

# ASPEK BERORIENTASI OBJEK DALAM RANCANG BANGUN SISTEM INFORMASI MANAJEMEN ZAKAT

Fatchurrochman

Jurusan Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Malang. Email: [fatchur\\_uin@yahoo.com](mailto:fatchur_uin@yahoo.com)

**Abstrak** – Dalam sistem berorientasi objek, aspek yang sangat penting adalah sifat reusable dari perangkat lunak yang telah dibangun. sifat reusable ini memungkinkan sebuah aplikasi menggunakan kembali bagian dari program yang sebelumnya telah dibangun sehingga proses pengembangan perangkat lunak menjadi lebih cepat karena tidak mengulang pekerjaan yang sama berkali-kali. Zakat adalah salah satu rukun Islam dimana seorang muslim wajib menjalankannya. manajemen zakat perlu dilaksanakan sebagai bagian dari akuntabilitas publik sehingga kepercayaan masyarakat semakin meningkat terhadap proses penyelenggaraan zakat yang dilaksanakan oleh lembaga amil zakat. Sistem informasi manajemen zakat adalah aplikasi yang digunakan untuk mencatat berbagai data dan transaksi sehingga semua data tersebut dapat diproses menjadi informasi dengan cepat dan akurat. Penelitian ini merupakan langkah awal dalam melakukan eksplorasi aspek berorientasi objek dalam rancang bangun sistem informasi. Penelitian ini menghasilkan sistem informasi manajemen zakat yang dirancang dan diimplementasi menggunakan paradigma berorientasi objek. Beberapa konsep utama dalam pemrograman berorientasi objek seperti enkapsulasi, inheritance dan polimorfisme digunakan dalam sistem informasi manajemen zakat ini.

**Kata Kunci** : Pemrograman berorientasi objek, Sistem Informasi Manajemen Zakat, Sistem Berorientasi Objek, Enkapsulasi, Inheritance, Polimorfisme.

## 1. PENDAHULUAN

Kualitas perangkat lunak dapat dilihat dari dua hal yaitu proses rekayasa dan kepuasan pemakai terhadap produk perangkat lunak. Dari sisi proses rekayasa, aspek kualitas yang penting diperhatikan adalah kebenaran, ketegaran, perluasan dan guna ulang. Dari sisi pemakai, kualitas yang baik adalah bila semua fungsi yang diharapkan pengguna dapat dipenuhi, kemudahan untuk dipelajari, kemudahan pengingatan, kemudahan penggunaan, efisiensi dan kehandalan. Rekayasa perangkat lunak mempunyai tugas untuk mewujudkan kedua hal tersebut dengan metode-metode pengembangan perangkat lunak seperti waterfall, prototyping, pengembangan formal, pengembangan komponen guna ulang dan lain-lain. Selain

itu juga diperlukan paradigma, yaitu cara pandang spesifik dalam pencarian solusi.

Salah satu paradigma dalam rekayasa perangkat lunak adalah paradigma berorientasi objek. Paradigma ini dapat diterapkan pada semua tahap pengembangan perangkat lunak. Identy adalah membangun aplikasi menggunakan komponen-komponen yang telah tersedia, baik yang dibuat sendiri atau komersial.

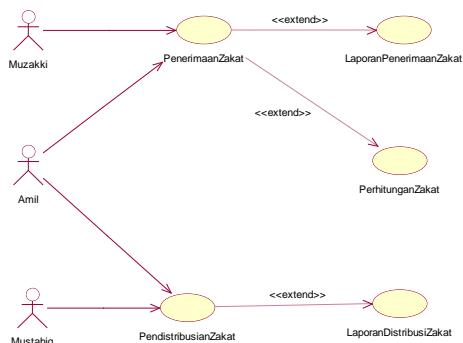
Penelitian ini merupakan eksplorasi terhadap proses pengembangan perangkat lunak dengan paradigma berorientasi objek terhadap persoalan sistem informasi zakat. Analisis dan perancangan dilakukan menggunakan kaskas rasional rose dan implementasinya menggunakan bahasa pemrograman java. Diharapkan bahwa akan ditemukan komponen-komponen yang dapat digunakan ulang dapat

pengembangan perangkat lunak lebih lanjut.

## 2. USE CASE DIAGRAM

Sistem informasi manajemen zakat adalah aplikasi yang digunakan untuk mencatat berbagai data dan transaksi sehingga semua data tersebut dapat diproses menjadi informasi dengan cepat dan akurat.

Sistem dimulai ketika ada seorang muzakki, yaitu orang yang akan mengeluarkan zakat, mendatangi amil zakat untuk membayar zakatnya. Bila muzakki tadi telah mengetahui bera zakat yang harus dikeluarkan maka amil akan segera menerima uang zakat dan memasukkan data pembayaran ke dalam sistem. Tetapi bila muzakki belum mengetahui berapa zakat yang harus dikeluarkan maka amil zakat dapat membantunya untuk menghitung besarnya zakat dengan memasukkan berbagai informasi jumlah kekayaan yang dimiliki muzakki ke dalam sistem dan sistem akan menghitung besarnya zakat yang harus dikeluarkan. Setelah mengetahui besarnya zakat barulah muzakki tadi membayar zakat. Pada waktu yang ditentukan, amil zakat akan mendistribusikan zakat yang telah terkumpul kepada yang berhak menerimanya yaitu mustahiq. Hasil akhir dari sistem ini adalah adanya laporan bulanan penerimaan zakat dan distribusinya.

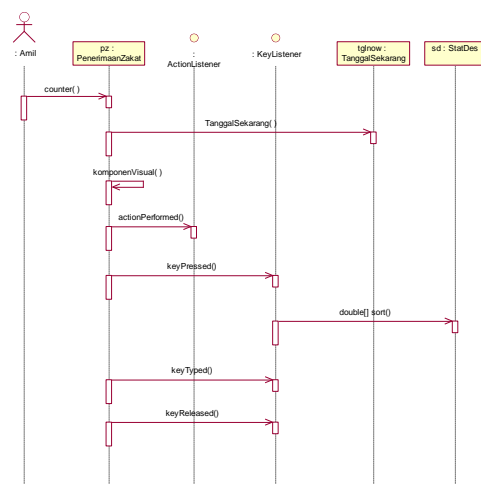


Gambar 1. Use Case Diagram SIM Zakat

Use case pada gambar 1 adalah fitur dasar dari sistem yang akan dibangun. Sistem terdiri dari tiga aktor yaitu Muzakki, Amil dan Mustahiq. Use case penerimaan zakat dan pendistribusian zakat digunakan untuk menangani transaksi sementara use case laporan dan perhitungan baru akan dilaksanakan bila memang diperlukan.

## 3. SEQUENCE DIAGRAM

Sequence diagram dibuat minimal satu buah untuk setiap use case. Bila ada perkecualian maka satu use case dapat memiliki lebih dari satu sequence diagram.

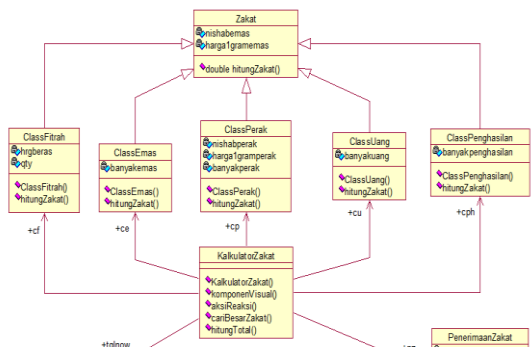


Gambar 2. Sequence Diagram untuk Use Case Penerimaan Zakat

Sequence diagram pada gambar 2 adalah sequence diagram untuk use case penerimaan zakat tanpa perkecualian. Dengan adanya sequence diagram maka method yang diperlukan dalam suatu class dapat ditemukan.

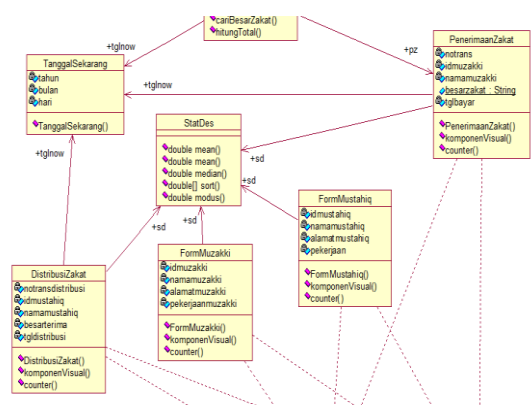
## 4. DIAGRAM CLASS

Diagram class adalah diagram yang penting bagi pengembang perangkat lunak karena dapat dijadikan dasar dalam membangun program. Dari hasil analisis yang dilakukan maka diagram class untuk sistem informasi manajemen zakat terdapat 13 class dan dua interface.



Gambar 3. Aspek inheritance dalam tahap perancangan sistem

Gambar 3 adalah diagram class yang menunjukkan konsep inheritance, dimana terdapat class Zakat sebagai class induk dan class tiap jenis zakat sebagai class anak. Class zakat berisi informasi yang bersifat umum yang dimiliki hampir oleh semua class anak seperti nishab emas dan harga 1 gram emas.



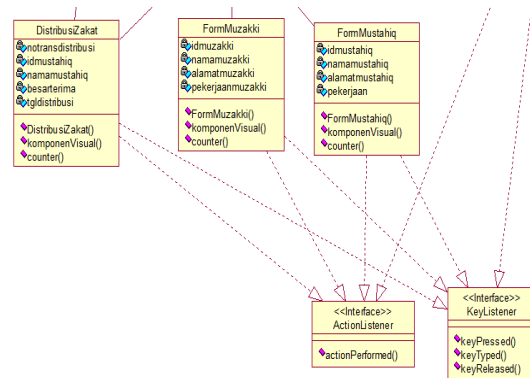
Gambar 4. Aspek Reusable

Gambar 4 menunjukkan interaksi antar class dalam menangani proses transaksi zakat. Class PenerimaanZakat digunakan untuk transaksi penerimaan zakat dan class DistribusiZakat digunakan untuk menangani pendistribusi zakat.

Class StatDes adalah contoh penggunaan class yang bersifat reusable. Class ini sebenarnya adalah class yang digunakan untuk berbagai keperluan pengolahan data yang berhubungan dengan statistik deskriptif. Class ini telah dibuat sebelum ide penelitian ini dikerjakan. Ternyata pada saat penelitian ini dirancang ada kebutuhan untuk menentukan nilai

maksimal dari sebuah array satu dimensi sehingga method sort yang ada pada class StatDes ini bisa digunakan dalam aplikasi zakat ini.

Class TanggalSekarang juga merupakan komponen yang bersifat reusable karena class ini dapat digunakan dimanapun ketika terdapat keperluan untuk menampilkan tanggal dari sistem komputer.



Gambar 5. Penggunaan Interface

Interface adalah fasilitas dalam pemrograman java yang digunakan agar sebuah class seolah-olah memiliki banyak class induk. Gambar 5 menunjukkan bahwa dalam sistem yang dibangun terdapat dua interface yaitu interface ActionListener untuk menangani penekanan tombol mouse dan KeyListener untuk menangani penekanan tombol keyboard.

## 5. IMPLEMENTASI ENKAPSULASI

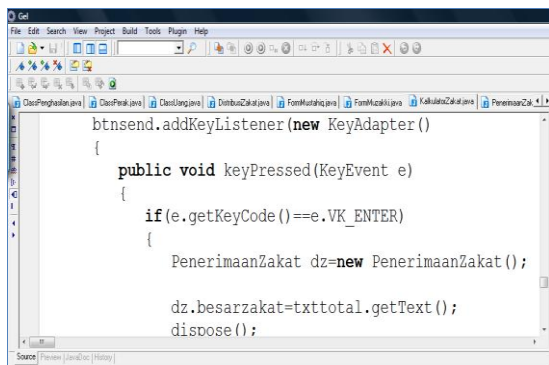
Enkapsulasi dapat dinyatakan dengan penggunaan access modifier variabel dan method, overloading dan package.

```
private String notrans;
private String idmuzakki;
private String namamuzakki;
public static String besarzakat;
private String tglbayar;
```

Penggunaan access modifier private ini menyebabkan variabel tidak dapat diakses dari luar class itu sendiri sehingga dapat melindungi nilai dari variabel tersebut dari

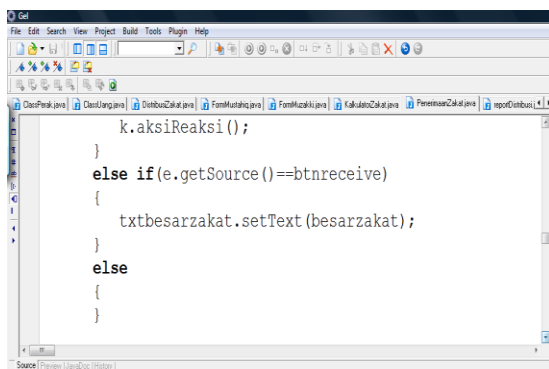
perubahan secara sengaja maupun tidak sengaja karena adanya proses perhitungan. Khusus untuk variabel *besarzakat* dinyatakan dengan *public static* karena ada kebutuhan untuk mengakses variabel tersebut dari luar class. Kebutuhan itu adalah pengiriman nilai dari proses perhitungan zakat ke aplikasi penerimaan zakat.

Gambar 6 adalah kode yang digunakan untuk mengirimkan nilai dari class KalkulatorZakat ke class PenerimaanZakat. Pengiriman nilai dilakukan dengan membentuk objek dari class PenerimaanZakat di class KalkulatorZakat, lalu dengan aksi penekanan tombol send dilakukan pemberian nilai pada variabel *besarzakat* yang ada di class PenerimaanZakat.



Gambar 6. Kode untuk pengiriman pesan dari KalkulatorZakat ke class Penerimaan Zakat

Gambar 7 adalah cara yang digunakan untuk menangkap atau mengambil nilai dari variabel *besarzakat* yang telah dikirimkan oleh class KalkulatorZakat.



Gambar 7. Kode untuk mengambil nilai besarzakat

## 6. IMPLEMENTASI INHERITANCE

Implementasi inheritance diwujudkan melalui class Zakat. Class ini merupakan class abstract yang berisi variabel dan method *hitungZakat()* yang akan diimplementasikan oleh class anak.

```

abstract class Zakat
{
    protected double nishabemas;
    protected double harga1gramemas;
    abstract public double hitungZakat();
}
    
```

Setiap jenis zakat dijadikan class anak sehingga class-class ini juga memiliki variabel dan method dari class Zakat meskipun tidak melakukan deklarasi.

```

public class ClassEmas extends Zakat
{
    private double banyakemas;

    ClassEmas(double nsb,double
    hrg,double jml)
    {
        nishabemas=nsb;
        harga1gramemas=hrg;
        banyakemas=jml;
    }
    public double hitungZakat()
    {
        double zakatemas=0.0;
        if(banyakemas>=nishabemas)
        {
            double
            jumlah=banyakemas*harga1gramemas;
            zakatemas=2.5/100*jumlah;
        }
        else
        {
            zakatemas=0;
        }
        return zakatemas;
    }
}
    
```

## 7. IMPLEMENTASI POLIMORFISME

Polimorfisme adalah salah satu karakteristik dalam pemrograman berorientasi objek dimana sebuah method dapat memiliki perilaku yang berbeda bergantung pada jenis objeknya. Dalam penelitian ini polimorfisme diimplementasikan dalam method `cariBesarZakat` yang memiliki argumen class induknya.

```
public double cariBesarZakat(Zakat z)
{
    return z.hitungZakat();
}
```

Method tersebut akan dijalankan di class `KalkulatorZakat` dengan cara membentuk objek dan menjadikan objek tersebut Sebagai argumen.

```
txtqtyemas.addListener(new
KeyListener()
{
    public void keyPressed(KeyEvent e)
    {
        if(e.getKeyCode()==e.VK_ENTER)
        {
            double
nishab=Double.parseDouble(txtnishabema
s.getText());
            double
hrngemas=Double.parseDouble(txtharga1gr
amemas.getText());
            double
qty=Double.parseDouble(txtqtyemas.getT
ext());

            ClassEmas ce=new
ClassEmas(nishab,hrngemas,qty);

            txtemas.setText(String.valueOf(cariBesa
rZakat(ce));
            txtqtyperak.requestFocus();
            hitungTotal();
        }
    }
});
```

Dengan cara sama, semua jenis zakat diimplementasikan dengan polimorfisme ini.

## 8. HASIL DAN UJI COBA

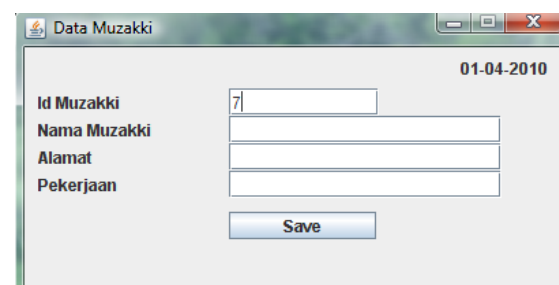
Hasil penelitian yang dilakukan adalah berupa rancangan sistem berorientasi objek dan beberapa aplikasi yang membentuk sistem informasi manajemen zakat. Rancangan sistem telah diuraikan di bagian sebelumnya dalam paper ini. Hasil implementasi dan uji coba ditampilkan dalam uraian berikut.

### 1. Hasil Impelementasi

Aplikasi ini dimulai dengan menjalankan menu utama yang menyediakan beberapa fasilitas seperti data muzakki, data mustahiq, transaksi penerimaan, transaksi distribusi, laporan data muzakki, laporan penerimaan zakat tiap bulan dan laporan distribusi tiap bulan (Gambar 8- 15).



Gambar 8. Menu Utama SIM Zakat



Gambar 9. Aplikasi Data Muzakki

Aplikasi data muzakki digunakan untuk mencatat data muzakki, untuk memasukkan data baru, sistem secara otomatis akan menampilkan id muzakki



baru berdasarkan no terakhir di database ditambah satu. Hal yang sama juga diberlakukan untuk data mustahiq.

Gambar 10. Aplikasi Data Mustahiq

Gambar 11. Aplikasi Penerimaan Zakat

Aplikasi Penerimaan zakat digunakan untuk mencatat data orang yang membayar zakat. No transaksi selalu bertambah dari data terbesar di database, sehingga no transaksi ini akan selalu unik untuk setiap transaksi.

Bila muzakki sudah tahu berapa besar zakat yang harus dibayar, maka amil dapat segera memasukkan data ke textfield besar zakat, tetapi bila belum tahu maka amil dapat membantu muzakki menghitung besarnya zakat yang harus dikeluarkan dengan aplikasi perhitungan zakat, baru hasilnya dikirimkan ke textfield besar zakat.

Gambar 12. Aplikasi Distribusi Zakat

Gambar 13. Aplikasi Perhitungan Zakat

No	Id Muzakki	Nama Muzakki	Alamat Muzakki
1	2	Fatchurrochman	Graha Sejahtera Residence G-3
2	3	Jono	Malang
3	4	Silvani Handayani	Graha Sejahtera
4	5	Fajar Rashid Arrochman	Graha Sejahtera
5	6	M.Faisal	Graha Sejahtera F-6

Gambar 14. Laporan Data Muzakki

No	Id Muzakki	Nama Muzakki	Besar Zakat	Tgl Bayar
1	5	Fajar Rashid Arrochman	1000000	26-03-2010
2	5	Fajar Rashid Arrochman	1000000	30-03-2010
3	2	Fatchurrochman	3725000.0	30-03-2010
4	6	M.Faisal	1000000	30-03-2010
5	5	Fajar Rashid Arrochman	3625000.0	30-03-2010
6	3	Jono	1125000.0	31-03-2010

Gambar 15. Laporan Penerimaan Zakat

## 2. Uji Coba Ketepatan Counter

Counter adalah method yang digunakan untuk menentukan nomor transaksi pada class penerimaan dan distribusi, serta menentukan nomor ID dari muzakki dan mustahiq. Counter ini memanfaatkan sifat reusable dari class StatDes.

NoTransaksi	IdMuzakki	BesarZakat	TglBayar
1	2	1000	25-3-2010
2	2	500000	25-3-2010
4	2	10000	26-3-2010
7	2	300000	26-3-2010
9	2	800000	26-3-2010
15	2	3725000.0	30-03-2010
3	3	5600000.0	25-3-2010
5	3	70000	26-3-2010
10	3	900000	26-3-2010
11	3	50000	26-3-2010
18	3	1125000.0	31-03-2010
6	4	1000000	26-3-2010
8	4	500000	26-3-2010
12	4	300000	26-3-2010
13	5	1000000	26-03-2010
14	5	100000	30-03-2010
17	5	3625000.0	30-03-2010
16	6	1000000	30-03-2010

Gambar 16. Data TransaksiPenerimaan Zakat

Dari gambar 16 diketahui bahwa no transaksi tertinggi adalah 18 sehingga bila ada transaksi baru no transaksinya harus 19. Bila aplikasi penerimaan zakat dijalankan maka hasilnya tampak pada gambar 17 yang menunjukkan bahwa no transaksi adalah 19.

Gambar 17. Uji coba counter

## 3. Uji Coba Kebenaran Perhitungan Zakat

Uji coba perhitungan zakat dilakukan untuk mengetahui apakah proses perhitungan yang dilakukan oleh sistem telah sesuai dengan kaidah yang benar. Perhitungan secara zakat secara manual seperti tampak pada gambar 18.

Jenis Zakat	Jml orang	harga 1 kg	2.5*10*10000	
Zakat Fitrah	10	10000	250000	
Jenis Zakat	nishab	harga 1 gram	Jml emas	2.5/100*100*350000
emas	91.92	350000	100	875000
Jenis Zakat	nishab	harga 1 gram	Jml perak	2.5/100*650*100000
perak	642	100000	650	1625000
Jenis Zakat	nishab	harga 1 gram	Jml penghasilan	2.5/100*3000000*12
emas	91.92	350000	3000000	900000
Jenis Zakat	nishab	harga 1 gram	Jml uang	2.5/100*40000000
emas	91.92	350000	40000000	1000000

Gambar 18. Perhitungan Zakat secara manual

## 9. KESIMPULAN DAN SARAN

Berdasarkan penelitian dan uji coba dapat disimpulkan sebagai berikut :

Beberapa aspek dalam pemrograman berorientasi objek seperti enkapsulasi, inheritance dan polimorfisme telah diimplementasikan dalam sistem yang dibangun. Sifat reusable dari komponen ditunjukkan pada class Statdes dan class TanggalSekarang. Pada saat kompilasi dan pada saat di running semuanya berjalan dengan baik tanpa ada kesalahan atau peringatan.

Penelitian ini merupakan eksplorasi awal terhadap proses rekayasa perangkat lunak pengembangan sistem informasi. Penelitian selanjutnya diharapkan dapat menemukan berbagai komponen yang bersifat reusable sehingga proses pengembangan perangkat lunak sistem informasi menjadi semakin cepat dan mudah untuk dikembangkan.

Penelitian selanjutnya juga dapat dikembangkan untuk *user requirement* yang lebih detil, misalnya pembuatan laporan untuk tiap jenis zakat sebagai bahan evaluasi bagi badan amil zakat dalam meningkatkan kinerja di masa depan.

## DAFTAR PUSTAKA

- [1]. Hariyanto, Bambang.2004. *Rekayasa Sistem Berorientasi Objek*. Informatika. Bandung.
- [2]. Fatchurrochman. 2008. *Pemrograman Berorientasi Objek menggunakan Bahasa Java*. UIN Press. Malang.

- [3]. Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak menggunakan UML dan Java*. Penerbit Andi. Yogyakarta.
- [4]. Buchori, Abdusshomad., KH. 2010. *Zakat, Sebuah potensi yang terlupakan*. BAZ Jatim.
- [5]. Said, Amir Al-Zibari., Dr. 2006. *124 Tanya Jawab Masalah Zakat*. Akbar Media Eka Sarana. Jakarta.
- [6]. Dwi, Didik Prasetyo. 2007. *150 Rahasia Pemrograman Java*. Elex Media Komputindo. Jakarta.
- [7]. Fatchurrochman. 2009. Ilustrasi Materi Kuliah Sistem Berorientasi Objek. [fatchurrochman.wordpress.com](http://fatchurrochman.wordpress.com)