KONSEP OOP PADA PEMROGRAMAN VISUAL BASIC

Oleh :
Sri Eniyati, S.Kom dan Rina Candra Noor Santi, S.Pd, M.Kom

ABSTRACT:

Object-Oriented Programming is inovation in development for software the last decade, moreover to appear visual programming language models. OOP concept needed for implementation components used in application preparation and preparation new component, so that to appear visual programming language models,

Preparation new component in veil VB programming is news interesting, except for to fill requisites unique new component, so using plentiful OOP concept.

Keywords : *OOP, OOP concept, Visual Programming Language*

ABSTRAK

Pemrograman berorientasi obyek adalah inovasi dalam pengembangan perangkat lunak dalam dekade terakhir, lebih-lebih setelah bermunculan berbagai macam bahasa pemrograman visual. Konsep OOP sangat diperlukan pada implementasi penggunaan komponen-komponen dalam perancangan aplikasi dan juga dalam perancangan/pembuatan komponen baru, sehingga dengan bermunculan berbagai macam bahasa pemrograman visual.

Perancangan komponen baru dalam lingkup pemrograman VB merupakan topik yang sangat menarik, selain untuk memenuhi kebutuhan komponen baru yang unik, juga menerapkan kelebihan-kelebihan yang dimiliki konsep OOP.

Kata Kunci : *OOP, konsep OOP, Bahasa Pemrograman Visual*

PENDAHULUAN

Pemrograman Visual merupakan salah satu unsur penting dalam perancangan tampilan, dengan adanya komponen ini maka perancangan program, khususnya tampilan akan sangat mempermudah, sehingga sebagian besar waktu dapat dikonsentrasikan pada isi programnya.

VB adalah salah satu software untuk membangun aplikasi dalam lingkungan windows. Pengembangan aplikasi, VB menggunakan pendekatan visual untuk merancang antar muka (user interface) dalam bentuk form, sedang untuk kode programnya menggunakan bahasa basic. VB program terdiri dari banyak sub program (prosedur), dimana setiap prosedur mempunyai kode tersendiri dan dapat dieksekusi sendiri, pada saat yang bersamaan dapat digabungkan menjadi satu. Untuk pengembangan aplikasi di dalam pemograman visual dimulai dengan pembentukan user interface, kemudian mengatur properti dari obyek-obyek yang digunakan dalam user interface. Kode pogram ditulis untuk menentukan kejadian (event). Yang menarik dari komponen visual adalah terbukanya kesempatan untuk merancang sendiri komponen tersebut. Bahkan komponen visual yang ada dapat dikembangkan lebih lanjut untuk mendapatkan hasil tertentu sesuai dengan keinginannya.

Pemodelan obyek menggambarkan struktur sistem data, obyek-obyek dan hubungan antar obyek data tersebut. Model ini dibangun dari kelas-kelas yang berhubungan dengan kelas lain melalui asosiasi. Berikut adalah beberapa komponen–komponen penting yang berhubungan dengan model menurut Blaha dan Pemerlani (1998):

1. Obyek

   Obyek adalah konsep, abstraksi atau sesuatu yang memiliki arti untuk aplikasi. Pemilihan obyek tergantung pada pendapat dan keadaan masalah. Masing–masing obyek ada dan dapat diidentifikasi. Identifikasi adalah sifat yang membedakan masing–masing obyek dari obyek–obyek lain.

2. Kelas

   Kelas adalah deskripsi dari grup obyek dengan sifat–sifat (atribut obyek) yang serupa, perilaku umum (operasi dan diagram status), hubungan–hubungan yang serupa (mirip) dengan obyek–obyek lain dan semantic umum. Kelas mendefinisikan maksud (tema, esensi, dan arti semantic dari suatu kelas) dari struktur dan perilaku obyek–obyek dari bentuk tertentu.

3. Atribut

   Atribut obyek adalah sifat yang diberi nama atau merupakan suatu property dari suatu kelas yang menggambarkan nilai–nilai yang dipenuhi oleh setiap obyek kelas.

4. Operasi

   Operasi adalah fungsi atau prosedur yang mungkin diaplikasikan terhadap obyek–obyek dalam kelas. Implementasi operasi kelas dinamakan metode.

5. Link

   Link adalah hubungan fisik atau konseptual antar obyek–obyek. Link dari suatu asosiasi menghubungkan obyek–obyek dari kelas –kelas yang sama dan memiliki sifat–sifat yang serupa (atribut link).

6. Asosiasi

   Asosiasi mendiskripsikan sekelompok link dengan struktur dan semantic yang lazim sehingga link sebagai instant dari asosiasi. Link dari asosiasi tersebut merelasikan obyek–obyek dari kelas–kelas yang sama dan mempunyai property (atribut link) yang sama.

7. Multiplicity

   Multiplicity menentukan banyaknya instance dari suatu kelas yang dapat berhubungan dengan instance tunggal dari suatu kelas yang sesuai yang saling berhubungan.

8. Qualifier

   Suatu asosiasi dimana obyek–obyek dalam aturan "many" sebagian atau seluruhnya di satu artikan (disambiguated) oleh suatu atribut yang dinamakan qualifier. Asosiasi yang dpat diberi sifat qualifier adalah One To Many dan Many To Many. Qualifier memilih diantara obyek–obyek target, mengurangi multiplicity efektif dari "Many" To "One".

9. Generalisasi

   Generalisasi adalah hubungan antara suatu kelas (superclass) dan satu atau lebih variasi dari kelas lain (subclass). Generalisasi mengorganisasikan kelas–kelas berdasarkan kemiripan dan perbedaan kelas–kelas tersebut. Superclass mengandung atribut–atribut umum, operasi–operasi, diagram status dan asosiasi. Sedangkan subclass menambahkan atribut, operasi, diagram status dan asosiasi tertentu.

10. Instansi

    Instansi merupakan keterhubungan diantara obyek dan kelasnya. Notasinya adalah garis terputus dari instant ke kelas dengan tanda panah terhadap kelas dan diberi label kata "instant".

Menurut Edward Yourdon (1994) suatu sistem yang dibangun dengan metode berorientasi obyek merupakan satu sistem dimana komponen–komponennya adalah data dan fungsi yang dienkapsulasi, yang menurunkan atribut dan perilaku dari komponen sejenis lainnya, dan komponen tersebut berkomunikasi melalui pesan dengan komponen lainnya. Sesuatu yang menarik dalam pemrograman visual adalah didukungnya konsep OOP dalam implementasinya. Hal yang mendasar dalam lingkungan pemrograman visual adalah obyek.

Keunggulan Pemodelan Berorientasi Obyek

Pemilihan Metode berorientasi obyek, karena berorientasi obyek memiliki banyak keunggulan, antara lain meningkatkan kualitas, mudah untuk dikembangkan, mudah untuk dideteksi kekurangannya dan meningkatkan komunikasi antara pengembang dan pengguna. Apabila diterapkan pada basisdata, maka pemodelan berorientasi obyek meningkatkan kualitas data, kinerja sistem lebih baik, mudah dalam pencarian kesalahan, dan pengembangan lebih cepat (Blaha dan Pemerlani, 1998).

Perbedaan OOA, OOD dan OOP

Perbedaan antara OOA, OOD, OOP adalah bahwa OOA singkatan object oriented analisis, menggambarkan daerah problem (problem domain) dengan mengidentifikasikan dan menspesifikasikan sejumlah obyek semantik yang berinteraksi dan berperilaku menurut kebutuhan sistem. Model OOA juga menggambarkan kolaborasi antar obyek untuk melakukan kerja sistem yang dispesifikasikan. Tujuan analisis adalah untuk memahami permasalahan dan merencanakan dengan baik model dunia nyata. Dalam analisis dijelaskan tentang apa yang harus dilakukan bukan bagaimana melakukan dan suatu masalah harus dipahami sebelum dipecahkan.

OOD adalah object oriented desain yaitu merupakan ekstensi dari OOA dengan memanfaatkan kelas dan obyek yang telah dihasilkan dari OOA. Metode OOD mendesain atau merancang suatu sistem. Perancangan sistem merupakan tahap awal dalam membuat suatu keputusan untuk memecahkan masalah.

Sedangkan OOP adalah object oriented programing yaitu inovasi dalam pengembangan perangkat lunak dalam dekade terakhir. Cepat atau lambat setiap programmer akan dipengaruhi oleh pendekatan berorientasi obyek dalam perancangan program.

Konsep Analisis dan Perancangan Berorientasi Obyek

Konsep Analisis Berorientasi Obyek

Pada proses pembuatan system informasi ini yang dilakukan pertama kali yaitu menganalisis sejumlah obyek yang berinteraksi dan berperilaku menurut kebutuhan sistem. Tujuan analisis adalah memahami permasalahan dan merencanakan dengan baik model dunia nyata. Dalam

analisis dijelaskan tentang apa yang harus dilakukan bukan bagaimana melakukan dan suatu masalah harus dipahami sebelum dipecahkan. Sehingga untuk memudahkan dalam memahami suatu permasalahan maka diperlukan langkah–langkah selama proses analisis. Hasil analisis pemodelan berorientasi obyek menurut Erdward Yourdon (1994) terdiri atas tahapan-tahapan:

1. Menentukan Obyek dan Kelas

- **Obyek**

  Ada beberapa penulis yang memberikan definisi "Obyek", diantaranya adalah:

  - Menurut Peter Coad (Yourdon, 1994), obyek adalah sebuah abstraksi dari sesuatu yang ada dalam kawasan masalah, menggambarkan kemampuan sistem untuk memuat informasi tentangnya, berinteraksi dengannya atau keduanya; suatu kapsulisasi nilai-nilai atribut serta pelayanan exclusive.
  - Menurut Yourdon (1994), obyek adalah nyata memiliki banyak karakteristik yang sama dalam kelas atau tipe obyek (same caracteristic in class or object type).

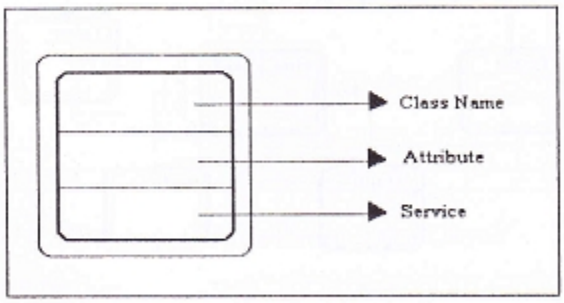
  Jadi obyek adalah abstraksi atau modal dari sebuah entitas yang menyimpan informasi enkapsulasi nilai atribut dan fungsi yang dimiliki.

- **Kelas**

  Kelas adalah deskripsi dari grup obyek dengan sifat–sifat (atribut obyek) yang serupa, perilaku umum (operasi dan diagram status), hubungan–hubungan yang serupa (mirip) dengan obyek–obyek lain dan semantic umum. Kelas mendefinisikan maksud (tema, esensi, dan arti semantic dari suatu kelas) dari struktur dan perilaku obyek–obyek dari bentuk tertentu.

- **Notasi kelas**

  Notasi untuk kelas ditunjukkan pada Gambar 2 garis tipis yang melingkar disekeliling batas kelas hanya menandakan bahwa kelas tetap hidup.



Gambar 2. Notasi kelas

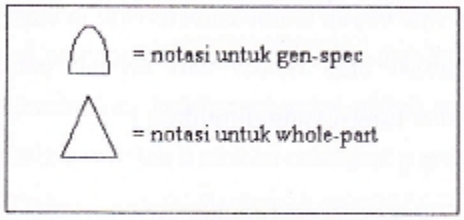
2. Menentukan Struktur

Struktur mengidentifikasikan relasi antar obyek yang membentuk komposisi seperti halnya relasi pewarisan. Terdapat 2 tipe struktur yaitu:

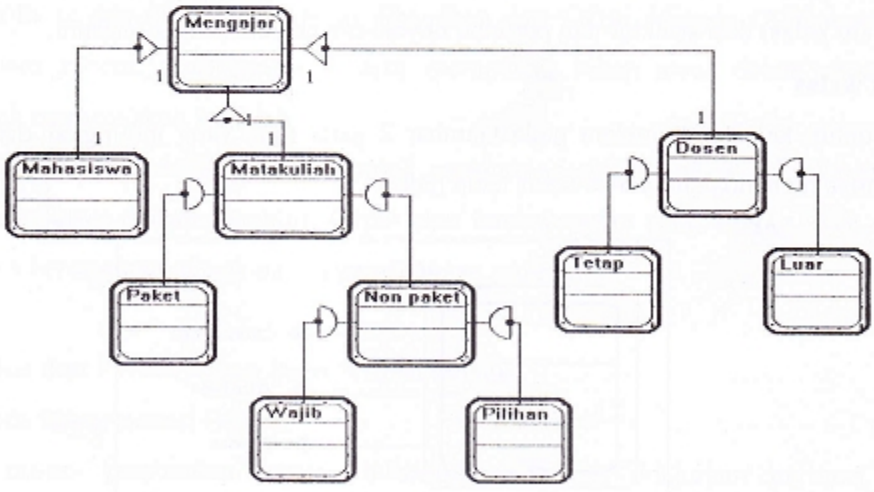
1. Gen-spec (Pewarisan)
2. Whole-part

Struktur Gen-Spec membentuk relasi pewarisan, sedangkan struktur Whole-Part untuk mengidentifikasikan relasi komposisi. Struktur Gen-Spec dibangun bila obyek atau generalisasi mempunyai atribut atau karakteristik yang dipakai bersama oleh semua obyek anak atau spesialisasi. Struktur Whole-Part dibangun bila obyek induk dikomposisi dari beberapa obyek anak. Struktur ini walaupun tidak memiliki pewarisan tetapi memiliki multiplicity dan participation (Yourdon, 1994).

- Notasi Gen-spec dan Whole-Part dapat dilihat pada Gambar 3.



Gambar 3. Notasi gambar Gen-Spec dan Whole-Part



Gambar 3.1 Contoh Notasi gambar Gen-Spec dan Whole Part

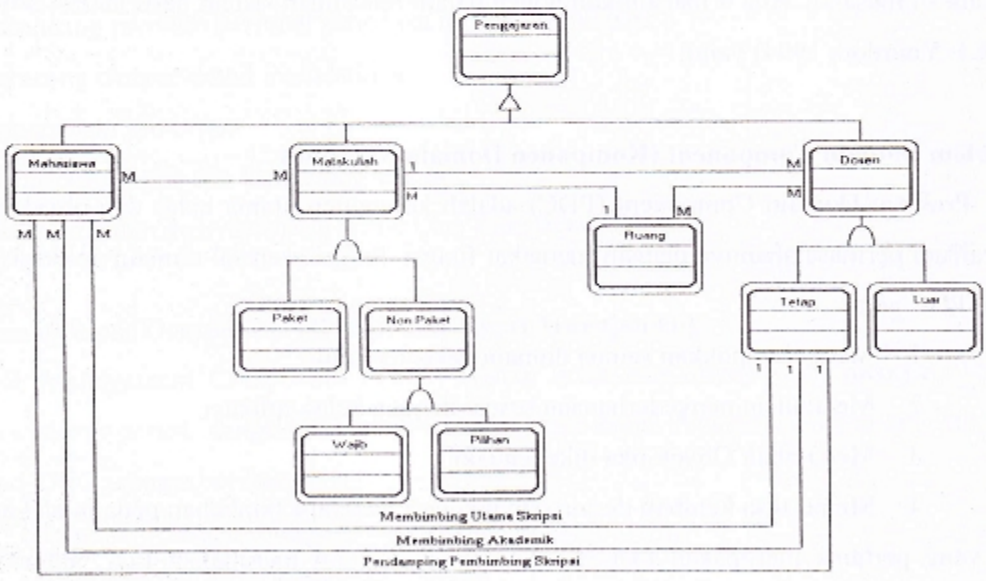
3. Menentukan Atribut

Atribut dalam pemodelan berorientasi obyek terkait dengan atribut obyek dan relasi antar obyek. Atribut dapat memberikan gambaran umum sebuah obyek dan memiliki nilai tertentu. Contoh atribut dapat dilihat pada Gambar 4

Gambar 4. Contoh Atribut pada Kelas Dosen

4. Menentukan Relasi Antar Obyek

Relasi menunjukkan adanya hubungan diantara sejumlah instance yang berasal dari obyek yang berbeda. Suatu relasi obyek adalah merupakan penggambaran statis dari kebijaksanaan pengguna (Yourdon, 1994). Sebagai contoh ditunjukkan dalam gambar 5:



Gambar 5. Contoh Relasi Antar Obyek

5. Menentukan Service

Service adalah fungsi khusus yang harus dimiliki oleh kelas atau obyek. dengan menentukan state yang dibutuhkan oleh tiap kelas. Service menentukan hubungan pesan (message connection) yang menghubungkan antara kelas pengirim dengan kelas penerima.

Gambar 6. Notasi service

Konsep Perancangan Berorientasi Obyek

Setelah analisis dikerjakan, maka langkah berikutnya adalah mendesain atau merancang sistem. Perancangan sistem merupakan tahap awal dalam membuat suatu keputusan untuk memecahkan masalah. Ada 4 macam komponen dalam rancangan sistem berorientasi obyek (Peter Coad & E.Yourdon, 1994) yaitu:

1. Problem Domain Component (Komponen Domain Masalah)

Problem Domain Component (PDC) adalah komponen utama kelas dan obyek yang akan memecahkan permasalahannya dengan memakai fungsi–fungsi esensial domain aplikasi. Langkah-langkah PDC yaitu:

1. Mengelompokkan semua domain kelas tertentu
2. Melakukan penyederhanaan sesuai dengan kelas aplikasi
3. Menambah Obyek jika dikehendaki
4. Memeriksa kembali desain dan menguji beberapa tambahan pada model analisis.

Untuk yang pertama merupakan PDC awal, sedangkan 2,3,4 merupakan PDC akhir. Jadi PDC terdiri dari dua yaitu PDC awal dan PDC akhir. Untuk PDC awalnya adalah menentukan subyek atau komponen masalah apa yang akan dikerjakan, sedangkan PDC akhir adalah menyederhanakan dari subyek masalah yang akan dikerjakan.

2. Data Management Component (Komponen Data Manajemen)

Data Managemet Component (DMC) adalah data manajemen kelas dan obyek yang diperlukan bagi interface ke database yang digunakan. Adapun langkah-langkah dalam menyusun DMC yaitu

1. Merancang dengan baik struktur datanya
2. Merancang tool atau servis yang akan digunakan untuk mengatur struktur data tersebut.
3. Merancang obyek dan kelas yang akan dipakai untuk aplikasi.
2. Menentukan data-data yang akan dipakai untuk sistem informasi tersebut

Pada DMC ini ditekankan pada basis datanya. Pada tesis ini pembuatan databasenya memakai Microsoft access, karena Microsoft access mudah untuk mengaplikasiannya dan mudah dipahami.

3. Human Interaction Component (Komponen Interaksi Manusia)

Human Interaction Component (HIC) adalah kelas dan obyek yang menyediakan interface antar user dengan sistem yang akan dibangun. Adapun langkah-langkah dalam menentukan HIC yaitu:

1. Menentukan siapa yang akan memakai sistem tersebut
2. Menyusun struktur masing-masing tugas
3. Merancang perintah-perintah yang bisa dipakai untuk user
4. Meracang dengan detail interaksinya
5. Meneruskan prototype
6. Merancang obyek dan kelas yang sesuai
7. Merancang dan menyesuaikan untuk User Interfacenya.

4.Task Management Component (Komponen Tugas Manajemen)

Task Management Component (TMC) adalah kelas dan obyek yang menspesifikasikan komponen sistem operasi, dengan mengimplementasikan sistem. Adapun langkah-langkah dalam menentukan TMC sebagai berikut:

1. Mengidentifikasikan tugas event-driven
2. Mengidentifikasikan tugas clock-driven
3. mengidentifikasikan tugas-tugas yang akan dipakai
4. Mengkoordinir masing-masing tugas
5. Merancang obyek yang sesuai untuk tiap-tiap tugas.

Konsep Object Oriented Pada Visual Basic

1. Obyek

Obyek pada visual basic adalah elemen pada user interface yang dibuat pada form Visual basic melalui kontrol toolbox. Dalam bahasa pemrograman visual telah disediakan banyak obyek, baik obyek visual maupun non visual. Semua obyek merupakan bagian dari pengelompokan dan penjenjangan yang membentuk pohon obyek, dimulai dari obyek dasar **TObject**. **Tobject** merupakan induk (ancestor) dari segala obyek, obyek lain merupakan keturunannya (pewarisnya/inheritance), sehingga akan mewarisi semua sifat induknya. Obyek merupakan kombinasi dari kode dan data yang diperlakukan sebagai satu kesatuan. Suatu obyek mungkin merupakan bagian dari aplikasi seperti suatu kontrol atau suatu form. Sesaat setelah menjadi obyek berdiri sendiri , masing-masing dapat memiliki properti yang berbeda-beda.

Contoh: Object command button

2. Class

Dalam lingkungan pemrograman visual, object atau class ini disebut komponen (Component). Dilihat dari sisi pengembang aplikasi, dengan mengetahui bahwa induk semua komponen visual adalah **Tcomponent** maka obyek akan lebih tepat disebut komponen. Sedangkan jka dilihat dari strukturisasi obyeknya, seperti sifat, fungsi, maka obyek akan lebih tepat disebut **class** (kelas).

2. Properti

Properti adalah nilai atau karakteristik yang dimiliki oleh obyek. Properti menjadi bagian yang paling menarik dari suatu komponen, nilainya dapat dilihat dan diubah-ubah pada tahap perancangan, hasilnya dapat dilihat langsung dari reaksi komponen terhadap perubahan tersebut. Sebuah properti dideklarasikan bersamaan dengan deklarasi kelas komponennya. Ada 3 hal yang perlu ditetapkan daam pendeklarasian auatu properti, yaitu:

a. Nama properti. Memilih nama yang dapat mewakili kegunaan properti tersebut, hindarkan penggunaan nama yang sama dengan nama properti induknya.
b. Tipe properti. Memilih validitas atau tingkat proteksi
c. Metode/Method yang digunakan untuk menulis (write) dan membaca (read) nilai propertinya.

Contohnya: caption, name, font, forecolor, dll

3. Metode (Method)

Metode (method) adalah pernyataan khusus yang melaksanakan suatu tindakan atau layanan untuk obyek tertentu dalam program. Metode adalah perintah yang ingin dipakai untuk mengubah obyek. Komponen sebaiknya tidak mempunyai banyak method dan sesedikit mungkin memanggil method dalam suatu aplikasi, sesuai dengan sifat lingkungan pemrograman yang berbasis visual sedapat mungkin method diubah menjadi properti.

Contoh: List1.Additem "check"

Memakai metode Additem untuk meletakkan kata check dalam object list.

4. Event

Prosedur kejadian, aksi, tindakan atau perbuatan yang akan dilaksanakan jika sebuah obyek dimanipulasi dalam sebuah program.

Contoh: DoubleClick

5. Form

Form merupakan tempat untuk meletakkan obyek/tempat area kerja. Pada form kita dapat menanamkan berbagai Kontrol seperti label, textbox, combobox, listbox, optionbutton, dll.

PENUTUP

Konsep OOP dalam pembuatan maupun dalam penggunaan komponen meliputi konsep pewaris digunakan untuk membuat komponen baru dengan cara menurunkan dari induk dengan maksud agar memiliki sifat dasar obyek dengan tanpa menulis ulang. Dengan memakai ketiga konsep yaitu OOA, OOD dan OOP maka dalam pembuatan program yang bermetode object oriented dapat dikerjakan dengan mudah, sekaligus menjadi tantangan untuk lebih bisa mendalami. Bagi para pemrogram/pengembang, tampilan yang unik diharapkan mampu mendongkrak nilai jual dari aplikasi yang dibuat.

DAFTAR RUJUKAN

Blaha, Michael and Wiliam Premerlani, 1998, ***Object Oriented Modelling and Design for Database Application***, Prentice Hall, New Jersey

Coad, Peter and Yourdon, E., 1991, ***Object-Oriented Analysis***, Second Edition, Prentice Hall, New Jersey

Coad, Peter and Yourdon, E., 1991, ***Object-Oriented Design***, Second Edition, Prentice Hall, New Jersey

Pamungkas, 2001, ***Microsoft Visual basic 6.0***, Elex Media Komputindo, Jakarta

Yourdon, E., 1994, ***Object Oriented System***, Prentice Hall, INC, New Jersey