

BAB II

TINJAUAN PUSTAKA

Dalam bab ini, penulis akan mencoba menguraikan teori yang relevan, lengkap, dan terkini sejalan dengan permasalahan yang dihadapi, juga diuraikan hubungan antara permasalahan tersebut dengan teknik yang digunakan dalam perancangan program aplikasi ini serta bagaimana cara penulis dalam mengatasi dan menyelesaikannya.

II.1. Pengertian Perancangan

Model perancangan sesungguhnya adalah modal objek yang mendeskripsikan realisasi fisik *use case* dengan cara berfokus pada bagaimana spesifikasi-spesifikasi kebutuhan fungsional dan *non-fungsional*, bersama dengan batasan-batasan lain yang berhubungan dengan lingkungan implemenatasi, memiliki imbas langsung pada pertimbangan-pertimbangan pada aktivitas-aktivitas yang dilakukan pada tahap implementasi. Tambahannya, model perancangan sesungguhnya secara langsung bertindak sebagai abstraksi implementasi sistem/perangkat lunak dan dengan sendirinya model perancangan suatu saat nanti akan menjadi asupan bagi aktivitas-aktivitas selanjutnya yang kelak akan terdefinisi pada tahap implementasi (Adi Nugroho ; 2010 : 212).

II.2. Pengertian Aplikasi

Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini menurut Rahmatillah (2011 ; 3) dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

II.3. Stop Motion

Stop motion disebut juga *frame-by-frame*, objek seakan bergerak karena mempunyai banyak *frame* yang dijalankan secara beraturan. Teknik Animasi Ini Mengambarkan berbagai manipulasi objek yang mungkin termasuk model, tanah liat dan boneka. *Stop motion animation* dibuat dengan cara *shot* satu demi satu gambar dengan menggunakan video atau *movie* kamera, bisa juga menggunakan kamera *photography* analog ataupun digital, hasil *shot stil image* (gambar diam) tadi lalu disusun menjadi movie.

Gerak berhenti (bahasa Inggris: *stop motion* atau *stop frame*) adalah sebuah teknik animasi untuk membuat objek yang dimanipulasi secara fisik agar terlihat

bergerak dengan sendirinya. Objek tersebut digerakan sedikit demi sedikit di setiap *frame* yang akan difoto, menciptakan ilusi pergerakan sat serangkaian *frame* tersebut dimainkan secara berurutan berkelanjutan. Boneka dengan sendi yang dapat digerakan atau figur tanah liat sering digunakan dalam gerak henti karena alasan kemudahan meletaknya kembali. Animasi gerak henti yang menggunakan tanah liat disebut animasi tanah liat atau *clay mation* (Arif Ranu ; 2013 : 1).

II.3.1. Jenis-Jenis Animasi Stop Motion

Menurut Arif Ranu (2013 ; 2) ada 6 (enam) jenis animasi stop-motion yaitu :

1. Tanah Liat (*Clay*)

Sering disingkat claymation, ini adalah jenis animasi stop-motion di mana masing-masing bagian animasi, baik objek karakter atau latar belakang, menggunakan "deformasi bentuk " dengan bahan elastis seperti tanah liat.

2. Guntingan (*Cutout*)

Animasi cutout adalah teknik yang dibentuk dengan mengerakan potongan lembaran 2- dimensi. Animasi ini dibuat dengan menggunakan karakter datar, alat peraga dan latar belakang dari bahan seperti kertas, kartu, kain yang kaku, benda-benda mati, kain atau bahkan foto.

3. Gambar (*Graphic*)

Animasi grafis adalah variasi dari stopmotion yang lebih konseptual daripada animasi cel tradisional bidang datar dan animasi kertas gambar (*cutout*). Tapi

secara teknis ia termasuk stop motion yang dibuat dengan foto (secara keseluruhan atau sebagian). Animasi grafis dapat dilakukan dengan kamera hanya paning ke atas dan ke bawah dan /atau difoto secara individu, satu per satu.

4. Aktor Hidup (*Pixilation*)

Pixilation adalah teknik *stop motion* di mana yang digunakan adalah aktor hidup sebagai subjek *frame-by-frame*, aktor tersebut berpose berulang kali untuk satu atau lebih *frame* yang diambil (foto) dan bergerak sedikit demi sedikit ke *frame* berikutnya. Aktor ini menjadi semacam wayang hidup gerak berhenti dan bergerak.

5. Wayang (*Puppet*)

Animasi boneka biasanya melibatkan tokoh boneka seperti wayang berinteraksi satu sama lain dalam lingkungan yang dibentuk. Boneka wayang umumnya memiliki tulang rangka untuk menjaga mereka tetap dan mantap serta menghambat mereka agar selalu bergerak pada sendi tertentu. Wayang digunakan dan dipotret setiap kali posisi mereka berubah sedikit demi sedikit untuk membuat gerakan. Dalam animasi ini kita memberikan kehidupan kepada benda mati dan membuat mereka terlihat seperti hidup.

6. Bayangan (*Silhouette*)

Animasi guntingan yang dijadikan serangkaian bayangan (hitam) gambar gelap, dan disebut sebagai animasi siluet. Teknik ini dipelopori oleh animator *Lote German Reiniger*.

II.4. *Smartphone*

Smartphone atau bisa disebut dengan telepon pintar/cerdas sudah menjadi sebuah kebutuhan bagi sekian orang di dunia ini sebagai penunjang aktivitas kerja maupun sekedar lifestyle atau gaya hidup. Telepon cerdas (*smartphone*) adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, kadang-kadang dengan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti telepon cerdas. Bagi beberapa orang, telepon pintar merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, telepon cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat elektronik), internet dan kemampuan membaca buku elektronik (e-book) atau terdapat papan ketik (baik sebagaimana jadi maupun dihubung keluar) dan penyambung VGA. Dengan kata lain, telepon cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon. Pertumbuhan permintaan akan alat canggih yang mudah dibawa kemana-mana membuat kemajuan besar dalam pemroses, ngingatan, layar dan sistem operasi yang diluar dari jalur telepon genggam sejak beberapa tahun ini.

Dengan menggunakan telepon pintar yang hanya merupakan sebuah evolusi dari jenjang-jenjang evolusi, jadi kemungkinan alat ini pada titik tertentu akan menjadi lebih kecil dan kita tidak akan menyebutnya telepon lagi, tetapi ia akan terintegrasi, kesepakatannya disini adalah untuk membuat alat ini menjadi se-tidak terlihat mungkin, antara anda, dan apa yang anda ingin lakukan" kata

Sacha Wunsch-Vincent pada OECD (Organisasi untuk Kerjasama dan Pengembangan Ekonomi).

Kebanyakan alat yang dikategorikan sebagai telepon pintar menggunakan sistem operasi yang berbeda. Dalam hal fitur, kebanyakan telepon pintar mendukung sepenuhnya fasilitas surel dengan fungsi pengatur personal yang lengkap. Fungsi lainnya dapat menyertakan miniatur papan ketik QWERTY, layar sentuh atau D-pad, kamera, pengaturan daftar nama, penghitung kecepatan, navigasi piranti lunak dan keras, kemampuan membaca dokumen bisnis, pemutar musik, penjelajah foto dan melihat klip video, penjelajah internet, atau hanya sekedar akses aman untuk membuka surel (surat elektronik) perusahaan, seperti yang ditawarkan oleh BlackBerry. Fitur yang paling sering ditemukan dalam telepon pintar adalah kemampuannya menyimpan daftar nama sebanyak mungkin, tidak seperti telepon genggam biasa yang mempunyai batasan maksimum penyimpanan daftar nama (Nekie Jocom ; 2013 : 7).

II.5. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi.”. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah *Open Handset Alliance*

(OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (M. Ichwan ; 2011 : 8).

II.6. Pengertian Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan oleh *Sun Microsystem* dan dirilis tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan *versi* yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, serta Sembilan pemrograman lainnya dari *Sun Microsystem*. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang (Wahana Komputer ; 2010 : 1).

II.7. UML (*Unified Modeling Language*)

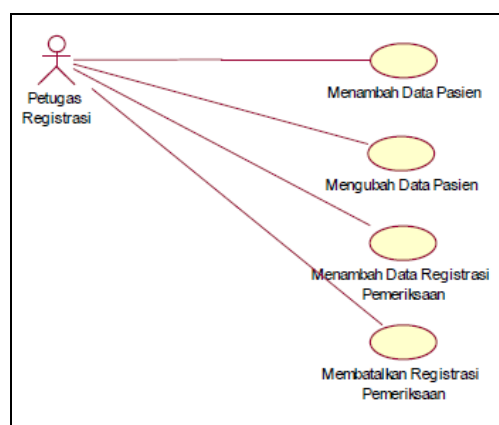
Menurut Havaluddin (2011 ; 3) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. Use Case Diagram

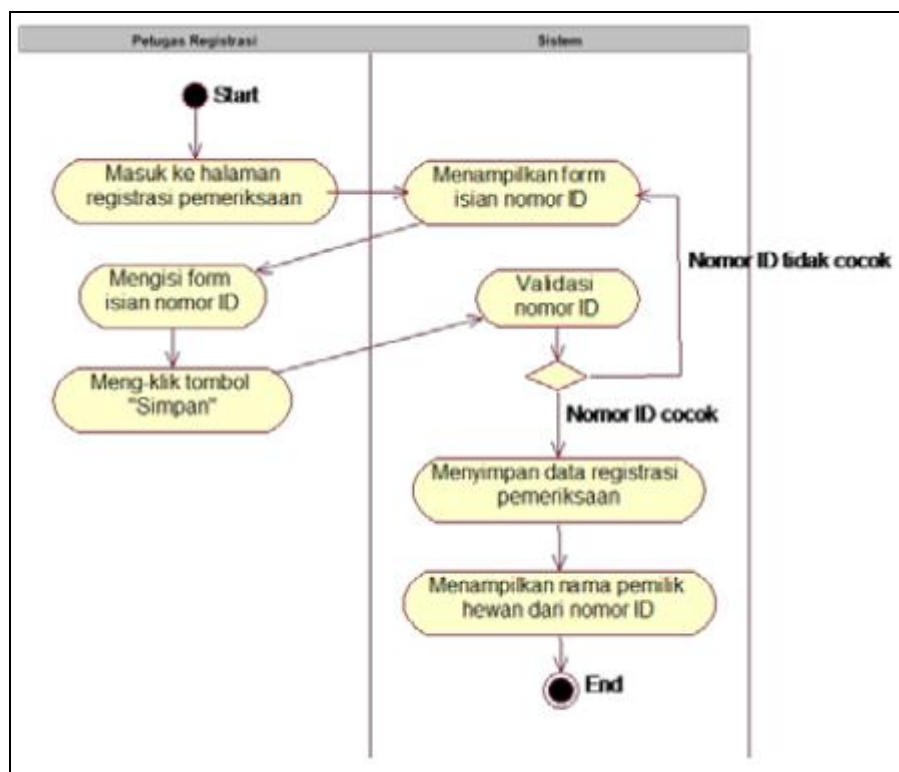
Use case diagram dibuat berdasarkan pengguna sistem informasi rawat jalan dan fungsi-fungsi yang ditangani oleh sistem informasi rawat jalan yang didefinisikan pada sub bab analisis sistem rawat jalan yang akan dibuat. Aktor-aktor pada *use case diagram* didapatkan dari pengguna sistem informasi rawat jalan, sedangkan *use case* untuk setiap aktor didapatkan dari fungsi-fungsi yang ditangani oleh sistem informasi rawat jalan. Contoh gambar *use case* untuk aktor petugas registrasi dan dokter pada rancangan sistem informasi rawat jalan yang akan dibuat dapat dilihat pada Gambar.



Gambar II.1. Contoh Usecase diagram
(Sumber : Haviluddin, 2011 ; 3)

2. Activity Diagram

Activity diagram dibuat berdasarkan aliran dasar dan aliran alternatif pada skenario *use case diagram*. Pada *activity diagram* digambarkan interaksi antara aktor pada *use case diagram* dengan sistem. Contoh *activity diagram* yang dibuat berdasarkan skenario pada *use case* menambah data registrasi pemeriksaan dan menambah data pemeriksaan medis lanjutan dapat dilihat pada Gambar

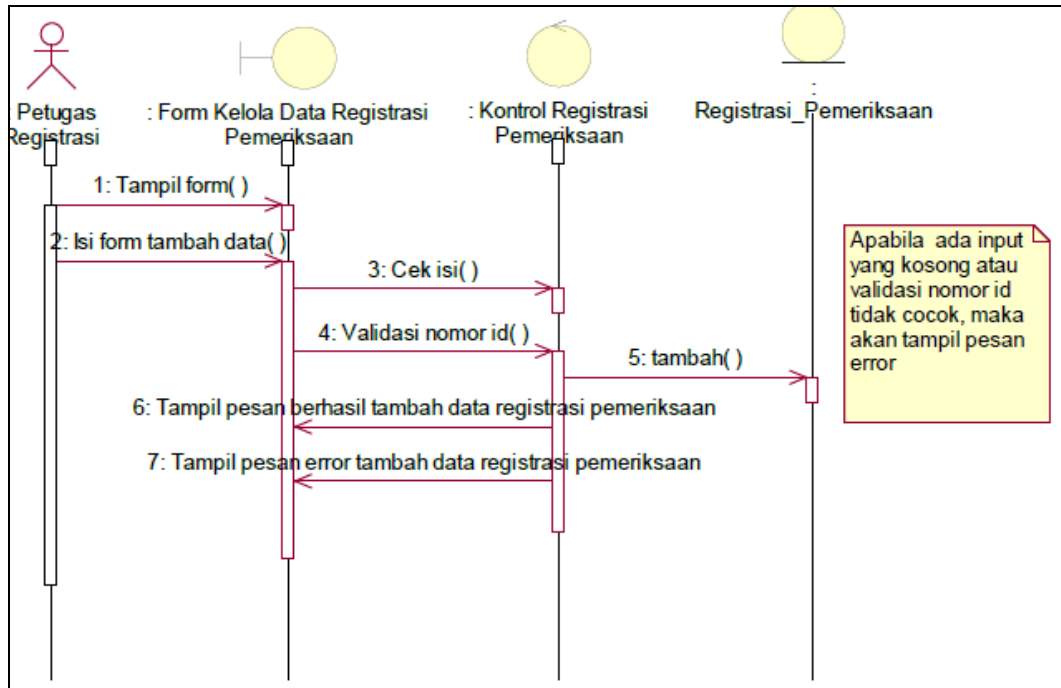


Gambar II.2. Contoh Activity Diagram
(Sumber : Haviluddin, 2011 ; 3)

3. Sequence diagram

Sequence diagram dibuat berdasarkan *activity diagram* dan *class diagram*. *Sequence diagram* menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada *class diagram* dengan menggunakan operasi yang dimiliki kelas tersebut. Untuk aliran pesan, *sequence diagram* merujuk pada alur sistem

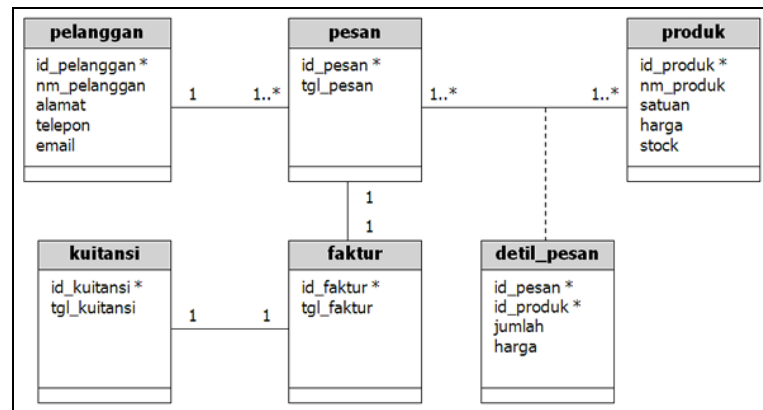
activity diagram yang telah dibuat sebelumnya. Berikut adalah contoh *sequence diagram* menambah data registrasi pemeriksaan :



Gambar II.3. Contoh Sequence Diagram
(Sumber : Haviluddin, 2011 ; 3)

4. Class Diagram

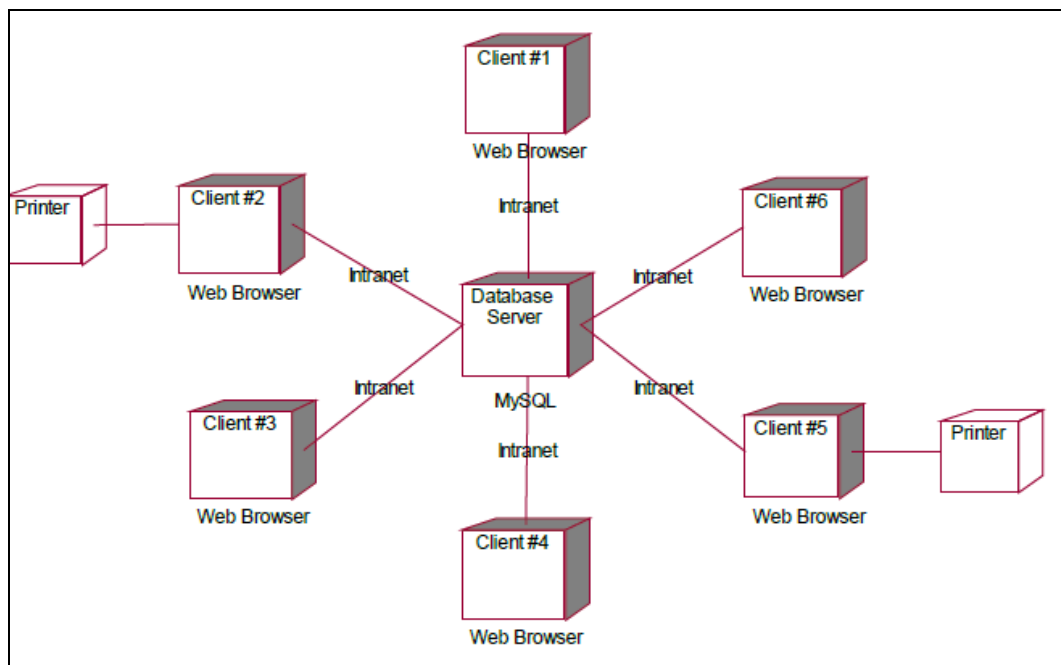
Class diagram dibuat berdasarkan *use case diagram* dan *activity diagram*. *Class diagram* dapat mendeskripsikan kelas-kelas yang digunakan dalam sistem informasi rawat jalan yang akan dibuat. Kelas dengan jenis *actor* diperoleh dari aktor-aktor yang digambarkan dalam *use case diagram*, sedangkan kelas dengan jenis lain seperti *entity*, *boundary*, dan *control* diperoleh dari gambaran alur sistem pada *activity diagram*. Berikut adalah contoh setiap jenis kelas :



Gambar II.4. Contoh Class Diagram
(Sumber : Haviluddin, 2011 ; 3)

5. Deployment Diagram

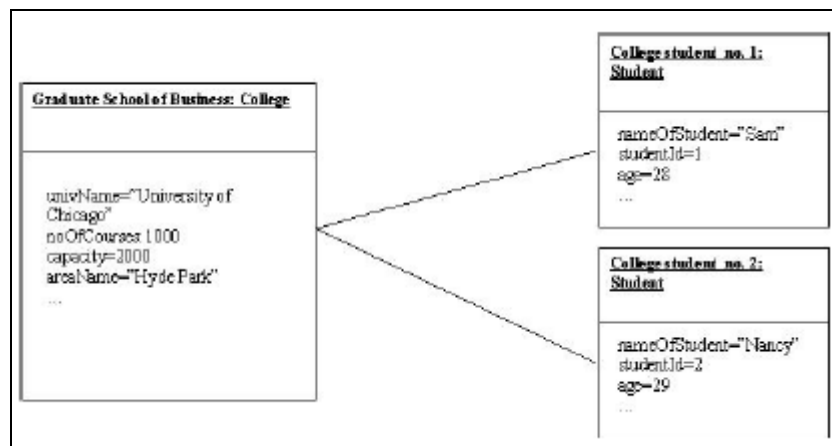
Deployment diagram dibuat untuk menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, dan keterhubungan antara komponen-komponen *hardware* tersebut.



Gambar II.5. Contoh Deployment Diagram
(Sumber : Haviluddin, 2011 ; 3)

6. Objek Diagram

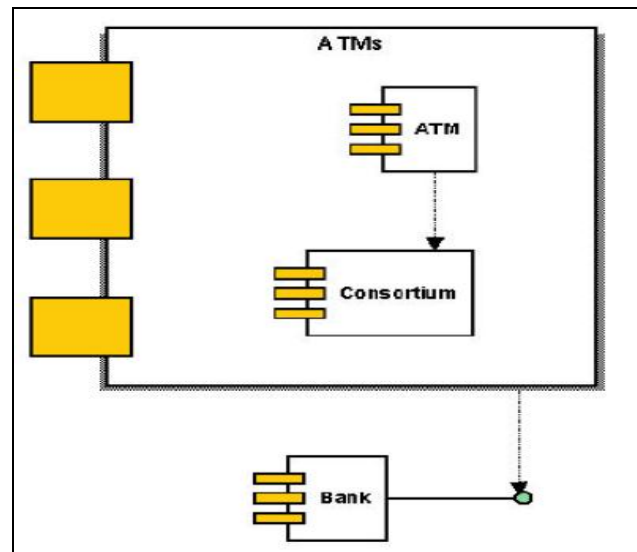
Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.



Gambar II.6. Contoh Objek Diagram
(Sumber : Haviluddin, 2011 ; 3)

7. Component diagram

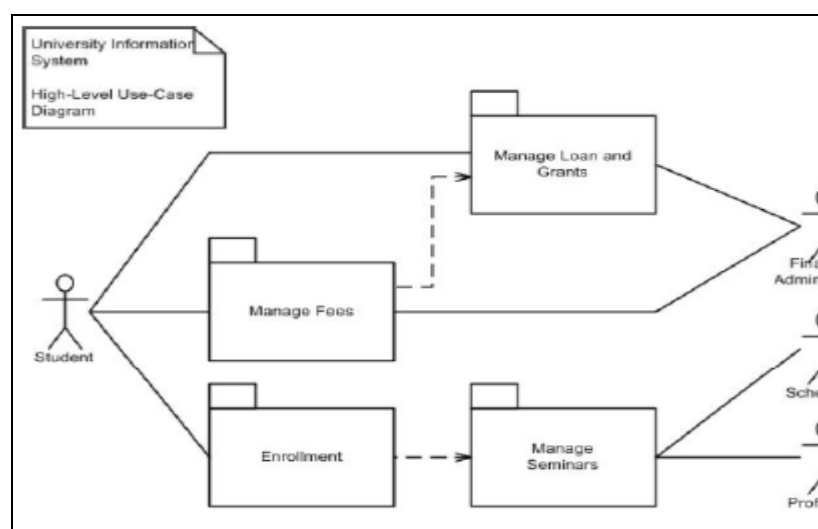
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.7. Contoh *Component Diagram*
(Sumber : Havaluddin, 2011 ; 3)

8. *Package Diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



Gambar II.8. Contoh *Package Diagram*
(Sumber : Havaluddin, 2011 ; 3)