DIAGRAM SEQUENCE UML

Makalah ini di susun oleh :

1) Banu Hardian (51412367)

2) Mutia Sulisetyani (55412178)

3) Raditya Rafian (55412868)

4) Tio Dwi Akbar (57412395)

GUNADARMA UNIVERSITY

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Kuasa atas segala limpahan Rahmat, Inayah, Taufik dan Hinayahnya sehingga saya dapat menyelesaikan penyusunan makalah ini dalam bentuk maupun isinya yang sangat sederhana. Semoga makalah ini dapat dipergunakan sebagai salah satu acuan, petunjuk maupun pedoman bagi pembaca dalam membuat sebuah diagram sequence UML.

Harapan saya semoga makalah ini membantu menambah pengetahuan dan pengalaman bagi para pembaca, sehingga saya dapat memperbaiki bentuk maupun isi makalah ini sehingga kedepannya dapat lebih baik.

Makalah ini saya akui masih banyak kekurangan karena pengalaman yang saya miliki sangat kurang. Oleh kerena itu saya harapkan kepada para pembaca untuk memberikan masukan-masukan yang bersifat membangun untuk kesempurnaan makalah ini.

Depok, Mei 2014

BAB I
PENDAHULUAN

1. Latar Belakang

UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2. Tujuan

Adapun tujuan yg ingin dicapai pada makalah ini, antara lain:

a. Mahasiswa mampu mendeskripsikan tentang UML

b. Mahasiswa dan dosen mampu menguasai UML dan Diagram Sequential

BAB II
PEMBAHASAN

1. Pengertian UML

UML (Unified Modeling Language) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

Bagian-bagian UML

Bagian-bagian utama dari UML adalah view, diagram, model element, dan general mechanism.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis view dalam UML antara lain: *use case view, logical view, component view, concurrency view, dan deployment view.*

Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. Actor yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

View ini digambarkan dalam use case diagrams dan kadang-kadang dengan activity diagrams. View ini digunakan terutama untuk pelanggan, perancang *(designer)*, pengembang *(developer)*, dan penguji sistem *(tester)*.

Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis *(class, object,danrelationship)* dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke objectlain dalam suatu fungsi tertentu.

View ini digambarkan dalam class diagrams untuk struktur statis dan dalam *state, sequence, collaboration, dan activity diagram* untuk model dinamisnya. View ini digunakan untuk perancang *(designer)* dan pengembang *(developer)*.

Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya.

View ini digambarkan dalam component view dan digunakan untuk pengembang (developer).

Concurrency view

Membagi sistem ke dalam proses dan prosesor. View ini digambarkan dalam diagram dinamis *(state, sequence, collaboration, dan activity diagrams)* dan diagram implementasi *(component dan deployment diagrams)* serta digunakan untuk pengembang *(developer)*, pengintegrasi *(integrator)*, dan penguji *(tester)*.

Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat *(nodes)* dan bagaimana hubungannya dengan lainnya.

View ini digambarkan dalam deployment diagrams dan digunakan untuk pengembang (developer), pengintegrasi *(integrator)*, dan penguji *(tester)*.

b. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Use case merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

Class Diagram

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku *(operasi)* dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram.Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system.

Component Diagram

Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis. Komponent merupakan implementasi software dari sebuah atau lebih class. Komponent dapat berupa source code, komponent biner, atauexecutable component. Sebuah komponent berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view.Sehingga component diagram merepresentasikan dunia riil yaitu component software yang mengandung component, interface dan relationship.

Deployment Diagram

Menggambarkan tata letak sebuah system secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat *(nodes)* satu sama lain dan jenis hubungannya. Di dalam nodes,executeable

component dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

State Diagram

Menggambarkan semua state *(kondisi)* yang dimiliki oleh suatu object dari suatu classdan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti sequence diagrams. Dalam menunjukkan pertukaran pesan, collaboration diagrams menggambarkan object dan hubungannya *(mengacu ke konteks)*. Jika penekannya pada waktu atau urutan gunakansequencediagrams, tapi jika penekanannya pada konteks gunakan collaboration diagram.

Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

Tujuan Penggunaan UML

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

4. UML bisa juga berfungsi sebagai sebuah *(blue print)* cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram *(reserve enginering)*.

2. Sequence Diagram

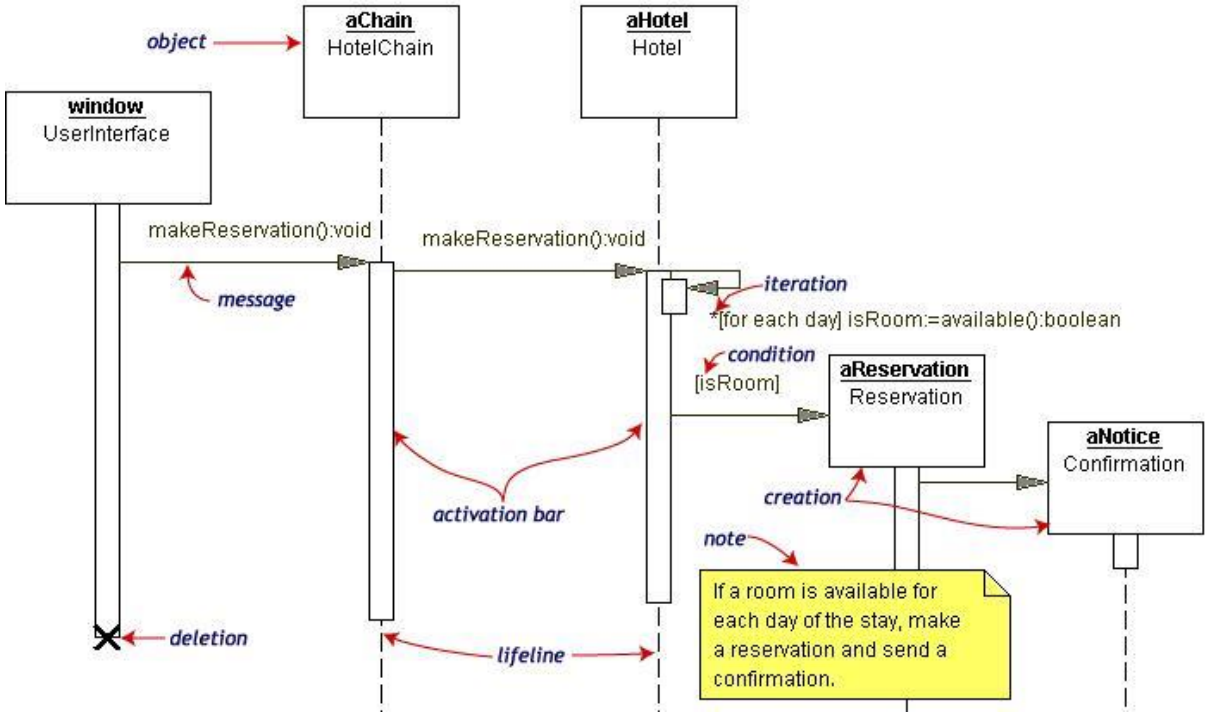
Pengertian Sequence Diagram

Sequence Diagram merupakan Intraction Diagram yang digunakan untuk menjelaskan eksekusi sebuah skenario semantik. Sequence Diagram juga digunakan untuk menjelelaskan interaksi antar objek dalam urutan waktu (Booch, Maksimchuk, Engle, Young, Conallen, & Houston, 2007).

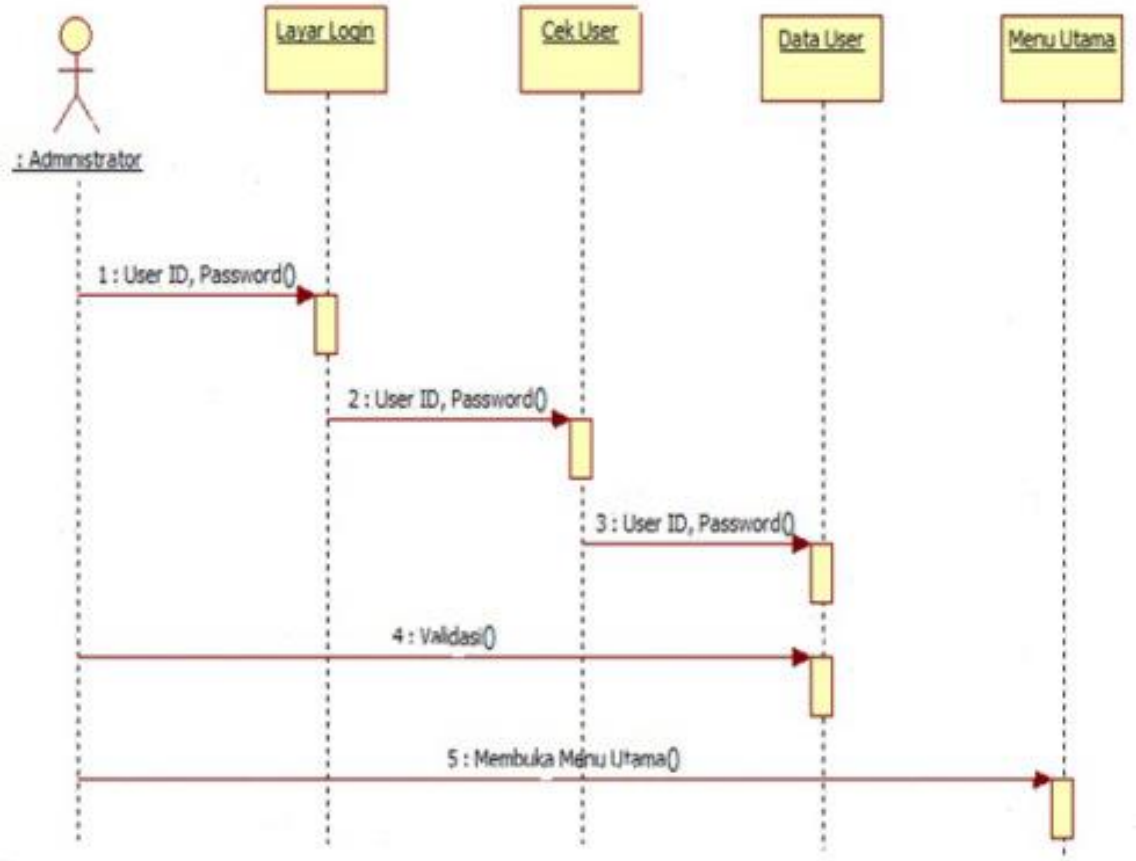
Sequence Diagram bisa digunakan untuk menjelaskan sebuah serangkaian langkah-langkah yang mengirimkan message antar satu lifeline ke lifeline yang lain. Setiap message yang dikirimkan bisa memberikan respon (return) relatif pada skenario yang dirancang di Use Case Diagram. Interaksi yang terjadi bisa bersifat instansiasi sebuah object maupun static method dari sebuah class.

Diagram Class dan diagram Object merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti Diagram Interaction. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Di bawah ini adalah diagram Sequence untuk pembuatan Hotel Reservation. Obyek yang mengawali urutan message adalah ‘aReservation Window’.

Contoh Sequence Diagram



'Reservation window' mengirim pesan makeReservation() ke 'HotelChain'. Kemudian 'HotelChain' mengirim pesan yang sama ke 'Hotel'. Bila 'Hotel' punya kamar kosong, maka dibuat 'Reservation' dan 'Confirmation'. Lifeline adalah garis dot (putus-putus) vertikal pada gambar, menerangkan waktu terjadinya suatu obyek. Setiap panah yang ada adalah pemanggilan suatu pesan. Panah berasal dari pengirim ke bagian paling atas dari batang kegiatan (activation bar) dari suatu pesan pada lifeline penerima. Activation bar menerangkan lamanya suatu pesan diproses. Pada gambar diagram , terlihat bahwa 'Hotel' telah melakukan pemanggilan diri sendiri untuk pemeriksaan jika ada kamar kosong. Bila benar, maka 'Hotel'membuat 'Reservation' dan 'Confirmation'. Pemanggilan diri sendiri disebut dengan iterasi. Expression yeng dikurung dengan "[ ]", adalah condition (keadaan kondisi). Pada diagram dapat dibuat note (catatan). Pada gambar, terlihat seperti selembar kertas yang berisikan teks. Note bisa diletakan dimana saja pada diagram UML.



Gbr 8. Contoh Diagram Sequence

Pada contoh diagram sequence di atas terdapat 1 administrator dan 3 objek, yaitu: Layar login, cek user, data user, menu utama. Pertama-tama administrator akan masuk ke layar login dengan menggunakan User ID dan Password(). Dari Layar login, admin akan melakukan cek user dengan memasukkan User ID dan Password(). Setelah melakukan cek user, admin akan memasukkan user ID dan password sekali lagi untuk melihat data user. User ID dan Password yang dimasukkan admin sebanyak 3 kali, digunakan untuk melakukan validasi. Validasi ini bertujuan untuk membuka menu utama().

BAB III
KESIMPULAN

1. KESIMPULAN

- Dengan menggunakan UML dapat memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
- Dapat menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
- Dapat memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- UML bisa juga berfungsi sebagai sebuah *(blue print)* cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram *(reserve enginering)*.
- Sequence Diagram merupakan Intraction Diagram yang digunakan untuk menjelaskan eksekusi sebuah skenario semantik.