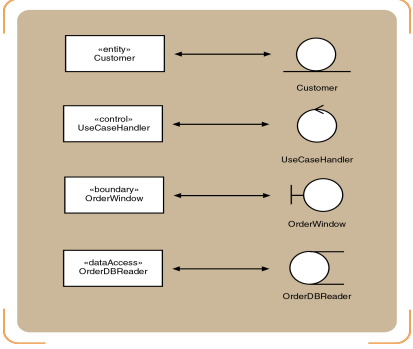


Essay :

- Standard stereotypes found in design models
- Three layer
- Internal symbols used to define a design class

Kasus :

- First cut design class diagram
- Ssd
- Sequence diagram

<p>Design Class Symbol Stereotypes -Notasi UML dikategorikan sebagai model element tipe yang berbeda Two types of notation -Full notation with guillemets(«») -Shorthand notation with circular icons Standard stereotypes -Entity, control, boundary, data access</p> <p>Design Class Notation -Class name & stereotype information -Attribute information visibility, type-expression, name, initial value, & properties -Method signature visibility, name, type-expression, & parameter list use the entire signature</p>	<p>First-Cut Sequence Diagram -Menentukan benda lain mungkin perlu dilibatkan untuk melaksanakan use case -Ganti: Sistem objek dengan objek use case controller -Tentukan pesan lain yang akan dikirim -Tentukan sumber objek dan tujuan untuk setiap pesan</p> <p>Developing a Multilayer Design -view layer Desain antarmuka pengguna untuk setiap use case Mengembangkan dialog desain untuk bentuk Tambahkan window kelas untuk sequence diagram</p> <p>-data access layer Inisialisasi objek domain dengan data dari database Query database dan mengirim objek referensi mengembalikan informasi dalam obyek referensi</p> <p>Ciri-ciri perusahaan yg memakai 2tier/3tier -2 tier untuk perusahaan berskala kecil dan networknya tidak terlalu besar(masih pakai LAN) -3 tier sudah mencakup luas dan bahkan beberapa perusahaan membutuhkan network layer khusus untuk networknya saja (diatas 3 layer belum ada buat network khusus).</p>
<p>Objek yg terdapat di Multi Layer Sequence Diagram</p> <ul style="list-style-type: none"> Entity class: desain identifier untuk kelas domain permasalahan. Dengan kata lain, itu berasal dari model domain. Mereka obyek biasanya pasif, dalam arti bahwa mereka menunggu acara bisnis terjadi sebelum mereka melakukan apa pun, dan mereka juga kelas biasanya gilig. Persistent Class: salah satu yang ada setelah program berhenti. Dengan kata lain, data harus bertahan setelah sistem dimatikan. Jelas, cara untuk melakukannya adalah untuk menulis itu keluar ke sebuah file atau database. Boundary class: kelas yang dirancang khusus untuk hidup pada batas otomatisasi sistem. Dalam sistem desktop, kelas-kelas ini akan menjadi jendela kelas dan semua kelas-kelas lain yang terkait dengan user interface. Control class: kelas yang menjadi perantara antara kelas-kelas batas dan kelas entitas. Dengan kata lain, tanggung jawab adalah untuk menangkap pesan dari objek batas kelas dan mengirim mereka ke kelas obyek entitas yang benar. Ini bertindak sebagai switchboard, atau controller, antara lapisan melihat dan lapisan domain. Data access class: kelas yang digunakan untuk mengambil data dari dan mengirim data ke database. Daripada memasukkan logika akses database, termasuk Structured Query Language (SQL) pernyataan, ke metode entitas kelas, lapisan terpisah dari kelas untuk mengakses database sering dimasukkan dalam desain 	<p>Design Principle</p> <ul style="list-style-type: none"> Encapsulation - Setiap objek adalah sebuah unit mandiri yang mengandung data dan logika program Object reuse - Obyek standar dapat digunakan berulang-ulang dalam sebuah sistem Information hiding - Data yang terkait dengan sebuah objek tidak terlihat - Method yang menyediakan akses ke data Navigation visibility - Menjelaskan objek yang dapat berinteraksi satu sama lain Coupling - Mengukur bagaimana suatu kelas terkait erat Kohesi (Cohesion) - Mengukur konsistensi fungsi dalam kelas Pemisahan tanggung jawab (Separation of responsibilities) - Membagi kelas menjadi beberapa kelas yang sangat kohesif
<p>Layer pada 3tier architecture</p> <p>1) Data Layer : yang mengatur penyimpanan data, biasanya pada satu atau lebih database Bagian dari three-layer architecture yang berinteraksi dengan database</p> <p>2) Business Logic Layer : yang mengimplementasi aturan dan prosedur dari business processing Bagian dari three-layer architecture yang berisi program yang mengimplementasi aturan bisnis dari suatu aplikasi</p> <p>3) View Layer : yang menerima input user dan format dan menampilkan hasil process Bagian dari three-layer architecture yang berisi user interface</p>	<p>Single computer & Multitier computer</p> <ul style="list-style-type: none"> Single Computer : arsitektur yang menggunakan single computer system untuk menjalankan application-related software (seluruh kegiatan komputasi terjadi pada satu komputer). Multitier Computer : arsitektur yang application-related software atau beberapa pengolahan di beberapa sistem komputer (kegiatan computing terjadi di beberapa komputer). <ul style="list-style-type: none"> Clustered Architecture => Hardware(server) yang digunakan pada komputer dengan spesifikasi yang sama Multicomputer Architecture => Hardware(server) yang digunakan pada komputer dengan spesifikasi yang berbeda
<p>Ciri-ciri perusahaan yg menggunakan DA/CA:</p> <p>Centralized Architecture :</p> <ol style="list-style-type: none"> Beberapa input transaksi tidak harus diproses secara real time (contoh : time sheet processed pada service payroll) Orang yang melakukan data-entry dapat terletak dipusat (contoh : order lewat telepon yang dapat melayani customer secara geografis) System menghasilkan output periodik yang besar (contoh : tagihan bulanan kartu kredit yang dikirimkan lewat surat kepada customer) Transaksi volume besar terjadi antara high-speed computers (contoh : business-to-business processing untuk supply chain management) gap <p>Distributed Architecture : Sebuah sistem informasi modern biasanya didistribusikan di banyak sistem komputer dan lokasi geografis. misalnya, data keuangan perusahaan mungkin disimpan pada komputer mainframe terpusat. Server kecil di kantor-kantor regional secara berkala mungkin menghasilkan laporan akuntansi dan lainnya berdasarkan data yang tersimpan pada mainframe. Workstation di banyak lokasi dapat digunakan untuk mengakses dan melihat laporan berkala serta langsung mengupdate database pusat. Pendekatan seperti untuk mendistribusikan perangkat lunak dan data di arsitektur bergantung pada jaringan komunikasi untuk terhubung secara geografis tersebar komponen perangkat keras komputer</p>	 <pre> classDiagram class Customer class UseCaseHandler class OrderWindow class OrderDBReader Customer --> UseCaseHandler UseCaseHandler --> OrderWindow OrderWindow --> OrderDBReader </pre>