

CRUD GUI JAVA SQL & MYSQL

STRUKTUR DATABASE SQL

Nama Database : biodataDB


Table : tbl_bio

	Column Name	Data Type	Allow Nulls
🔑	id_biodata	int	<input type="checkbox"/>
	nama_lengkap	varchar(50)	<input checked="" type="checkbox"/>
	jenis_kelamin	char(1)	<input checked="" type="checkbox"/>
▶	agama	varchar(20)	<input checked="" type="checkbox"/>
	hobi	varchar(50)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	no_telp	varchar(15)	<input checked="" type="checkbox"/>
	alamat	text	<input checked="" type="checkbox"/>
	created_date	datetime	<input checked="" type="checkbox"/>
	updated_date	datetime	<input checked="" type="checkbox"/>
	is_deleted	char(1)	<input type="checkbox"/>
			<input type="checkbox"/>

STRUKTUR DATABASE MYSQL

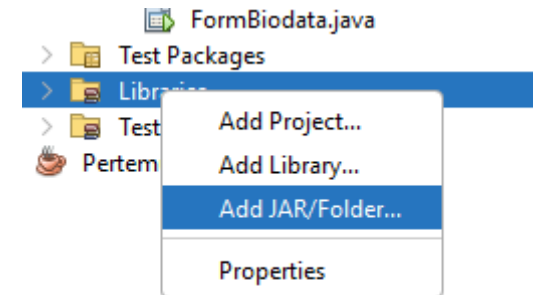
Nama Database : biodataDB

Table : tbl_bio

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/>	1	id_biodata 	int(10)			Tidak	<i>Tidak ada</i>		AUTO_INCREMENT
<input type="checkbox"/>	2	nama_lengkap	varchar(50)	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	3	jenis_kelamin	char(1)	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	4	agama	varchar(20)	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	5	hobi	varchar(50)	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	6	email	varchar(50)	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	7	no_telp	varchar(15)	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	8	alamat	text	utf8mb4_general_ci		Ya	<i>NULL</i>		
<input type="checkbox"/>	9	created_date	datetime			Ya	<i>NULL</i>		
<input type="checkbox"/>	10	updated_date	datetime			Ya	<i>NULL</i>		
<input type="checkbox"/>	11	is_deleted	char(1)	utf8mb4_general_ci		Tidak	1		

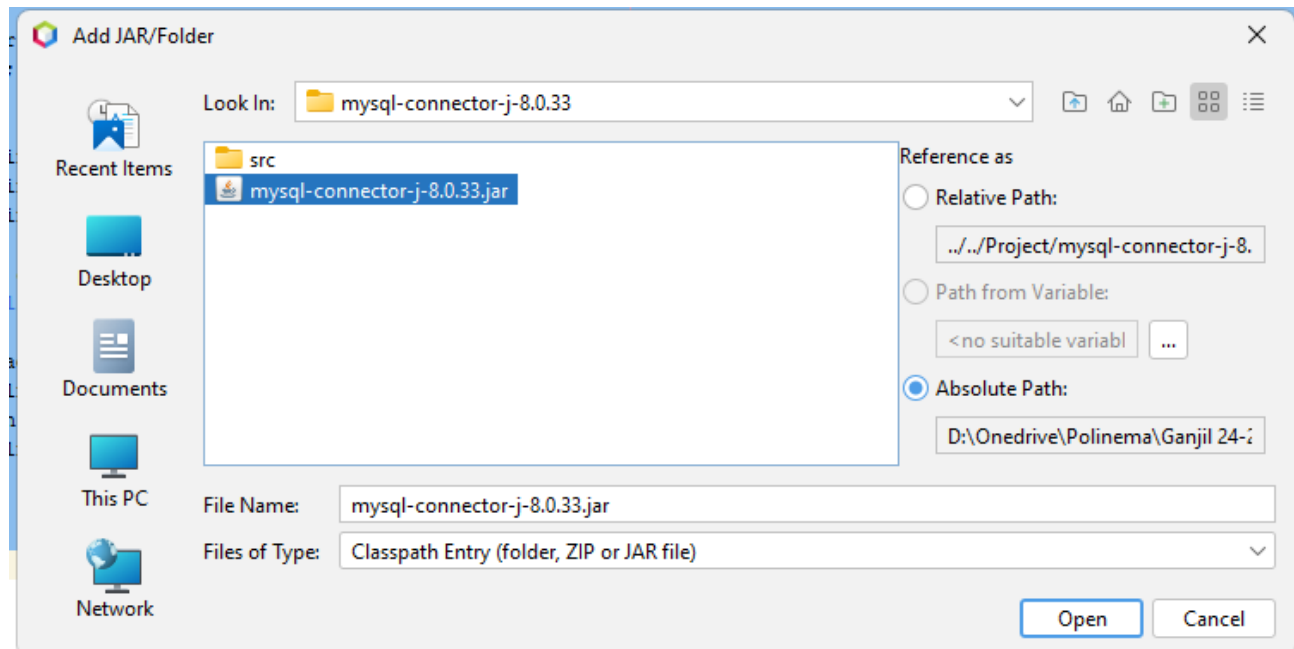
PENAMBAHAN JDK DI PROJECT

1. Klik kanan pada folder “Libraries”, pilih add jar/folder



2. Kemudian pilih file .jar dari mysql atau sql server, klik open

- Mysql : [mysql-connector-j-8.0.33.jar](#)
- SQL : [sqljdbc_12.8](#)



KONEKSI SQL “CLASSKONEKSI”

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4
5 public class ClassKoneksi {
6     private static final String URL = "jdbc:sqlserver://localhost:1433;databaseName=cobaDB;encrypt=true;trustServerCertificate=true;";
7     private static final String USERNAME = "sa";
8     private static final String PASSWORD = "xxxxxxxxx";
9
10    public static Connection getConnection() {
11        Connection conn = null;
12        try {
13            conn = DriverManager.getConnection(URL, USERNAME, PASSWORD);
14            System.out.println("Koneksi ke database berhasil!");
15        } catch (SQLException e) {
16            System.out.println("Koneksi gagal: " + e.getMessage());
17        }
18        return conn;
19    }
20 }
```

KONEKSI MYSQL “CLASSKONEKSI”

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4
5 public class ClassKoneksi {
6     private static final String URL = "jdbc:mysql://localhost:3306/biodatadb";
7     private static final String USERNAME = "root";
8     private static final String PASSWORD = "";
9
10    public static Connection getConnection() {
11        Connection conn = null;
12        try {
13            conn = DriverManager.getConnection(URL, USERNAME, PASSWORD);
14            System.out.println("Koneksi ke database berhasil!");
15        } catch (SQLException e) {
16            System.out.println("Koneksi gagal: " + e.getMessage());
17        }
18        return conn;
19    }
20 }
```

CLASS BIODATA → UNTUK MENYIMPAN VARIABLE FORM KE QUERY

```
1 public class Biodata {
2     private int id_biodata;
3     private String nama_lengkap;
4     private String jenis_kelamin;
5     private String agama;
6     private String hobi;
7     private String email;
8     private String no_telp;
9     private String alamat;
10
11     public void setId_biodata(int id_biodata) {
12         this.id_biodata = id_biodata;
13     }
14
15     public int getId_biodata() {
16         return id_biodata;
17     }
18
19     public String getNama_lengkap() {
20         return nama_lengkap;
21     }
22
23     public void setNama_lengkap(String nama_lengkap) {
24         this.nama_lengkap = nama_lengkap;
25     }
26
27     public String getJenis_kelamin() {
28         return jenis_kelamin;
29     }
30
31     public void setJenis_kelamin(String jenis_kelamin) {
32         this.jenis_kelamin = jenis_kelamin;
33     }
34
35     public String getAgama() {
36         return agama;
37     }
38
39     public void setAgama(String agama) {
40         this.agama = agama;
41     }
42 }
```

```
43     public String getHobi() {
44         return hobi;
45     }
46
47     public void setHobi(String hobi) {
48         this.hobi = hobi;
49     }
50
51     public String getEmail() {
52         return email;
53     }
54
55     public void setEmail(String email) {
56         this.email = email;
57     }
58
59     public String getNo_telp() {
60         return no_telp;
61     }
62
63     public void setNo_telp(String no_telp) {
64         this.no_telp = no_telp;
65     }
66
67     public String getAlamat() {
68         return alamat;
69     }
70
71     public void setAlamat(String alamat) {
72         this.alamat = alamat;
73     }
74 }
```

CLASS DATA ACCESS OBJECT (DAO) – SQL (1)

Script Code : [BiodataDAO.png](#)

```
1 public class BiodataDAO {
2     private final Connection conn;
3     private final String curr_time;
4
5     public BiodataDAO() {
6         conn = ClassKoneksi.getConnection();
7
8         LocalDateTime currentDateTime = LocalDateTime.now(); // Mendapatkan waktu saat ini
9         // Konversi LocalDateTime ke Timestamp
10        Timestamp timestamp = Timestamp.valueOf(currentDateTime);
11
12        this.curr_time = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").format(timestamp);
13    }
14
15    public String insert(Biodata bd) throws SQLException{
16
17        try (conn) {
18            String sql = "INSERT INTO [biodataDB].[dbo].[tbl_bio]\n" +
19                "        ([nama_lengkap]\n" +
20                "        ,[jenis_kelamin]\n" +
21                "        ,[agama]\n" +
22                "        ,[hobi]\n" +
23                "        ,[email]\n" +
24                "        ,[no_telp]\n" +
25                "        ,[alamat]\n" +
26                "        ,[created_date])\n" +
27                "VALUES\n" +
28                "        ('"+bd.getNama_lengkap()+"'\n" +
29                "        , '"+bd.getJenis_kelamin()+"'\n" +
30                "        , '"+bd.getAgama()+"'\n" +
31                "        , '"+bd.getHobi()+"'\n" +
32                "        , '"+bd.getEmail()+"'\n" +
33                "        , '"+bd.getNo_telp()+"'\n" +
34                "        , '"+bd.getAlamat()+"'\n" +
35                "        , '"+this.curr_time+"')";
36
37            PreparedStatement stmt = conn.prepareStatement(sql);
38            System.out.println(sql);
39
40            // Eksekusi query
41            int rowsInserted = stmt.executeUpdate();
42            if (rowsInserted > 0) {
43                return "Data "+bd.getNama_lengkap()+" berhasil ditambahkan!";
44            }else{
45                return "Data "+bd.getNama_lengkap()+" gagal ditambahkan!";
46            }
47        }
48
49    }
50}
```

```
51 public List<ArrayList> readBiodata() throws SQLException {
52     String sql = "SELECT * FROM [biodataDB].[dbo].[tbl_bio] WHERE tbl_bio.is_deleted = '1'";
53     List<ArrayList> list = new ArrayList<>();
54
55     try (Statement stmt = conn.createStatement();
56         ResultSet rs = stmt.executeQuery(sql)) {
57
58         while (rs.next()) {
59             ArrayList bio = new ArrayList();
60
61             bio.add(rs.getInt("id_biodata")); //0
62             bio.add(rs.getString("nama_lengkap")); //1
63             bio.add(rs.getString("jenis_kelamin")); //2
64             bio.add(rs.getString("agama")); //3
65             bio.add(rs.getString("hobi")); //4
66             bio.add(rs.getString("email")); //5
67             bio.add(rs.getString("no_telp")); //6
68             bio.add(rs.getString("alamat")); //7
69             bio.add(rs.getString("created_date")); //8
70             bio.add(rs.getString("updated_date")); //9
71
72             list.add( bio);
73         }
74     }
75     return list;
76 }
77
78 public String updateBiodata(Biodata bd) throws SQLException {
79
80     try(conn){
81         String sql = "UPDATE [biodataDB].[dbo].[tbl_bio]\n" +
82             "        SET [nama_lengkap] = '"+bd.getNama_lengkap()+"'\n" +
83             "        ,[jenis_kelamin] = '"+bd.getJenis_kelamin()+"'\n" +
84             "        ,[agama] = '"+bd.getAgama()+"'\n" +
85             "        ,[hobi] = '"+bd.getHobi()+"'\n" +
86             "        ,[email] = '"+bd.getEmail()+"'\n" +
87             "        ,[no_telp] = '"+bd.getNo_telp()+"'\n" +
88             "        ,[alamat] = '"+bd.getAlamat()+"'\n" +
89             "        ,[updated_date] = '"+this.curr_time+"'\n" +
90             "WHERE [id_biodata] = '"+bd.getId_biodata()";
91
92
93         PreparedStatement stmt = conn.prepareStatement(sql);
94
95         // Eksekusi query
96         int rowsUpdated = stmt.executeUpdate();
97         if (rowsUpdated > 0) {
98             return "Data "+bd.getNama_lengkap()+" berhasil diubah!";
99         }else{
100             return "Data "+bd.getNama_lengkap()+" gagal diubah!";
101         }
102     }
103
104 }
105 }
```


CLASS DATA ACCES OBJECT (DAO) – SQL (2)

Script Code : [BiodataDAO.png](#)

```
105
106     public String deleteBiodata(int id) throws SQLException {
107         try(conn){
108             String sql = "UPDATE [biodataDB].[dbo].[tbl_bio]\n" +
109                         "    SET [is_deleted] = '0'\n" +
110                         "    WHERE [id_biodata] = "+id;
111
112
113             PreparedStatement stmt = conn.prepareStatement(sql);
114
115             // Eksekusi query
116             int rowsUpdated = stmt.executeUpdate();
117             if (rowsUpdated > 0) {
118                 return "Data berhasil dihapus!";
119             }else{
120                 return "Data gagal diubah!";
121             }
122
123         }
124     }
125 }
```

CLASS DATA ACCES OBJECT (DAO) – MYSQL (1)

[BiodataDAOMySQL.png - OneDrive](#)

```
1 import java.sql.*;
2 import java.util.ArrayList;
3 import java.util.List;
4 import java.sql.Timestamp;
5 import java.text.SimpleDateFormat;
6 import java.time.LocalDateTime;
7
8 public class BiodataDAO {
9     private final Connection conn;
10    private final String curr_time;
11
12    public BiodataDAO() {
13        conn = ClassKoneksi.getConnection();
14
15        LocalDateTime currentDateTime = LocalDateTime.now(); // Mendapatkan waktu saat ini
16        // Konversi LocalDateTime ke Timestamp
17        Timestamp timestamp = Timestamp.valueOf(currentDateTime);
18
19        this.curr_time = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").format(timestamp);
20    }
21
22    public String insert(Biodata bd) throws SQLException{
23
24        try (conn) {
25            String sql = "INSERT INTO tbl_bio \n" +
26                "        (`nama_lengkap`\n" +
27                "        ,`jenis_kelamin`\n" +
28                "        ,`agama`\n" +
29                "        ,`hobi`\n" +
30                "        ,`email`\n" +
31                "        ,`no_telp`\n" +
32                "        ,`alamat`\n" +
33                "        ,`created_date`\n" +
34                "        VALUES\n" +
35                "        ('"+bd.getNama_lengkap()+"'\n" +
36                "        , '"+bd.getJenis_kelamin()+"'\n" +
37                "        , '"+bd.getAgama()+"'\n" +
38                "        , '"+bd.getHobi()+"'\n" +
39                "        , '"+bd.getEmail()+"'\n" +
40                "        , '"+bd.getNo_telp()+"'\n" +
41                "        , '"+bd.getAlamat()+"'\n" +
42                "        , '"+this.curr_time+"')";
43
44            PreparedStatement stmt = conn.prepareStatement(sql);
45            System.out.println(sql);
46
47            // Eksekusi query
48            int rowsInserted = stmt.executeUpdate();
49            if (rowsInserted > 0) {
50                return "Data "+bd.getNama_lengkap()+" berhasil ditambahkan!";
51            }else{
52                return "Data "+bd.getNama_lengkap()+" gagal ditambahkan!";
53            }
54        }
55    }
56 }
57
```

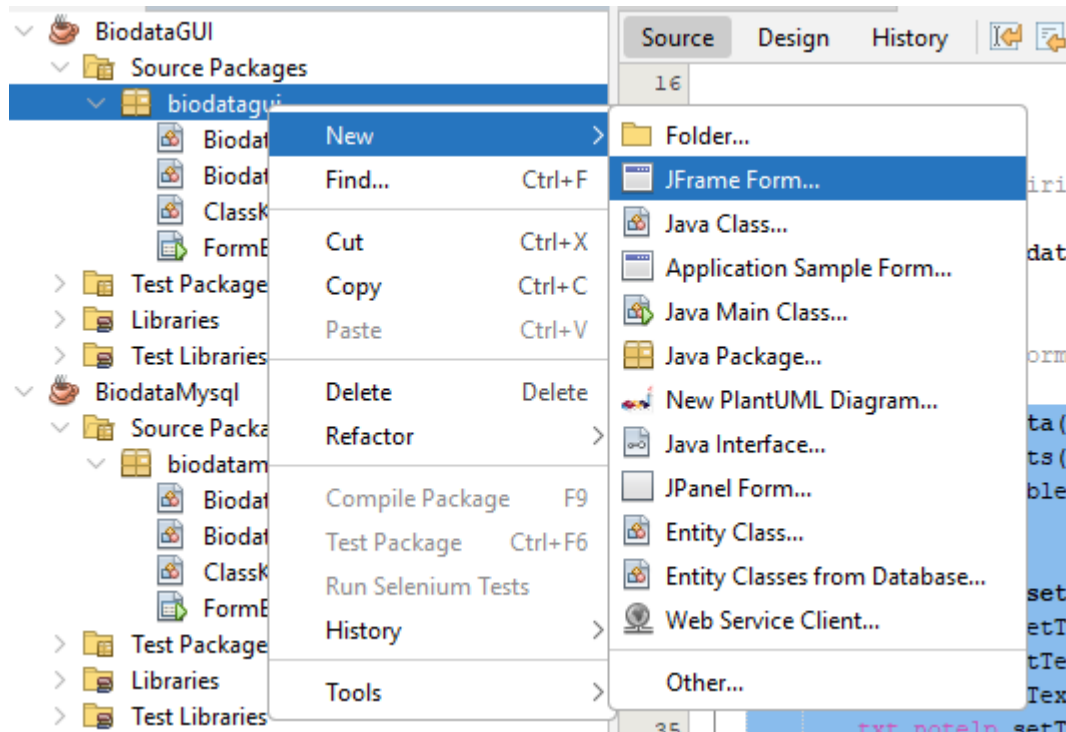
```
57
58 public List<ArrayList> readBiodata() throws SQLException {
59     String sql = "SELECT * FROM `tbl_bio` WHERE `is_deleted` = '1'";
60     List<ArrayList> list = new ArrayList<>();
61
62     try (Statement stmt = conn.createStatement();
63         ResultSet rs = stmt.executeQuery(sql)) {
64
65         while (rs.next()) {
66             ArrayList bio = new ArrayList();
67
68             bio.add(rs.getInt("id_biodata")); //0
69             bio.add(rs.getString("nama_lengkap")); //1
70             bio.add(rs.getString("jenis_kelamin")); //2
71             bio.add(rs.getString("agama")); //3
72             bio.add(rs.getString("hobi")); //4
73             bio.add(rs.getString("email")); //5
74             bio.add(rs.getString("no_telp")); //6
75             bio.add(rs.getString("alamat")); //7
76             bio.add(rs.getString("created_date")); //8
77             bio.add(rs.getString("updated_date")); //9
78
79             list.add( bio);
80         }
81     }
82     return list;
83 }
84
85 public String updateBiodata(Biodata bd) throws SQLException {
86
87     try(conn){
88         String sql = "UPDATE tbl_bio \n" +
89             "        SET `nama_lengkap` = '"+bd.getNama_lengkap()+"'\n" +
90             "        , `jenis_kelamin` = '"+bd.getJenis_kelamin()+"'\n" +
91             "        , `agama` = '"+bd.getAgama()+"'\n" +
92             "        , `hobi` = '"+bd.getHobi()+"'\n" +
93             "        , `email` = '"+bd.getEmail()+"'\n" +
94             "        , `no_telp` = '"+bd.getNo_telp()+"'\n" +
95             "        , `alamat` = '"+bd.getAlamat()+"'\n" +
96             "        , `updated_date` = '"+this.curr_time+"'\n" +
97             "        WHERE `id_biodata` = '"+bd.getId_biodata()+'";
98
99         PreparedStatement stmt = conn.prepareStatement(sql);
100
101         // Eksekusi query
102         int rowsUpdated = stmt.executeUpdate();
103         if (rowsUpdated > 0) {
104             return "Data "+bd.getNama_lengkap()+" berhasil diubah!";
105         }else{
106             return "Data "+bd.getNama_lengkap()+" gagal diubah!";
107         }
108     }
109 }
110
111 }
112
```

CLASS DATA ACCES OBJECT (DAO) – MYSQL (2)

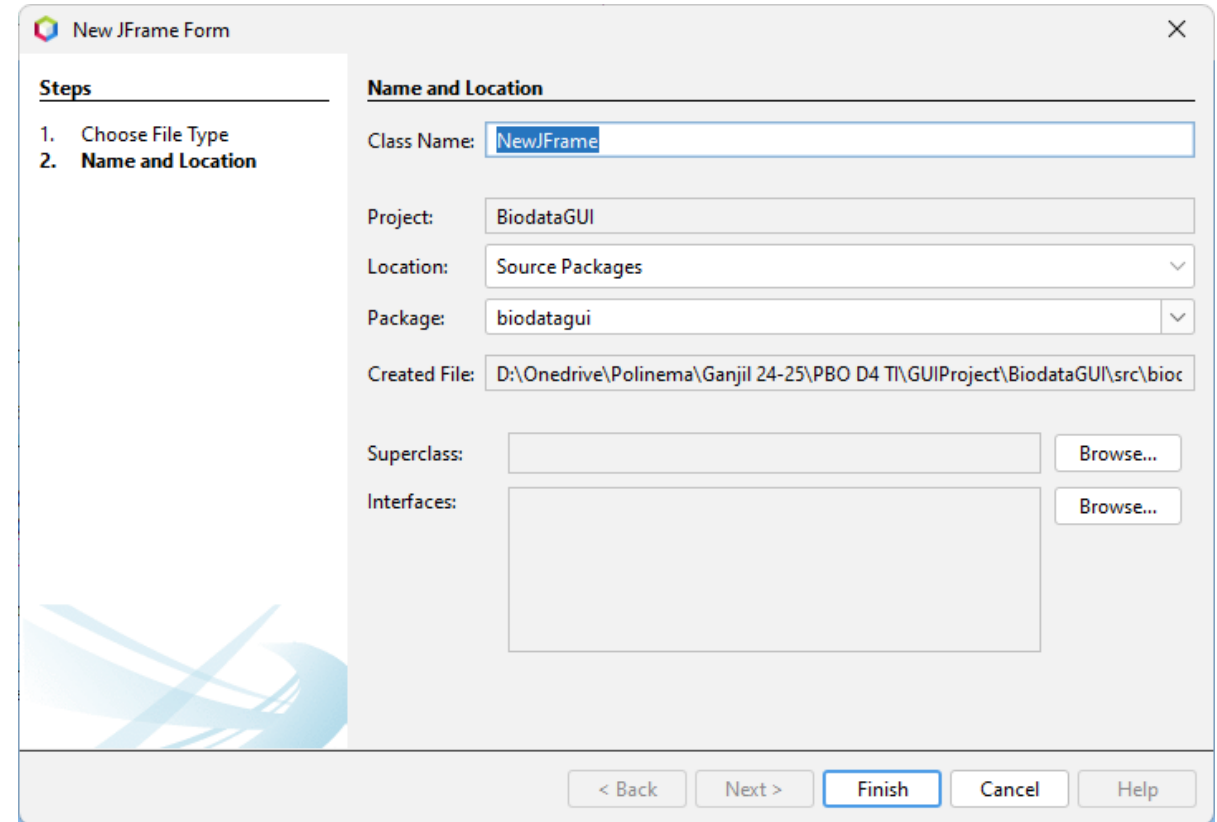
[BiodataDAOMySql.png - OneDrive](#)

```
112
113     public String deleteBiodata(int id) throws SQLException {
114         try(conn){
115             String sql = "UPDATE tbl_bio\n" +
116                 "    SET is_deleted = '0'\n" +
117                 "    WHERE id_biodata = "+id;
118
119
120             PreparedStatement stmt = conn.prepareStatement(sql);
121
122             // Eksekusi query
123             int rowsUpdated = stmt.executeUpdate();
124             if (rowsUpdated > 0) {
125                 return "Data berhasil dihapus!";
126             }else{
127                 return "Data gagal diubah!";
128             }
129
130         }
131     }
132 }
```

MEMBUAT FILE GUI JAVA



1



2

DESAIN GUI

FORM BIODATA

Nama Lengkap :

Jenis Kelamin : ☐ Laki-Laki ☐ Perempuan

Agama :

Hobi : ☐ Makan ☐ Nonton Film ☐ Game
☐ Olahraga ☐ Tidur

Email :

No. Telp

Alamat :

DATA BIODATA

ID	Nama_L...	Jenis_Kel...	Agama	Hobi	Email	No_Telp	Alamat	Created...	Updated...
----	-----------	--------------	-------	------	-------	---------	--------	------------	------------

MENAMBAHKAN FUNGSI RESET DAN READ DATA DI DATABASE

```
1 public FormBiodata() {
2     initComponents();
3     loadDataToTable();
4 }
5
6 private void reset(){
7     txt_alamat.setText("");
8     txt_email.setText("");
9     txt_nama.setText("");
10    txt_notelp.setText("");
11    cmb_agama.setSelectedIndex(0);
12
13    rb_jk_laki.setSelected(false);
14    rb_jk_perempuan.setSelected(false);
15
16    btn_tambah.setEnabled(true);
17    btn_simpan.setEnabled(false);
18    jLabelForm.setText("FORM BIODATA");
19
20    for (java.awt.Component comp : jPanelHobi.getComponents()) {
21        if (comp instanceof JCheckBox checkbox) { // Cek apakah komponen adalah JCheckBox
22            if (checkbox.isSelected()) { // Cek apakah checkbox dipilih
23                checkbox.setSelected(false);
24            }
25        }
26    }
27 }
28
29 private void loadDataToTable() {
30     try {
31         // inisiasi object BiodataDAO
32         BiodataDAO dao = new BiodataDAO();
33         // inisiasi variable untuk menampung data dari database
34         List<ArrayList> biodataList = dao.readBiodata();
35         // inisiasi model dari JTable
36         DefaultTableModel model = (DefaultTableModel) jTableBio.getModel();
37         model.setRowCount(0); // Hapus semua baris
38
39         for (ArrayList biodata : biodataList) {
40             String jk = biodata.get(2).equals("P") ? "Perempuan" : "Laki-Laki";
41             Object[] row = {
42                 biodata.get(0), biodata.get(1), jk,
43                 biodata.get(3), biodata.get(4),
44                 biodata.get(5), biodata.get(6),
45                 biodata.get(7), biodata.get(8),
46                 biodata.get(9)
47             };
48             model.addRow(row);
49         }
50     } catch (SQLException e) {
51         JOptionPane.showMessageDialog(this, "Gagal memuat data: " + e.getMessage());
52     }
53 }
```

MENAMBAHKAN FUNGSI BUTTON TAMBAH

```
1 private void btn_tambahActionPerformed(java.awt.event.ActionEvent evt) {  
2     try {  
3         // ArrayList untuk menampung hobi yang dipilih  
4         ArrayList<String> hobiArr = new ArrayList<>();  
5         // Iterasi semua komponen di dalam panel  
6         for (java.awt.Component comp : jPanelHobi.getComponents()) {  
7             if (comp instanceof JCheckBox checkbox) { // Cek apakah komponen adalah JCheckBox  
8                 if (checkbox.isSelected()) { // Cek apakah checkbox dipilih  
9                     hobiArr.add(checkbox.getText()); // Tambahkan teks checkbox ke array  
10                }  
11            }  
12        }  
13        // join text delimiter ","  
14        String hobi = String.join(",", hobiArr);  
15  
16        // set variable  
17        Biodata bd = new Biodata();  
18        bd.setNama_lengkap(txt_nama.getText());  
19        bd.setJenis_kelamin(rb_jk_laki.isSelected() ? "L" : "P");  
20        bd.setAgama(cmb_agama.getSelectedItem().toString());  
21        bd.setAlamat(txt_alamat.getText());  
22        bd.setEmail(txt_email.getText());  
23        bd.setHobi(hobi);  
24        bd.setNo_telp(txt_notelp.getText());  
25  
26        BiodataDAO dao = new BiodataDAO();  
27        String ins = dao.insert(bd);  
28  
29        JOptionPane.showMessageDialog(rootPane, ins, "Success", HEIGHT);  
30  
31        loadDataToTable();  
32        reset();  
33    } catch (SQLException ex) {  
34        Logger.getLogger(FormBiodata.class.getName()).log(Level.SEVERE, null, ex);  
35    }  
36 }
```

MENAMBAHKAN FUNGSI BUTTON EDIT

```
1 private void jButtonEditActionPerformed(java.awt.event.ActionEvent evt) {  
2     int i = jTableBio.getSelectedRow();  
3  
4     TableModel mdl = jTableBio.getModel();  
5  
6     jLabelForm.setText("Edit Data ID : "+mdl.getValueAt(i, 0).toString());  
7  
8     txt_nama.setText(mdl.getValueAt(i, 1).toString());  
9  
10    if(mdl.getValueAt(i, 2).toString().equals("Perempuan")){  
11        rb_jk_perempuan.setSelected(true);  
12    }else{  
13        rb_jk_laki.setSelected(true);  
14    }  
15  
16    cmb_agama.setSelectedItem(mdl.getValueAt(i, 3).toString());  
17  
18    String[] hobi = mdl.getValueAt(i, 4).toString().split("\\\\,");  
19  
20    for (String hobi1 : hobi) {  
21        for (java.awt.Component comp : jPanelHobi.getComponents()) {  
22            if (comp instanceof JCheckBox checkbox) {  
23                // Cek apakah komponen adalah JCheckBox  
24                if (checkbox.getText().equals(hobi1)) {  
25                    checkbox.setSelected(true);  
26                }  
27            }  
28        }  
29    }  
30  
31    txt_email.setText(mdl.getValueAt(i, 5).toString());  
32    txt_notelp.setText(mdl.getValueAt(i, 6).toString());  
33    txt_alamat.setText(mdl.getValueAt(i, 7).toString());  
34  
35    btn_tambah.setEnabled(false);  
36    btn_simpan.setEnabled(true);  
37 }
```


MENAMBAHKAN FUNGSI BUTTON SIMPAN

```
1 private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {  
2     try {  
3         int i = jTableBio.getSelectedRow();  
4         TableModel mdl = jTableBio.getModel();  
5         // ArrayList untuk menampung hobi yang dipilih  
6         ArrayList<String> hobiArr = new ArrayList<>();  
7         // Iterasi semua komponen di dalam panel  
8         for (java.awt.Component comp : jPanelHobi.getComponents()) {  
9             if (comp instanceof JCheckBox checkbox) { // Cek apakah komponen adalah JCheckBox  
10                if (checkbox.isSelected()) { // Cek apakah checkbox dipilih  
11                    hobiArr.add(checkbox.getText()); // Tambahkan teks checkbox ke array  
12                }  
13            }  
14        }  
15        // join text delimiter ","  
16        String hobi = String.join(",", hobiArr);  
17        // set variable  
18        Biodata bd = new Biodata();  
19        bd.setId_biodata((int) (mdl.getValueAt(i, 0)));  
20        bd.setNama_lengkap(txt_nama.getText());  
21        bd.setJenis_kelamin(rb_jk_laki.isSelected() ? "L" : "P");  
22        bd.setAgama(cmb_agama.getSelectedItem().toString());  
23        bd.setAlamat(txt_alamat.getText());  
24        bd.setEmail(txt_email.getText());  
25        bd.setHobi(hobi);  
26        bd.setNo_telp(txt_notelp.getText());  
27        BiodataDAO dao = new BiodataDAO();  
28        String upd = dao.updateBiodata(bd);  
29        JOptionPane.showMessageDialog(rootPane, upd, "Success", HEIGHT);  
30        reset();  
31        loadDataToTable();  
32    } catch (SQLException ex) {  
33        Logger.getLogger(FormBiodata.class.getName()).log(Level.SEVERE, null, ex);  
34    }  
35 }  
36  
37  
38  
39  
40 }
```

MENAMBAHKAN FUNGSI BUTTON DELETE

```
1 private void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt) {  
2     int response = JOptionPane.showConfirmDialog(null,  
3         "Anda yakin menghapus data ini?",  
4         "Konfirmasi Hapus Data",  
5         JOptionPane.OK_CANCEL_OPTION);  
6  
7     int i = jTableBio.getSelectedRow();  
8  
9     TableModel mdl = jTableBio.getModel();  
10  
11     if (response == JOptionPane.OK_OPTION) {  
12         try {  
13             BiodataDAO dao = new BiodataDAO();  
14             String del = dao.deleteBiodata((int) mdl.getValueAt(i, 0));  
15  
16             JOptionPane.showMessageDialog(rootPane, del, "Success", HEIGHT);  
17             loadDataToTable();  
18         } catch (SQLException ex) {  
19             Logger.getLogger(FormBiodata.class.getName()).log(Level.SEVERE, null, ex);  
20         }  
21     }  
22 }
```

FULL CODE : LINK