

Tugas Proyek

Makassar, 20 Juni 2020

**PEMROGRAMAN BERORIENTASI OBJEK
(MENGHITUNG GAJI BERSIH PEGAWAI CV.XXXI)**



Nama : Siti gayatri hehanussa
Stambuk : 13020180076
Kelas : B2

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2020**

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Puji syukur kehadirat Allah SWT, dengan rahmat dan karunia-Nya, saya memiliki kesempatan untuk menyelesaikan laporan akhir. Dan diberi kesempatan untuk menyelesaikan laporan ini tentang konsep **Menghitung Gaji Bersih Pegawai** sebagai salah satu tugas Proyek Pemrograman Berorientasi Objek.

Tidak lupa saya ucapkan terima kasih kepada bapak/ibu dosen serta berbagai pihak yang telah memberikan dukungan serta memberikan petunjuk dalam menyelesaikan laporan ini. Saya menyadari bahwa dalam penulisan laporan ini masih banyak kekurangan, saya sangat mengharapkan kritik dan saran yang membangun.

Semoga dengan selesainya laporan ini dapat bermanfaat bagi pembaca dan teman-teman.

Makassar, **20 Juni 2020**

Siti Gayatri hehanussa

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dari tahun ke tahun selalu mengalami perkembangan yang sangat pesat. Banyaknya fasilitas kemudahan-kemudahan yang ditimbulkan oleh perkembangan teknologi informasi secara langsung berdampak kepada kegiatan organisasi. Dampak dari perkembangan teknologi informasi yang terjadi memacu organisasi-organisasi untuk tetap serta dapat meningkatkan prestasi yang dijalanannya. Peran teknologi informasi menitikberatkan pada pengaturan sistem informasi, selain itu teknologi informasi dapat memenuhi kebutuhan organisasi dengan sangat cepat, tepat waktu, Zaman sekarang ini persaingan di dunia IT sangatlah ketat, maka kita sebagai salah seorang calon IT harus pintar –pintar berkreasi dalam meningkatkan

Teknologi di dunia ini terutama melalui bahasa pemrograman seperti menciptakan berbagai jenis aplikasi canggih untuk mempermudah kinerja. Semua teknologi yang ada di dunia ini tidak luput dari kerja keras dan kepandaian tangan seorang para ahli teknologi terutama teknologi IT dalam menciptakan karya –karyanya. pemanfaatan teknologi informasi merupakan sarana penunjang/pendorong bagi organisasi dalam mencapai tujuan organisasi. Pemanfaatan teknologi informasi dapat dilakukan secara efektif jika anggota dalam organisasi dapat menggunakan teknologi tersebut dengan baik. Pemanfaatan teknologi yang efektif dapat meningkatkan kinerja

1.2 Maksud dan Tujuan

Adapun maksud dan tujuan pembuatan laporan ini adalah untuk memenuhi salah satu tugas mata kuliah pemrograman berbasis objek (PBO), yang merupakan laporan tugas proyek. Atau tugas yang telah diberikan oleh dosen yang bersangkutan

BAB II

LANDASAN TEORI

2.1 Pemrograman Berorientasi Objek

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu: Programming-language specification, Application-programming interface, Virtual-machine.

Java pada awalnya disebut Oak tapi kemudian diubah menjadi Java. Nama Java diambil dari nama sebuah biji kopi yang berasal dari Jawa (dalam bahasa Inggris, Jawa=Java) kesukaan Gosling. Karena orang-orang di Green Project terbiasa menggunakan UNIX yang berbasis C++ dan bukan Pascal, maka kebanyakan sintaks diambil dari bahasa C serta mengadopsi orientasi objek C++ dan bukan prosedural. Java dibekali dengan kemampuan untuk dapat digunakan pada berbagai platform atau berbagai sistem operasi dan arsitektur komputer.

Pemrograman Java berorientasikan objek, artinya semua aspek yang terdapat dalam Java adalah objek. Hal ini sangat memudahkan pemrograman untuk mendesain, membuat, mengembangkan dan mengalokasikan kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Java bersifat mudah didekompilasi yaitu proses membalikan dari kode menjadi source code, ini karena kode Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama class, metode dan tipe data.

Pada bahasa pemrograman Java ada yang disebut dengan Java API terdiri dari tiga bagian utama yaitu Java Standard Edition (SE), sebuah standar API aplikasi desktop dan applets dengan bahasa dasar yang mendukung grafis, M/K, keamanan, konektivitas basis data dan jaringan. Java Enterprise Edition (EE), sebuah inisiatif API untuk merancang aplikasi server dengan mendukung untuk basis data. Java Micro Edition (ME), sebuah API untuk

merancang aplikasi yang jalan pada alat kecil seperti telepon genggam, komputer genggam dan pager.

Ada juga yang disebut dengan MVC yang merupakan sebuah konsep dalam membangun sebuah aplikasi dengan memisahkan antara data dari tampilan dan aksi pemrosesannya. Model berfungsi sebagai sumber data. View berfungsi sebagai desain interface yang berinteraksi langsung dengan user. Controller berguna sebagai “otak” atau “business logic” yang memproses data yang dilakukan oleh user.

2.2 Objek

Yang dimaksud dengan objek pada java adalah sekumpulan software yang terdiri dari variable dan method-method yang terkait. Objek juga merupakan benda nyata yang di buat berdasarkan rancangan yang di definisikan di dalam class.

Object adalah instance dari class. Jika class secara umum mepresentasikan (template) sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri. Contoh : Dari class Fruit kita dapat membuat object Mangga, Pisang, Apel, dan lain-lain.

2.3 Class

Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variable dan method-methode pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program.

Class merupakan suatu blueprint atau cetakan untuk menciptakan suatu instant dari object. class juga merupakan grup suatu object dengan kemiripan attributes/properties, behaviour dan relasi ke object lain. Contoh : Class Person, Vehicle, Tree, Fruit dan lain-lain.

2.4 Method

Method adalah kumpulan program yang mempunyai nama. Method merupakan sarana bagi programmer untuk memecah program menjadi bagian-bagian yang kecil agar jadi lebih kompleks sehingga dapat di gunakan berulang-ulang.

Method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object. Contoh, pada object pear : terdapat method ambilRasa , kupasKulit dan lainlain

2.5 Pewarisan

Pewarisan adalah salah satu cirri pemograman berorientasi objek, yang menyatakan suatu kelas dapat diturunkan lagi menjadi kelas-kelas baru yang lainnya sehingga dapat membentuk sebuah hiraki. Kelas yang merupakan kelas turunan ini bisa disebut dengan kelas anak (subclass) dan kelas yang menjadi dasar penurunan disebut kelas orang tua (superclass).

- a. Dasar Pewarisan (Superclass dan Subclass) Pemograman java mengizinkan kita untuk mendefinisikan suatu kelas yang bersifat generic. Selanjutnya kelas tersebut diturunkan lagi menjadi kelas baru yang bersifat lebih spesifik. Adapun kelas baru yang hasil turunan disebut subclass. Pada proses penurunan kelas ini kelas turunan akan mewarisi sifat-sifat yang terdapat pada kelas induknya, selanjutnya, kelas turunan tersebut dapat memiliki sifat-sifat spesifik yang sebelumnya tidak dimiliki oleh kelas induk. Sebagai contoh, yakni terdapat kelas binatang, kelas tersebut selanjutnya akan diturunkan lagi menjadi kelas-kelas baru yang lebih spesifik tanpa meninggalkan sifat-sifat dari kelas binatang, menjadi kelas: Herbivora (binatang pemakan tumbuhan) dan Karnivora (binatang pemakan tumbuhan). Kelas turunan yang lain akan diturunkan lagi menjadi kelas turunan yang lainnya seperti kelas herbivora bisa diturunkan lagi menjadi kelas Kambing dan Kerbau, sedangkan untuk kelas Karnivora dapat diturunkan lagi menjadi Anjing, Harimau, Singa dsb.() kelas binatang adalah superclass dari kelas herbivora dan karnivora itu sendiri berperan sebagai subclass. kemudian dari superclass tersebut terdapat subclass nya yakni pada superclass Herbivora dapat menjadi Subclass dari kelas Kambing dan Kerbau. Begitupun pada superclass

Karnivora dapat menjadi subclass dari kelas Harimau dan Kucing. Pada pemograman java menyediakan kata kunci `extends` yang digunakan untuk proses penurunan terhadap suatu kelas. Bentuk umum dari penggunaan kata kunci tersebut adalah sebagai berikut: `Class nama-subclass extends nama-superclass{ // badan kelas }` Ketika kita sedang membentuk sebuah hirarki kelas constructor yang akan dipanggil pertama kali dalam setiap pembentukan objek adalah constructor dari kelas induk paling dasar. Sebagai contoh kita memiliki induk dengan nama A dan diturunkan lagi menjadi kelas C, pada situasi saat ini setiap kita membentuk objek dari kelas C, maka urutan yang dipanggil adalah constructor dari kelas A disusul dengan constructor dari kelas B, baru kemudian constructor dari kelas c. Pada saat kita membentuk atau melakukan instansiasi terhadap suatu objek turunan, maka sebelumnya java juga akan membentuk objek dari kelas induk-induk nya.

- b. Pengertian Polimorfisme Polimorfisme adalah kemampuan suatu objek untuk mengungkap banyak hal melalui satu cara yang sama. Polimorfisme merupakan salah satu hal esensial dalam pemograman berorientasi objek karena alasan beriku: yakni polimorfisme mengizinkan kelas induk untuk mendefinisikan sebuah method general (bersifat umum) untuk semua kelas turunannya. Polimorfisme dapat berupa overloading atau overriding. Overloading merupakan bentuk polimorfisme yaitu beberapa metode yang dapat memiliki nama yang sama dengan isi dan parameter yang berbeda didalam sebuah kelas. Biasanya eksekusi program akan langsung mengacu pada metode yang dipanggil sesuai dengan parameter
- c. Overriding merupakan bentuk polimorfisme yaitu berupa metode pada kelas orang tua yang dapat di tulis ulang pada kode kelas anak dalam pewarisan (inheritance) dengan memiliki nama yang sama dan memiliki isi ataupun parameter yang sama atau berbeda.

2.6 NetBeans

Java Netbeans atau Netbeans adalah sebuah proyek software OpenSource. Proyek Netbeans mulai diprakarsai oleh Perusahaan Sun Microsystems sejak bulan Juni 2000 dan terus berkembang hingga saat ini. NetBeans mengacu pada dua hal, yaitu NetBeans platform untuk pengembangan aplikasi desktop java dan sebuah NetBeans Integrated Development Environment (Lutfiana Rahmawati: 2012). Netbeans merupakan sebuah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu debugger.

2.7 Xampp

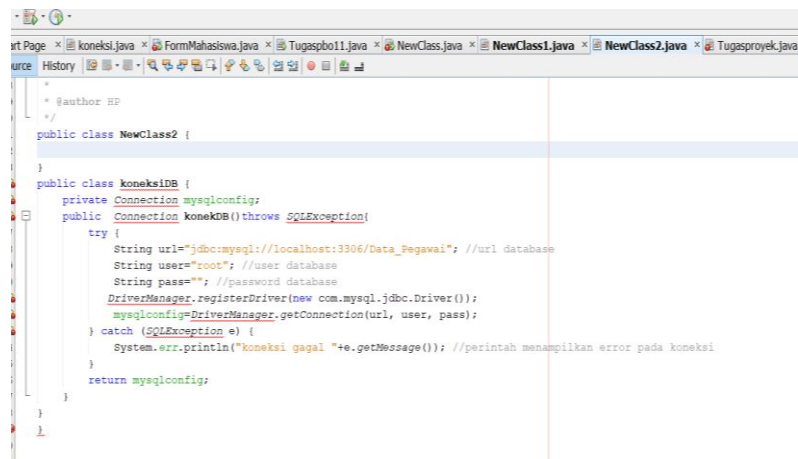
Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, Mysql, phpmyadmin, PHP, Perl, Freetype2, dan lain-lain. Instalasi Mysql dan Apache langsung dapat dilakukan melalui Xampp tanpa menginstall aplikasi aplikasi tersebut satu persatu. Setelah berhasil menginstall Xampp, kita dapat langsung mengaktifkan Mysql dengan cara mengaktifkan Xampp.

BAB IV

IMPLENTASI & PEMBAHASAN

4.1 Implementasi

dalam membuat frame, kita harus membuat kelas terlebih dahulu, dan dalam percobaan ini dibuat class connection untuk meyambungkan ke database MYSQL.



```

* @author HP
*/
public class NewClass2 {

}

public class koneksiIDB {
    private Connection mysqlconfig;
    public Connection koneksiIDB() throws SQLException {
        try {
            String url="jdbc:mysql://localhost:3306/Data_Pegawai"; //url database
            String user="root"; //user database
            String pass=""; //password database
            DriverManager.registerDriver(new com.mysql.jdbc.Driver());
            mysqlconfig=DriverManager.getConnection(url, user, pass);
        } catch (SQLException e) {
            System.err.println("koneksi gagal "+e.getMessage()); //perintah menampilkan error pada koneksi
        }
        return mysqlconfig;
    }
}

```

Sintaks pada from / new class java



```

String nama;
int nip,gol,gapok,lembur,jamker,t1,galem,geber,tun,pjk,ul,GP,tg;
float x=0.15f, y=0.1f;
/**
 * Creates new form formcrud
 */
public formcrud() {
    initComponents();
    kosong();
    load_table();

    add(gol1);
    add(gol2);
    add(gol3);
    add(gol4);
    add(gol5);
}

public void load_data(){
    nama = tnama.getText();
    nip = Integer.parseInt(tnip.getText());
}

private void load_table() {

```

```

/ membuat tampilan model tabel
DefaultTableModel model = new DefaultTableModel();
odel.addColumn("No");
odel.addColumn("NIP");
odel.addColumn("NAMA");
odel.addColumn("GOLONGAN");
odel.addColumn("GAJI POKOK");
odel.addColumn("LEMBUR");
odel.addColumn("JAM KERJA");
odel.addColumn("JAM LEMBUR");
odel.addColumn("GAJI PEGAWAI");
odel.addColumn("GAJI LEMBUR");
odel.addColumn("TOTAL GAJI PEGAWAI");
odel.addColumn("GAJI BERSIH");

/ menampilkan data database kedalam tabel
try {
    int no=1;
    String sql = "select * from pegawai";
    java.sql.Connection conn=(Connection)connection.konekDB();

```

```

        model.addRow(new Object[] {no++,res.getString(1),res.getString(2),res.getString(3),res.getString(4),
        res.getString(5), res.getString(6),res.getString(7), res.getString(8), res.getString(9),
        res.getString(10), res.getString(11),});
    }
    tabel.setModel(model);
} catch (Exception e) {
}

private void kosong() {
    tnip.setText(null);
    tnama.setText(null);
    tgapok.setText(null);
    tlembur.setText(null);
    tjaker.setText(null);
    tjalem.setText(null);
    tgapeg.setText(null);
    tgalem.setText(null);
    tttotal.setText(null);
    tgaber.setText(null);
}

```

```

/
@SuppressWarnings("unchecked")
}
Generated Code
}
private void tnipActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void tnamaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void bgkeluarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
}
private void bgprosesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String sql ="UPDATE karyawan SET nama = '"+nip+"',nama = '"+nama+"', gol = '"+gol+"', gapok = '"+gapok+"', ul = '"+tjaker+"',
        jamker = '"+jamker+"', lembur = '"+lembur+"', GaPe = '"+GP+"', galem = '"+galem+"',
        total = '"+tg+"', gaber = '"+gaber+"' WHERE nip = '"+nip+"'";
        java.sql.Connection conn=(Connection)connection.konekDB();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "data berhasil di edit");
    } catch (Exception e) {

```

```

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Perubahan Data Gagal"+e.getMessage());
    }
    load_table();
    kosong();
}

private void bgsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    load_data();
    try {
        String sql = "INSERT INTO karyawan VALUES ('"+nip+"','"+nama+"','"+gol+"','"+gapok+"','"+t1+"','"+jamker+"','"+lembur+
            "','"+GP+"','"+galem+"','"+tg+"','"+gaber+"')";
        java.sql.Connection conn=(Connection)connection.konekDB();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Penyimpanan Data Berhasil");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    load_table();
    kosong();
}

private void gol1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tgapok.setText("500000");
    tlembur.setText("5000");
    gol = 1;
}

}

private void gol2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tgapok.setText("300000");
    tlembur.setText("3000");
    gol = 2;
}

private void gol3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tgapok.setText("250000");
    tlembur.setText("2000");
    gol = 3;
}

private void gol4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tgapok.setText("100000");
    tlembur.setText("1500");
    gol = 4;
}

private void gol5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tgapok.setText("50000");
    tlembur.setText("1000");
    gol = 5;
}

}

private void hitung1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void hitung2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    gapok = Integer.parseInt(tgapok.getText());
    jamker = Integer.parseInt(tjamker.getText());
    GP = gapok*jamker;
    lembur = Integer.parseInt(tjalem.getText());
    t1 = Integer.parseInt(tlembur.getText());
    galem = lembur*t1;
    tg = GP+galem;
    tun = (int) (tg*x);
    pjK = (int) ((tg*y)+tun);
    gaber = (tg + tun) - pjK;
    tgapeg.setText(""+GP);
    tgalem.setText(""+galem);
    tttotal.setText(""+tg);
    tgaber.setText(""+gaber);
}

private void tabelAncestorAdded(javax.swing.event.AncestorEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */

```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new formcrud().setVisible(true);
        }
    });
}

```

```

// Variables declaration - do not modify
private javax.swing.JTextField bgkeluar;
private javax.swing.JTextField bgproses;
private javax.swing.JTextField bgsimpan;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.ButtonGroup buttonGroup2;
private javax.swing.ButtonGroup buttonGroup3;
private javax.swing.ButtonGroup buttonGroup4;
private javax.swing.JRadioButton gol1;
private javax.swing.JRadioButton gol2;
private javax.swing.JRadioButton gol3;
private javax.swing.JRadioButton gol4;
private javax.swing.JRadioButton gol5;
private javax.swing.JTextField hitung1;
private javax.swing.JTextField hitung2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField jTextField1;

```

```

private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
private javax.swing.JTable tabel;
private javax.swing.JTextField tgaber;
private javax.swing.JTextField tgalem;
private javax.swing.JTextField tgapeg;
private javax.swing.JTextField tgapok;
private javax.swing.JTextField tjaker;
private javax.swing.JTextField tjalem;
private javax.swing.JTextField tlembur;
private javax.swing.JTextField tnama;
private javax.swing.JTextField tnip;
private javax.swing.JTextField tttotal;
// End of variables declaration

```

Tampilan akhir

The screenshot shows a Java Swing application window titled "formcrud". The window contains a form with various input fields and buttons. The fields are: NIP (13020180076), Nama pegawai (Gayatri), Golongan (radio buttons 1, 2, 3, 4, 5), Gaji pokok (5000), Lembur (50), Jam kerja (5), Jam lembur (2), Gaji lembur (5000), Gaji pegawai (10000), Total gaji, and gaji bersih. The buttons are: hitung, hitung1, Proses, Simpai, and keluar. The form is styled with a light blue border and a white background.

4.2 Pembahasan

Bagian ini membahas tiap-tiap hasil implementasi untuk setiap fitur / fungsi atau komponen. Dan membahas tentang penjelasan materi yang dingkst tentang PBO

BAB V

KESIMPULAN & SARAN

5.1 Kesimpulan

Pada praktikum Pemrograman Berorientasi Objek ini saya menggunakan Netbeans 8.2 sebagai editor. Dari hasil pembelajaran, saya bisa memahami dan mengerti cara merancang tampilan form dari sebuah aplikasi dan mengerti fungsi dari source code masing-masing form dan menghubungkannya ke database.

5.2 Saran

Semoga teori dan praktikum Pemrograman Berorientasi Objek dapat ditingkatkan lagi. Untuk kedepannya dengan adanya tugas proyek ini bisa membantu untuk menjadi lebih baik mudah dalam menghadapi hal hal yang bersangkutan dengan java

Daftar Pustaka

<https://www.codepolitan.com/aplikasi-crud-java-sederhana-menggunakan-database-mysql>

<https://student-activity.binus.ac.id/himsisfo/2016/07/pengertian-methode-class-dan-objek-dalam-oop/>