MODUL VI

ACTIVITY DIAGRAM

1. Tujuan Praktikum

- Praktikan mengenal tentang Activity Diagram
- Praktikan dapat membuat Activity Diagram
- Praktikan mengimplementasikan Activity Diagram pada kasus modul 1

2. Dasar Teori

2.1. Pengertian Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistemyang sedang dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Pada umumnya activity diagram tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya. Sehingga seringkali diagram ini digunakan untuk memodelkan aktivitas bisnis dalam level konseptual. Diagram ini sangat mirip dengan sebuah flowchart karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas keaktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat(state), akan tetapi perbedaannya dengan flowchart adalah Activity diagram bisa mendukung perilaku pararel sedangkan flowchart tidak bisa.

Membuat Activity diagram terlebih dahulu dalam memodelkan sebuah proses dapat membantu kita memahami proses secara keseluruhan. Activity diagram juga berguna ketika kita ingin menggambarkan perilaku pararel atau menjelaskan bagaimana perilaku dalam berbagai use case berinteraksi.Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Activity diagram adalah variasi dari state diagram yang mana "state"merepresentasikan operasi, dan transisinya merepresentasikan aktivitas yang terjadi pada saat operasi sudah selesai. Sama seperti state, standar UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

2.2. Tujuan Penggunaan Activity Diagram

Activity diagram juga bermanfaat untuk menggambarkan parallel behaviouratau menggambarkan interaksi antara beberapa use case.

2.3. Action dan Activity State

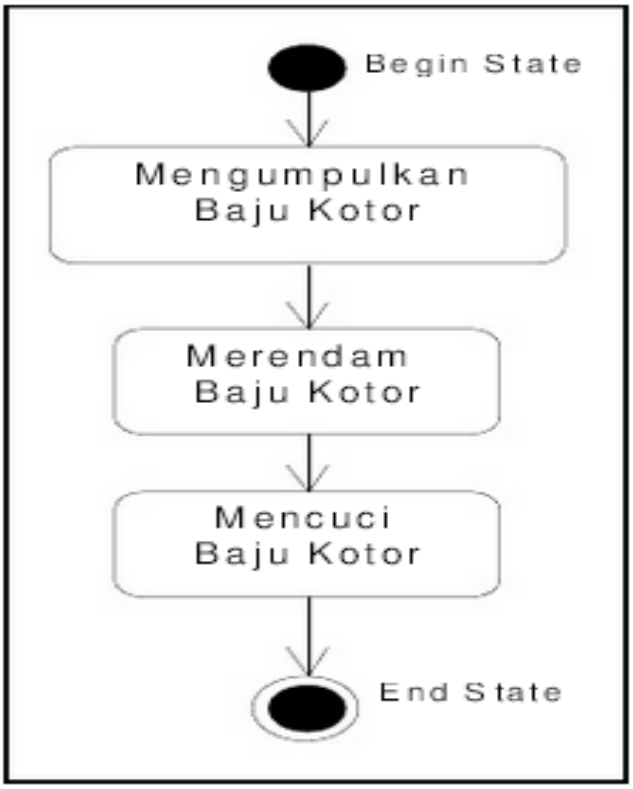
Pada activity diagram, state adalah aksi atau ekspresi yang terjadi pada suatu objek. Action state tidak dapat didekomposisi lebih lanjut, pada umumnya bersifat

atomik, yang berarti event-event mungkin terjadi, tetapi pekerjaan suatu action state tidak akan terinterupsi. Pekerjaan dari suatu action state secara umum dipertimbangkan mengambil waktu eksekusi yang tidak signifikan dibandingkan waktu kerja sistem secara keseluruhan. Sedangkan activity state dapat didekomposisi lebih lanjut, ia tidak bersifat atomik, yang berarti mereka dapat diinterupsi dan secara umum membutuhkan beberapa waktu dari perjalanan sistem atau perangkat lunak untuk terselesaikan dengan tuntas.

Action State Activity State

2.4. Transition

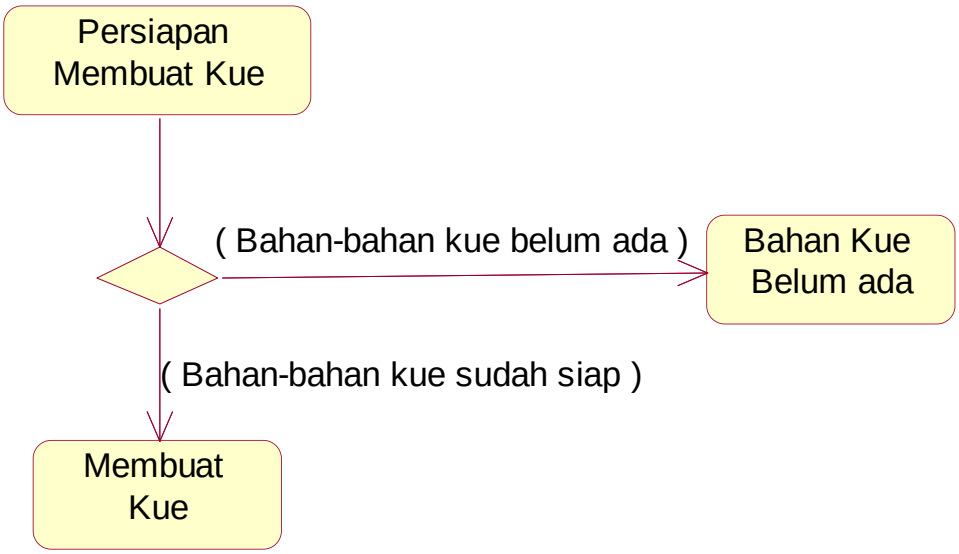
Ketika aksi atau aktifitas dari suatu state diselesaikan, aliran kendali akan menuju ke aksi/aktifitas berikutnya. Kita dapat menspesifikasikan aliran tersebut menggunakan transisi-transisi untuk memperlihatkan lintasan dari satu aksi keaksi berikutnya. Transisi seperti ini dinamakan pemicuan(fired atau trigger)karena control dilewatkan dengan segera setelah suatu state diselesaikan. Kita dapat mengeksekusi aksi keluar(exit action) jika ada, setelah suatu state diselesaikan. Selanjutnya control mengikuti transisi dan dilewatkan ke aksi atau aktifitas berikutnya.Seperti dapat kita lihat pada gambar 4.3 dimana kita mendapatkan gambaran tentang bagaimana langkah-langkah mencuci baju yang kotor . Pertama kita mengumpulkan baju-baju kotor yang akan dicuci terlebih dahulu,kemudian merendamnya, setelah itu baru mencuci baju-baju kotor tersebut.



Gambar 4.3 contoh transisi

2.5. Percabangan(Branching)

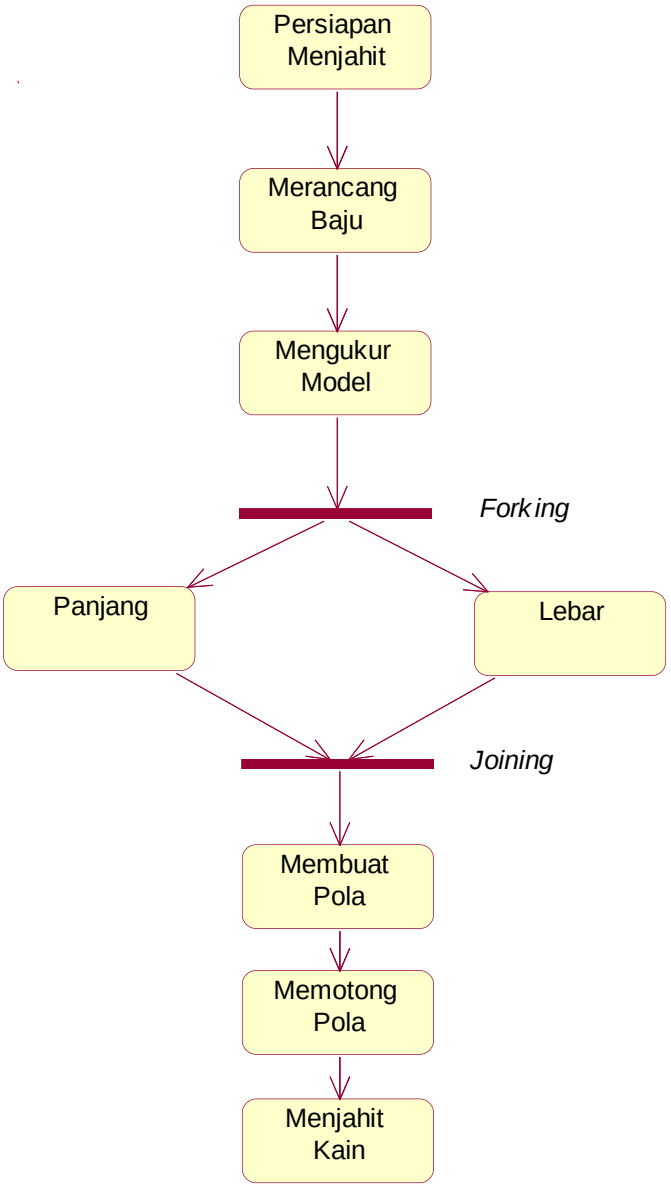
Umumnya transisi terjadi secara berurutan, tetapi bukan berarti semua transisi terjadi secara berurutan, namun kita dapat melibatkan percabangan seperti halnya dalam diagram alir(flowchart). Percabangan dilukiskan dalam bentuk jajaran genjang. Seperti pada gambar 4.4 diperlihatkan proses persiapan membuat kue, jika bahan-bahan kue telah tersedia, maka peroses membuat kue dapat dilakukan, tetapi jika bahan-bahannya belum ada, maka kita harus membeli bahan-bahan kuenya terlebih dulu.



Contoh Percabangan

2.6. Forking dan Joining

Dalam UML kita menggunakan garis sinkronisasi untuk menspesifikasikan forking(satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran) serta joining(beberapa aliran sekaligus yang secara bersamaan masuk menjadi satu titik). Garis sinkronisasi digambarkan dengan garis horisontal tebal. Seperti diperlihatkan pada gambar 4.5, pada gambar tersebut diperlihatkan forking dan joining sekaligus pada proses menjahit kain, mulai dari persiapan sampai proses menjahit kain. Dimana forking terjadi pada saat mengukur model,yang terbagi menjadi dua aliran yaitu mengukur panjang dan lebar, sedangkan joining terjadi pada saat membuat pola dari aliran mengukur panjang dan lebar tadi.



Contoh Forking dan Joining

2.7. Unsur-unsur utama Activity Diagram

Sebelum kita membuat activity diagram untuk menjelaskan aliran-aliransecara rinci dari setiap use case, terlebih dahulu kita akan berkenalan dengantoolbar yang nantinya akan dipergunakan untuk membuat activity diagram,seperti diperlihatkan pada tabel 4.1 di bawah ini:

Ikon	Nama button	Fungsi
	Select/Deselect an item	Mengembalikan pointer mouse ke ikon ini memungkinkan kita memilih item-item tertentu
	Text Box	Menambahkan boks teks ke diagram
	Note	Menambahkan catatan pada diagram
	Anchor Note to item	Melekatkan catatan pada state atau activity tertentu dalam diagram
	State	Menambahkan state untuk suatu objek
	Activity	Menambahkan aktifitas baru pada diagram
	Start State	Memperlihatkan dimana aliran kerja berawal
	End State	Memperlihatkan dimana aliran kerja berakhir
	State Transition	Menambah transisi dari suatu aktifitas ke aktifitas lainnya
	Transition to Self	Menambah transisi rekursif
	Horizontal Synchronization	Menambahkan sinkronisasi horisontal pada diagram
	Vertical Synchronization	Menambahkan sinkronisasi vertikal pada diagram
	Decision	Menambahkan titik keputusan pada aliran kerja
	Swimlane	Menambahkan swimlane(umumnya digunakan pada pemodelan bisnis)

Tabel 4.1 Toolbar untuk activity diagram

Adapun unsur-unsur utama dari activity diagram meliputi :

- **Swimlane**

Memperlihatkan siapa yang bertanggungjawab untuk melaksanakan tugas- tugas tertentu pada activity diagram.

- **Activity**

Menggambarkan langkah-langkah dalam aliran kerja

- **Action**

Merupakan langkah-langkah dalam activity. Aksi atau action mungkinterjadi saat event tertentu memasuki activity, saat event meninggalkan(*exit*) activity, terjadi di dalam activity atau setelah event tertentu (spesifik).

- **Objek**

Merupakan entitas-entitas yang dipengaruhi aliran kerja

- **Transitions**

Memperlihatkan bagaimana aliran-aliran kerja bergerak dari satu activity keactivity lainnya.

- **Desicions**

Memperlihatkan dimana keputusan perlu diambil selama terjadi aliran-alirankerja.

- **Synchronizations**

Memperlihatkan bagaimana dua atau lebih langkah pada aliran-aliran kerjaterjadi secara simultan.

- **Start State**
- **End State**

3. Langkah Praktikum

3.1. Membuat Activity Diagram

Untuk menambahkan activity diagram pada suatu use case, lakukan langkah-langkah berikut ini :

1. **Klik kanan** pada use case tertentu.
2. Pilih **New Activity Diagram**

.3. Rational Rose akan menambahkan **State/Activity Model** di dalam usecase.

4. Beri nama activity diagram yang baru dengan **BookActivity.**

5. **Klik ganda** activity diagram yang baru untuk membukanya.

3.2. Menambahkan ikon-ikon pada Activity Diagram

1. **Klik** ikon **Start State** pada toolbar dan taruh pada diagram kemudian ubah namanya menjadi **Mulai.**
2. **Klik** ikon **Activity** pada toolbar dan taruh pada diagram kemudian ubah namanya menjadi **Cari Barang (by Nama Barang).**
3. Kemudian tambahkan lagi tujuh buah (**7**) ikon **Activity** dan ubah namanya:

- **Pilih barang yang akan dipesan**
- **Pesan barang**
- **Pengiriman Informasi Pesanan via Email**
- **Transfer Uang**
- **Konfirmasi SMS**
- **Pengiriman barang**

4. Tambahkan sebuah ikon **Decision** dan ubah nama menjadi **Persetujuan Pembelian**

5. Tambahkan sebuah ikon **End State** dan ubah nama menjadi **Selesai.**
6. Hubungkan antar **Start State** dengan **Activity Decision** dan **End State** menggunakan **State Transition.**
7. Dan tambahkan dua buah (2) ikon **Text Box** untuk **Batal** dan **OK** pada Decision.8. Sehingga tampilan seperti dibawah ini :

3.3. Menghapus ikon pada Activity Diagram

1. **Klik** pada ikon yang akan dihapus kemudian tekan tombol **Delete.**

3.4. Menghapus Diagram

1. Pilih **activity** diagram yang akan kita hapus.

2. **Klik kanan** dan pilih **Delete**

Latihan

1. **Buatlah Diagram Activity yang menjelaskan setiap skenario yang ada dalam kasus pada modul 1 (karena contoh diatas adalah Diagram Activity secara global untuk kasus pemesanan), yaitu :**

a. Proses Registrasi

b. Proses Pemesanan yang terdiri dari Login dan Informasi Pemesanan(melalui Email).

3. Proses Pembayaran yang terdiri dari Transfer Uang dan PengirimanBarang.

2. **Buatlah diagram Activity dari system perpustakaan**

LEMBAR KERJA PRAKTIKUM

Paraf Dosen/Asisten

Tanggal :

DAFTAR PUSTAKA

Erwin, M. AH, *Rekayasa Perangkat Lunak,*Teknik Informatika, FakultasTeknologi Industri, Universitas Islam Indonesia, Yogyakarta, 2004.

Nugroho, Adi, *Rational Rose untuk Pemodelan Berorientasi Objek,* INFORMATIKA, Bandung, 2005.Setiawan, M. Andri

A.S Rosa , dan M.Shalahuddin. 2014. Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek. Bandung : Informatika.