



## MODUL 2 - WhiteStartUML Design Tools

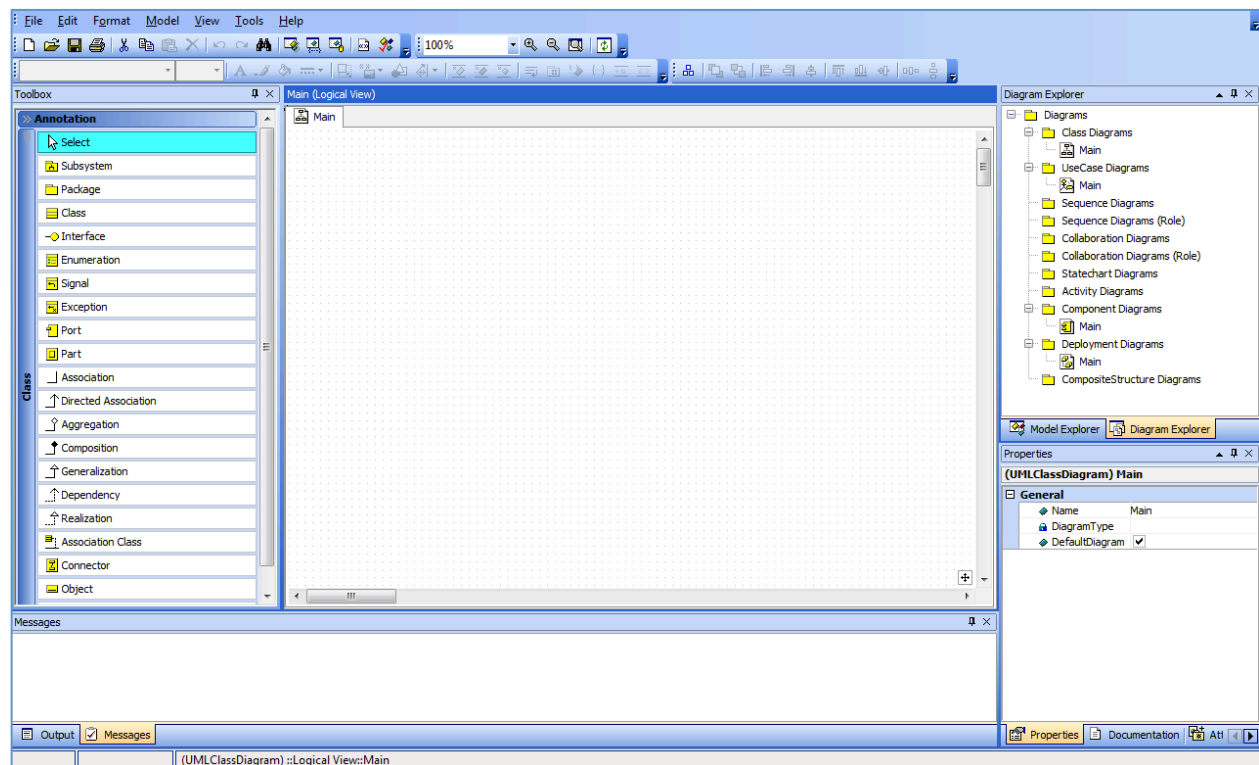
RKPPL  
TEKNIK INFORMATIKA

UNIVERSITAS PASUNDAN  
BANDUNG 2014/2015

**WhiteStarUml** adalah sebuah perangkat lunak pemodelan yang berasal dari StarUML yang mendukung UML (Unified Modeling Language). Didasarkan pada UML versi 1.4 dan menyediakan sebelas jenis diagram, dan dapat menerima dengan notasi UML 2.0.

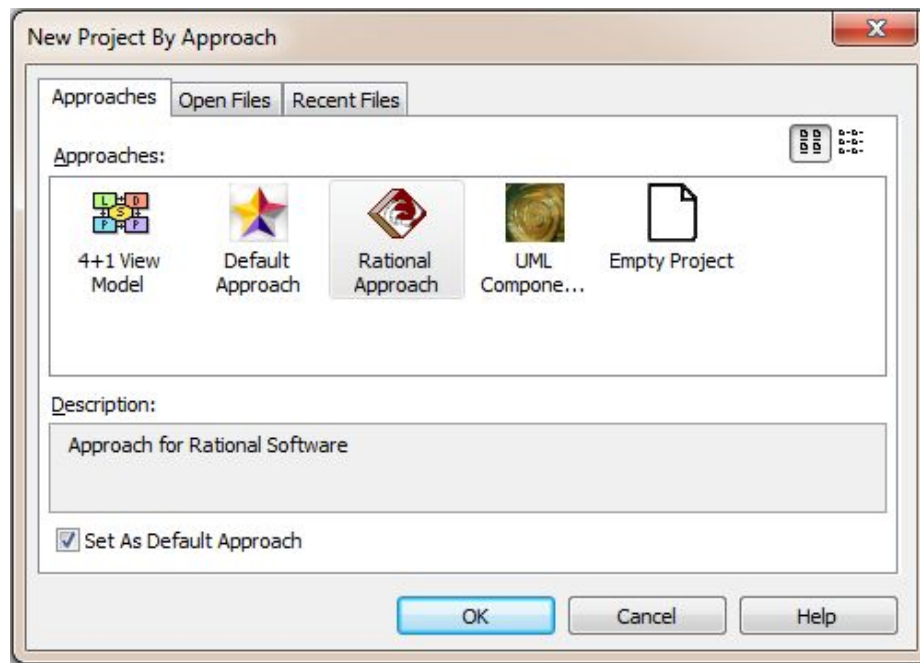
## Fitur

Feature	Description
Accurate UML standard model	WhiteStarUml strictly adheres to the UML standard specification specified by the OMG for software modeling. Considering the fact that the results of design information can reach 10 years or more into the future, dependence on vendor-specific irregular UML syntax and semantics can be quite risky. WhiteStarUml maximizes itself to order UML 1.4 standard and meaning, and it accepts UML 2.0 notation on the basis of robust meta model.
Open software model format	Unlike many existing products that manage their own legacy format models inefficiently, WhiteStarUml manages all files in the standard XML format. Codes written in easy-to-read structures and their formats can be changed conveniently by using the XML parser. Given the fact that XML is a world standard, this is certainly a great advantage, ensuring that the software models remain useful for more than a decade.
True MDA support	WhiteStarUml truly supports UML Profile. This maximizes extensibility of UML, making modeling of applications possible even in areas like finance, defense, e-business, insurance, and aeronautics. Truly Platform Independent Models (PIM) can be created, and Platform Specific Model (PSM) and executable codes can be automatically generated in any way.
Applicability of methodologies and platforms	WhiteStarUml manipulates the approach concept, creating environments that adapt to any methodologies/processes. Not only the application framework models for platforms like .NET and J2EE, but also basic structures of software models (e.g. 4+1 view-model, etc.) can be defined easily
Excellent extensibility	All functions of the WhiteStarUml tools are automated according to Microsoft COM. Any language which supports COM (Visual Basic Script, Java Script, VB, Delphi, C++, C#, VB.NET, Python, etc.) can be used to control WhiteStarUml or develop integrated Add-In elements.
Software model verification function	Users can make many mistakes during software modeling. Such mistakes can be very costly if left uncorrected until the final coding stage. In order to prevent this problem, WhiteStarUml automatically verifies the software model developed by the user, facilitating early discovery of errors, and allowing more faultless and complete software development.
Useful Add-Ins	WhiteStarUml includes many useful Add-Ins with various functionalities: it generates source codes in programming languages and converts source codes into models, imports Rational Rose files, exchanges model information with other tools using XML, and supports design patterns. These Add-Ins offer additional reusability, productivity, flexibility and interoperability for the modeling information.



## Project

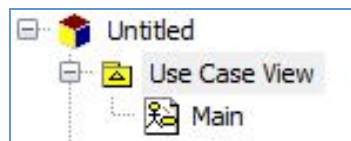
Saat menjalankan WhiteStarUML, akan muncul jendela 'Project Wizard' yang menanyakan tentang pendekatan/approach yang akan digunakan untuk project saat ini.



Gambar 1. Project Wizard

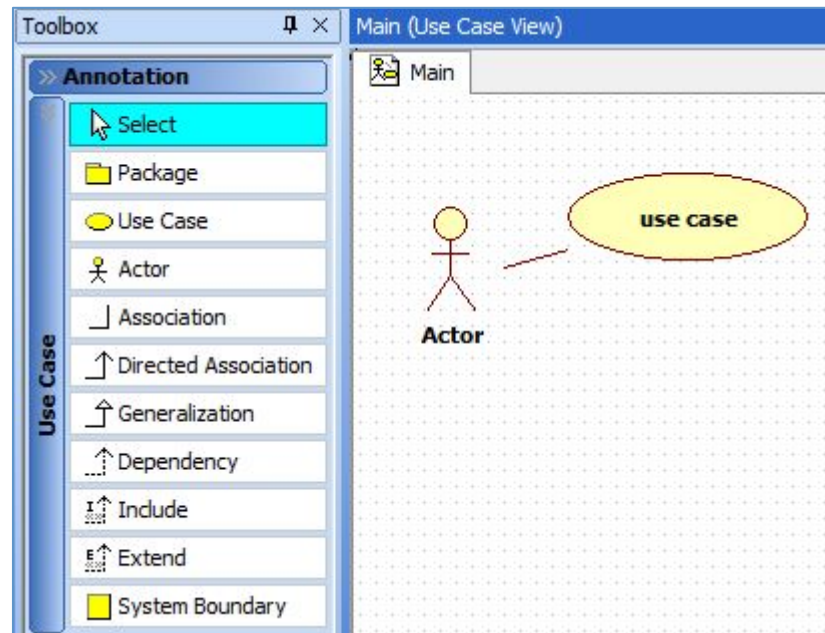
## Diagram Use Case

- Untuk menggambar diagram use case, klik dua kali Use Case View.
- Gambar use case menggunakan diagram standar dengan nama 'Main' atau dapat menambahkan diagram baru dengan mengklik kanan Use Case View dan kemudian pilih [Add Diagram] \_ [Use Case Diagram].



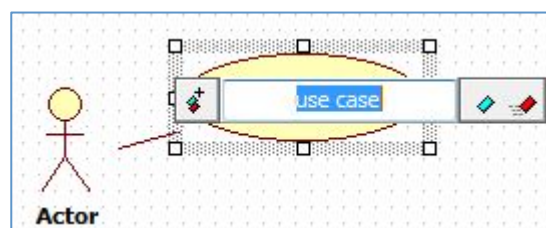
- Menggambar notasi pada diagram
  - Klik dua kali diagram use case
  - Gunakan notasi diagram dari Toolbox yang ada di sebelah kiri.

- Untuk menggambar elemen diagram (misalnya, use case), klik elemen diagram elemen dalam panel Use Case (di Toolbox) dan klik pada posisi yang diinginkan di kanvas.



*Gambar 2. Membuat diagram use case*

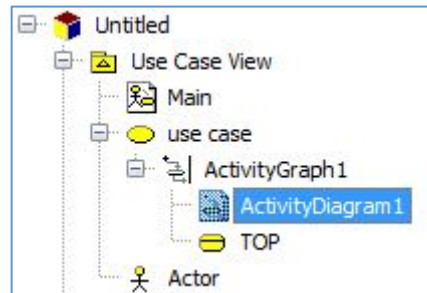
- Membuat diagram use case Mengganti nama elemen diagram
  - Klik dua kali pada elemen yang akan diganti nama
  - Ketik nama elemen diagram pada textbox yang sudah disediakan



*Gambar 3. Mengganti nama elemen*

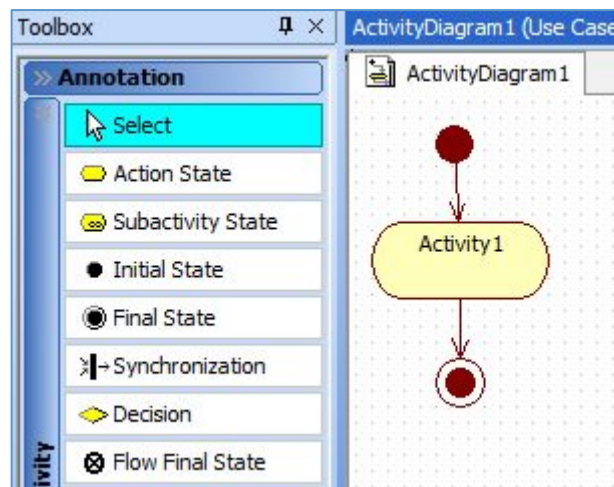
## Diagram Activity

- Untuk menggambar diagram activity, klik kanan pada use case yang dipilih dan kemudian pilih [Add Diagram] \_ [Activity Diagram].



Gambar 4. Diagram Activity

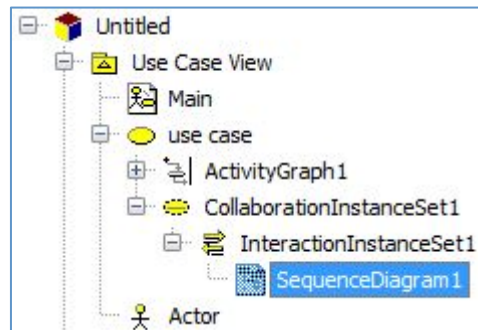
- Klik dua kali diagram activity
  - Gunakan notasi diagram dari Toolbox yang ada di sebelah kiri.
  - Untuk menggambar elemen diagram (misalnya, activity), klik elemen diagram elemen dalam panel Activity (di Toolbox) dan klik pada posisi yang diinginkan di kanvas.



Gambar 5. Membuat diagram activity

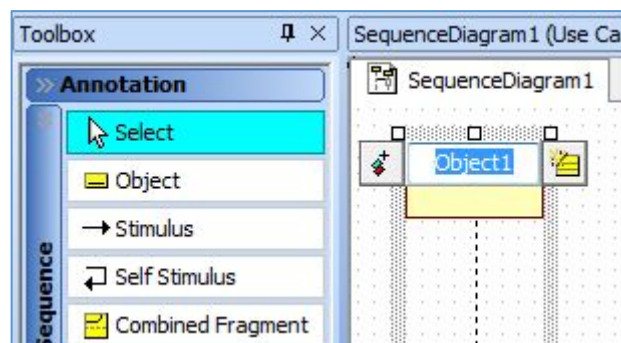
## Sequence Diagram

- Untuk menggambar diagram sequence, klik kanan pada use case yang dipilih dan kemudian pilih [Add Diagram] \_ [Sequence Diagram].



Gambar 6. Diagram Sequence

- Menggambar obyek
  - Klik dua kali diagram sequence
  - Untuk menambahkan objek baru dalam diagram, klik tombol Object di Toolbox dan klik di posisi apapun di kanvas.
  - Untuk menambahkan kelas obyek ini untuk model, klik dua kali objek dan klik class figure di pane sebelah kanan objek.
  - Memasukkan nama kelas, lalu klik OK dan kelas baru akan dibuat di Model Explorer.

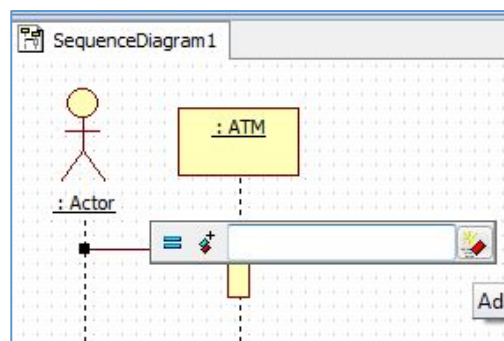


Gambar 7. Membuat diagram activity

- Untuk menambahkan objek baru dari kelas yang ada, lakukan drag dan drop class ke

kanvas.

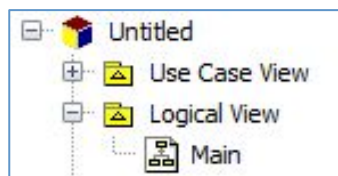
- Untuk menambahkan objek dari satu objek ke yang lain, pilih stimulus tombol di Toolbox dan tarik dari objek sumber ke objek tujuan.
- klik tanda '=' untuk merujuk pesan ke operasi kelas yang ada
- Jika operasi belum ada dan hendak menambahkan ke dalam kelas, klik symbol persegi panjang merah di sisi kanan. Selanjutnya, tentukan nama operasi dan, jika perlu, visibilitas. Setelah melakukan langkah ini, operasi akan ditambahkan ke kelas yang sesuai. Perhatikan bahwa prosedur ini hanya akan mengeksekusi jika kelas sudah ada dalam model



Gambar 8. Menambah Object Sequence

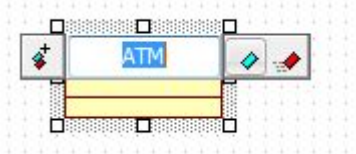
## Class Diagram

- Untuk membuat diagram kelas, dapat menggambar diagram elemen dalam diagram 'Main' pada Logical view atau membuat diagram kelas baru.
- Untuk membuat diagram kelas baru, klik kanan Logical View dan pilih [Add Diagram] \_ [Class Diagram].
- Untuk mulai membuat elemen diagram, double-klik diagram kelas yang diinginkan dalam model explorer



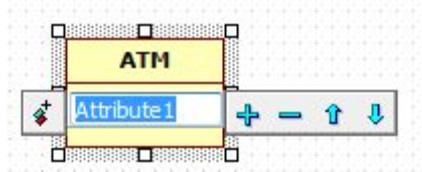
Gambar 9. Diagram class

- Pilih elemen Kelas dalam toolbox (panel kiri) dan klik pada posisi yang diinginkan pada kanvas.
- Untuk menambahkan atribut ke kelas, klik dua kali kelas dan pilih symbol persegi panjang biru yang muncul di sisi kanan dari kelas.
- Untuk menambahkan operasi untuk kelas, klik dua kali kelas dan pilih persegi panjang merah yang muncul di sisi kanan dari kelas.



*Gambar 10. Menambahkan atribut dan operasi kelas*

- Untuk mengedit atau menghapus atribut kelas / operasi, klik dua kali atribut atau operasi yang Anda ingin memodifikasi / menghapus. Untuk mengubah visibilitas pilih panel kiri kelas atau untuk menambah, menghapus, atau menyusun ulang atribut / operasi pilih panel kanan kelas



*Gambar 11. Edit/hapus atribut dan operasi kelas*



## Latihan

- Buatlah Project WhiteStartUML menggunakan Dokumen Design yang dapat di download di Source
- Ekplorasi fitur pada WhiteStartUML untuk generate model diagram menjadi dokumen .DOC

## Tugas

Buatlah Tutorial menggunakan tools WhiteStartUML atau StartUML dengan menjalankan fitur

- Export diagram kelas ke kode program (generate)
- Import kode program menjadi diagram (reverse engineering)

Kumpulkan tugas via E-Mail pada Asisten dengan nama file dan subject:

- **[RKPPL]-TUGAS1-NRP-SHIFT**
- contoh **[RKPPL]-TUGAS1-123040199-SABTU-07**

## Reference

- <https://www.scribd.com/doc/235383396/Pengantar-OOAD-UML-WhiteStarUML-2>  
diakses pada 20 February 2015