BAB 2

TEORI PENUNJANG

2.1 Antrian

Menurut Siagian (1987), antrian adalah suatu garis tunggu dari nasabah (satuan) yang memerlukan dari satu atau lebih pelayanan (fasilitas layanan). Pada umumnya, sistem antrian dapat diklarifikasikan menjadi sistem yang berbeda-beda dimana teori antrian dan simulasi sering diterapkan secara luas. Klarifikasi menurut Hil Ier dan Liberman adalah sebagai berikut[2] :

1. Sistem pelayanan komersial
2. Sistem pelayanan bisnis-industri
3. Sistem pelayanan transportasi
4. Sistem pelayanan social

2.1.1 Komponen Dasar Antrian

1. Kedatangan

   Setiap masalah antrian melibatkan kedatangan, misalnya orang, mobil, panggilan telepon untuk dilayani, dan lain – lain. Unsur ini sering dinamakan proses input. Proses input meliputi sumber kedatangan atau biasa dinamakan calling population, dan cara terjadinya kedatangan yang umumnya merupakan variabel acak. Menurut Levin, dkk (2002), variabel acak adalah suatu variabel yang nilainya bisa berapa saja sebagai hasil dai percobaan acak. Variabel acak dapat berupa diskrit atau kontinu. Bila variabel acak hanya dimungkinkan memiliki beberapa nilai saja, maka ia merupakan variabel acak diskrit. Sebaliknya bila nilainya dimungkinkan bervariasi pada rentang tertentu, ia dikenal sebagai variabel acak kontinu.

2. Pelayanan

   Pelayan atau mekanisme pelayanan dapat terdiri dari satu atau lebih pelayan, atau satu atau lebih fasilitas pelayanan. Tiap – tiap fasilitas pelayanan kadang-kadang disebut sebagai saluran (channel) (Schroeder, 1997). Contohnya, jalan tol dapat memiliki beberapa pintu tol. Mekanisme pelayanan dapat hanya terdiri dari satu pelayan dalam satu fasilitas

pelayanan yang ditemui pada loket seperti pada penjualan tiket di gedung bioskop atau penjualan tiker kereta

3. Antri

   Timbulnya antrian tergantung dari sifat kedatangan dan proses pelayanan. Jika tak ada antrian berarti terdapat pelayan yang menganggur atau kelebihan fasilitas pelayanan (Mulyono, 1991). Penentu antrian lain yang penting adalah disiplin antri. Disiplin antri adalah aturan keputusan yang menjelaskan cara melayani pengantri. Menurut Siagian (1987), ada 5 bentuk disiplin pelayanan yang biasa digunakan, yaitu :

   a. FirstCome FirstServed (FCFS) atau FirstIn FirstOut (FIFO) artinya, lebih dulu datang (sampai), lebih dulu dilayani (keluar). Misalnya, antrian pada loket pembelian tiket bioskop.
   b. LastCome FirstServed (LCFS) atau LastIn FirstOut (LIFO) artinya, yang tiba terakhir yang lebih dulu keluar. Misalnya, sistem antrian dalam elevator untuk lantai yang sama.
   c. Service In Random Order (SIRO) artinya, panggilan didasarkan pada peluang secara acak, tidak soal siapa yang lebih dulu tiba.
   d. Priority Service (PS) artinya, prioritas pelayanan diberikan kepada pelanggan yang mempunyai prioritas lebih tinggi dibandingkan dengan pelanggan yang mempunyai prioritas lebih rendah, meskipun yang terakhir ini kemungkinan sudah lebih dahulu tiba dalam garis tunggu. Kejadian seperti ini kemungkinan disebabkan oleh beberapa hal, misalnya seseorang yang dalam keadaan penyakit lebih berat dibanding dengan orang lain dalam suatu tempat praktek dokter.

2.1.2 Model Antrian

2.1.2.1 Single Channel (M/M/1)

Model antrian *single channel* memiliki beberapa komponen, antara lain:

1. Populasi input tak terbatas yaitu jumlah kedatangan pelanggan potensial tak terbatas.
2. Distribusi kedatangan pelanggan potensial mengikuti distribusi poisson. Rata-rata kedatangan persatuan waktu adalah variable random suatu distribusi probabilitas poisson. Dalam notasi (M/M/1), tanda M pertaman menunjukan rata-rata kedatangan yang mengikuti distribusi probabilitas poisson. Sedangkan arti M kedua adalah tingkat pelayanan yang mengikuti distribusi probabilitas poisson. Angka satu menunjukan jumlah fasilitas pelayanan dalam sistem atau satu saluran (*one channel).*
3. Disiplin pelayanan mengikuti pedoman FCFS.
4. Fasilitas pelayanan terdiri dari saluran tunggal.
5. Distribusi pelayanan mengikuti distribusi posisson. Diasumsikan bahwa lamda lebih kecil dari miu ($\lambda < \mu$) yaitu rata-rata jumlah kedatangan pelanggan persatuan waktu lebih kecil dari rata-rata jumlah pelanggan yang dapat dilayani persatuan waktu dalam sistem.
6. Kapasitas sistem diasumsikan tak terbatas.
7. Tidak ada penolakan maupun pengingkaran.

Persamaan yang digunakan dalam sistem (M/M/1) adalah sebagai berikut:

$$p = \frac{\lambda}{\mu} \quad (1)$$

$$p_n = p^n (1-p) \quad (2)$$

$$L = \frac{p}{1-p} = \frac{\lambda}{\mu-\lambda} \quad (3)$$

$$Lq = \frac{\lambda^2}{\mu(\mu-\lambda)} \quad (4)$$

$$W = \frac{1}{\mu-\lambda} \quad (5)$$

$$Wq = \frac{\lambda}{\mu(\mu-\lambda)} \quad (6)$$

Notasi dalam sistem antrian adalah sebagai berikut:

n = Jumlah pelanggan dalam sistem

P_n = Probabilitas kepastian n pelanggan dalam sistem

λ = Jumlah rata-rata pelanggan yang datang per satuan waktu

μ = Jumlah rata-rata pelanggan yang dilayani per satuan waktu

Po = Probabilitas tidak ada pelanggan dalam sistem

P = Tingkat itensitas fasilitas pelanggan

L = Jumlah rata-rata pelangan yang diharapkan dalam sistem

Lq = Jumlah rata-rata pelanggan yang menunggu dalam antrian

W = Waktu yang diharapkan pelanggan selama dalam sistem

Wq = Waktu yang diharapkan pelanggan selama menunggu antrian

1/ μ = Watu rata-rata pelayanan

1/ λ = Waktu rata-rata antar kedatangan

S = Jumlah fasilitas pelayanan

2.1.2.2 Mltiple Channel (M/M/s)

Perbedaan dengan single channel model terletak pada jumlah pelayanan. Fasilitas pelayanan yang dimiliki model (M/M/s) lebih dari satu. Huruf (s) yang terdapat pada (M/M/s) menyatakan jumlah fasilitas pelayanan. Notasi yang digunanak multiple channel model, pada dasarnya sama dengan single channel model, kecuali untuk dua persamaan berikut:

$$P = \frac{\lambda}{S.\mu} \tag{7}$$

$$Po = \left[\sum_{n=0}^{s-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \frac{1}{1-(\frac{\lambda}{S\mu})}\right]^{-1} \tag{8}$$

$$Po = \begin{cases} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} \ (P_0), \ jika \ 0 \le n \le s \\ \frac{\left(\frac{\lambda}{\mu}\right)^n}{S!S^{n-s}} \ (P_0), \ jika \ n \ge s \end{cases} \tag{9}$$

$$Lq = \frac{P_0\left(\frac{\lambda}{\mu}\right)^2 P}{S!(1-P)^2} \tag{10}$$

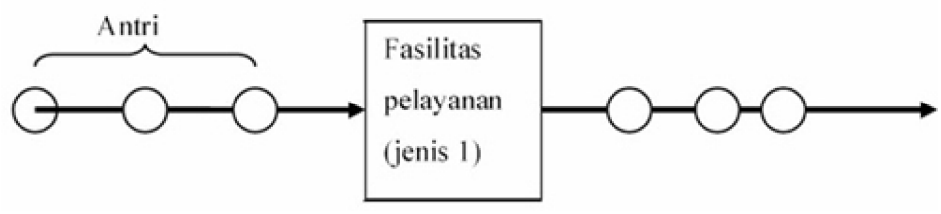
$$Wq = \frac{Lq}{\lambda} \tag{11}$$

$$W = Wq + \frac{1}{\mu} \tag{12}$$

$$L = \lambda W = Lq + \frac{\lambda}{\mu} \tag{13}$$

2.1.3 Struktur Antrian

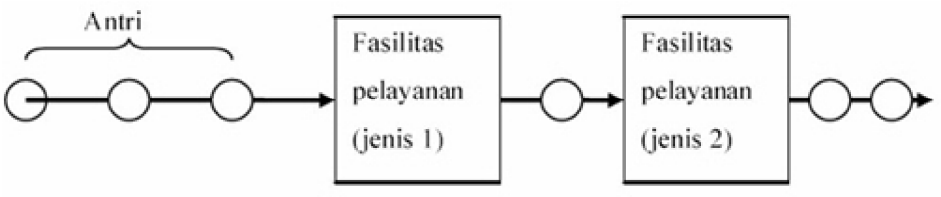
1. Singel Channel – Single Phase



Gambar 2.1 Single Channel-Single Phase

Single Channel berarti hanya ada satu jalur yang memasuki system pelayanan atau ada satu fasilitas pelayanan. Single Phase berarti hanya ada satu pelayanan.

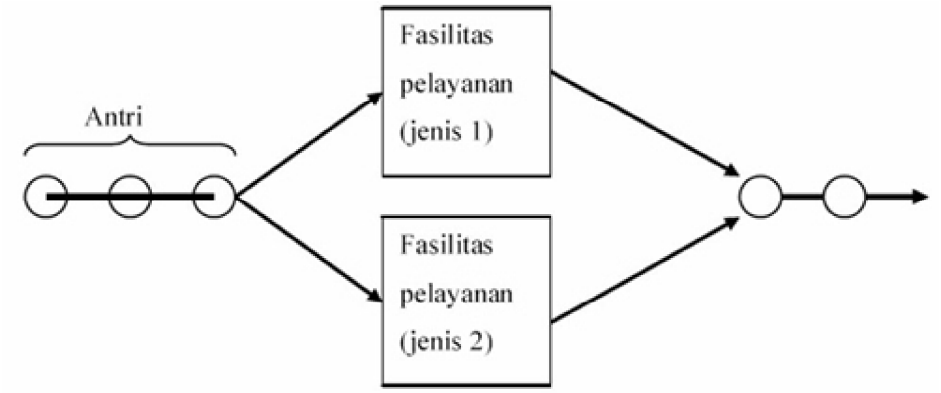
2. Single Channel – Multi Phase



Gambar 2.2 Single Channel-Multi Phase

Istilah Multi Phase menunjukkan ada dua atau lebih pelayanan yang dilaksanakan secara berurutan (dalam phasephase). Sebagai contoh : pencucian mobil.

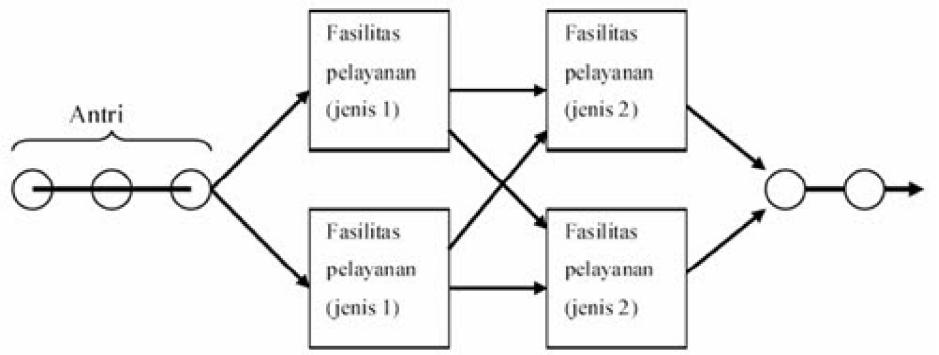
3. Multi Channel – Single Phase



Gambar 2.3Multi Channel-Single Phase

Sistem Multi Channel – Single Phase terjadi kapan saja di mana ada dua atau lebih fasilitas pelayanan dialiri oleh antrian tunggal, sebagai contoh model ini adalah antrian pada teller sebuah bank.

4. Multi Channel – Multi Phase



Gambar 2.4 Multi Channel-Multi Phase

Salah satu Model antrian yang akan digunakan adalah model *multi chanel single phase,* yaitu antrian dengan jalur banyak dengan fasilitas pelayanan lebih dari dua. Untuk contohnya model antrian ini banyak digunakan pada proses pelayanan dirumah sakit, karena rumah sakit memiliki beberapa tahap dalam pelayanannya, serpeti regristrasi, pesan antrian, pengobantan/pemeriksaan dan pembayaran.

Model antrian ini bisa terjadi kapan saja saat suatu fasilitas pelayanan dialiri oleh antrian tunggal untuk dapaat melayani suatu populasi yang membutuhkan fasilitas pelayanan yang berbeda.

2.2 Gammu SMS

Aplikasi gammu adalah aplikasi yang dapat digunakan untuk mengelola fungsi-fungsi pada mobile phone, modem dan perangkat serupa lain nya. Beberapa yang dapat dikelola oleh Gammu adalah fungsi kontak (Phonebook) dan fungsi SMS.

Gammu merupakan salah satu library opensource yang dibuat sebagai gateway atau penghubung antara handphone dengan perangkat komputer. Pengembangan Gammu awalnya dari gnooki yang dari menggunakan konsep masih sangat sederhana dan rumit.

Gammu dibuat untuk menangani hal tersbut dengan instalasi dan pembangunan yang lebih mudah. Dibuatna gammu ini tak luput dari keterlibatan beberapa developer. Gammu dibuat menggunakan gabungan phyton dan C. Terdapat pula

versi lain yang telah dikemas menjadi aplikasi jadi bernama wammu yang dibangun menggunakan bahasa C++. Gammu juga dapat di gunakan diberbagai Operating System. seperti linux, unix, windows dan yang membedakan hanya proses instalasinya saja[7].

2.3 Website

Website adalah sebuah halaman web yang memiliki bagian saling terkait, terkadang juga sering disertai dengan berkas-berkas, gambar, video, atau jenis-jenis berkas lainnya. Dengan kata lain, website adalah sebuah cara untuk menampilkan diri di Internet. Dapat diibaratkan website adalah sebuah tempat di Internet, siapa saja di dunia ini dapat mengunjunginya, kapan saja mereka dapat mengetahui tentang informasi yang disajikan, memberikan pertanyaan, memberikan masukan atau bahkan mengetahui dan membeli produk. Sebuah halaman web merupakan berkas yang ditulis sebagai berkas teks biasa (plain text) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML, atau XHTML, kadang-kadang pula disisipi dengan sekelumit bahasa skrip. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer.

Penemu situs web adalah Sir Timothy John “Tim” Berners-Lee, sedangkan situs web yang tersambung dengan jaringan pertamakali muncul pada tahun 1991. Maksud dari “Tim” ketika merancang situs web adalah untuk memudahkan tukar menukar dan memperbarui informasi pada sesama peneliti di tempat ia bekerja[4].

Secara garis besar website terbagi menjadi tiga, yaitu:

1. Website Statis
2. Website Dinamis
3. Website Interaktif

2.3.1 Website Statis

Situs web statis merupakan situs web yang memiliki isi tidak dimaksudkan untuk diperbarui secara berkala sehingga pengaturan ataupun pemutakhiran isi atas situs web tersebut dilakukan secara manual.

2.3.2 Website Dinamis

Situs web dinamis merupakan situs web yang secara spesifik didisain agar isi yang terdapat dalam situs tersebut dapat diperbarui secara berkala dengan mudah. Sesuai dengan namanya, isi yang terkadung dalam situs web ini umumnya akan berubah setelah melewati satu periode tertentu. Situs berita adalah salah satu contoh jenis situs yang umumnya mengimplementasikan situs web dinamis.

2.3.3 Website Interaktif

Website interaltif adalah website yang berifat interaktif dan lebih dikenal dengan istilah website generasi 2.0 (Web 2.0). dimana masing-masing user dapat memberikan pertanyaan, jawaban, argument, ide dan lain-lain.

Dalam website interaktif terjadi komunikasi dua arah antara pengunjung dan pengurus website atau antara pengunjung dengan sesame pengunjung. Contohnya facebook dan tewitter.

2.4 UML



Gambar 2.5 Logo UML

Unified Modeling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industru untuk visualisasi, merangcang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan Menggunakan UML dapat membatu untuk membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation*

dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, atau VB.NET. walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi procedural dalam VB dan C[6].

2.4.1 Sejarh UML

Sejarah UML cukup panjang. Sampai era tahun 199 seperti kita ketahui puluhan meodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah metodologi booth, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirst-brock, dan sebagainya. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi mebawa notasi sendiri-sendiri, yang mengakibatkan tumbul maslah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimuali pada bulan oktober 1994, Booth, Rmbaugh dan Jacobson, yang merrupakan tiga tokoh yang boleh dikatakan metodologinya banyak digunakan mempolopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft oertana dari UML (bersi 0.8). sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG-http://www.omg.org). ahun 1997 UML versi 1.1 muncul, dan saat ini versi erbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buka serial tentang UML pada tahun 1999. Sejak saat itulah UML menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

2.4.2 Konsep Dasar UML

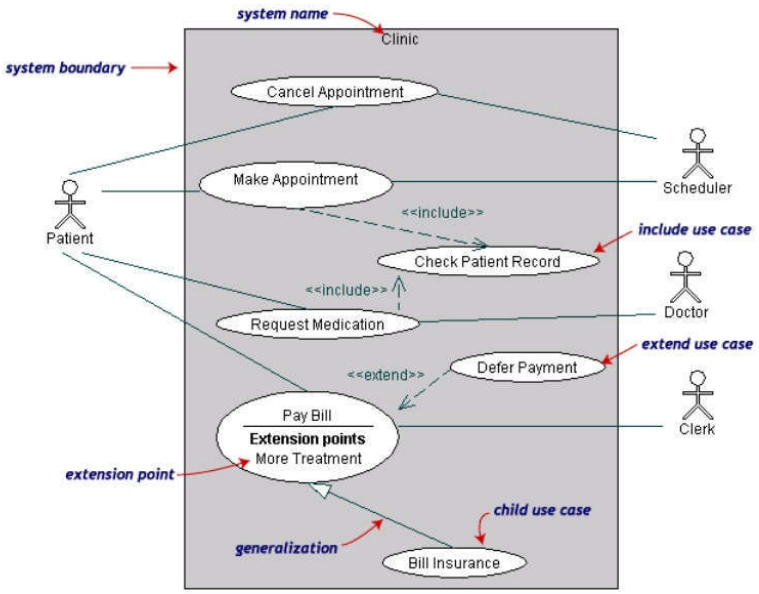
2.4.2.3 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dana bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara actor dengan sistem. *Use case* merrupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanjam dan sebagainya. Seorang/sebuah actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu

bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klienm dan merancang *test case* untuk semua fitur yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsional *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case yang* di-*include* akan dipanggil seriap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari denga cara menarik keluar fungsionalitas yang *common.*

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behavior*-nya sendiri. Semtara hubungan generalitas antar *use case* menunjukan bahaw *use case* yang satu meruoakan spesialitas dari yang lain.



Gambar 2.6 Diagram Use Case

2.4.2.4 Class Diagram

Class adalah sebuah sepesifikasi yang jika diinstansikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem. Sekaligus menawarkan layanan untuk memanipuasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktru dan deskripsi *class, package,* dan objek berserta hubungan satu sama lain seperti *containment,* pewarisanm asosiasi, dan lain-lain.

class memiliki tiga area pokok :

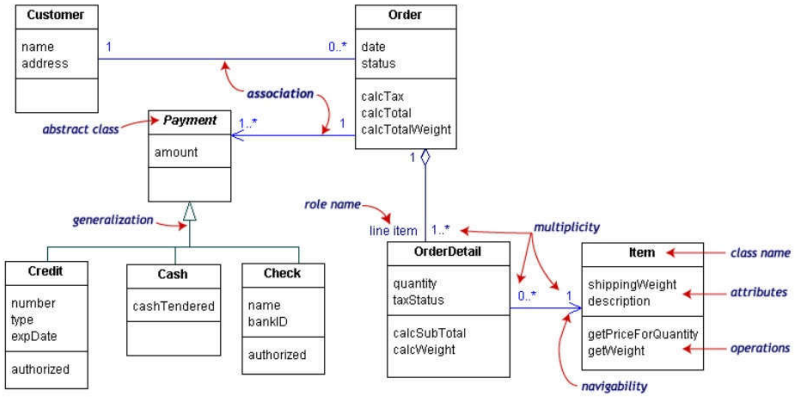
5. Nama (dan stereotype)
6. Atribut
7. Metoda

Atribut dan metoda dapat memiliki satu sifat berikut L

- *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 diagram memiliki hububungan antara lain:

4. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan classyang memiliki atribut berupa classlain, atau classyang harus mengetahui eksistensi classlain. Panah navigabilitymenunjukkan arah queryantar class.
5. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
6. Pewarisan, yaitu hubungan hirarkis antar class. Classdapat diturunkan dari classlain dan mewarisi semua atribut dan metoda classasalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari classyang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
7. Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passingdari satu classkepada classlain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

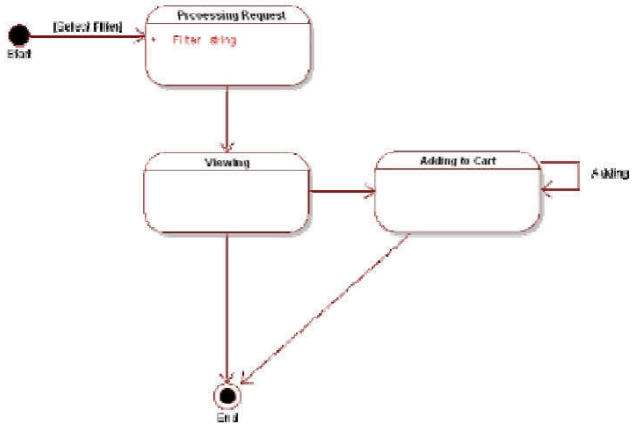


Gambar 2.7 Class Diagram

2.4.2.5 Statecart Diagram

Statecart diagram menggambarkan tansisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) subtu 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* diangam berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisiya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Berikut adalah contoh *statechart diagram*:



Gambar 2.8 Statechart Diagram

2.4.2.6 Activity Diagram

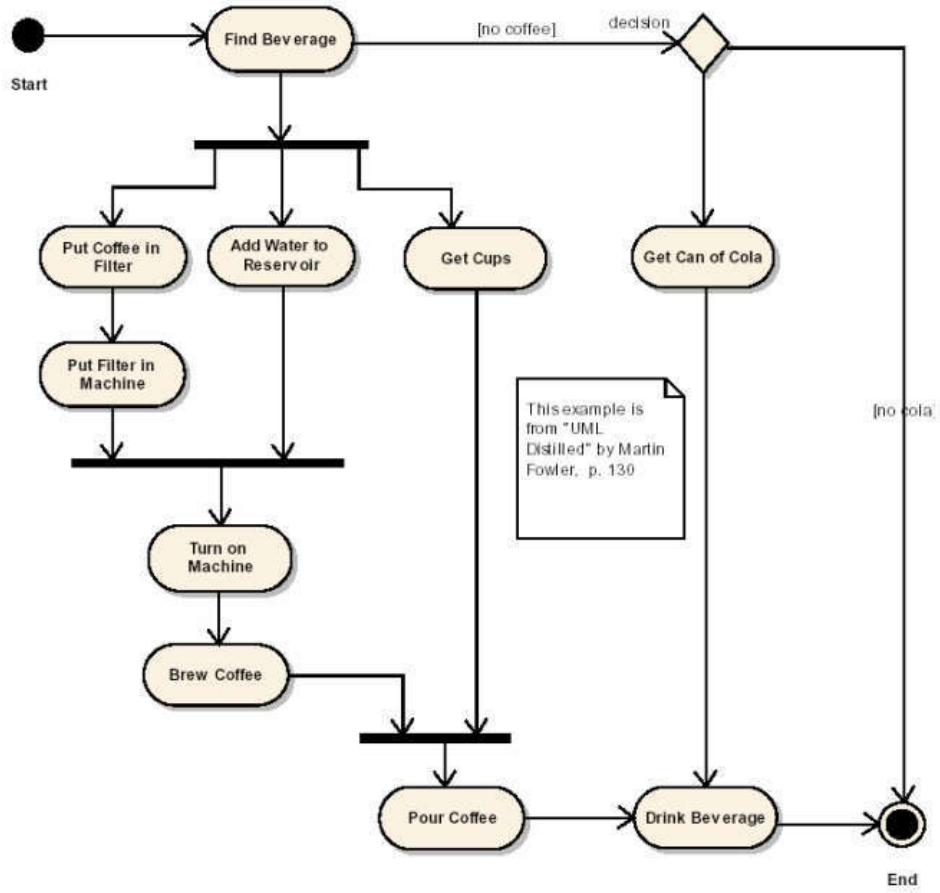
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *dicision* yang mungkin terjadi, dan bagaimana meraka berakhir. *Activity diagram* juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi dipicu oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior* internal sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

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 diagramdapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Berikut contoh *activit diagram* tanpa *swimlane*:



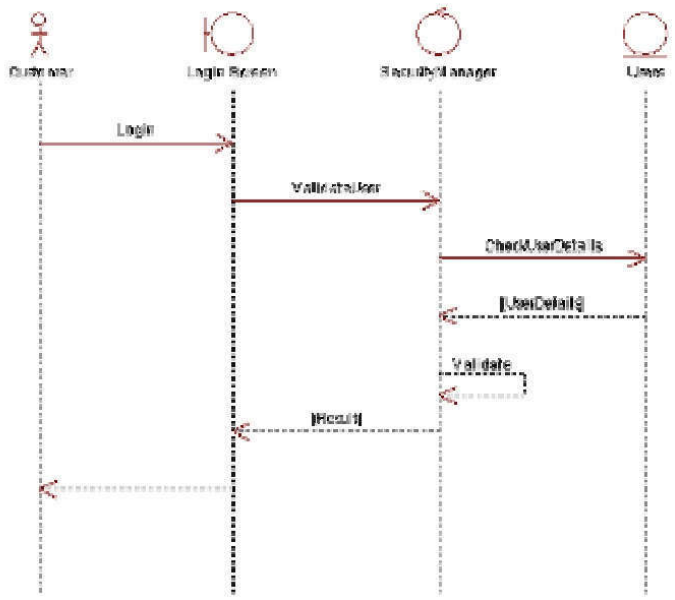
Gambar 2.9 Actuvity Diagram

2.4.2.7 Sequience Diagram

Squence diagram menggambarkan interaksi antar objek di dalam dan disekitas sistem (termasuk pengguna, *display,* dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram bisa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

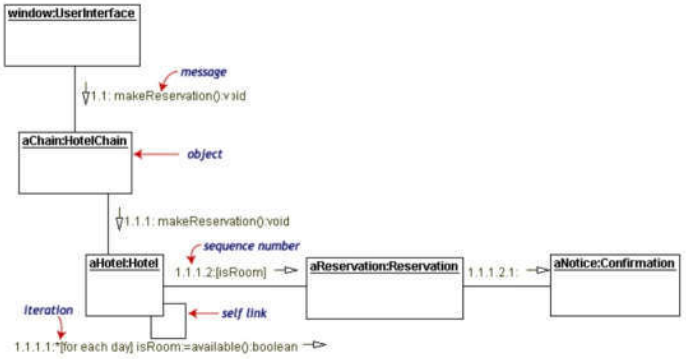
Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari class. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*. Berikut adalah contoh *sequence diagram*:



Gambar 2.10 Squence Digram

2.4.2.8 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message.* Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.



Gambar 2.11 Colaboration Diagam

2.5 PHP

PHP atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side". PHP termasuk dalam open source product, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis melalui situs resmi PHP : http://www.php.net.

PHP juga dapat berjalan pada berbagai web server seperti IIS (Internet Information Server), PWS (Personal Web Server), Apache, Xitami. PHP juga mampu berjalan di banyak sistem operasi yang beredar saat ini, diantaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac Os, Solaris. PHP dapat dibangun sebagai modul web server Apache dan sebagai binary yang dapat berjalan sebagai CGI (Common Gateway Interface). PHP dapat mengirim HTTP header, dapat mengatur cookies , mengatur authentication dan redirect user.

Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data atau Database Management Sistem (DBMS), sehingga dapat menciptakan suatu halaman web dinamis. PHP mempunyai koneksitas yang baik dengan beberapa DBMS seperti Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm, dan tidak terkecuali semua database ber-interface ODBC[3].

2.6 Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, controller, dan user interface[8].

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view.

Beberapa fitur yang terdapat di Laravel :

1. Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
2. Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang menagatasi masalah pada hubungan objek database.
3. Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route.
4. Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route.
5. Restful controllers, memisahkan logika dalam melayani HTTP GET and POST.
6. Class Auto Loading, menyediakan loading otomatis untuk class PHP.

7. View Composer, adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading.
8. IoC Container, memungkin obyek baru dihasilkan dengan pembalikan controller.
9. Migration, menyediakan sistem kontrol untuk skema database.
10. Unit Testing, banyak tes untuk mendeteksi dan mencegah regresi.
11. Automatic Pagination, menyederhanakan tugas dari penerapan halaman.

2.7 Vue.Js

Vue js merupakan salah satu framework yang banyak digunakan. Vue.js mempunyai kemiripan di bagian sintak dengan angular.js, hanya saja angular js memiliki development stack yang lebih rumit dibandingkan dengan vue js. Vue js juga menggunakan fitur data binding sama seperti angular js. Jika di angular js kita menggunakan fungsi controller, maka di vue js kita akan menggunakan pendekatan dom yaitu kita akan menggunakan id, name atau class dari suatu component html. Vue js juga memiliki component routing yaitu vue-router [9].