

# SISTEM PENGELOLAAN KOMPONEN GUNA-ULANG BERBASIS ONTOLOGI

*Lola*

*Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta,  
lola.rezak@gmail.com*

## **Abstrak**

Ontologi merupakan konsep tentang makna dari suatu objek, properti dari suatu objek, serta relasi objek tersebut yang mungkin terjadi pada suatu domain pengetahuan. Ontologi digunakan untuk menerangkan pengetahuan pada suatu domain, untuk guna ulang (*reuse*) dan pengorganisasian komponen.

Untuk melakukan guna-ulang komponen perangkat lunak disediakan pengaksesan komponen berupa pencarian dan pengambilan komponen ke dalam ontologi. Sistem pengelolaan komponen guna-ulang ini dibuat dalam struktur pohon (taksonomi) yang terdiri dari pustaka guna-ulang dan katalog komponen.

Pada makalah ini diberikan prototipe perangkat lunak bantu untuk pengelolaan komponen guna-ulang dengan menggunakan konsep ontologi. Prototipe perangkat lunak bantu sistem pengelolaan komponen guna-ulang ini dibuat tidak berbasis web.

Kata Kunci : ontologi, taksonomi, komponen guna-ulang, pustaka guna-ulang, katalog komponen

## **1. PENDAHULUAN**

Produktivitas dan kualitas merupakan hal yang kritis dalam industri perangkat lunak. Proyek perangkat lunak diharapkan dapat menghasilkan lebih banyak produk dengan sumber daya yang lebih sedikit, waktu yang lebih cepat, *turnaround time* tahap pemeliharaan yang lebih singkat serta keandalan dan keamanan sistem yang lebih tinggi. Pemikiran ini melahirkan kebutuhan akan perubahan dalam proses produksi perangkat lunak.

Peningkatan kualitas dan produktivitas industri perangkat lunak mempunyai tiga tujuan yaitu meningkatkan efektivitas proses, mengurangi jumlah pengulangan pekerjaan, dan mengguna-ulangkan siklus hidup produk (Pressman et al, 2020). Produksi perangkat lunak dengan menggunakan komponen guna-ulang (*reusable component*) merupakan salah satu langkah penting untuk mencapai ketiga tujuan diatas, serta mengurangi biaya dan waktu pembangunan perangkat lunak.

Guna-ulang (*reuse*) merupakan konsep menggunakan sesuatu yang sama lebih dari satu kali, dalam hal ini adalah komponen perangkat lunak. Komponen dalam sebuah perangkat lunak bisa berupa program, bagian dari program, spesifikasi, kebutuhan, arsitektur, kasus uji dan rencana, yang saling berhubungan

atau dengan yang lainnya (Burton et al, 2012). Untuk memudahkan proses guna-ulang komponen perangkat lunak dibutuhkan organisasi dan strukturisasi komponen perangkat lunak yang baik.

Salah satu cara pengorganisasian dan penstrukturan komponen guna-ulang adalah dengan menggunakan sistem ontologi. Ontologi merupakan teori tentang makna dari suatu objek, properti dari suatu objek, serta relasi objek tersebut yang mungkin terjadi pada suatu domain pengetahuan (Gruber, 2015). Ontologi dapat digunakan dalam konteks berbagi pengetahuan dan guna-ulang (*reuse*). Tujuan digunakan ontologi pada proses guna-ulang komponen perangkat lunak ini adalah untuk mempercepat pencarian komponen ke dalam struktur ontologi.

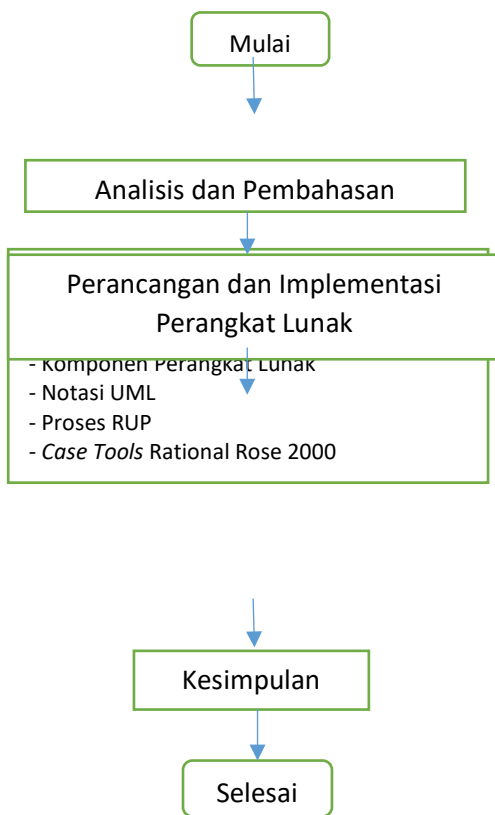
Ontologi secara formal terdiri dari empat komponen utama yaitu konsep (*concepts*), relasi (*relations*), instans (*instances*), dan aksioma (*axioms*) (Gruber, 2015). Konsep merepresentasikan himpunan atau *class* dari entitas atau 'things' pada suatu domain, contohnya : Protein adalah konsep dengan domain biologi molekular. Relasi menggambarkan interaksi antar konsep atau properti konsep, contohnya : *subclass-of* dan *connected-to*. Konsep dan relasi dapat diorganisasikan dalam sebuah taksonomi yaitu

mengorganisasikan konsep ke dalam struktur pohon sub-super konsep. Aksioma digunakan untuk batasan nilai kelas dan instans yang selalu bernilai true, dan instans merupakan objek/individu yang diinstansiasi dari suatu konsep dalam ontologi.

Riset-riset tentang repositori ontologi komponen perangkat lunak telah ada dibuat yaitu repositori ontologi *on-line* untuk perangkat lunak tempelan (*An Online Repository for Embedded Software*)(Gruber, 2013). Dengan mengacu pada riset tersebut, maka pada makalah ini dibuat repositori ontologi untuk perangkat lunak aplikasi tidak terdistribusi.

**2. METODOLOGI**

Metodologi penelitian digambarkan dalam bentuk diagram alir sebagai berikut :



Gambar 1. Diagram Alir Metodologi Penelitian OntoKomp

**3. HASIL DAN PEMBAHASAN**

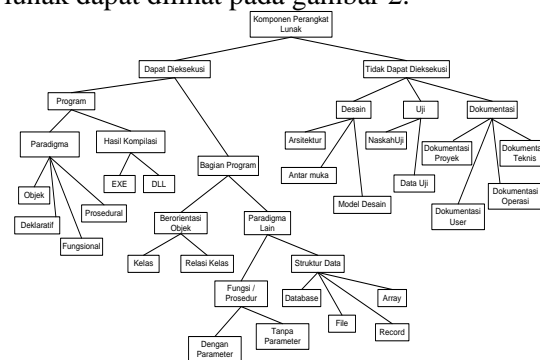
**3.1 HASIL**

Makalah ini bertujuan untuk mengembangkan sebuah aplikasi sistem

pengelolaan komponen guna-ulang berbasis ontologi untuk mendukung pencarian dan pengambilan komponen perangkat lunak secara cepat. Aplikasi ini diberi nama OntoKomp (Ontologi Komponen). Analisis akan dilakukan dengan metode berorientasi objek Jacobson<sup>2</sup>.

Pada makalah ini dijelaskan ontologi kategori komponen perangkat lunak yang dibuat dalam bentuk taksonomi. Komponen perangkat lunak disimpan berdasarkan deskripsi/kategori komponen tersebut. Deskripsi komponen adalah konsep konkrit/abstrak pada ontologi. Untuk guna ulang (pencarian dan pengambilan) komponen pada repositori berbasis ontologi dilakukan pendefinisian komponen yang akan dicari, pemasukan komponen ke repositori, dan pengambilan komponen dari repositori. Repositori (pustaka guna ulang) berbasis ontologi yang dibuat berisikan katalog komponen.

Ontologi dalam bentuk taksonomi berikut menekankan pengelompokkan komponen berdasarkan karakteristik komponen perangkat lunak yaitu dapat dieksekusi atau tidak dapat dieksekusi. Karakteristik komponen dapat dieksekusi dijadikan acuan untuk membedakan komponen perangkat lunak berupa dokumen tekstual biasa. Taksonomi komponen perangkat lunak dapat dilihat pada gambar 2.



Gambar 2. Taksonomi Kategori Komponen Perangkat Lunak

Relasi antar konsep yang digunakan pada ontologi komponen perangkat lunak ini adalah jenis relasi 'is-a' atau 'part-of'. Deskripsi relasi tersebut dapat dilihat pada tabel 1.

Tabel 1. Deskripsi Relasi Pada Ontologi Komponen Perangkat Lunak

Nama Relasi	Nama Konsep_1	Nama Konsep_2
-------------	---------------	---------------

Adalah	Dapat Dieksekusi	Program
Adalah	Dapat Dieksekusi	Bagian Program
Adalah	Program	Paradigma
Adalah	Program	Tipe Hasil Kompilasi
Adalah	Paradigma	Objek
Adalah	Paradigma	Deklaratif
Adalah	Paradigma	Prosedural
Adalah	Paradigma	Fungsional
Adalah	Tipe Hasil Kompilasi	EXE
Adalah	Tipe Hasil Kompilasi	DLL
Adalah	Bagian Program	Berorientasi Objek
Adalah	Bagian Program	Paradigma Lain
BagianDari	Berorientasi Objek	Kelas
BagianDari	Berorientasi Objek	Relasi Kelas
Adalah	Paradigma Lain	Fungsi / Prosedur

Pada makalah ini, ada dua jenis aksioma yang bisa dibuat, yaitu aksioma konsep disjoint dan aksioma relasi. Aksioma relasi yang dapat direalisasikan adalah aksioma simetrik, aksioma antisimetrik dan aksioma transitif. Contoh aksioma relasi dapat dilihat pada tabel 2.

Konsep disjoint hanya berlaku untuk konsep yang bersifat konkrit karena berkaitan dengan pembentukan instans. Pada tabel 3 dapat dilihat aksioma disjoint ontologi komponen perangkat lunak ini.

Tabel 2. Contoh Aksioma Relasi Ontologi Komponen Perangkat Lunak

Nama Aksioma	Nama Relasi_1	Nama Relasi_2
Aksioma Simetrik	Adalah(Dapat Dieksekusi, Program)	Adalah(Dapat Dieksekusi, Bagian Program)
Aksioma Anti Simetrik	Adalah(Dapat Dieksekusi, Program)	Adalah (Program, Paradigma)

Aksioma Transitif	Adalah(Dapat Dieksekusi, Program)	Adalah (Program, Paradigma)
Aksioma Inversi	Adalah(Dapat Dieksekusi, Program)	Adalah (Tidak Dapat Dieksekusi, Desain)

Tabel 3. Contoh Aksioma Disjoin Ontologi Komponen Perangkat Lunak

Nama Aksioma	Nama Konsep_1	Nama Konsep_2
Aksioma Disjoin	Dapat Dieksekusi	Tidak Dapat Dieksekusi
Aksioma Disjoin	Program	Bagian Program

### 3.1.1 Deskripsi Komponen

Deskripsi komponen berupa deskripsi formal yang dihasilkan dengan mengikuti metoda tertentu sesuai dengan metoda klasifikasi yang digunakan. Untuk mendeskripsikan komponen digunakan dua pendekatan yaitu :

1. Membuat struktur katalog secara umum.
2. Membuat struktur katalog berdasarkan model yang diwakili komponen.

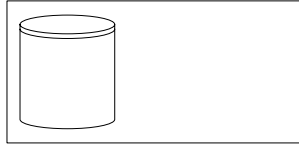
### 3.1.2 Tahap Guna-Ulang

Guna-ulang komponen memerlukan tiga tahap yaitu:

1. Definisi (*definition*) yaitu mendeskripsikan komponen yang diperlukan (*target*) dalam sebuah pustaka guna ulang (*repository*).
2. Pengambilan (*retrieval*) yaitu mencari deskripsi komponen yang cocok dengan target, menyusun list kandidat (komponen yang mirip dengan target), memilih salah satu kandidat, dan mengambil komponen yang dipilih dari pustaka guna-ulang.
3. Memasukkan (*incorporation*) yaitu mengambil komponen yang baru dikonstruksi, menyisipkan deskripsi komponen ke dalam katalog, dan menyimpan komponen ke dalam pustaka guna-ulang.

### 3.1.3 Pustaka Guna Ulang Komponen Perangkat Lunak

Pustaka guna ulang komponen berisi katalog komponen. Katalog komponen merupakan basis data yang berisi data petunjuk untuk mengakses komponen guna-ulang pada pustaka guna-ulang.



Gambar 3. Pustaka Guna Ulang Komponen Perangkat Lunak

### 3.1.4 Struktur Katalog Komponen

Komponen dapat dideskripsikan dengan memperhatikan karakteristik komponen tersebut, yaitu memiliki :

- a. *Interface* : layanan-layanan fungsional yang disediakan oleh komponen agar dapat berinteraksi dengan komponen lain.
- b. *Contract* : spesifikasi yang lebih akurat dari kelakuan komponen.

```

Component SpellChecker =
  Interface = -ISpellCheck

  Contract      =
    pre: word <> ""
    post:
      SUCCEDED(result)implies correct=word-
        >includes(word)

End component
    
```

Gambar 4. Contoh Deskripsi Umum Komponen

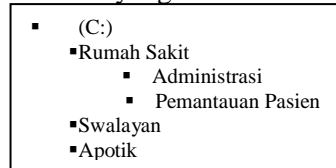
Pada deskripsi komponen ini digunakan atribut tambahan pada *interface* komponen untuk menyimpan tipe dan kelakuan komponen. Deskriptor ini menyimpan deskripsi sebuah rutin yang ditulis dalam bahasa PASCAL bernama *SearchInFile* yang berfungsi untuk mencari sebuah string tertentu pada sebuah file. Contoh spesifikasi rutin tersebut adalah sebagai berikut :

```

function SearcherInFile (S:string):boolean;
{ mengembalikan TRUE jika string S
  ditemukan pada file dan FALSE jika
  tidak ketemu }
    
```

### 3.1.5 Struktur Pustaka Guna-Ulang

Pustaka guna-ulang merupakan sebuah pustaka virtual yang menyimpan kumpulan komponen yang disimpan secara terorganisir menurut klasifikasi/kategori tertentu. Penyimpanan komponen secara fisik dilakukan di dalam media penyimpanan komputer berupa kumpulan file yang diletakkan di dalam susunan direktori yang teratur.

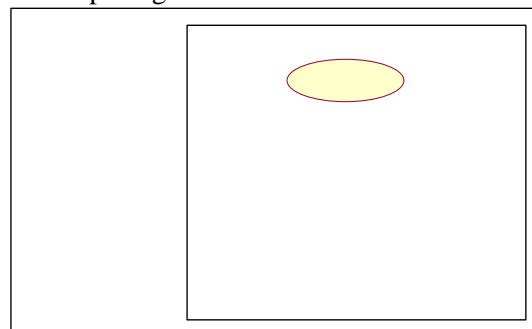


Gambar 5. Struktur Pustaka Guna Ulang

## 3.2 PEMBAHASAN

### 3.2.1. Model Kebutuhan

Model kebutuhan menjelaskan kebutuhan fungsionalitas masalah dalam bentuk *use-case*. Dalam hal ini dijelaskan bahwa pada sistem terdapat *actor* sebagai pelaku sistem dan *use-case* sebagai fungsi-fungsi sistem. *Actor* sistem pengelolaan komponen guna-ulang ini terdiri dari pemodel dan pemakai akhir. Sedangkan *use-case* sistemnya terdiri dari pembuatan ontologi komponen guna-ulang, ambil komponen guna-ulang dan evaluasi fleksibilitas komponen. Model kebutuhan sistem pengelolaan komponen guna-ulang ini dapat dilihat pada gambar 6 berikut.

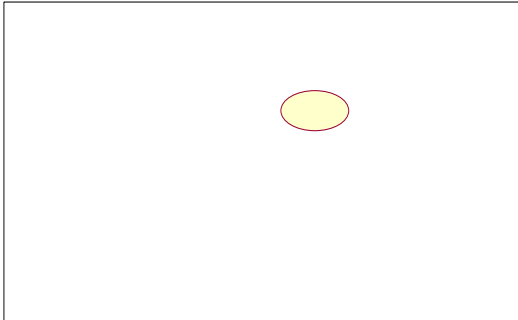


Gambar 6. Diagram Kebutuhan

### 3.2.2 Model Use Case

Model *use-case* menjelaskan interaksi fungsi-fungsi sistem yang ada dengan pelaku sistem (*actor*). Pada sistem pengelolaan komponen guna-ulang (OntoKomp) ini *actor* pemodel bertugas untuk melakukan fungsi pembuatan ontologi komponen guna-ulang,

sedangkan *actor* pemakai akhir bertugas untuk melakukan fungsi ambil dan definisi komponen guna-ulang. Model *use-case* OntoKomp ini dapat dilihat pada gambar 7 berikut.



Gambar 7. Model *Use Case*

### 3.2.3 Objek Domain Persoalan (*Problem Domain Object*)

Diagram objek domain persoalan digunakan untuk mendefinisikan objek-objek yang mempunyai peran aktif dalam lingkungan aplikasi dan harus diketahui oleh sistem. Objek-objek tersebut untuk aplikasi OntoKomp dapat dilihat pada gambar 8.



Gambar 8. Objek Domain Persoalan

### 3.2.4 Diagram Interaksi

Dengan mengacu kepada model kebutuhan yang telah dibuat pada tahap analisis, maka dibuatlah diagram interaksi untuk tiap use case dan objek domain persoalan. Diagram ini menggambarkan aliran perilaku tiap use case sesuai dengan deskripsinya. Interaksi model use case dan diagram interaksi OntoKomp ini dapat dilihat pada gambar 9 berikut :

Gambar 9. Interaksi Model *Use Case* dan Diagram Interaksi *OntoKomp*

### 3.2.5 Arsitektur *OntoKomp*



Gambar 10. Arsitektur *OntoKomp*

### 3.2.6 Perancangan Layar

Perancangan layar ditujukan untuk merancang antarmuka kepada pengguna dalam aplikasi. Perancangan layar didasarkan objek-objek antarmuka yang telah dibuat pada pembuatan model kebutuhan dengan beberapa penyesuaian dengan lingkungan implementasi.

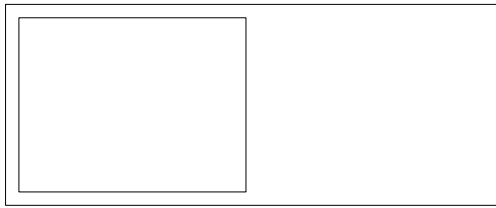
#### 1. Panel Menu



Gambar 11. Rancangan Panel Menu

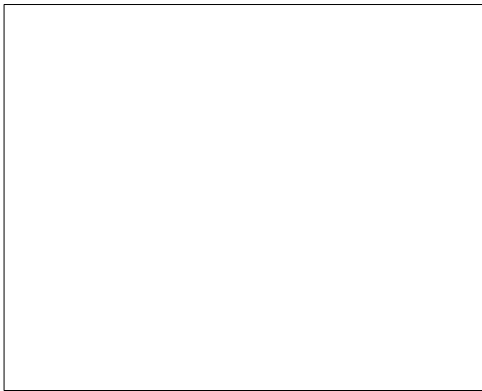
#### 2. Panel Pembuatan Ontologi

Pembuatan ontologi secara keseluruhan ditangani di Panel Pembuatan Ontologi. Panel ini merupakan suatu panel bertab yang memiliki beberapa panel anak tempat pengguna dapat membangun objek-objek ontologi yang diinginkannya.



Gambar 12. Rancangan Panel Pembuatan Ontologi

### 3. Jendela Guna Ulang Komponen Perangkat Lunak



Gambar 13. Rancangan Jendela Guna Ulang Komponen Perangkat Lunak

## 4. KESIMPULAN

Kesimpulan-kesimpulan yang didapat adalah:

1. Pada makalah ini telah dibuat sebuah prototipe perangkat lunak bantu pengelolaan komponen guna-ulang berbasis ontologi (OntoKomp) yang dapat berkembang. Pengelolaan komponen guna-ulang berbasis ontologi ini dibangun dengan menggunakan metodologi berorientasi objek, notasi UML, proses RUP dan penggunaan *CASE Tools Rational Rose 2000*. Perangkat lunak bantu ini membantu pengelolaan hasil proses setiap tahap pengembangan perangkat lunak ke dalam suatu pustaka komponen perangkat lunak, dan meningkatkan *reusability* komponen, sehingga mempercepat proses dokumentasi dan implementasi dari proses pengembangan perangkat lunak.
2. OntoKomp ini dirancang dapat berkembang dengan membuat dan menyimpan model deskripsi komponen pada sebuah basis data khusus yang

dinamakan basis data Model Deskripsi Komponen.

3. OntoKomp menyediakan fasilitas untuk menyimpan dan mengambil kembali komponen guna-ulang tanpa bergantung jenisnya dengan syarat model deskripsi dan deskripsi komponen sudah terdefinisi. Dalam hal ini komponen guna-ulang tersebut berada pada hierarki konsep ontologi yaitu pada konsep abstrak.
4. OntoKomp menerima query berupa nama komponen dalam format teks standar. Query dibutuhkan oleh pengguna untuk melakukan pencarian komponen pada ontologi. Pencarian ini dinamakan pencarian tematik yaitu pencarian berdasarkan suatu tema, dalam hal ini temanya adalah Kategori Komponen Perangkat Lunak.
5. Pemakai sistem dibagi atas dua kelompok yaitu pemodel dan pengguna akhir. Kedua pemakai diberi hak akses yang berbeda terhadap OntoKomp sesuai dengan peran masing-masing, serta untuk menjaga keamanan sistem.

## DAFTAR PUSTAKA

- Burton, B.A., Aragon, R.W., Bailey, S.A., Koehler, K.D., & Mayes, L.A. (2012). The Reusable Software Library, *IEEE Transaction on Software Engineering and Methodology*, (July 2012), 25-33.
- Gruber, T.R. (2013). *A Translation Approach to Portable Ontology Specifications Knowledge Acquisition*, Sixth Edition, 199-220.
- Gruber, T.R. (2015). Towards principles for the design of ontologies used for knowledge sharing, *International Journal of Human-Computer Studies*, 907-928.
- Pressman, R.S., & Maxim, B.R. (2020). *Software Engineering : A Practitioner's Approach*, Ninth Edition, Mc Graw-Hill. Education.