

Mata Kuliah

ANALISIS & DESAIN SISTEM (ADS)

System Analysis & Design

Dosen:

Budi Nugroho, S.Kom, M.Kom

Materi 7

UML: Interaction Diagram



Program Studi S1 Informatika

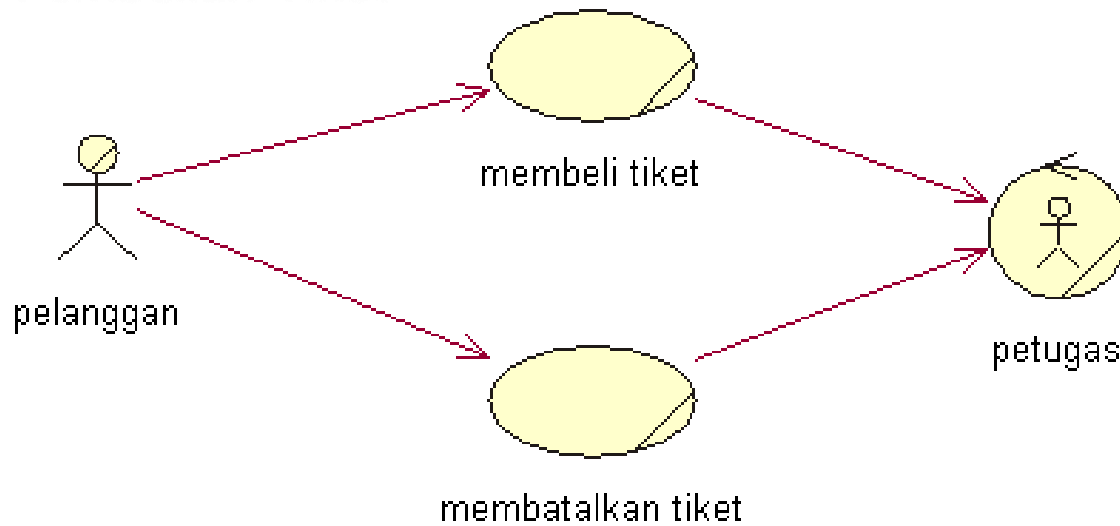
Fakultas Ilmu Komputer UPN "Veteran" Jawa Timur

TA 2019/2020 Semester Genap

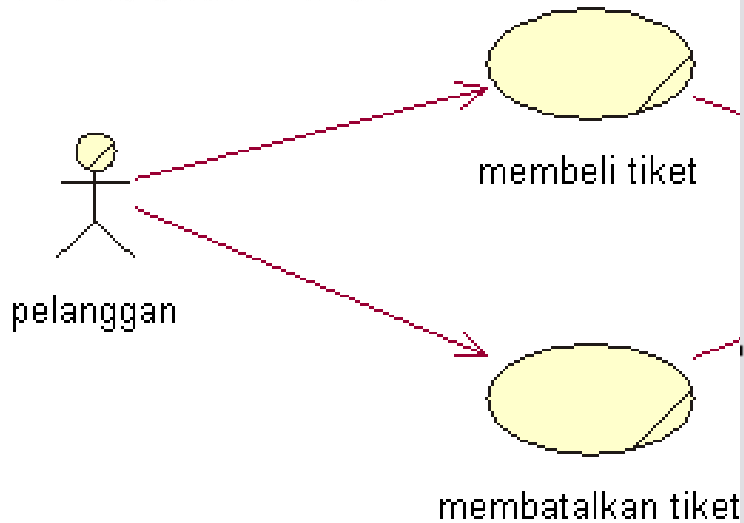
UML: Review Materi

Business Process View	Business Functionality	Business Use Case Diagram
	Business Workflow	Business Activity Diagram
Functional System View	System Functionality	Use Case Diagram
	System Workflow	Activity Diagram

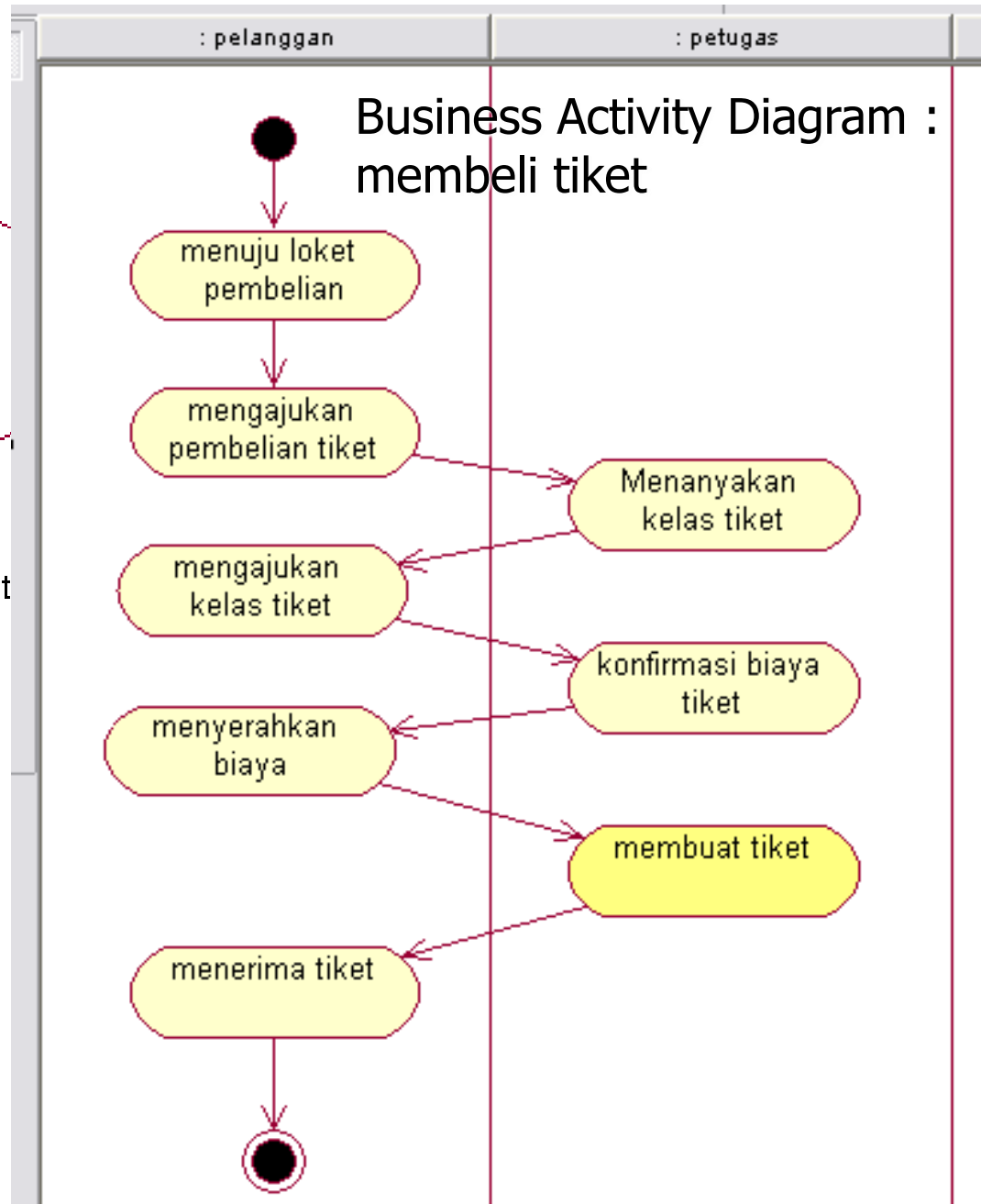
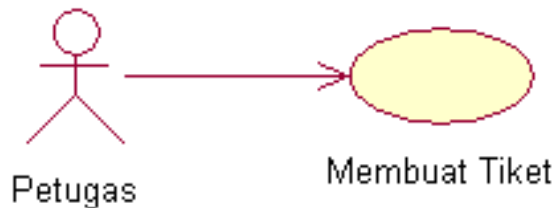
Business Use Case Diagram : Pembelian Tiket



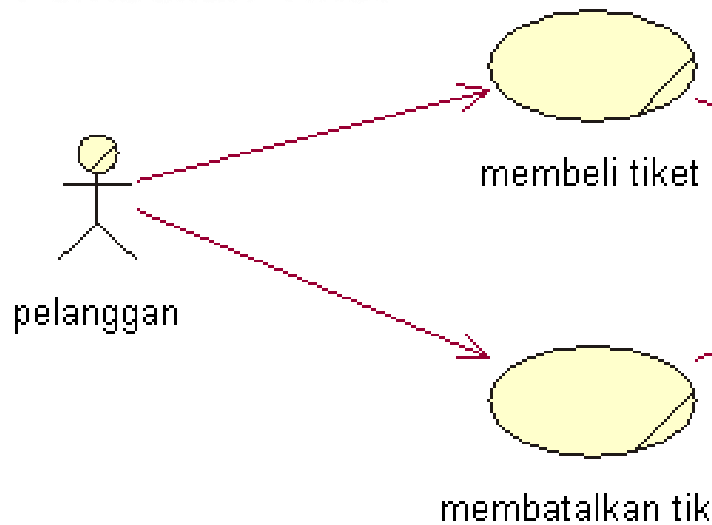
Business Use Case Diagram : Pembelian Tiket



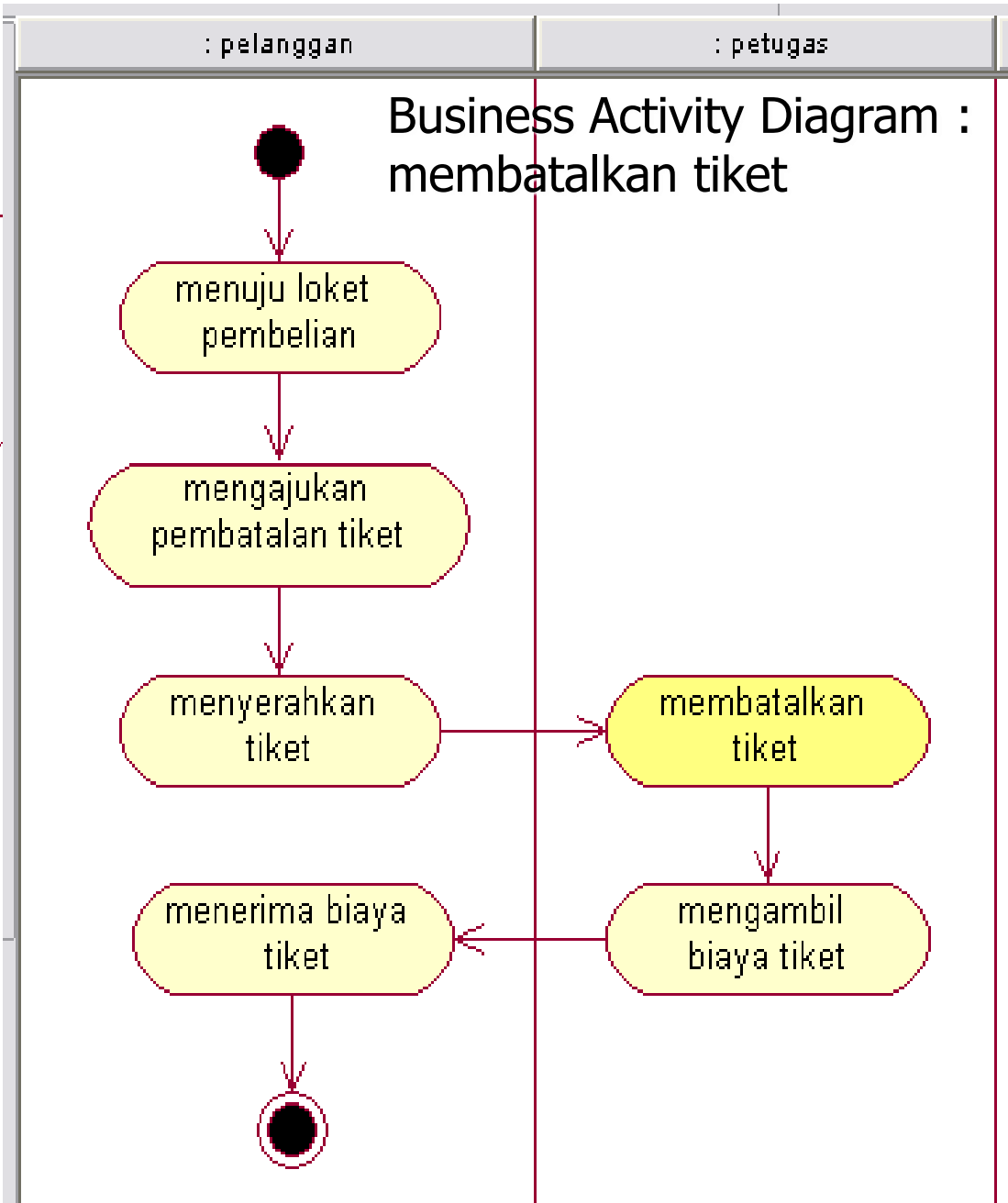
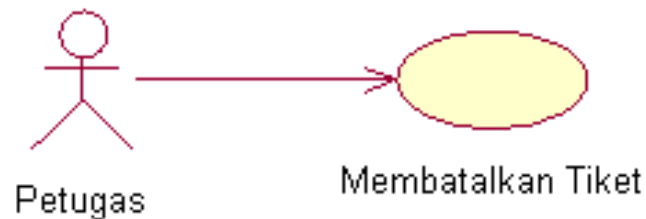
Use Case Diagram : membeli tiket



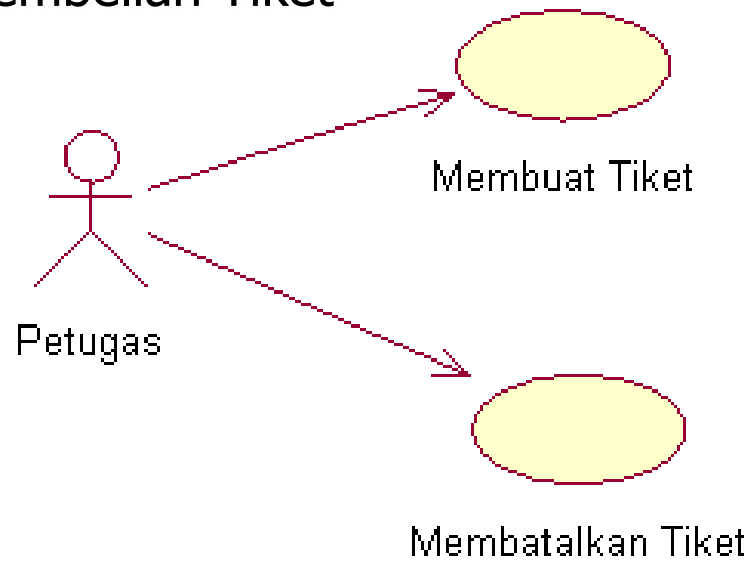
Business Use Case Diagram : Pembelian Tiket



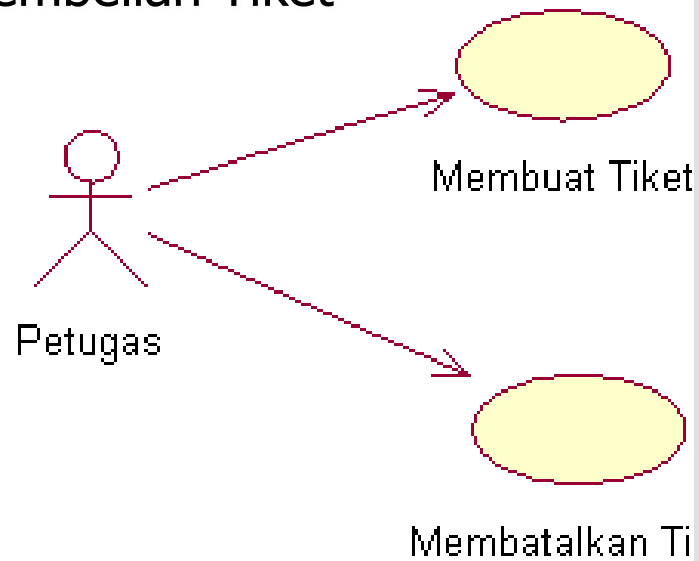
Use Case Diagram : membatalkan tiket



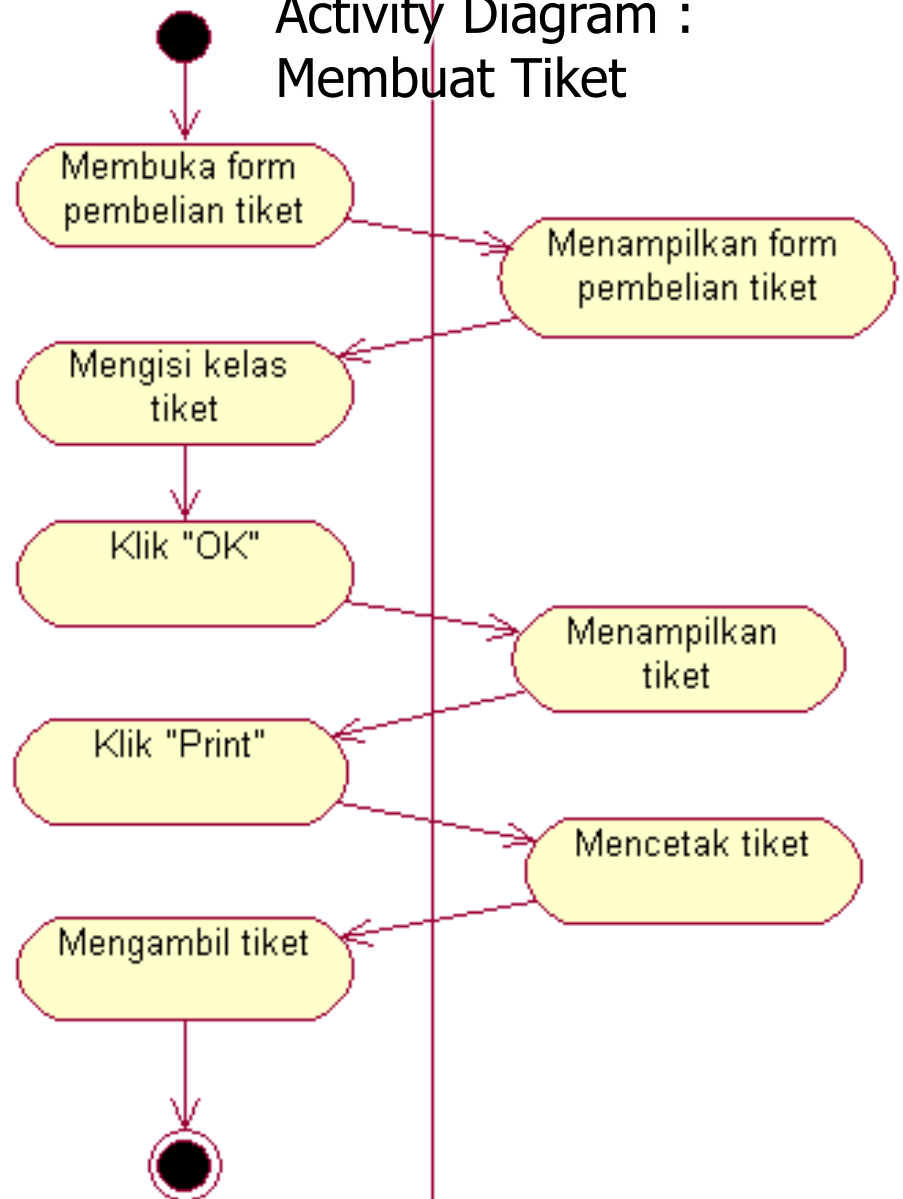
Use Case Diagram : Pembelian Tiket



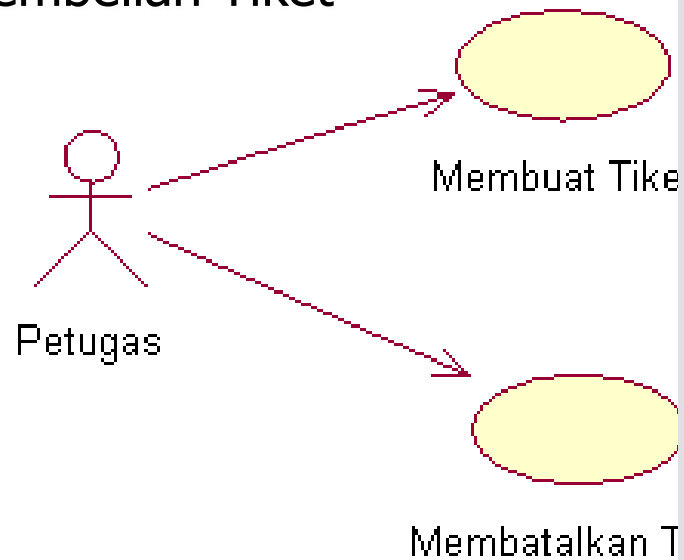
Use Case Diagram : Pembelian Tiket



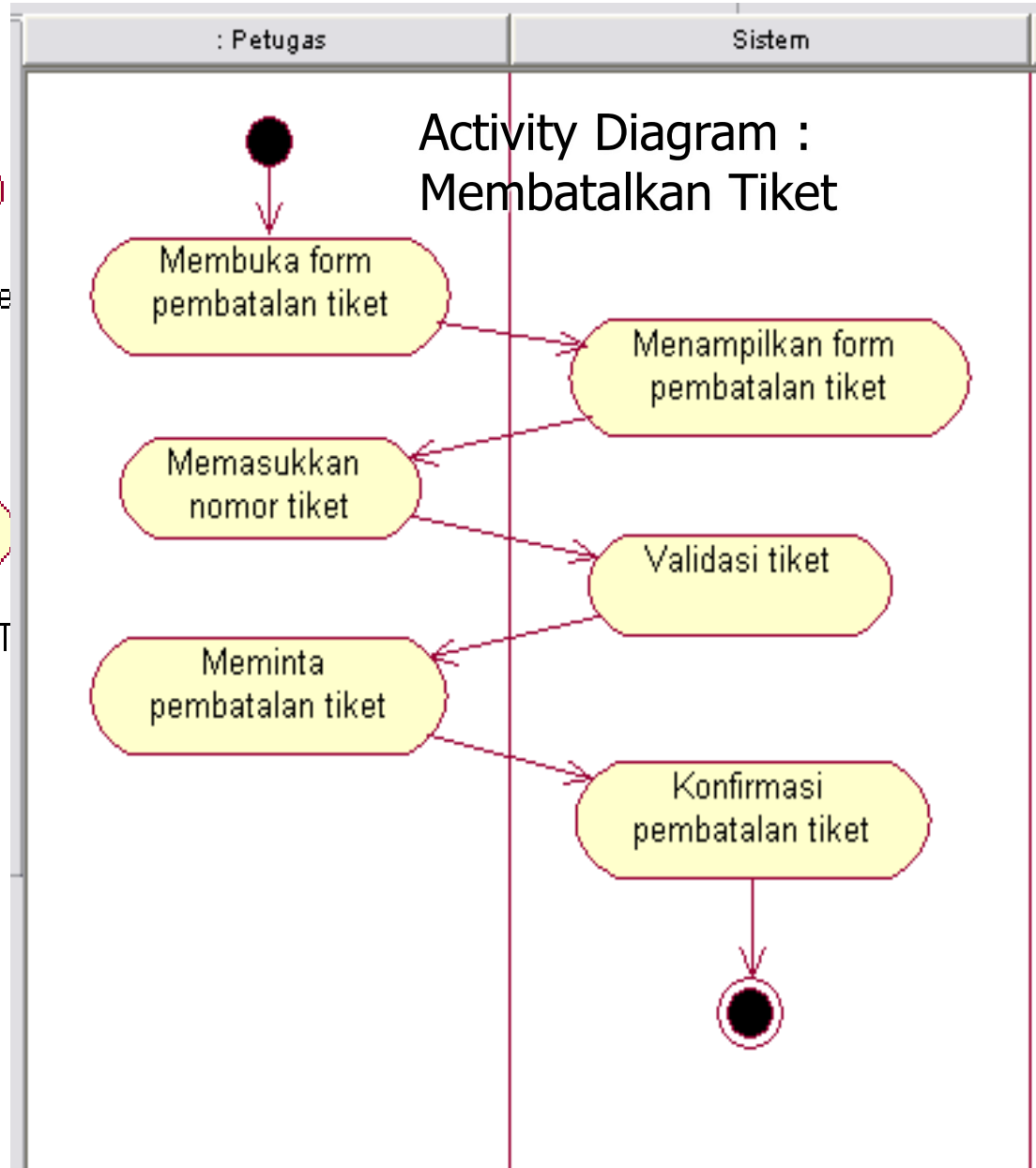
Activity Diagram : Membuat Tiket



Use Case Diagram : Pembelian Tiket



Activity Diagram : Membatalkan Tiket



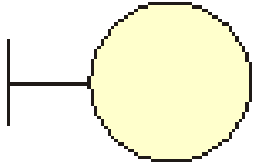
UML

Business Process View	Business Functionality	Business Use Case Diagram
	Business Workflow	Business Activity Diagram
Functional System View	System Functionality	Use Case Diagram
	System Workflow	Activity Diagram
Organizational System View	Dinamic View	Interaction Diagram <ul style="list-style-type: none">• Sequence Diagram• Collaboration Diagram
		Statechart Diagram
	Logical View	Class Diagram
	Architectural View	Component Diagram
		Deployment Diagram

Interaction Diagram

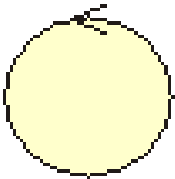
- Memodelkan interaksi antara object-object dalam sistem.
 - ✎ Menunjukkan bagaimana object-object saling berkomunikasi dalam setiap aliran kejadian sistem (system workflow / flow of events).
- Interaksi dilakukan dengan menggunakan link dan message.
Link : arah komunikasi (dari suatu object ke object lain).
Message : pesan kepada object lain untuk melakukan sesuatu.
- Setiap message pada Interaction Diagram nantinya akan dipetakan ke setiap operasi pada Class dari object tujuan.
- Ada 2 Bentuk :
 - ✓ Interaksi antar object berdasarkan urutan waktu → fokus pada aliran message.
 - ✎ **Sequence Diagram.**
 - ✓ Interaksi antar object tidak berdasarkan urutan waktu → fokus pada pola hubungan antar object.
 - ✎ **Collaboration Diagram.**

Klasifikasi Class



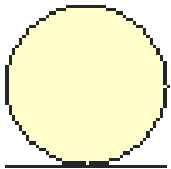
Boundary Class

- Menangani informasi pada User Interface (input / output sistem dari/ke Pengguna).



Control Class

- Menangani informasi yang membutuhkan pengolahan lebih lanjut (dari/ke User Interface / Basisdata).
- Mengendalikan aliran informasi dari/ke Boundary Class & dari/ke Entity Class.



Entity Class

- Menangani informasi yang berkaitan pengolahan Basisdata.
- Sebagian diantaranya atau semua mungkin akan dipetakan menjadi tabel pada tahap desain Basisdata.

Sequence Diagram : Interaksi antar Object

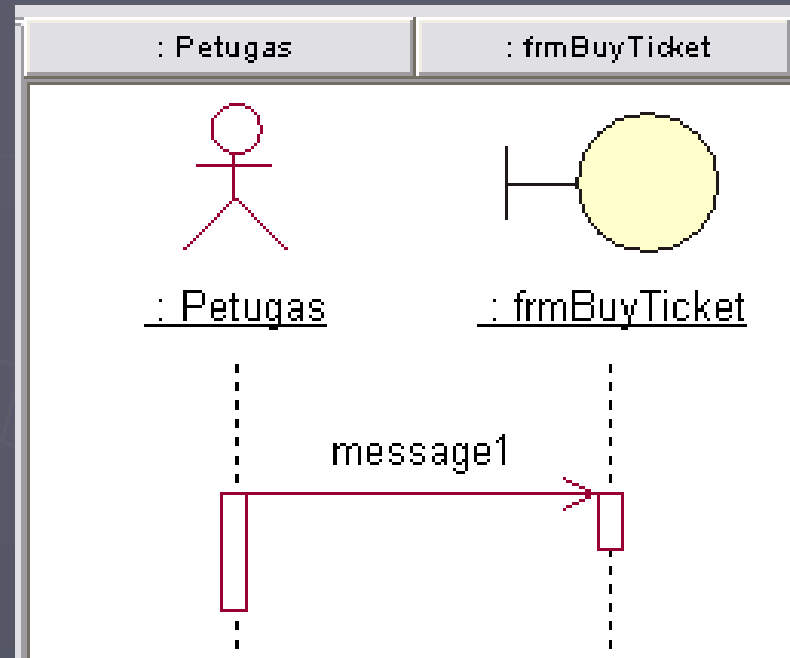
- Interaksi antara Pengguna dengan Sistem.
→ antara Actor dengan Object dari Class Boundary.
- Interaksi antar bagian dalam Sistem.
→ antar Object.
- Interaksi antar bagian dalam Sistem.
→ antara operasi dalam Object.

Sequence Diagram : Interaksi antar Object

- Interaksi antara Pengguna dengan Sistem.
→ antara Actor dengan Object dari Class Boundary.

- Interaksi antar bagian dalam Sistem.
→ antar Object.

- Interaksi antar bagian dalam Sistem.
→ antara operasi dalam Object.



➡ Actor memicu terjadinya aktivitas pada Sistem.

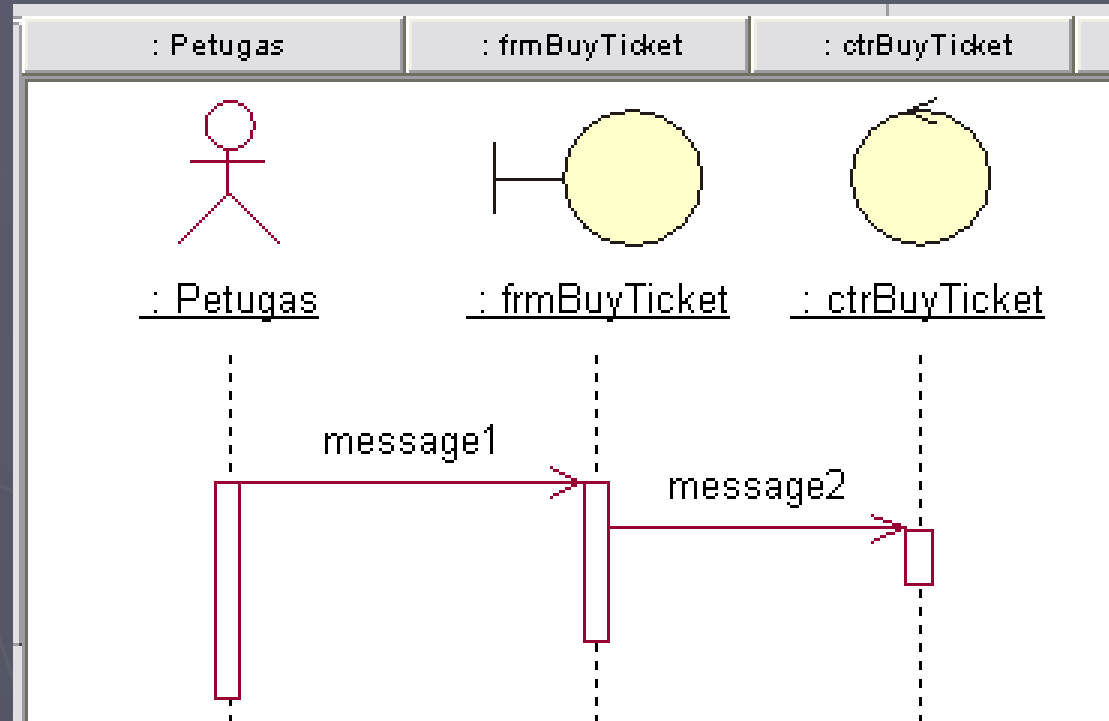
➡ Actor memicu Object (dari Class 'frmBuyTicket') untuk melakukan operasi 'message1'.

Sequence Diagram : Interaksi antar Object

- Interaksi antara Pengguna dengan Sistem.
→ antara Actor dengan Object dari Class Boundary.

- Interaksi antar bagian dalam Sistem.
→ antar Object.

- Interaksi antar bagian dalam Sistem.
→ antara operasi dalam Object.

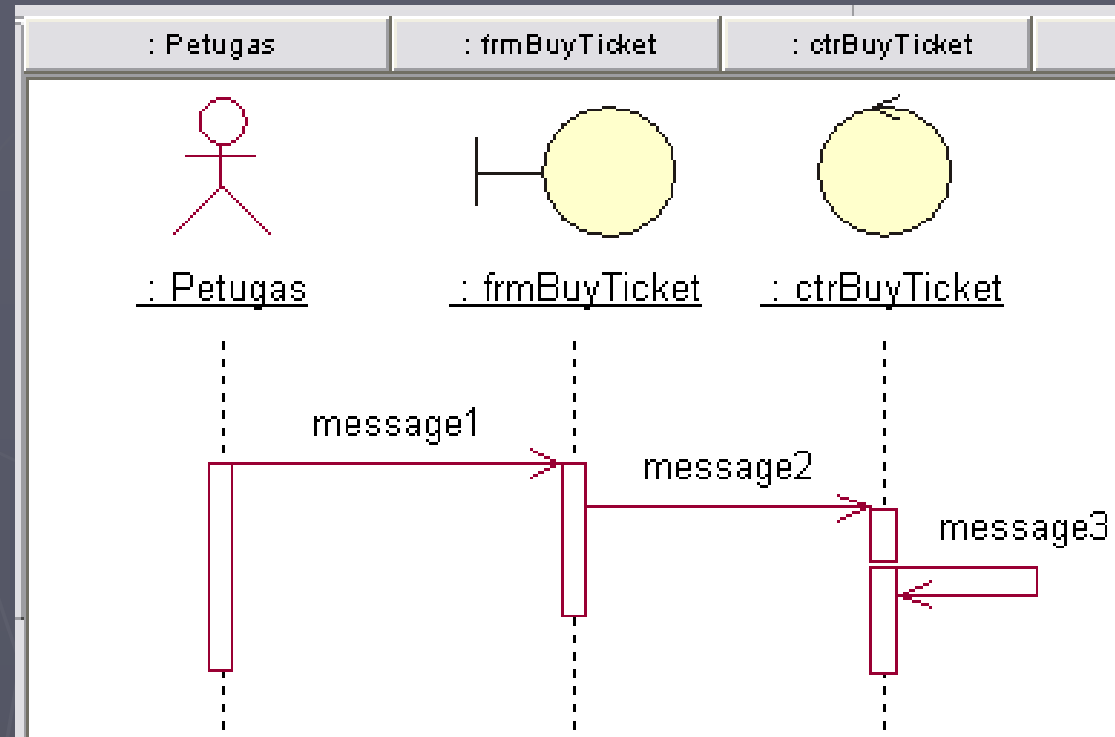


☞ Suatu Object memicu terjadinya aktivitas pada Object lain.

☞ Object dari Class 'frmBuyTicket' (melalui 'message1') memicu Object dari Class 'ctrBuyTicket' untuk melakukan operasi 'message2'.

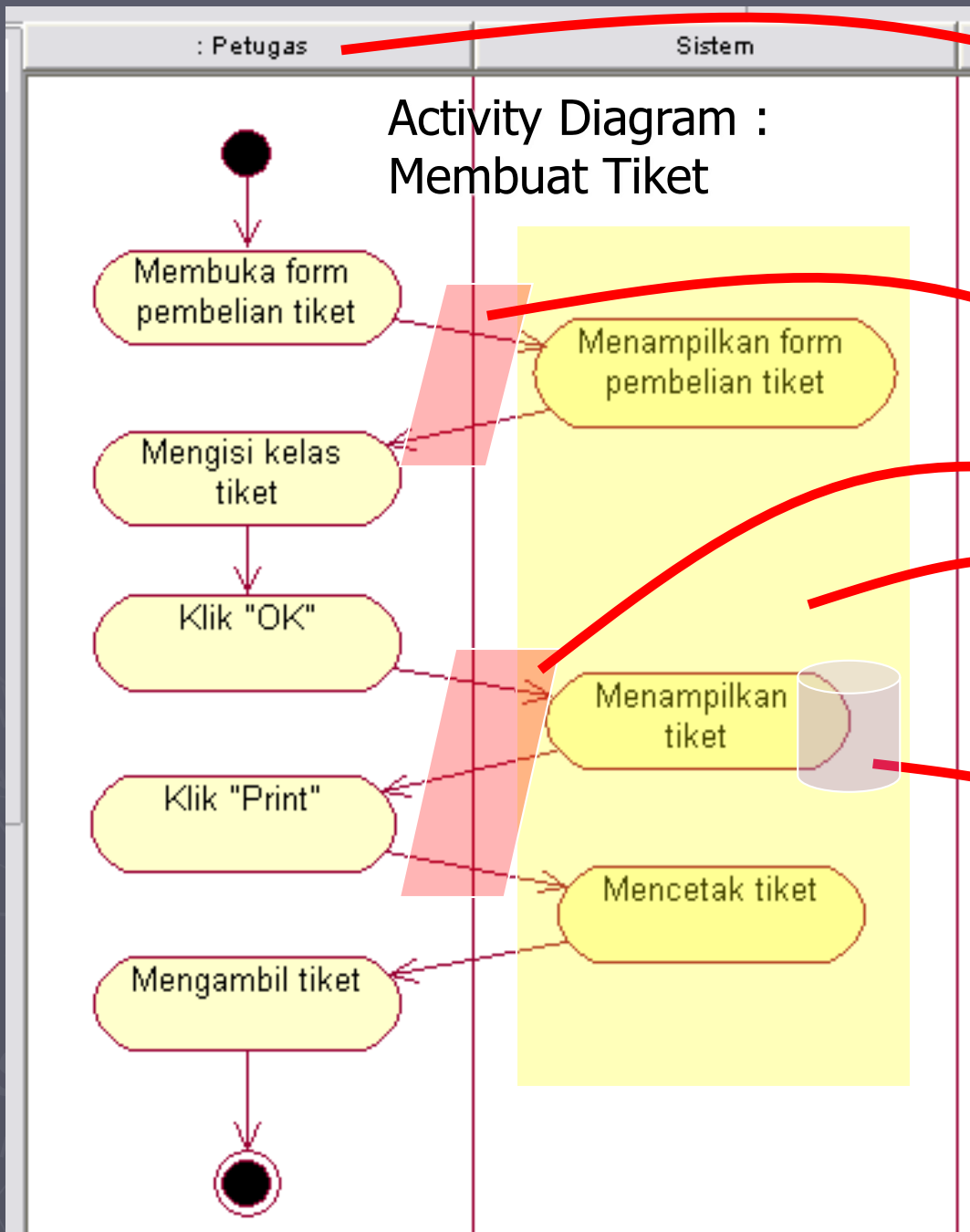
Sequence Diagram : Interaksi antar Object

- Interaksi antara Pengguna dengan Sistem.
→ antara Actor dengan Object dari Class Boundary.
- Interaksi antar bagian dalam Sistem.
→ antar Object.
- Interaksi antar bagian dalam Sistem.
→ antara operasi dalam Object.

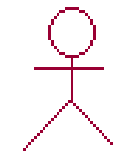


☞ Suatu Object memicu terjadinya aktivitas pada dirinya sendiri (refleksif).

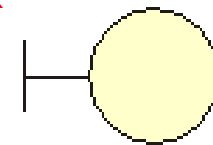
☞ Object dari Class 'ctrBuyTicket' (melalui 'message2') memicu dirinya sendiri untuk melakukan operasi 'message3'.



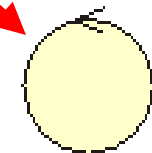
Identifikasi Object



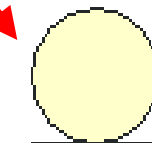
: Petugas



: frmBuyTicket



: ctrBuyTicket

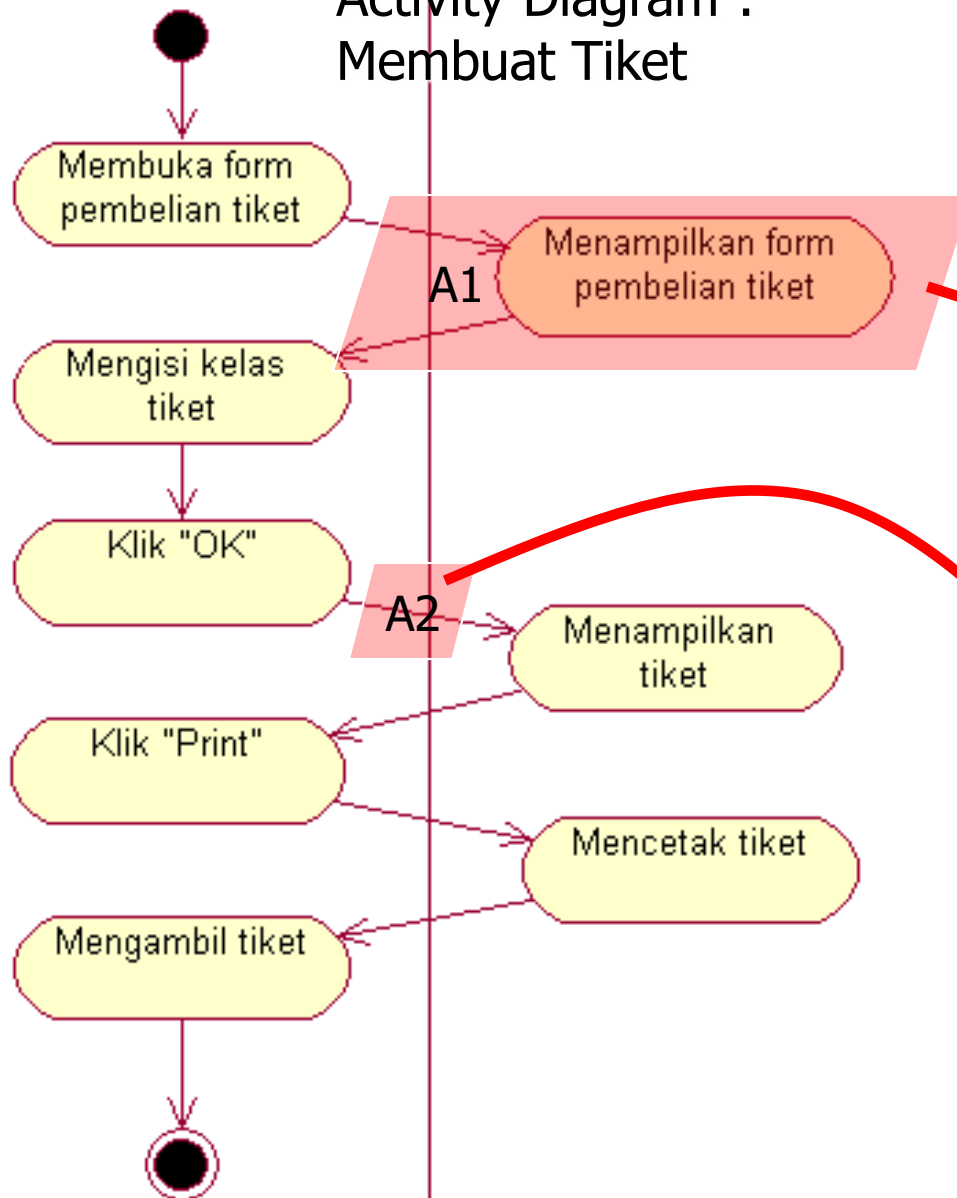


: Ticket

: Petugas

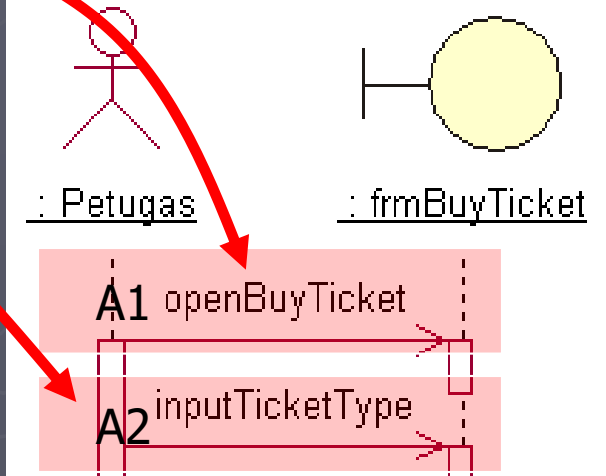
Sistem

Activity Diagram : Membuat Tiket



: Petugas

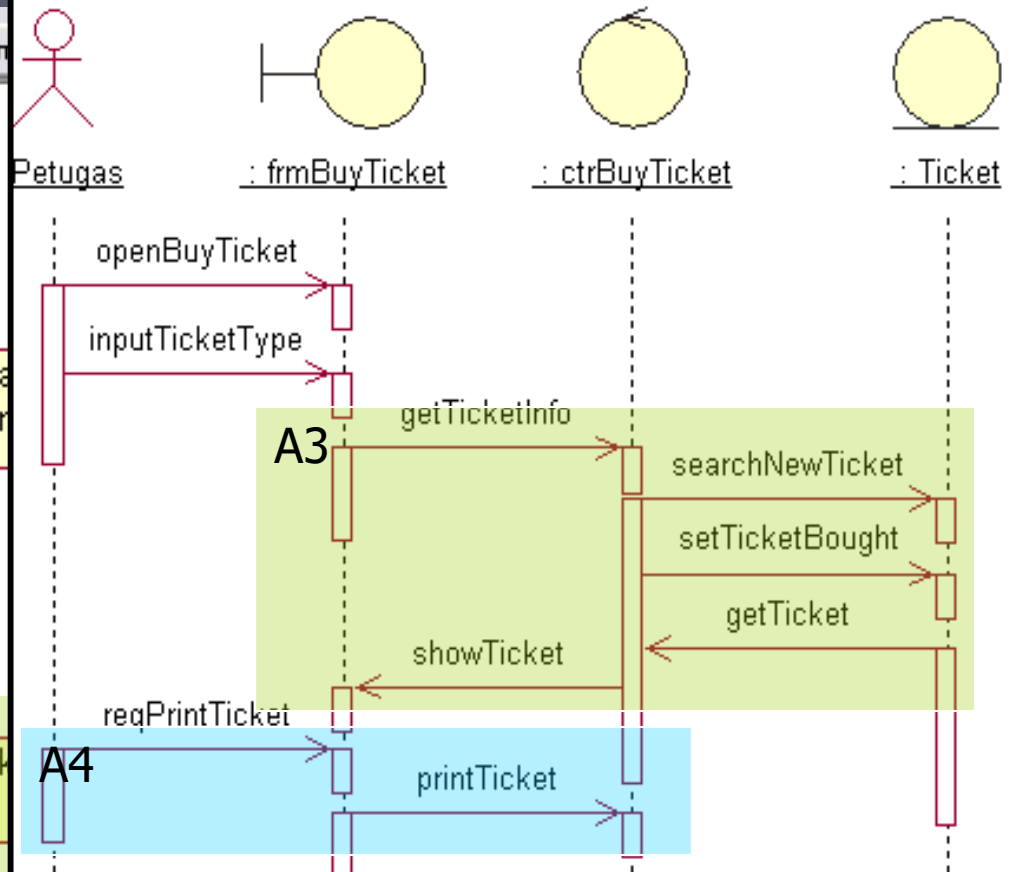
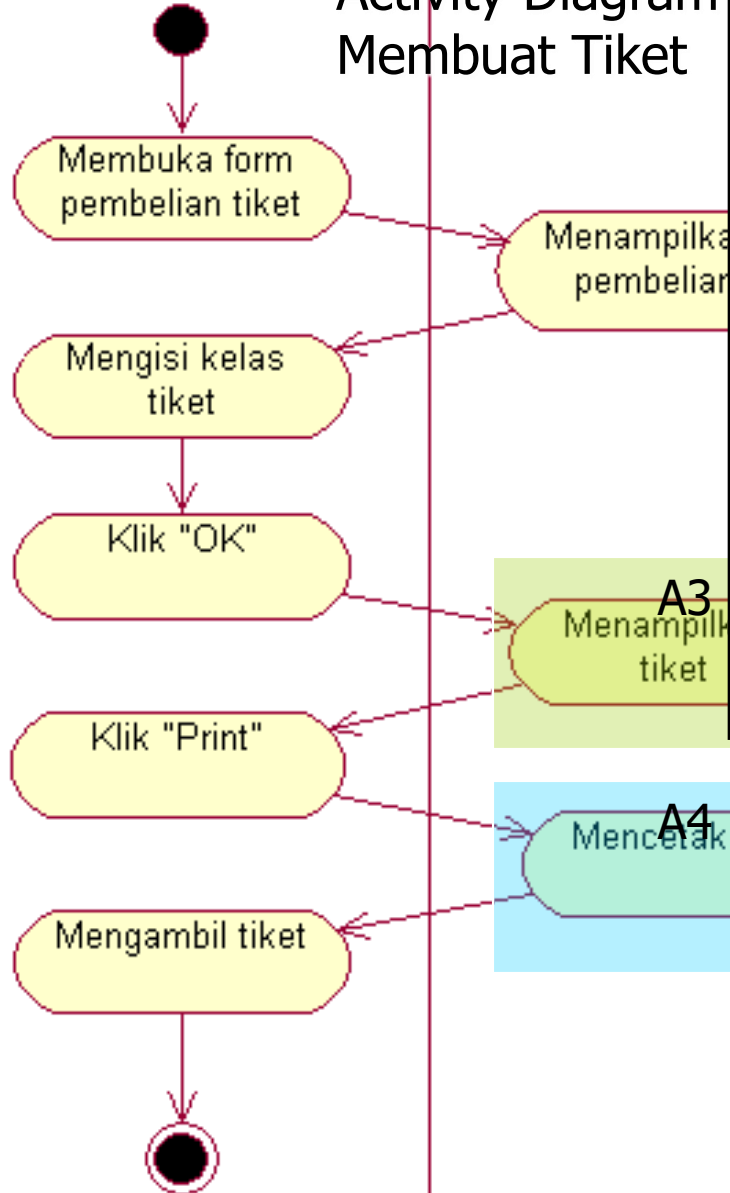
: frmBuyTicket



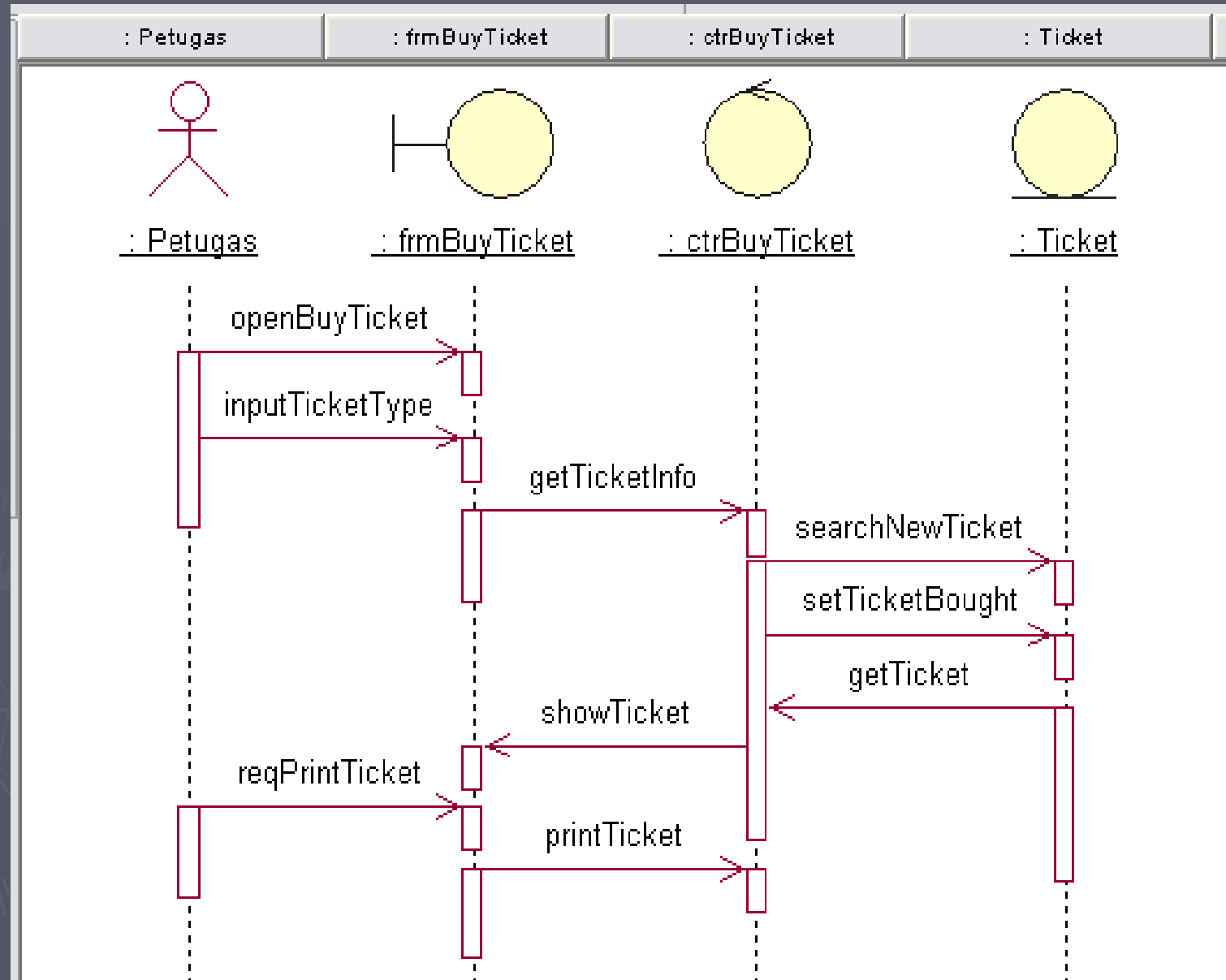
: Petugas

Sistem

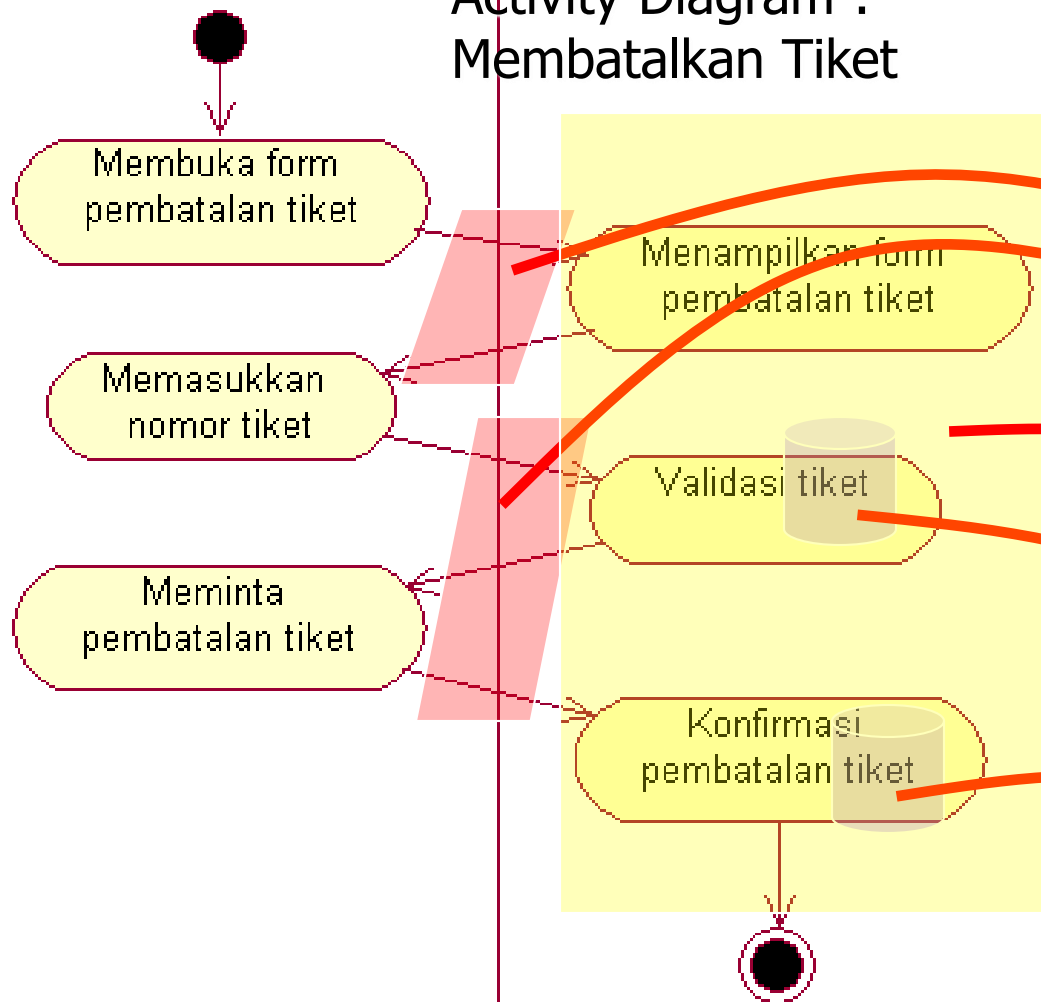
Activity Diagram Membuat Tiket



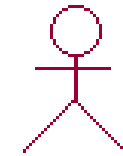
Sequence Diagram : Membuat Tiket



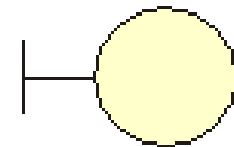
Activity Diagram : Membatalkan Tiket



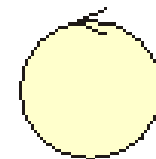
Identifikasi Object



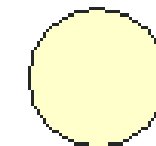
: Petugas



: frmCancelTicket

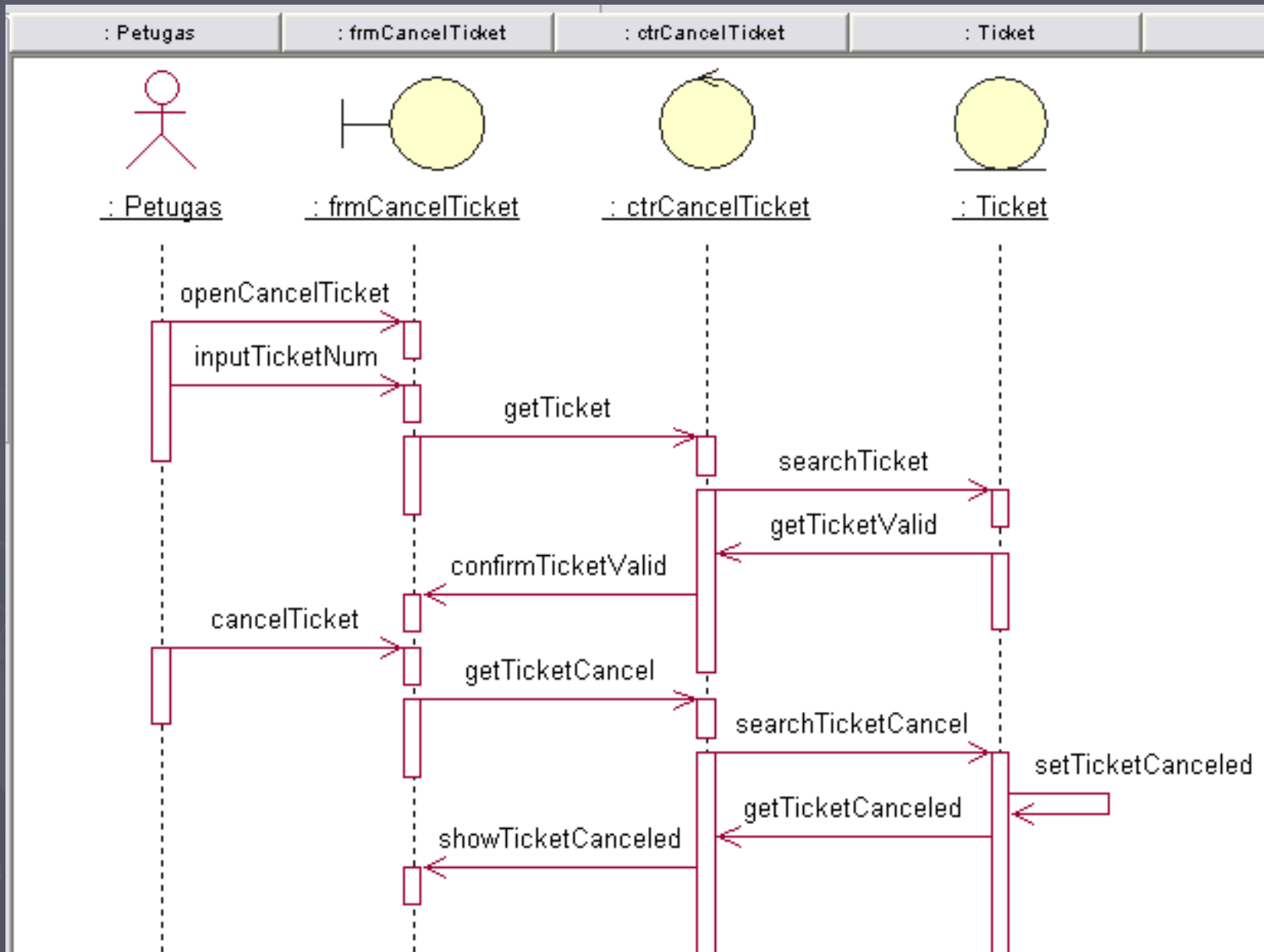


: ctrCancelTicket

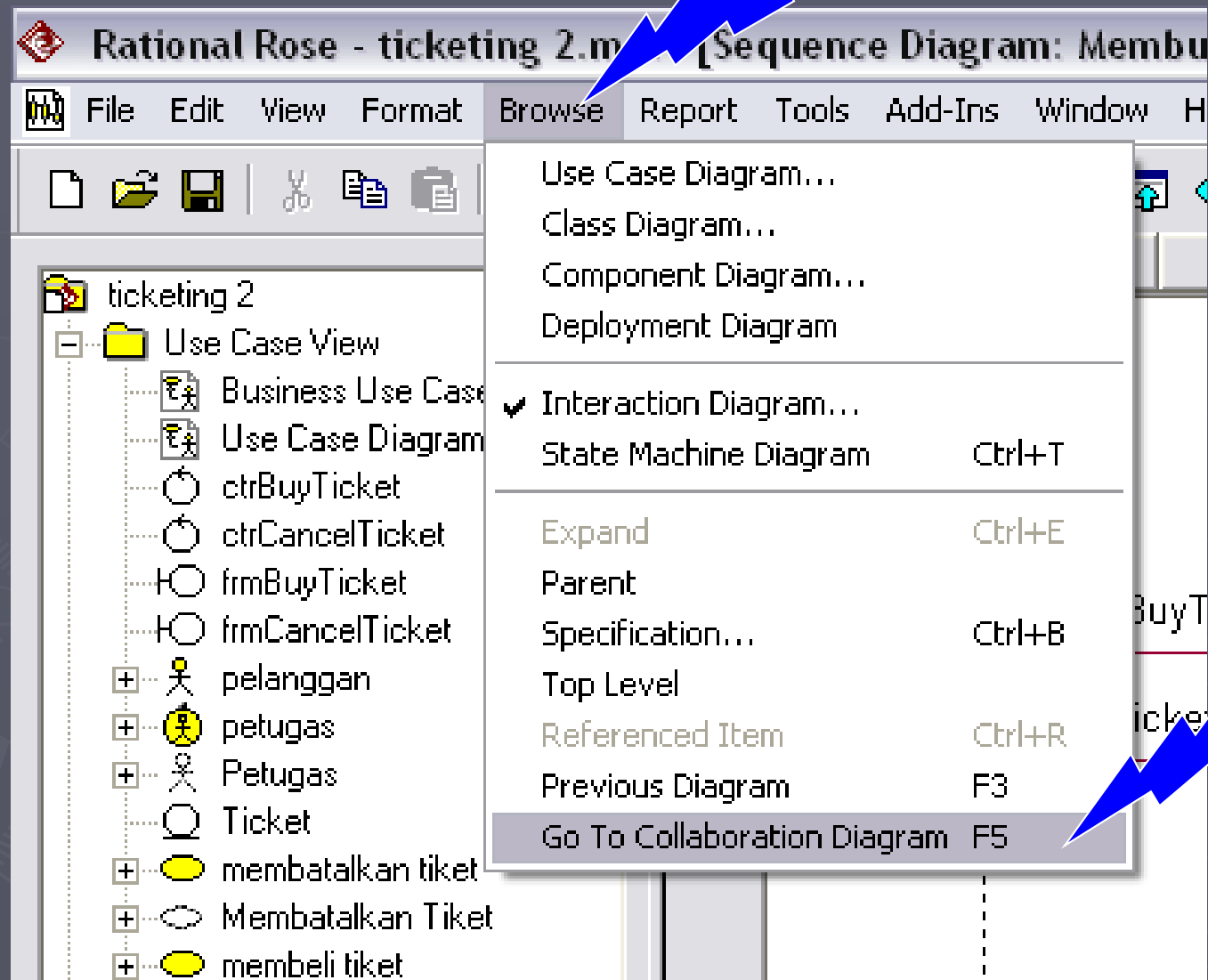


: Ticket

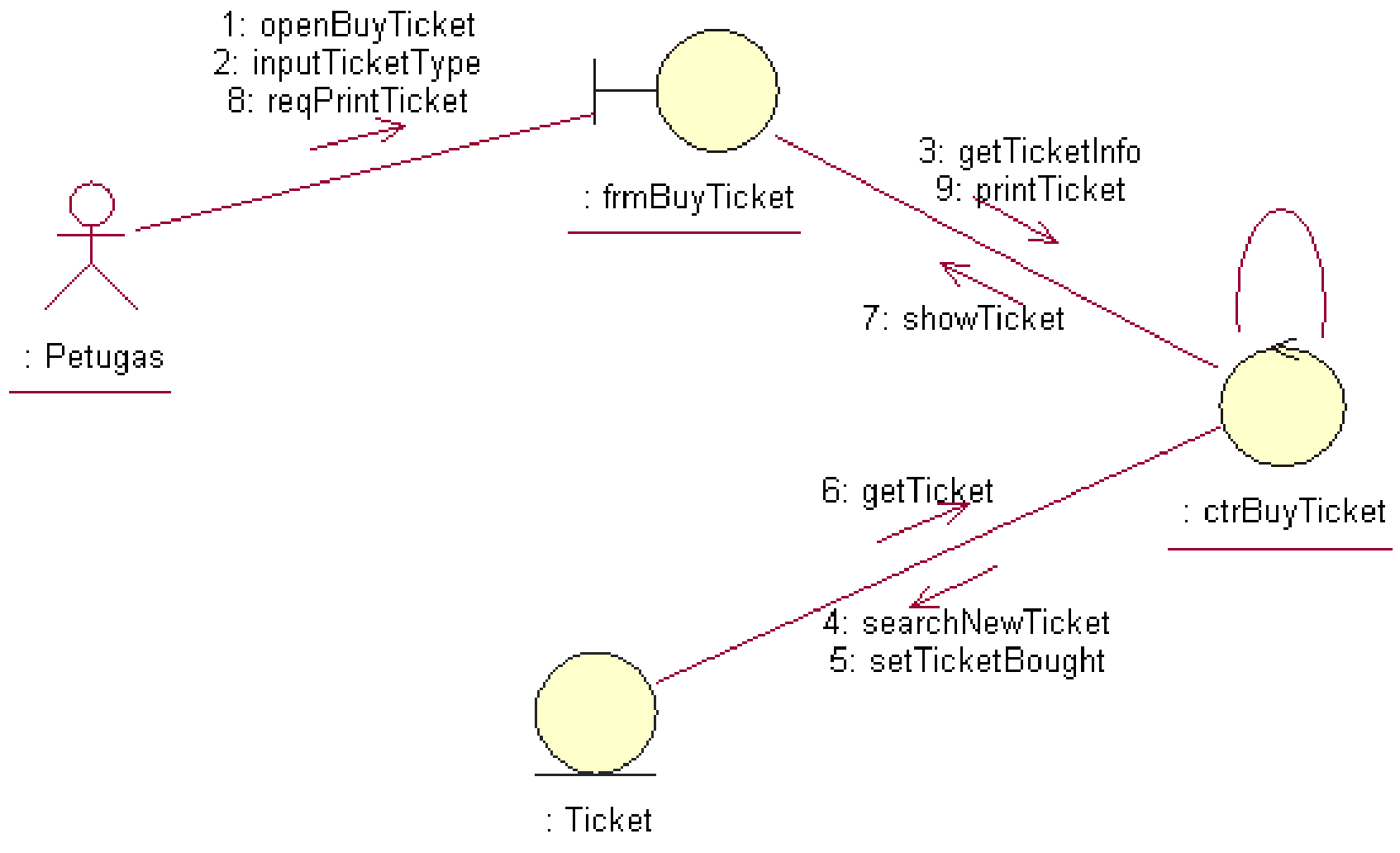
Sequence Diagram : Membatalkan Tiket



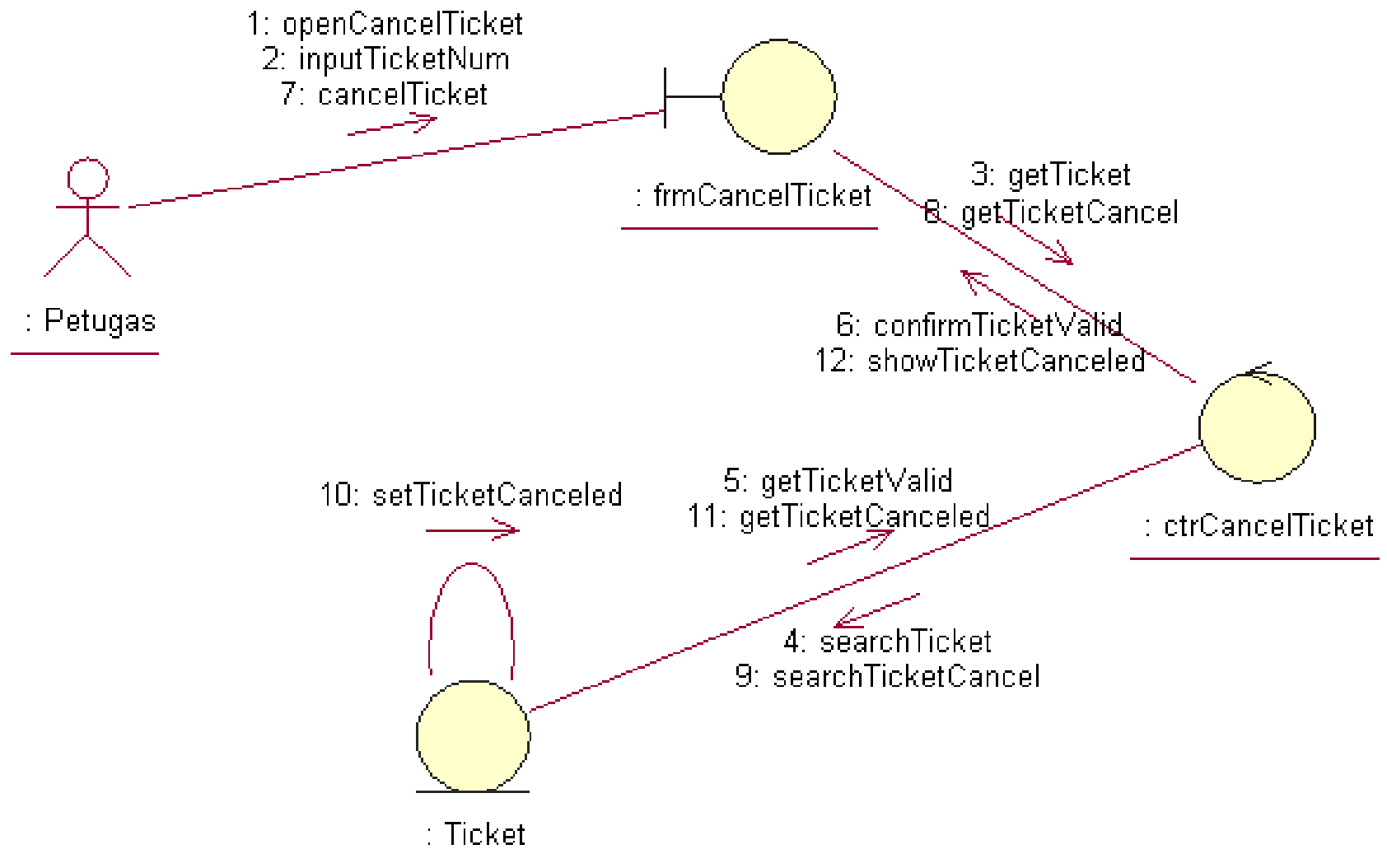
Collaboration Diagram



Collaboration Diagram : Membuat Tiket



Collaboration Diagram : Membatalkan Tiket





Thank You!
😊