BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Objek yang dijadikan penelitian didalam penulisan skripsi ini adalah Toko retail sepatu Original Skate Shoes. Berikut adalah sejarah singkat perusahaan, visi dan misi, struktur organisasi dan deskripsi tugas yang penulis peroleh langsung dari tempat yang penulis teliti.

3.1.1 Sejarah singkat perusahaan

Original Skate shoes adalah perusahaan retail sepatu skate import yang berdiri tanggal 1 Desember 2009 oleh Evi Hasan. Perusahaan ini bertempat di Jl.Cikajang No.65 bandung. Berdirinya original skate shoes berawal dari sebuah organisasi pecinta skateboard yang kemudian mereka mengembangkan kegiatan organisasinya dengan mendirikan toko sepatu skate.

Dengan bermodal awal sebesar Rp.3.500.000.- yang berasal dari uang pribadi pemiliknya, original skate shoes kemudian membeli beberapa model sepatu langsung dari luar dan menjualnya ke orang-orang yang dikenal dan sebagian lagi dipasarkan di forum jual beli online, pada jaman dimana semakin maraknya produk-produk yang palsu khususnya untuk produk sepatu, Original skate shoes bertekad untuk memberikan pemahaman tentang pentingnya kita lebih menghargai sepatu yang asli dan memerangi produk-produk palsu. Selain itu

Original skate shoes ingin menjadi salah satu retail sepatu original terbesar di indonesia

3.1.2 Visi dan Misi perusahaan

Mengingat perlunya suatu badan usaha atau perusahaan untuk memiliki visi dan misi agar semakin terarahnya tujuan dan target yang pasti dari perusahaan tersebut. Maka berikut visi dan misi pada Original Skate Shoes.

Visi Original Skate Shoes

"Memberikan kualitas dan pelayanan yang terbaik untuk setiap customer**"**

Misi Original Skate Shoes

"Menjadi salah satu perusahaan retail sepatu original terbaik di indonesia**"**

3.1.3 Struktur Organisasi

Dalam melakukan kegiatan operasional kerja dalam suatu perusahaan harus memiliki struktur organisasi yang jelas, dan pada Original skate shoes pun terdapat berbagai fungsi kerja yang jelas dan terstruktur menurut fungsinya masing masing. Berikut adalah struktur organisasi pada Original skate shoes yang berlaku sejak januari 2011.



Gambar 3.1 Struktur Organisasi

2.1.4 Deskripsi Perkerjaan

Setiap posisi dalam suatu organisasi baik perusahaan maupun bukan perusahaan memiliki *job description* yang jelas, sama halnya pada Original skate shoes yang memiliki beberapa fungsi yang jelas dari setiap posisi kerjanya tidak terkecuali posisi sebagai pemilikpun memiliki peranan dan fungsi dari perusahaan ini.

Berikut adalah fungsi, tanggung jawab dan tugas pokok dari struktur organisasi pada Original skate shoes :

1. Pemilik

   Pemilik adalah orang yang berhak atau memiliki kewenangan dalam setiap aktifitas perusahaan ini, selain itu pemilik disini berperan sebagai orang yang mengurusi proses pemasaran dan orang yang memberi tugas-tugas terhadap bawahannya.

2. Bag. Pembelian

   Adalah orang yang mengurusi pemesanan barang dari distributor dan orang yang bertanggung jawab atas proses pemesanan barang dari mulai proses pemesanan barang, mengurusi pembayaran pada distributor, mengurusi bea cukai barang sampai barang sampai di tempat atau di gudang original skate shoes.

3. Bag. Penjualan

   Adalah orang yang mengurusi penjualan barang dari mulai promosi, penjualan langsung pada pembeli, mempaketkan barang jika ada pembeli

dari luar kota dan orang yang melakukan transaksi penjualan langsung dengan pembeli.

4. Bendahara

   Adalah orang yang bertanggung jawab dengan pengelolaan keuangan perusahaan, mengurusi pendapatan dan pengeluaran perusahaan serta orang yang membuat laporan keuangan perusahaan.

3.2 Metode Penelitian

Metode merupakan suatu kerangka kerja untuk melakukan suatu tindakan yang terarah, beraturan dan terkonteks untuk melakukan suatu tindakan pada sebuah penelitian.

Penelitian adalah suatu kegiatan dalam mengkaji secara teliti dan teratur menurut kaidah tertentu untuk menghasilkan fakta-fakta atau prinsip-prinsip baik kegiatan untuk penemuan, pengujian atau pengembangan. dan kaidah yang dianut adalah kaidah metode tersebut.

3.2.1 Desain Penelitian

Pendekatan kasus pada original skate shoes penulis menggunakan metode deskriptif, yaitu suatu metode dengan tujuan untuk membuat gambaran secara sistematis, faktual dan akurat mengenai fakta-fakta dan sifat-sifat pada suatu objek penelitian tertentu. tahap pertama yang dilakukan penulis adalah dengan cara mengumpulkan data dan bahan yang diperlukan pada objek penelitian yang bersangkutan, setelah itu tahap berikutnya penulis mengolah, membahas dan menjelaskannya sampai pada suatu kesimpulan dan pada akhirnya dapat dibuat

suatu laporan untuk melampirkan semua kegiatan yang dikerjakan selama dilakukannya penelitian di Original skate shoes.

3.2.2 Jenis dan Metoda Pengumpulan Data

Sesuai dengan tujuan yang ingin dicapai dalam penelitian ini, penulis mengumpulkan data sebagai bahan penyusunan laporan penelitian ini. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah observasi dan wawancara.

3.2.2.1 Sumber Data Primer

Sumber data primer yaitu sumber data yang langsung didapat dari pihak yang bersangkutan dengan objek penelitian seperti pegawai, pemilik atau orang-orang yang berkaitan dengan objek penelitian tersebut. Dalam memperoleh data primer penulis melakukan wawancara dan observasi di tempat penelitian.

1. Observasi

   Proses untuk mendapatkan data dengan mengamati langsung objek yang akan diteliti dan mengamati kegiatan-kegiatan yang sedang berjalan di tempat penelitian yaitu di Original skate shoes.

2. Wawancara

   Yaitu dengan melakukan proses dialog langsung dengan orang yang dianggap bisa memberikan informasi yang dibutuhkan oleh penulis yaitu dengan pegawai atau pemilik pada Original skate shoes.

3.2.2.2 Sumber Data Skunder

Data sekunder yaitu data – data yang diperoleh secara tidak langsung yang dapat dijadikan data pendukung dari sumber data primer. Data sekunder yang penulis ambil yaitu struktur organisasi, sejarah perusahaan, serta dokumen

dokumen yang ada pada tempat penelitian seperti katalog produk, daftar harga, data transaksi dan juga jadwal event yang diperoleh dari tempat penelitian.

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Metode pendekatan dan pengembangan sistem yaitu menggambarkan tahapan-tahapan dalam proses penelitian guna memecahkan masalah pada penelitian dari awal perencanaan hingga hasil akhir penelitian.

3.2.3.1 Metode Pendekatan Sistem

Metode merupakan suatu kerangka kerja untuk melakukan suatu tindakan yang terarah, beraturan dan terkonteks untuk melakukan suatu tindakan. Metode pendekatan sistem yang penulis gunakan adalah metode pendekatan sistem berorientasi objek (*Object-Oriented*).

Object Oriented Programming atau dapat juga disebut dengan pemrograman berorientasi pada objek merupakan metode pemrograman yang sedang 'booming' saat ini. Metode ini digunakan untuk mencari cara atau bagaimana membuat sebuah program yang lebih mudah dan meminimalisir kesalahan dalam proses compiling -nya. Dalam dunia nyata kita sering melihat berbagai macam objek yang memang dapat dilihat secara kasat mata dan dapat dirasakan karena mempunyai bentuk. Tetapi dalam sebuah proses program, objek terkadang merupakan sekumpulan perintah-perintah yang dijalankan oleh sebuah compiler tertentu sehingga dapat menampilkan hasil yang diinginkan oleh si pembuat program.

Pada bagian awal pembahasan sudah sedikit disinggung mengenai objek itu sendiri. Sebagai contoh dalam objek kehidupan nyata, kita dapat melihat

adanya sebuah objek yang bernama mobil. Sebuah mobil tentunya memiliki beberapa bagian-bagian yang mendukung agar objek mobil tersebut dapat beroperasi sesuai dengan keinginan si pembuatnya. Dalam Objek mobil tersebut mempunyai fungsi masing-masing yang kesemuanya itu harus berjalan dengan "seirama" dan "selaras" agar objek mobil itu dapat bergerak dengan baik dan sesuai dengan perintah seorang penegemudinya.Pemrograman berorientasi objek dapat diimplementasikan terhadap berbagai macam aplikasi bahasa pemrograman. Diantaranya pemrograman berbasis visual sekarang ini sudah "support" terhadap OOP, misalnya Visual Basic 6.0, Visual Foxpro 9.0 dan lain-lain.Pemrograman PHP, Java paling sering digunakan untuk mengaplikasikan pemrograman berbasis Objek.

Mekanisme pemrograman OOP

Class dan objek: merupakan konsep object-oriented yang meng-encapsulasi data dan prosedur secara abstrak yang diperoleh untuk mendeskripsikan content atau state dan behaviour terhadap beberapa entitas didunia nyata. Content atau state dapat artikan keadaan dari sebuah objek,sebagai contoh mobil memiliki state warna, model, bentuk dan lain-lain. Sedangkan behaviour merupakan kelakukan dari objek tersebut, misalnya mobil maju kedepan, belok kiri atau kanan,memberi tanda lampu (sen) dan lainnya. Objek dapat menyimpan state dalam beberapa variabel dan mengimplementasikan behaviour- nya menggunakan method. Berdasarkan penjelasan tersebut apabila diimplementasikan dalam pemrograman, objek dapat diartikan sebagai bagian dari sebuah aplikasi program (software) dimana didalamnya dihubungkannya variabel-

variabel dan juga method-mthodnya yang keduanya saling keterkaitan satu dengan lainnya.

Contoh umum untuk konsep abstrak seperti ini adalah objek Event dalam pemrograman berbasis visual, yaitu objek untuk mewakili peristiwa klik atau tombol ditekan. Kemudian bagaimana dengan pengertian dari class itu sendiri. Class dapat diartikan seperti sebuah cetakan atau template. Apabila kita ingin membuat sebuah kue bolu, maka kita diharuskan mempunyai cetakan kue bolu. Dari cetakan kue bolu yang sudah ada, kita dapat mencetak berbagai macam jenis kue bolu. Berbagai macam jenis kue bolu ini yang dapat dikatakan sebagai objek, yaitu hasil dari class.

Hal ini dapat dilakukan karena semua objek kue bolu yang dibuat pada dasarnya memiliki karakteristik yang sama, sehingga dapat dibuatkan semacam samplenya. Tetapi objek-objek yang dibangun tetap akan memiliki bentuk fisik tertentu sendiri-sendiri, seperti variabel dalam sebuah program. Dengan penjelasan ini, kelas dapat kita definisikan kembali menjadi sebuah blueprint, atau prototipe, yang mendefinisikan variabel dan metode yang sama untuk semua objek sejenis.

Dalam sebuah class terdapat dua komponen penting, diantaranya yaitu :

a. Method yang menggambarkan behaviour atau perilaku dari sebuah objek
b. Properties menggambarkan karakteristik atau sifat dari sebuah objek

Konsep pemrograman OOP

Ada beberapa konsep mengenai pemroraman berorientasi ada Objek, diantaranya :

a. Inheritance atau biasa disebut dengan istilah pewarisan, dalam konsep ini kita dapat membuat suatu class baru yang bentuknya dapat disamakan dengan class yang sebelumnya.

   Sebagai contoh, sebuah kelas MobilMewah akan mewarisi state dan behaviour dari kelas Mobil. Begitu juga dengan kelas mobilbiasa. Kelas MobilMewah dan mobilbiasa disebut subkelas, atau kelas anak, dari kelas Mobil, yang disebut superkelas, atau kelas induk.Seluruh subkelas akan mewarisi (inherits) state dan behaviour dari superkelasnya. Dengan begitu, semua subkelas dari superkelas yang sama akan memiliki state dan behaviour yang sama. Namun, masing-masing subkelas bisa menambah sendiri state atau behaviournya. Misalkan, pada kelas Mobil tidak terdapat variable nontonTV, namun subkelas MobilMewah memiliki variabel tersebut. Contoh lain misalnya kelas Mobil tidak memiliki metode nyalakanTV, namun MobilMewah memiliki metode itu.

   Dalam kasus tertentu subkelas mungkin memiliki implementasi behaviour yang berbeda dengan superkelasnya. Hal seperti ini disebut override. Contohnya subkelas SepedaBalap memiliki implementasi metode ubahGigi yang berbeda dengan implementasi metode tersebut pada superkelas Sepeda.

   Tingkat pewarisan tidak hanya terbatas pada dua tingkatan. Dari contoh di atas, kita bisa saja membuat subkelas dari kelas SepedaBalap, dan seterusnya. Kita bisa terus memperpanjang tingkat pewarisan ini sepanjang yang kita butuhkan. Dengan begitu,subkelas-subkelas yang dibuat akan

lebih khusus dan lebih terspesialisasi. Namun terdapat batasan pewarisan dalam Java yang disebut single inheritance. Artinya sebuah kelas hanya dapat mewarisi sifat dari satu dan hanya satu superkelas saja. Dalam beberapa bahasa pemrograman berorientasi objek lain, yang berlaku adalah multiple inheritance. Artinya sebuah kelas dapat mewarisi sifat dari beberapa superkelas sekaligus. Dalam Java, terdapat kelas Object yang merupakan superkelas dari semua kelas dalam Java, baik yang builtin ataupun yang kita buat sendiri, langsung maupun tidak langsung. Karena itu sebuah variabel bertipe Object akan dapat menyimpan referensi ke objek apapun dalam bahasa Java.

Manfaat penggunaan konsep pewarisan antara lain: pertama, kita dapat menggunakan kembali kelas-kelas yang kita buat (sebagai superkelas) dan membuat kelas-kelas baru berdasar superkelas tersebut dengan karakteristik yang lebih khusus dari behaviour umum yang dimiliki superkelas. Kedua, kita dapat membuat superkelas yang hanya mendefinisikan behaviour namun tidak memberi implementasi dari metode-metode yang ada. Hal ini berguna jika kita ingin membuat semacam template kelas.

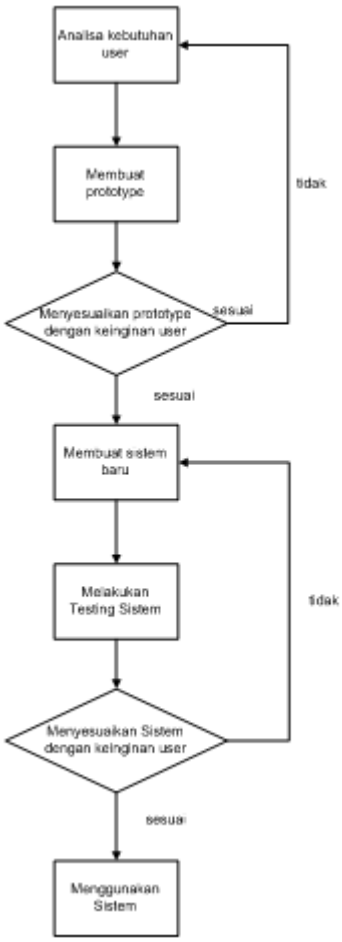
b. Encapsulation biasa disebut dengan istilah membungkus atau mengemas sebuah informasi yang akan menghasilkan sebuah objek. Tujuannya agar informasi yang dihasilkan tidak menyebar keluar dari aturan yang ada.

c. Polymorphism menghasilkan banyak bentuk. Konsep ini mengembangkan konsep pada pewarisan dimana kita dapat membuat dan menghasilkan sebuah class baru yang dapat diperluas kembali menjadi banyak bentuk.

3.2.3.2 Metode Pengembangan Sistem

Desain penelitian ini dimodelkan dengan model proses Prototype, yang merupakan suatu metode dalam pengembangan system yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga dapat segera di evaluasi oleh pemakai (user). Dari pengertian metode prototype tersebut penulis akan memberikan beberapa alasan mengapa menggunakan metode prototype, dikarenakan penulis akan lebih mudah dalam merancang sistem yang diinginkan oleh pemakai sistem (user), penulis bisa lebih memaksimalkan rancangan yang diusulkan dengan kebutuhan dan keinginan user, bisa memperlihatkan dan mempresentasikan bentuk rancangan yang akan dibuat oleh penulis kepada calon pemakai (user) terlebih dahulu sebelum ketahap perampungan pengerjaan sistem sehingga penulis bisa menerima masukan dari pemakai (user).

Prototype yang dibuat oleh Penulis dengan mendifiniskan fungsi dan prosedur sistem dimana pengguna atau pemilik sistem tidak bisa mendefinisikan sistem tersebut. Berikut ini langkah-langkah dari requirement prototype yang penulis pakai :



Gambar 3.2 Pendekatan Prototype

(Raymond McLeod, Jr. and Goorge P. Schell)

[Sumber : http://zulvani.wordpress.com/2010/03/24/rekayasa-perangkat-lunak-bagian-3/]

Analisis kebutuhan user, pengembang dan pengguna atau pemilik sistem melakukan diskusi dimana pengguna atau pemilik sistem menjelaskan kepada penulis tentang kebutuhan sistem yang mereka inginkan.

1. **Membuat prototype,** pengembang membuat prototype dari sistem yang telah dijelaskan oleh pengguna atau pemilik sistem.

2. **Menyesuaikan prototype,** dengan keinginan user, pengembang menanyakan kepada pengguna atau pemilik sistem tentang prototype yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.
3. **Membuat sistem baru,** pengembang menggunakan prototype yang sudah dibuat untuk membuat sistem baru.
4. **Melakukan testing sistem,** pengguna atau pemilik sistem melakukan uji coba terhadap sistem yang dikembangkan
5. **Menyesuaikan dengan keinginan user,** sistem disesuaikan dengan keinginan user dan kebutuhan sistem, jika sudah sesuai sistem siap digunakan.
6. **Menggunakan sistem.**

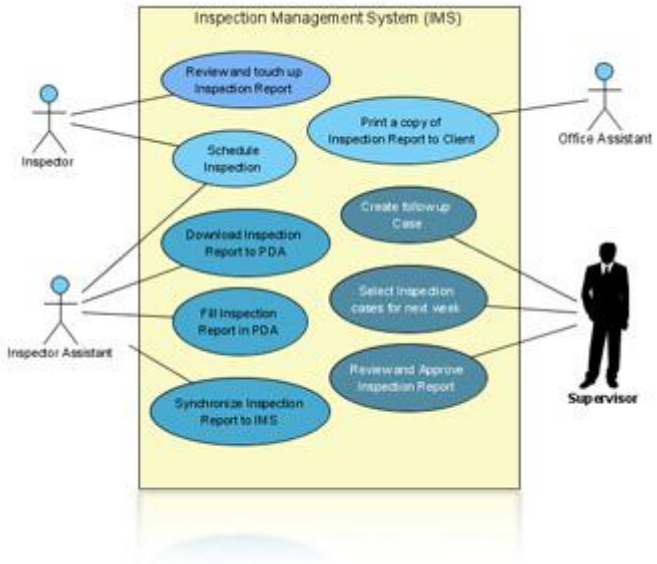
3.2.3.3 Alat Bantu Analisis dan Perancangan

Alat bantu untuk analisis dan perancangan yang penulis pakai adalah dengan menggunakan UML. Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. UML mendefinisikan diagram-diagram sebagai berikut :

1. Use Case Diagram

   Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.



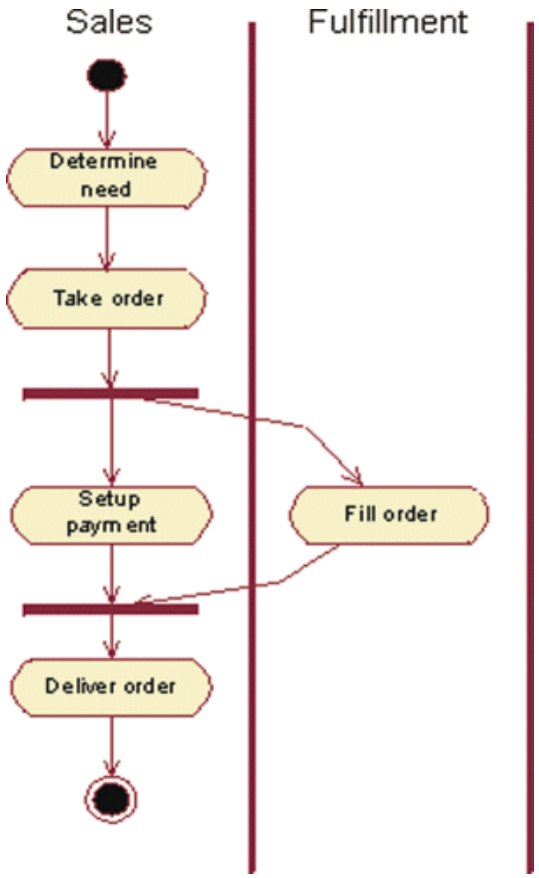
Gambar 3.3 Contoh Use Case Diagram

2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*).

Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih

menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.



Gambar 3.4 Contoh Activity Diagram

3. Class Diagram

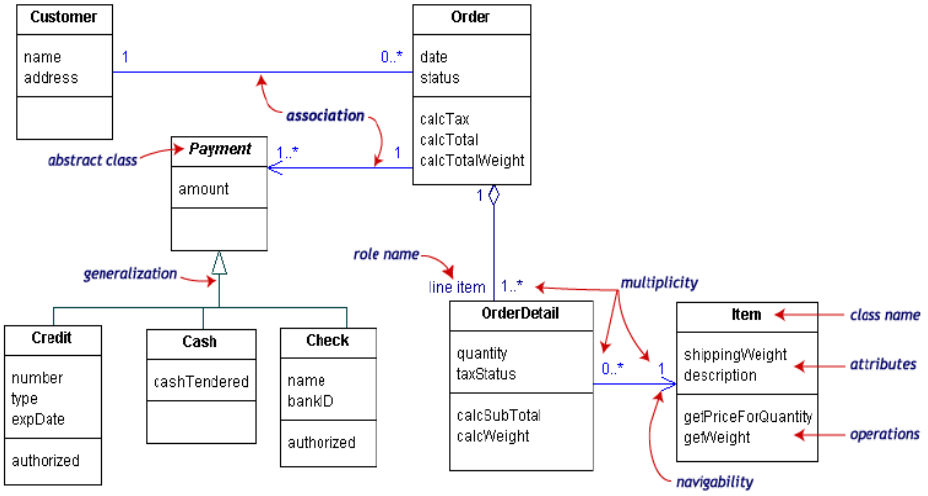
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class memiliki tiga area pokok :

a. Nama dan stereotype

b. Atribut

c. Metoda

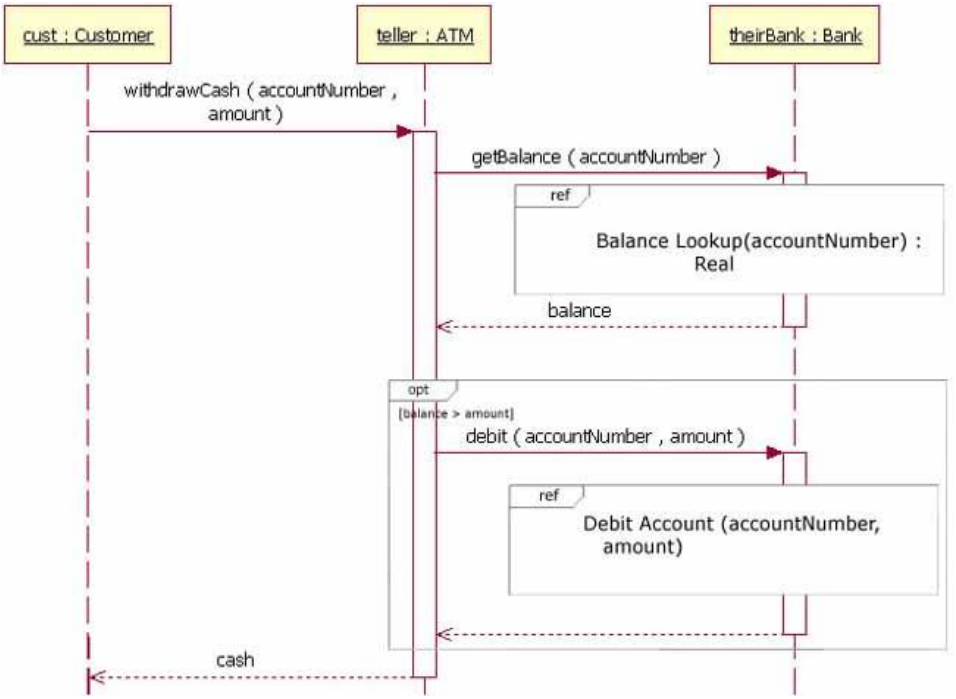


Gambar 3.5 Contoh Class Diagram

4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

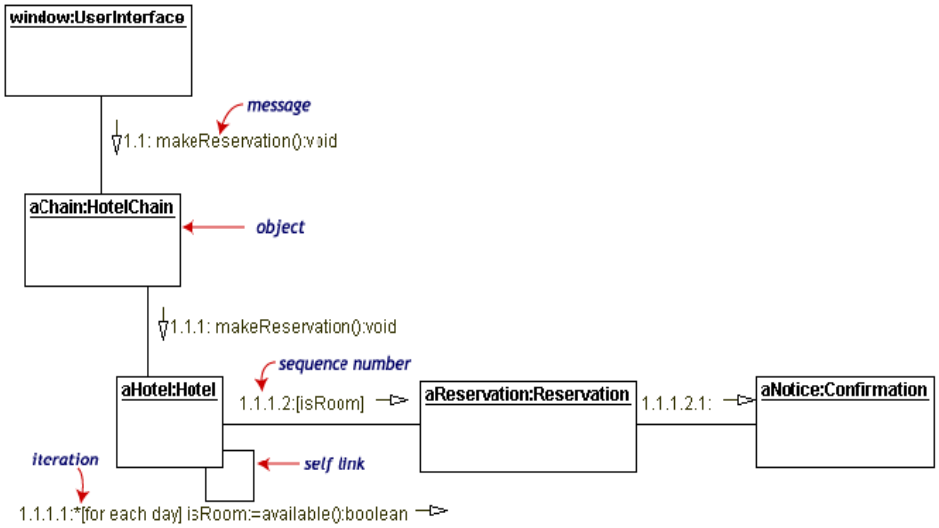
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output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



Gambar 3.6 Contoh Sequence Diagram

5. 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.

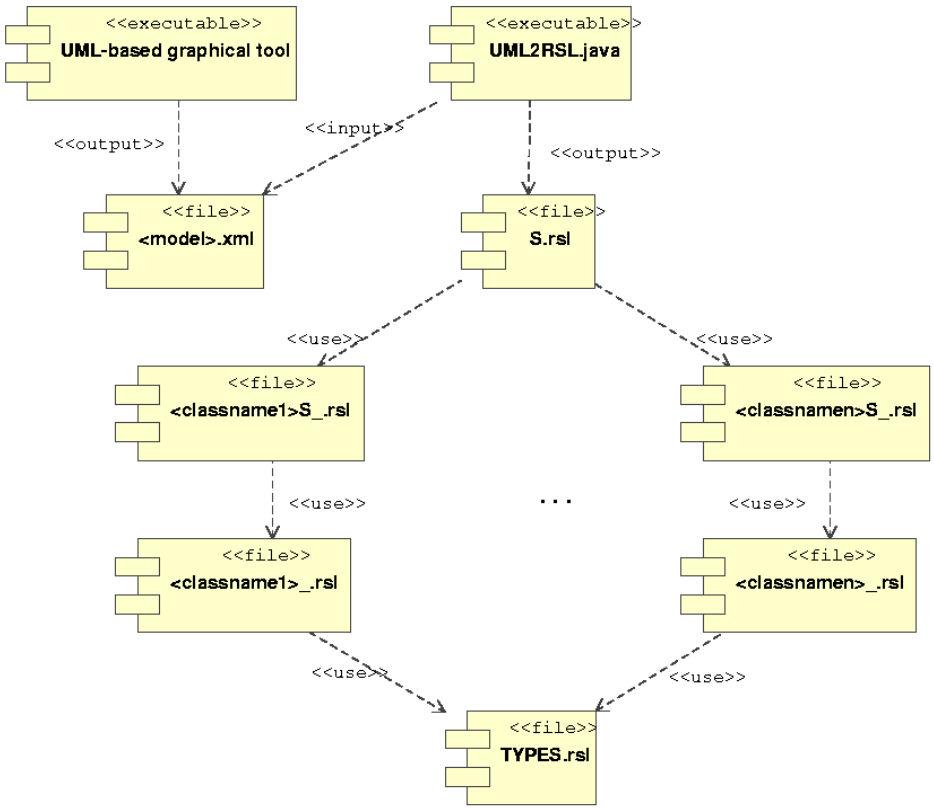


Gambar 3.7 Contoh Collaboration Diagram

6. Componen 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 maupun executable, 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.

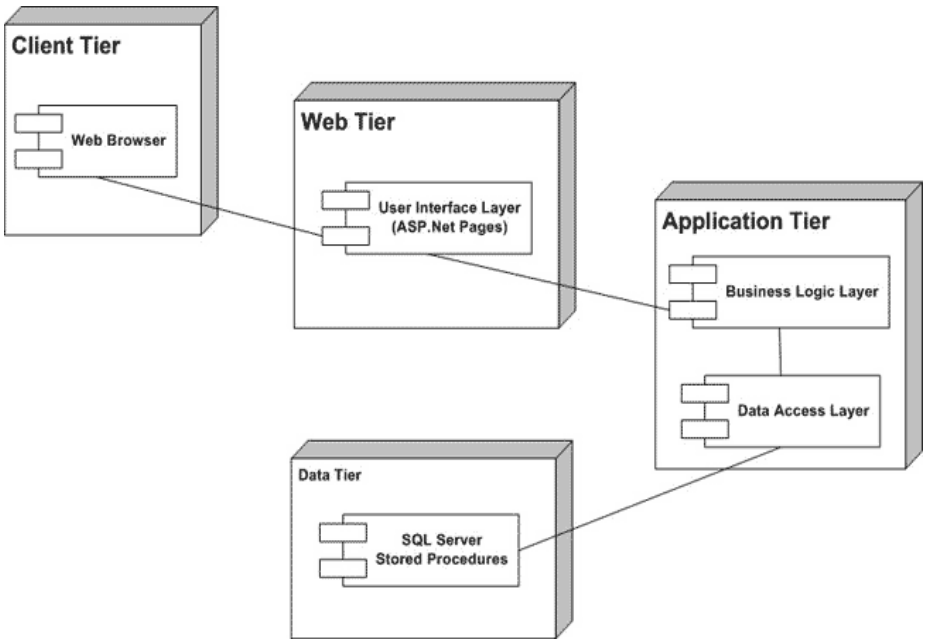
Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.



Gambar 3.8 Contoh Componen Diagram

7. 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 3.9 Contoh Deployment Diagram

3.2.4 Pengujian Software

Pengujian software yang digunakan dalam penelitian ini adalah black box. Pengujian black box adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian black box merupakan metode peracangan data uji yang didasarkan pada pengujian fungsional sistem.

Faktor Pengujian Black Box :

1. Reliability

   Menekankan bahwa aplikasi akan dilaksanakan dalam fungsi sesuai yang diminta dalam periode waktu tertentu. Pembetulan proses tersangkut kemampuan sistem untuk memvalidasi proses secara benar.

2. Access control

   Menekankan sumberdaya sistem harus dilindungi dari kemungkinan modifikasi, pengrusakan, penyalahgunaan dan prosedur keamanan harus dijalankan secara penuh untuk menjamin integritas data dan program aplikasi.

3. Correctness

   Menjamin pada data yang dimasukkan, proses dan output yang dihasilkan dari aplikasi harus akurat dan lengkap. Kelengkapan dan akurasi akan dicapai melalui kontrol transaksi dan elemen data.