

Modul Analisis Dan Evaluasi Sistem Informasi (Kode: MIK 452)

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

**Dosen Pengampu:
Tria Saras Pertiwi, SKM., MPH**

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

**Manajemen Informasi Kesehatan
Universitas Esa Unggul**

2018

Daftar Isi

				Halaman
Modul 1	Universitas Esa Unggul	Universitas Esa Unggul	Universitas Esa Unggul	1
Modul 2				26



Modul 1

UML (*UNIFIED MODELING LANGUAGE*)

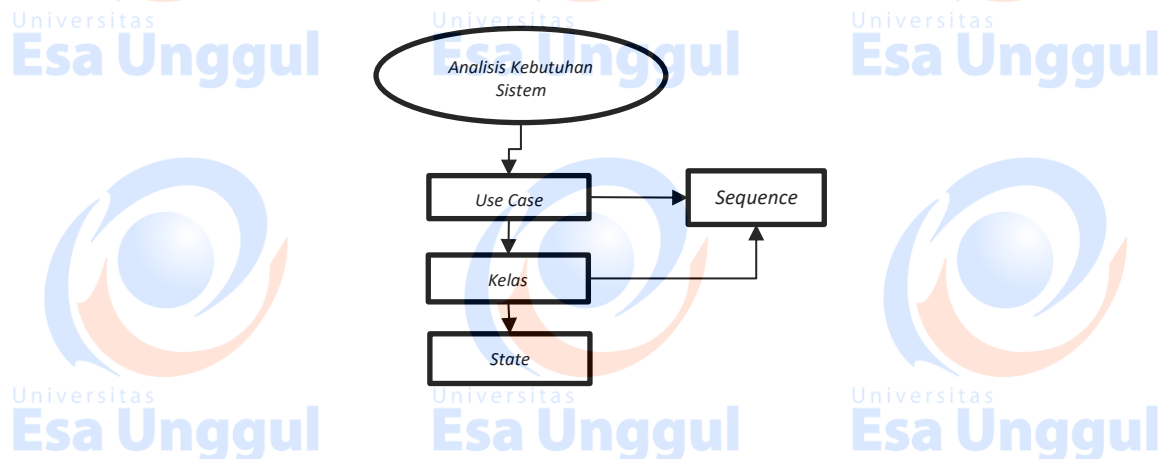
Modul pertemuan ini berisi tentang Dasar teori UML, Jenis-jenis diagram UML serta latihan membuat diagram *use case*, kelas, *sequence*, state dengan aplikasi ArgoUML.

Pada modul ini diharapkan mahasiswa mampu membuat *use case*, memahami komponen-komponen yang terlibat dalam *use case diagram*, memahami hubungan antara *actor* dengan *use case diagram*, dan mampu membuat *use case diagram* dari skenario yang telah ada.



Topik 1

UML (*UNIFIED MODELING LANGUAGE*)



Perancangan dan pembangunan aplikasi perangkat lunak berbasis objek **Object Oriented Analysis & Design (OOAD)** menganggap segala sesuatunya adalah objek serta sistem dipandang sebagai interaksi dari banyak objek dimodelkan menggunakan *Unified Modelling Language (UML)*.

Unified Modelling Language (UML) merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk pembangunan perangkat lunak berbasis objek. UML memiliki banyak diagram, diantaranya **Use Case Diagram, Class Diagram, Sequence Diagram & State Diagram**.

Saat spesialis informasi memilih membuat pemodelan arsitektur dengan pendekatan struktur (membuat diagram kontkes & Diagram Alir Data), maka pemodelan arsitektur berbasis objek (membuat diagram-diagram UML) tidak lagi digunakan, demikian pula sebaliknya. UML ditujukan untuk mendapatkan gambaran arsitektur sistem informasi dengan sudut pandang yang berbeda, hal ini terpaut pada faktor kebutuhan. Kebutuhan sistem adalah fungsionalitas apa yang mesti disediakan oleh sistem, apakah didokumentasikan pada model *use case* yang menggambarkan fungsi sistem yang diharapkan (*use case*), yang mengelilinginya (*actor*) dan hubungan antara *actor* dengan *use case* (*use case diagram*). Untuk lebih jelasnya, berikut urutan pembuatan diagram dengan UML.

DIAGRAM USE CASE

Use case adalah konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna potensial. *Use-case* terdiri dari sekumpulan skenario yang dilakukan oleh seorang aktor (orang, perangkat keras, urutan waktu atau sistem yang lain). Sedangkan **use-case diagram memfasilitasi komunikasi di antara analis dan pengguna serta diantara analis dan klien**.

Diagram use case harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan. Diagram use case akan menggambarkan apa yang

dikerjakan aktor. Untuk menggambar diagram use case mengacu pada proses sebelumnya, yaitu analisis kebutuhan sistem atau analisis kebutuhan pada RPL (rekayasa perangkat lunak).

Langkah-langkah yang dibutuhkan untuk menyusun diagram use case adalah:

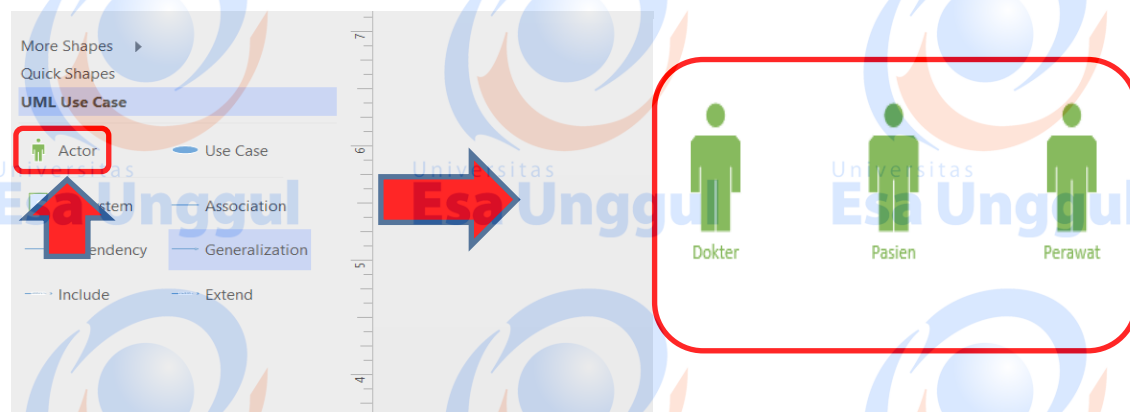
1. Mengidentifikasi pelaku bisnis
2. Mengidentifikasi use case persyaratan bisnis
3. Membuat diagram model use case
4. Mendokumentasikan naratif use case persyaratan bisnis

1. Komponen-Komponen Yang Terlibat Dalam Use Case Diagram

a. Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor* dimana *actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem, **actor hanya berinteraksi dengan use case tetapi tidak memiliki kontrol atas use case.**

Actor digambarkan dengan *stick man* dan penamaan aktor menggunakan kata benda. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*. Simbol *Actor* dan contoh relasi gambaran *Actor* pada Ms. Visio.



Ada beberapa kemungkinan yang menyebabkan *actor* tersebut terkait dengan sistem antara lain:

- Yang berkepentingan terhadap sistem dimana adanya arus informasi baik yang diterimanya maupun yang dia inputkan ke sistem.
- Orang ataupun pihak yang akan mengelola sistem tersebut.
- *External resource* yang digunakan oleh sistem.
- Sistem lain yang berinteraksi dengan sistem yang akan dibuat.

b. Use Case

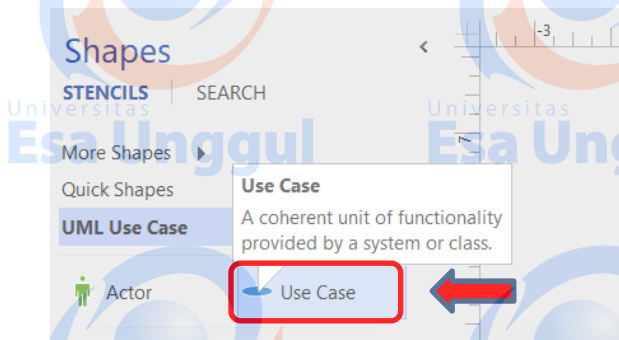
Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun dan penamaan use case menggunakan kata kerja.

Catatan:

- Use case diagram adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (user), sehingga pembuatan use case lebih dititikberatkan pada

- fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.

Simbol use case dalam visio:

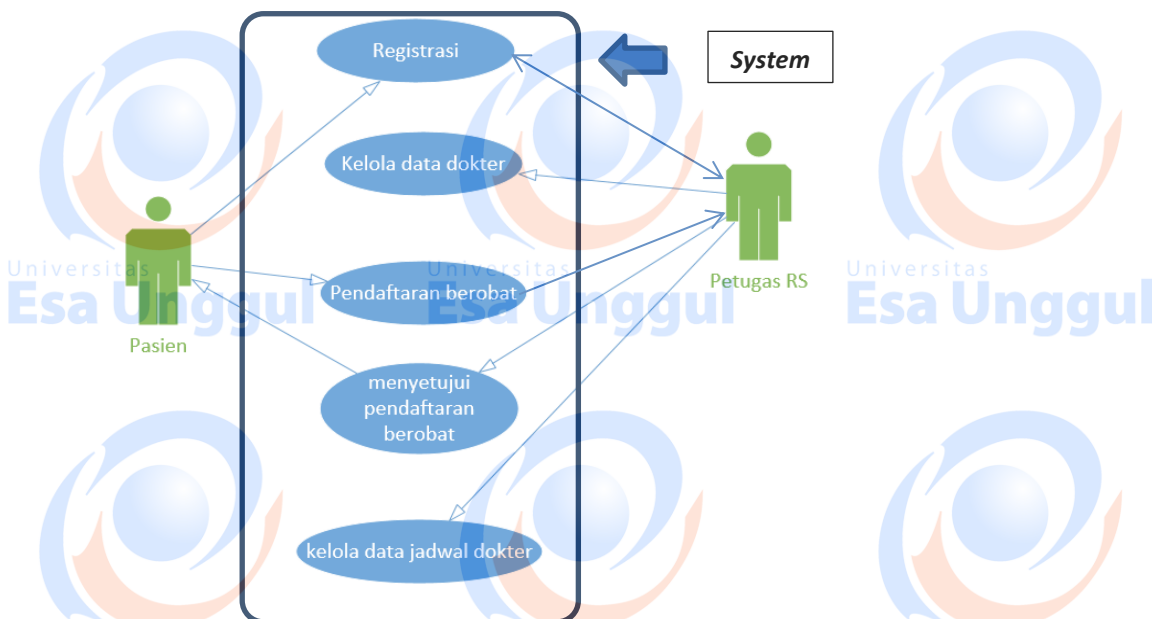


Cara menentukan Use Case dalam suatu sistem:

- Pola perilaku perangkat lunak aplikasi.
- Gambaran tugas dari sebuah actor.
- Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada actor.
- Apa yang dikerjakan oleh suatu perangkat lunak (bukan bagaimana cara mengerjakannya).

c. System

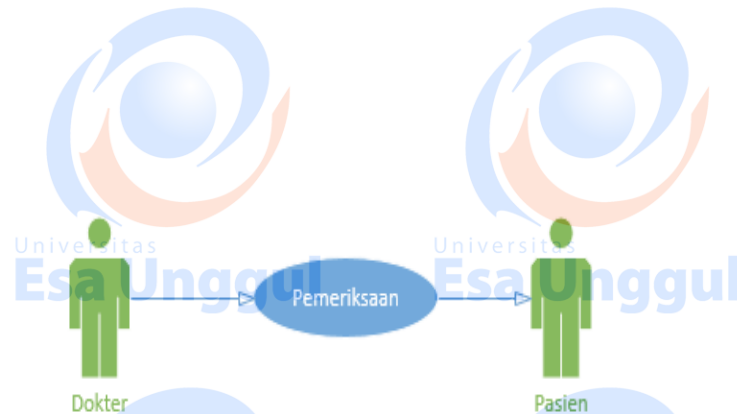
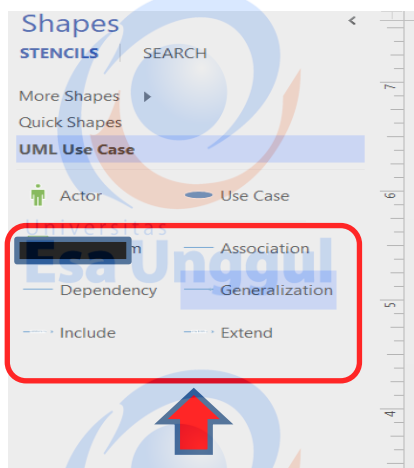
System yaitu menyatakan batasan sistem dalam relasi dengan actor-actor yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Digambarkan dengan segiempat yang membatasi semua use case dalam sistem terhadap pihak mana sistem akan berinteraksi. Sistem disertai label yang menyebutkan nama dari sistem, tapi umumnya tidak digambarkan karena tidak terlalu memberi arti tambahan pada diagram.



d. Relasi dalam Use Case

- **Association**, menghubungkan link antar element (hubungan antar aktor dan use case).
- **Generalization**, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
- **Dependency**, sebuah element bergantung dalam beberapa cara ke element lainnya.
- **Aggregation**, bentuk assosiation dimana sebuah elemen berisi elemen lainnya.

Relasi dalam use case (contoh menggunakan visio)



Extend

Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri meski tanpa use case tambahan itu arah panah mengarah pada use case yang ditambahkan. Extend menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

Include

Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan membutuhkan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini arah panah include mengarah pada use case yang dipakai (dibutuhkan) atau mengarah pada use case tambahan. Include menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.

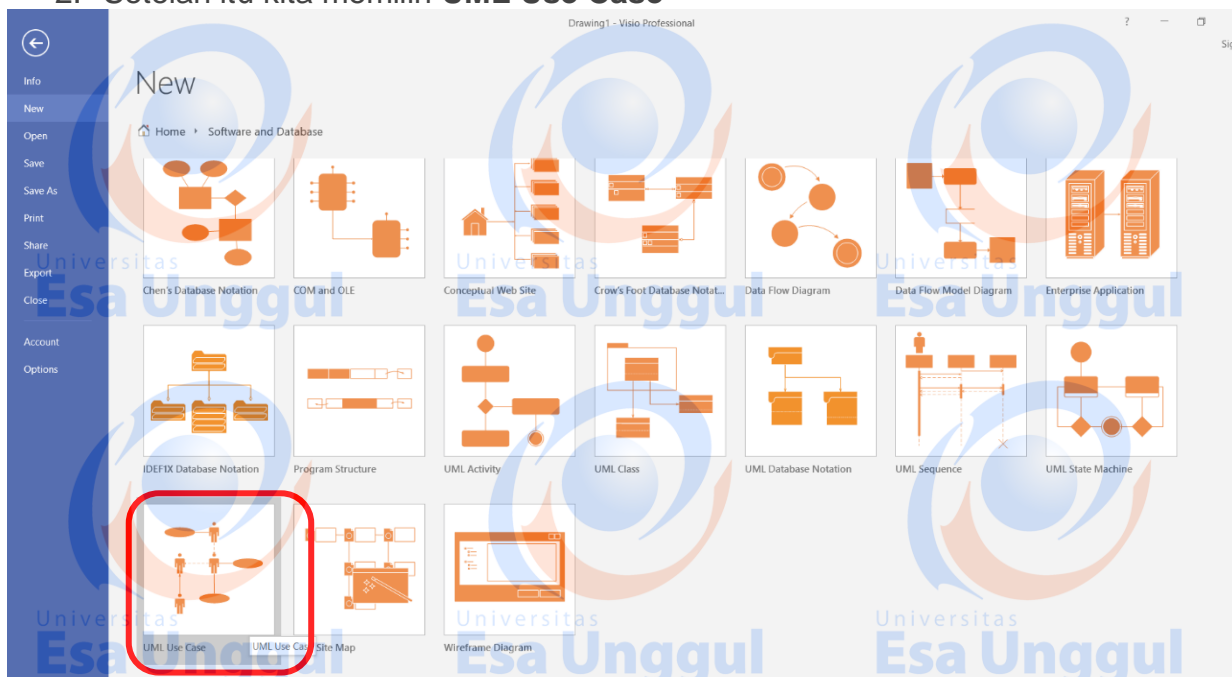
2. Menggambar Use Case diagram

Use case diagram adalah gambaran *graphical* dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. Use case diagram biasanya untuk mempermudah kita dalam mengingat fitur-fitur yang ada pada aplikasi tersebut. Visio (dalam contoh modul ini yang digunakan yaitu Microsoft Visio 2016) adalah aplikasi diagram yang memungkinkan kita bekerja secara visual untuk membuat segala macam diagram. Dalam pembuatan *use case* yang berada dalam Microsoft Visio ini terdapat beberapa Shapes yang membantu kita untuk membuat *use case*.

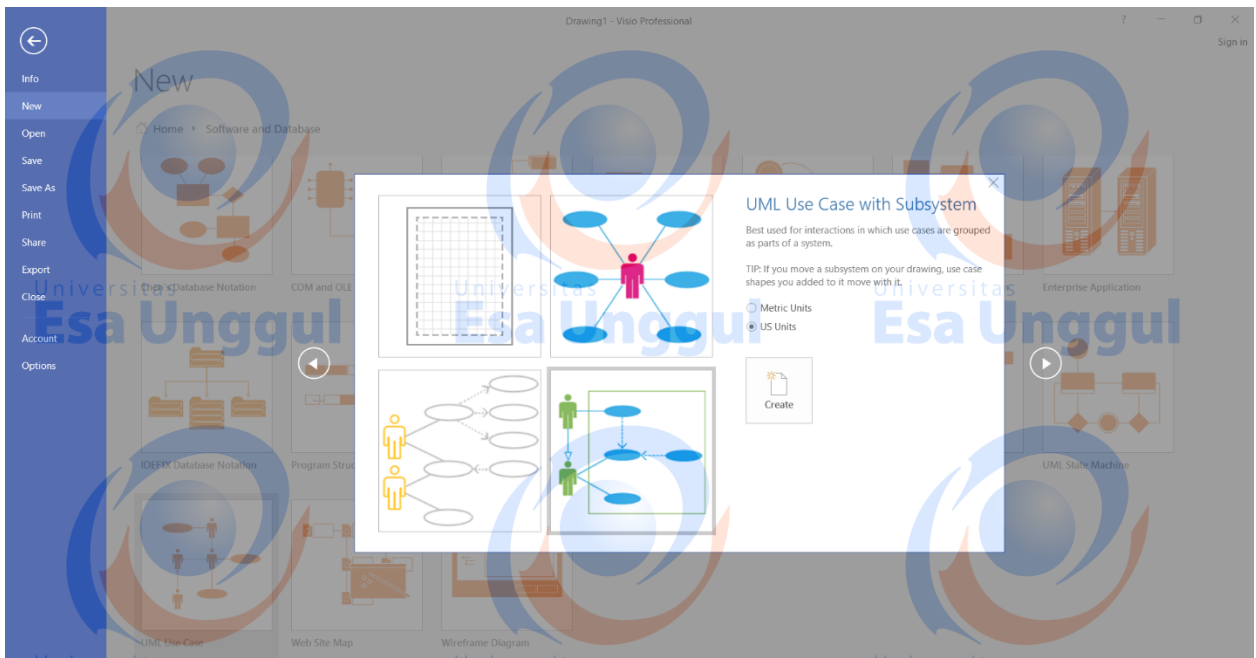
1. Pertama membuka aplikasi visio dan tampilannya seperti gambar di bawah ini, kita dapat memilih berbagai kategori diagram, untuk melanjutkan memilih jenis-jenis diagram UML dapat memilih diagram untuk **Software And Database**.



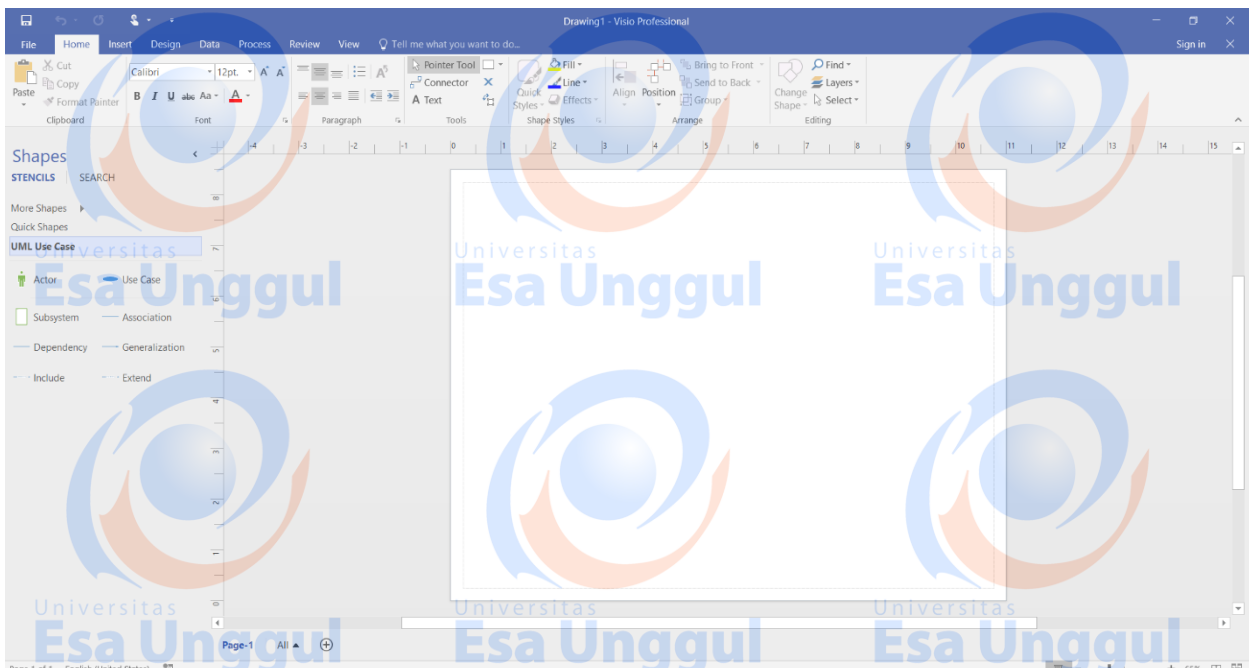
2. Setelah itu kita memilih **UML Use Case**



3. Setelah memilih **UML Use case**, kita dapat menggunakan template yang ada seperti di bawah ini atau menggunakan template yang kosong dan kemudian pilih **create**



- Setelah dipilih maka akan muncul tampilan seperti di bawah ini jika kita memilih salah satu template yang disediakan kita dapat langsung mengganti nama.



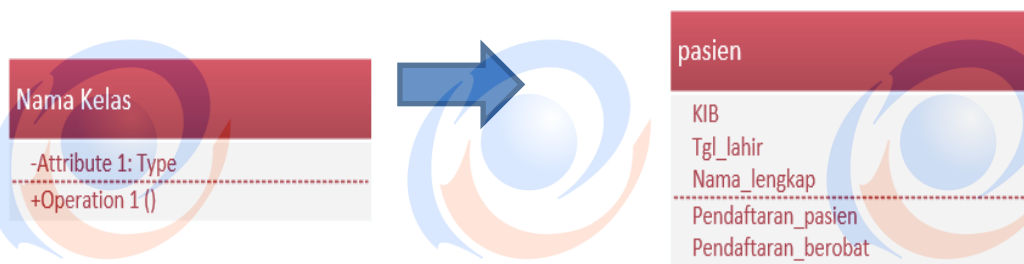
- Pada Tab **UML Use Case (Metric)** terdapat *Shapes* yang sudah dikatakan sebelumnya, dan *shapes* yang sudah ada bisa di letakkan di *Paper* yang sudah tersedia. dan sisanya tinggal menghubungkan satu shape dengan shape lainnya tergantung kebutuhan sistem yang akan kalian buat.

DIAGRAM KELAS

Diagram kelas dibuat setelah diagram use case dibuat terlebih dahulu. Pada pembuatan diagram ini harus menjelaskan **hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuklah suatu system aplikasi**. Kelas adalah rancangan dari suatu objek, sebagai contoh kelas mobil merupakan rancangan dari objek mobil sedan, *pick up*, mini bus, truk, dan lain-lain. Jadi di sini kelas mobil adalah rancangan saja, belum menjadi objek, belum bisa dikendarai, dan belum bisa dioperasikan. Sebaliknya, jika rancangan mobil tersebut sudah menjadi objek nyata, mobil sedan misalnya, objek tersebut barulah bisa dikendarai dan dioperasikan.

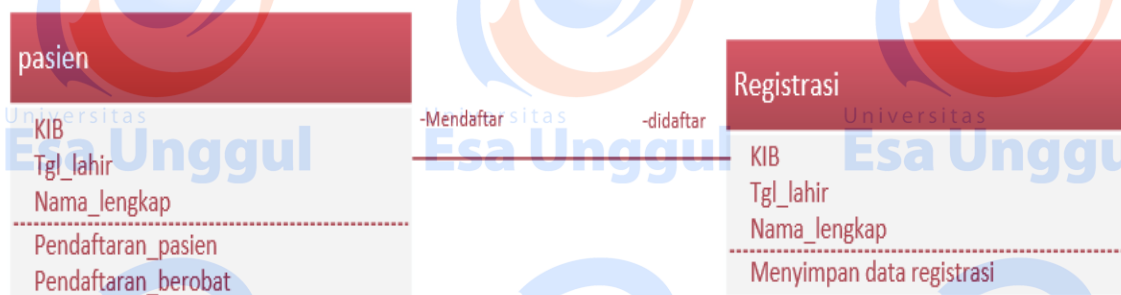
Pembuatan diagram kelas dibagi menjadi dua bagian, yaitu kelas itu sendiri dan relasi antarkelas. Kelas dibagi menjadi tiga bagian, yakni nama kelas, atribut kelas, dan operasi kelas (methods). Nama kelas adalah nama dari kelas itu sendiri, misalnya kelas pasien, kelas dokter, dan lain-lain. Atribut adalah data yang dimiliki oleh kelas tersebut, misalnya kelas pasien memiliki atribut nama pasien, tanggal lahir, jenis kelamin, dan lain-lain. Lalu operasi kelas adalah apa yang kelas itu kerjakan, misalnya kelas pasien dapat melakukan pendaftaran pasien.

contoh klaifikasi diagram kelas adalah sebagai berikut.

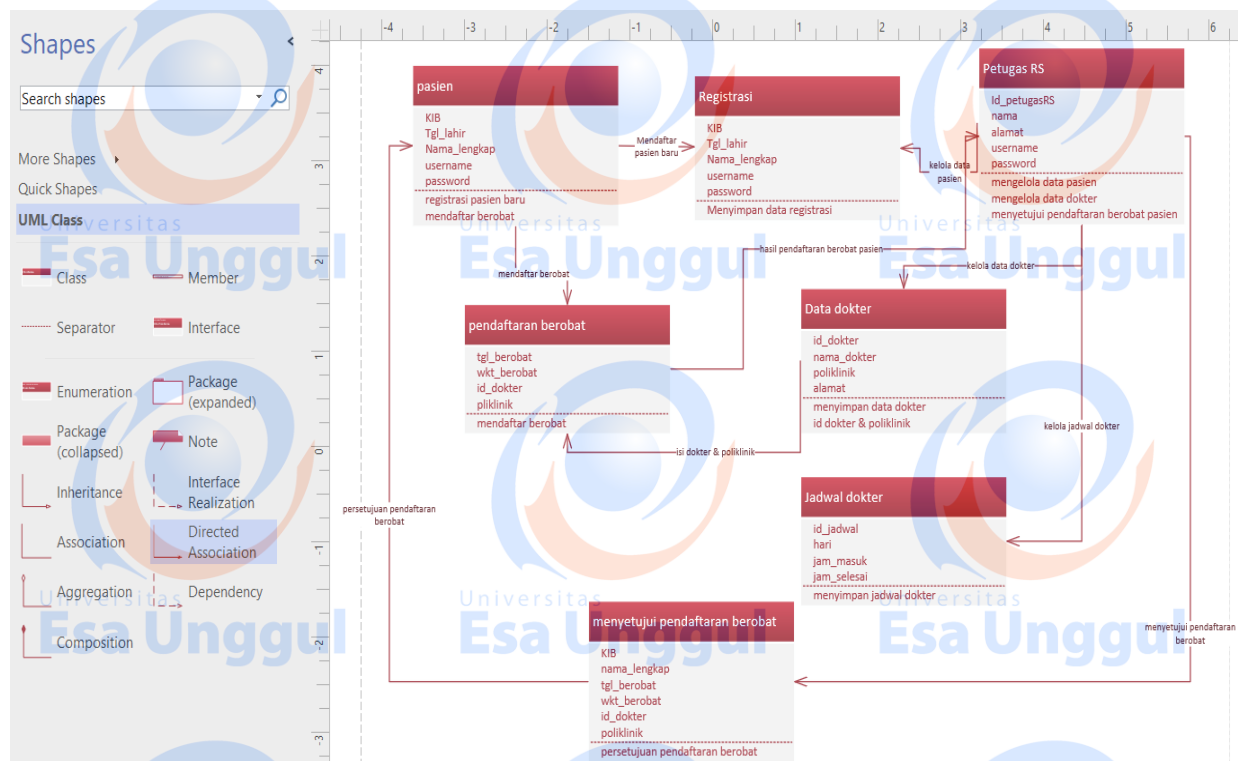


Setelah membuat kelas, selanjutnya harus membuat relasi antar kelas. Misalnya pasien melakukan pendaftaran pasien di bagian registrasi dalam hal ini pasien dan registrasi adalah kelas, pendaftaran pasien adalah relasi antarkelas.

Contoh diagram kelas



Contoh relasi diagram kelas



- Class Pasien memiliki relasi langsung dengan class registrasi karena di registrasi pasien wajib memasukkan identitas pasien dan Petugas Rumah Sakit mengelola kembali hasil registrasi pasien.
- Petugas Rumah Sakit memiliki relasi dengan Mengolah Data Dokter Karena data Dokter akan di kelola oleh Petugas Rumah Sakit.
- Begitu juga Petugas Rumah Sakit memiliki relasi dengan Mengolah Data Jadwal Dokter Karena data Jadwal Dokter akan di kelola oleh Petugas Rumah Sakit.
- Pasien Melakukan Pendaftaran Berobat dimana dalam Class Pendaftaran Berobat\terdapat atribut yang berada di Class Mengolah Data Dokter yaitu ID_Dokter dan Poliklinik.
- Petugas Rumah Sakit Akan menerima hasil pendaftaran berobat pasien sehingga Petugas Rumah Sakit memiliki relasi langsung dengan class Menyetujui Pendaftaran Berobat, untuk memberikan persetujuan pendaftaran berobat terhadap pasien.

DIAGRAM SEQUENCE

Diagram sequence adalah diagram yang dibuat untuk mengetahui alur interaksi antar objek. Isi dari diagram sequence harus sama dengan use case dan diagram kelas. Satu use case tunggal akan digambarkan satu diagram sequencenya.

Contoh sequence diagram pada use case tunggal : Petugas Rumah Sakit pada sistem yang dibuat.

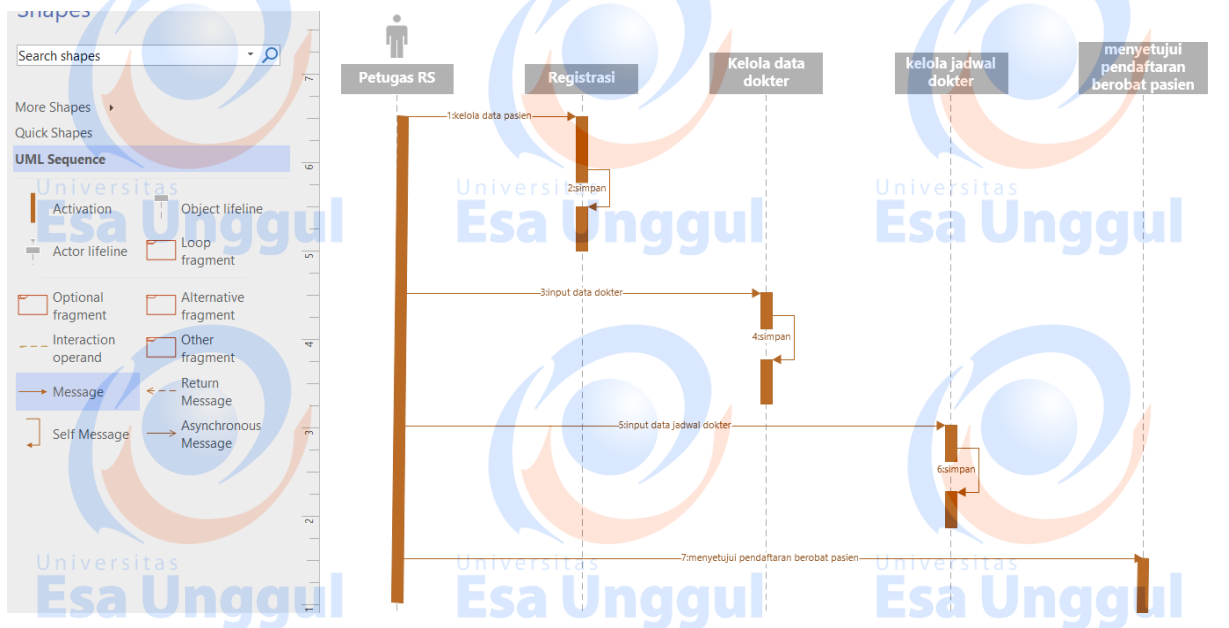
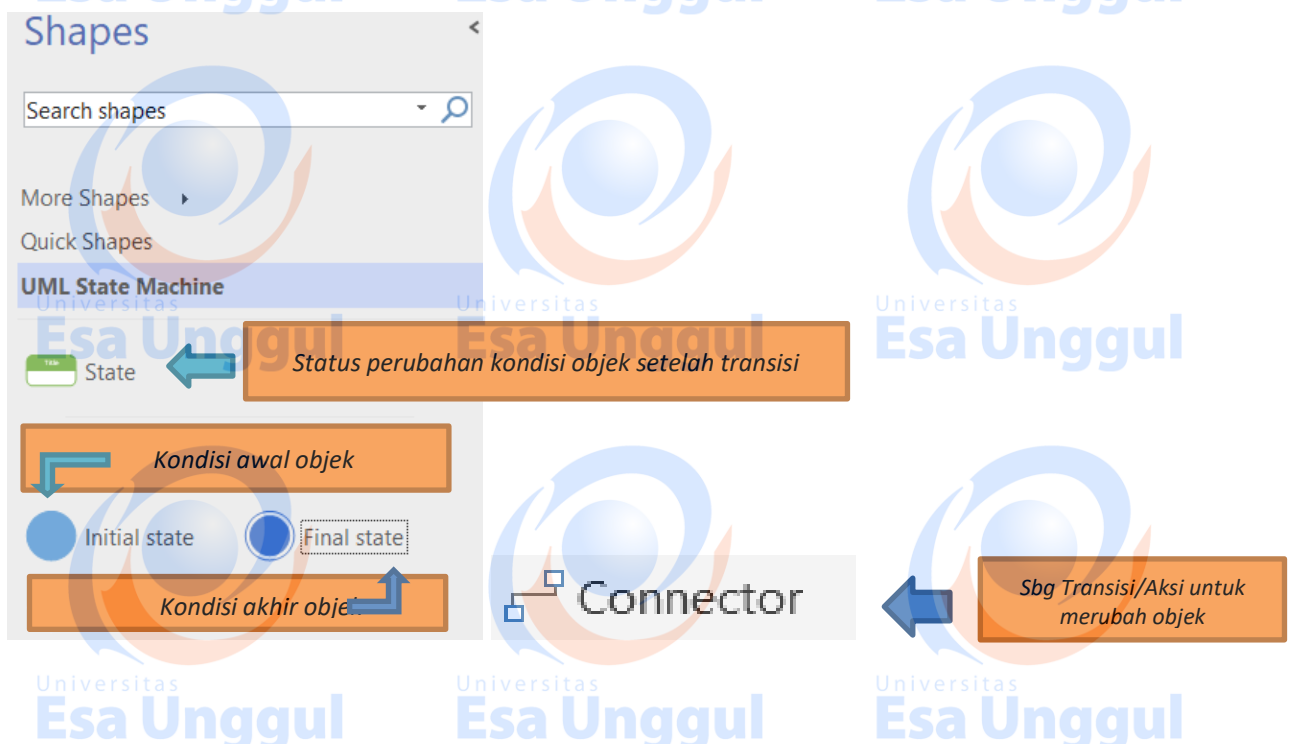


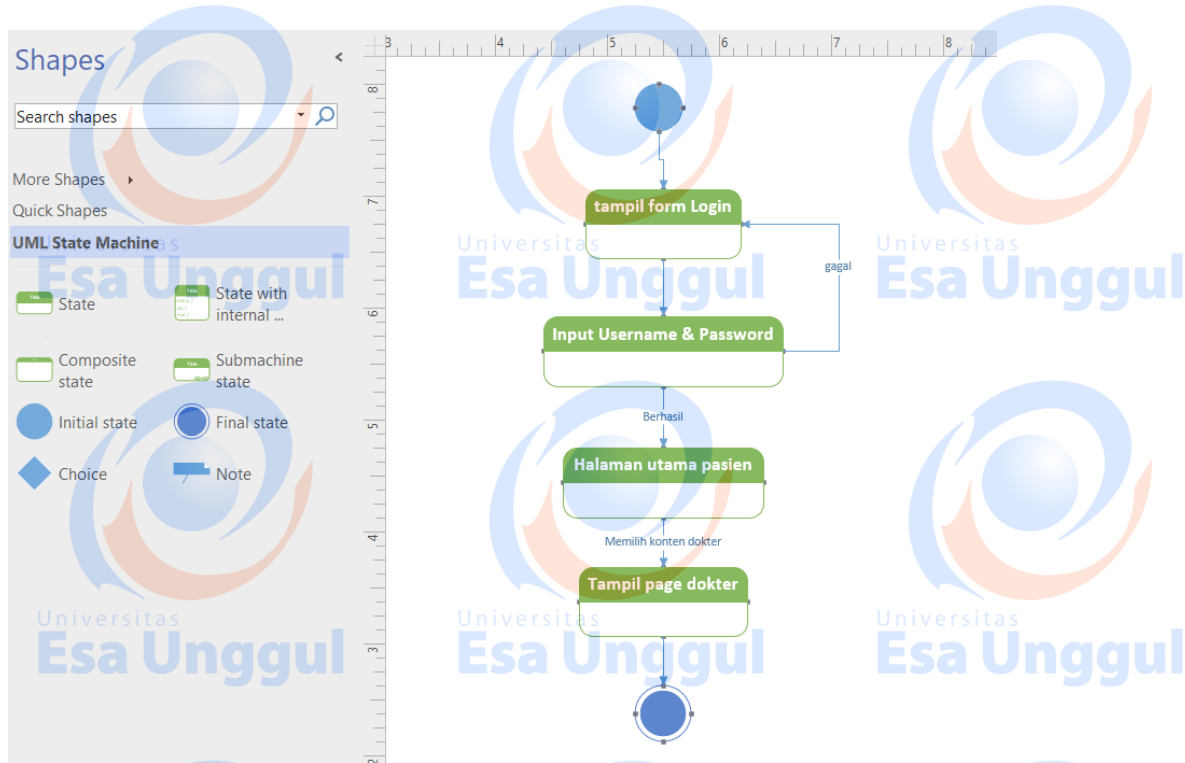
DIAGRAM STATE

Diagram state menggambarkan perilaku dari objek. Saat menggambar diagram state, diagram kelas harus sudah dibuat terlebih dahulu. Objek yang dibuat dari kelas yang sama akan memiliki perilaku yang sama, maka jumlah diagram state yang dibuat harus sama dengan jumlah kelas yang dibuat.

Symbol, nama dan keterangan yang digunakan pada diagram state



Contoh diagram state pasien memilih dokter



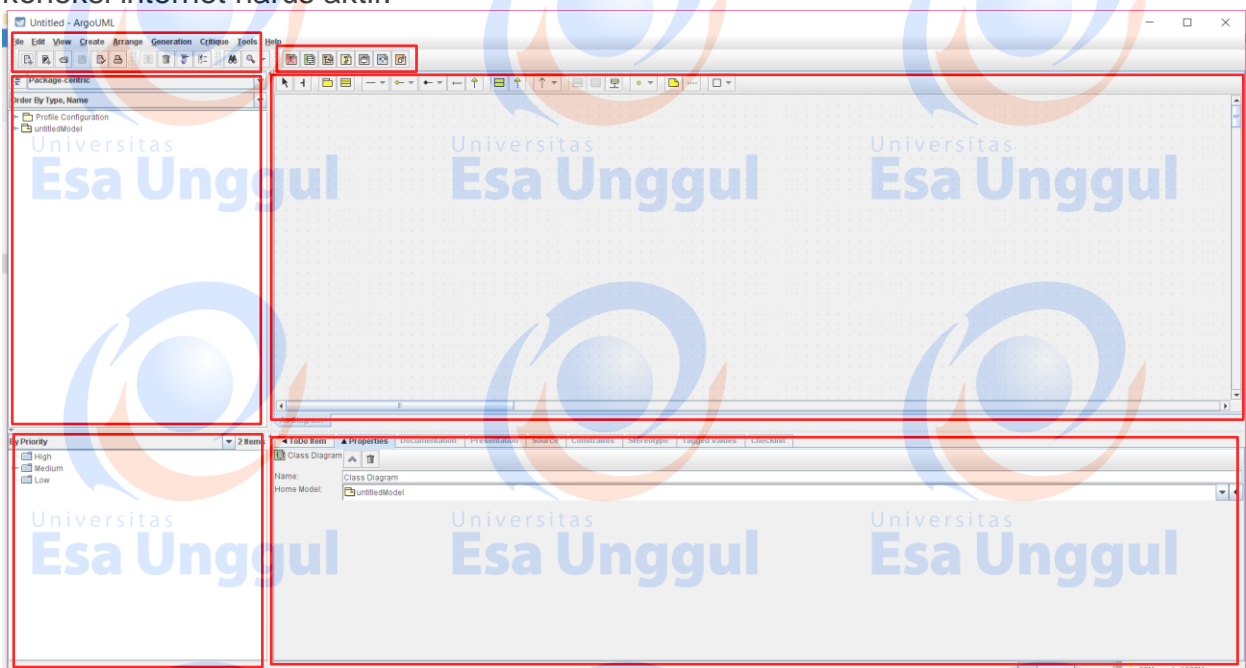
LATIHAN 1

Contoh Kasus

Aplikasi perangkat lunak yang akan dibangun di perpustakaan Univ X berbasis web dengan ruang lingkupnya adalah hanya diakses internal oleh pustakawan sebagai pengelola, serta dosen dan mahasiswa sebagai anggota. Lalu mahasiswa memiliki batasan transaksi peminjaman buku selama seminggu dan jika lewat seminggu tidak mengembalikan buku maka akan dikenakan denda, berlaku untuk dosen dan mahasiswa. Buatlah use case diagram, class diagram, sequence diagram (use case tunggal pada pustakawan dengan use case peminjaman) dan state diagram menggunakan aplikasi argoUML.

Untuk membuat UML, tersedia aplikasi gratis bahkan aplikasi *open source*, misalnya: StarUML, Aceleo & ArgoUML. Jika hendak menggunakan aplikasi ArgoUML gratis dan *open source*, dapat mengunduh aplikasi pada link www.argouml.tigris.org.

Setelah diunduh, lakukan instalasi seperti biasa. ArgoUML memerlukan aplikasi JAVA sebagai dasarnya. Bila komputer atau laptop tidak tersedia aplikasi tersebut, maka ArgoUML akan mengunduhnya secara otomatis dengan syarat koneksi internet harus aktif.



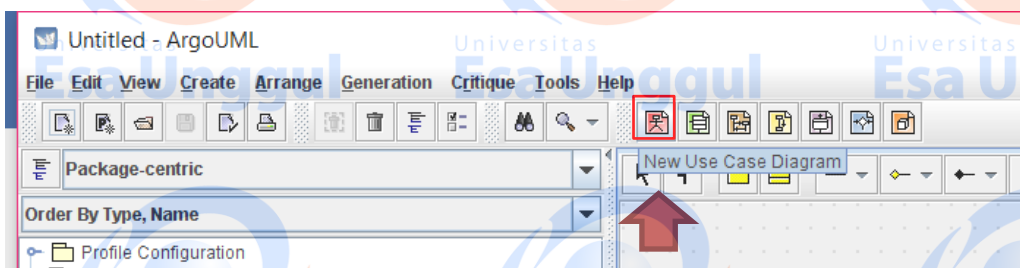
Terdapat enam bagian pada aplikasi ini:

- 1) **Menu navigasi:** menu standar seperti new, save, open project, dll
- 2) **Menu model :** menu untuk membuat diagram use case, kelas, sequence, dan state
- 3) **List model :** tempat semua diagram yang sudah dibuat
- 4) **Kanvas :** tempat menggambar model
- 5) **List diagram :** menampilkan diagram apa saja yang dibuat
- 6) **Menu properti :** properti dari diagram yang dibuat

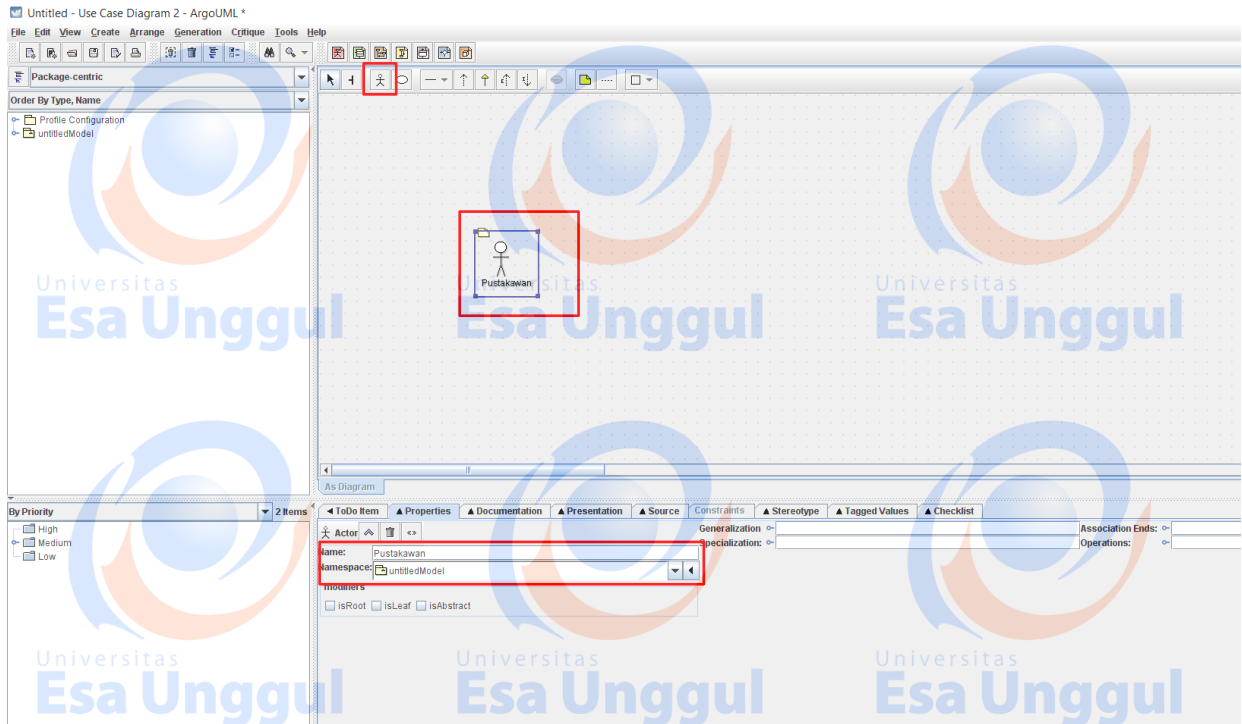
Membuat diagram:

1. Use case diagram

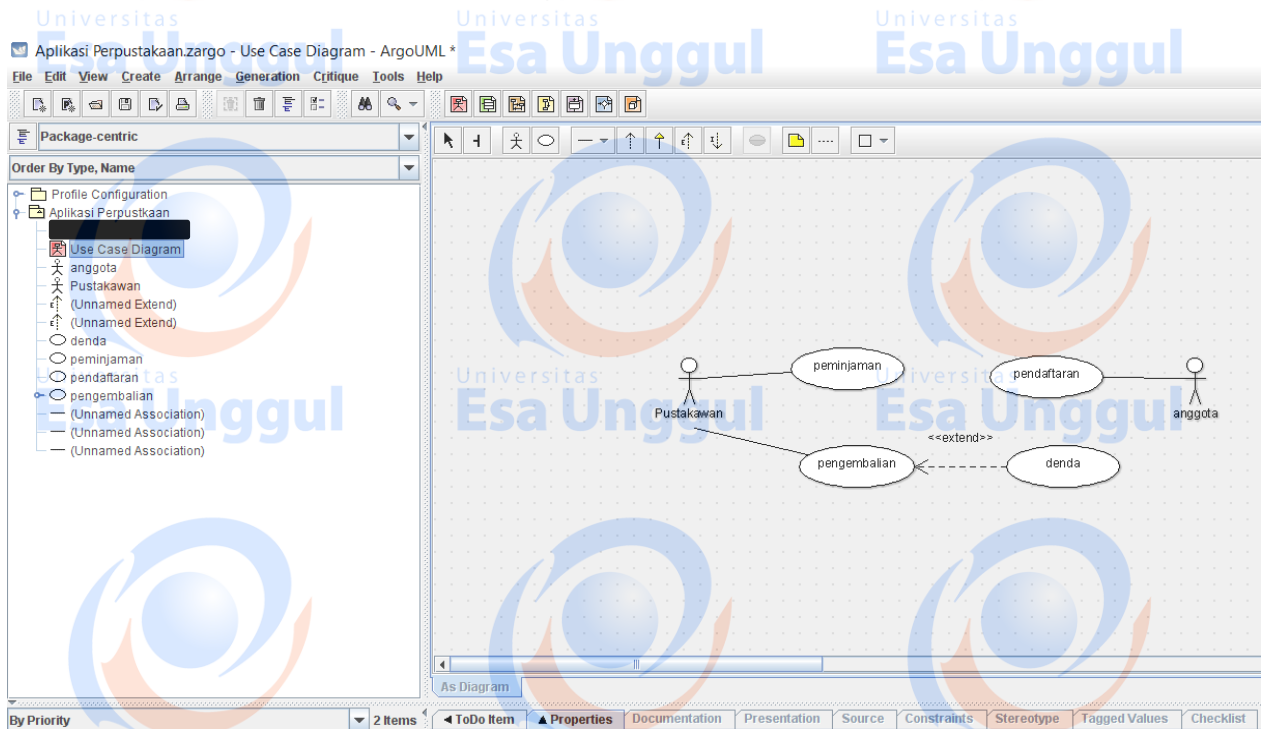
- a. Pada bagian “menu model”, pilih “new use case diagram” (terletak di paling kiri berwarna merah”.



- b. Gambar use case seperti contoh dengan cara klik simbolnya (aktor, use case, dan asosiasi) di bagian “kanvas”, lalu klik kembali ke arah kanan “kanvas” tepat di bawahnya. Kemudian beri nama simbol di bagian “menu properti”.



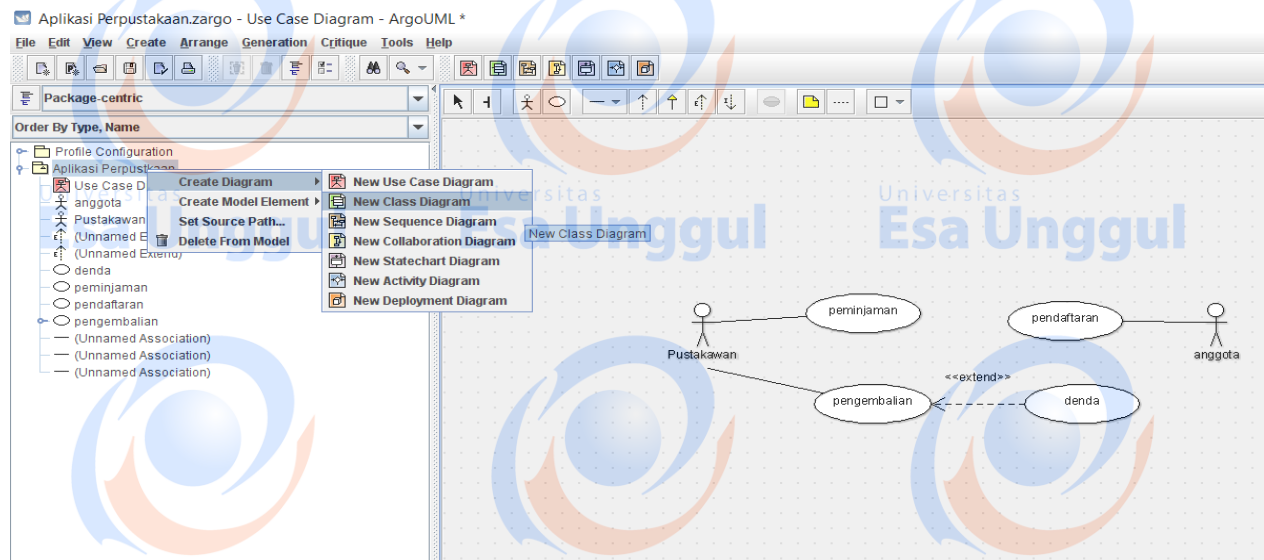
- c. Lanjutkan hingga hasilnya sama dengan contoh pada subbab diagram use case di atas dan contoh kasus sebelumnya.
- d. Setelah selesai membuat use case, hasilnya akan terlihat seperti gambar di bawah ini. Diagram yang dibuat akan muncul di bagian "list model". Simpan dalam bentuk project dengan memilih menu "file" pada bagian "menu navigasi", pilih "save project as". Hasilnya akan berupa file dengan ekstensi ZARGO.



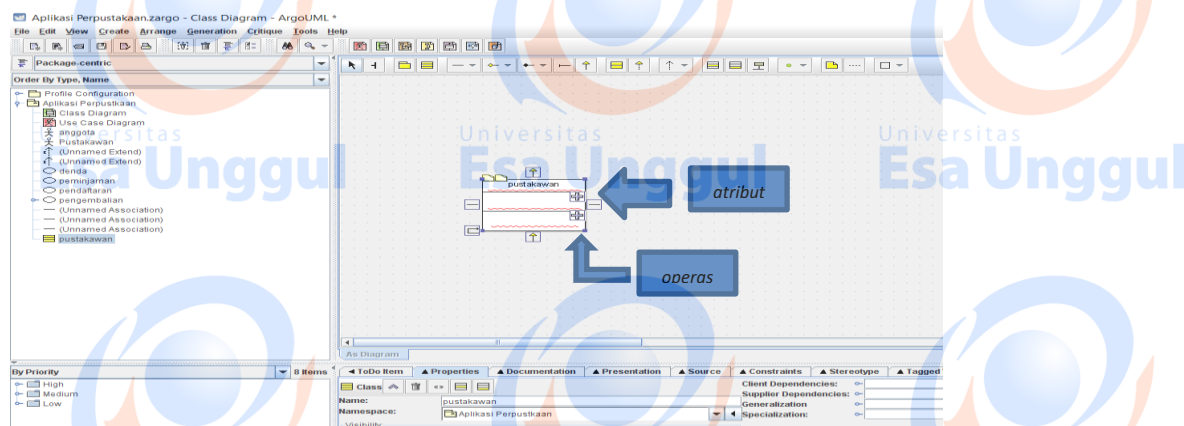
- e. Jangan lupa untuk mengubah nama folder project dengan nama “aplikasi perpustakaan”.
- f. Untuk membuat diagram baru, tidak perlu menggunakan project yang baru pula, cukup dengan menggunakan project yang sudah ada karena sifatnya adalah melanjutkan diagram dengan aplikasi yang sama.

2. Class diagram

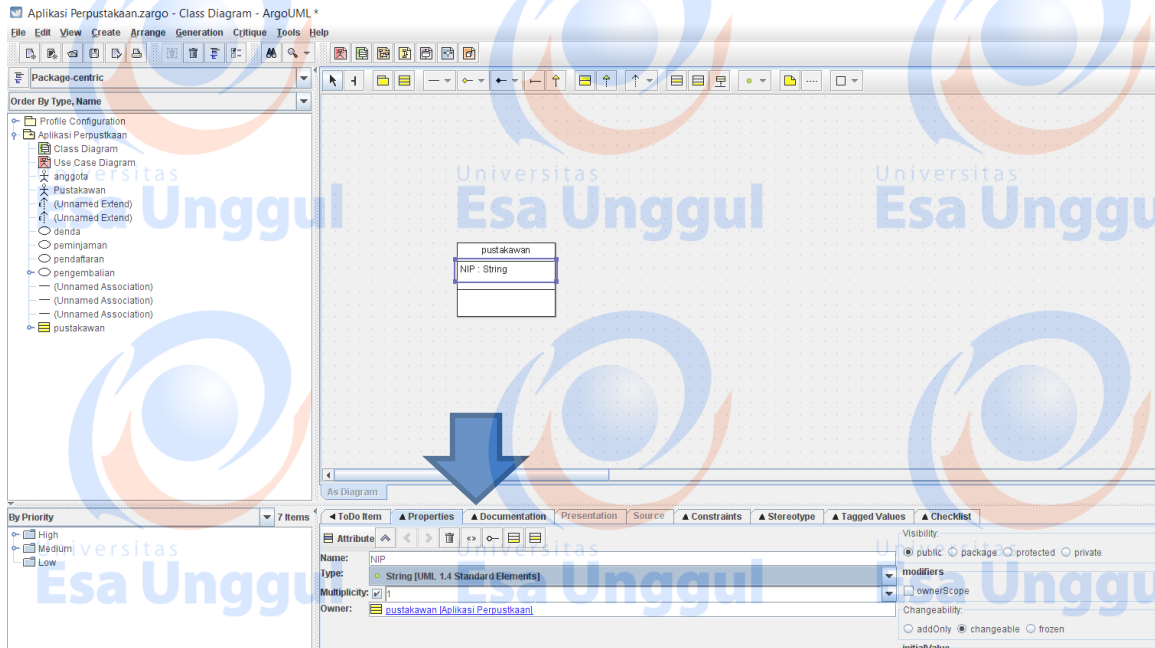
- a. Selanjutnya untuk membuat diagram kelas, klik kanan pada folder project, pilih “create diagram”, lalu “new class diagram”. Diagram kelas akan muncul pada bagian “list model” dan siap untuk dibuat atau juga bisa membuatnya dengan memilih “new class diagram” pada bagian “menu model”.



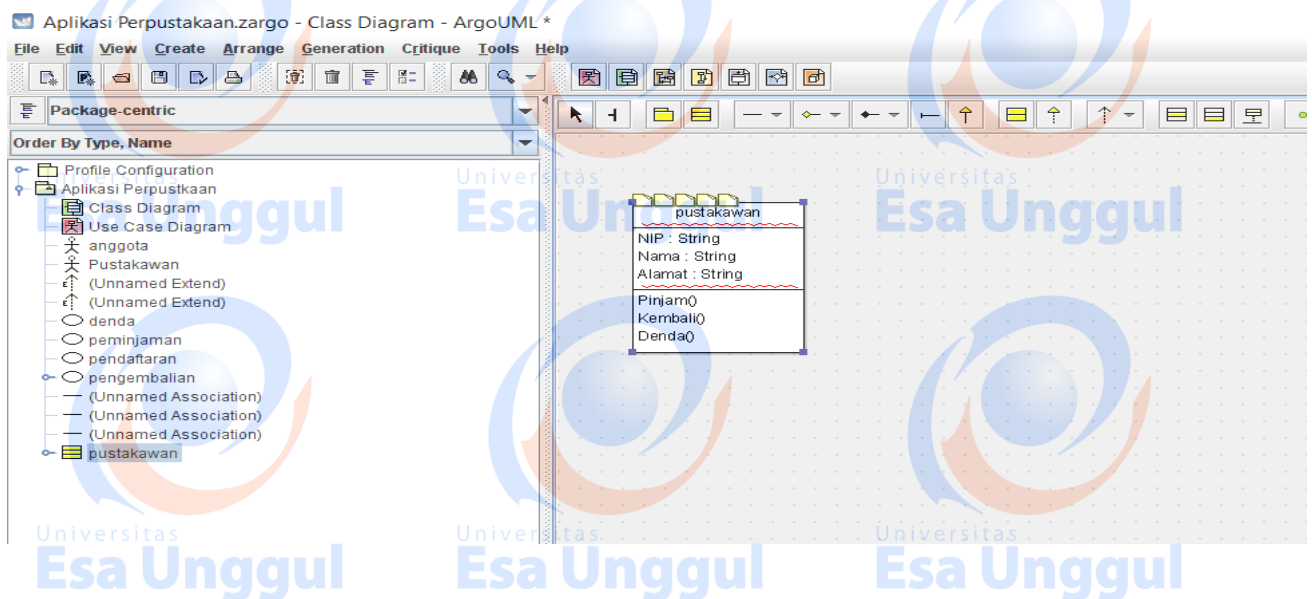
- b. Untuk menambahkan atribut dan operasi pada kelas yang dibuat, klik kelas yang diinginkan, akan tampil tanda “+” di bagian kanan kelas tersebut. Klik yang pertama untuk atribut, klik kedua untuk operasi.



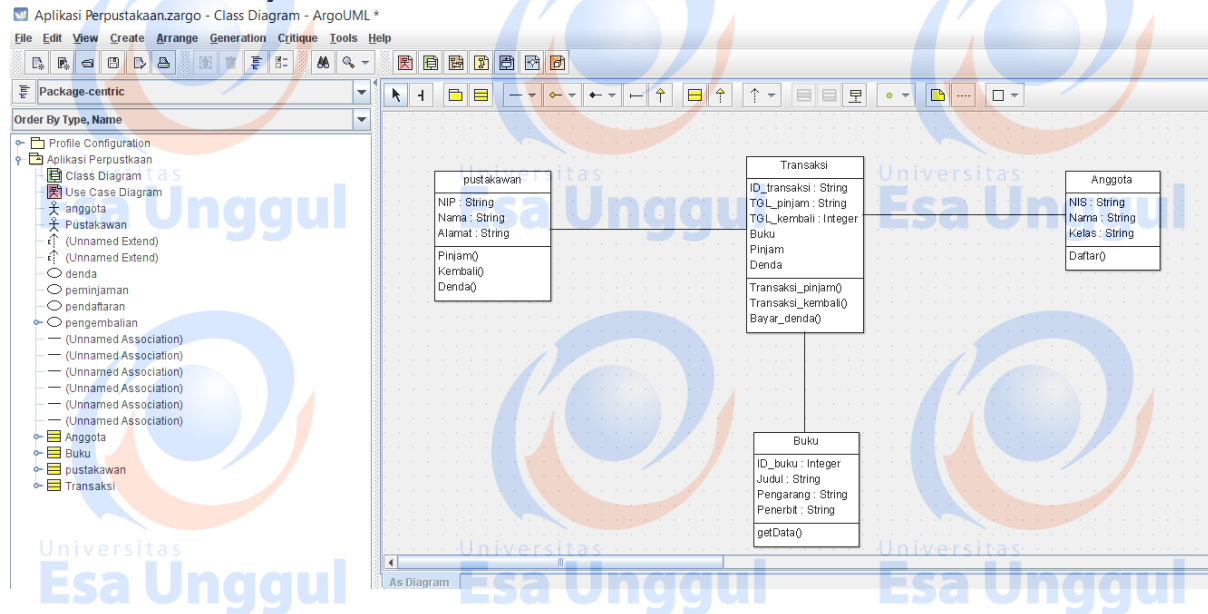
c. Isi atribut dan operasi pada bagian menu property



d. Untuk menambah atribut dan operasi, lakukan cara yang sama

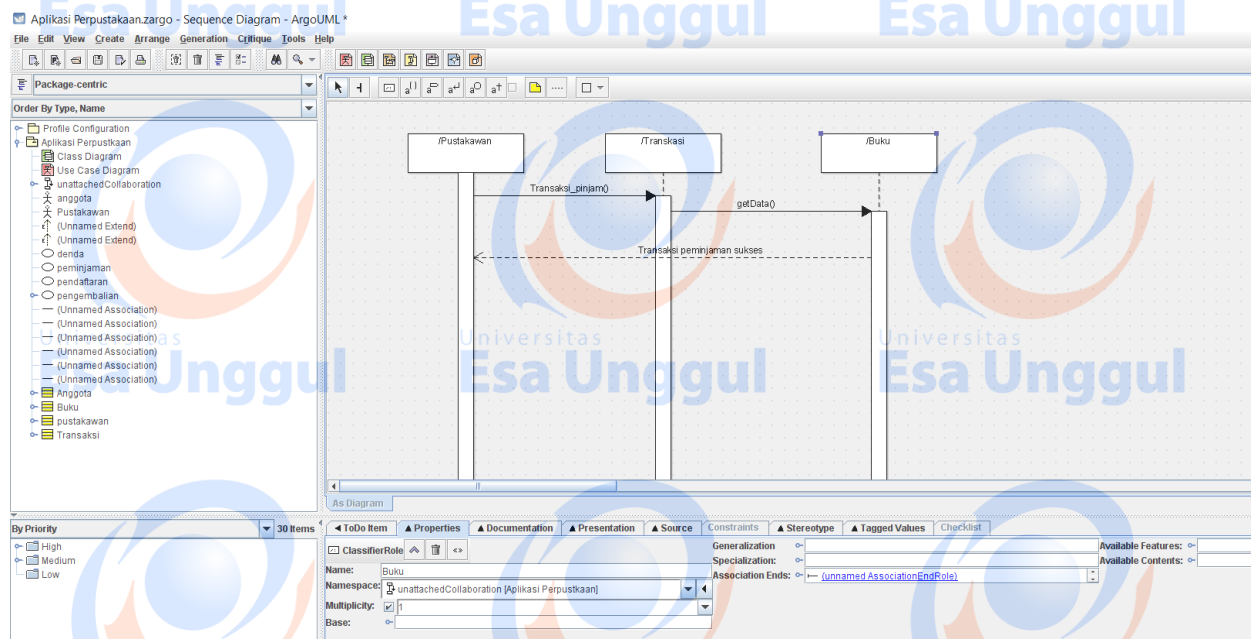


e. Buatlah diagram kelas yang sama dengan contoh kasus yang sudah dijelaskan di atas sebelumnya



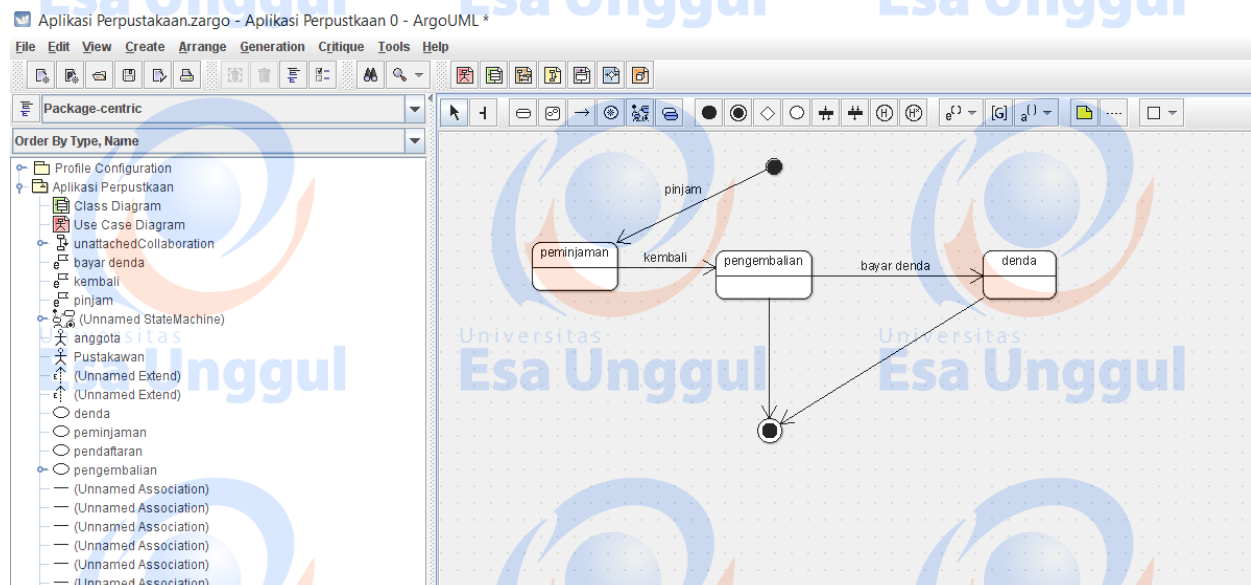
3. Sequence diagram

Use case aplikasi perpustakaan terdapat empat use case tunggal, yaitu pustakawan dengan peminjaman, pustakawan dengan pengembalian, pustakawan dengan denda, dan anggota dengan pendaftaran. Buat pula sequence diagramnya sesuai contoh use case tunggal (pustakawan dengan use case peminjaman) berdasarkan kasus dan penjelasan definisi sequence diagram sebelumnya.



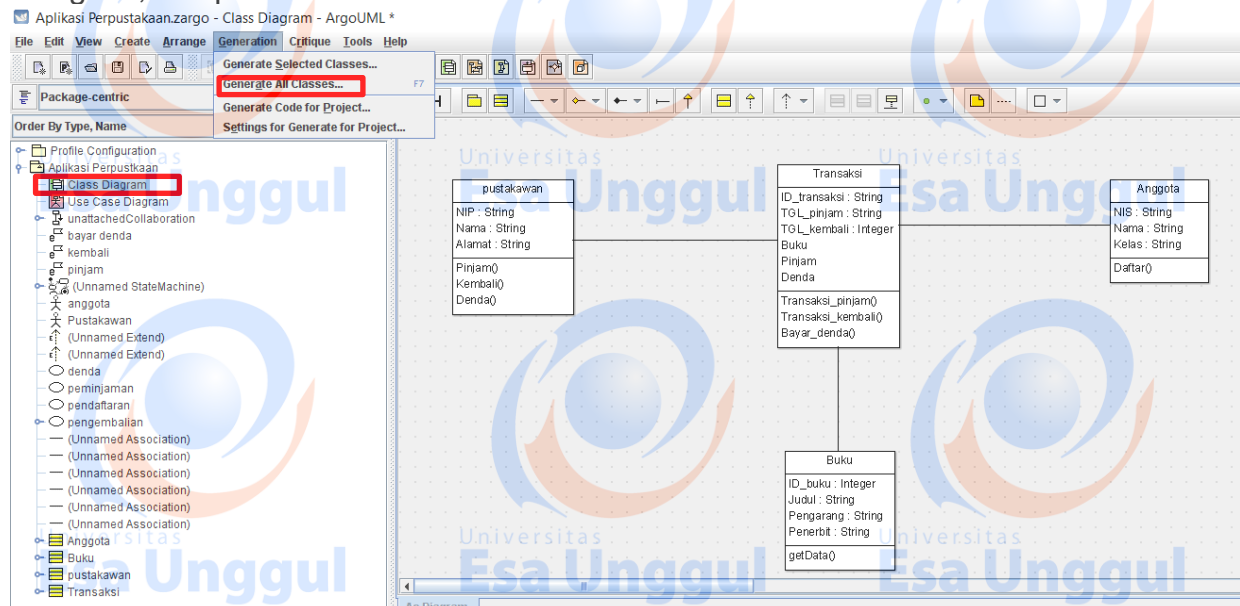
4. State diagram

Setelah membuat sequence diagram, selanjutnya membuat state diagram. Ambil salah satu kelas dari aplikasi perpustakaan berdasarkan contoh kelas, misalnya kelas transaksi. Kelas ini memiliki tiga layanan, yaitu peminjaman, pengembalian, dan denda. Kelas ini akan memiliki status peminjaman jika aksinya adalah pinjam, akan memiliki status pengembalian jika aksinya adalah kembali dan akan menjadi status denda jika aksinya adalah bayar denda. Berikut adalah diagramnya.

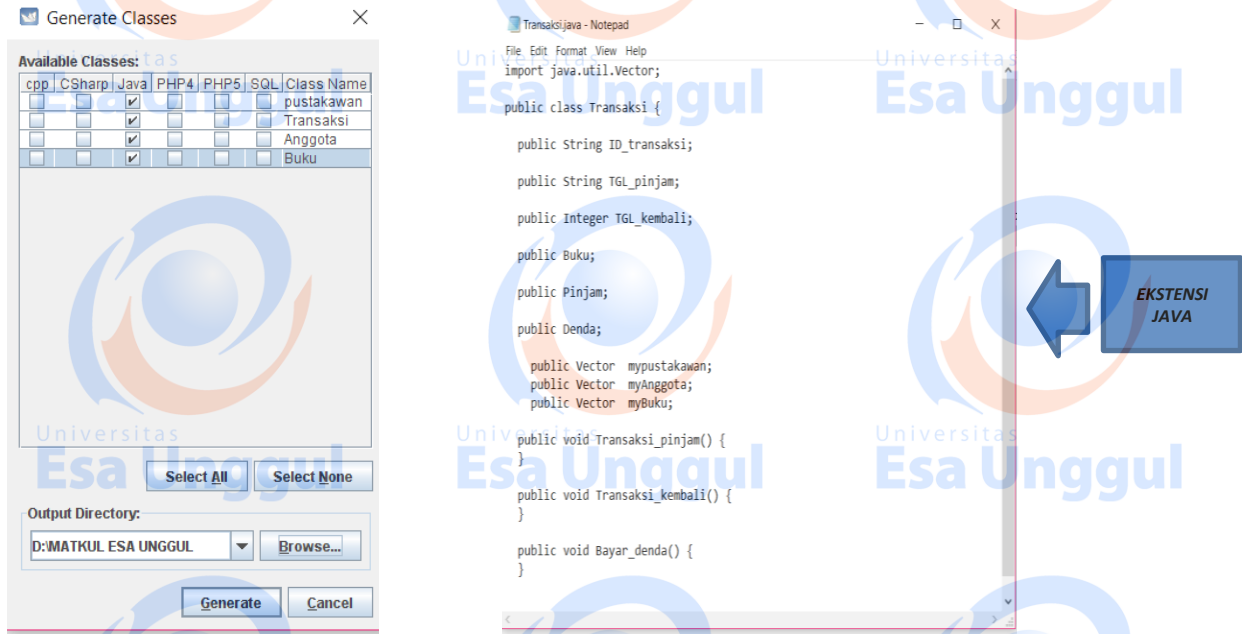


5. Ekstraksi kode dari diagram UML

Programmer yang merancang sistem menggunakan perangkat lunak berorientasi objek tidak perlu melakukan koding dari awal. Setelah diagram UML dibuat, hasilnya dapat diekstraksi menjadi kode. Untuk melakukan hal tersebut pilih diagram kelas terlebih dahulu di bagian “list model”, kemudian pilih “generation” pada bagian “menu navigasi”, lalu pilih “Generate All Classes”.



Lalu akan tampil jendela baru, sesuaikan dengan bahasa pemrograman yang digunakan untuk koding. Misalnya, koding akan menggunakan JAVA, pilih semua kelas dan centang pada bagian JAVA, pilih tempat penyimpanan, lalu file pun akan terbentuk dengan ekstensi JAVA.



Setelah memodelkan dengan UML, maka langkah selanjutnya adalah memodelkan garis data menggunakan entitiy relationship diagram (ERD) yang akan menjadi dasar dalam membangun basis data.

LATIHAN 2

A. Contoh Kasus

Pengarsipan data pasien yang dilakukan di klinik X masih menggunakan metode konvensional dengan menggunakan media kertas sebagai bahan penyimpanan. Hal tersebut menjadi kendala dalam pencarian data pasien yang memerlukan waktu, bahkan arsip seringkali hilang atau tidak ditemukan. Masalah yang terjadi adalah belum adanya sistem pengarsipan digital pada Klinik X dalam mengelola data pasien yang berobat di Klinik tersebut. Penggunaan metode konvensional dalam pengelolaan data di Klinik X menyebabkan pencarian data menjadi kurang efektif dan efisien.

Aplikasi perangkat lunak yang akan dibangun di Klinik X adalah **sistem informasi pengarsipan data pasien** dimana perangkat lunak yang di-design memiliki fitur penambahan data diri pasien baru, menyimpan data diri pasien, menghapus data pasien yang sudah lama tidak berobat, serta mengupdate data diri pasien, jika sewaktu mendaftar ada ketentuan yang belum terpenuhi.

Contoh design sistem informasi pengarsipan data pasien di klinik X menggunakan pendekatan object oriented adalah sebagai berikut.

B. Membuat Diagram Dengan ArgoUML

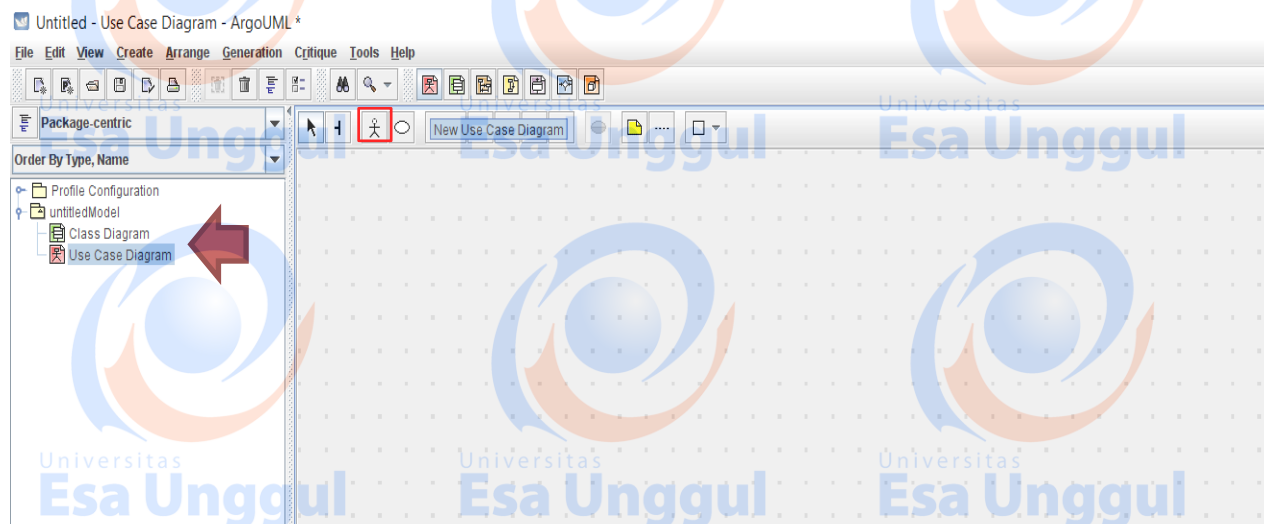
Arsitektur sistem informasi pengarsipan data pasien di Klinik X yang akan dibangun terdiri dari **tiga komponen** yaitu: **pengguna**, **antarmuka sistem** dan **basisdata**. **Komponen utama** dari sistem ini adalah **1) pengelolaan data pasien, 2) pengelolaan data penyakit dan 3) pembuatan laporan**.

Pengguna sistem adalah **administrator** aplikasi dan **dokter** yang bekerja di klinik tersebut. Pengguna **admin** mempunyai hak akses dalam **mengelola data pasien, mengelola data penyakit dan pembuatan laporan**. Sedangkan **dokter** hanya mempunyai hak akses dalam **mengelola data pasien dan mengelola data penyakit**.

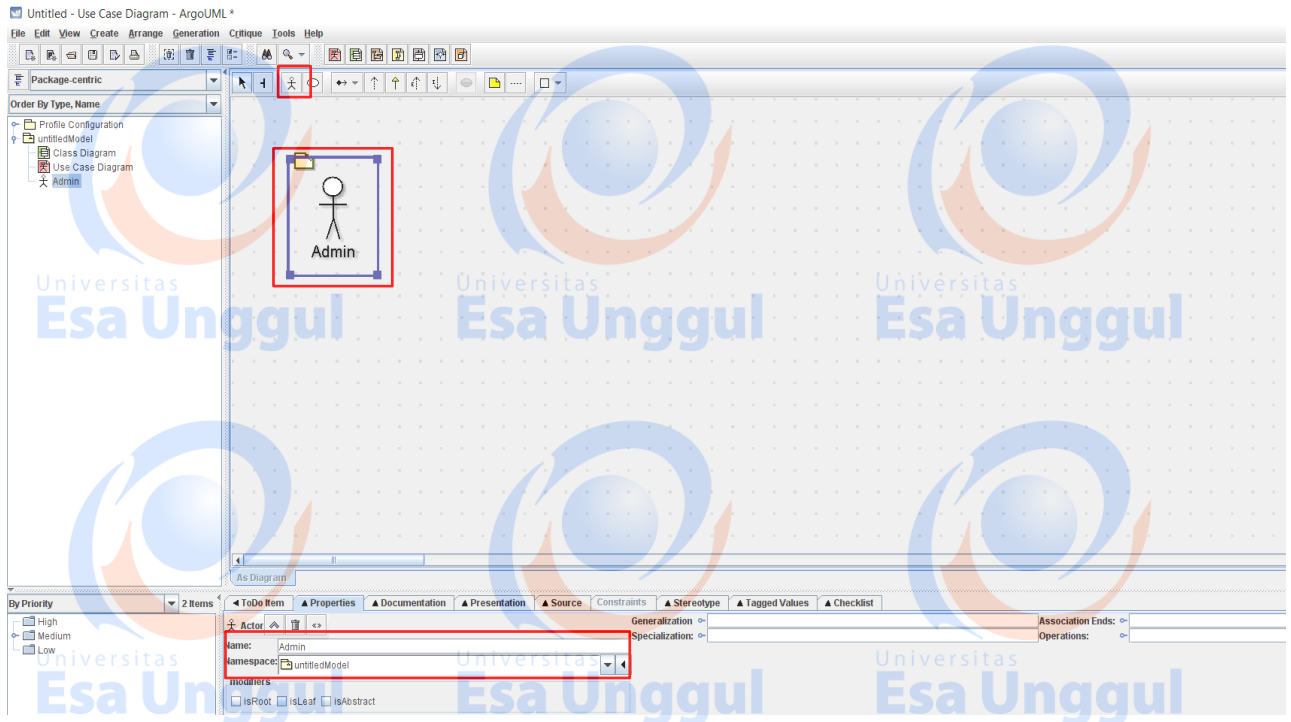
1. Use case diagram

Diagram Use Case menggambarkan fungsionalitas sistem yang dibangun. Diagram Use Case untuk Sistem Informasi Pengarsipan Data Pasien Klinik X ditunjukkan pada gambar berikut.

- a. Pada bagian “menu model”, pilih “new use case diagram” (terletak di paling kiri berwarna merah).



- b. Gambar use case seperti contoh dengan cara klik simbolnya (aktor, use case, dan asosiasi) di bagian “kanvas”, lalu klik kembali ke arah kanan “kanvas” tepat di bawahnya. Kemudian beri nama simbol di bagian “menu properti”.



c. Lanjutkan hingga hasilnya sama dengan contoh diagram use case berikut.

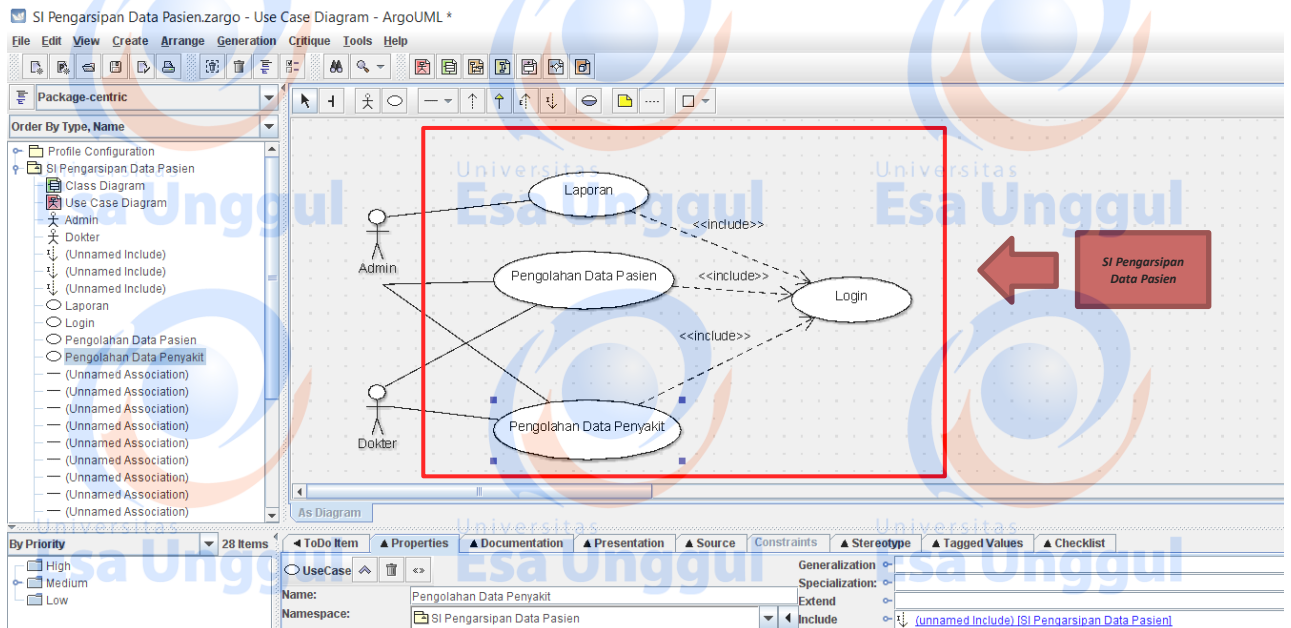
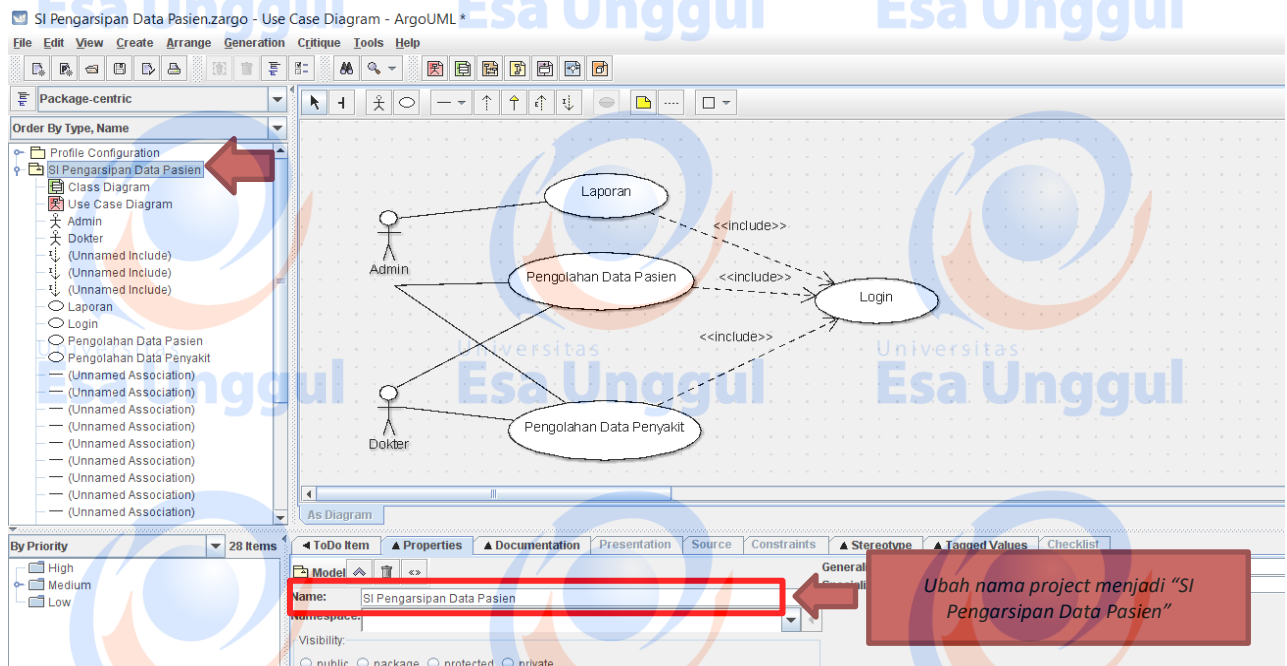


Diagram use case sistem informasi Klinik X terdiri dari **dua actor** yaitu **admin** dan **dokter**, selanjutnya terdapat **empat use case**. Admin memiliki asosiasi dengan melakukan login, pengolahan data pasien, pengolahan data penyakit dan pembuatan laporan. Sedangkan dokter hanya memiliki asosiasi melakukan login dan mengolah data pasien dan data penyakit.

- d. Setelah selesai membuat use case diagram, hasilnya akan terlihat seperti gambar di atas. Diagram yang dibuat akan muncul di bagian list “untitledModel”. Simpan dalam bentuk project dengan memilih menu “file” pada bagian “menu navigasi”, pilih “save project as”. Hasilnya akan berupa file dengan ekstensi ZARGO.
- e. Jangan lupa untuk mengubah nama folder project dengan nama “SI Pengarsipan Data Pasien”.



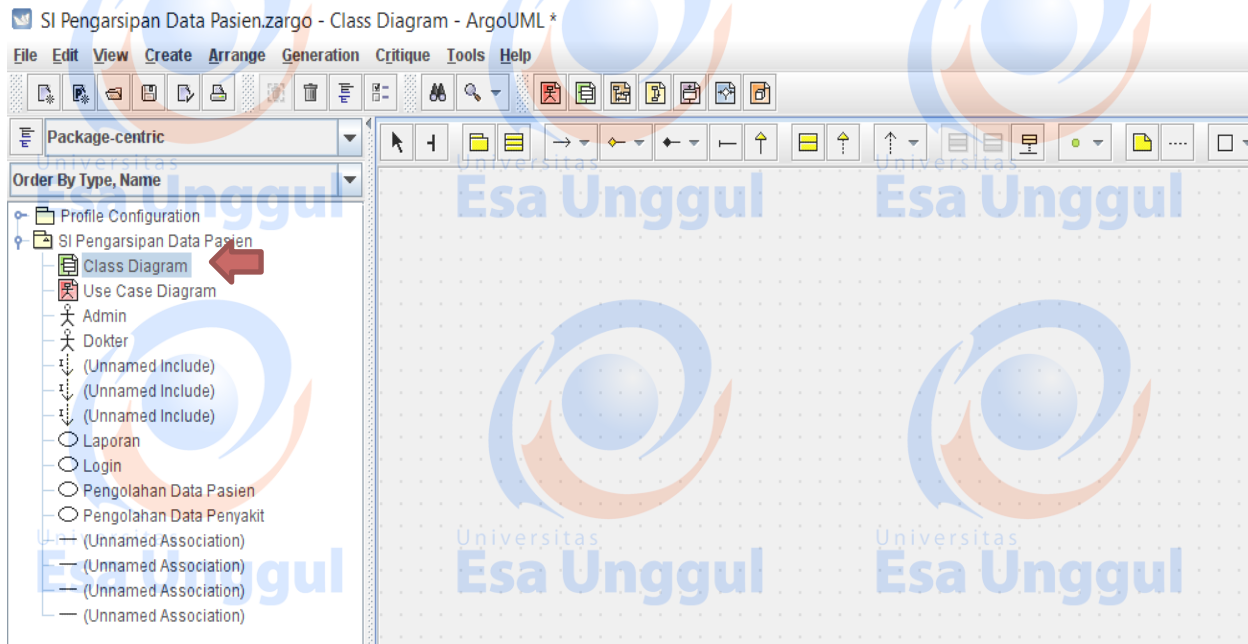
- f. Untuk membuat diagram baru, tidak perlu menggunakan project yang baru pula, cukup dengan menggunakan project yang sudah ada karena sifatnya adalah melanjutkan diagram dengan aplikasi yang sama.

2. Class diagram

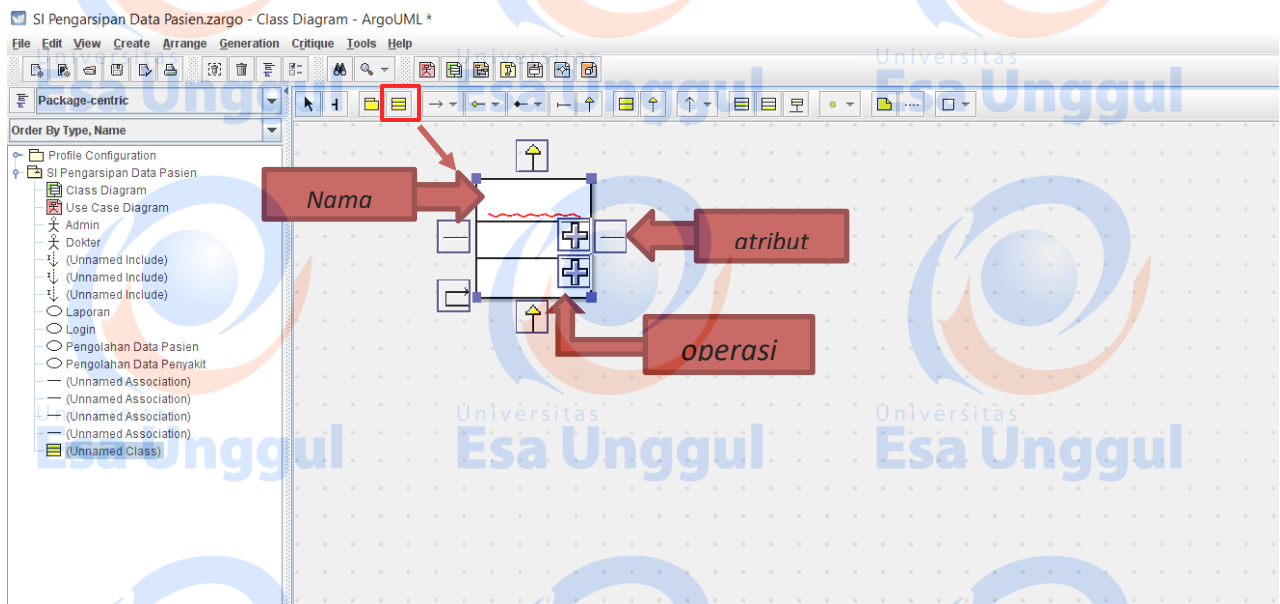
Diagram kelas menunjukkan relasi antar kelas dan menggambarkan jenis objek dalam sistem dan berbagai macam hubungan statis yang ada di antara mereka. Setiap kelas yang terlibat masing-masing memiliki atribut dan operasi.

SI Pengarsipan Data Pasien di Klinik X menunjukkan terdapat kelas **admin**, **dokter**, **olah_data_pasien**, **olah_data_penyakit** dan **pembuatan_laporan**. Relasi terjadi antara admin dengan beberapa kelas seperti olah_data_pasien, olah_data_penyakit dan pembuatan_laporan dimana relasi yang terjadi adalah satu admin dapat melakukan proses olah data pasien, olah data penyakit dan pembuatan laporan secara banyak/ berkali kali/ secara berkala. Sedangkan dokter hanya memiliki relasi ke kelas olah data pasien dan olah data penyakit dimana seorang dokter dapat mengolah data pasien dan penyakitnya satu kali tanpa ada batas maksimal.

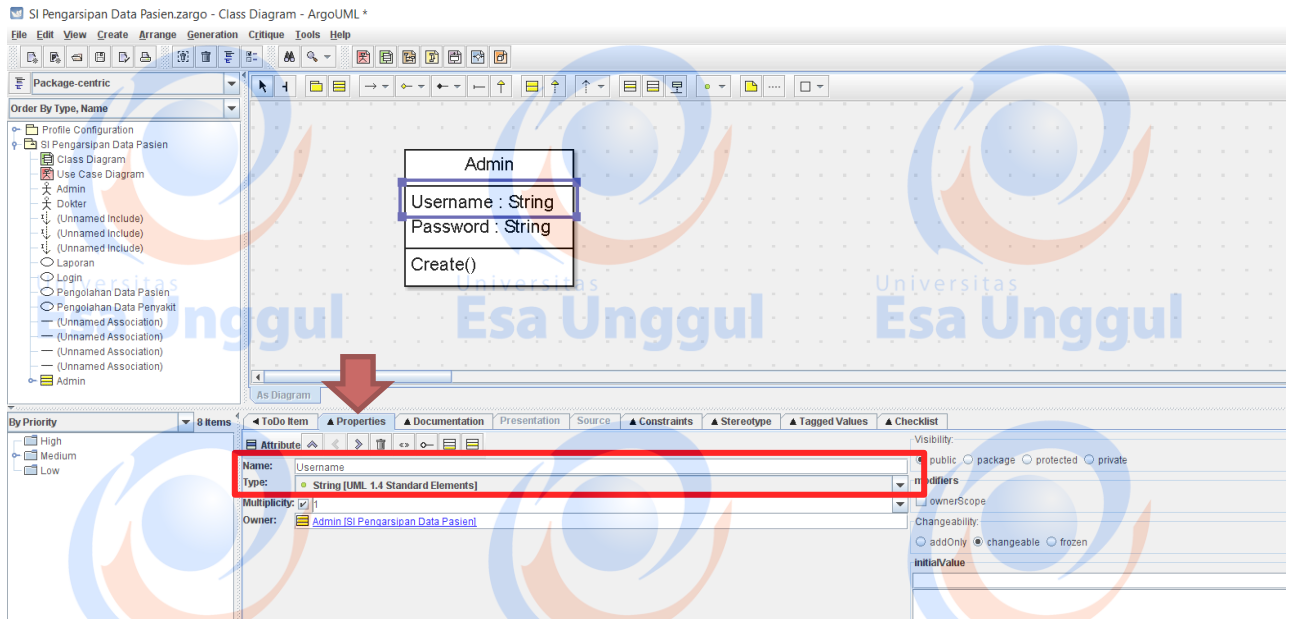
- a. Untuk membuat diagram kelas, klik **“class diagram”** pada bagian **“list model”** di toolbar menu kanan dan siap untuk dibuat, atau juga bisa membuatnya dengan memilih **“new class diagram”** pada bagian **“menu model”**.



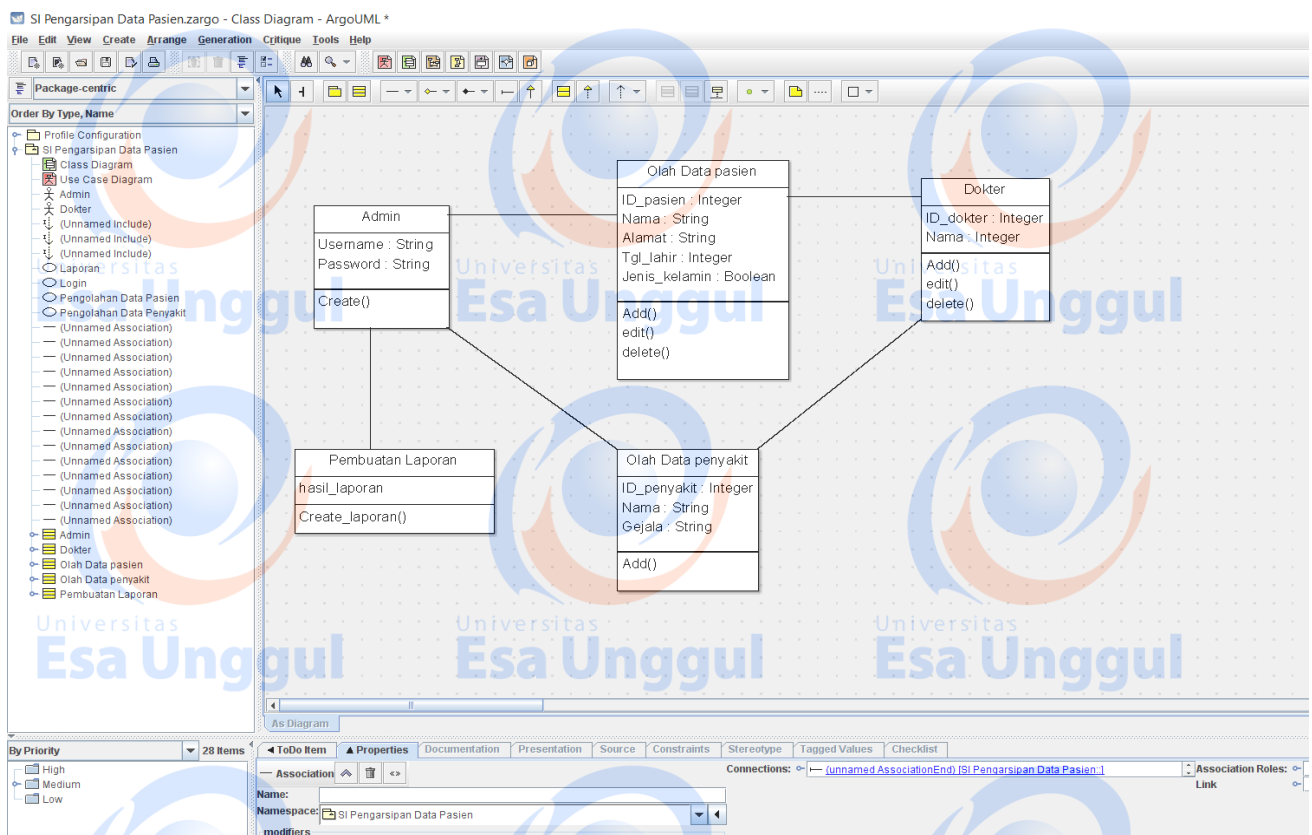
- b. Untuk menambahkan atribut dan operasi pada kelas yang dibuat, klik kelas yang diinginkan, akan tampil tanda **“+”** di bagian kanan kelas tersebut. Klik yang pertama untuk atribut, klik kedua untuk operasi.



- c. Isi nama, atribut dan operasi pada bagian **menu properties “Name”**. sebelumnya kita juga harus menentukan tipe data atribut yang akan diisi. Untuk tipe data teks dan karakter yaitu **string**. Untuk tipe data angka yaitu **integer**, tipe data kategorik yaitu tipe datanya **boolean**, ataupun tidak memiliki tipe data misal **“pembuatan hasil laporan”** yaitu klik **none** pada tipe data.



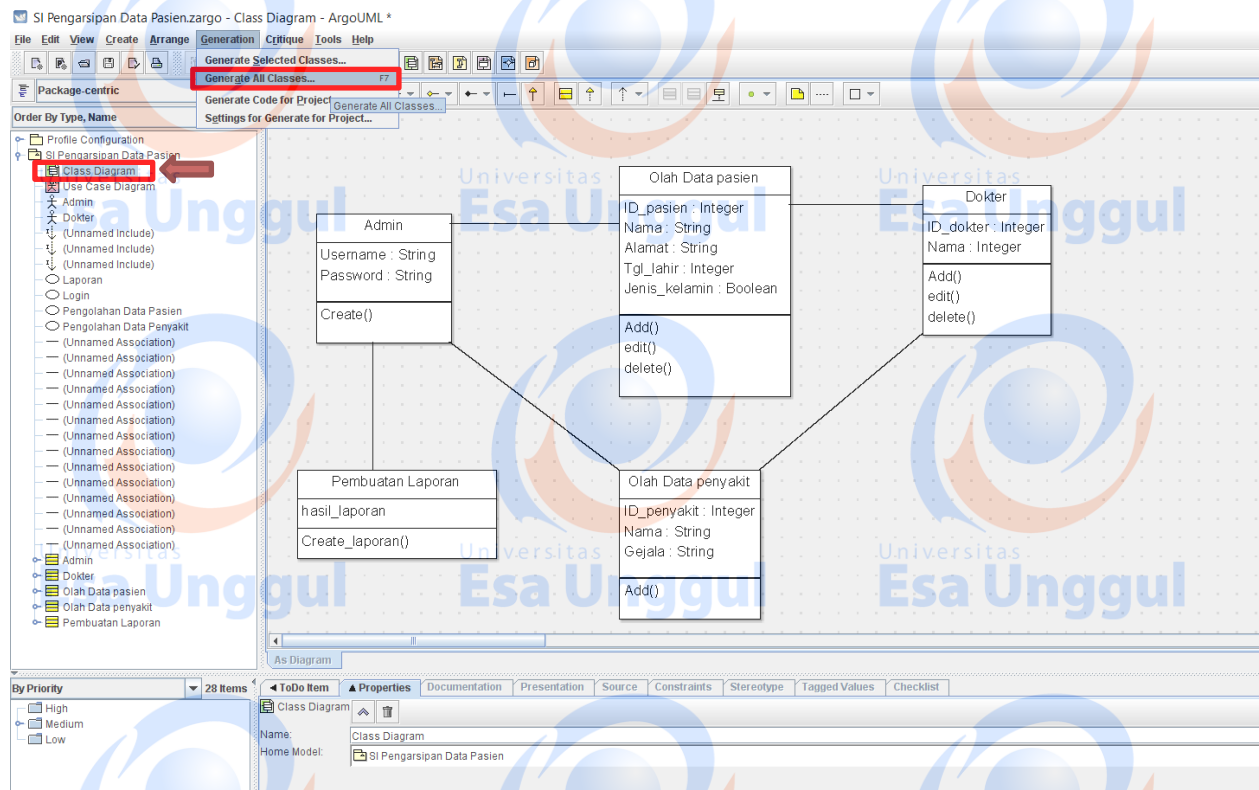
d. Selanjutnya buatlah diagram kelas yang sama dengan contoh kasus yang sudah dijelaskan di atas sebelumnya



3. Ekstraksi kode dari daigram UML

Programmer yang merancang sistem menggunakan perangkat lunak berorientasi objek tidak perlu melakukan koding dari awal. Setelah diagram UML dibuat, hasilnya dapat diekstraksi menjadi kode. Untuk melakukan hal tersebut **pilih diagram kelas**

terlebih dahulu di bagian list model, kemudian pilih **“generation”** pada bagian **“menu navigasi”**, lalu pilih **“Generate All Classes”**.



Lalu akan tampil jendela baru, sesuaikan dengan bahasa pemrograman yang digunakan untuk koding. Misalnya, koding akan menggunakan JAVA, pilih semua kelas dan centang pada bagian JAVA, pilih tempat penyimpanan, lalu file pun akan terbentuk dengan ekstensi JAVA.

Available Classes:	cpp	CSharp	Java	PHP4	PHP5	SQL	Class Name
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Admin
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Olah Data ...
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Olah Data ...
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pembuata...
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dokter

```

import java.util.Vector;

public class Admin {

    public String Username;

    public String Password;

    public Vector myOlah Data pasien;
    public Vector myOlah Data penyakit;
    public Vector myPembuatan Laporan;
    public Vector myOlah Data pasien;

    public void create() {
    }
}
    
```

Setelah memodelkan dengan UML, maka langkah selanjutnya adalah memodelkan garis data menggunakan entitij relationship diagram (ERD) yang akan menjadi dasar dalam membangun basis data.

EVALUASI

1. Jelaskan apa yang dimaksud UML dan 3 diagram UML (use case diagram, class diagram, dan activity diagram)!
2. Buatlah use case diagram, class diagram, sequence diagram & state diagram untuk (pilih salah satu): sistem informasi pendaftaran di RSX, sistem informasi rekam medis di RS X, sistem informasi di apotikX & system informasi di klinikX. (Dengan asumsi dan batasan anda sendiri)



Modul 2

Evaluasi Sistem Informasi

Esa Unggul Esa Unggul Esa Unggul

Modul pertemuan ini terdiri atas topik pembahasan yang membahas tentang Evaluasi Sistem Informasi dengan menggunakan teori model antara lain: *End User Computing (EUC) satisfaction, Task Technology Fit (TTF) Analysis, Human-Organization-Fit (HOT-Fit) Model, TAM (Technology Acceptance Model), dan DeLone & McLean*. Topik selanjutnya membahas tentang PLS (Partial Least Squares).

Kompetensi yang akan dicapai diantaranya; Mahasiswa mampu memahami teori dasar dan perbedaan dari beberapa model evaluasi sistem informasi, mampu menganalisis & mengevaluasi sistem dengan menggunakan uji PLS (Partial Least Squares) pada penelitian kuantitatif evaluasi sistem.

Universitas
Esa Unggul Universitas
Esa Unggul Universitas
Esa Unggul

Universitas
Esa Unggul Universitas
Esa Unggul Universitas
Esa Unggul

Universitas
Esa Unggul Universitas
Esa Unggul Universitas
Esa Unggul

Topik 2

Evaluasi Sistem

Pada hakekatnya evaluasi adalah penelitian tentang bagaimana program dijalankan, apakah proses dan dampaknya sudah sesuai dengan yang diharapkan, serta mengecek faktor-faktor penghambat yang dihadapi, dan faktor-faktor pendukung yang dimiliki untuk mencapai tujuan. Evaluasi juga merupakan proses pengumpulan data sebagai bahan pertimbangan untuk mengambil keputusan. Evaluasi diperlukan untuk mengukur tingkat keberhasilan mencapai tujuan.

Ada dua pendapat tentang evaluasi sistem informasi. Pertama evaluasi sistem informasi kesehatan menurut (Ammenwerth et al, 2004) adalah suatu tindakan untuk mengukur atau mengeksplorasi suatu sistem informasi kesehatan dari segi perencanaan, pengembangan, implementasi untuk memperoleh hasil informasi dalam pengambilan keputusan terutama konteks yang spesifik. Faktor utama yang menentukan keberhasilan dari penerapan teknologi informasi dan komunikasi dalam organisasi adalah sumber daya manusia. Sumber daya manusia khususnya adalah pengguna dari teknologi informasi dan komunikasi tersebut. Kedua, evaluasi sistem informasi merupakan suatu pengujian terhadap pengendalian infrastruktur sistem informasi. Dalam suatu evaluasi ada beberapa aspek yang diperiksa menyangkut efektivitas, efisiensi, availability system, reliability, confidentiality, integrity serta aspek security (Nugroho, 2008).

Tahapan dalam evaluasi sistem informasi menurut (Berg, 2004), yaitu latar belakang perlunya evaluasi dilaksanakan, kapan evaluasi dilaksanakan, hal-hal yang akan dievaluasi, bagaimana evaluasi dilakukan, melaksanakan evaluasi dan melaporkan hasilnya dan menentukan rekomendasi dan tidak lanjutnya. Menurut Davis (1996) mengelompokkan evaluasi sistem informasi dalam tiga ukuran: teknis, operasional dan ekonomi.

- b. Evaluasi teknis bertujuan untuk mengetahui apakah secara teknis sistem layak untuk menjalankan pengolahan informasi yang diusulkan. Banyak aplikasi diluar jangkauan kemampuan teknis dari perangkat keras dan perangkat lunak yang tersedia untuk pemakaian. Beberapa contoh pertanyaan dalam evaluasi teknis yaitu apakah tingkat transmisi data cukup ketat untuk menangani data, apakah terdapat cukup penyimpanan tambahan untuk merekam file yang diperlukan, dapatkah unit pusat pengolahan menanggapi semua permintaan dalam waktu yang telah ditentukan, apakah terdapat metode perhitungan masalah, atau apakah sistem pengoperasian mendukung pendekatan operasional yang diusulkan. Bilamana aplikasi sudah dijalankan, evaluasi berikutnya bisa menunjukkan bahwa aplikasi berjalan tidak efektif, hal ini disebabkan karena kemampuan teknis perangkat lunak tidak mampu mendukung aplikasi yang mantap.
- c. Evaluasi operasional yaitu pertimbangan kelayakan operasional berkaitan dengan masalah apakah data masukan dapat disediakan dan keluaran dapat digunakan dan benar dipakai. Evaluasi aplikasi setelah pelaksanaan harus menelusuri seberapa baik aplikasi itu bekerja dalam hubungan dengan masukan, tingkat kesalahan, ketepatan waktu hasil pengolahan, dan pemanfaatannya.

Pemanfaatan hasil pengolahan memerlukan perlunya evaluasi secara periodik dari aplikasi.

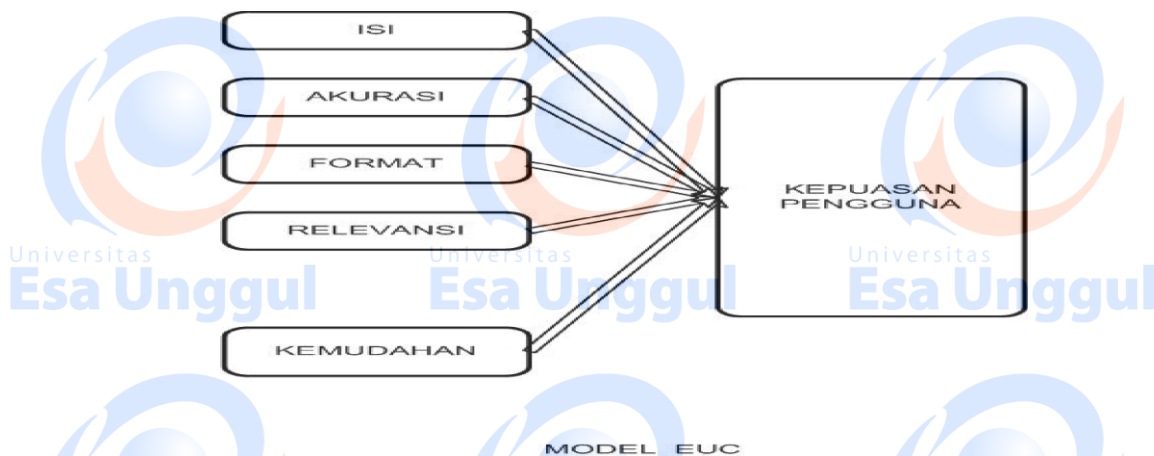
- d. Evaluasi ekonomi yaitu manfaat ekonomis sistem informasi perlu mengelompokkan biaya dan manfaat ke dalam kelompok perkiraan dengan perbedaan yang kecil, perbedaan sedang dan perbedaan besar. Pemasangan aplikasi, perlu ditelaah secara periodik menurut ukuran biaya/efektivitas. Dalam menilai kelayakan ekonomis dari proyek dan mengevaluasi manfaat ekonomis dan sistem informasi manajemen.

Model Evaluasi Sistem Informasi

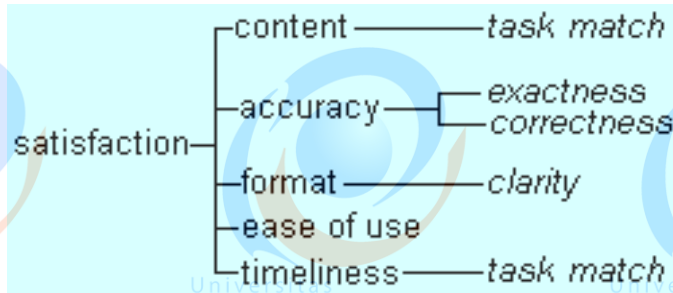
Evaluasi sistem informasi (SI) dapat dilakukan dengan cara berbeda dan pada tingkatan berbeda, tergantung pada tujuan evaluasi. Tujuan evaluasi system yaitu, untuk Menilai Kemampuan Teknis Suatu Sistem, Pelaksanaan Oprasional dan Pendayagunaan Sistem

Model untuk mengevaluasi suatu sistem bermacam-macam tergantung dengan jenis penelitiannya, berikut ini adalah berbagai model evaluasi sistem informasi:

1. End User Computing (EUC) Satisfaction

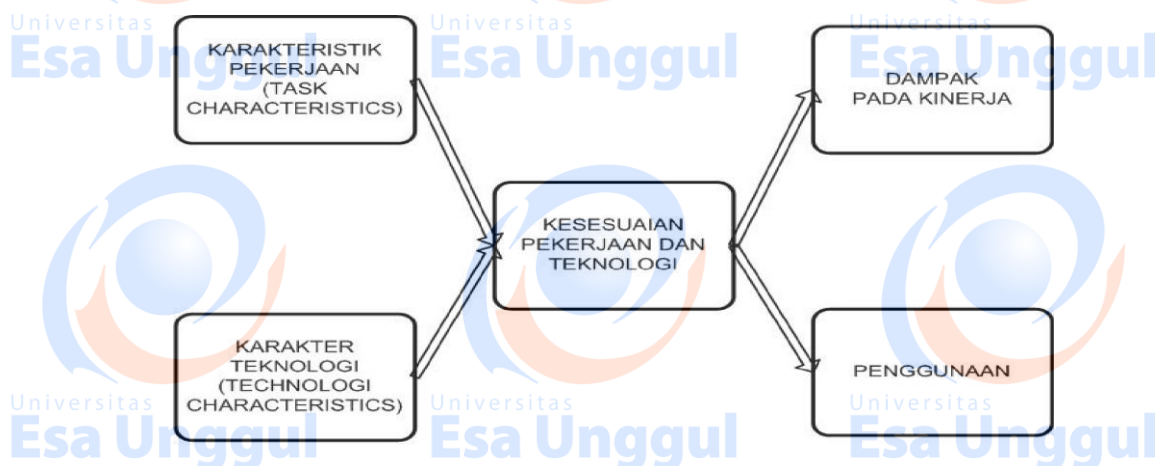


Pengukuran terhadap kepuasan telah mempunyai sejarah yang panjang dalam disiplin ilmu sistem informasi. Dalam lingkup EUC sejumlah studi telah dilakukan meng-capture keseluruhan evaluasi, yaitu pengguna akhir telah menganggap pengguna dari suatu sistem informasi (misalnya kepuasan) dan juga faktor-faktor yang membentuk kepuasan (Nugroho, 2008). Model evaluasi ini dikembangkan oleh Doli & Torkzadeh, dimana lebih menekankan pada kepuasan (satisfaction) pengguna akhir terhadap aspek teknologi, dengan menilai isi, keakuratan, format, waktu dan kemudahan penggunaan sistem.



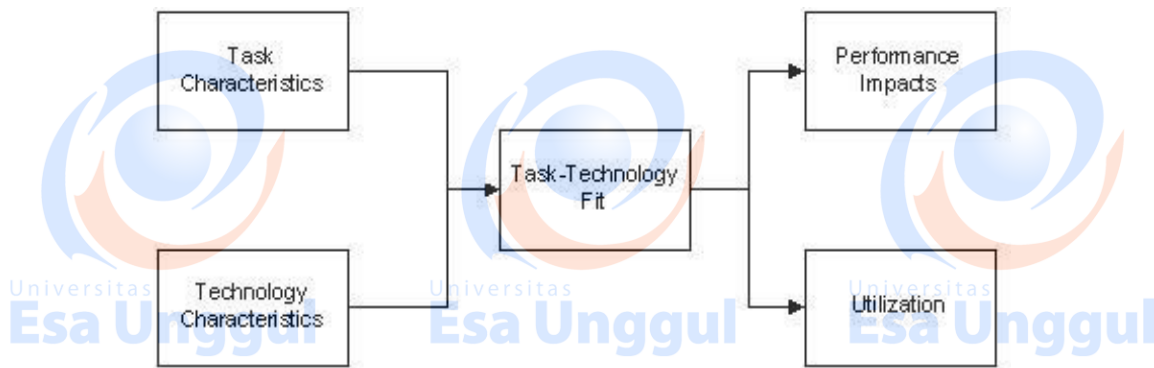
Model evaluasi system dengan menggunakan EUC lebih menekankan kepuasan (satisfaction) pengguna akhir terhadap aspek teknologi, dengan menilai isi, keakuratan, format, waktu & kemudahan penggunaan dari sistem. Model ini telah banyak diujicobakan oleh peneliti lain untuk menguji reliabilitasnya & hasilnya menunjukkan tidak ada perbedaan bermakna meskipun instrumen ini diterjemahkan dalam berbagai bahasa yang berbeda.

2. Task Technology Fit (TTF) Analysis



MODEL TTF

Model TTF adalah sebuah konstruk formal yang merupakan kesesuaian dari kapabilitas teknologi untuk kebutuhan tugas dalam pekerjaan yaitu kemampuan teknologi informasi untuk memberikan dukungan terhadap pekerjaan. Model TTF memiliki 4 konstruk kunci yaitu Task Characteristic, Technology Characteristic, yang bersama-sama mempunyai konstruk ketiga TTF yang balik mempengaruhi variabel outcome yaitu Performance atau Utilization. Model TTF menyatakan bahwa teknologi informasi hanya akan digunakan jika fungsi dan manfaatnya tersedia untuk mendukung aktifitas pengguna.

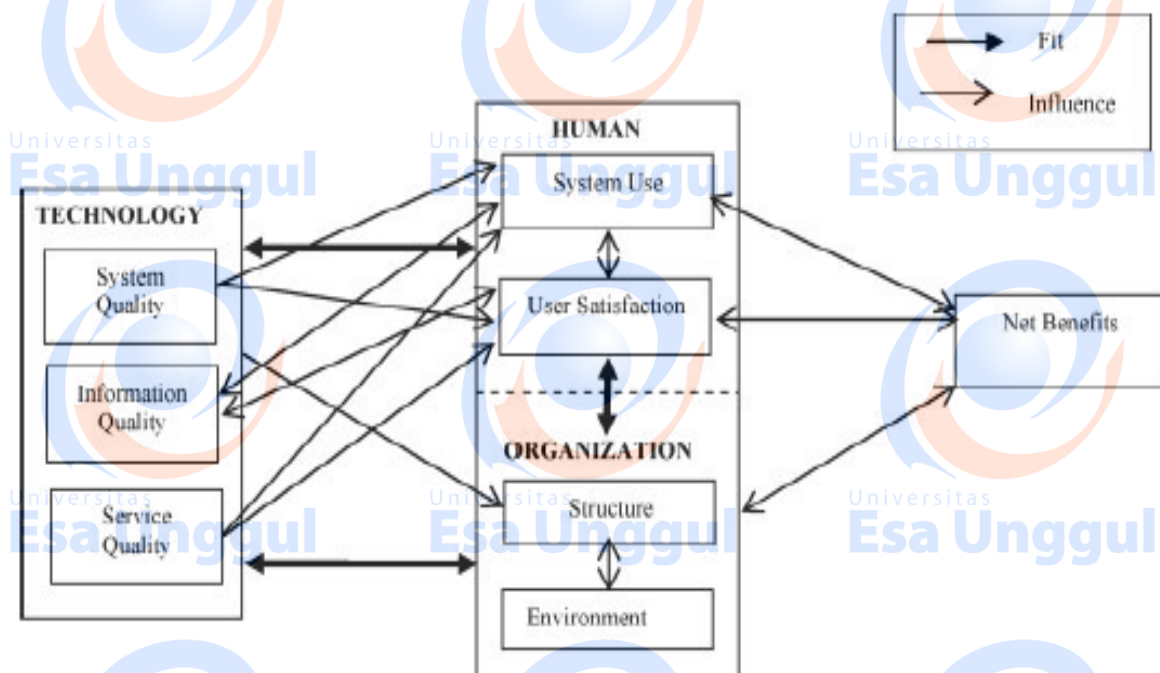


Source: Goodhue and Thompson, (1995)

Model evaluasi ini pertama kali dikembangkan oleh Goodhue dan Thompson pada tahun 1995. Teori ini berpegang bahwa teknologi informasi memiliki dampak positif terhadap kinerja individu dan dapat digunakan jika kemampuan teknologi informasi cocok dengan tugas-tugas yang harus dihasilkan oleh pengguna (Furneaux, 2006a)

3. Human-Organization-Technology (HOT) Fit Model

Model ini menempatkan komponen penting dalam sistem informasi, yakni manusia (human), organisasi (organization), dan teknologi (technology) serta kesesuaian hubungan diantaranya. Hot fit model diakomodir variabel struktur dan lingkungan organisasi dimana variabel tersebut tidak terdapat pada model sebelumnya.

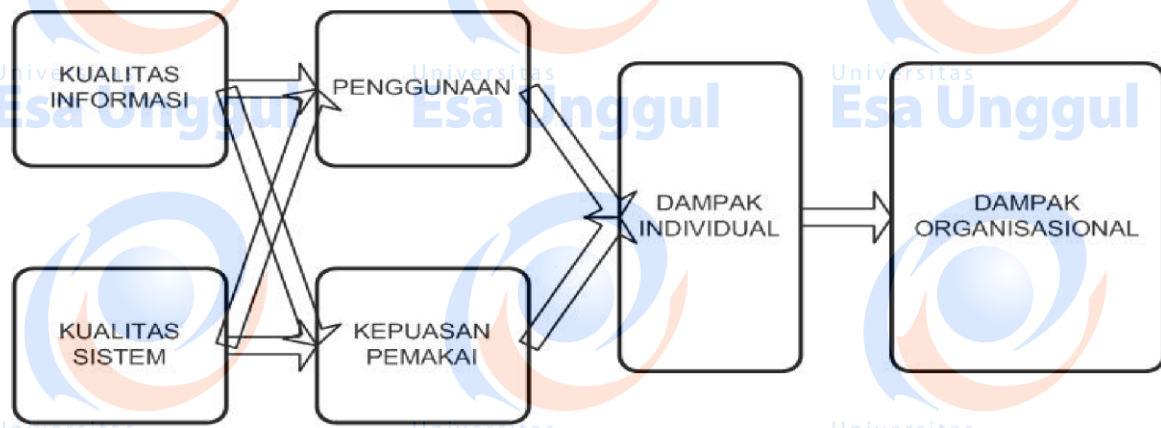


- a. Komponen manusia (human) komponen manusia menilai sistem informasi dari sisi penggunaan sistem (system use) pada frekwensi dan luasnya fungsi dan penyelidikan sistem informasi. System use juga berhubungan dengan siapa yang menggunakan (who use it), tingkat penggunaanya (level of user), pelatihan, pengetahuan, harapan dan sikap menerima (acceptance) atau menolak (resistance) sistem. Komponen ini juga menilai sistem dari aspek kepuasan

pengguna (user satisfaction). Kepuasan pengguna adalah keseluruhan evaluasi dari pengalaman pengguna dalam menggunakan sistem informasi dan dampak potensial dari sistem informasi. User satisfaction dapat dihubungkan dengan persepsi manfaat (usefulness) dan sikap pengguna terhadap sistem informasi yang dipengaruhi oleh karakteristik personal.

- b. Komponen organisasi (organization) kepemimpinan, dukungan dari top manajemen dan dukungan staf merupakan bagian yang penting dalam mengukur keberhasilan sistem. Sedangkan lingkungan organisasi terdiri dari sumber pembiayaan, pemerintahan, politik, kompetisi, hubungan interorganisasional dan komunikasi.
- c. Komponen teknologi (technology) komponen teknologi terdiri dari kualitas sistem (system quality), kualitas informasi (information quality) dan kualitas layanan (service quality). Kualitas sistem dalam sistem informasi menyangkut keterkaitan fitur dalam sistem termasuk performa sistem dan user interface. Kemudahan penggunaan (ease of use), kemudahan untuk dipelajari (ease of learning), response time, usefulness, ketersediaan, fleksibilitas, dan sekuritas merupakan variabel atau faktor yang dapat dinilai dari kualitas sistem. Kriteria yang dapat digunakan untuk menilai kualitas informasi antara lain adalah kelengkapan, keakuratan, ketepatan waktu, ketersediaan, relevansi, konsistensi, dan data entri. Sedangkan kualitas layanan berfokus pada keseluruhan dukungan yang diterima oleh service provider sistem atau teknologi. Service quality dapat dinilai dengan kecepatan respon, jaminan, empati dan tindak lanjut layanan (Yusof et al, 2008).

4. Model DeLone & McLean



MODEL KESUKSESAN SISTEM INFORMASI DARI DELONE DAN MCLEAN

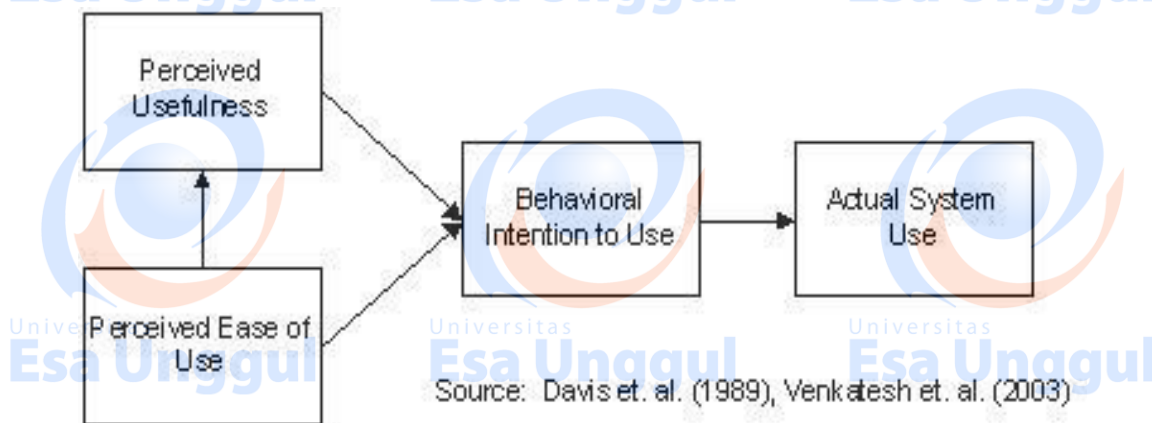
Menurut DeLone & McLean agar sistem informasi sukses, mempunyai dampak positif terhadap organisasi maka terlebih dahulu sistem informasi harus mempunyai dampak pada individual. Agar mempunyai dampak terhadap individual, maka kepuasan pemakai harus tercapai, disamping bahwa sistem informasi sudah mulai dipergunakan secara rutin operasional. Agar kedua hal tersebut tercapai, maka kualitas sistem dan kualitas informasi haruslah bagus terlebih dahulu (Nugroho, 2008).

Dalam perkembangannya, DeLone & McLean menyempurnakan modelnya dengan menambah tahapan kualitas pelayanan, tahapan penggunaan diubah menjadi intense penggunaan, manfaat pada individual dan pada organisasi diubah menjadi dampak keberhasilan secara neto.

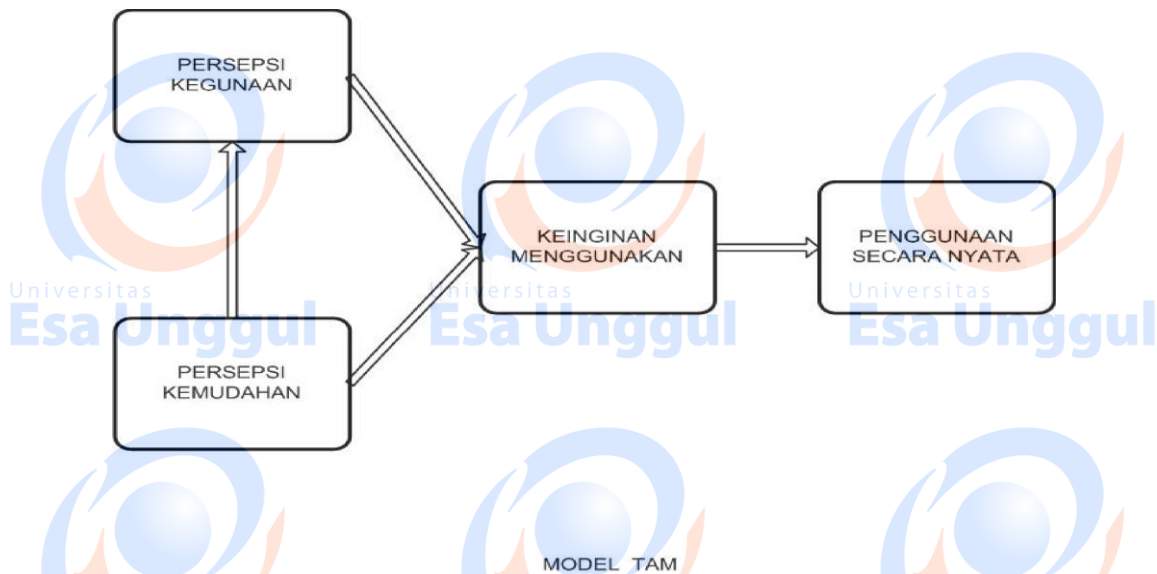
faktor-faktor kesuksesan system informasi :

- a. Kualitas Sistem (System Quality): Kualitas dari kombinasi hardware & software dalam system informasi.
- b. Kualitas Informasi (Information Quality): Merujuk pada output system, menyangkut nilai, manfaat, relevansi, & urgensi dari informasi yang dihasilkan.
- c. Intensitas Penggunaan Sistem Informasi: Seberapa sering user menggunakan system
- d. User Satisfaction: Respon/feedback user setelah memakai system (kepuasan)
- e. Individual impact: pengaruh keberadaan & pemakaian system terhadap kualitas kinerja user secara individual termasuk produktivitas, efisiensi & efektivitas kinerja.

5. Technology Acceptance Model (TAM)



Model penerimaan teknologi atau TAM ini pertama kali diperkenalkan oleh Davis pada tahun 1989. TAM adalah sistem informasi yang membuat model tentang bagaimana pengguna bersedia menerima dan menggunakan teknologi. Model ini mengusulkan bahwa ketika pengguna ditawarkan untuk menggunakan suatu sistem yang baru, sejumlah faktor mempengaruhi keputusan mereka tentang bagaimana dan kapan akan menggunakan sistem tersebut, khususnya dalam hal usefulness pengguna yakin bahwa dengan menggunakan sistem ini akan meningkatkan kinerjanya, ease of use (pengguna yakin bahwa menggunakan sistem ini akan membebaskannya dari kesulitan, dalam artian bahwa sistem ini mudah dalam penggunaannya).



TAM yang memiliki elemen yang kuat tentang perilaku (behavioral) mengasumsikan bahwa ketika seseorang membentuk suatu bagian untuk bertindak, mereka akan bebas untuk bertindak tanpa batasan. Beberapa penelitian telah mereplikasi studi Davis untuk memberi bukti empiris terhadap hubungan yang ada antara usefulness, ease of use dan system use (Furneaux, 2006b).

Kelebihan model TAM:

- a. TAM merupakan model perilaku (behavior) yang bermanfaat untuk menjawab pertanyaan mengapa banyak sistem teknologi gagal diterapkan karena pemakainya tidak mempunyai minat (intention) untuk menggunakannya.
- b. TAM dibangun dengan dasar teori yang kuat.
- c. TAM telah diuji dengan banyak penelitian.
- d. TAM merupakan model yang parsimony (parsimonius) yaitu model yang sederhana dan valid.

Kelemahan model TAM:

- a. Hanya memberikan informasi/hasil yang sangat umum saja tentang minat & perilaku pemakai sistem dalam menerima system informasi.
- b. Hanya menjelaskan kepercayaan (beliefs) mengapa pemakai mempunyai minat perilaku menggunakan sistem.
- c. Belum memberikan informasi menjelaskan mengapa pemakai sistem mempunyai kepercayaan-kepercayaan tersebut.



Topik 3

PLS (*Partial Least Squares*)

Structural Equation Modelling (SEM) merupakan salah satu metode yang saat ini digunakan untuk menutup kelemahan yang ada pada metode regresi. Para ahli metode penelitian mengelompokkan SEM menjadi dua pendekatan. Pendekatan pertama disebut sebagai *Covariance Based SEM (CBSEM)* dan pendekatan lainnya adalah *Variance Based SEM* atau yang lebih dikenal dengan Partial Least Squares (PLS). Untuk melakukan analisa dengan menggunakan CBSEM maka software yang sering digunakan adalah AMOS dan LISREL sedangkan untuk PLS software yang sering digunakan adalah smartPLS, warpPLS dan XLStat.

Modul ini membahas mengenai panduan melakukan analisa dalam penelitian dengan menggunakan teknik SEM. Khususnya, modul ini akan membahas mengenai alasan menggunakan PLS, melakukan evaluasi outer model (*outer model evaluation*), evaluasi inner model (*inner model evaluation*), analisa jalur, pengujian mediasi/pengaruh tidak langsung dan perbandingan multi group.

Mengapa Menggunakan SEM

Sebelum mengoperasikan PLS sebagai bagian dari SEM, maka alangkah lebih baiknya jika dipahami terlebih dahulu alasan dan tujuan melakukan analisa dengan menggunakan PLS. Seperti telah diuraikan sebelumnya bahwa pada metode regresi terdapat kelemahan-kelemahan yang mengurangi komprehensifitas dari analisis. Berikut ini akan dibahas mengenai kelemahan-kelemahan tersebut.

Asumsi Normalitas Data

Dalam metode analisa menggunakan regresi, terdapat asumsi-asumsi yang harus diperiksa oleh peneliti untuk memastikan bahwa persamaan regresi yang dibentuk BLUE (Best Linier Unbiased Estimate). Salah satu asumsi yang sering kali menjadi sandungan peneliti adalah asumsi normalitas.

PLS menggunakan metode bootstrapping atau penggandaan secara acak. Oleh karenanya asumsi normalitas tidak akan menjadi masalah bagi PLS. Selain terkait dengan normalitas data, dengan dilakukannya bootstrapping maka PLS tidak mensyaratkan jumlah minimum sample. Penelitian yang memiliki sampel kecil dapat tetap menggunakan PLS.

Melakukan Analisa Dengan Variabel Dependen Lebih Dari Satu

Secara umum persamaan regresi dapat ditulis sebagai berikut:

Persamaan regresi tersebut menunjukkan bahwa:

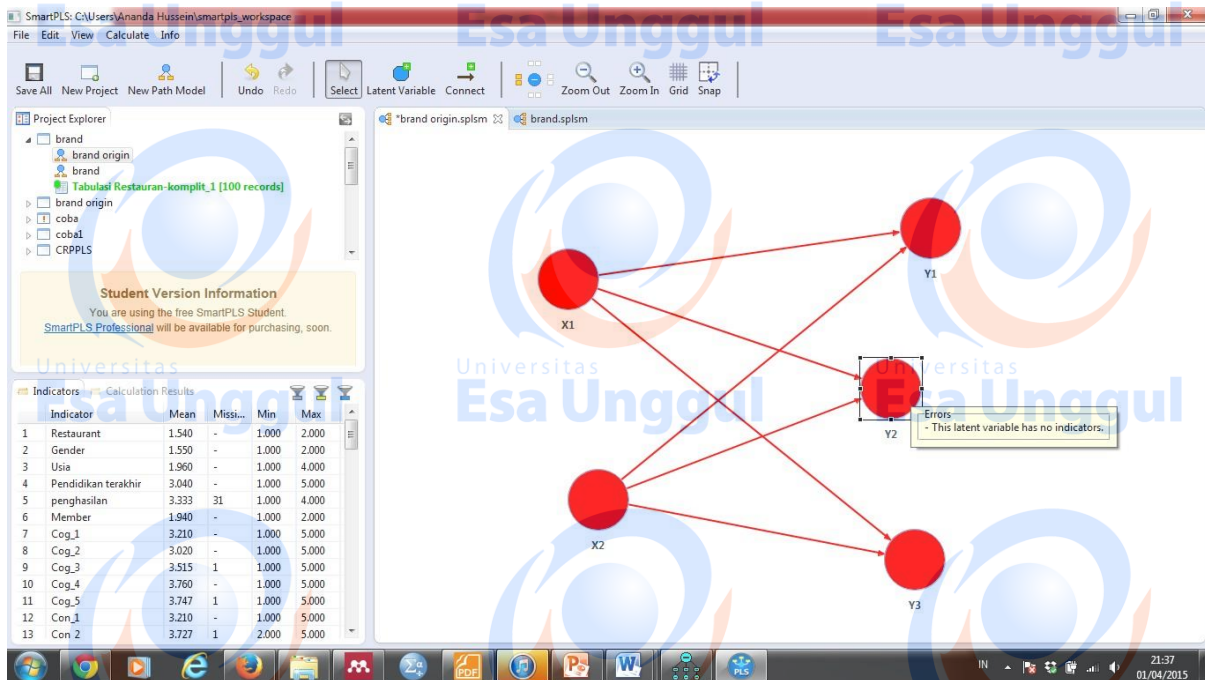
Y = variabel dependen

X = variabel independen

Tampak dalam persamaan tersebut bahwa terdapat hanya satu variabel terikat/dependen dan banyak variabel bebas/independen. Yang menjadi pertanyaan adalah bagaimana melakukan analisa jika variabel dependennya lebih dari satu.

Dengan menggunakan pendekatan jalur (path) PLS akan mampu memberikan analisa jika variabel dependennya lebih dari satu.

Gambar 1. Analisa Jalur Dengan Variabel Dependen Lebih Dari Satu



Memberikan Gambaran Pengaruh Langsung dan Tidak Langsung

Pada analisa multivariate seperti halnya regresi linier berganda ataupun Multivariate ANOVA (MANOVA), pengujian hanya dilakukan secara langsung. Metode-metode tersebut hanya mampu memberikan analisa mengenai pengaruh variabel bebas terhadap variabel terikat.

Persamaan tersebut memberikan informasi bahwa variabel bebas X memiliki pengaruh terhadap Y. Akan tetapi dalam pengujian teori banyak pendugaan yang menyatakan bahwa pengaruh suatu konstruk/variabel itu dimediasi oleh sebuah variabel medias.

Secara manual konsep pengujian mediasi dapat digunakan. Akan tetapi karena basis pengujiannya adalah regresi maka komprehensifitas nya akan berkurang jika dibandingkan SEM yang menguji secara serentak.

Perbedaan PLS dengan CSEM

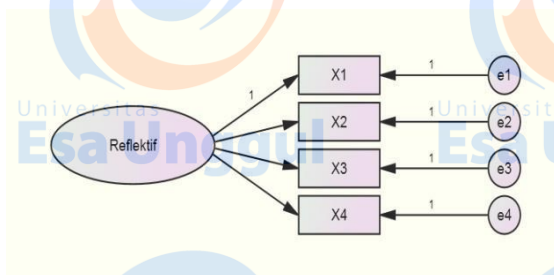
Banyak yang mengatakan bahwa PLS bukanlah SEM. Akan tetapi sebenarnya PLS itu adalah SEM yang berbasis varians. Jika PLS juga SEM maka timbul pertanyaan baru yaitu apa perbedaan PLS dengan SEM yang menggunakan program AMOS atau LISREL.

Walaupun sama-sama dapat dikategorikan sebagai SEM, PLS dan CBSEM

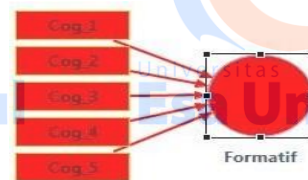
memiliki perbedaan yang jelas. Hal utama yang membedakan antara PLS dan CBSEM adalah tujuan dari penggunaan metode. Tujuan dari penggunaan dari PLS adalah melakukan prediksi. Prediksi yang dimaksud disini adalah prediksi hubungan antar konstruk. Berbeda dengan PLS yang bertujuan untuk melakukan prediksi, penggunaan CBSEM lebih ditujukan sebagai metode untuk melakukan konfirmasi teori. Sehingga dapat dikatakan bahwa PLS lebih berorientasi kepada prediksi sedangkan CBSEM berorientasi pada teori.

Berdasarkan asumsi statistiknya, PLS digolongkan sebagai jenis non-parametrik sedangkan CBSEM lebih kepada Multivariate normal distribution dan independent observation (parametrik). Oleh karena itu dalam pemodelan PLS tidak diperlukan data dengan distribusi normal.

Dari sisi konstruk, CBSEM hanya dapat mengakomodir konstruk yang berbentuk reflektif. Sedangkan PLS dapat mengakomodir baik formatif maupun reflektif.



Gambar 3
Konstruk Reflektif



Gambar 4
Konstruk Formatif

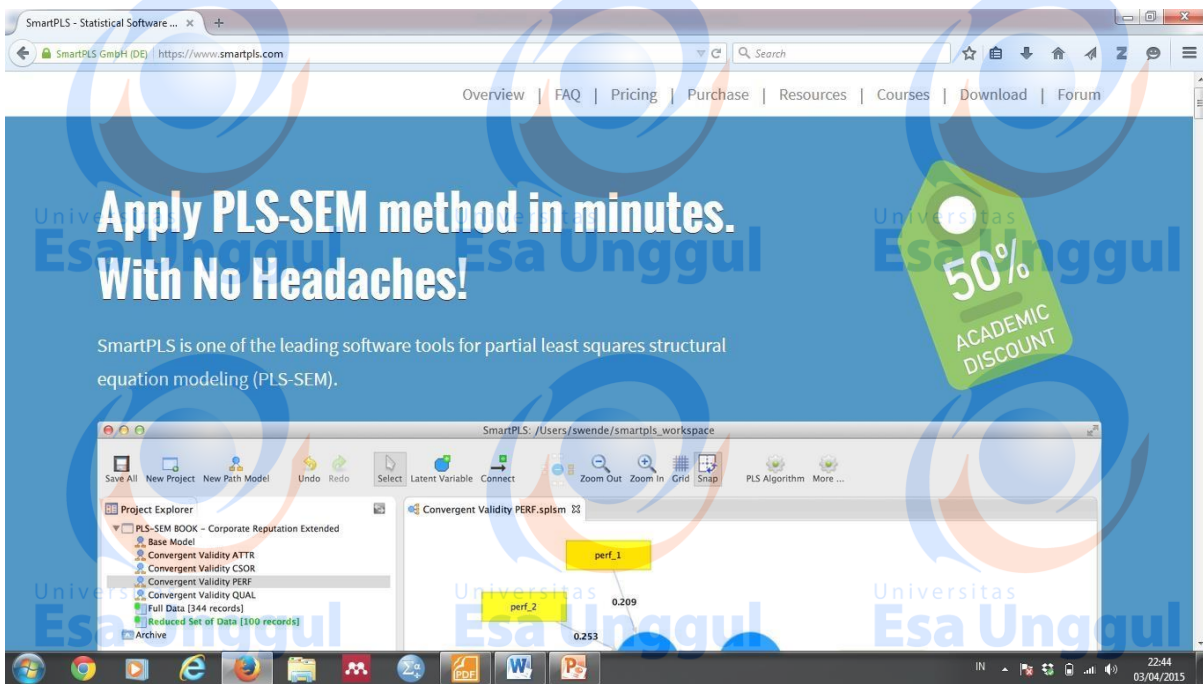
Perbedaan lainnya antara CBSEM dengan PLS adalah mengenai jumlah sampel. Dikarenakan PLS ber basis pada variance maka jumlah sampel yang digunakan tidak perlu besar. Sampel yang digunakan dapat berkisar antara 30 – 100. Sedangkan CBSEM, karena dasarnya adalah covariance, maka sampel yang digunakan berkisar 200-800 sampel.

Untuk jumlah konstruk dan indikator yang digunakan pun PLS dan CBSEM berbeda. PLS dapat mengakomodir hingga 100 konstruk dan 1000 indikator sedangkan CBSEM hanya bisa mengakomodir maksimal 100 indikator.

Mengoperasikan Program smartPLS 3.0

Untuk melakukan analisa dengan PLS salah satu program yang bisa digunakan adalah smartPLS 3.0. Untuk mendapatkannya bisa download langsung di website nya www.smartPLS.de.

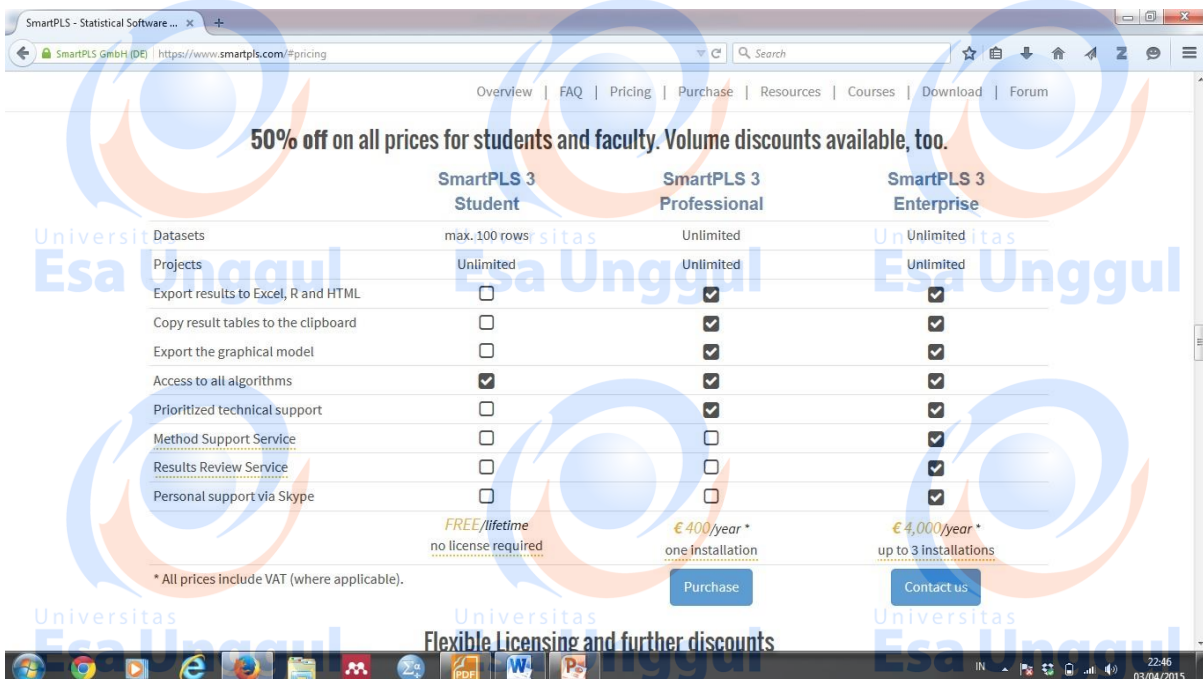
Gambar 2. Tampilan Web smartPLS



Setelah membuka web dari smartPLS maka program bisa didownload pada segmen download.

Terdapat tiga pilihan yang ditawarkan oleh pengembang program ini. Yang pertama versi student. Versi ini dapat didownload secara gratis tetapi terbatas hanya maksimum 100 observasi. Selain itu output tidak bisa di copy. Pilihan lainnya adalah smartPLS 3.0 professional dan smartPLS 3.0 enterprise. Masing-masing fitur yang ditawarkan dapat dilihat pada gambar dibawah ini.

Gambar 3. Pilihan Program smartPLS 3.0

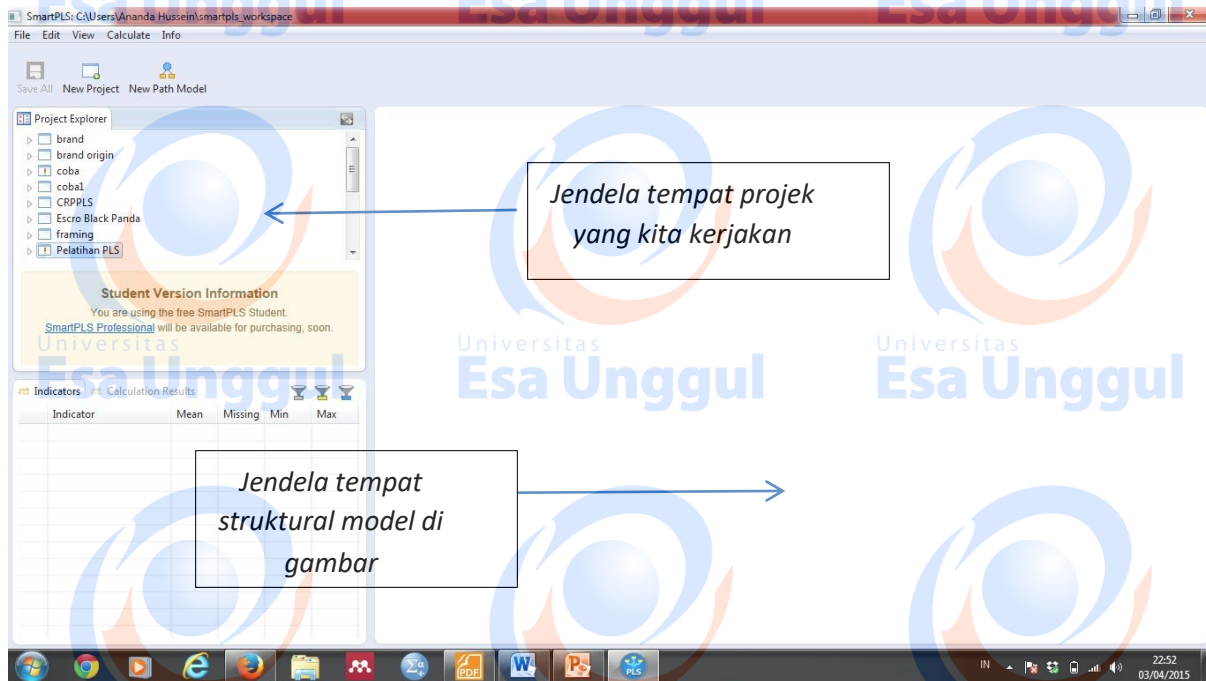


Setelah program didownload maka di desktop akan muncul icon smartPLS 3.0



Menggambar Model dan Memasukkan Data

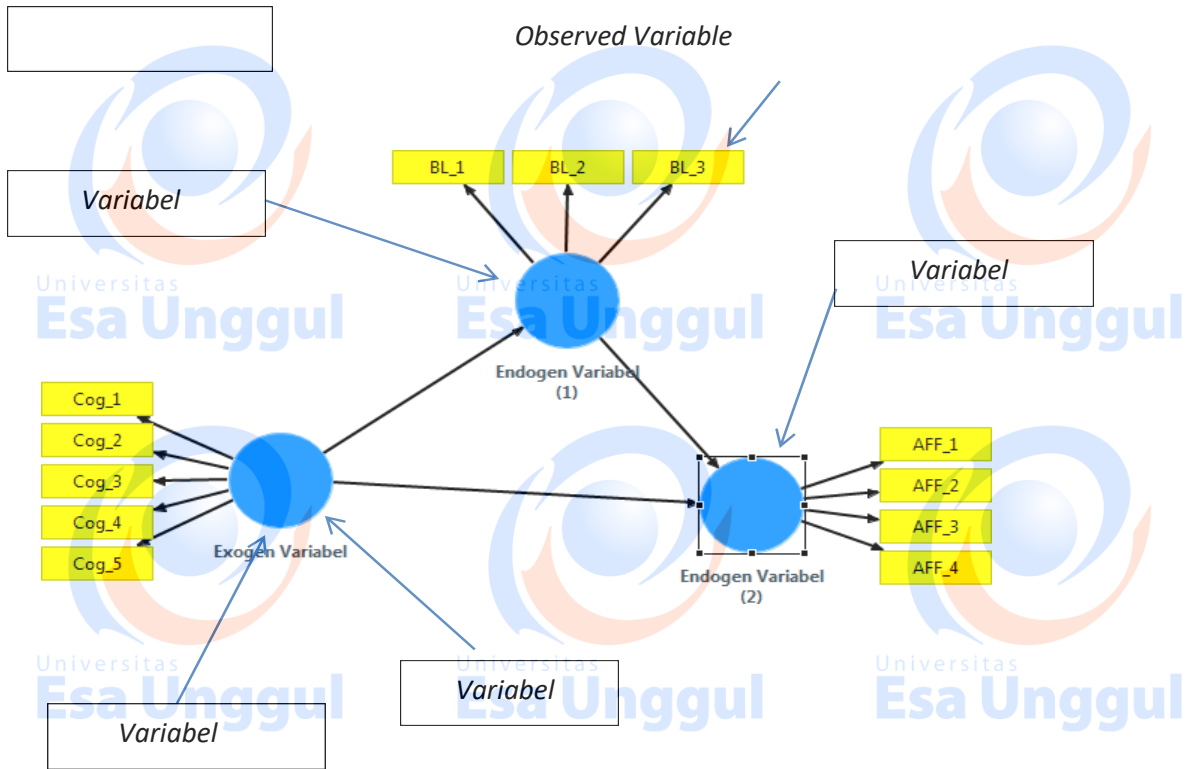
Ketika baru pertama kali membuka program PLS dengan smartPLS 3.0 maka tampilan yang akan pertama kali dilayar adalah sebagai berikut:



Sebelum menggambar ada beberapa istilah yang harus diketahui:

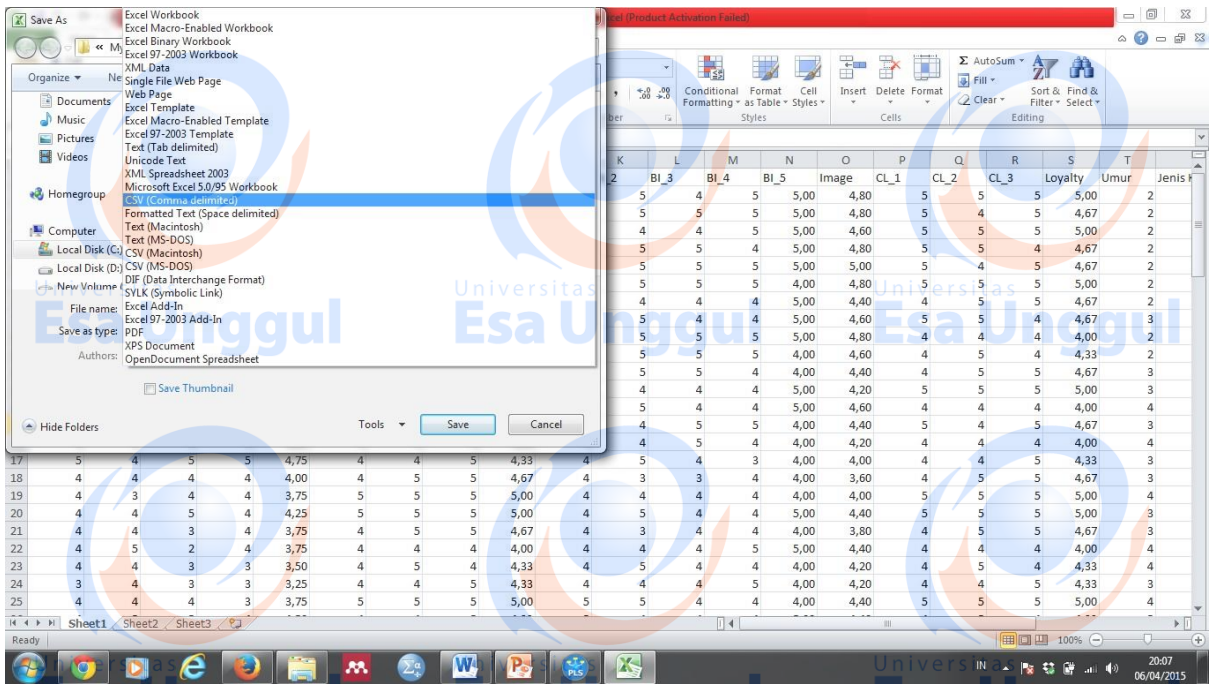
Istilah	Pengertian
Konstruk Exogen	Variabel yang tidak diprediksi oleh variabel-variabel yang lain yang terdapat dalam model. Konstruk eksogen adalah konstruk yang dituju oleh garis dengan satu ujung anak panah.
Konstruk Endogen	yang merupakan faktor-faktor yang diprediksi oleh satu atau beberapa konstruk. Konstruk endogen dapat memprediksi satu atau beberapa konstruk endogen lainnya, tetapi konstruk endogen hanya dapat berhubungan kausal dengan konstruk endogen.
Latent Variabel	Variabel laten adalah variabel yang nilai kuantitatifnya tidak dapat diketahui secara tampak.

<p>Observed Variabel</p>	<p>Variabel manifest adalah variabel yang besar kuantitatifnya dapat diketahui secara langsung misalnya dari skor respons subjek terhadap instrumen pengukuran.</p>
--------------------------	---

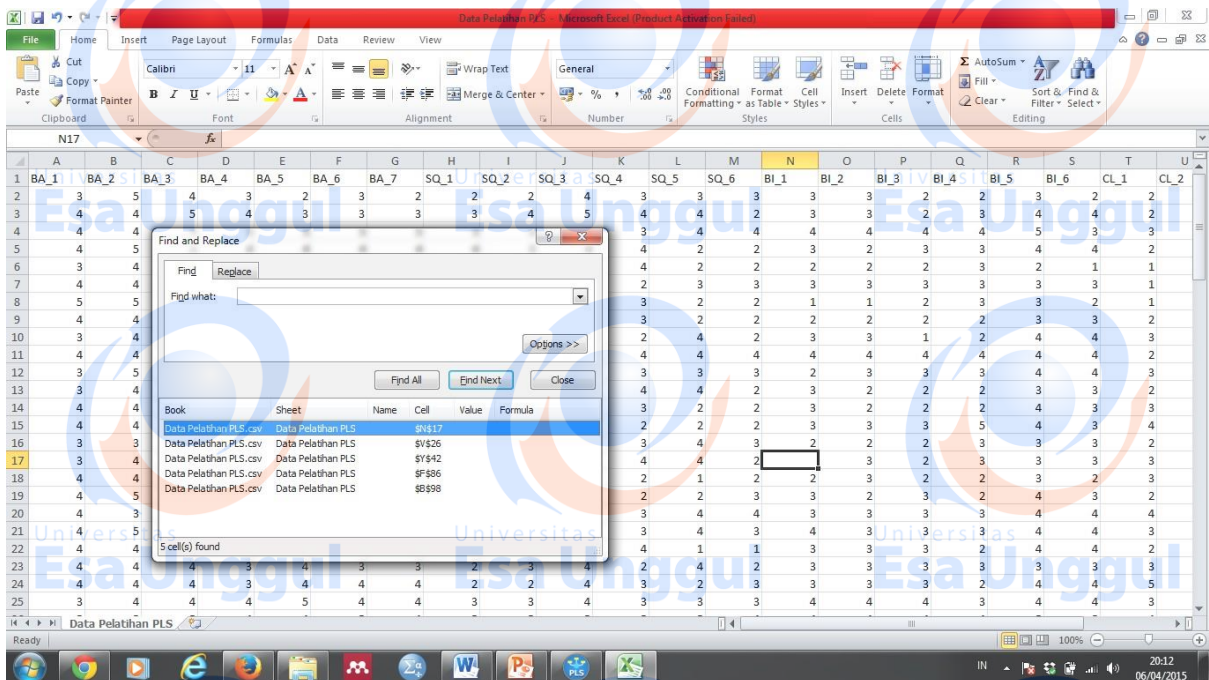


Gambar di atas menunjukkan mana saja yang merupakan variabel endogen dan eksogen serta laten dan observend variabel.

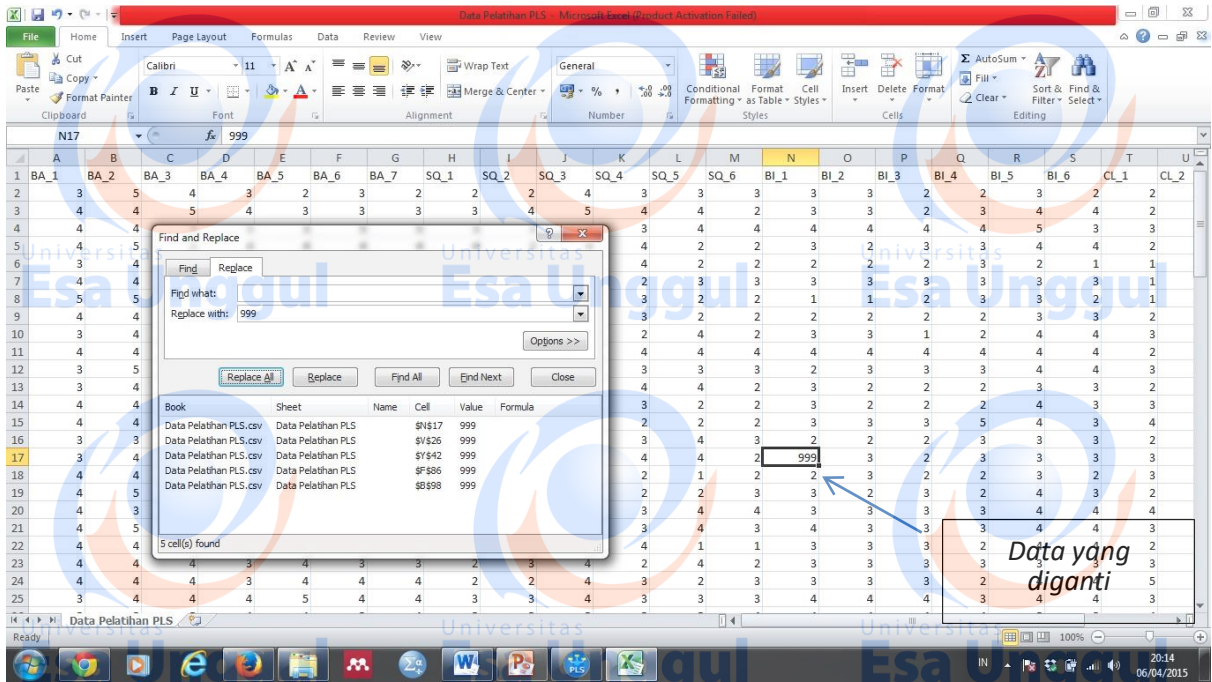
Sebelum mulai menggambar model struktural yang harus dilakukan adalah menginput file yang akan kita analisis. Untuk menjalankan smartPLS 3.0 data harus disimpan dalam format .csv.



Setelah data disimpan dalam format csv sebelum dimasukkan/ di input ke dalam smartPLS 3.0 alangkah lebih baik nya jika data di cek terlebih dahulu apakah ada missing value atau tidak.

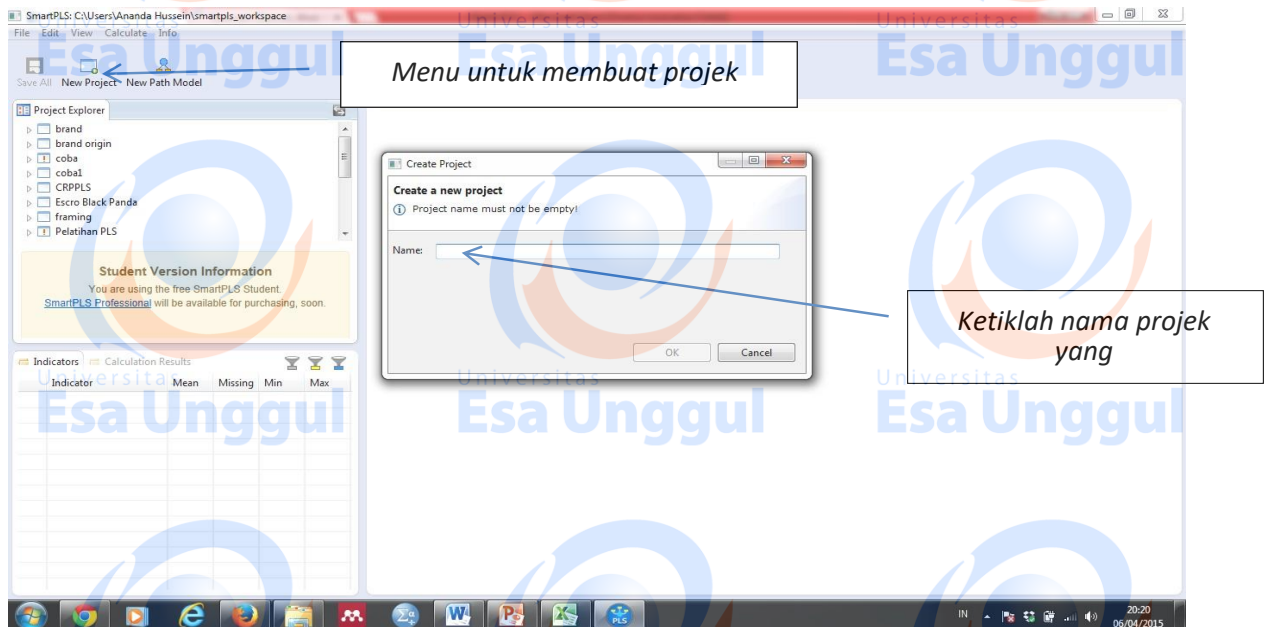


Memeriksa missing value bisa dilakukan secara manual. Akan tetapi jika data berjumlah banyak maka lebih baik gunakan menu find yang disediakan oleh excel. Jika didapati ada data yang missing. Maka beri tandalah data missing tersebut. Simbol yang sering digunakan untuk mengidentifikasi missing data adalah 999. Maka gantilah data yang kosong dengan 999.

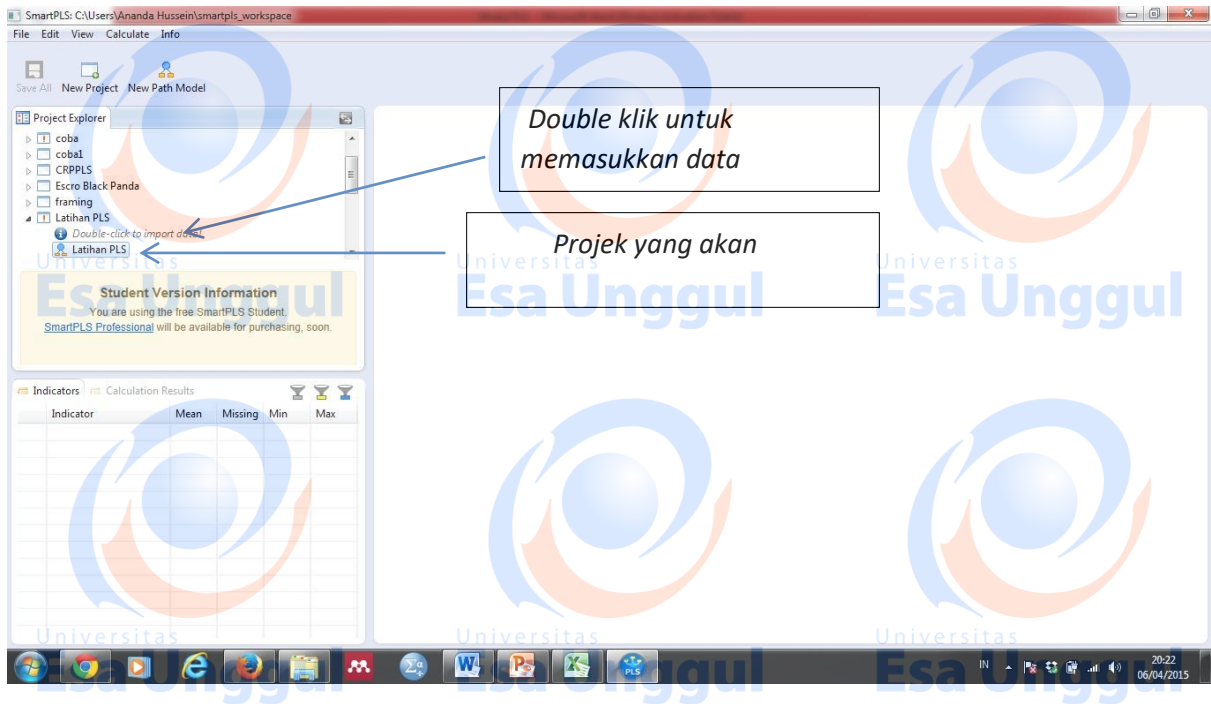


Jika data sudah di pastikan tidak ada yang missing maka langkah berikutnya adalah memasukkan data ke dalam program smartPLS 3.0.

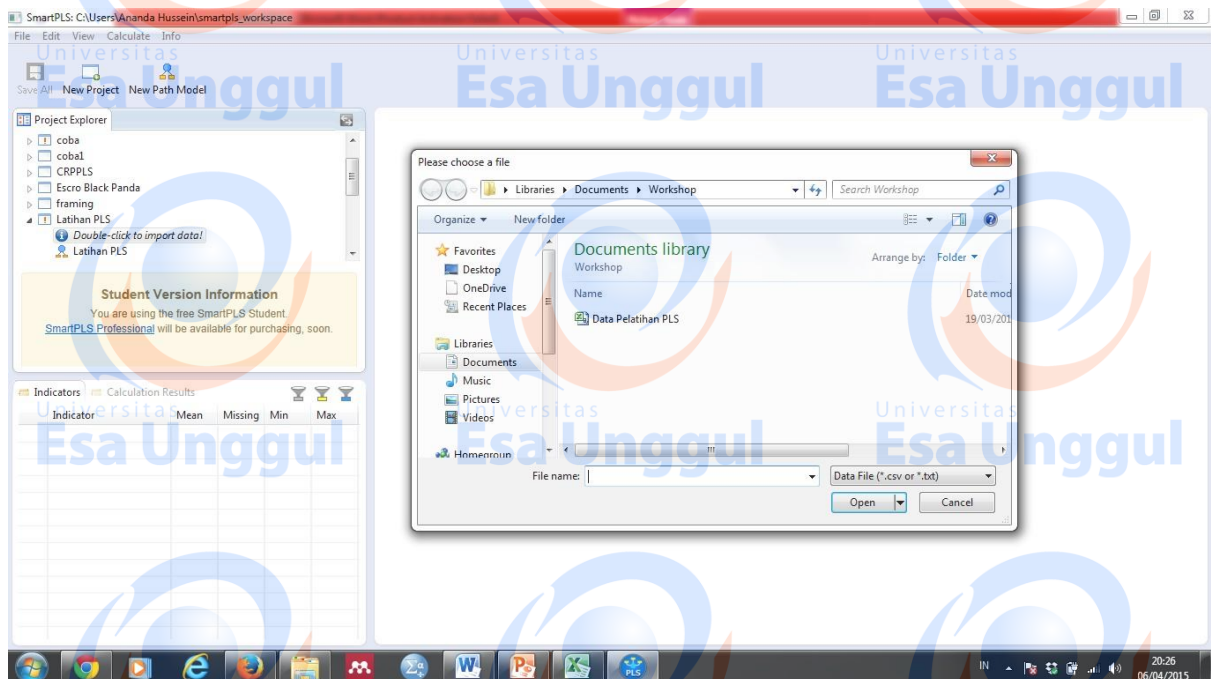
Langkah pertama untuk memasukkan data ke dalam smartPLS 3.0 adalah dengan cara membuat proyek yang akan kita kerjakan. Disudut kiri atas ada pilihan new project.



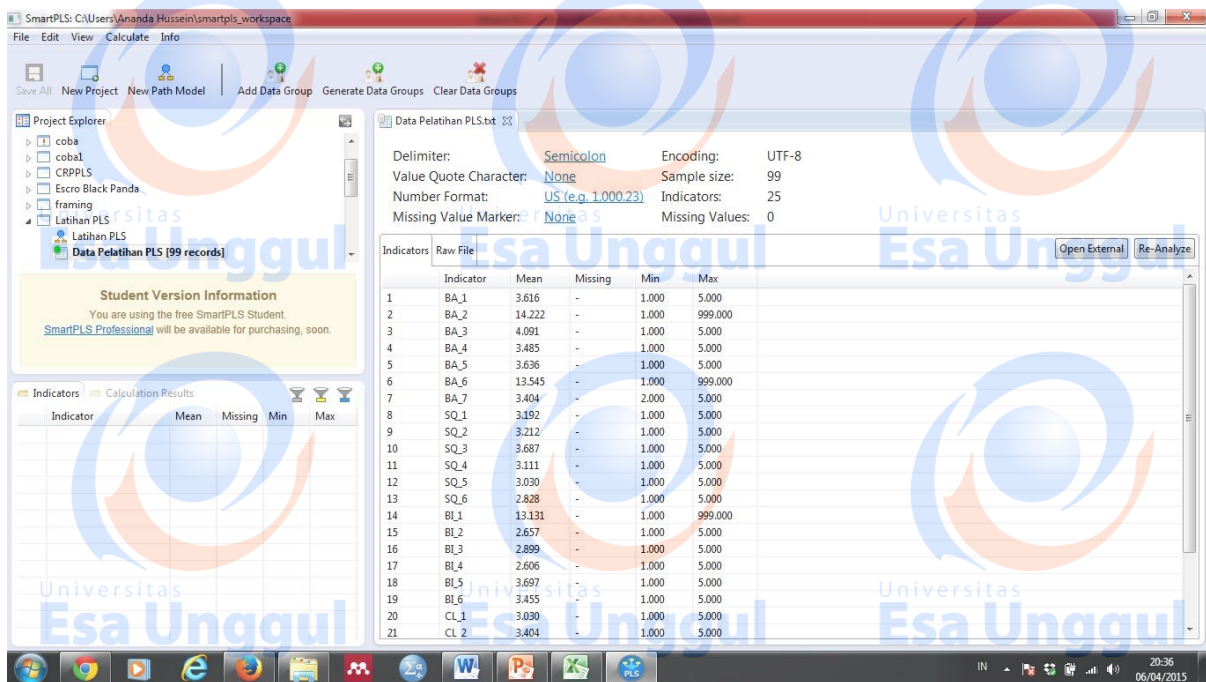
Setelah diketikkan nama proyek dan di klik tombol OK maka proyek akan muncul pada layar di sebelah kiri.



Untuk memasukkan data maka double klik icon menu yang berada di atas nama file proyek yang ingin dikerjakan. Setelah di double klik maka akan muncul layar yang berisikan file dimana kita menaruh data mentah. Jika layar yang dibuka bukan tempat kita menaruh data mentah yang akan dikerjakan maka kita bisa mencari pada folder-folder lainnya.



Jika data sudah dimasukkan maka akan tampak dilayar smartPLS yang dibuka.

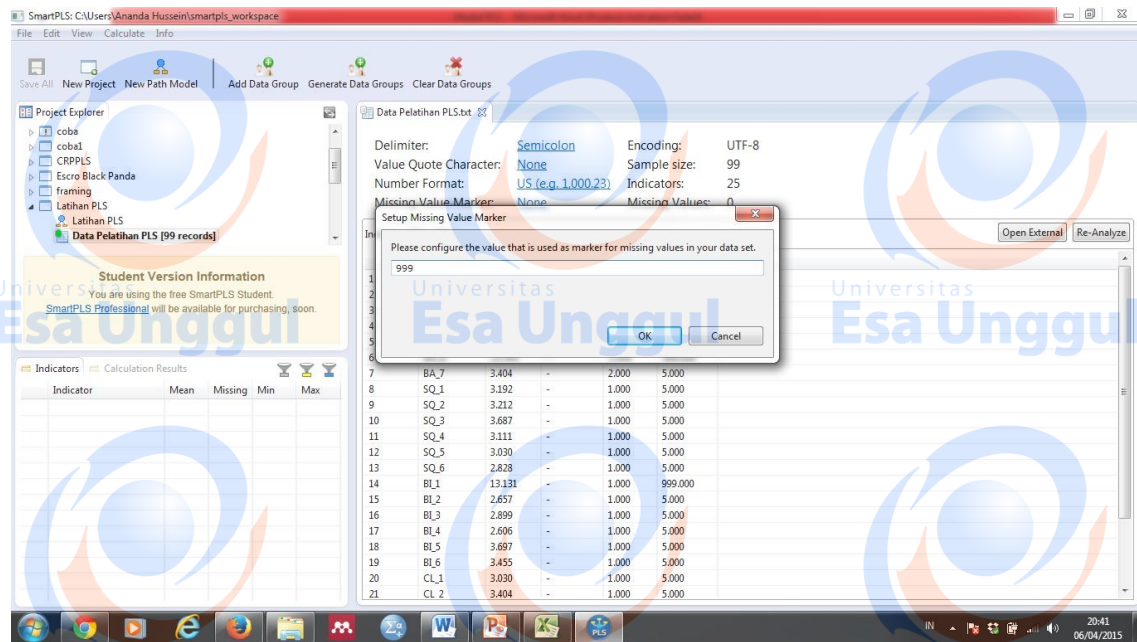


Tampilan layar tersebut menunjukkan bahwa data mentah yang kita inputkan sudah masuk kedalam program smartPLS. Indikator lainnya jika data sudah masuk secara sempurna adalah icon warna hijau di bawah nama proyek yang kita kerjakan. Jika warna icon tersebut hijau maka berarti data sudah komplit. Tetapi jika icon berwarna kuning maka maknanya data masing mengandung missing value. Missing value tersebut harus diberi kode 999 seperti kode missing value lainnya.

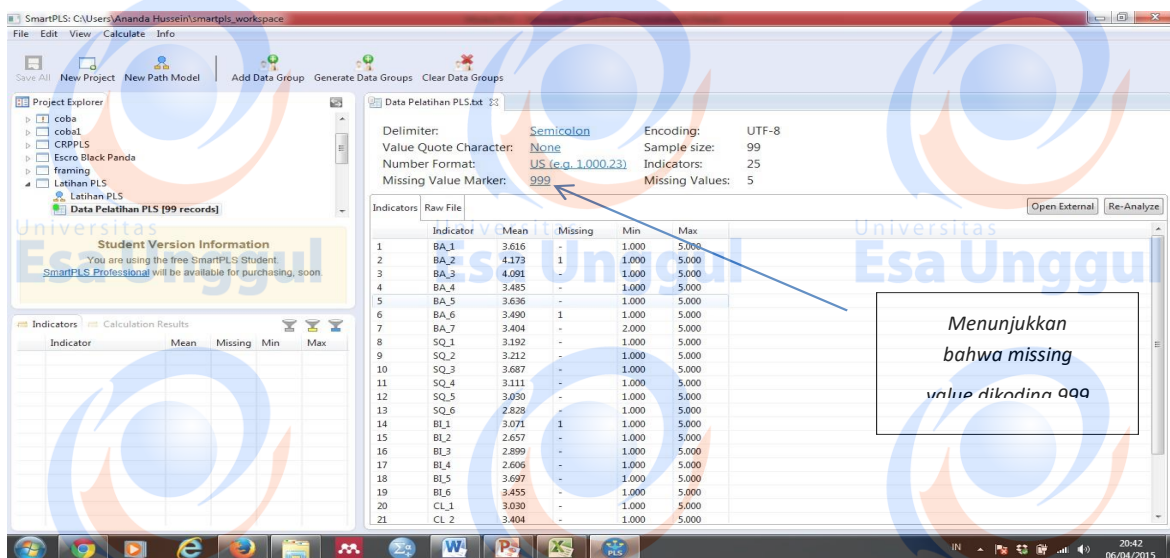
Di layar tengah dapat dilihat bahwa ada deretan angka. Deretan angka tersebut menunjukkan indikator yang kita gunakan, mean/rata-rata, nilai minimal dan nilai maksimal. Di kolom maksimal terlibat beberapa angka tertulis nilai maksimal adalah 999. Padahal nilai 999 merupakan koding dari missing value. Oleh karena itu harus diinputkan bahwa 999 adalah koding.

Pada layar di atas nya ada tulisan Missing Value Marker. Klik tulisan tersebut. Setelah di klik akan muncul sebuah layar baru. Maka isikan 999 pada layar tersebut. Pengisian angka 999 tersebut memberikan informasi kepada program smartPLS 3.0 bahwa angka 999 adalah koding untuk missing value.

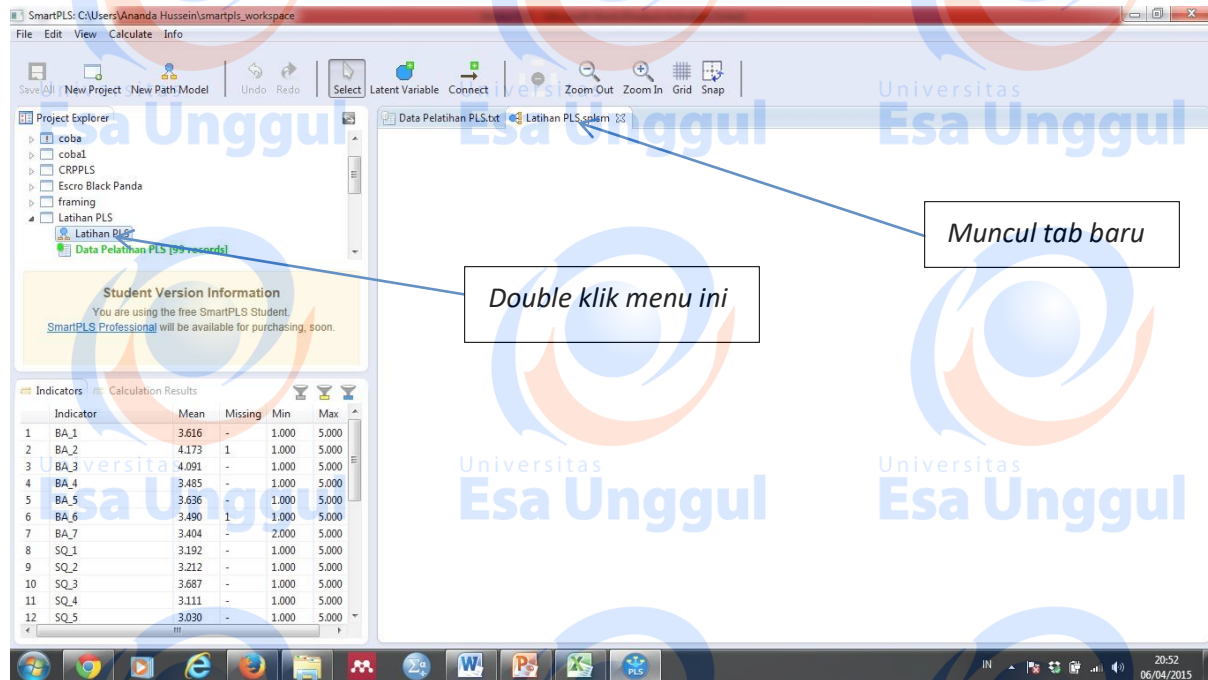




Jika sudah diisikan maka tekan OK. Setelah tekan OK maka dapat dilihat bahwa angka 999 hilang. Di kolom tengah tampak bahwa berapa jumlah data yang hilang.



Setelah data berhasil diinput dengan sempurna maka langkah berikutnya adalah menggambar model struktural. Menggambar model struktural dapat dimulai dari melakukan double klik pada menu “Latihan PLS” yang ada pada layar sebelah kiri. Setelah double klik maka akan muncul tab baru di bagian atas.

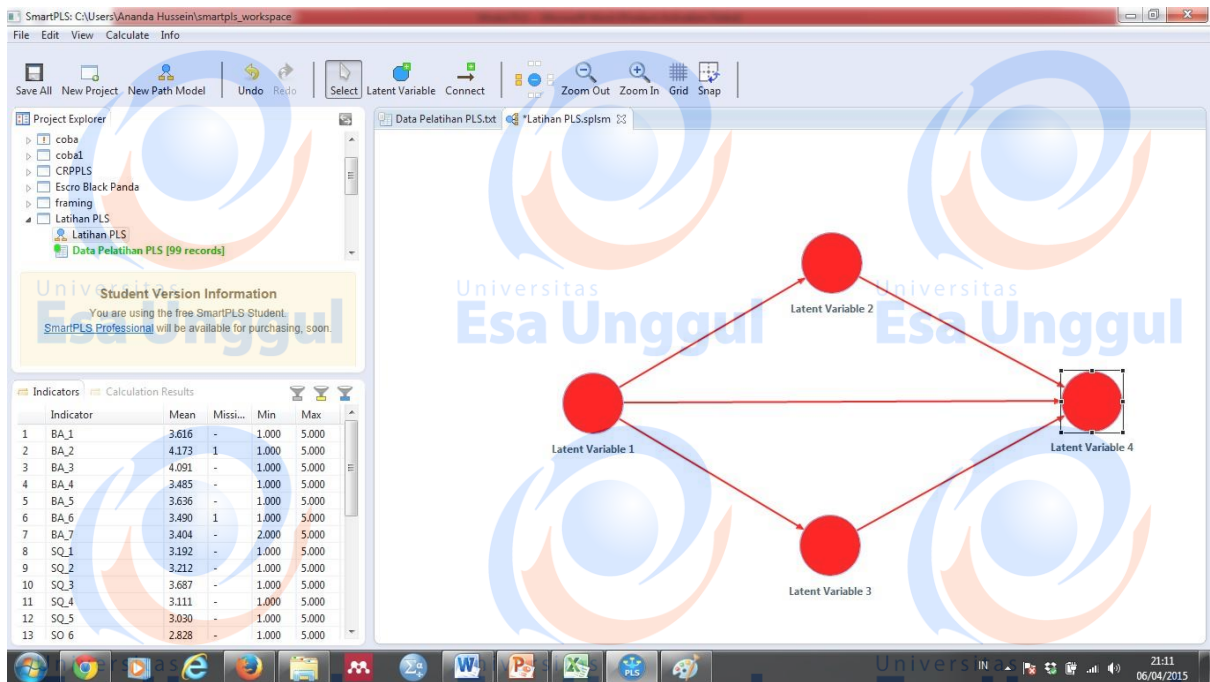


Nah jika tampilan sudah seperti gambar di atas maka model struktural siap untuk digambarkan. Dalam menggambar dua icon memiliki peran yang penting.

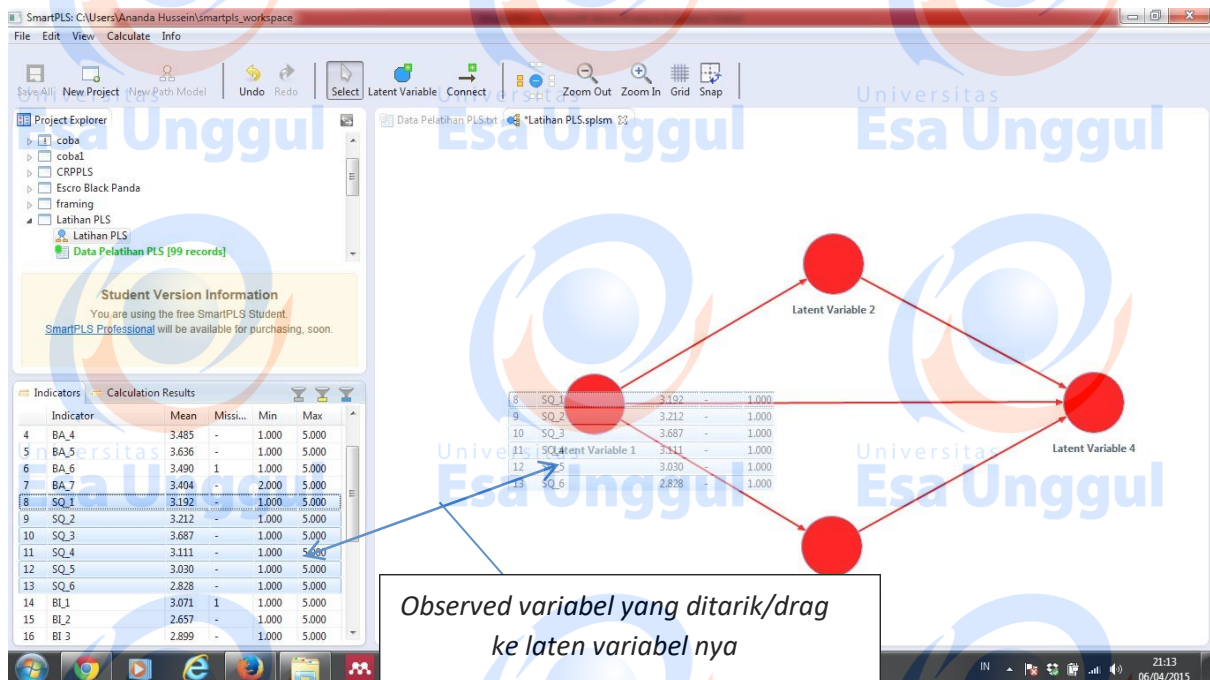
Simbol  merupakan simbol untuk menggambar laten variabel.

Untuk memulai menggambar maka klik icon laten variabel satu kali dan klik sekali lagi dibidang gambar. Jika didalam model yang akan dibuat memiliki empat laten variabel maka empat kali pula harus diulang langkah yang sama. Setelah ke empat laten variabel tergambar di bidang gambar maka langkah berikutnya adalah menggambar hubungan sebab-akibatnya. Hubungan sebab akibat digambar dengan meng klik icon connect dan menghubungkan antara laten variabel satu dengan lainnya.

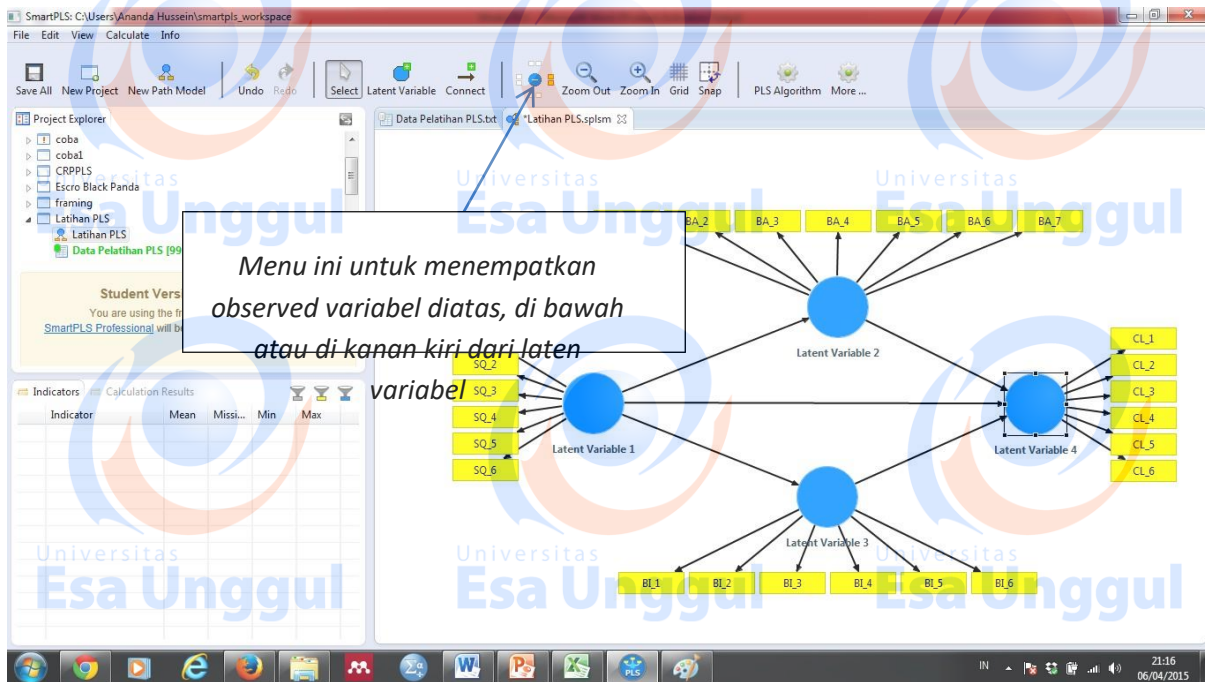




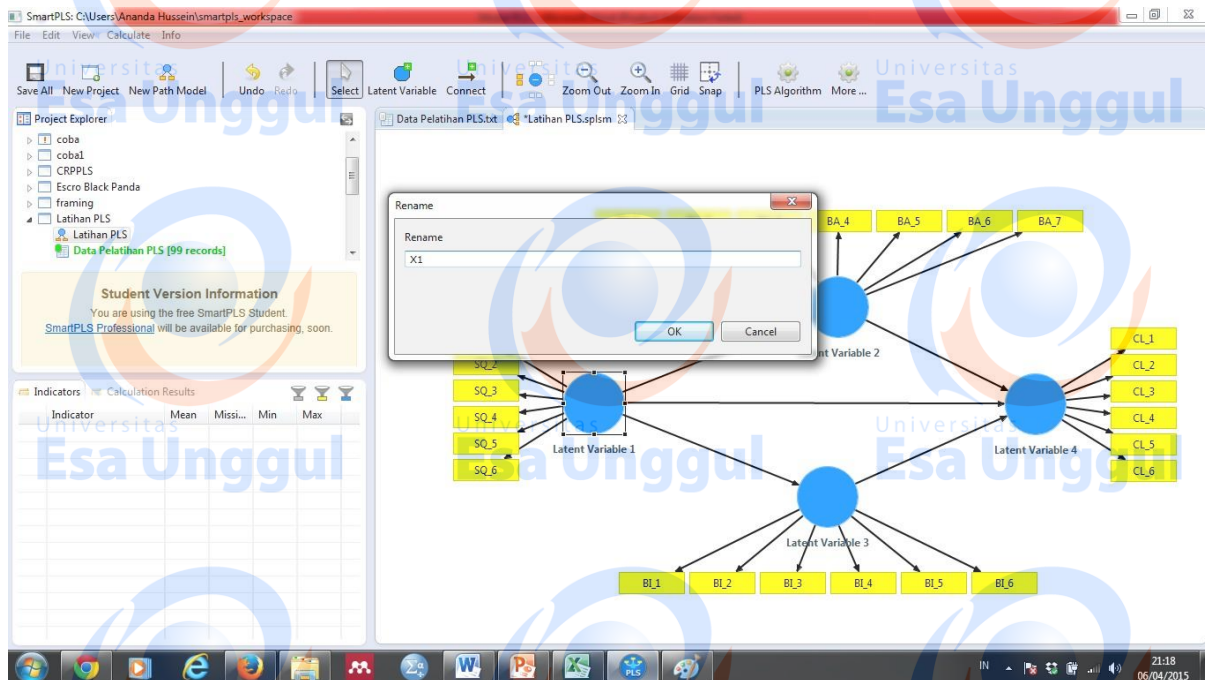
Hasil gambar akan tampak seperti gambar di atas. Gambar tersebut tampak berwarna merah. Hal tersebut dikarenakan masing-masing laten variabel masih belum memiliki variabel manifest/observed variable. Untuk memasukkan observed variabel ke dalam laten variabel maka kita cukup men-drag satu persatu observed variabel yang ada pada layar sebelah kiri.



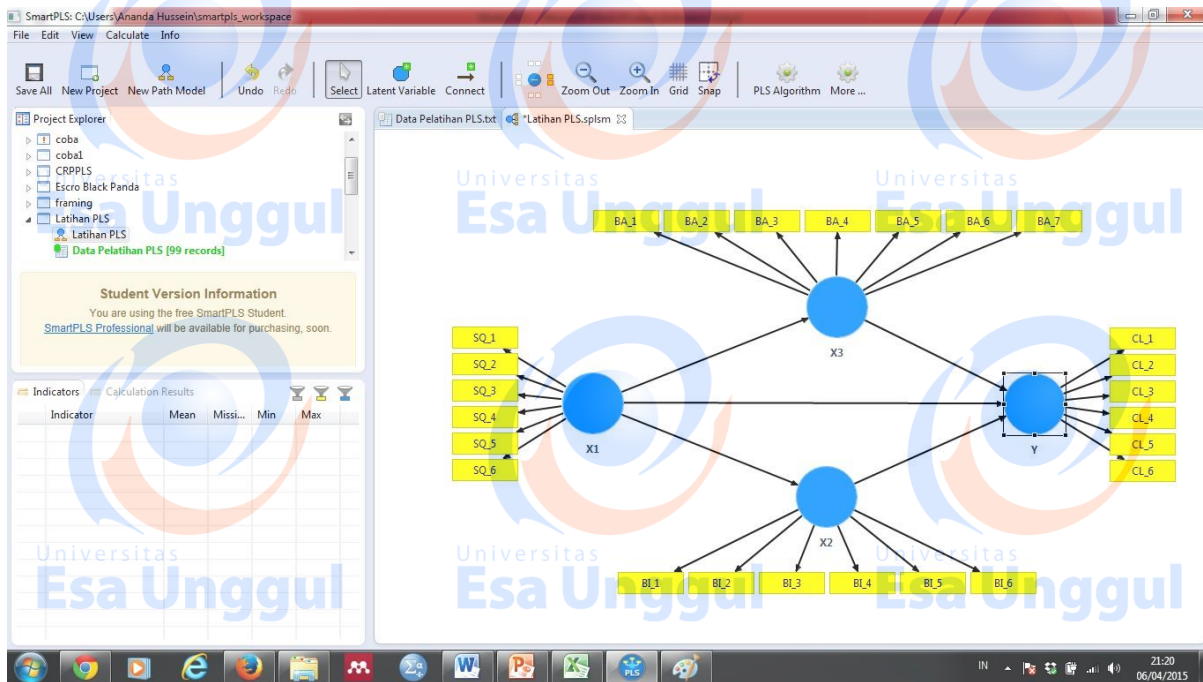
Jika observed variabel sudah dimasukkan semua kedalam laten variabel maka model akan berubah menjadi biru.



Untuk mengganti nama dari masing-masing laten variabel cukup klik kanan laten variabel yang ingin diganti namanya kemudian pada pilihan rename bisa langsung dituliskan nama yang diinginkan.



Setelah nama laten variabel sudah diganti maka hasil akhir dari model yang digambar adalah sebagai berikut



Analisa Model Struktural

Analisa pada PLS dilakukan dengan tiga tahap:

1. *Analisa outer model*
2. *Analisa inner model*
3. *Pengujian Hipotesa.*

Analisa outer model dilakukan untuk memastikan bahwa measurement yang digunakan layak untuk dijadikan pengukuran (valid dan reliabel). Analisa outer model dapat dilihat dari beberapa indikator:

1. *Convergent validity*
2. *Discriminant validity*
3. *Unidimensionality*

Sedangkan analisa inner model/analisa struktural model dilakukan untuk memastikan bahwa model struktural yang dibangun robust dan akurat. Evaluasi inner model dapat dilihat dari beberapa indikator yang meliputi:

1. *Koefisien determinasi (R^2)*
2. *Predictive Relevance ()*
3. *Goodness of Fit Index (GoF)*

Untuk pengujian Hipotesa dilakukan dengan melihat nilai probabilitasnya dan t-statistiknya. Untuk nilai probabilitas, nilai p-value dengan alpha 5% adalah kurang dari 0,05. Nilai t-tabel untuk alpha 5% adalah 1,96. Sehingga kriteria penerimaan Hipotesa adalah ketika t-statistik > t-tabel.

Analisa Outer Model

Analisa Outer Model ini menspesifikasi hubungan antar variabel laten dengan indikator-indikatornya. atau dapat dikatakan bahwa outer model mendefinisikan bagaimana setiap indikator berhubungan dengan variabel latennya. Uji yang dilakukan pada outer model :

- *Convergent Validity*. Nilai *convergen validity* adalah nilai *loading faktor* pada variabel laten dengan indikator-indikatornya. Nilai yang diharapkan >0.7.
- *Discriminant Validity*. Nilai ini merupakan nilai *cross loading faktor* yang berguna untuk mengetahui apakah konstruk memiliki diskriminan yang memadai yaitu dengan cara membandingkan nilai *loading* pada konstruk yang dituju harus lebih besar dibandingkan dengan nilai *loading* dengan konstruk yang lain.
- *Composite Reliability*. Data yang memiliki *composite reliability* >0.7 mempunyai reliabilitas yang tinggi.
- *Average Variance Extracted (AVE)*. Nilai AVE yang diharapkan >0.5.
- *Cronbach Alpha*. Uji reliabilitas diperkuat dengan *Cronbach Alpha*. Nilai diharapkan >0.6 untuk semua konstruk.

Uji yang dilakukan diatas merupakan uji pada outer model untuk indikator reflektif. Untuk indikator formatif dilakukan pengujian yang berbeda. Uji untuk indikator formatif yaitu :

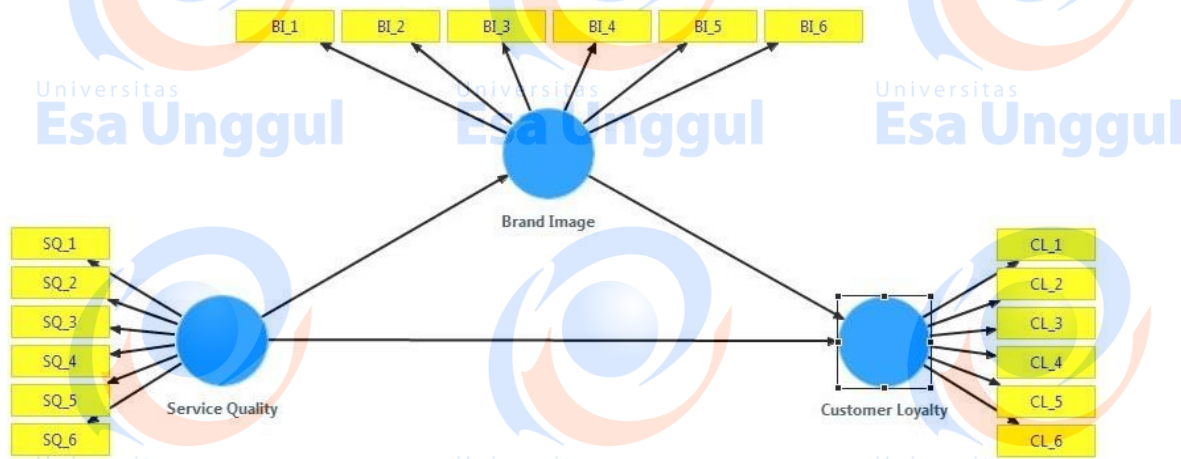
- *Significance of weights*. Nilai *weight indikator formatif* dengan konstruksya harus signifikan.
- *Multicollinearity*. Uji *multicollinearity* dilakukan untuk mengetahui hubungan antar indikator. Untuk mengetahui apakah indikator formatif mengalami *multicollinearity* dengan mengetahui nilai *VIF*. Nilai *VIF* antara 5- 10 dapat dikatakan bahwa indikator tersebut terjadi *multicollinearity*.

Pengujian Hipotesa

Secara umum metode *explanatory research* adalah pendekatan metode yang menggunakan PLS. Hal ini disebabkan pada metode ini terdapat pengujian Hipotesa. Menguji hipotesis dapat dilihat dari nilai t-statistik dan nilai probabilitas. Untuk pengujian hipotesis menggunakan nilai statistik maka untuk alpha 5% nilai t-statistik yang digunakan adalah 1,96. Sehingga kriteria penerimaan/penolakan Hipotesa adalah H_a diterima dan H_0 di tolak ketika t-statistik > 1,96. Untuk menolak/menerima Hipotesis menggunakan probabilitas maka H_a di terima jika nilai $p < 0,05$.

Menggunakan PLS-SEM Pada Contoh Penelitian

Sebuah penelitian akan menguji sebuah model struktural sebagai berikut

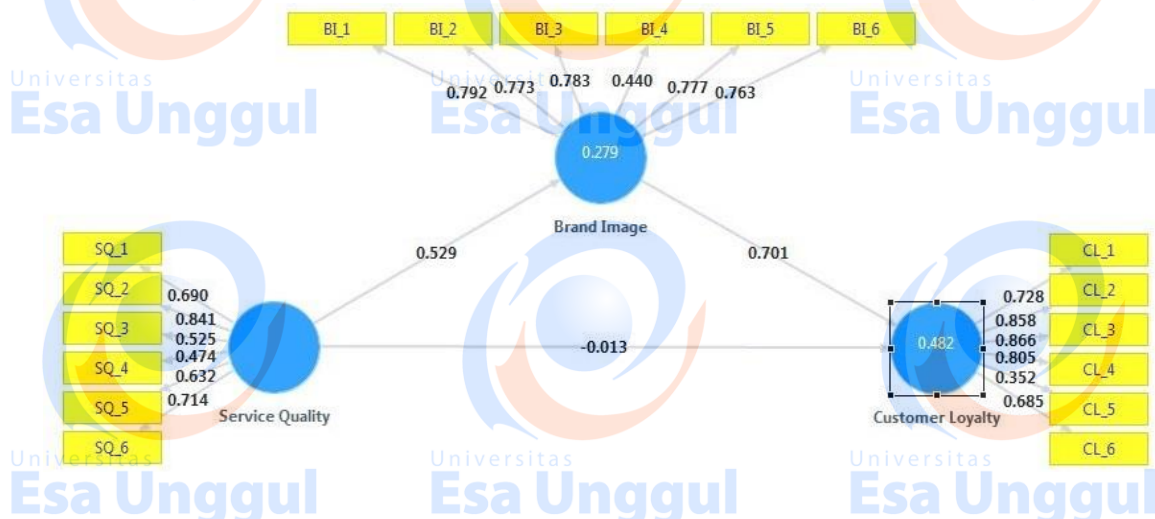


Dari model tersebut dirumuskan permasalahan sebagai berikut:

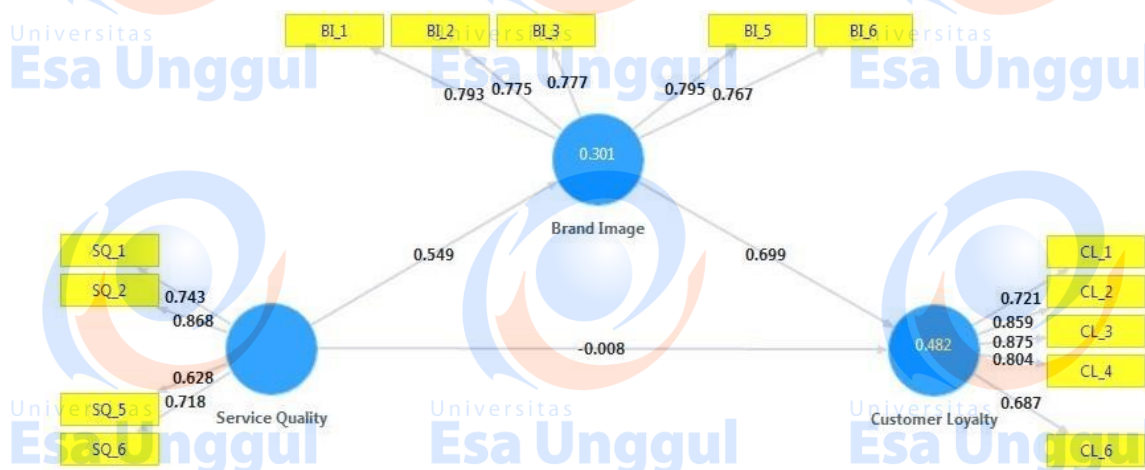
1. Apakah Service quality memiliki pengaruh yang signifikan terhadap customer loyalty
2. Apakah brand image memiliki pengaruh yang signifikan terhadap loyalty
3. Apakah service quality memiliki pengaruh yang signifikan terhadap brand image
4. Apakah brand image memediasi pengaruh service quality terhadap customer loyalty

Untuk menjawab permasalahan –permasalahan tersebut yang pertama kali dilakukan adalah melakukan pengukuran Outer dan Inner model.

Pengujian Outer Model



Dari gambar tersebut terlihat bahwa item BI_4, CL_5, SQ_3 dan SQ_4 memiliki nilai faktor loading dibawah 0,6. Oleh karenanya ketiga item tersebut harus dihapus dari model.



Selain dilihat dari nilai faktor loading, convergent validity juga dapat dilihat dari nilai Average Variance Extracted (AVE). Pada penelitian ini nilai AVE masing-masing konstruk berada di atas 0,5. Oleh karenanya tidak ada permasalahan konvergen validity pada model yang diuji.

Konstruk	AVE
Brand Image	0,611
Customer Loyalty	0,629
Service Quality	0,554

Dikarenakan tidak ada nya permasalahan convergen validity maka berikutnya yang diuji adalah permasalahan yang terkait dengan discriminant validity. Discriminant validity dapat di uji dengan membandingkan nilai cr akar kuadrat AVE dengan nilai korelasi antar konstruk.

	Brand Image	Customer Loyalty	Service Quality
Brand Image	0,781		
Customer Loyalty	0,694	0,793	
Service Quality	0,549	0,375	0,744

Akar kuadrat AVE

Dari tabel di atas terlihat bahwa nilai akar kuadrat dari AVE (0,781; 0,793 dan 0,744) lebih besar dari korelasi masing-masing konstruk.

Metode lain yang dapat digunakan untuk menguji discriminant validity adalah dengan melihat dari tabel cross loading.

	Brand Image	Customer Loyalty	Service Quality
BI_1	0.793	0.583	0.447
BI_2	0.775	0.443	0.436
BI_3	0.777	0.488	0.384
BI_5	0.795	0.639	0.504
BI_6	0.767	0.525	0.352
CL_1	0.468	0.721	0.226
CL_2	0.617	0.859	0.229
CL_3	0.601	0.875	0.393
CL_4	0.547	0.804	0.327
CL_6	0.501	0.687	0.312
SQ_1	0.370	0.327	0.743
SQ_2	0.489	0.375	0.868
SQ_5	0.373	0.090	0.628
SQ_6	0.395	0.266	0.718

Tabel di atas menunjukkan bahwa nilai loading dari masing-masing item terhadap konstraknya lebih besar dari pada nilai cross loadingnya. Dari hasil analisa cross loading tampak bahwa tidak terdapat permasalahan discriminant validity.

Untuk memastikan bahwa tidak ada masalah terkait pengukuran maka langkah terakhir dalam evaluasi outer model adalah menguji unidimensionalitas dari model. Uji unidimensionality dilakukan dengan menggunakan indikator Composite reliability dan alpha cronbach. Untuk kedua indikator ini titik cut-off value adalah 0,7.

	Composite Reliability
Brand Image	0.887
Customer Loyalty	0.893
Service Quality	0.831

Tabel di atas menunjukkan bahwa seluruh konstruk memiliki nilai composite reliability di atas 0,7. Oleh karena itu tidak ditemukan permasalahan reliabilitas/unidimensionality pada model yang dibentuk.

Pengujian Inner Model

Evaluasi inner model dapat dilakukan dengan tiga cara. Ketiga cara tersebut adalah dengan melihat dari R², Q² dan GoF.

Konstruk	Nilai R ²
Brand Image	0,301
Customer Loyalty	0,482

Berikut untuk pengujian Inner model dapat dilakukan dengan melihat nilai Q² (predictive relevance). Untuk menghitung Q² dapat digunakan rumus

$$Q^2 = 1 - (1 - R^2) (1 - R^2) \dots (1 - R^2) \dots$$

$$Q^2 = 1 - (1 - 0,301)(1 - 0,482)$$

$$Q^2 = 0,637$$

Yang terakhir adalah dengan mencari nilai Goodness of Fit (GoF). Berbeda dengan CB-SEM, untuk nilai GoF pada PLS-SEM harus dicari secara manual.

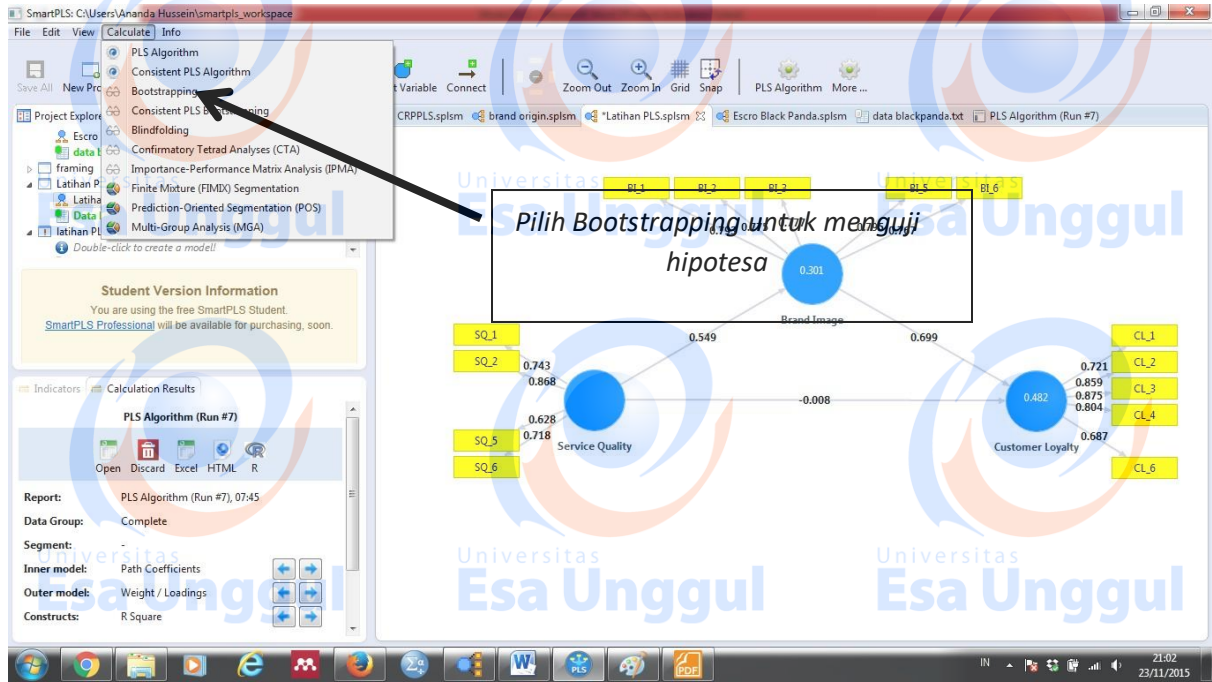
$$\sqrt{\dots} \dots \dots \text{Tenenhaus (2004)}$$

$$\text{GoF} = 0,483$$

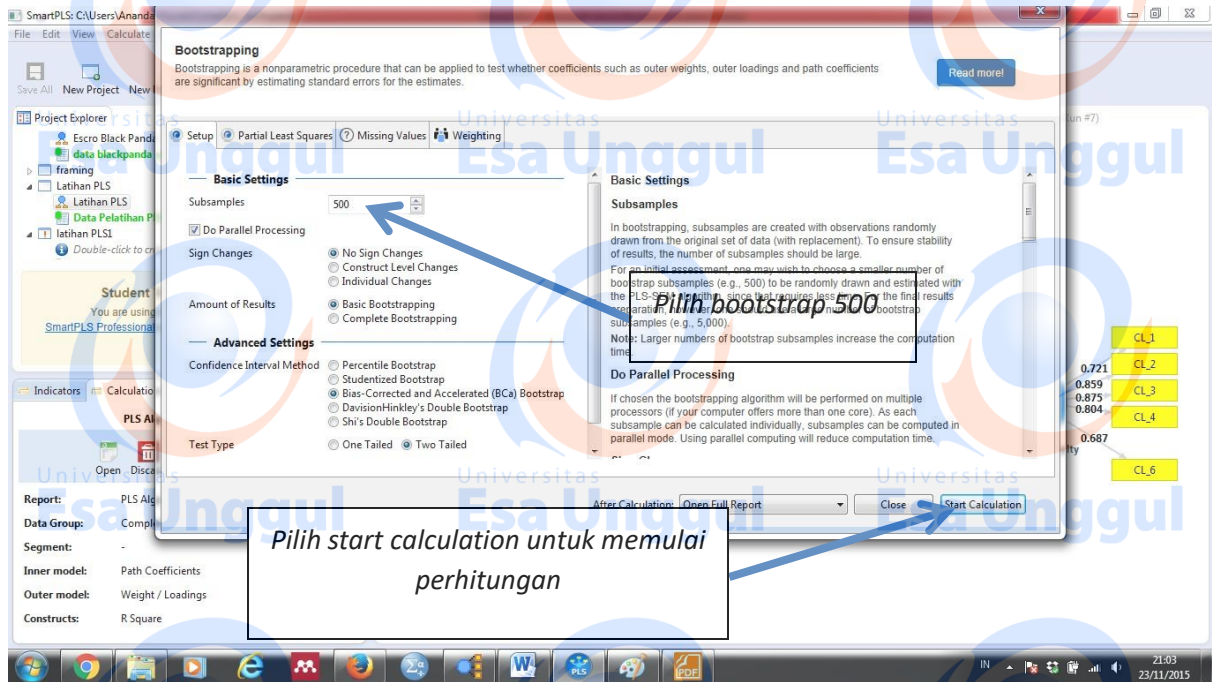
Menurut Tenenhaus (2004), nilai GoF small = 0,1, GoF medium = 0,25 dan GoF besar = 0,38.

Dari pengujian R², Q² dan GoF terlihat bahwa model yang dibentuk adalah robust. Sehingga pengujian hipotesa dapat dilakukan.

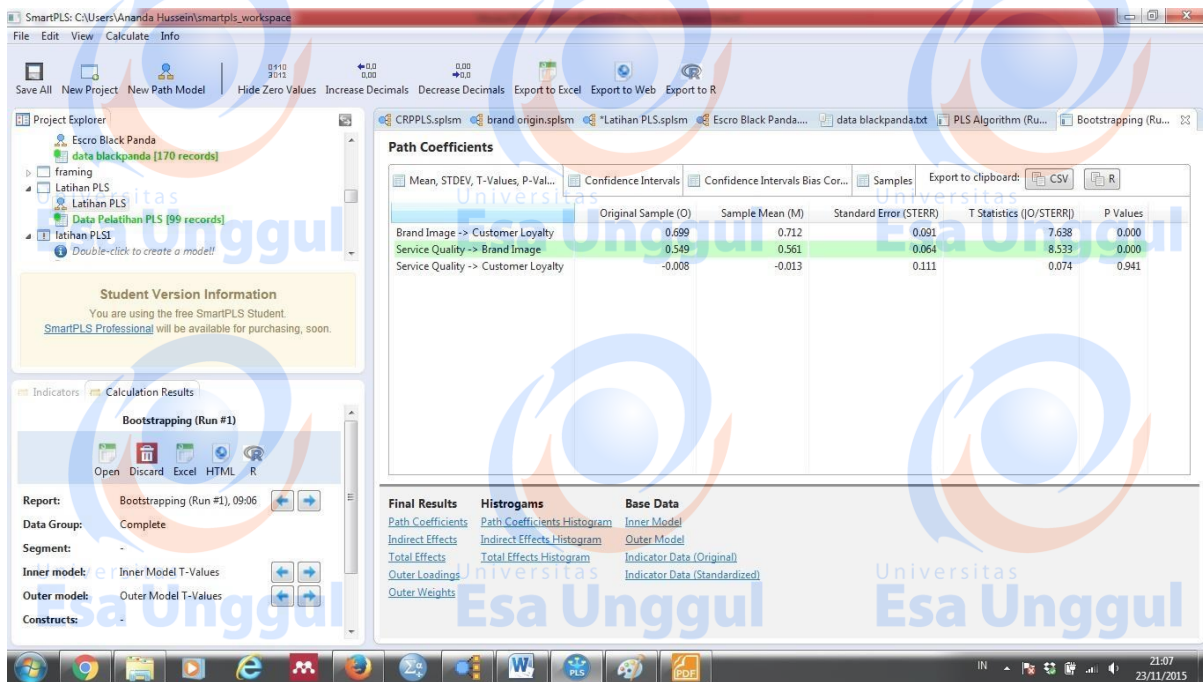
Pengujian Hipotesa



Kemudian



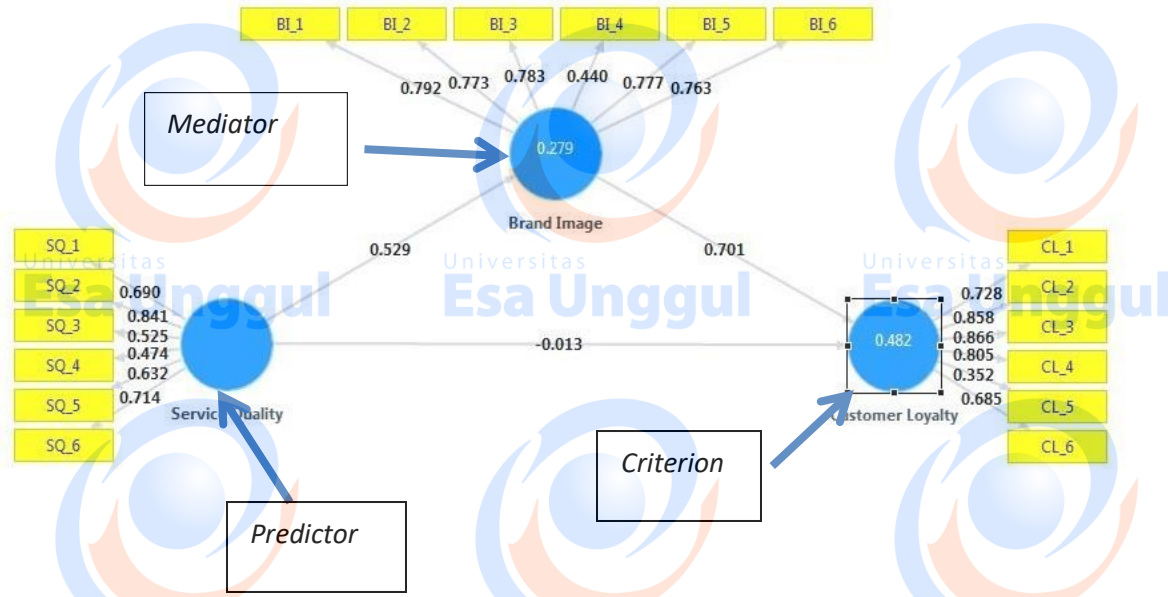
Hasil analisa akan muncul demikian



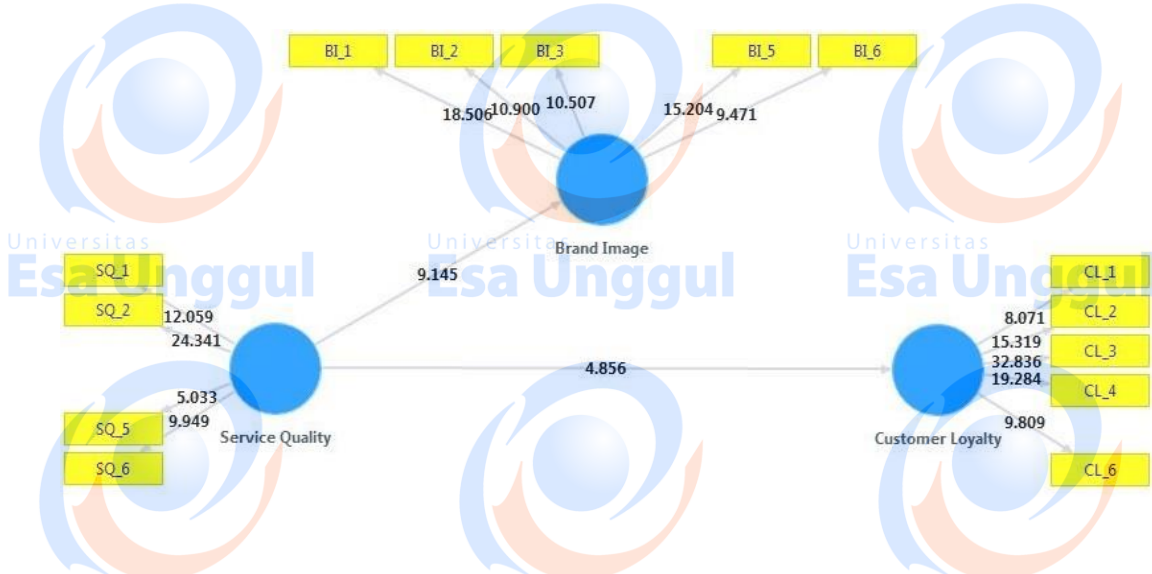
Dari hasil estimation dapat dilihat bahwa pengaruh Brand Image terhadap Customer loyalty memiliki koefisien jalur sebesar 0,712. Dimana pengaruh nya adalah signifikan ($t= 7,638$; $p < 0,005$).

Dari hasil pengujian hipotesa ini dapat dijawab rumusan masalah 1,2 dan 3. Untuk menguji pengaruh tidak langsung seperti halnya ditanyakan pada rumusan masalah keempat maka perlu dilakukan pengujian lebih lanjut. Untuk menguji pengaruh tidak langsung dilakukan dengan cara menggunakan teori yang dirumuskan oleh Kenny dan Baron (1986).

Menurut Kenny dan Baron, dalam menguji pengaruh tidak langsung dikenal tiga macam variabel. Ketiga variabel tersebut adalah predictor, criterion dan mediator.

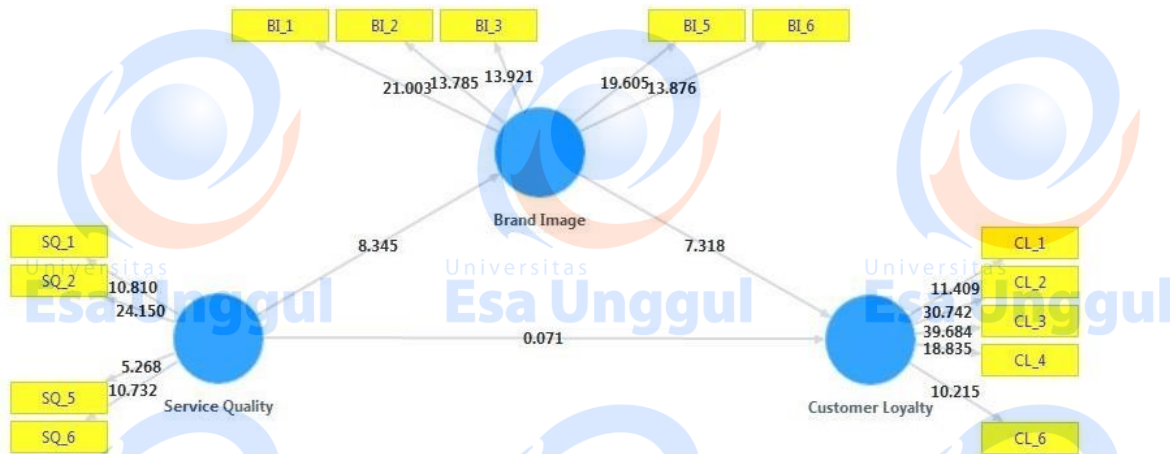


Untuk menguji pengaruh tidak langsung dapat dilakukan melalui empat tahap. Tahap pertama menguji pengaruh langsung dari predictor kepada criterion



Pada langkah pertama ini terlihat bahwa Service Quality memiliki pengaruh secara langsung terhadap customer loyalty ($t = 4,856$; $p < 0,05$).

Kemudian dilanjutkan dengan langkah kedua dan ketiga yaitu melihat apakah predictor memiliki pengaruh terhadap mediator dan mediator memiliki pengaruh terhadap criterion.



Pada langkah kedua terlihat bahwa predictor (Service Quality) memiliki pengaruh terhadap mediator (Brand Image) dan pada langkah ketiga terlihat bahwa mediator memiliki pengaruh terhadap criterion (customer loyalty). Dengan mengikuti langkah-langkah pada Kenny dan Baron (1986) terlihat bahwa Brand image memediasi pengaruh service quality terhadap Customer loyalty.

Untuk melihat apakah brand image memberikan pengaruh mediasi sempurna (full mediation) atau mediasi sebagian (partial mediation), maka dapat dilanjutkan pada langkah keempat. Langkah keempat adalah melihat pengaruh predictor (service quality) terhadap criterion (Customer loyalty) dengan tetap memasukkan pengaruh mediator (Brand Image). Dari pengujian tampak bahwa ternyata ketika pengaruh dari mediator dimasukkan pada model terlihat bahwa service quality memiliki pengaruh yang signifikan terhadap customer loyalty. Hal ini berbeda dengan langkah pertama ketika pengaruh mediator tidak dimasukkan kedalam model. Dengan hilangnya pengaruh langsung dari predictor (service quality) terhadap criterion (customer loyalty) dapat dikatakan bahwa Brand Image memiliki pengaruh mediasi sempurna (full mediaton).



DAFTAR PUSTAKA

1. Ammenwerth, E., Iller, C., Mahler, C., 2004. Visions and strategies to improve evaluation of health information systems Reflections and lessons based on the HIS-EVAL workshop in Innsbruck. *International Journal of Medical Informatics*. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/15171977> [Accessed January 18, 2017].
2. Berg, 2004. *Health Information Management: Integrating Information Technology In Health Care Work*, London: Routledge.
3. Burke, Wang & Wan, 2002. Exploring Hospitals Adoption of Information Technology. *Journal of Medical System*, 26, pp.349–355 [Accessed January 10, 2017].
4. Davis, G.B., 1996. *Human Resources and personnel Management*.
5. Davis, F.D., 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of. *Information Technology MIS Quarterly*, 13(3), p.319–340
6. Furneaux, B., 2006a. Theories Used in IS Research: Task Technology Fit.
7. Furneaux, B., 2006b. Theories Used in IS Research: Technology Acceptance Model.
8. Husein, A.S., 2015. *Penelitian Bisnis dan Manajemen Menggunakan Partial Least Squares (PLS) dengan smartPLS 3.0*. Universitas Brawijaya, Malang.
9. Jogiyanto dan Willy Abdillah, 2011. *Sistem Tatakelola Teknologi Informasi Edisi 1.*, Yogyakarta: CV. Andi Offset
10. Koswara, E., 1998. *Dinamika Informasi dalam Era Global*, Bandung: Pengurus Daerah Ikatan Pustakawan Indonesia Jawa Barat bekerja sama dengan PT. Remaja Rosdakarya.
11. Nugroho, E., 2008. *Sistem Informasi Manajemen: Konsep, Aplikasi, dan Pengembangannya*, Yogyakarta: Andi
12. Venkatesh, Viswanath & Bala, H., 2008. Technology Acceptance Model 3 and a research agenda on interventions. *Decision Science*, 39(2), pp.273–315.
13. Venkatesh, V. & Davis, F.D., 2000. A Theoretical Extension of the Technology Acceptance Model : Four Longitudinal Field Studies. *Management Science*, 46(2), pp.186–204
14. Yusof, M.M., Kuljis, J., Papazafeiropoulou, A., Stergioulas, L.K., 2008. An evaluation framework for Health Information Systems: human, organization and technology-fit factors (HOT-fit). *International Journal of Medical Informatics*, 77(6), pp.386–398 [Accessed December 8, 2016]