**LAPORAN PRAKTIKUM**

**OBJECT ORIENTED PROGRAMING**

**SEMESTER GANJIL TAHUN AKADEMIK 2022/2023**

****

**Disusun oleh :**

**Nama : Tesalonika Dua Nurak**

**NIM : 22181015**

**Prodi : Teknik Informatika S-1**

**Kelompok :**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**2023**

**LEMBAR PERSETUJUAN**

**PRAKTIKUM OBJECT ORIENTED PROGRAMING**

**SEMESTER GENAP TAHUN AKADEMIK 2022/2023**

****

Disusun Oleh

NAMA : Tesalonika Dua Nurak

NIM : 2218015

PRODI : Teknik Informatika S-1

|  |  |
| --- | --- |
| Mengetahui  Ka. Lab. Pemrograman Komputer  **(Ahmad Faisol, ST, MT)**  **NIP.P : 1031000431** | Menyetujui  Dosen Pembimbing  **(……………………………………….)**  **NIP.P : …** |

**PROGRAM STUDI TEKNIK INFORMATIKA S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**2023**

# KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT, karena atas berkah rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Praktikum Pemrogramman Web, guna persyaratan dalam menempuh mata kuliah tersebut.

Laporan ini disusun berdasarkan percobaan dan teori dasar yang ada dalam buku panduan praktikum, teori yang diperoleh praktikan dari perkuliahan, dan tidak lupa yaitu Internet sehingga penulis dapat menambah tidak hanya menguasai teori saja namun juga memahami serta mengaplikasikannya.

Terwujudnya laporan ini, tentunya tidak lepas dari bantuan-bantuan yang telah penulis terima. Pada kesempatan ini, penulis menyampaikan terima kasih yang sebesar-besarnya kepada yang terhormat:

1. Bapak/Ibu … selaku dosen pembimbing Praktikum Pemrograman Web.
2. Bapak Yoseph Agus Pranoto, ST, MT. dan Dedy Rudhistiar, S.Kom, M.Cs. Selaku dosen matakuliah Object Oriented Programing.
3. Bapak Ahmad Faisol, ST, MT selaku Ketua Pelaksana Praktikum Pemrograman Web Program Studi Teknik Informatika ITN Malang.
4. Instruktur Lab.Pemrograman Komputer Teknik Informatika yang telah memberi petunjuk kepada penulis selama pelaksanaan praktikum.
5. Rekan-rekan yang telah membantu dalam pelaksanaan dan penyelesaian laporan ini.

Harapan penulis, laporan praktikum ini bermanfaat bagi penulis sendiri maupun pembaca sekalian.

Malang, ... Desember 2024

Penulis

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc146489740)

[DAFTAR ISI ii](#_Toc146489741)

[DAFTAR GAMBAR iv](#_Toc146489742)

[DAFTAR TABEL v](#_Toc146489743)

[BAB I PENDAHULUAN I-1](#_Toc146489744)

[I.1 Latar Belakang I-1](#_Toc146489745)

[I.2 Rumusan Masalah I-2](#_Toc146489746)

[I.3 Tujuan I-2](#_Toc146489747)

[I.4 Manfaat I-2](#_Toc146489748)

[BAB II KONSEP DASAR OOP II-3](#_Toc146489749)

[II.1 Landasan Teori II-3](#_Toc146489750)

[II.2 Langkah – Langkah Praktikum II-3](#_Toc146489751)

[II.3 Tugas Praktikum –1: II-3](#_Toc146489752)

[II.4 Tugas Rumah 1 : II-7](#_Toc146489753)

[II.5 Kesimpulan II-8](#_Toc146489754)

[BAB III KONSTRUCTOR DAN INHERITANCE III-1](#_Toc146489755)

[III.1 Landasan Teori III-1](#_Toc146489756)

[III.2 Langkah – Langkah Praktikum III-1](#_Toc146489757)

[III.3 Tugas Praktikum 1 : III-1](#_Toc146489758)

[III.4 Tugas Praktikum 2 : III-3](#_Toc146489759)

[III.5 Tugas Rumah 1 : III-4](#_Toc146489760)

[III.6 Tugas Rumah 2 : III-5](#_Toc146489761)

[III.7 Kesimpulan III-10](#_Toc146489762)

[BAB IV ENKAPSULASI , OVERRIDING DAN OVERLOADING IV-1](#_Toc146489763)

[IV.1 Landasan Teori IV-1](#_Toc146489764)

[IV.2 Langkah – Langkah Praktikum IV-1](#_Toc146489765)

[IV.3 Tugas Praktikum 1 : IV-1](#_Toc146489766)

[IV.4 Tugas Praktikum 2 : IV-2](#_Toc146489767)

[IV.5 Tugas Rumah 1 : IV-4](#_Toc146489768)

[IV.6 Tugas Rumah 2: IV-7](#_Toc146489769)

[IV.7 Kesimpulan IV-11](#_Toc146489770)

[BAB V ABSTRACT, POLIMORFISME, DAN EXCEPTION HANDLING V-1](#_Toc146489771)

[V.1 Landasan Teori V-1](#_Toc146489772)

[V.2 Langkah – Langkah Praktikum V-1](#_Toc146489773)

[V.3 Tugas Praktikum 1 : V-1](#_Toc146489774)

[V.4 Tugas Praktikum 2 : V-2](#_Toc146489775)

[V.5 Tugas Praktikum 3 : V-4](#_Toc146489776)

[V.6 Tugas Rumah 1 : V-5](#_Toc146489777)

[V.7 Tugas Rumah 2 : V-6](#_Toc146489778)

[V.8 Tugas Rumah 3 : V-7](#_Toc146489779)

[V.9 Kesimpulan V-9](#_Toc146489780)

[BAB VI INTERFACE DAN PENGENALAN DATABASE VI-1](#_Toc146489781)

[VI.1 Landasan Teori VI-1](#_Toc146489782)

[VI.2 Langkah – Langkah Praktikum VI-1](#_Toc146489783)

[VI.3 Tugas Praktikum 1 : VI-1](#_Toc146489784)

[VI.4 Tugas Praktikum 2 : VI-3](#_Toc146489785)

[VI.5 Tugas Praktikum 3 : VI-6](#_Toc146489786)

[VI.6 Tugas Praktikum 4 : VI-9](#_Toc146489787)

[VI.7 Tugas Rumah 1 : VI-12](#_Toc146489788)

[VI.8 Tugas Rumah 2 : VI-12](#_Toc146489789)

[VI.9 Tugas Rumah 3 : VI-15](#_Toc146489790)

[VI.10 Tugas Rumah 4 : VI-18](#_Toc146489791)

[VI.11 Kesimpulan VI-21](#_Toc146489792)

[BAB VII KESIMPULAN VII-1](#_Toc146489793)

[DAFTAR PUSTAKA VII-2](#_Toc146489794)

# DAFTAR GAMBAR

# DAFTAR TABEL

# PENDAHULUAN

## Latar Belakang

*OOP (Object Oriented Programming)* adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari *OOP* diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Salah satu contoh perbedaan yang sangat sederhana antara pemrograman prosedural dengan Pemrograman berorientasi objek adalah pada pendefinisian , konstanta atau fungsi. Didalam pemrograman prosedural seluruh , konstanta, ataupun fungsi yang dibutuhkan didalam rogram wajib didefinisikan sementara didalam programan beorientasi objek semua kebutuhan , konstanta atau fungsi tersebut cukup dibuatkan di dalam sebuah objek.

Konsep dari OOP sendiri adalah semua pemecahan masalah dibagi ke dalam objek. Proses perancangan atau desain dalam suatu pemrograman merupakan proses yang tidak terpisah dari proses yang mendahului, yaitu analisis dan proses yang mengikutinya. Pembahasan mengenai orientasi objek tidak akan lepas dari konsep objek seperti *inheritance* atau penurunan, *encapsulation* atau pembungkusan, dan *polymorphism* atau kebanyakrupaan. Konsep – konsep ini merupakan fundamental dalam orientasi objek yang perlu sekali dipahami serta digunakan dengan baik, dan menghindari penggunaanya yang tidak tepat.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

## Rumusan Masalah

1. Apa itu OOP?
2. Apa tujuan dari OOP?
3. Bagaimana pengaplikasian OOP?

## Tujuan

1. Mampu memahami dengan konsep OOP
2. Mampu menerapkan dan menguasai berdasarkan konsep OOP
3. Mampu membuat program dari OOP

## Manfaat

1. Dapat mengetahui konsep dasar maupun lanjutan dari OOP
2. Dapat mengaplikasikan OOP
3. Dapat mengerti fungsi-fungsi keyword yang terdapat dalam OOP

# KONSEP DASAR OOP

|  |  |  |
| --- | --- | --- |
| **Jumlah Pertemuan** | **:** | **2 x 60 menit** |
| **Tujuan Praktikum** | **:** | 1. Praktikan mampu menerapkan konsep dasar pemrograman berorientasi objek. 2. Praktikan mampu membuat sebuah *Class*, atribut, *method* dan objek. 3. Praktikan mampu mengetahui fungsi *method* *getter* dan *setter*. 4. Praktikan mampu membuat program untuk memasukkan data mahasiswa berbasis *GUI* (*Graphical User Interface* ). |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: *Netbeans*. 3. Modul Praktikum *OOP* 2022. |

## Landasan Teori

1. Pengenalan *OOP*

OOP (*Object Oriented Programming*) adalah suatu metode pemrograman yang berbasis kepada objek. OOP mengadopsi model yang mirip dengan objek-objek di kehidupan sehari-hari. sebuah objek dalam program dapat terdiri dari beberapa atribut dan metode yang saling berhubungan dan berinteraksi, hal tersenbut memungkinkan attribut dan method memiliki peran dan fungsinya sendiri dalam membentuk suatu objek secara keseluruhan.

1. Class

Class juga merupakan rancangan dari sebuah objek yang medefenisikan atribut (ciri/variabel) dan method (perilaku) umum dari suatu objek tertentu.

1. Drive Class : Class yang menggunakan main atau dapat di running.
2. Object Class : Class yang tidak ada main, biasanya berisi method dan attribute yang nantinya akan di panggil di Driver Class. Class dideklarasikan sebagai berikut:
3. Object

Object adalah realisasi dari class. Dapat dibayangkan, class adalah sebuah cetakan/tamplate , dan object adalah bentuk dari representasi cetakan/tamplate dari class. Setiap object akan mempunyai state / keadaan (instance variabel/properties) yang membedakan satu object dengan object lain. Terdapat istilah “instansiasi” dalam OOP yaitu proses pembuatan object real dari class. Ketika kita membuat sebuah class bukan berarti kita membuat sebuah objek. Atribut adalah karakteristik unik atau ciri dari sebuah objek. Karakteristik tersebut dapat berupa data/variabel yang akan dimiliki oleh objek dari kelas tersebut.

1. Attribut

Atribut adalah karakteristik unik atau ciri dari sebuah objek. Karakteristik tersebut dapat berupa data/variabel yang akan dimiliki oleh objek dari kelas tersebut. Atribut dapat memiliki hak akses private, public maupun protected (akan dijelaskan di bab 4).

Jenis atribut umum dalam OOP Java:

1. Atribut Instance (Instance Variables)

Atribut yang berkaitan dengan objek yang dihasilkan dari class yang dipakai. Atribut ini mendefinisikan karakteristik atau properti unik dari objek.

1. Atribut Static (Static Variables)

Atribut yang berkaitan dengan class itu sendiri, bukan dengan objek. Biasanya digunakan untuk menyimpan data yang bersifat bersama antara semua objek kelas.

1. Atribut Final (Final Variables)

Atribut yang nilai awalnya harus ditetapkan saat deklarasi dan tidak dapat diubah lagi setelahnya. Attribut ini digunakan untuk menyimpan konstanta atau nilai tetap yang tidak boleh diubah.

1. Method

Method adalah aksi atau tindakan yang dapat dilakukan oleh objek dari suatu kelas. Method beroperasi pada objek atau kelas dan dapat digunakan untuk melakukan tindakan tertentu seperti mengembalikan nilai, atau mengubah keadaan objek. Method berfungsi mengorganisir dan mengelompokkan kode, sehingga tindakan atau fungsi tertentu dapat dijalankan dengan cara yang terstruktur.

Jenis metode dalam OOP Java:

1. Instance Methods

Instance Methods adalah metode dalam OOP yang terkait langsung dengan objek spesifik yang diciptakan dari suatu kelas. Instance method berjalan pada atribut dan perilaku objek tertentu yang diwakili oleh instansiasi objek yang telah dibuat dari kelas tersebut.

1. Static Methods

Static methods adalah method yang terkait dengan kelas itu sendiri, bukan dengan objek yang dibuat dari kelas. method static diakses melalui nama kelas tanpa perlu membuat objek dari kelas tersebut. Karena method ini tidak beroperasi pada objek spesifik, mereka tidak dapat mengakses atribut non-static kelas atau variabel anggota non-static.

1. Final Methods

Pada metode final tidak dapat di-override oleh kelas turunan atau "child class". Final methods digunakan ketika ingin mencegah kelas turunan untuk mengubah perilaku khusus dari metode yang ada dalam kelas dasar. Method final digunakan untuk menjaga konsistensi perilaku metode.

## Langkah – Langkah Praktikum

1. Buka Aplikasi *Netbeans*.
2. Buat *Class*.
3. Memberi *script* pada kelas tersebut.
4. Buat *form*.
5. Memberi *script* pada *form* tersebut.
6. Menjalankan program.

## Tugas Praktikum –1:

Membuat kelas Mahasiswa.java dan GUI\_Mahasiswa.java

Diagram Class (Class Mahasiswa):

|  |
| --- |
| Mahasiswa |
| ~ nim :String  ~ nama : String  ~ JenisKelamin : String  ~ prodi : String  ~ Angkatan : String  ~ Alamat : String |
| ~ void dataNIM(String Nim)  ~ void dataNama(String Nama)  ~ void dataJK(String JenisKelamin)  ~ void dataProdi(String Prodi)  ~ void dataAngkatan(String Angkatan)  ~ void dataStatus(String Status)  ~ cetakNIM() : String  ~ cetakNama() : String  ~ cetakJenisKelamin() : String  ~ cetakProdi() : String  ~ cetakAngkatan() : String  ~ cetakStatus() : String |

*Source code* (Mahasiswa.java) :

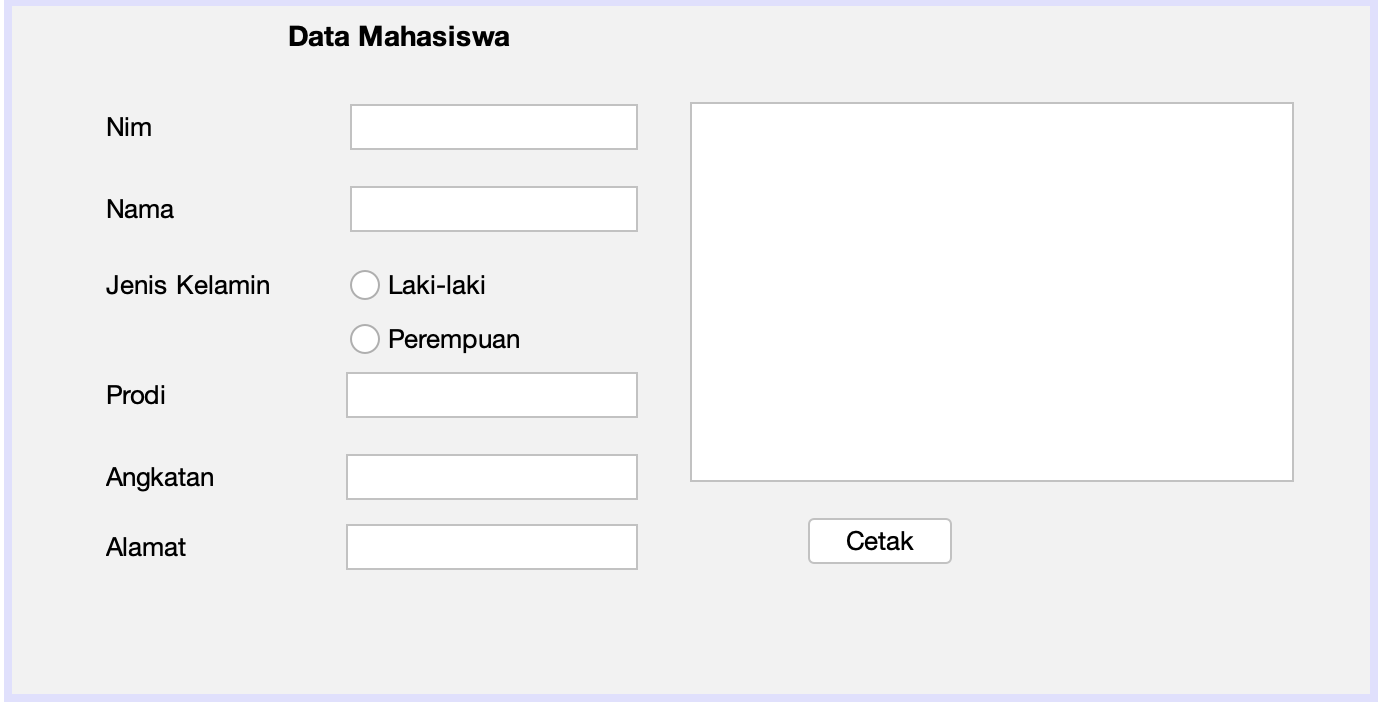
**Method**

**Attribute**

**Class**

|  |
| --- |
| public class Mahasiswa {  String nim, nama, prodi, angktn;  void dataNIM(String Nim){  this.nim = Nim;  }  void dataNama(String Nama){  this.nama = Nama;  }  void dataProdi(String Prodi){  this.prodi = Prodi;  }  void dataAngkatan(String Angktn){  this.angktn = Angktn;  }    String cetakNIM(){  return nim;  }  String cetakNama(){  return nama;  }  String cetakProdi(){  return prodi;  }  String cetakAngkatan(){  return angktn;  }  } |

Desain *form* (GUI\_Mahasiswa.java):



### Desain form mahasiswa

#### Properti Desain GUI\_Mahasiswa.java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Komponen** | **Properti** | **Nilai** |
| 1 | jLabel1 | Text | DATA MAHASISWA |
| 2 | jLabel2 | Text | NIM |
| 3 | jLabel3 | Text | Nama |
| 4 | jLabel4 | Text | Jenis Kelamin |
| 5 | jLabel5 | Text | Prodi |
| 6 | jLabel6 | Text | Angkatan |
| 7 | jLabel7 | Text | Alamat |
| 8 | jTextField1 | Name | txtNim |
| Text |  |
| 9 | jTextField2 | Name | txtNama |
| Text |  |
| 10 | jTextField3 | Name | txtProdi |
| Text |  |
| 11 | jTextField4 | Name | txtAngkatan |
| Text |  |
| 12 | jTextField5 | Name | txtAlamat |
| Text |  |
| 13 | jRadioButton1 | Name | radiobtnLaki |
| Text | Laki-laki |
| 14 | jRadioButton2 | Name | radiobtnPerempuan |
| Text | Perempuan |
| 15 | jButton1 | Name | btnKTM |
| Text | Cetak KTM |
| 16 | jButton2 | Name | btnCLOSE |
| Text | Close |
| 17 | jTextArea | Name | memoKTM |
| Text |  |

*Source code* pada button Cetak KTM*:*

|  |
| --- |
| private void CetakActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  // TODO add your handling code here:  memoKTM.setText("");  Mahasiswa mhs = new Mahasiswa();  mhs.dataNIM(txtNim.getText());  mhs.dataNama(txtNama.getText());  String JenKel="";  if (radiobtnLaki.isSelected()){  mhs.dataJenisKelamin(radiobtnLaki.getText());  }else{  mhs.dataJenisKelamin(radiobtnPerempuan.getText());  }  mhs.dataProdi(txtProdi.getText());  mhs.dataAngkatan(txtAngkatan.getText());  mhs.dataAlamat(txtAlamat.getText());    memoKTM.append("Kartu Tanda Mahasiswa\n");  memoKTM.append("--------------------------------------------------------\n");  memoKTM.append("NIM : " + mhs.cetakNIM() +"\n");  memoKTM.append("Nama : " + mhs.cetakNama() + "\n");  memoKTM.append("Jenis Kelamin : " + mhs.cetakJenisKelamin() + "\n");  memoKTM.append("Prodi : " + mhs.cetakProdi() +"\n");  memoKTM.append("Angkatan : " + mhs.cetakAngkatan() + "\n");  memoKTM.append("Alamat : " + mhs.cetakAlamat() + "\n");  } |

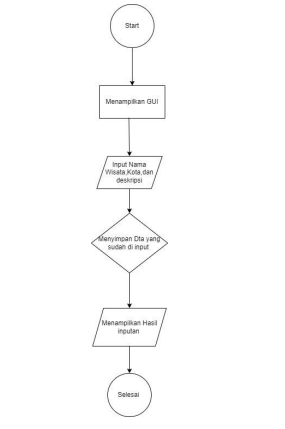
Tampilan Hasil :



### Hasil Gui\_mahasiswa

## Tugas Rumah 1 :

Flowchart (Sistem Informasi Wisata )



### Flowchart Wisata

Analisa :

Pertama program akan dimulai dengan menginput nama,Kota,Dan deskripsi wisata. Kemudian program akan menyimpan semua data inputan tadi. Berikutnya program akan menampilkan data input dalam bentuk List wisata yang sesuai.

## Tugas Rumah 2 :

Judul : Sitem Informasi Wisata

Tema : Sistem Informasi

*Diagram Class* (Class List.java):

|  |
| --- |
| List |
| ~ Nama : String  ~ Kota : String  ~ Deskripsi: String |
| ~ void dataNIM(String Nim)  ~ void dataNama(String Nama)  ~ void dataJK(String JenisKelamin)  ~ void dataProdi(String Prodi)  ~ void dataAngkatan(String Angkatan)  ~ void dataStatus(String Status)  ~ cetakNIM() : String  ~ cetakNama() : String  ~ cetakJenisKelamin() : String  ~ cetakProdi() : String  ~ cetakAngkatan() : String  ~ cetakStatus() : String |

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## 

## Kesimpulan

* + - 1. OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berbasis kepada objek.
      2. Class merupakan rancangan dari sebuah objek yang mendefenisikan atribut (ciri/variabel) dan method (perilaku) umum dari suatu objek yang dibuat .
      3. Atribut adalah karakteristik unik atau ciri dari sebuah objek. Karakteristik tersebut dapat berupa data/variabel yang akan dimiliki oleh objek dari kelas tersebut

# KONSTRUCTOR DAN INHERITANCE

|  |  |  |
| --- | --- | --- |
| **Jumlah Pertemuan** | **:** | **2 x 60 menit** |
| **Tujuan Praktikum** | **:** | 1. Praktikan mampu mengetahui tipe data dan *java* *identifier*. 2. Praktikan mampu merubah tipe data (*Casting*). 3. Praktikan mampu mengetahui dan membuat *method* konstruktor. 4. Praktikan mampu mengetahui Inheritance |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: *Netbeans*. 3. Modul Praktikum *OOP* 2023. |

## Landasan Teori

1. Konstructor

Penjelasan inheritance……………..

1. Inheritance

Penjelasan inheritance……………..

## Langkah – Langkah Praktikum

* + - 1. Buka Aplikasi *Netbeans*.
      2. Buat *Class*.
      3. Memberi *script* pada kelas tersebut.
      4. Buat *form*.
      5. Memberi *script* pada *form* tersebut.
      6. Menjalankan program.

## Tugas Praktikum 1 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):



### Desain Gui\_Matakuliah .java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Praktikum 2 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* (Cetak) pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Rumah 1 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Tugas Rumah 2 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

Membuat *Class* …………… dan GUI\_...............

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

Membuat *Class* …………… dan GUI\_...............

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Kesimpulan

* + - 1. …………………………………………………….
      2. ………………………………………………………
      3. …………………………………………………………….
      4. ………………………………………………………….

# ENKAPSULASI , OVERRIDING DAN OVERLOADING

|  |  |  |
| --- | --- | --- |
| **Jumlah Pertemuan** | **:** | **2 x 60 menit** |
| **Tujuan Praktikum** | **:** | 1. Praktikan mampu memahami tentang hak akses (Enkapsulasi). 2. Praktikan mampu mengimplementasikan konsep *Encapsulasi* kedalam program. 3. Praktikan mampu memahami dan membuat *method override*. |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: *Netbeans*. 3. Modul Praktikum *OOP* 2022. |

## Landasan Teori

## Langkah – Langkah Praktikum

1. Buka Aplikasi *Netbeans*.
2. Buat *Class*.
3. Memberi *script* pada kelas tersebut.
4. Buat *form*.
5. Memberi *script* pada *form* tersebut.
6. Menjalankan program.

## Tugas Praktikum 1 :

Implementasi Enkapsulasi *Class* ……………

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Praktikum 2 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Rumah 1 :

Implementasi Enkapsulasi *Class* ……………

Judul Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

Implementasi Enkapsulasi *Class* ……………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

Implementasi Enkapsulasi *Class* ……………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Tugas Rumah 2:

Implementasi Overrloading dan Overloading

Judul : ……………………………………

*Diagram Class* Overriding(nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

*Diagram Class* Overloading (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Kesimpul

# ABSTRACT, POLIMORFISME, DAN EXCEPTION HANDLING

|  |  |  |
| --- | --- | --- |
| **Jumlah Pertemuan** | **:** | **2 x 60 menit** |
| **Tujuan Praktikum** | **:** | 1. Mampu menerapkan konsep dasar pemrograman web. 2. Mampu mengetahui fungsi tag-tag pada html. 3. Mampu membuat halaman web menggunakan html. |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: VS Code, XAMPP. 3. Modul Praktikum WEB 2023. |

## Landasan Teori

1. Abstract
2. Polimorfisme
3. Exception

## Langkah – Langkah Praktikum

1. Buka Aplikasi *Netbeans*.
2. Buat *Class*.
3. Memberi *script* pada kelas tersebut.
4. Buat *form*.
5. Memberi *script* pada *form* tersebut.
6. Menjalankan program.

## Tugas Praktikum 1 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

* + - * 1. **SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Praktikum 2 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Praktikum 3 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

## Tugas Rumah 1 :

Implementasi Class Abstract

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Tugas Rumah 2 :

Menambahkan Exception Handling pada project

Judul : ……………………………………

*Source code Object Class ……..:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Tugas Rumah 3 :

Menerapkan Polimorfisme pada *Class* ……………

Judul : ……………………………………

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

Hasil Tampilan:

**SS Hasil Tampilan GUI**

### Hasil Tampilan GUI\_.............java

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Kesimpulan

# INTERFACE DAN PENGENALAN DATABASE

|  |  |  |
| --- | --- | --- |
| **Jumlah Pertemuan** | **:** | **2 x 60 menit** |
| **Tujuan Praktikum** | **:** | 1. Praktikan mampu memahami tentang *exception* dan menerapkan *exception* ke dalam program. 2. Praktikan mampu membuat *database* dengan menggunakan MySQL. 3. Praktikan mampu mengkoneksikan *database* dengan program yang sudah dibuat sebelumnya. 4. Praktikan mampu memberi akses *insert, update, delete,* dan *searching* (memasukkan, merubah, menghapus, dan pencarian). |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: *Netbeans*. 3. Modul Praktikum *OOP* 2022. |

## Landasan Teori

1. Interface
2. Pengenalan Database

## Langkah – Langkah Praktikum

1. Buka Aplikasi *Netbeans*.
2. Buat *Class*.
3. Memberi *script* pada kelas tersebut.
4. Buat *form*.
5. Memberi *script* pada *form* tersebut.
6. Menjalankan program.

## Tugas Praktikum 1 :

Membuat *Class* …………… dan GUI\_...............

Judul : ……………………………………

*Diagram Class* (nama\_diagram):

Masukkan class diagram disini

*Source code Object Class:*

|  |
| --- |
|  |

Desain *form* (GUI\_............java):

**SS Desain GUI**

### Desain Gui\_.............java

#### Properti Desain GUI\_..................java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |
| 7 |  |  |  |
|  |  |  |
| 8 |  |  |  |
|  |  |  |
| 9 |  |  |  |
|  |  |  |

*Source code Button* Cetak pada GUI:

|  |
| --- |
|  |

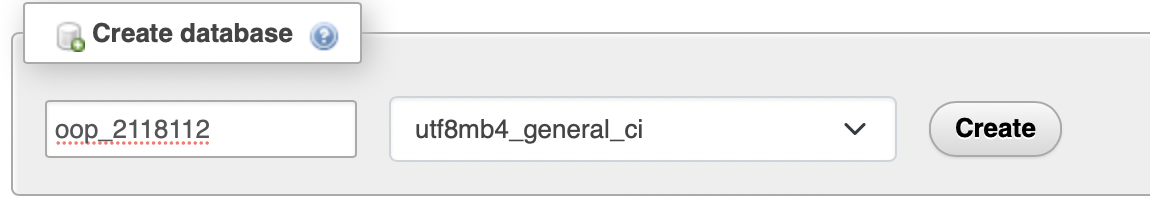
Hasil Tampilan:

**SS Hasil Tampilan GUI**

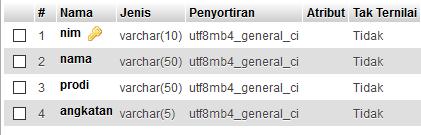
### Hasil Tampilan GUI\_.............java

## Tugas Praktikum 2 :

Mengkoneksikan GUI\_Mahasiswa Kedalam Database

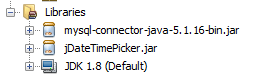
**

*Structure* tb\_mahasiswa pada database oop\_NIM



### Membuat table(tb\_mahasiswa)

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**



### Menambahkan *Library Connection*

*Source code* pada GUI (import *Library*):

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code* di bawah Konstruktor GUI\_Mahasiswa*:*

|  |
| --- |
| public Connection conn; |

*Source code method* koneksi *:*

|  |
| --- |
|  |

*Source code* di *method* tampil*:*

|  |
| --- |
|  |

*Source code method refresh :*

|  |
| --- |
|  |

*Source code method tambah :*

|  |
| --- |
|  |

*Source code method ubah :*

|  |
| --- |
|  |

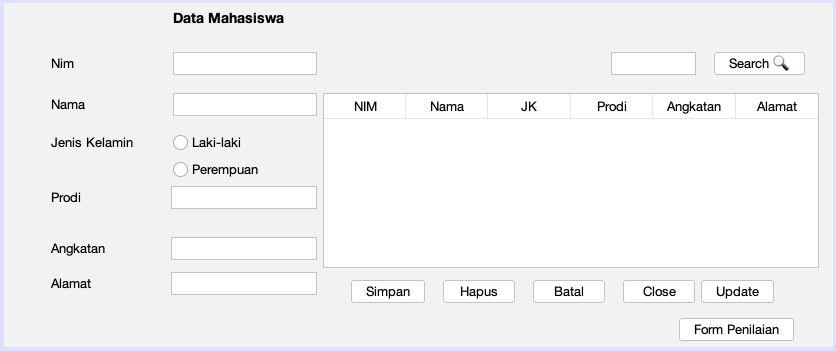
*Source code method hapus :*

|  |
| --- |
|  |

*Source code method cari :*

|  |
| --- |
|  |

Desain *form* GUI Mahasiswa:



### Desain Form Gui\_Mahasiswa

#### Properti Desain GUI\_Mahasiswa (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Komponen** | **Properti** | **Nilai** |
| 1 | jLabel1 | Text | DATA MAHASISWA |
| 2 | jLabel2 | Text | NIM |
| 3 | jLabel3 | Text | Nama |
| 4 | jLabel4 | Text | Jenis Kelamin |
| 5 | jLabel5 | Text | Prodi |
| 6 | jLabel6 | Text | Angkatan |
| 7 | jLabel7 | Text | Alamat |
| 8 | jLabel8 | Text | Search |
| 9 | jTextField1 | Name | txtNim |
| Text |  |
| 10 | jTextField2 | Name | txtNama |
| Text |  |
| 11 | jTextField3 | Name | txtProdi |
| Text |  |
| 12 | jTextField4 | Name | radiobtnLaki |
| Text | Laki-laki |
| 13 | jTextField5 | Name | radiobtnPerempuan |
| Text | Perempuan |
| 14 | jTextField6 | Name | txtAngkatan |
| Text |  |
| 15 | jTextField7 | Name | txtAlamat |
| Text |  |
| 16 | jTextField8 | Name | txtSearch |
| Text |  |
| 17 | jButton1 | Name | btnSimpan |
| Text | Simpan |
| 18 | jButton2 | Name | btnHapus |
| Text | Hapus |
| 19 | jButton3 | Name | btnClose |
| Text | Close |
| 20 | jButton4 | Name | btnBatal |
| Text | Batal |
| 21 | jButton5 | Name | btnPenilaian |
| Text | Form Penilaian |
| 22 | jTable | Name | table\_data\_mahasiswa |
| Text |  |

*Source code* pada tombol tambah*:*

|  |
| --- |
|  |

*Source code* pada tombol Ubah*:*

|  |
| --- |
|  |

*Source code* pada tombol hapus*:*

|  |
| --- |
|  |

*Source code* pada tombol cari*:*

|  |
| --- |
|  |

*Source code* pada tombol batal*:*

|  |
| --- |
|  |

Tampilan Hasil:

SS HASIL

### Tampilan Hasil Method tampil() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method tambah() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method ubah() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method hapus() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method batal() Gui\_Mahasiswa

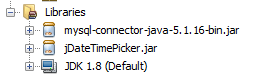
## Tugas Praktikum 3 :

Mengkoneksikan GUI\_Nilai Kedalam Database

**SS Struktur Tabel tb\_nilai**

### Membuat table (tb\_nilai)

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**



### Menambhakan *Library Connection*

*Source code* pada GUI (import *Library*):

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code* di bawah Konstruktor GUI\_Nilai*:*

|  |
| --- |
| public Connection conn; |

*Source code method* koneksi *:*

|  |
| --- |
|  |

*Source code* di *method* tampil*:*

|  |
| --- |
|  |

*Source code method refresh :*

|  |
| --- |
|  |

*Source code method tambah :*

|  |
| --- |
|  |

*Source code method ubah :*

|  |
| --- |
|  |

*Source code method hapus :*

|  |
| --- |
|  |

*Source code method cari :*

|  |
| --- |
|  |

Desain *form* GUI Nilai:

**SS Desain GU\_Nilai**

### Desain Form Gui\_Nilai

#### Properti Desain GUI\_ Nilai(Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |

*Source code* pada tombol tambah*:*

|  |
| --- |
|  |

*Source code* pada tombol Ubah*:*

|  |
| --- |
|  |

*Source code* pada tombol hapus*:*

|  |
| --- |
|  |

*Source code* pada tombol cari*:*

|  |
| --- |
|  |

*Source code* pada tombol batal*:*

|  |
| --- |
|  |

Tampilan Hasil:

SS HASIL

### Tampilan Hasil Method tampil() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method tambah() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method ubah() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method hapus() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method batal() Gui\_Mahasiswa

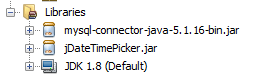
## Tugas Praktikum 4 :

Mengkoneksikan GUI\_Matkul Kedalam Database

**SS Struktur Tabel tb\_matkul**

### Membuat table (tb\_matkul)

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**



### Menambhakan *Library Connection*

*Source code* pada GUI (import *Library*):

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code* di bawah Konstruktor GUI\_Matkul*:*

|  |
| --- |
| public Connection conn; |

*Source code method* koneksi *:*

|  |
| --- |
|  |

*Source code* di *method* tampil*:*

|  |
| --- |
|  |

*Source code method refresh :*

|  |
| --- |
|  |

*Source code method tambah :*

|  |
| --- |
|  |

*Source code method ubah :*

|  |
| --- |
|  |

*Source code method hapus :*

|  |
| --- |
|  |

*Source code method cari :*

|  |
| --- |
|  |

Desain *form* GUI Matkul:

**SS Desain GU\_Matkul**

### Desain Form Gui\_Matkul

#### Properti Desain GUI\_ Matkul (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |

*Source code* pada tombol tambah*:*

|  |
| --- |
|  |

*Source code* pada tombol Ubah*:*

|  |
| --- |
|  |

*Source code* pada tombol hapus*:*

|  |
| --- |
|  |

*Source code* pada tombol cari*:*

|  |
| --- |
|  |

*Source code* pada tombol batal*:*

|  |
| --- |
|  |

Tampilan Hasil:

SS HASIL

### Tampilan Hasil Method tampil() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method tambah() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method ubah() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method hapus() Gui\_Mahasiswa

SS HASIL

### Tampilan Hasil Method batal() Gui\_Mahasiswa

## Tugas Rumah 1 :

Menerapkan *Interface*  pada class\_..................

Diagram Class…….:

*Source code* pada *Super Class* nama\_class:

|  |
| --- |
|  |

*Souce code* pada *Sub Class* ………………:

|  |
| --- |
|  |

Tampilan Hasil:

**SS Hasil Tampilan Program**

### 1Tampilan Hasil Method tampil() GUI……….

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program keseluruhan dari class objek hingga alur GUI

## Tugas Rumah 2 :

Menerapkan CRUD Pada GUI\_............................

Membuat Database oop\_judul



### Membuat database oop\_judul

*Structure* tb\_................ pada database oop\_judul

*SS hasil*

### Membuat table(tb\_...........)

*Source code* pada GUI:

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code* di bawah Konstruktor GUI\_........................*:*

|  |
| --- |
|  |

*Source code method* koneksi *:*

|  |
| --- |
|  |

*Source code* di *method* tampil*:*

|  |
| --- |
|  |

*Source code method refresh :*

|  |
| --- |
|  |

*Source code method* tambah *:*

|  |
| --- |
|  |

*Source code method* ubah *:*

|  |
| --- |
|  |

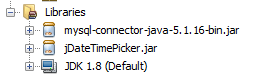
*Source code method* hapus *:*

|  |
| --- |
|  |

*Source code method cari :*

|  |
| --- |
|  |

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**



### Menambahkan *Library Connection*

Desain *form* GUI\_..............................:

**SS Desain GUI\_...............................**

### Desain Form GUI\_…….

#### Properti Desain GUI\_.................... (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |

*Source code* pada tombol Tambah*:*

|  |
| --- |
|  |

*Source code* pada tombol Ubah*:*

|  |
| --- |
|  |

*Source code* pada tombol Hapus*:*

|  |
| --- |
|  |

*Source code* pada tombol cari*:*

|  |
| --- |
|  |

*Source code* pada tombol Batal*:*

|  |
| --- |
|  |

Tampilan Hasil:

SS HASIL

### Tampilan Hasil Method Tampi() GUI\_.......

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program GUI keseluruhan

## Tugas Rumah 3 :

Menerapkan CRUD Pada GUI\_............................

*Structure* tb\_................ pada database oop\_judul

*SS hasil*

### Membuat table(tb\_...........)

*Source code* pada GUI:

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code* di bawah Konstruktor GUI\_........................*:*

|  |
| --- |
|  |

*Source code method* koneksi *:*

|  |
| --- |
|  |

*Source code* di *method* tampil*:*

|  |
| --- |
|  |

*Source code method refresh :*

|  |
| --- |
|  |

*Source code method* tambah *:*

|  |
| --- |
|  |

*Source code method* ubah *:*

|  |
| --- |
|  |

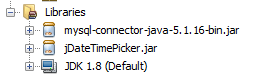
*Source code method* hapus *:*

|  |
| --- |
|  |

*Source code method cari :*

|  |
| --- |
|  |

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**



### Menambahkan *Library Connection*

Desain *form* GUI\_..............................:

**SS Desain GUI\_...............................**

### Desain Form GUI\_…….

#### Properti Desain GUI\_.................... (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |

*Source code* pada tombol Tambah*:*

|  |
| --- |
|  |

*Source code* pada tombol Ubah*:*

|  |
| --- |
|  |

*Source code* pada tombol Hapus*:*

|  |
| --- |
|  |

*Source code* pada tombol cari*:*

|  |
| --- |
|  |

*Source code* pada tombol Batal*:*

|  |
| --- |
|  |

Tampilan Hasil:

SS HASIL

### Tampilan Hasil Method Tampi() GUI\_.......

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program GUI keseluruhan

## Tugas Rumah 4 :

Menerapkan CRUD Pada GUI\_............................

*Structure* tb\_................ pada database oop\_judul

*SS hasil*

### Membuat table(tb\_...........)

*Source code* pada GUI:

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code* di bawah Konstruktor GUI\_........................*:*

|  |
| --- |
|  |

*Source code method* koneksi *:*

|  |
| --- |
|  |

*Source code* di *method* tampil*:*

|  |
| --- |
|  |

*Source code method refresh :*

|  |
| --- |
|  |

*Source code method* tambah *:*

|  |
| --- |
|  |

*Source code method* ubah *:*

|  |
| --- |
|  |

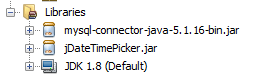
*Source code method* hapus *:*

|  |
| --- |
|  |

*Source code method cari :*

|  |
| --- |
|  |

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**



### Menambahkan *Library Connection*

Desain *form* GUI\_..............................:

**SS Desain GUI\_...............................**

### Desain Form GUI\_…….

#### Properti Desain GUI\_.................... (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objek** | **Properti** | **Nilai** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |

*Source code* pada tombol Tambah*:*

|  |
| --- |
|  |

*Source code* pada tombol Ubah*:*

|  |
| --- |
|  |

*Source code* pada tombol Hapus*:*

|  |
| --- |
|  |

*Source code* pada tombol cari*:*

|  |
| --- |
|  |

*Source code* pada tombol Batal*:*

|  |
| --- |
|  |

Tampilan Hasil:

SS HASIL

### Tampilan Hasil Method Tampi() GUI\_.......

Analisa:

Minimal 3 kalimat dalam 1 Paragraf, menyatakan alur program GUI keseluruhan

## Kesimpulan

# KESIMPULAN

Dengan adanya kegiatan praktikum Pemrograman Web. Penulis dapat menyimpulkan sebagai berikut

# DAFTAR PUSTAKA

**LEMBAR ASISTENSI PRAKTIKUM OBJECT ORIENTED PROGRAMING**

**SEMESTER GENAP TAHUN AKADEMIK 2023/2024**

Foto latar merah

Almamater

**Nama : Tesalonika Dua Nurak**

**NIM : 2218015**

**Kelompok :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Tanggal** | **Asistensi** | | | | **Paraf** |
| **Konsep** | | **Program** | **Hasil Akhir** |
| 1 |  | **Instruktur** | Konsep Dasar OOP  *Konstructor*, *Inheritance*  *Enkapsulasi*  *Overloading*, *Overriding*  *Abstract* , *Polimorfisme*  *Interface*  CRUD Java |  |  |  |
| 2 |  | **Dosen** |  |  |  |  |
| **Batas Akhir:** | | | | | | |

**Asisten,**

**(………………………………………….)**

**NIM :**

Malang, 2024

**Dosen Pembimbing,**

**(……………………………………………..)  
NIP.P :**