Nama : Farhan Aziz

NPM : 2113191097

Pelajaran : Pemodelan Sistem Berbasis Objek

Jurusan : S1 Teknik Informatika/A2

Soal :

1. Buat Artikel mengenai Sejarah **UML**, Fungsi **UML**, dan penjelasan **Setiap Diagram UML!**

## JAWABAN!!

- **PENGERTIAN UML**

**UML** adalah bahasa untuk menspesifikasi,memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak)dari sistem perangkat lunak,seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object.UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corps. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespetktif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

- **SEJARAH UML**

**UML** dimulai secara resmi pada Oktober 1994, ketika Rumbaugh menggabungkan kekuatan dengan Booch. Mereka berdua lalu bekerja bersama di Relational Software Cooperation. Proyek ini memfokuskan pada penyatuan metode booch dan Rumbaugh(OMT). Pada bulan October 1995,

UML merilis versi 0.8 dan pada waktu yang sama juga Jacobson bergabung dengan Relational. Cakupan dari UML pun semakin meluas. Kemudian dibangunlah persatuan untuk UML dengan beberapa organisasi yang akan menyumbangkan sumber dayanya untuk bekerja, mengembangkan,dan melengkapi UML.

Banyak partner yang berkontribusi pada UML 1.0, diantaranya Digital Equipment Corporation, Hawlett-Packard, I-Logix, IBM, ICON Computing, MCI systemhouse, Microsoft, Oracle, Relation, Texas Insturments dan Unisys. Dari kolaborasi ini dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, expressive, kuat dan cocok untuk lingkungan masalah yang luas. Dan pada January 1997, UML dijadikan sebagai standar bahasa pemodelan.

- **FUNGSI UML**

  Inilah beberapa tujuan atau fungsi dari penggunaan UML, yang diantaranaya:

  - Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemerograman maupun proses rekayasa.
  - Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
  - Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
  - Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
  - Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (software) saja.
  - Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

- **BAGIAN-BAGIAN UML**

**1. 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.


Gambar Use Case Diagram

Diagram Use Case berguna dalam tiga hal :

- Menjelaskan fasilitas yang ada (requirements)
- Use Case baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan design menjadi lebih jelas.
- Komunikas dengan klien
- Penggunaan notasi dan simbol dalam diagram Use Case membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.
- Membuat test dari kasus-kasus secara umum
- Kumpulan dari kejadian-kejadian untuk Use Case bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.

**2. Activity Diagram**

Pada dasarnya diagram Activity sering digunakan oleh flowchart. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus padaobyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram Activity berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata

lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Sebagai contoh, perhatikan proses yang terjadi. “Pengambilan uang dari bank melalui ATM.” Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait yaitu : Customer, ATM, and Bank. Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran stop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi. Lihat gambar di bawah ini, agar lebih jelas :

Contoh Diagram **Activity Pengambilan Uang melalui ATM.**

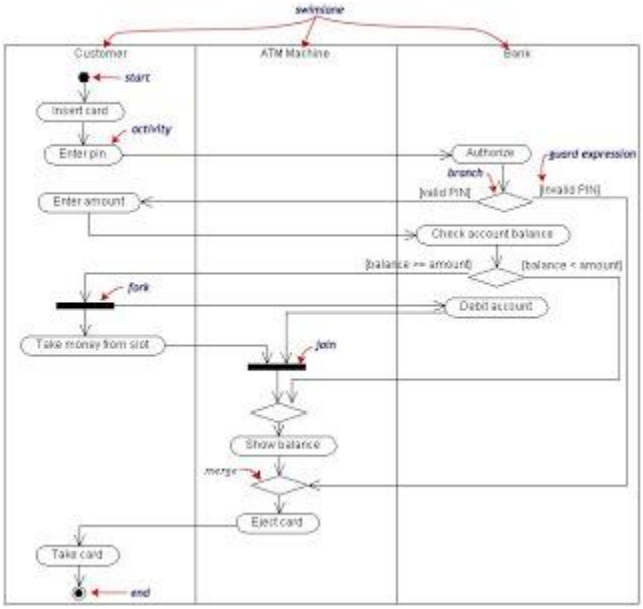
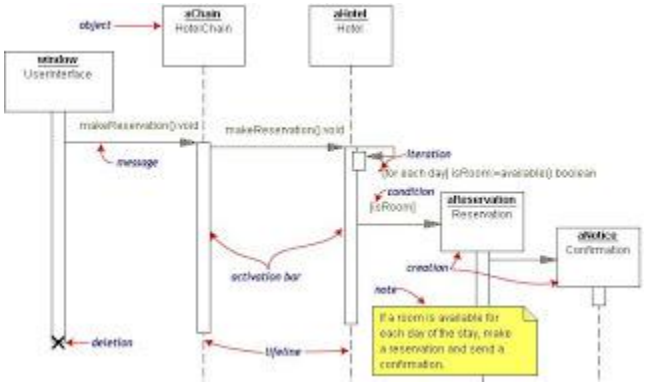



Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal (single transition) timbul dari setiap adanya activity (aktifitas), yang saling menghubungi pada aktifitas berikutnya. Sebuah transition (transisi) dapat membuat cabang ke dua atau lebih percabangan exclusive transition (transisi eksklusif). Label Guard Expression(ada didalam [ ]) yang menerangkan output (keluaran) dari percabangan. Percabangan akan menghasilkan bentuk menyerupai bentuk intan. Transition bisa bercabang menjadi beberapa aktifitas paralel yang disebut Fork. Fork beserta join (gabungan dari hasil output fork) dalam diagram berbentuk solid bar (batang penuh).

**3. Sequence Diagram**

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 Diagram Sequence **‘Pemesanan kamar di Hotel’.**



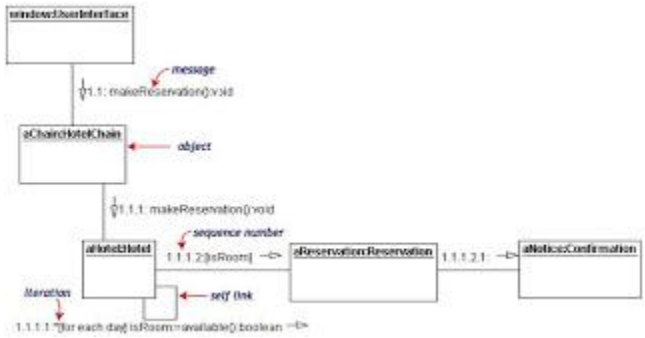

‘Reservation window’ mengirim pesan makeReservation() ke ‘HotelChain’. Kemudian ‘HotelChain’ mengirim pesan yang sama ke ‘Hotel’. Bila ‘Hotel’ punya kamar kosong, maka dibuat ‘Reservation’ dan ‘Confirmation’. Lifeline adalah garis dot (putus-putus) vertikal pada gambar, menerangkan waktu terjadinya suatu 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.

**4. Communication Diagram (Collaboration diagram in versi 1.x)**

Collaboration diagram menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek. Setiap messagememiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama. Diagram Collaboration juga merupakan diagram interaction. Diagram membawa informasi yang sama dengan diagram Sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.

Contoh Diagram Collaboration ‘Pemesanan kamar di Hotel’.




Kotak kegiatan obyek diberi label dengan nama kelas atau obyek (atau keduanya). Nama kelas dibatasi dengan colons / titik dua ( : ). Setiap pesan pada diagram Collaboration mempunyai angka yang terurut. Pesan yang tingkatannya tertinggi adalah angka 1. Pesan yang berada pada tingkat yang sama memiliki prefix yang sama, namunsuffix berbeda bergantung pada posisinya; hanya untuk angka 1, 2, dan seterusnya.

## 5. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Classmenggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperticontainment , pewarisan, asosiasi, dan lain-lain.

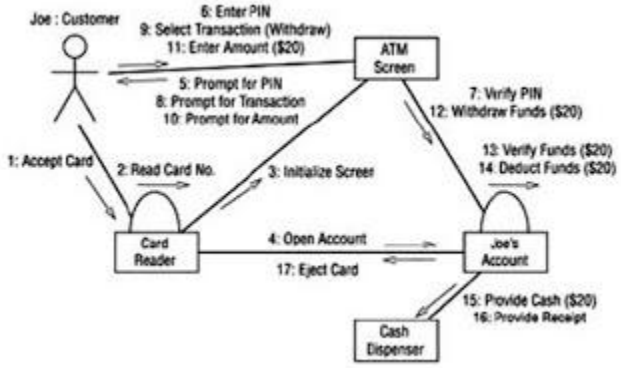
Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- Private, tidak dapat dipanggil dari luar class yang bersangkutan
- Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- Public, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Sesuai dengan perkembangan class model, classdapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.




Gambar Class Diagram

Hubungan Antar Class

1. Asosiasi, yaitu hubungan statis antar class . Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi classlain. Panah navigability menunjukkan arah query antar class.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar class . Class dapat diturunkan dari classlain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan ( message ) yang di-passing dari satuclass kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

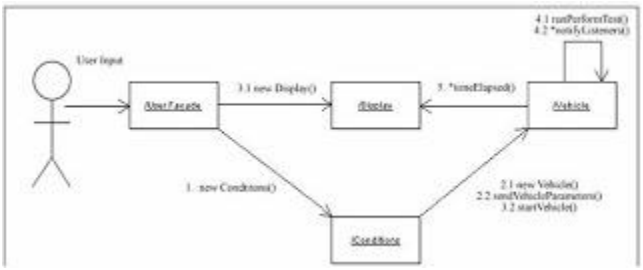
**6. State Machine Diagram (Statechart diagram in versi 1.x)**

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state kestate lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram ). Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar stateumumnya memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari eventtertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.


Gambar State Machine Diagram (Statechart diagram in versi 1.x)

**7. Component Diagram**

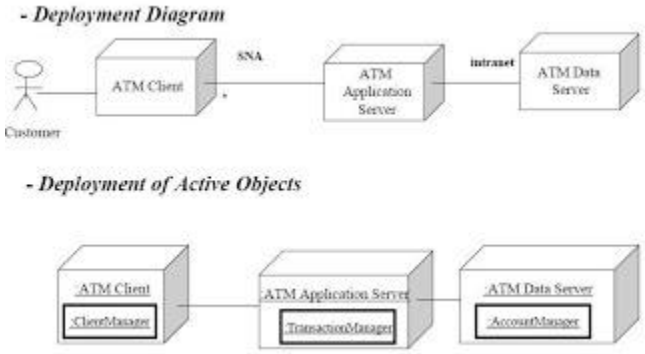
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan ( dependency ) di antaranya. Komponen piranti lunak adalah modul berisi code , baik berisi source code maupun binary code , baik library maupunexecutable , baik yang muncul pada compile time, link time , maupun run time . Umumnya komponen terbentuk dari beberapa class dan/atau package , tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface , yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.




Gambar Component Diagram

**8. Deployment Diagram**

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah node adalah server, workstation , atau piranti keras lain yang digunakan untuk men- deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.




Gambar Deployment Diagram

**9. Composite Structure Diagram**

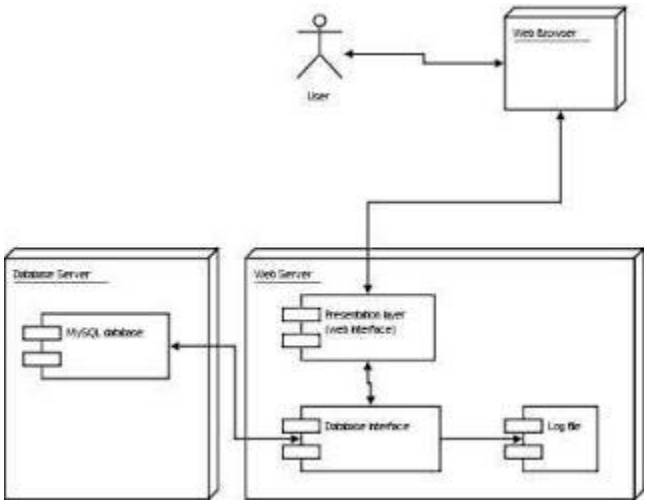
Diagram struktur komposit adalah diagram yang menunjukkan struktur internal classifier, termasuk poin interaksinya ke bagian lain dari sistem. Hal ini menunjukkan konfigurasi dan hubungan bagian, yang bersama-sama melakukan perilaku classifie. Diagram struktur komposit merupakan jenis diagram struktur statis dalam Unified Modeling

Language (UML), yang menggambarkan struktur internal kelas dankolaborasi.

Struktur komposit dapat digunakan untuk menjelaskan :

- Struktur dari bagian-bagian yang saling berkaitan
- Run-time struktur yang saling berhubungan

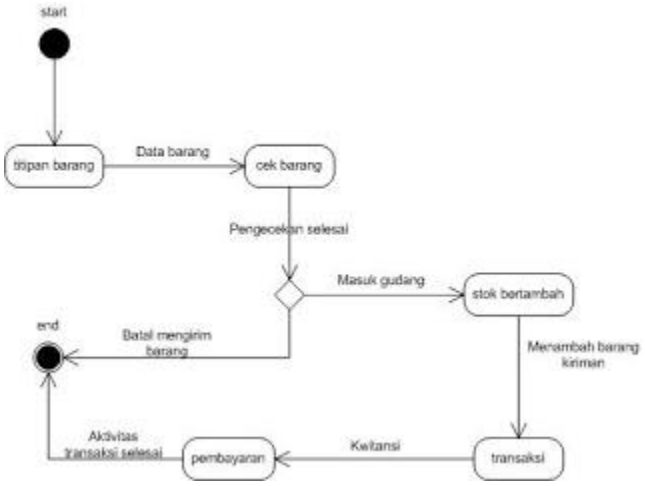
Contoh : Deskripsi dari bagian-bagian mesin yang saling berhubungan untuk melakukan fungsi mesin.




Gambar Composite Structure Diagram

**10. Interaction Overview Diagram**

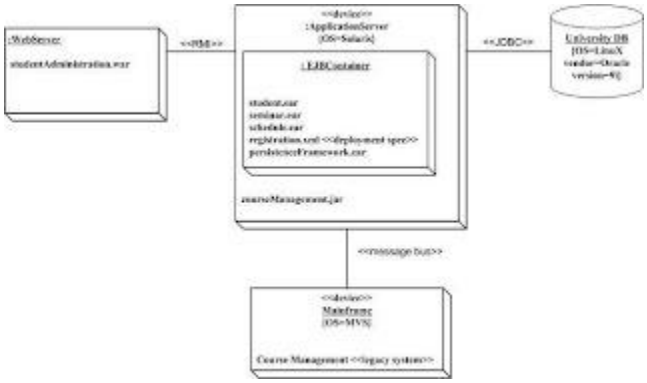
Interaction Overview Diagram adalah pencangkokan secara bersama antara activity diagram dengan sequence diagram. Interaction Overview Diagram dapat dianggap sebagai activity diagram dimana semua aktivitas diganti dengan sedikit sequence diagram, atau bisa juga dianggap sebagai sequence diagram yang dirincikan dengan notasi activitydiagram yang digunakan untuk menunjukkan aliran pengawasan.




Gambar Interaction Overview Diagram

**11. Object Diagram**

Object diagram merupakan sebuah gambaran tentang objek-objek dalam sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah 29 daripada class,object diagram lebih sering disebut sebagai sebuah diagram perintah.



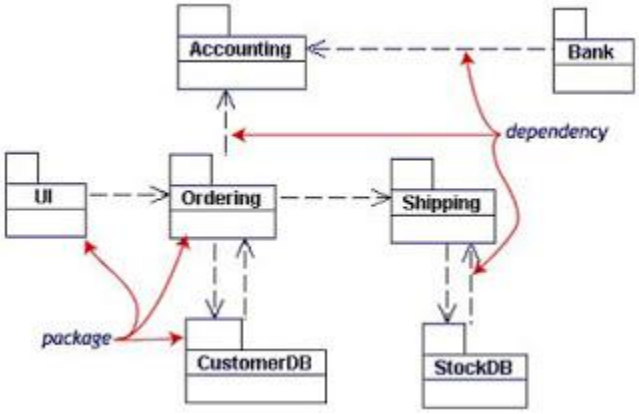
Gambar Object Diagram

**12. Package Diagram**

Diagram objek melengkapi notasi grafik untuk pemodelan objek, kelas dan relasinya dengan yang lain. Diagram objek bermanfaat untuk pemodelan abstrak dan membuat perancangan program. Untuk mengatur pengorganisasian diagram Class yang kompleks, dapat dilakukan pengelompokan kelas-kelas berupa package (paket-paket). Packageadalah kumpulan elemen-elemen logika UML. Gambar di bawah

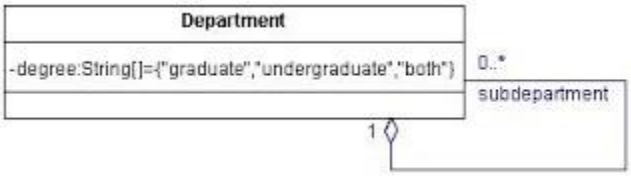
ini mengenai model bisnis dengan pengelompokan kelas-kelas dalam bentuk paket-paket :

Contoh Diagram Package.

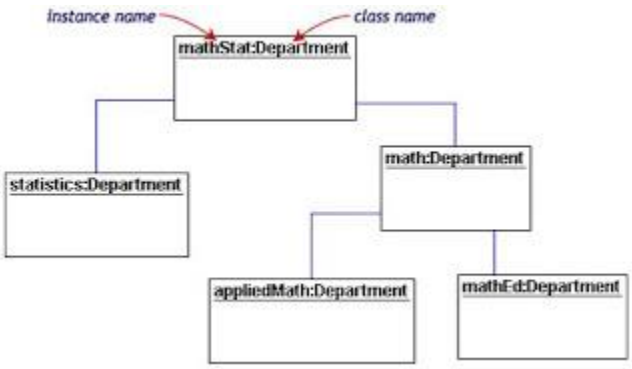



Gambar Package Diagram

Ada jenis khusus dari diagram Class yaitu diagram Object. Kegunaannya untuk penjelasan yang sedikit dengan relasi yang sulit, khususnya relasi rekursif. Lihat gambar dibawah, diagram Class kecil menunjukkan bahwa ‘department’ dapat mengandung banyak ‘department’ yang lain.
Class yang relasinya rekursif.

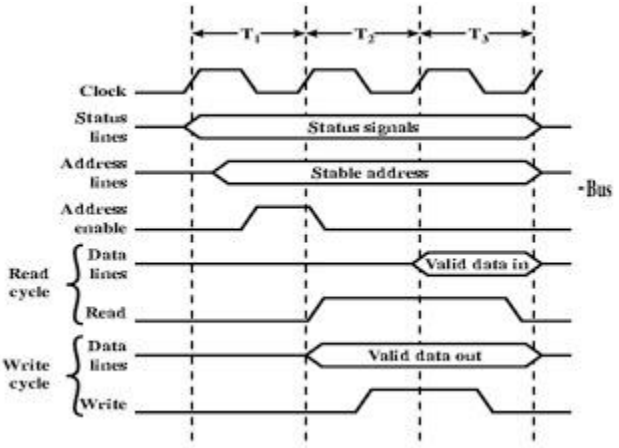



Setiap tingkatan pada diagram berpengaruh pada single instance (bagian tunggal). Nama bagian digarisbawahi dalam diagram UML. Untuk Class name (nama kelas) maupuninstance name (nama bagian) bisa mengambil dari diagram Object selama arti diagram tersebut masih jelas. Instance name memiliki huruf yang digarisbawahi.




**13. Timing Diagram**

Timing Diagram adalah bentuk lain dari interaction diagram, dimana fokus utamanya lebih ke waktu. Timing diagram sangat berdaya guna dalam menunjukkan faktor pembatas waktu diantara perubahan state pada objek yang berbeda.




Gambar Timing Diagram