

## MAKALAH SEMINAR TUGAS AKHIR

# PERANCANGAN APLIKASI SISTEM INFORMASI PERUMAHAN DENGAN MENGGUNAKAN PEMROGRAMAN BERORIENTASI OBJEK ZOPE

Andriyanto, Adian F.R, S.T, MT, Aghus Sofwan, S.T,MT  
Jurusan Teknik Elektro Teknik Universitas Diponegoro

*Abstrak - Saat ini teknologi informasi sudah berkembang sedemikian luas sehingga merambah kepada aplikasi sistem informasi dan perdagangan melalui dunia maya. Salah satunya adalah Sistem Informasi Perumahan yang memudahkan pihak pengembang perumahan ( developer ) untuk memperkenalkan tempat hunian kepada pengunjung situs web. Pemasaran perumahan konvensional akan membutuhkan biaya, waktu dan tenaga yang cukup besar sedangkan calon pembeli masih sangat membutuhkan informasi yang detail sebelum melakukan transaksi pembelian rumah. Mengatasi hal tersebut maka dibuatlah Sistem informasi perumahan lewat media internet yang dapat memberikan informasi detail dan dapat diakses langsung oleh calon pembeli dari luar maupun dalam negeri.*

*Metode yang digunakan dalam aplikasi ini adalah berorientasi objek, sedangkan perancangannya dibuat dengan menggunakan bahasa pemodelan Unified Modeling Language (UML). Sistem Informasi Perumahan ini dibuat dengan menggunakan bahasa pemrograman DTML ( Document Template Markup Language Zope) yang dipadukan dengan bahasa pemrograman HTML, basis data yang digunakan adalah ZODB, dan perintah SQL untuk melakukan akses basis data.*

*Sistem informasi ini diujikan dengan sebuah komputer server dan dapat diakses dengan beberapa komputer client. Bahasa scripting Zope yaitu DTML memungkinkan untuk memprogram sistem ZOPE via web dengan lebih ringkas. Basis data terintegrasi ZODB dapat melakukan kemampuan undo dan dapat memetakan model data hierarki dengan baik.*

*Kata Kunci :Zope, Zodb*

## I. Pendahuluan

### 1.1 Latar Belakang

Saat ini World Wide Web sudah tidak lagi menjadi sebuah kumpulan web statis, tetapi

sudah berkembang menjadi media transportasi berbagai layanan aplikasi web. Sebagai contohnya adalah basis data pengetahuan, mesin pencari internet dan berbagai macam kategori software baru seperti sistem manajemen data, sistem portal komunitas dan weblog. Sistem ini membutuhkan sebuah browser web untuk mengaksesnya tanpa membutuhkan tambahan software lainnya.

Perancangan program aplikasi untuk sistem informasi perumahan ini digunakan untuk memudahkan dalam proses penjualan, promosi dan informasi detail tentang perumahan. Dengan program aplikasi ini diharapkan pengunjung situs informasi perumahan ini akan mendapatkan informasi yang lebih detail seperti bentuk bangunan, kondisi fisik, lingkungan, tempat, akses keluar selain dapat pula mendapatkan informasi perubahan bentuk bangunan luar maupun dalam jika menginginkan suatu perubahan bangunan. Demikian pula tentang harga jual sebelum dilakukan perubahan maupun setelah dilakukan perubahan bentuk bangunan.

### 1.2 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah :

1. Mempelajari Zope yang digunakan sebagai server aplikasi dan server manajemen data untuk pembuatan situs Informasi Perumahan
2. Merancang sebuah aplikasi web dengan menggunakan pemrograman berorientasi objek Zope sebagai server aplikasi dan server manajemen data.

### 1.3 Batasan Masalah

Pembatasan masalah pada penulisan tugas akhir ini sebagai berikut :

1. Pembuatan aplikasi web dengan server Zope yang berjalan dalam platform linux dan dengan pemrograman berorientasi objek
2. Zope digunakan sebagai server aplikasi web dan server manajemen data

## II Landasan Teori

### 2.1 ZOPE

Zope adalah aplikasi server *opensource* yang dikembangkan oleh badan hukum Zope dan komunitas aktif dari pengguna. Zope dapat berjalan pada sistem operasi Linux dan Windows dan dapat pula digunakan dengan server web populer lain. Zope dikembangkan dengan bahasa pemrograman python yang merupakan bahasa pemrograman berorientasi objek dengan komponen yang ditulis dalam bahasa C. Teknologi yang hadir dalam Zope adalah berdasar pada pemrograman berorientasi objek yang tepat untuk diaplikasikan pada pemrograman web.

#### 2.1.2 Zope Sebagai Server Aplikasi Web

Sebuah server aplikasi web berfungsi sebagai jembatan (*middleware*) dari browser, aplikasi software dan database. Zope menyediakan dasar yang memungkinkan untuk menggabungkan HTML, DTML, SQL, dan Python.

#### 2.1.3 Zope Sebagai Server Manajemen Data

Zope menawarkan sistem manajemen data dasar sebagai tampilan administrasi web, dan banyak fungsi yang telah terintegrasi dalam Zope, diantaranya banyak fungsi dari *content management system* (CMS) yang telah terintegrasi dalam Zope.

### 2.2 Database Relasional ZOPE

Secara umum aplikasi internet kompleks menggunakan sebuah sistem database, tidak terkecuali dengan Zope, ZODB merupakan basis data yang menyatu dengan sistem Zope. Namun ada aplikasi database lain yang dapat menyimpan data lebih efisien menggunakan database relasional tradisional atau dapat menggunakan basis data yang ada.

Sistem basis data merupakan suatu sistem terkomputerisasi yang tujuannya adalah memelihara pemakai sistem basis data dan dapat melakukan operasi, yaitu:

1. Menambahkan file baru.
2. Menyisipkan data ke suatu berkas.
3. Mengambil data yang ada pada suatu berkas.
4. Mengubah data pada suatu berkas.
5. Menyajikan suatu informasi yang diambil dari sejumlah berkas.

### 2.3 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan cara berpikir atau berlogika dalam menghadapi masalah dengan bantuan komputer.

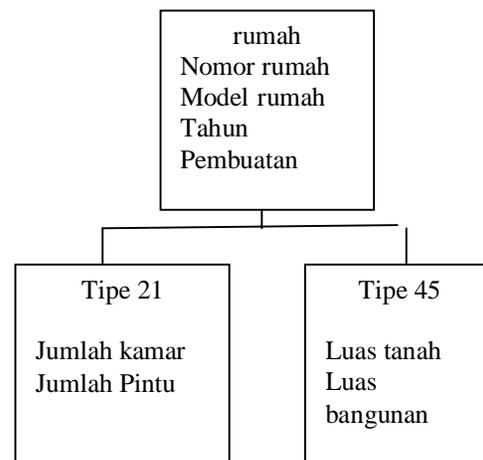
OOP (*Object Oriented Programming*) menganalisa permasalahan sebagai bagian dari dunia nyata dimana setiap objek merupakan entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu. Dalam pemrograman berorientasi objek terdapat 3 metodologi dasar yaitu: Pemodulan / pengkapsulan (*Encapsulation*), Penurunan (*Inheritance*), Polimorfisme (*Polimorphism*).

#### 2.3.1 Enkapsulasi

Enkapsulasi adalah suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem itu sendiri. Akses ke dalam sistem diatur sedemikian rupa oleh seperangkat interface. Contoh dari kasus ini adalah penggunaan kelas pencarian Janis rumah. Pemrogram bisa memanggil dan menggunakan kelas tersebut tanpa harus mengetahui isi program dan cara kerja dalam program tersebut.

#### 2.3.2 Pewarisan

Pewarisan adalah suatu konsep untuk berbagi dari apa yang dimiliki oleh suatu kelas atau objek bagi kelas atau objek yang lain. Sebagai contoh kelas rumah yang berbagi atribut yang sama misalnya : tahun pembuatan, model, jumlah kamar dan sebagainya. Seperti dijelaskan pada Gambar 2.1 yang menggambarkan tentang fenomena pewarisan.



Gambar 2.1 pewarisan atribut dari superkelas ke subkelas

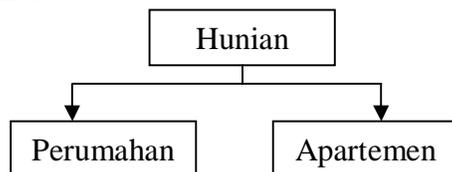
#### 2.3.3 Polymorphism

Polymorphism berasal dari bahasa Yunani yang berarti “banyak bentuk”. Dalam OOP, konsep ini memungkinkan digunakannya

suatu interface yang sama untuk memerintah suatu objek agar melakukan tindakan yang secara prinsip sama tetapi memiliki proses yang berbeda. Dalam istilah umum *polymorphism* memiliki istilah satu interface banyak aksi. Contoh nyata dalam polimorfisme adalah rumah, dimana rumah terdiri dari berbagai tipe tetapi kesemuanya memiliki interface yang sama, yaitu desain dalam dan tampak depan, luas bangunan dan luas tanah serta interface yang kesemuanya memiliki dasar yang sama.

## 2.4 Zope dan Pemrograman Berorientasi Objek

Objek merupakan konsep, abstrak, atau sesuatu yang memiliki arti bagi aplikasi yang akan kita kembangkan. Objek bisa berupa kata benda, akan tetapi objek dalam konsep OOP bukanlah sesuatu yang dapat dilihat atau diraba dalam dunia nyata, seperti halnya mobil, kuda, sepeda, tetapi juga menyangkut entitas seperti persamaan kuadrat, liberalisme, marxisme dan sebagainya. Selain itu dalam objek dikenal dengan identitas dimana identitas adalah sesuatu yang membedakan suatu objek dengan objek lainnya. Berlawanan dengan konsep identitas, yaitu objek dapat dibedakan satu sama lainnya, dan beberapa objek dapat dikelompokkan menjadi satu kelas tunggal. Seperti halnya pada Gambar 2.2 sebagai contoh jenis rumah yang dapat dikategorikan dalam satu kelas yaitu kelas hunian.



Gambar 2.2 Klasifikasi

### 2.4.1 Kelas (*class*)

Sekarang apabila perumahan dibandingkan dengan perumahan lain, perumahan 2. Perumahan 2 memiliki luas tanah yang luas, penghuni yang banyak dan tempat yang strategis dari jalan raya. Dari keduanya terdapat perbedaan pada atribut dari objek perumahan tetapi memiliki kemampuan dan sifat yang sama. Kedua perumahan tersebut memiliki persamaan pada objeknya, maka kedua perumahan tersebut dapat dikatakan dalam kelas (*class*). Sebuah kelas mendeskripsikan objek – objek yang memiliki atribut dan method yang sama.



Gambar 2.3. Kelas

### 2.4.2 Objek

Sebuah objek diturunkan dari sebuah kelas dan dibangun sesuai dengan property dari sebuah kelas. Penurunan memiliki analogi yang sama seperti halnya jika membuat folder seperti direktori file dengan menurunkan sebuah objek folder dari sebuah kelas yang mendefinisikan atribut dan method folder. Maka ketika itu pula nama diberikan pada objek dan atribut atau property lainnya.

### 2.4.3 Method dan Atribut

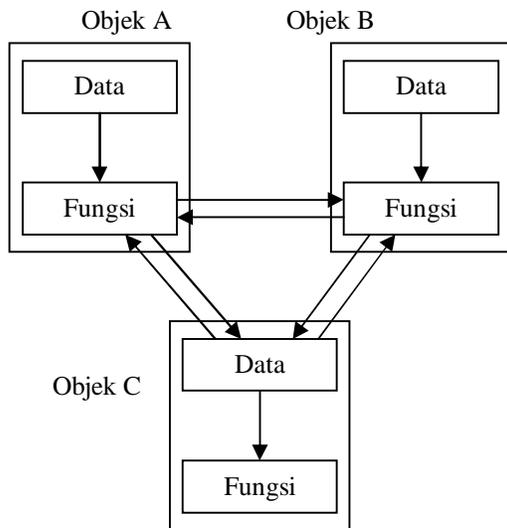
Sebagai contoh perumahan memiliki atribut nama, desain, jumlah rumah, lokasi dan lain - lain yang masing – masing memiliki nilai tertentu. Jika perumahan tersebut adalah memiliki gaya mediterania maka atribut dari perumahan tersebut adalah jenis mediterania.

Sebagai contoh pemilik perumahan mengetahui kapan harus ditempati oleh penghuni, kapan melakukan renovasi bangunan, keamanan dan sebagainya. Aksi pada pemilik perumahan tersebut dinamakan dengan method.

### 2.5 Metode Berorientasi Objek

Pendekatan berorientasi objek akan membuat data terbungkus pada setiap fungsi / prosedur dan melindunginya terhadap perubahan yang tidak dikehendaki dari fungsi yang berada di luar seperti yang ditunjukkan pada gambar 2.4. Beberapa karakteristik yang menjadi ciri – ciri dari pendekatan berorientasi objek adalah :

- ❖ Pendekatan lebih pada data dan bukan pada prosedur/fungsi
- ❖ Program besar dibagi pada apa yang dinamakan objek.
- ❖ Struktur data dirancang dan menjadi karakteristik dari objek
- ❖ Fungsi yang mengoperasikan data tergabung dalam suatu objek yang sama
- ❖ Data tersembunyi dan terlindung dari fungsi / prosedur yang ada di luar
- ❖ Objek dapat saling berkomunikasi dengan saling mengirim pesan. Pendekatan yang dilakukan adalah dari bawah ke atas.



Gambar 2.4 Pengorganisasian Data serta Fungsi pada Pendekatan Berorientasi Objek.

## 2.6 Unified Modeling Language ( UML )

UML adalah bahasa standar untuk membuat pemodelan. UML sendiri menyediakan beberapa jenis diagram untuk merepresentasikan entitas dan hubungan yang terdapat dalam aplikasi. Proses pembangunan aplikasi dibagi menjadi beberapa siklus. Setiap kali satu siklus dilakukan, maka dilakukan evaluasi sebagai bahan untuk memulai siklus berikutnya. Setiap siklus biasanya terdiri dari atas beberapa tahap yaitu analisa permintaan, analisa sistem, desain dan implementasi. UML akan digunakan dalam tahap analisa dan desain. Diagram yang disediakan UML dan paling sering digunakan antara lain yaitu :Use Case Diagram, Diagram Sekuensial ( Sequence ), Diagram Kolaborasi, dan Diagram Kelas.

## 2.7. Klasifikasi Rumah Dengan Pemrograman Berorientasi Objek.

Dalam melakukan proses perancangan perangkat lunak untuk klasifikasi rumah sebelum melakukan pengkodean, maka terlebih dahulu harus melakukan pemodelan (*modeling*). Pemodelan adalah suatu penyederhanaan dari kenyataannya. Alasan yang fundamental kenapa harus membuat suatu model adalah karena saat memodelkan, akan lebih mengerti sistem yang akan dibangun. Melalui pemodelan, ada 4 tujuan yang akan dicapai:

1. Pemodelan membantu untuk memvisualisasikan suatu sistem.
2. Pemodelan memberikan spesifikasi struktur dan tingkah laku sistem.

3. Pemodelan memberikan bentuk yang memandu untuk membangun sistem.
4. Pemodelan mendokumentasikan keputusan yang telah dibuat.

Pemodelan berorientasi objek direpresentasikan dengan beberapa diagram, yaitu:

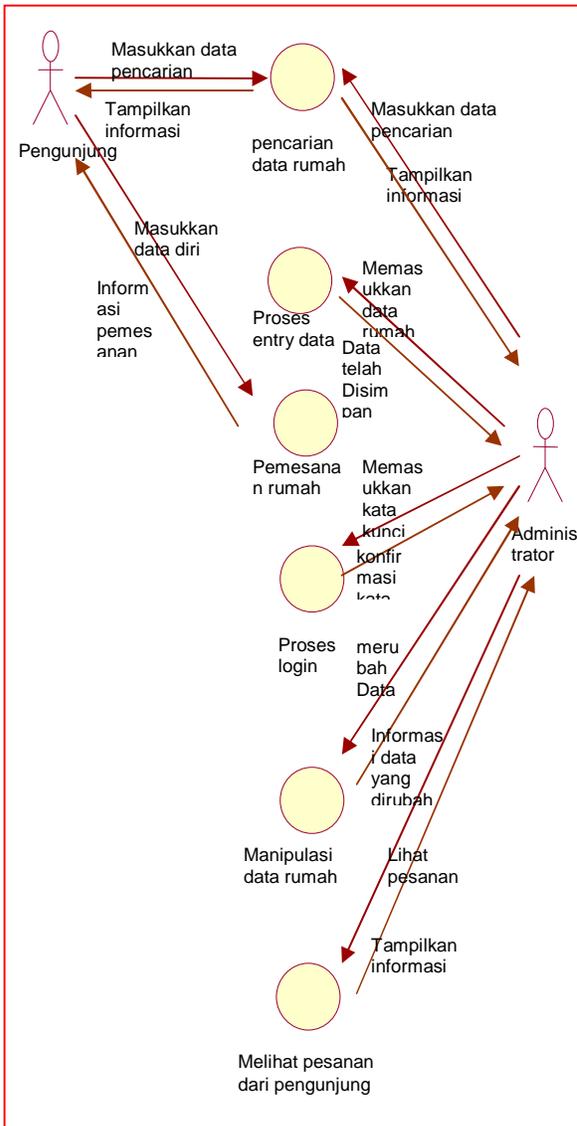
Diagram Use Case, Diagram Kelas, Diagram Statechart, Diagram Aktivitas, Diagram Sequence, Diagram Kolaborasi

## III PERANCANGAN APLIKASI

Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model adalah deskripsi dari suatu permasalahan atau topic dari aplikasi yang akan dibuat. Dengan menggunakan model ini akan membantu perancang dalam memahami lingkup permasalahan yang akan dipecahkan. Model itu sendiri adalah visualisasi dari aplikasi yang akan dibangun. Model piranti lunak dapat dianalogikan seperti cetak biru pada suatu perancangan. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin kompleks sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Untuk membuat pemodelan digunakan UML (*Unified Modeling Language*). UML menyediakan beberapa jenis diagram untuk merepresentasikan entitas dan hubungan yang terdapat dalam aplikasi.

### 3.1 Diagram Use case ( Use Case Diagram )

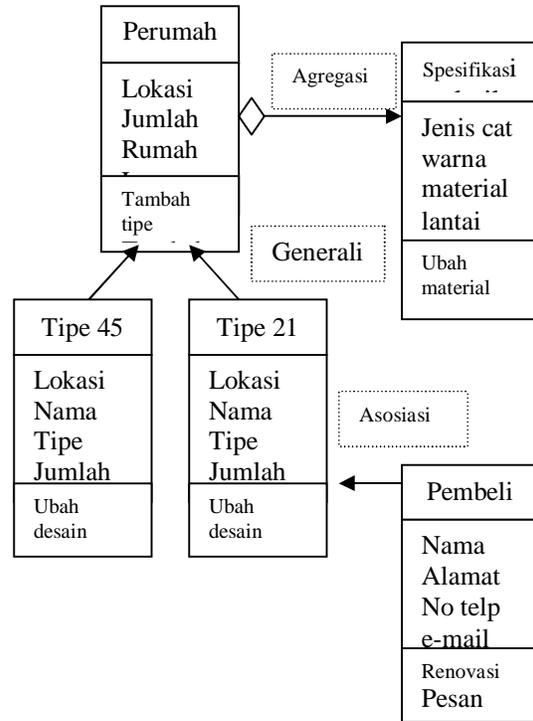
Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Gambar 3.1 menunjukkan main use case untuk Sistem Informasi Perumahan. Sistem Informasi Perumahan pada dasarnya membantu pengunjung dalam mencari tipe rumah berdasarkan salah satu atribut rumah, sistem ini terdiri dari use case untuk proses mencari jenis rumah yang terdiri dari memasukkan data-data, melihat data-data rumah dan membandingkan jenis rumah berdasarkan atributnya.



Gambar 3.1 Diagram Use case

**3.2. Diagram Kelas**

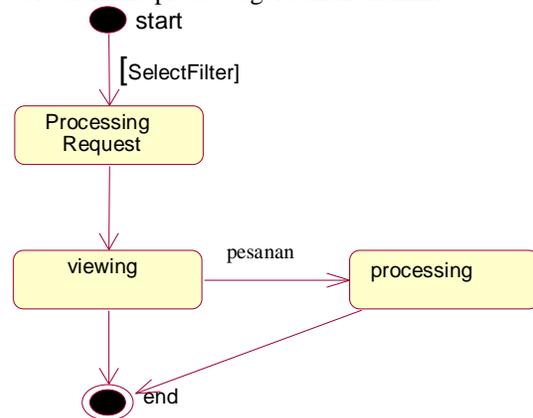
Kelas adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Kelas menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda / fungsi / operasi). Diagram kelas menggambarkan struktur dan deskripsi kelas, paket dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. kelas memiliki tiga area pokok : yaitu Nama kelas, Atribut dan Metode



Gambar 3.2 Diagram Kelas

**3.3 Diagram Statechart**

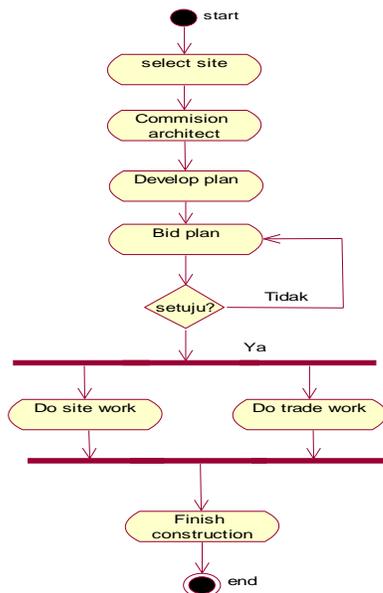
Diagram statechart menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya diagram statechart menggambarkan kelas tertentu (satu kelas dapat memiliki lebih dari satu diagram statechart). Penjelasan pada Gambar 3.3 adalah saat dimulai proses terjadi permintaan tampilan halaman sistem informasi. Kemudian terjadi interaksi antara pengunjung dengan sistem informasi. Apabila terjadi kontak antar pengunjung dengan pengelola maka akan diproses pada kotak proses. Setelah semuanya selesai maka proses segera akan diakhiri.



Gambar 3.3 Diagram statechart

### 3.4 Diagram Aktivitas

Diagram aktivitas menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram aktivitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas merupakan diagram *state* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

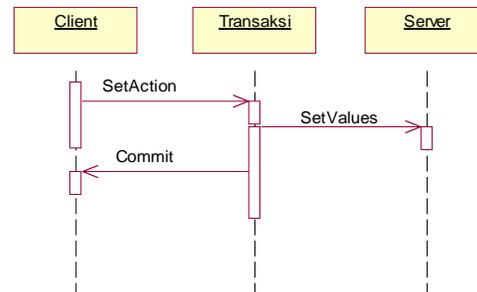


Gambar 3.4 Diagram Aktivitas

### 3.5 Diagram Sequence

Diagram sequence menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Diagram sequence terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Diagram sequence biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan keluaran tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan keluaran apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. Pesan digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, pesan akan dipetakan

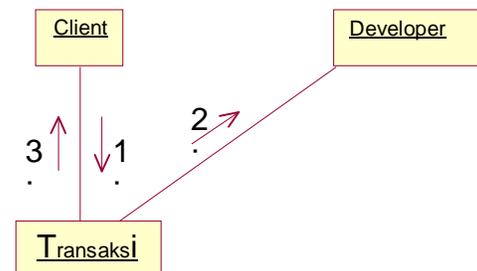
menjadi operasi / metoda dari kelas. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah pesan. Gambar 3.5 menunjukkan diagram sequence.



Gambar 3.5 Diagram Sequence

### 3.6 Diagram Kolaborasi

Diagram kolaborasi juga menggambarkan interaksi antar objek seperti diagram sequence, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian pesan. Setiap pesan memiliki urutan sequence, di mana pesan dari level tertinggi memiliki nomor 1. Pesan dari level yang sama memiliki prefiks yang sama. Gambar 3.6 menunjukkan diagram kolaborasi.



Gambar 3.6 Diagram Kolaborasi

## IV. IMPLEMENTASI DAN PENGUJIAN

### 4.1 Implementasi

Implementasi dari tugas akhir ini adalah menampilkan antarmuka untuk pengguna, admin dan tampilan antarmuka untuk proses manipulasi basis data.

### 4.1 Implementasi Basis Data

Basis data dari aplikasi *virus information centre* ini mempunyai beberapa objek yang disimpan dalam direktori *zope/var/zgadfly*.

## 4.2 Pengujian

Pengujian aplikasi sistem informasi ini meliputi pengujian basis data, Tampilan Sistem Informasi Perumahan, dan pengujian pada tampilan menu admin. Pengujian dilakukan pada sisi client pada jaringan komputer.

### 4.2.1 Pengujian Basis Data

Database yang digunakan pada sistem informasi ini adalah ZODB ( Zope Object Database). Database ini dihubungkan dengan server zope dengan menggunakan *ZGadfly Database Connection* yang merupakan koneksi database internal yang telah disediakan oleh Zope. Untuk pemanggilan isi data dari ZODB digunakan *ZSQL Method* yang merupakan fungsi yang disediakan oleh Zope untuk memanggil isi database dengan menggunakan perintah SQL (*SQL Query*).

### 4.2.2 Pengujian Tampilan Sistem Informasi Perumahan

Untuk tampilan aplikasi ini disajikan menu utama beserta isi dari menu Sistem Informasi Perumahan. Menu utama berisikan menu untuk pencarian database atribut dari rumah, link ke tampilan lengkap database, link ke denah perumahan, link ke pengembangan rumah baru.

### 4.2.3 Pengujian Tampilan Administrator

Menu untuk administrator memiliki fungsi untuk melihat isi data, melakukan perubahan isi data dan melakukan perubahan sistem. Sebelum masuk kedalam menu administrator admin diharuskan mengisi nama dan password untuk otentifikasi peran. Tampilan login Zope ditampilkan pada gambar 4.1



Gambar 4.1 Tampilan Menu Login Admin

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari pembuatan tugas akhir Perancangan Sistem Informasi Perumahan Dengan Menggunakan Pemrograman Berorientasi Objek Zope serta pengujian terhadap aplikasi yang dibuat dapat disimpulkan :

1. Pada perancangan menggunakan bahasa pemodelan *Unified Modeling Language* terdapat 3 *use case*, yaitu *use case* untuk memasukkan data-data rumah, *use case* untuk memasukkan kata kunci, dan *use case* untuk memanipulasi basis data.
2. Masing-masing *use case* pada sistem informasi perumahan ini dijabarkan dalam diagram kelas, diagram kolaborasi, diagram statechart dan diagram aktivitas.
3. Dengan bahasa pemrograman DTML Zope memungkinkan untuk mengakses kumpulan fungsi yang banyak jumlahnya dan melakukan aksi kompleks dengan penggunaan template dengan beberapa baris kode pemrograman saja.
4. Pengelompokkan data dalam sistem informasi perumahan ini berdasarkan pada tipe, no blok, tampak muka, harga, kondisi, lokasi dan status.
5. Pemrograman berorientasi objek terbukti telah memberikan banyak kemudahan, diantaranya dalam penciptaan maupun penghilangan suatu objek tanpa mengubah / mempengaruhi keseluruhan sistem

### Saran

Pembuatan Sistem informasi kedepan diharapkan dapat memberikan fasilitas transaksi online via web ( E-commerce) dan pengunjung dapat memberikan ide atau desain untuk dikembangkan oleh pihak pengembang.

### DAFTAR PUSTAKA

- [1] Irwanto, Djon, S.kom, MM, *Perancangan Object Oriented Software dengan UML*, Yogyakarta, Andi, 2006
- [2] Nugroho, Adi, ST., MMSI, *Analisis dan perancangan sistem informasi dengan metodologi berorientasi objek*, Bandung, Informatika, 2002
- [3] Aziz, Farid M.M.Kom, *Object Oriented Programming dengan PHP*, Jakarta, Elex Media Komputindo, 2005
- [4] Hermawan, Julius, *UML dan VB.Net*, Yogyakarta, Andi, 2004
- [5] Suhendar, A.S.Si, Gunadi, Hariman, S.Si, MT, *Visual Modelling menggunakan UML dan Rational Rose*, Bandung, Informatika, 2002
- [6] Rumbaugh, Booch Jacobson, *UML Distilled*, Yogyakarta, Andi, 2004
- [7] Beehive, *ZOPE*, Jakarta, Elex Media Komputindo, 2003
- [8] --- <http://www.zope.org/Members/maryniak/zope-on-suse64-howto/>
- [9] --- <http://www.sampublishing.com/about/help.asp>



Andriyanto (L2F304213)  
Desember 1982  
Mahasiswa Teknik  
Elektro 2004,  
konsentrasi Informatika  
dan komputer,  
Universitas Diponegoro  
Email :  
andrian\_28@yahoo.com

Menyetujui dan Mengesahkan

Pembimbing I

Pembimbing II

Adian F R, ST, MT  
NIP. 132 205 680

Aghus Sofwan . S.T, MT  
NIP. 132 163 757