

LAPORAN TUGAS PROYEK
“Menghitung Nilai Akhir Mahasiswa
Program Studi Teknik Informatika”



ST. HATIYAH H. ILYAS

13020180061

B1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MUSLIM INDONESIA

MAKASSAR

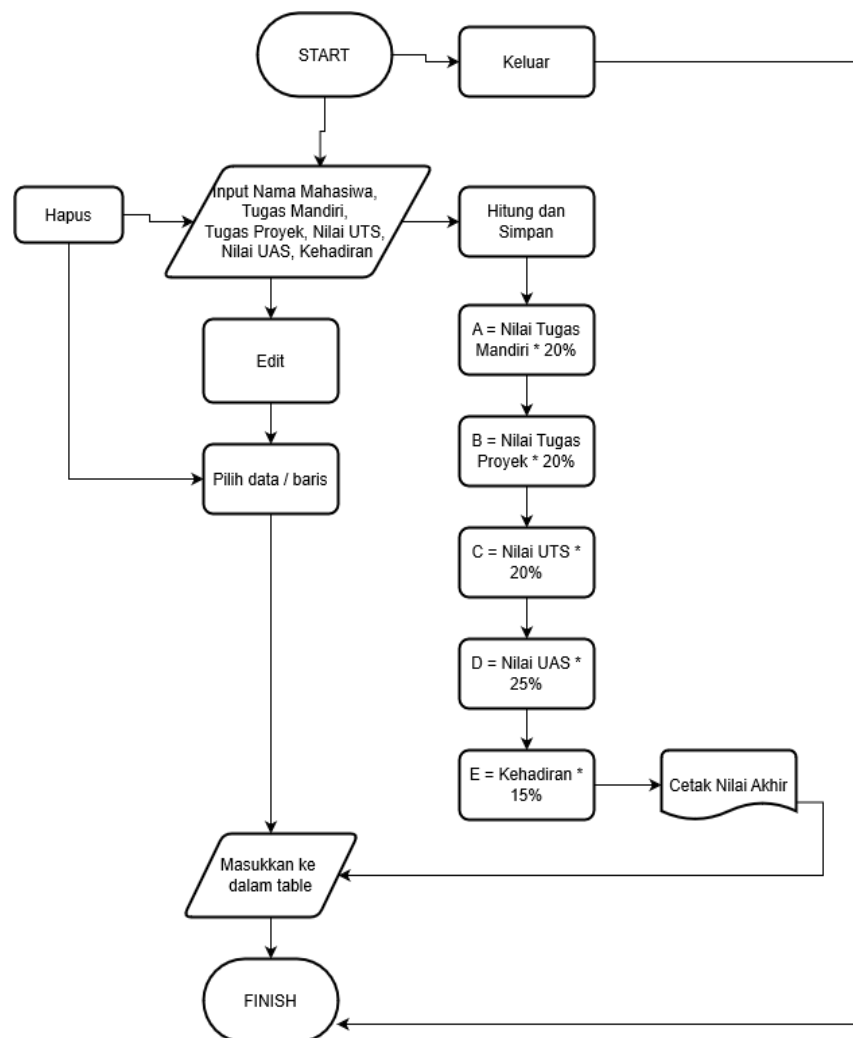
2020

I. PENDAHULUAN

Di era globalisasi saat ini perkembangan teknologi sangatlah maju dengan pesat dan cepat. Hal seperti ini telah membuat setiap individu terus berusaha dengan berbagai cara agar dapat mengikuti setiap perkembangan teknologi yang ada, sehingga pengetahuan mereka semakin bertambah dan maju seiring perkembangan zaman. Computer merupakan sebuah perkembangan teknologi yang sering dimanfaatkan untuk mencari berbagai macam informasi yang dibutuhkan oleh manusia. Bahkan sekarang ini computer secara tidak langsung telah menjadi suatu unsur yang sangat penting dan sangat berguna bagi kehidupan manusia. Oleh karena itu, teknologi computer sudah dapat mengatasi berbagai masalah dalam pekerjaan manusia, termasuk menciptakan aplikasi yang dapat berguna untuk mempermudah manusia untuk memanfaatkan waktu dengan lebih baik.

Program Menghitung Nilai Mahasiswa ini sangat diperlukan didalam sebuah universitas, khususnya untuk mahasiswa dalam mengetahui nilai akhir dan grade dengan menghitung nilai dari Tugas Mandiri/Kuis, Tugas Proyek, UTS, UAS, dan kehadiran.

II. DESKRIPSI UMUM SISTEM



ANALISIS

Dalam pembuatan program ini, penulis menggunakan beberapa software dan hardware sebagai pendukung dalam pembuatan aplikasi ini. Software dan hardware yang digunakan adalah:

Hardware

Harddisk	: 500 GB
RAM	: 2 GB
Prosesor	: 1.10 GHz
VGA	: -

Software

A. Netbeans IDE

Netbeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). Sun Microsystems mendirikan proyek kode terbuka Netbeans pada bulan juni 2000 dan terus menjadi sponsor utama.

The Netbeans IDE adalah sebuah lingkungan pengembangan – sebuah kakas untuk pemrograman menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java – namun dapat mendukung Bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa Batasan bagaimana digunakan.

B. XAMMP

XAMMP adalah perangkat lunak bebas yang mendukung banyak system operasi dan merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah Bahasa yang ditulis dengan Bahasa pemrograman PHP dan Perl. Nama XAMMP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (General Public License) dan merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

C. MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (Bahasa inggris: database management system) atau DBMS yang multithread, multi user dengan sekitar 6 juta instalasi di seluruh dunia, MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

D. Encapsulation

Encapsulation atau pengkapsulan adalah konsep tentang pengikatan data atau metode yang berbeda yang disatukan atau “dikapsulkan” menjadi satu unit data. Encapsulation dapat mempermudah dalam pembacaan code karena informasi yang disajikan tidak perlu dibaca secara rinci dan sudah merupakan satu kesatuan.

E. Inheritance

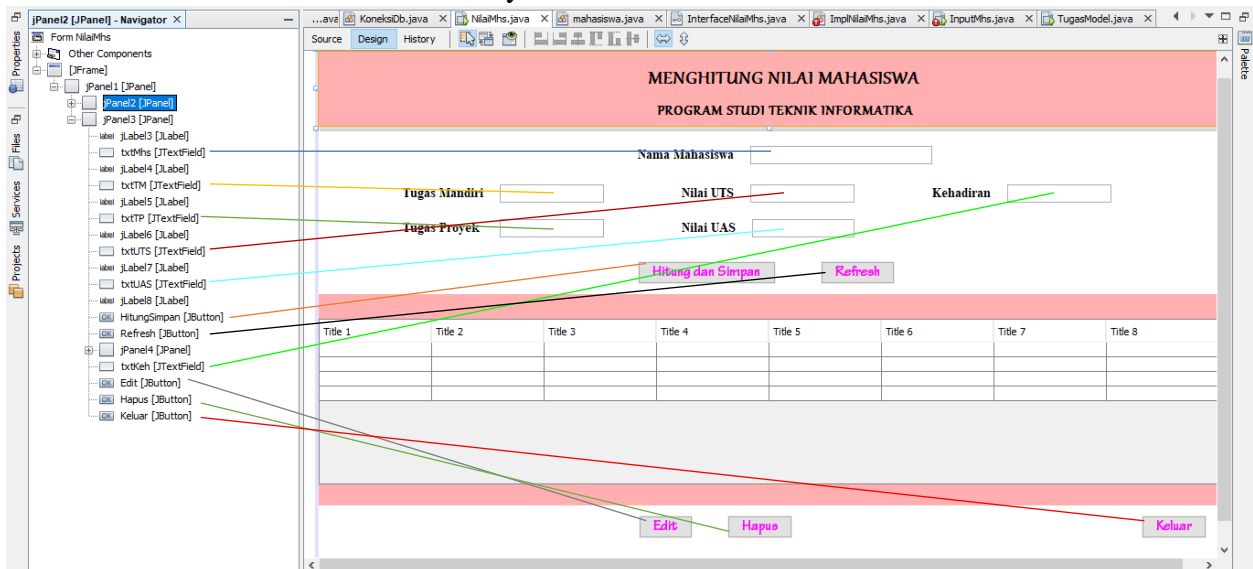
Inheritance adalah konsep OOP dimana kita dapat membentuk class baru yang “mewarisi” atau memiliki bagian-bagian dari class yang sudah ada sebelumnya. Konsep ini menggunakan system hirarki atau bertingkat.

F. Polymorphism

Polymorphism adalah konsep dimana suatu objek yang berbeda-beda dapat diakses melalui interface yang sama. Sebuah objek yang polymorphic dapat beradaptasi dengan metode apapun yang di implementasikan pada objek tersebut, dan setiap class memiliki interpretasinya tersendiri terhadap interfacenya.

IMPLEMENTASI

A. Desain form beserta nama variable - nya



B. Syntax MySQL dan class *KoneksiDb* untuk menghubungkan database ke netbeans

```
1 • create database Mahasiswaaaa;
2 • use Mahasiswaaaa;
3
4 • create table table_mahasiswaa ( `Stambuk` varchar(5) primary key,
5   `Nama` varchar(30),
6   `Jurusan` varchar(15), `Email` varchar(30));
```

```
10 public class KoneksiDb {
11     static Connection con = null;
12
13     public static Connection getConnection() {
14         if (con == null) {
15             try {
16                 String url = "jdbc:mysql://localhost:3306/nilai_mhs";
17                 String user = "root";
18                 String password = "";
19                 DriverManager.registerDriver(new com.mysql.jdbc.Driver());
20                 con = DriverManager.getConnection(url, user, password);
21                 System.out.println("Terkoneksi");
22             } catch (SQLException t) {
23                 System.out.println("Error membuat koneksi");
24             }
25         }
26         return con;
27     }
28 }
```

C. Syntax class *mahasiswa*

```

Source History
9 public class mahasiswa {
10     String nama;
11     int tgsM;
12     int tgsP;
13     int uts, uas, kehadiran;
14
15     public String getNama() {
16         return nama;
17     }
18
19     public void setNama(String nama) {
20         this.nama = nama;
21     }
22
23     public int getTgsM() {
24         return tgsM;
25     }
26
27     public void setTgsM(int tgsM) {
28         this.tgsM = tgsM;
29     }
30
31     public int getTgsP() {
32         return tgsP;
33     }
34
35     public void setTgsP(int tgsP) {
36         this.tgsP = tgsP;
37     }
38
39     public int getUts() {
40         return uts;
41     }
42
43     public void setUts(int uts) {
44         this.uts = uts;
45     }
46
47     public int getUas() {
48         return uas;
49     }
50
51     public void setUas(int uas) {
52         this.uas = uas;
53     }
54
55     public int getKehadiran() {
56         return kehadiran;
57     }
58
59     public void setKehadiran(int kehadiran) {
60         this.kehadiran = kehadiran;
61     }
62 }

```

D. Syntax class *InterfaceNilaiMhs*

```

Source History
1 import java.sql.SQLException;
2 import java.util.List;
3
4 /**
5  * Nama : St. Hatijah H. Ilyas
6  * Stambuk : 13020180061
7  * Tanggal : 17 Juni 2020
8  * Waktu : 14.19 WITA
9  */
10
11 public interface InterfaceNilaiMhs {
12     mahasiswa insert(mahasiswa siswa) throws SQLException;
13     void update(mahasiswa siswa) throws SQLException;
14     void delete(int id) throws SQLException;
15     List getAll() throws SQLException;
16 }

```

E. Syntax class *ImplNilaiMhs*

```

Source History
9  import java.sql.*;
10 import java.util.*;
11
12 public class ImplNilaiMhs implements InterfaceNilaiMhs {
13
14     @Override
15     public mahasiswa insert(mahasiswa siswa) throws SQLException {
16         PreparedStatement st = KoneksiDb.getConnection().prepareStatement("insert into mhs values (?, ?, ?, ?, ?)");
17         st.setString(1, siswa.getNama());
18         st.setInt(2, siswa.getTgsM());
19         st.setInt(3, siswa.getTgsP());
20         st.setInt(4, siswa.getUts());
21         st.setInt(5, siswa.getUas());
22         st.setInt(6, siswa.getKehadiran());
23         st.executeUpdate();
24         return siswa;
25     }
26
27     @Override
28     public void update(mahasiswa siswa) throws SQLException {
29         PreparedStatement st = KoneksiDb.getConnection().prepareStatement("update mhs set `Tugas Mandiri/Kuis`=?,`"
30             + "`Tugas Proyek`=?,`UTS`=?,`UAS`=?,`Kehadiran`=? where `Nama Mahasiswa`=?");
31
32         st.setString(1, siswa.getNama());
33         st.setInt(2, siswa.getTgsM());
34         st.setInt(3, siswa.getTgsP());
35         st.setInt(4, siswa.getUts());
36         st.setInt(5, siswa.getUas());
37         st.setInt(6, siswa.getKehadiran());
38         st.executeUpdate();
39     }
40
41     @Override
42     public void delete(String nama) throws SQLException {
43         PreparedStatement st = KoneksiDb.getConnection().prepareStatement("delete from mhs where `Nama Mahasiswa`=?");
44         st.setString(1, nama);
45         st.executeUpdate();
46     }
47
48     @Override
49     public List getAll() throws SQLException {
50         Statement st = KoneksiDb.getConnection().createStatement();
51         ResultSet rs = st.executeQuery("select * from mhs");
52         List list = new ArrayList();
53         while (rs.next()) {
54             mahasiswa p = new mahasiswa();
55             p.setNama(rs.getString("nama"));
56             p.setTgsM(rs.getInt("tgsM"));
57             p.setTgsP(rs.getInt("tgsP"));
58             p.setUts(rs.getInt("uts"));
59             p.setUas(rs.getInt("uas"));
60             p.setKehadiran(rs.getInt("kehadiran"));
61             list.add(p);
62         }
63         return list;
64     }
65 }

```

F. Method *setTable()* untuk men - setting isi table yang akan kita buat, membuat header table (nama kolom), variable field untuk menampung nama kolom, lalu akan dijadikan table model.

```

31 private void setTable() {
32     try {
33         record = profileServices.getAll();
34         Object data[][] = new Object[record.size()][6];
35         int x = 0;
36         for (Iterator it = record.iterator(); it.hasNext(); ) {
37             mahasiswa p = (mahasiswa) it.next();
38             data[x][0] = p.getNama();
39             data[x][1] = p.getTgsM();
40             data[x][2] = p.getTgsP();
41             data[x][3] = p.getUts();
42             data[x][4] = p.getUas();
43             data[x][5] = p.getKehadiran();
44             x++;
45         }
46         Object[] field = {"Nama Mahasiswa", "Tugas Mandiri", "Tugas Proyek", "Nilai UTS", "Nilai UAS",
47             "Kehadiran", "Nilai Akhir", "Nilai Huruf"};
48         model = new DefaultTableModel(field, 0);
49         tabel.setModel(model);
50         JTableHeader th = tabel.getTableHeader();
51         th.setReorderingAllowed(false);
52         tabel.setRowHeight(20);
53     } catch (SQLException ex) {
54         Logger.getLogger(ImplNilaiMhs.class.getName()).log(Level.SEVERE, null, ex);
55     }
56 }

```

G. Method *calculate (String... utm)* yang berada di class *NilaiMhs* untuk menghitung semua nilai sesuai dengan persentase nilai yang ada, disini saya membuat program dengan nilai Tugas Mandiri 20%, Tugas Proyek 20%, Nilai UTS 20%, Nilai UAS 25% dan kehadiran 15%. Arti dari **String... utm** pada parameter method adalah berapapun parameter yang dibutuhkan maka sebenarnya program tetap bisa berjalan dengan lancar selama nilai parameter pemanggilnya bertipe string. Tetapi di program ini tidak dibuat seperti itu karena ada code **double t1=0, t2=0, t3=0, t4=0, t5=0**; lalu switch case dan **double jumlah = t1+t2+t3+t4+t5**; yang sudah jelas parameternya hanya bisa menampung 4 nilai. *Math.round* digunakan untuk membulatkan bilangan float(pecahan).

```

Source Design History
60 private String calculate(String... utm){
61     for(int i=0; i<utm.length; i++){
62         if(utm[i].isEmpty()){
63             utm[i] = "0";
64         }
65     }
66     double t1=0, t2=0, t3=0, t4=0, t5=0;
67     for (int i=0; i<utm.length; i++){
68         double cv = Double.valueOf(utm[i]);
69         switch(i){
70             case 0:
71                 t1=cv*0.2;
72                 break;
73             case 1:
74                 t2=cv*0.2;
75                 break;
76             case 2:
77                 t3=cv*0.2;
78                 break;
79             case 3:
80                 t4=cv*0.25;
81                 break;
82             case 4:
83                 t5=cv*0.15;
84                 break;
85             default:
86                 break;
87         }
88     }
89     double jumlah=t1+t2+t3+t4+t5;
90     double as = Math.round(jumlah);
91     String cvJumlah = indexOF(as);
92     return cvJumlah;
93 }

```

H. Method *indexOF (double b)* digunakan untuk mendeteksi bilangan, jika bilangan tersebut bernilai 0 maka hilangkan, selain itu hanya merubah tipe data saja. Method ini digunakan pada method *calculate* saja.

```

95 public String indexOF(double b){
96     String hsl;
97     if(b%1==0){
98         hsl=String.valueOf(b).substring(0, String.valueOf(b).lastIndexOf("."));
99     }else{
100         hsl=String.valueOf(b);
101     }
102     return hsl;
103 }

```


- I. Method *calAbjad* digunakan untuk memberikan nilai berupa huruf yang sudah diprogram pada method ini.

```
Source Design History
105 private String calAbjad(String jumlah) {
106     String nilai="";
107     double b=Double.valueOf(jumlah);
108     if(b>85) {
109         nilai="A";
110     }else if(b<=85 && b>=81) {
111         nilai="A-";
112     }else if(b<=80 && b>=76) {
113         nilai="B+";
114     }else if(b<=75 && b>=71) {
115         nilai="B";
116     }else if(b<=70 && b>=66) {
117         nilai="B-";
118     }else if(b<=65 && b>=61) {
119         nilai="C+";
120     }else if(b<=60 && b>=51) {
121         nilai="C";
122     }else if(b<=50 && b>=45) {
123         nilai="D";
124     }else if(b<45) {
125         nilai="E";
126     }
127     return nilai;
128 }
```

- J. Untuk mengaktifkan *Action Performed* pada tombol button *Refresh* menggunakan code dibawah ini dan menggunakan method *Reset* yang berfungsi untuk mererefresh atau membersihkan isi field (kolom isi) yang sebelumnya kita ketikkan. Menggunakan *setText* karena akan men – setting variable dari pallet text field untuk men – setting menjadi kosong (“”).

```
483 private void RefreshActionPerformed(java.awt.event.ActionEvent evt) {
484     // TODO add your handling code here:
485     reset();
486 }
```

```
579 private void reset() {
580     txtMhs.setText("");
581     txtTM.setText("");
582     txtTP.setText("");
583     txtUTS.setText("");
584     txtUAS.setText("");
585     txtKeh.setText("");
586 }
```

- K. Untuk mengaktifkan *Action Performed* pada tombol button *Hitung dan Simpan* menggunakan code dibawah ini. Fungsinya yaitu untuk menghitung nilai yang telah di input dan dihitung sesuai dengan perhitungan sebelumnya. Setelah itu, inputan yang diketikkan beserta hasil nilai akhir dan grade (nilai berupa huruf) tersebut akan di munculkan ke dalam table yang telah disiapkan. Dalam code tersebut terdapat *JOptionPane* atau *penanganan kesalahan* yaitu dengan diharuskan memasukkan nama terlebih dahulu sebelum angka nilai yang diinputkan akan dihitung dan disimpan.

```
488 private void HitungSimpanActionPerformed(java.awt.event.ActionEvent evt) {  
489     // TODO add your handling code here:  
490     String MS = txtMhs.getText();  
491     String tgs = txtTM.getText();  
492     String tgs1 = txtTP.getText();  
493     String tgs2 = txtUTS.getText();  
494     String tgs3 = txtUAS.getText();  
495     String tgs4 = txtKeh.getText();  
496  
497     if(MS.isEmpty()){  
498         JOptionPane.showMessageDialog(this, "Masukkan Nama Terlebih Dahulu");  
499         txtMhs.requestFocus();  
500     }else{  
501         String[] data= {tgs, tgs1, tgs2, tgs3, tgs4};  
502         String jml = calculate(data);  
503         Object[] data2 = {MS, tgs, tgs1, tgs2, tgs3, tgs4, jml, calAbjad(jml)};  
504         model.addRow(data2);  
505     }  
506     reset();  
507 }
```

- L. Untuk mengaktifkan *Action Performed* pada tombol button *Hapus* menggunakan code dibawah ini. Untuk menghapus baris pada table, bisa langsung semua, bisa satu-satu karena modelnya menggunakan *getSelectedRows*, yaitu mengambil pilihan (select) yang user masukkan. Jadi ketika user memilih (select) 1 baris atau 2 baris itu akan terhapus dari table, jika tidak ada baris yang dipilih maka tidak ada aksi.

```
430 private void HapusActionPerformed(java.awt.event.ActionEvent evt) {  
431     // TODO add your handling code here:  
432     int[] pilih = tabel.getSelectedRows();  
433     if(pilih.length > 0){  
434         for(int i = pilih.length-1; i >=0; i--){  
435             model.removeRow(pilih[i]);  
436         }  
437     }else{  
438         JOptionPane.showMessageDialog(this, "Pilih Data yang ingin Dihapus");  
439     }  
440     reset();  
441 }
```

- M. Untuk mengaktifkan *Action Performed* pada tombol button *Edit* menggunakan code dibawah ini. Berfungsi untuk mengganti/update data yang telah dimasukkan kedalam table sebelumnya.

```

443 private void EditActionPerformed(java.awt.event.ActionEvent evt) {
444     try {
445         // TODO add your handling code here:
446         mahasiswa p = new mahasiswa();
447
448         if (txtMhs.getText().equals("") && txtTM.getText().equals("") && txtTP.getText().equals("") &&
449             txtUTS.getText().equals("") && txtUAS.getText().equals("") && txtKeh.getText().equals("")) {
450             JOptionPane.showMessageDialog(this, "Silahkan Isi Data Yang Akan Diubah");
451         } else {
452             p.setNama(txtMhs.getText());
453             p.setTgsM(Integer.parseInt(txtTM.getText()));
454             p.setTgsP(Integer.parseInt(txtTP.getText()));
455             p.setUts(Integer.parseInt(txtUTS.getText()));
456             p.setUas(Integer.parseInt(txtUAS.getText()));
457             p.setKehadiran(Integer.parseInt(txtKeh.getText()));
458             profileServices.update(p);
459             ambil_data_tabel();
460             JOptionPane.showMessageDialog(this, "Data Berhasil Diubah");
461         }
462     } catch (SQLException ex) {
463         JOptionPane.showMessageDialog(this, "Data Gagal Diubah");
464     }
465 }

```

N. Untuk mengaktifkan *Action Performed* pada tombol button *Keluar* menggunakan code dibawah ini. Berfungsi untuk apakah user ingin keluar atau tidak dari aplikasi tersebut.

```

513 private void KeluarActionPerformed(java.awt.event.ActionEvent evt) {
514     // TODO add your handling code here:
515     int reply= JOptionPane.showConfirmDialog(null, "Yakin Ingin Keluar?", "Konfirmasi Keluar Aplikasi",
516         JOptionPane.YES_NO_OPTION);
517     if(reply==JOptionPane.YES_OPTION){
518         System.exit(0);
519     }
520 }

```

PENGUJIAN PROGRAM

A. Tampilan Awal

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

Tugas Mandiri

Nilai UTS

Kehadiran

Tugas Proyek

Nilai UAS

Hitung dan Simpan

Refresh

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
----------------	---------------	--------------	-----------	-----------	-----------	-------------	-------------

Edit

Hapus

Keluar

B. Tampilan Hitung dan Simpan

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

ulfa

Tugas Mandiri

80

Nilai UTS

78

Kehadiran

65

Tugas Proyek

77

Nilai UAS

65

Hitung dan Simpan

Refresh

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
tjah	87	89	88	88	91	88	A

Edit

Hapus

Keluar

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

Tugas Mandiri

Nilai UTS

Kehadiran

Tugas Proyek

Nilai UAS

Hitung dan Simpan
Refresh

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
tijah	87	89	88	88	91	88	A
ulfa	80	77	78	65	65	73	B

Edit
Hapus
Keluar

Jika user telah menginputkan semua nilai sedangkan Nama Mahasiswa belum terinput maka akan menghasilkan ouput seperti ini

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

Tugas Mandiri

Nilai UTS

Kehadiran

Tugas Proyek

Nilai UAS

i

Masukkan Nama Terlebih Dahulu

OK

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Kehadiran	Nilai Akhir	Nilai Huruf
tijah	87	89		88	A

Edit
Hapus
Keluar

C. Tampilan Edit

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

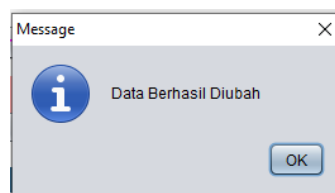
Nama Mahasiswa

Tugas Mandiri Nilai UTS Kehadiran

Tugas Proyek Nilai UAS

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
tjah	87	89	88	88	91	88	A
ulfa	80	77	78	65	65	73	B

Sebelum di edit Nilai UAS = 65, setelah di tekan tombol EDIT maka akan muncul pemberitahuan seperti ini



Dan data telah berubah menjadi Nilai UAS = 77.

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

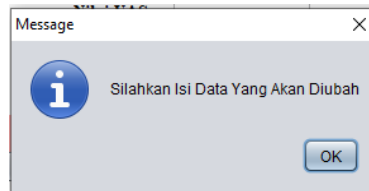
Nama Mahasiswa

Tugas Mandiri Nilai UTS Kehadiran

Tugas Proyek Nilai UAS

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
tjah	87	89	88	88	91	88	A
ulfa	80	77	78	77	65	73	B

Dan akan muncul pemberitahuan seperti ini jika user belum memilih dan meng-edit data yang akan diubah.



D. Tampilan Hapus

MENGHITUNG NILAI MAHASISWA

PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

Tugas Mandiri

Nilai UTS

Kehadiran

Tugas Proyek

Nilai UAS

Hitung dan Simpan

Refresh

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
tjah	87	89	88	88	91	88	A
ulfa	80	77	78	77	65	73	B

Edit

Hapus

Keluar

MENGHITUNG NILAI MAHASISWA

PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

Tugas Mandiri

Nilai UTS

Kehadiran

Tugas Proyek

Nilai UAS

Hitung dan Simpan

Refresh

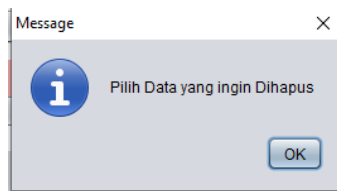
Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Nilai UTS	Nilai UAS	Kehadiran	Nilai Akhir	Nilai Huruf
tjah	87	89	88	88	91	88	A

Edit

Hapus

Keluar

Dan akan muncul pemberitahuan seperti ini jika user belum memilih data atau baris yang akan di hapus.



E. Tampilan Keluar

MENGHITUNG NILAI MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa

Tugas Mandiri

Tugas Proyek

Nilai UTS

Nilai UAS

Kehadiran

?

Yakin Ingin Keluar?

YesNo

Nama Mahasiswa	Tugas Mandiri	Tugas Proyek	Kehadiran	Nilai Akhir	Nilai Huruf
tijah	87	89	91	88	A

Edit

Hapus

Keluar

PENUTUP

A. Kesimpulan

Program Menghitung Nilai Mahasiswa ini sangat diperlukan didalam sebuah universitas, khususnya untuk mahasiswa dalam mengetahui nilai akhir dan grade dengan menghitung nilai dari Tugas Mandiri/Kuis, Tugas Proyek, UTS, UAS, dan kehadiran. Program ini mempunyai Create untuk membuat data yang di inputkan, Read untuk membaca data yang telah di inputkan, Update untuk mengubah data, Delete untuk menghapus data dan Keluar untuk keluar dari program.

B. Saran

Saya berharap semoga program ini dapat digunakan dan berguna bagi fakultas dan mahasiswa. Bila ada kekurangan baiknya dibahas agar bisa berkembang menjadi lebih baik lagi.