

IMPLEMENTASI KONSEP PEMROGRAMAN BERORIENTASI OBJEK PADA APLIKASI SISTEM PARKIR MENGGUNAKAN BAHASA PEMROGRAMAN JAVA

Nadifah Adya Ilham¹, Naziro²

^{1,2}Jurusan Teknik Elektro FT Universitas Negeri Malang,
Email: nadifaadya@gmail.com

ABSTRACT

The development of Information Technology has been increasingly rapid. Many agencies or companies that have used information systems for operational processes that manually become an automated and integrated system. One of which is a parking system. In Indonesia there are still many vehicle day care stations that use a parking system manually. This makes the parking attendant ignorance with the price set and the lengths of time parked the vehicle and make it difficult in the process of calculating the report of parked vehicles. For this reason, an integrated parking system is needed so that it can be applied to manual vehicle storage. So in making the parking system uses the concept of programming. The concept of the program to be used is the concept of Object Oriented Programming with the Java programming language and the editor uses Netbeans 8.2 to run the program. The method in conducting this research is to use 2 methods, namely the method for collecting data with the description method and the method for developing a waterfall system. With the parking system will provide convenience for parking users and also make efficient parking attendant performance. So expect speed, and accuracy in collecting vehicle parking at a vehicle storage in Indonesia.

Keywords: java, object oriented programming, parking

ABSTRAK

Perkembangan Teknologi Informasi sudah semakin pesat. Banyak instansi atau perusahaan yang telah mengubah sistem informasi dari proses operasional secara manual menjadi sistem yang otomatis dan terintegrasi. Satu diantaranya ialah sistem parkir. Di Indonesia masih banyak tempat penitipan kendaraan yang menggunakan sistem parkir secara manual. Hal ini membuat ketidaktahuan petugas parkir mengenai harga tarif yang telah ditetapkan, lamanya waktu terparkir kendaraan tersebut, dan menyulitkan dalam proses laporan perhitungan kendaraan yang terparkir. Untuk itu diperlukan sebuah sistem parkir terintegrasi yang dapat diterapkan pada tempat penitipan kendaraan yang masih manual. Pembuatan sistem parkir terintegrasi menggunakan konsep pemrograman. Konsep program yang digunakan adalah konsep Pemrograman Berorientasi Objek dengan bahasa pemrograman Java dan editor menggunakan Netbeans 8.2. Metode penelitian ini yaitu menggunakan 2 metode, yakni metode untuk pengumpulan data dengan metode deskripsi dan metode untuk pengembangan sistem dengan waterfall. Adanya sistem parkir yang berbasis aplikasi desktop dapat memberikan kenyamanan bagi pengguna parkir dan membuat kinerja petugas parkir lebih efisien. Sehingga diharapkan adanya kecepatan, dan keakuratan dalam pendataan parkir kendaraan pada suatu tempat penitipan kendaraan di Indonesia.

Kata kunci: Java, pemrograman berorientasi objek, parkir

PENDAHULUAN

Perkembangan sebuah teknologi merupakan suatu hal yang sangat dinikmati manfaatnya oleh masyarakat luas. Karena teknologi telah mempengaruhi manusia di segala aspek kehidupannya.

Perkembangan Teknologi dan Informasi atau ICT (*Information and Communication Technology*) sudah maju secara signifikan dan

selalu dikembangkan supaya menghasilkan manfaat dengan dampak positif bagi masyarakat.

Salah satu contoh penerapan yang paling sering dijumpai dalam kehidupan sehari-hari adalah penerapan program komputer dalam mengatur perparkiran di mall, gedung, hotel, dan banyak tempat lainnya. Hampir seluruh tempat seperti mall, gedung, hotel yang ada di kota-kota besar yang ada di Indonesia

menggunakan penerapan program komputer untuk mengatur proses perparkiran. Program komputer tersebut disebut dengan aplikasi parkir (Wahyudiantoro, 2013:21).

Parkir adalah keadaan tidak bergerak suatu kendaraan yang bersifat sementara ditinggalkan oleh pengemudinya. Secara hukum dilarang parkir di tengah jalan raya, namun parkir di sisi jalan umumnya diperbolehkan. Fasilitas parkir dibangun bersama-sama dengan kebanyakan gedung untuk memfasilitasi kendaraan pemakai gedung (Utomo, 2013:12).

Program aplikasi sistem parkir merupakan sebuah aplikasi perangkat lunak yang digunakan untuk mengelola transaksi parkir baik di pintu masuk maupun pintu keluar. Jumlah transaksi atau jumlah kendaraan yang keluar masuk area parkir tidak sedikit sehingga memungkinkan terjadinya kesalahan transaksi yang akan merugikan pengelola parkir dengan nilai rupiah yang tidak sedikit (Isnanto, 2013:13).

Sistem parkir merupakan sebuah sistem perangkat lunak yang berfungsi untuk mempermudah dalam pendataan parkir kendaraan di sebuah tempat. Seperti sebuah tempat perbelanjaan atau tempat yang memiliki parkir khusus.

Banyak sekali tempat umum di Indonesia masih menggunakan cara manual untuk pembayaran parkir. Seperti pada swalayan atau market lokal di kota-kota tertentu, contohnya adalah Sardo Swalayan di Malang, Roxy Square di Banyuwangi, dan Persada Swalayan di Malang. Cara manual yang digunakan yaitu hanya dengan sistem karcis yang diberikan kepada pemilik kendaraan saat masuk ke tempat penitipan kendaraan. Belum lagi jika parkir dipenuhi oleh pengunjung maka keamanan dan keefektifan sistem karcis ini pastinya berkurang dan membuat karyawan atau pun petugas parkir tidak dapat bekerja dengan baik dan maksimal.

Oleh karena itu, sistem parkir terotomatisasi ini sangat diperlukannya untuk membantu pekerjaan karyawan dalam mendapatkan data yang akurat tentang parkir.

Tujuan perancangan sistem parkir ini adalah untuk mengimplementasikan konsep Pemrograman Berorientasi Objek pada sebuah sistem parkir dengan bahasa pemrograman java menggunakan editor NetBeans 8.2. Untuk mempermudah pekerjaan karyawan atau petugas parkir dalam penginputan data parkir dan perhitungan biaya parkir.

METODE

1. Metode Pengumpulan Data

Metode penelitian ini menggunakan metode deskripsi. Tujuan dari penelitian deskriptif ini adalah untuk mengumpulkan fakta-fakta aktual yang terjadi pada saat ini. Menurut Sugiyono (2005: 21) menyatakan bahwa metode deskriptif adalah suatu metode yang digunakan untuk menggambarkan atau menganalisis suatu hasil penelitian tetapi tidak digunakan untuk membuat kesimpulan yang lebih luas.

Pada sistem parkir banyak memuat fakta yang ada di lingkungan kita. Saat ini masih banyak tempat umum yang menyediakan tempat parkir atau tempat penitipan kendaraan yang masih manual yaitu dengan karcis. Dengan fakta-fakta yang tersebut, dapat disimpulkan bahwa sistem pendataan parkir yang terotomatisasi dengan komputer ini masih sangat minim diterapkan .

2. Metode Pengembangan Sistem

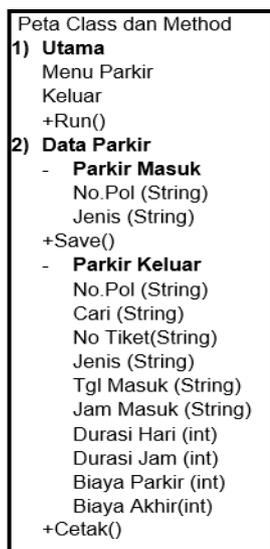
Pengembangan sistem metodologi yang digunakan adalah *System Development Life Cycle* (SDLC) yang sering disebut dengan model pengembangan air terjun (waterfall). Metode waterfall merupakan model pengembangan sistem informasi yang sistematis dan sekuensial (Sasmito, 2017).

Metode penelitian yang digunakan untuk pengembangan perangkat lunaknya yaitu menggunakan model air terjun (waterfall). Model ini merupakan pendekatan perangkat lunak secara terurut yang dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (Rosa dan Shalahudin, 2013).



Gambar 1. Kerangka Penelitian

Analisis Kebutuhan yakni untuk menganalisis kebutuhan apa saja yang nantinya akan dibutuhkan pada saat pembuatan program, misalnya *hardware* dan *software* yang akan digunakan. Pada kasus ini menggunakan bahasa pemrograman java dengan Netbeans 8.2. Untuk desain *database* dan aplikasi, yakni memetakan apa saja kolom dan atribut yang digunakan dalam aplikasi serta menganalisis class dan method dalam aplikasinya. Berikut adalah gambaran singkat tentang desain aplikasi yang digunakan.



Gambar 2. Desain Aplikasi

Tahap pengkodean atau implementasi aplikasi menggunakan penerapan dari OOP atau *Objected Oriented Programming*. Tahap terakhir adalah tahap pengujian aplikasi, dimana aplikasi tersebut digunakan dan dicek apakah bisa berjalan atau tidak, ada error atau tidak. Sehingga keluaran dari program tersebut dapat bermanfaat. Model pengujian pada penelitian ini yaitu menggunakan angket dalam skala likert yang dilakukan pada 10 orang

mahasiswa. Skor yang digunakan dalam rentang 0 sampai 4 yang ditunjukkan pada Tabel 1, serta aspek penilaian yang dievaluasi ditunjukkan pada Tabel 2.

Tabel 1. Skala Likert

Kriteria	Skor
Sangat Baik	5
Baik	4
Cukup	3
Kurang	2
Sangat Kurang	1

Tabel 2. Aspek Penilaian

No	Aspek Penilaian
1	Operasional Sistem Aplikasi
2	User Interface Aplikasi

Tabel 3. Kriteria Interpretasi Skor Berdasarkan Interval Skala Likert

Kriteria	Keterangan
Angka 0% – 19,99%	Tidak setuju/buruk Tidak
Angka 20% – 39,99%	Setuju/Kurang Baik
Angka 40% – 59,99%	Cukup/Netral
Angka 60% – 79,99%	Setuju/Baik
Angka 80% – 100%	Sangat Setuju/Sangat Baik

HASIL DAN PEMBAHASAN

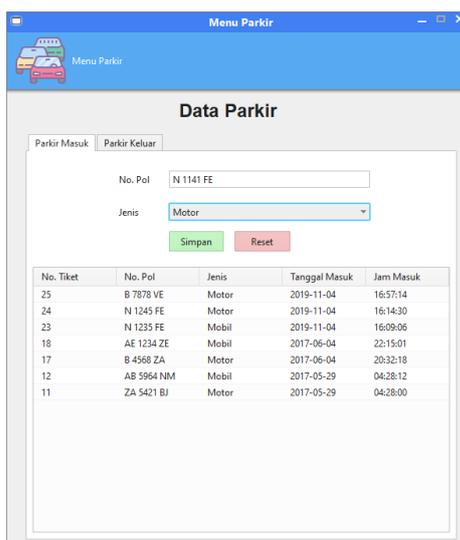
1. Jalannya Program dan Penerapan GUI

Pada tahapan pertama pengguna akan dihadapkan pada menu utama, pada menu utama tersebut terdapat menu keluar dan menu parkir untuk masuk kedalam inti dari program aplikasi parkir.



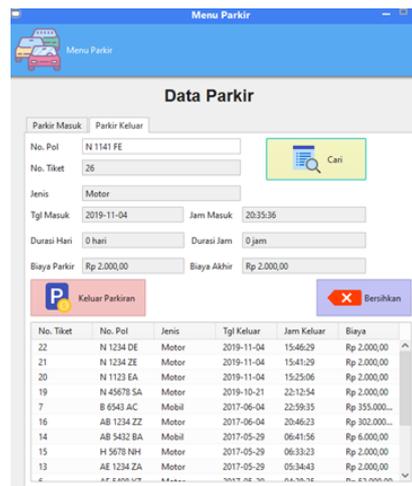
Gambar 3. Tampilan Utama Aplikasi

Lalu selanjutnya, dari tampilan utama mengarah pada tampilan inti setelah tombol Parkir diklik. Sehingga muncul tampilan seperti berikut.



Gambar 4. Tampilan Menu Parkir Masuk

Pada tampilan ini *user* menginputkan data berupa nomor plat dan jenis kendaraan, motor atau pun mobil. Lalu selanjutnya klik simpan untuk menyimpan data pada databasenya. Tombol reset berfungsi untuk meniadakan data yang sewaktu-waktu salah menginputkan. User bisa langsung menekan reset dan menginput kembali data yang benar. Maka pada tabel di bawah, otomatis melakukan penambahan data beserta nomor tiketnya. Sehingga data otomatis tersimpan.



Gambar 5. Tampilan Menu Parkir Keluar

Ketika kendaraan sudah ingin keluar dan melakukan pembayaran. Maka petugas sebagai *user* langsung bisa menginputkan nomor plat dari kendaraan tersebut, lalu secara otomatis dengan mengklik *button* cari akan keluar data-data yang diperlukan termasuk total biaya parkirnya. Sehingga dapat langsung menghitung total bayar secara efisien sesuai dengan data yang tepat dan jelas. Lalu klik keluar ketika proses pembayan telah selesai, karena menandakan bahwa kendaraan tersebut sudah selesai parkir di tempat tersebut. Data langsung bisa terupdate pada *database*. Tombol Bersihkan disediakan untuk memudahkan *user* menginputkan datanya kembali saat kendaraan selanjutnya ingin melakukan proses transaksi begitu seterusnya.

2. Penerapan OOP

- Penerapan Tipe Data

Ada beberapa tipe data yang digunakan pada program ini, contohnya adalah String yang digunakan untuk memasukkan karakter seperti no polisi dan jenis kendaraanya dan Integer untuk menginisialisasi objek biaya parkir, hari, dan waktu. Tipe data yang digunakan didalam program ini ditunjukkan pada source code :

```
new String [] {
    "No. Tiket", "No. Pol", "Jenis", "Tgl Keluar", "Jam Keluar", "Biaya"
}
int hari = Integer.parseInt(konek.rs.getString("durasi_hari"));
int jam = Integer.parseInt(konek.rs.getString("durasi_jam"));
int biaya, biaya_akhir;
```

Gambar 6. Penerapan Tipe Data pada Aplikasi

- Penerapan Kelas dan Objek

OOP tidak bisa terlepas dari kelas dan objek yang digunakan. Tentunya hal ini sebagai langkah untuk membuat suatu sistem berbasis aplikasi yang bisa terintegrasi dan terhubung antara fitur satu dan fitur lainnya. Berikut adalah penerapan Kelas terdapat pada *source code*

```
public class koneksi {  
    public class TA_parkir {
```

Gambar 7. Penerapan Kelas pada Aplikasi

Penerapan Objek terdapat pada *source code*:

```
DecimalFormatSymbols formatRp = new DecimalFormatSymbols();  
new Object [] {  
},  
new String [] {  
    "No. Tiket", "No. Pol", "Jenis", "Tgl Keluar", "Jam Keluar", "Biaya"  
}
```

Gambar 8. Penerapan Objek pada Aplikasi

- Penerapan Enkapsulasi

Salah satu penerapan enkapsulasi pada program ini digunakan untuk memberikan hak akses pada beberapa tombol, sehingga tidak terdapat 2 fungsi dalam satu tombol. Hal ini sangat dibutuhkan untuk membangun sebuah aplikasi yang memiliki banyak fitur di setiap halamannya, seperti aplikasi parkir ini. Penerapan enkapsulasi terdapat pada *source code*:

```
// Variables declaration - do not modify  
private javax.swing.JButton btnCari;  
private javax.swing.JButton btnClear;  
private javax.swing.JButton btnKeluar;  
private javax.swing.JButton btnReset;  
private javax.swing.JButton btnSimpan;  
private javax.swing.JComboBox<String> cbJenis;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel10;  
private javax.swing.JLabel jLabel11;  
private javax.swing.JLabel jLabel12;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JMenuBar jMenuBar1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;
```

Gambar 9. Penerapan Enkapsulasi pada Aplikasi

- Penerapan *Inheritance*

Class yang mengandung member yang sama dari beberapa *class* lain dinamakan *superclass* atau *parent class*. *Class* yang mewarisi dinamakan *subclass* atau *child class*. *Inheritance* menghasilkan *class hierarchy* yang memudahkan untuk menghubungkan halaman tampilan GUI dengan kelas-kelas fitur yang berisi metode-metode setiap tampilan pada aplikasi tersebut terhubung. Pada program ini penerapan *Inheritance* terdapat pada *source code*:

```
public class fParkir extends javax.swing.JFrame {  
    public class awal extends javax.swing.JFrame {
```

Gambar 10. Penerapan *Inheritance* pada Aplikasi

- Penerapan *Polymorphism*

Polymorphism artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphism*, bila objek-objek itu mempunyai antar muka yang identik namun mempunyai perilaku yang berbeda. Hal ini sangat berguna untuk membuat suatu aplikasi lebih efisien, karena tidak perlu membuat objek baru lebih banyak untuk menyesuaikan dengan sebuah *method*. Pada program ini penerapan *Polymorphism* terdapat pada *source code*:

```
public void mousePressed(java.awt.event.MouseEvent evt) {  
    mnParkirMousePressed(evt);  
}  
};  
public void mousePressed(java.awt.event.MouseEvent evt) {  
    mnKeluarMousePressed(evt);  
}  
};
```

Gambar 10. Penerapan *Polymorphism* pada Aplikasi

- Penerapan *Interface Class*

Interface pemrogram, untuk berbagi konstanta atau menentukan bentuk metode yang dapat digunakan oleh sejumlah kelas. *Interface* diimplementasikan ke dalam suatu *class* dengan menggunakan kata kunci *implements*. Sebuah kelas dapat mengimplementasikan lebih dari satu *interface*. *Interface* berfungsi untuk mendeklarasikan terlebih dahulu *method-method* yang ingin dibuat. Jika ingin memakai

method tersebut di kelas lain, hanya perlu melakukan pemanggilan dan *implements*. Pada program ini penerapan *Interface* terdapat pada *source code*:

```
package ta_parkir;

public interface tampil {
    public void tampil_masuk();
    public void keluar();
}

public class fParkir extends javax.swing.JFrame implements tampil {
    DefaultTableModel masuk, keluar;
```

Gambar 11. Penerapan *Interface* pada Aplikasi

- Penerapan *Method Mutator*

Method Mutator adalah *Method* yang digunakan untuk akses objek lain agar bisa mengubah data. *Method* ini biasanya sering digunakan ketika membuat sebuah program pendataan, karena banyak inputan yang sama dan sedikit yang berbeda. Seperti pada aplikasi ini yang menginputkan nomor polisi dan jenis kendaraan. Pada umumnya penulisan *method* ini biasanya diawali dengan penulisan *set*.

```
public void resetAll() {
    txtNoPol.setText("");
    cbJenis.setSelectedIndex(0);
}
```

Gambar 11. Penerapan *Method* pada Aplikasi

- Penerapan *Exception Handling*

Exception Handling adalah *event* yang terjadi ketika program menemui kesalahan pada saat instruksi program dijalankan. Pada program ini mendeteksi apabila terjadi kesalahan ketika menginputkan dan menyimpan data pada *database*.

```
public void konek(String server, String db, String user, String passwd) {
    System.out.println("Keterangan");
    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        System.out.println("Driver JDBC tidak berhasil Load");
        return;
    }
}
```

Gambar 10. Penerapan *Exception Handling* pada Aplikasi

3. Hasil Pengujian Aplikasi

Pada penelitian ini, dilakukan uji coba pengguna kepada 10 responden yakni pada mahasiswa Jurusan Teknik Informatika di Universitas Negeri Malang. Data hasil uji coba

diperoleh dengan cara menyebarkan angket berupa form pertanyaan beserta skala penilaiannya. Hasil uji coba Pengguna disajikan pada Tabel 4 dan Tabel 5.

Tabel 4. Hasil Uji Coba Pengguna

Indikator Penilaian	Hasil Responden				
	1	2	3	4	5
Tampil Awal	0	0	0	0	10
Fitur Parkir	0	0	0	1	9
Cara Kerja (Operasional) Aplikasi	0	0	0	1	9
Tampilan Keseluruhan/ <i>User Interface</i> Aplikasi	0	0	0	0	10

Tabel 5. Persentase Uji Coba Aplikasi

No	Penilaian	Presentase Skor
1	Operasional Sistem Aplikasi	99%
2	<i>User Interface</i> Aplikasi	99%
Rata-Rata Persentase		99%

SIMPULAN

Konsep OOP atau *Objected Oriented Programming* pada java ini, memiliki banyak sekali kegunaan untuk menciptakan sebuah program dari masalah kehidupan sehari-hari ini. Tidak salah memang bahwa banyak hal yang bisa diimplementasikan dari kehidupan sehari-hari dengan OOP java ini. Berdasarkan hasil uji coba pengguna, hasil kepuasan pengguna berdasarkan aspek penilaian operasional sistem dan *user interface* aplikasi sebesar 99% yang berarti sangat baik. Sehingga dapat disimpulkan bahwa aplikasi manajemen parkir berbasis java ini layak untuk digunakan. Program sistem parkir ini, dapat diterapkan pada masyarakat khususnya tempat penitipan kendaraan di kota-kota Indonesia. Adanya system parkir dengan Konsep OOP dapat memberikan kenyamanan

bagi pengguna parkir dan membuat kinerja petugas parkir lebih efisien.

DAFTAR RUJUKAN

Isnanto, Iskandar. 2013. Perancangan Sistem Parkir Pada Kampus UKSW Salatiga Kabupaten Salatiga. *Jurnal Informatika dan Komputer*, Vol.2 No.2 halaman 13.

Rosa dan Shalahuddin M. 2011. *Terstruktur dan Berorientasi Objek*. Bandung: Modula.

Sasmito, A. 2017. Rancangan Pembelajaran Pemrograman Java Dengan Pendekatan Object-First Untuk Mempermudah Memahami Konsep Object Oriented Programming (OOP). *Jurnal Pilar Nusa Mandiri*, Vol.10 No.2 Halaman 149.

Utomo, A. P. 2013. Analisa dan Perancangan Sistem Informasi Parkir di Universitas Muria Kudus. *Jurnal SIMETRIS*, Vol.3 No.1 Halaman 17-18.

Wahyudianto, E. 2013. Perancangan Sistem Parkir Pada Dinas Perhubungan Komunikasi dan Informatika Kabupaten Pacitan. *Jurnal Informatika dan Komputer*, Vol.2 No.1 Halaman 21.