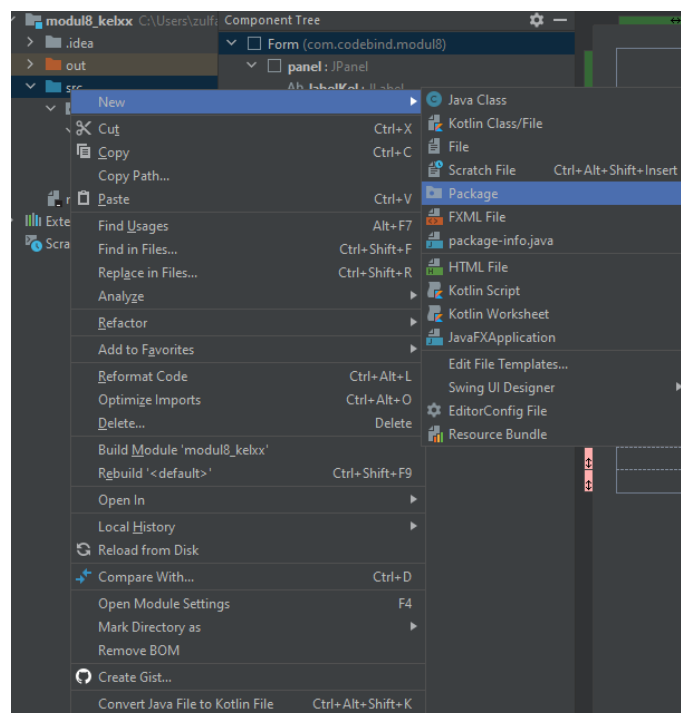
Pada bahasa pemrograman java untuk membuat aplikasi berbasis GUI digunakanlah Java Swing atau Java AWT. Dalam praktikum kali ini kita menggunakan IntelliJ sebagai IDE dari bahasa pemrograman java. IntelliJ, terdapat beberapa komponen GUI, diantaranya: JFrame, JLabel, JTextField, JTextArea, JComboBox, JButton, dan masih banyak lagi. Dalam praktikum kali ini kita akan menggunakan beberapa komponen GUI pada IntelliJ untuk membentuk suatu program sederhana berbasis GUI. Program yang akan kita buat merupakan program dengan input berupa *text*. Berikut merupakan langkah–langkah nya:

1. Buka aplikasi IntelliJ
2. Pilih *File* □ *New Project*
3. Pada langkah pertama akan muncul pilihan, pilih Java lalu pilih SDK yang digunakan, lalu klik *next*



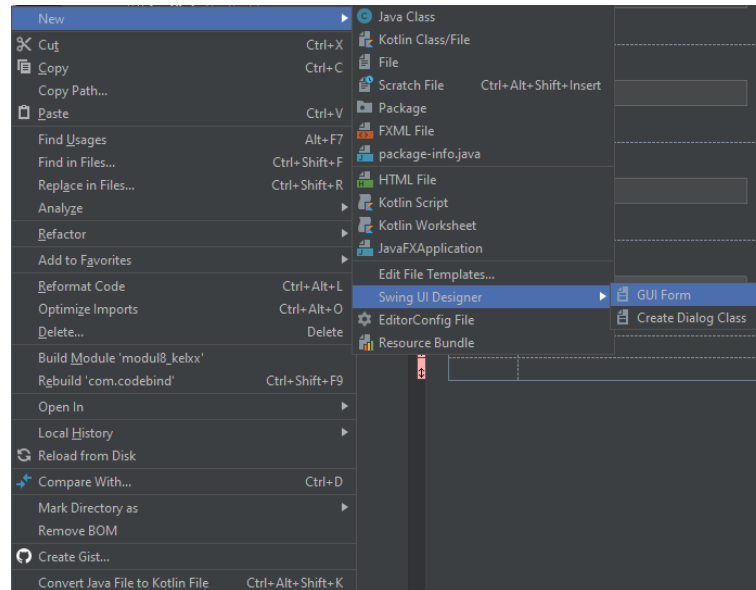
Gambar 9.2 New project

4. Selanjutnya, tidak usah memakai *template*, beri nama *project* dengan nama modul8dkp_kelxx (xx diisi dengan nomor kelompok) setelah selesai pilih *finish*.
5. Setelah itu klik kanan pada folder *src* modul8dkp_kelxx pilih *New* kemudian *Package* dan isikan *package* dengan nama *com.company*



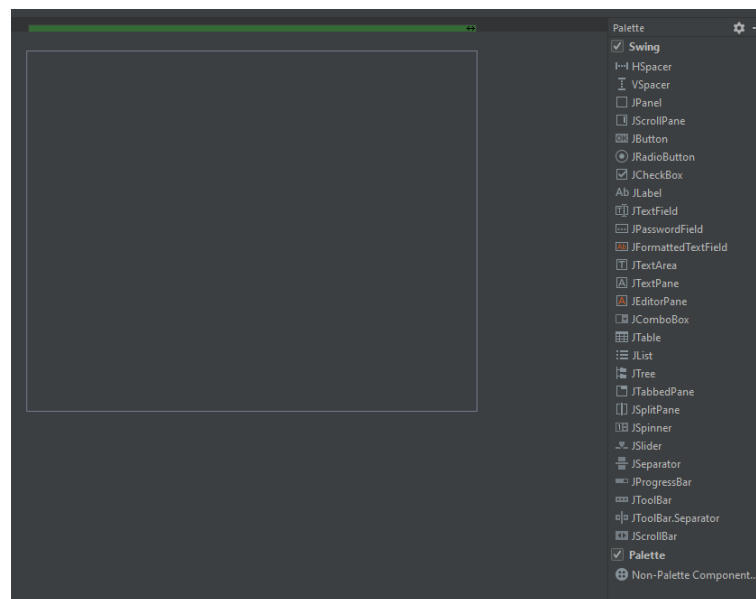
Gambar 9.5 Memilih Package

6. Kemudian dalam *package* tadi, klik kanan pilih *new* lalu *Swing UI Designer* kemudian pilih *GUI Form*. Beri nama dengan “modul8” setelah selesai pilih OK.



Gambar 9.6 Memilih GUI Form

7. Kemudian akan muncul tampilan seperti gambar berikut



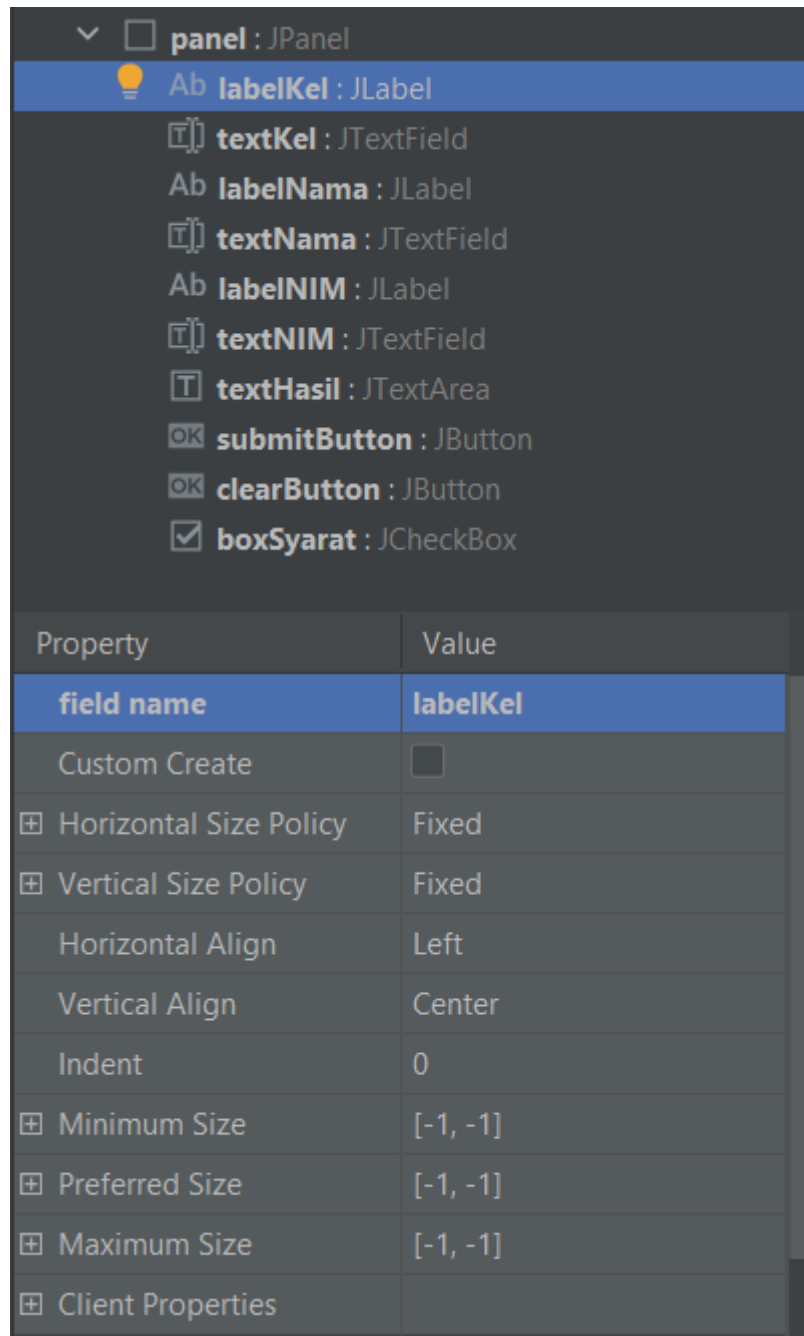
Gambar 9.7 Tampilan GUI IntelliJ

8. Langkah selanjutnya masukan komponen label, *text field*, *button* dan *text area*.
9. Desain komponen tersebut seperti gambar berikut

The image shows a Java Swing window titled "modul8.form". Inside the window, there is a form with a dark gray background. The form is organized into a grid-like structure. On the left side, there are three labels: "Kelompok", "Nama", and "NIM", each followed by a text input field. Below these fields is a checkbox labeled "Syarat dan ketentuan". At the bottom of the form, there are two buttons: "Submit" and "Clear". On the right side of the form, there is a large text area labeled "textHasil:JTextArea". The form has a light gray border and a dark gray background. There are also some green and blue handles for resizing the window and its components.

Gambar 9.8 Tampilan modul8.form

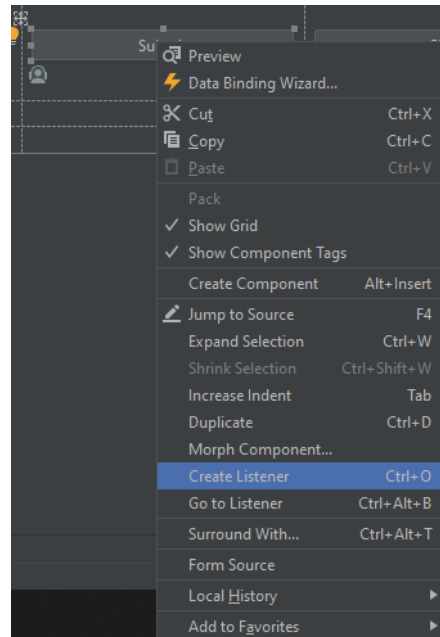
10. Kemudian beri nama variabel pada panel, label, *text field*, dan *text area* pada desain, dengan cara isikan pada *field name* nama variabelnya. Ganti nama variabelnya sesuai pada gambar ini.



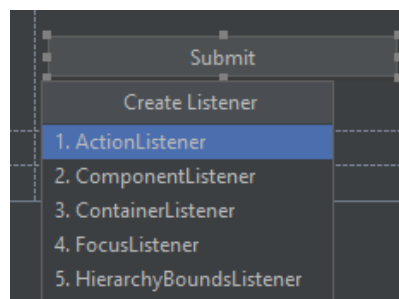
Gambar 9.9 Nama variabel

11. Setelah itu kita akan memasukkan *source code* untuk menjalankan fungsi dari aplikasi sederhana ini.
12. Pada langkah ini kita akan memanfaatkan fungsi *listener* untuk mengaktifkan fungsi yang kita inginkan, klik kanan pada *button Submit* □ □

Create listener □ pilih “*ActionListener*”



Gambar 9.3 Memilih *Create Listener*



Gambar 9.4 Memilih *ActionListener*

13. Tuliskan *source code* berikut pada *action listener* submit button tersebut.

```
textHasil.setText("Halo, saya dari kelompok " +
    textKel.getText() + " bernama " +
    textNama.getText() + " dengan NIM " +
    textNIM.getText() + "\nBelajar GUI
    bersama kak dan kak George itu menyenangkan bukan? bukan");
```

```
31 public modul8() {
32     submitButton.addActionListener(new ActionListener() {
33         @Override
34         public void actionPerformed(ActionEvent e) {
35             textHasil.setText("Halo, saya dari kelompok " +
36                 textKel.getText() + " bernama " +
37                 textNama.getText() + " dengan NIM " +
38                 textNIM.getText() + "\nBelajar GUI bersama kak Tito dan kak George itu menyenangkan bukan? bukan");
```

Gambar 9.5 *submit button code*

14. Lakukan hal yang sama pada *clear button* dan masukan *source code* seperti dibawah.

```
textKel.setText("");  
textNama.setText("");  
textNIM.setText("");  
textHasil.setText("");
```

```
clearButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        textKel.setText("");  
        textNama.setText("");  
        textNIM.setText("");  
        textHasil.setText("");  
    }  
});
```

Gambar 9.6 *source code clear button*

15. Jangan lupa tambahkan *main method* dalam *class modul8* agar GUI-nya bisa di-run dengan cara menuliskan *source code* berikut di dalam *class*.

```
public static void main(String[] args) {  
    JFrame frame = new JFrame("DKP GUI");  
    frame.setContentPane(new modul8().panel);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.pack();  
    frame.setVisible(true);  
    frame.setSize(600, 600);  
    frame.setResizable(true);  
}
```

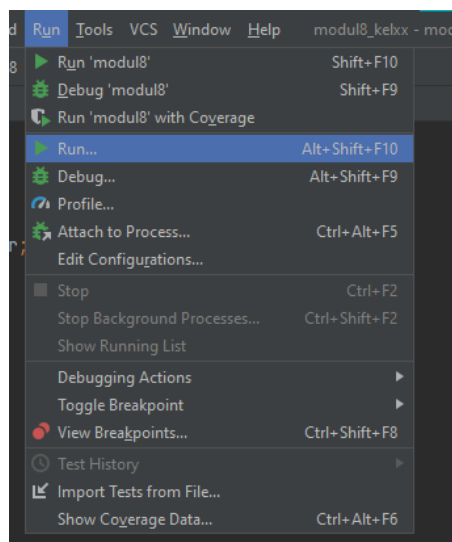
```

public static void main(String[] args) {
    JFrame frame = new JFrame( title: "DKP GUI");
    frame.setContentPane(new modul8().panel);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
    frame.setSize( width: 600, height: 600);
    frame.setResizable(true);
}

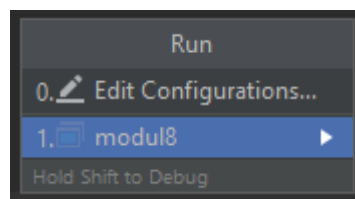
```

Gambar 9.7 Main method

16. Jalankan *project* tersebut dengan menekan “Alt + Shift + F10” atau dengan klik *run* pada *toolbar*. Kemudian pilih “modul8”.

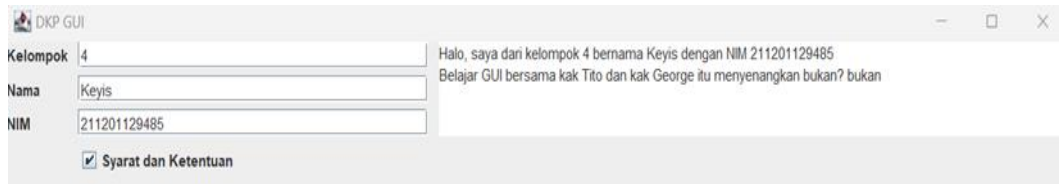


Gambar 9.8 Memilih *run*



Gambar 9.9 Memilih modul8

17. Berikut merupakan hasil *running project* (sebelum dan setelah menekan tombol Submit) dan liat hasilnya!



Gambar 9.10 Hasil program (1)

18. Full source code:

```
package com.company;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class modul8 extends Container {
    private JTextField textKel;
    private JTextField textNama;
    private JTextField textNIM;
    private JTextArea textHasil;
    private JButton submitButton;
    private JButton clearButton;
    private JCheckBox boxSyarat;
    private JLabel labelKel;
    private JLabel labelNama;
    private JLabel labelNIM;
    private JPanel panel;

    public static void main(String[] args) {
        JFrame frame = new JFrame("DKP GUI");
        frame.setContentPane(new modul8().panel);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
        frame.setSize(600, 600);
        frame.setResizable(true);
    }

    public modul8() {
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textHasil.setText("Halo, saya dari kelompok "
+
                textKel.getText() + " bernama " +
                textNama.getText() + " dengan NIM " +
                textNIM.getText() + "\nngerjain Modul
8 menyenangkan bukan? bukan");
            }
        });

        clearButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textKel.setText("");
            }
        });
    }
}
```

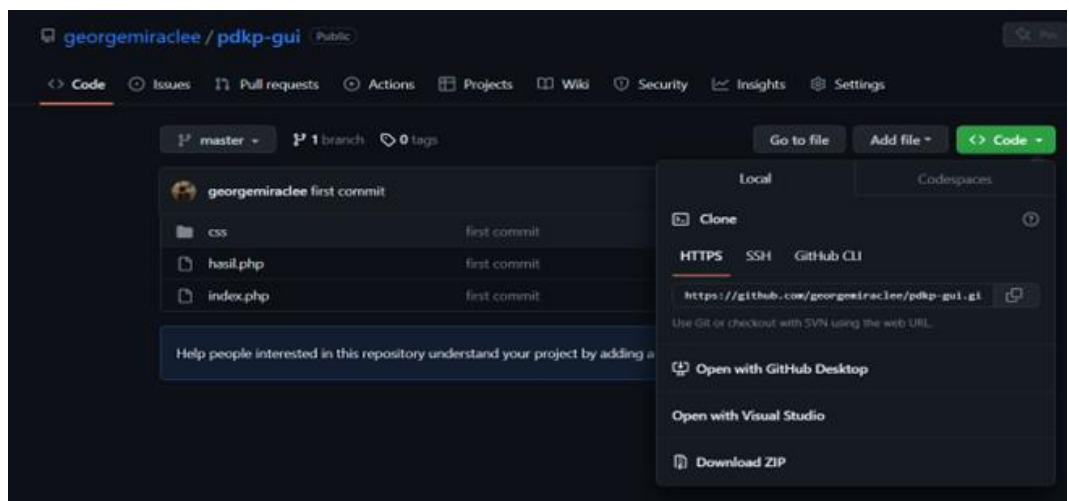
```
        textNama.setText("");  
        textNIM.setText("");  
        textHasil.setText("");  
    }  
});  
}  
}
```

9.2.4. GUI Programming pada PHP

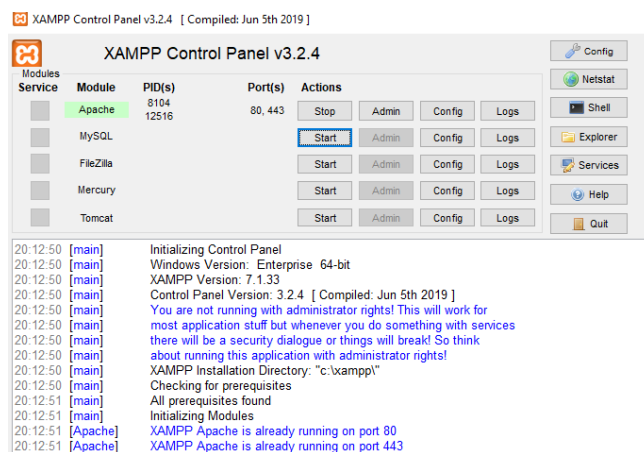
Pada bahasa pemrograman PHP untuk membuat aplikasi berbasis GUI, kita bisa menggunakan aplikasi Visual Studio Code. Visual Studio Code adalah sebuah aplikasi penyunting teks dan penyunting *source code* yang berjalan di sistem operasi Windows. Visual Studio Code dapat menampilkan dan menyunting teks dan *source code* berbagai bahasa pemrograman.

Untuk mencoba membuat aplikasi berbasis GUI pada bahasa pemrograman PHP, ikuti langkah berikut ini:

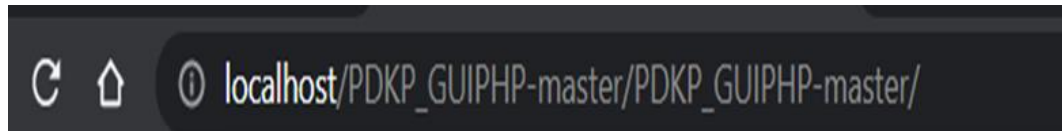
1. **Buka link github berikut:** [PHP Full Program](#)
2. Clone program yang ada pada github tersebut.
3. Letakan folder program di folder `xampp/htdocs/`



4. Buka program menggunakan Visual Studio Code.
5. Buka aplikasi XAMPP. Jalankan modul Apache dengan menekan tombol *Start*.



6. Jalankan program PHP menggunakan *browser*. Akses



7. Selamat mencoba!

Praktikum DKP: GUI PHP

Tambah Data

Nama Anda

NIM

Jenis Kelamin

Pilih Jenis kelamin

Peminatan

☐ Software

☐ Embedded System

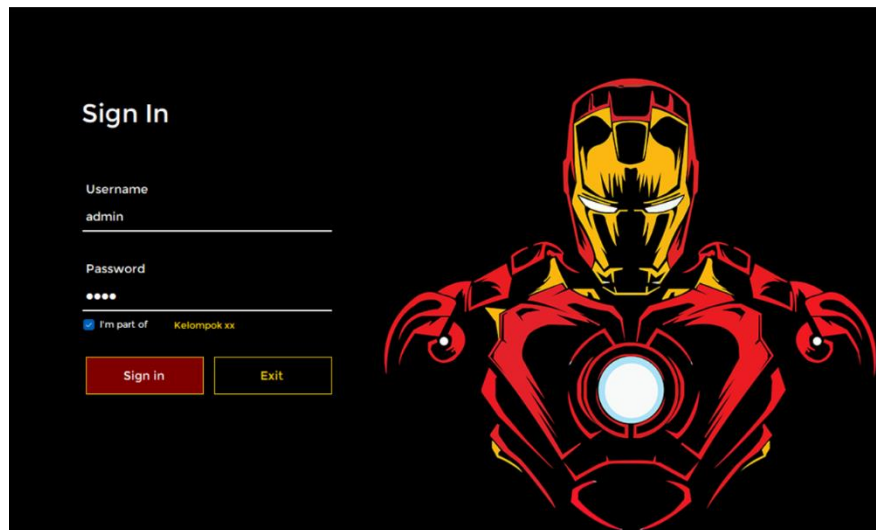
☐ Network

☐ Multimedia

Submit

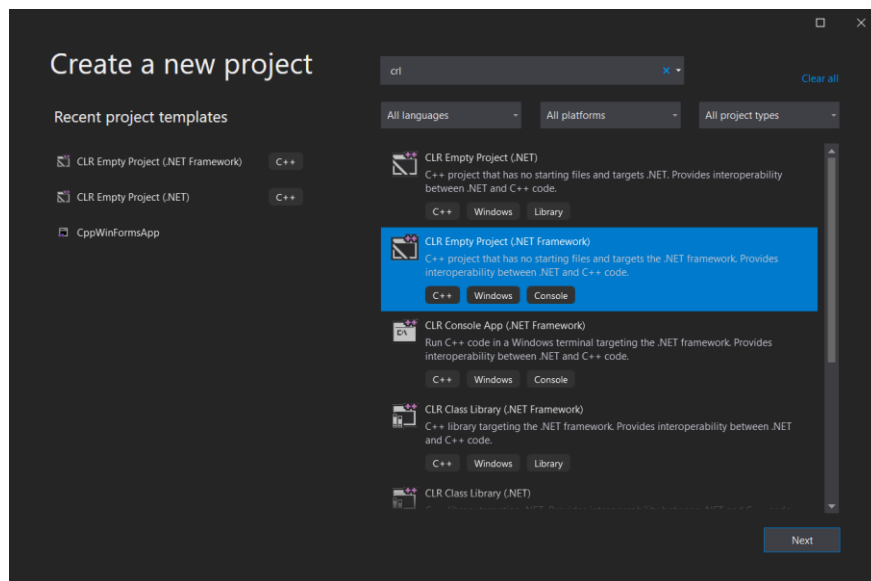
9.2.5. GUI Programming pada C++

Pada bahasa pemrograman C++ untuk membuat aplikasi berbasis GUI, kita akan menggunakan aplikasi Visual Studio. Berbeda dengan Visual Studio Code, Visual Studio termasuk ke dalam Integrated Development Environment (IDE) Tools dimana kita dapat melakukan debugging, refactoring, compiling, pembuatan packages dan lain-lain. Sedangkan VSCode termasuk kedalam Text Editor atau dalam artian, VSCode digunakan hanya untuk menulis code saja, tidak termasuk dengan Debugging, Compiling, dll.

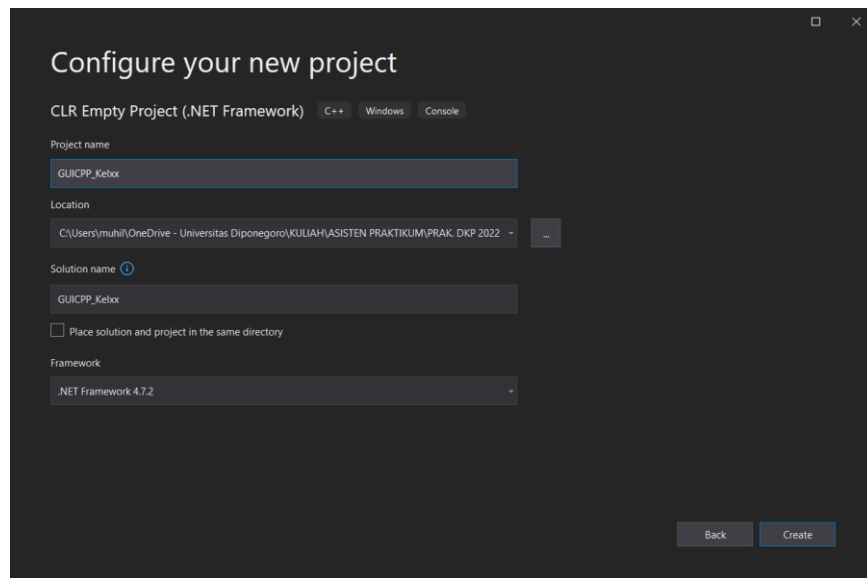


Pada percobaan ini kita akan membuat sebuah program Sign in seperti di atas. Silahkan ikuti langkah berikut ini untuk mencoba membuat program C++ GUI tersebut.

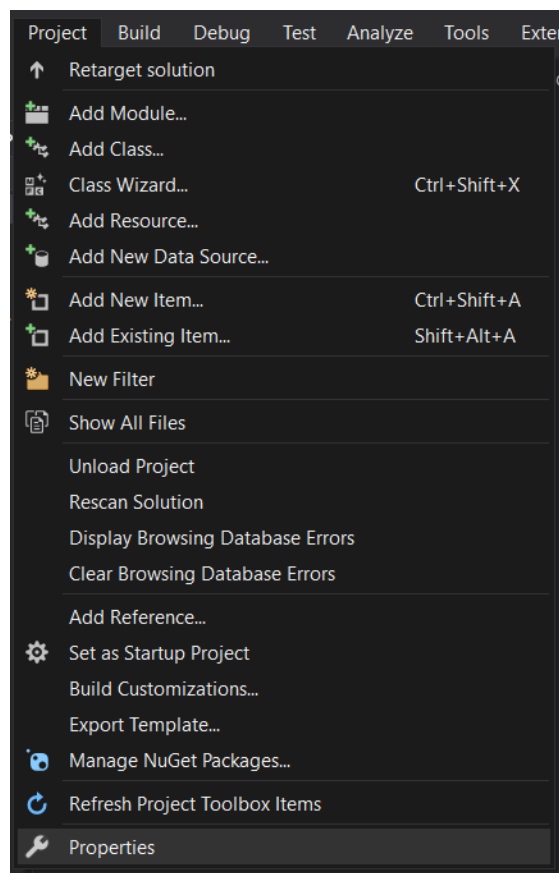
1. Buka aplikasi Visual Studio lalu pilih *Create a new project*.



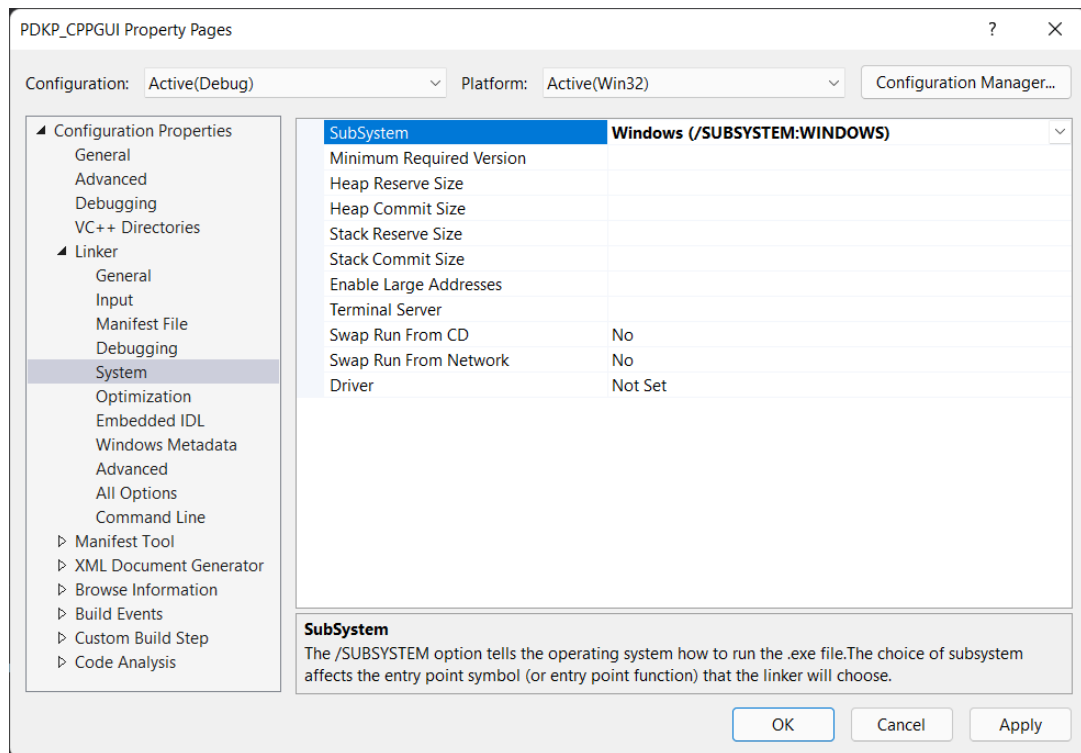
2. Ketikan “clr” pada kolom pencarian, pilih CLR Empty Project (.NET Framework), lalu tekan tombol Next seperti gambar di atas.



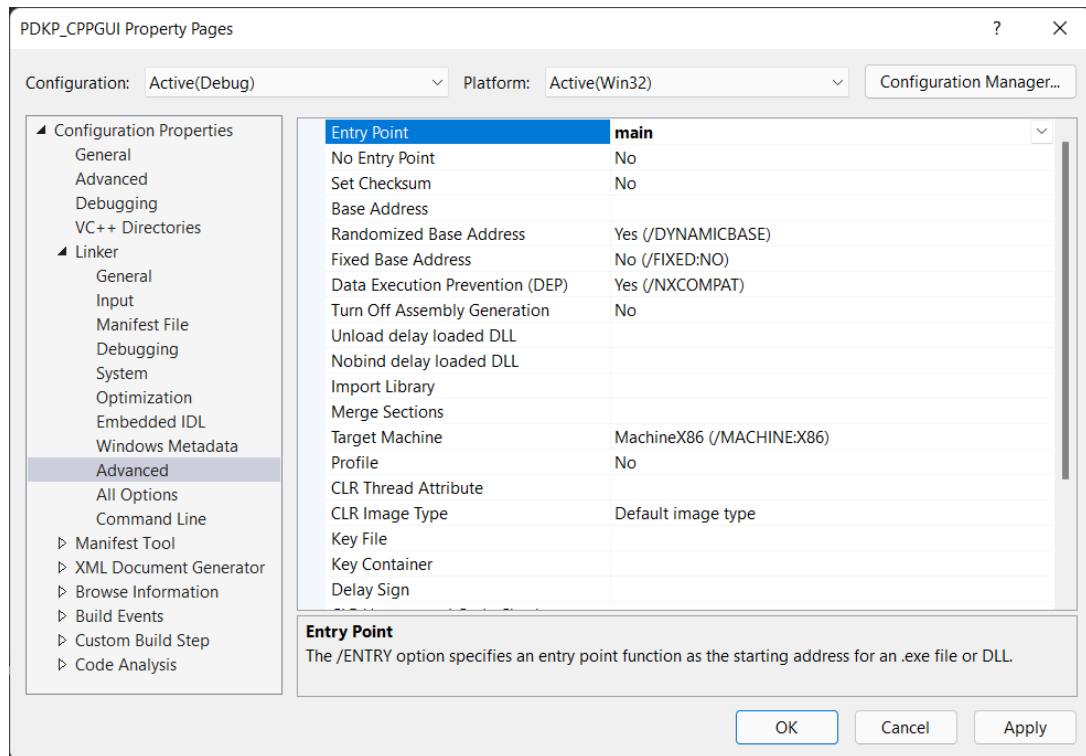
3. Masukkan nama *project* seperti gambar di atas dengan xx adalah nomor kelompok kalian. Tentukan lokasi penyimpanan proyek lalu tekan next.



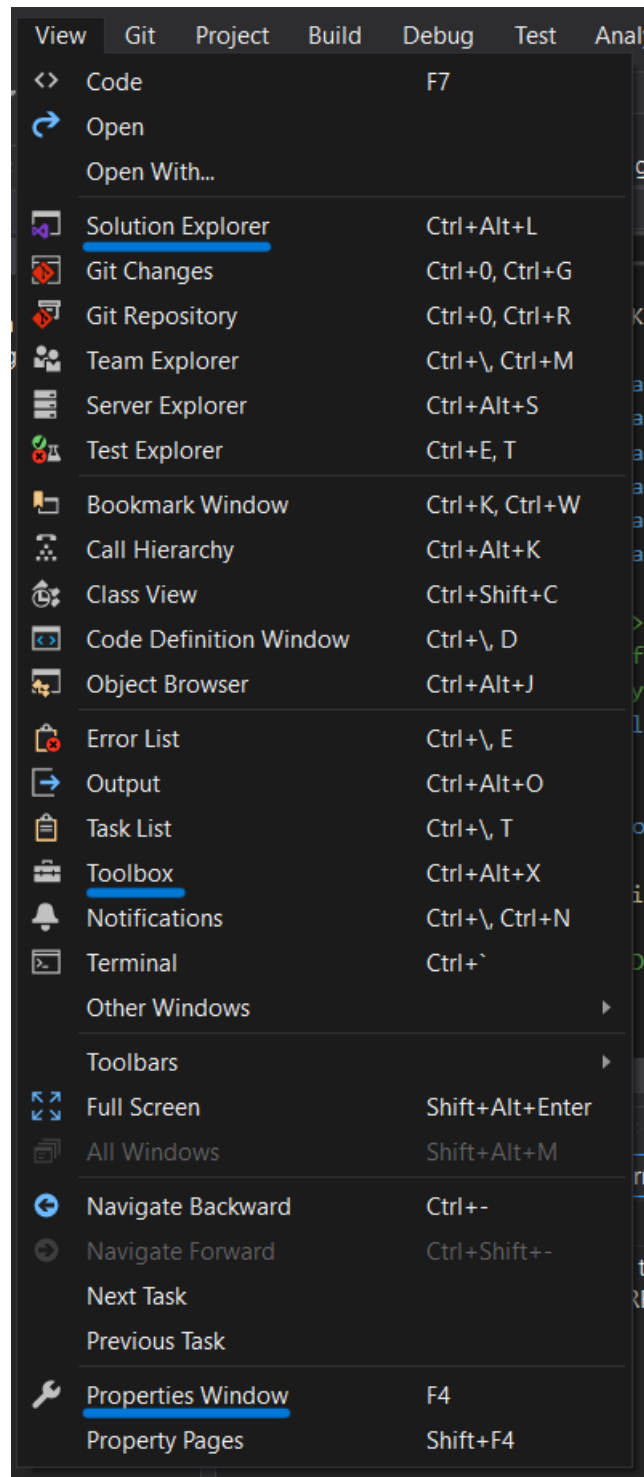
4. Setelah *project* terbuat langsung arahkan *cursor* ke menu **Project** dan pilih *option* Properties seperti gambar di atas.



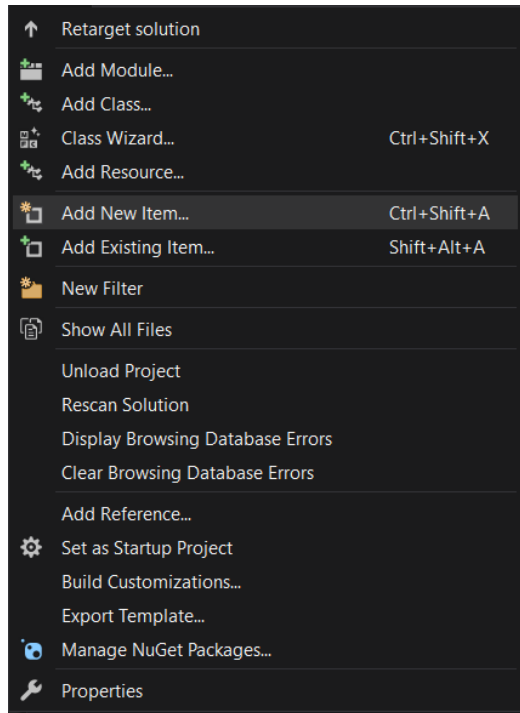
5. Pada halaman Properties pilih menu System lalu pada kolom SubSystem pilih **Windows (/SUBSYSTEM:WINDOWS)** seperti di atas. Lalu pilih menu Advanced dan ketikkan **“main”** di kolom Entry Point seperti gambar di bawah.



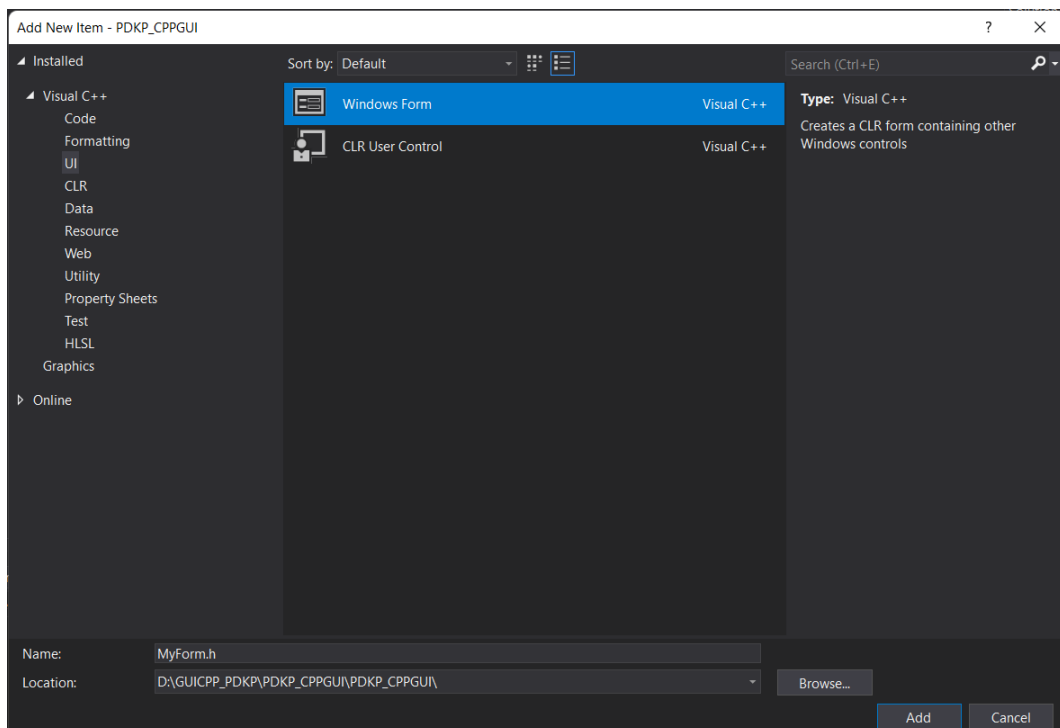
6. Setelah itu, klik *Apply* lalu *Ok*.
7. Sebelum mulai membuat program, kita perlu menampilkan beberapa *tools* yang akan memudahkan kita nanti dengan cara pilih menu **View** lalu pilih “Solution Explorer”, “Toolbox”, dan “Properties Windows”.

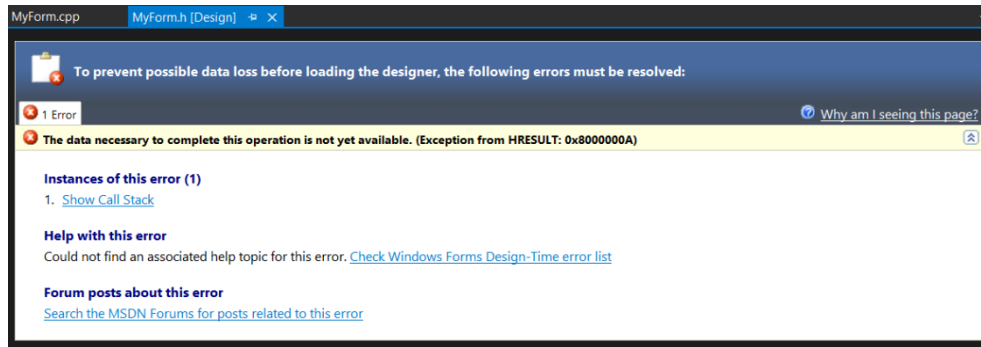


8. Buat Form baru dengan menekan “Add New Item...” di menu **Project** seperti gambar di bawah.



9. Pilih *option* “UI” lalu pilih “Windows Form” lalu tekan tombol “Add” seperti gambar di bawah.





10. Setelah *item* terbuat, jika kalian mengalami error seperti gambar di atas, biarkan saja dulu. Buka halaman kode MyForm.cpp dan masukan kode seperti gambar di bawah ini.

```
1  #include "MyForm.h"
2
3  using namespace System;
4  using namespace System::Windows::Forms;
5  [STAThread]
6  void main(array<String^>^ args) {
7      Application::EnableVisualStyles();
8      Application::SetCompatibleTextRenderingDefault(false);
9      GUICPPKelxx::MyForm form; //xx diubah ke nama kelompok kalian. Sesuaikan dengan nama project
10     Application::Run(% form);
11 }
```

```
#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThread]

void main(array<String^>^ args) {

    Application::EnableVisualStyles();

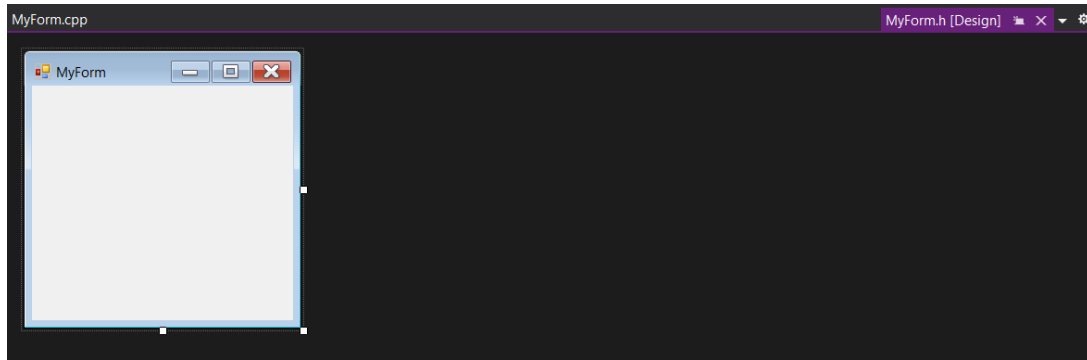
    Application::SetCompatibleTextRenderingDefault(false);

    GUICPPKelxx::MyForm form;

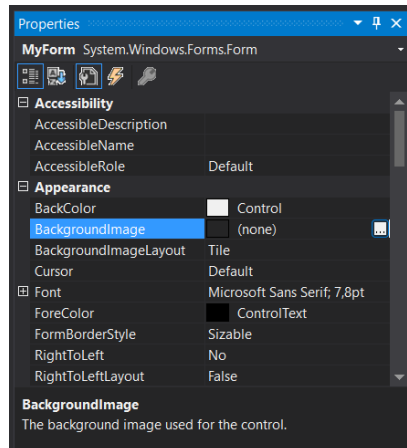
    Application::Run(% form);

}
```

11. Simpan project kalian lalu tutup aplikasi Visual Studio. Setelah itu, buka Visual Studio lagi dan buka project kalian tadi. Seharusnya error di atas sudah tidak ada lagi dan kalian akan melihat form seperti gambar di bawah.

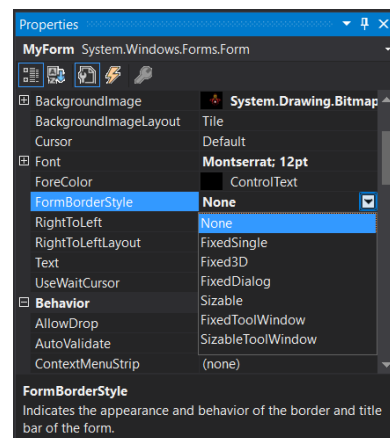
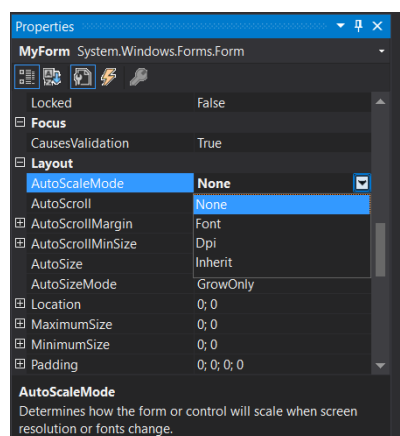


12. Silahkan sesuaikan ukuran form sesuai dengan keinginan kalian.

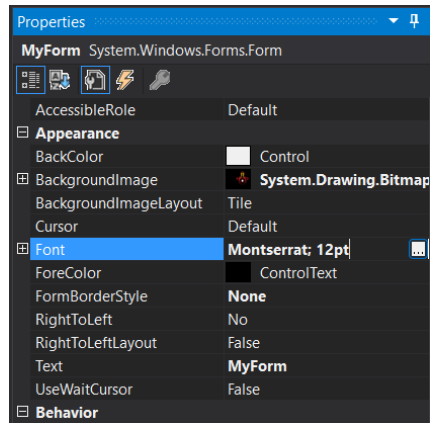


13. Tambahkan *background* dengan klik titik tiga di kolom BackgroundImage pada Properties seperti gambar di atas. Jika kalian ingin menggunakan *background* yang sama dengan contoh, kalian bisa download [Di sini](#).

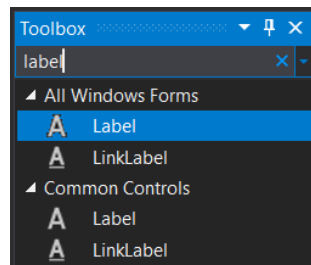
14. Untuk menghilangkan tombol-tombol windows dan membuat form tidak berubah-ubah ukuran, ubah FormBorderStyle dan AutoScaleMode seperti di bawah ini.



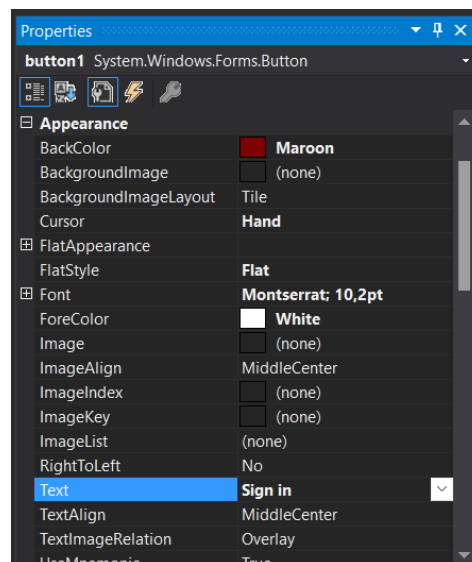
15. Untuk mengubah *font family* dari program bisa dilakukan dengan mengubahnya seperti di gambar di bawah ini.



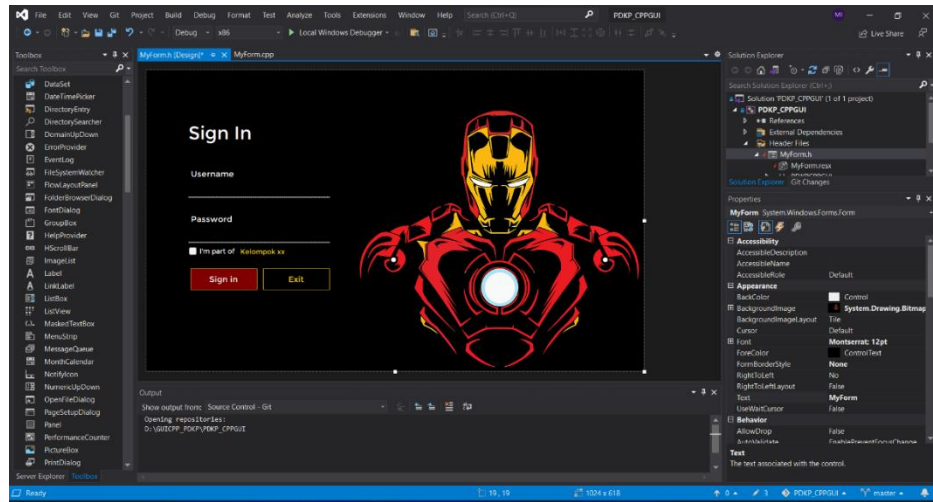
16. Tambahkan beberapa komponen dari *Toolbox* untuk membuat *label*, *textbox*, *checkbox* dan *button* seperti pada contoh.



17. Warna, posisi, ukuran, teks, nama dan behaviour tiap komponen dapat kalian ubah-ubah pada tab Properties.



18. Atur posisi, warna, teks dan ukuran tiap komponen sesuai dengan contoh atau silahkan jika ingin dikreasi sendiri.



19. Full source code: [C++ Full Program](#)