

BAB II

TINJAUAN PUSTAKA

II.1. Akuntansi

Akuntansi adalah sistem informasi yang menghasilkan informasi keuangan kepada pihak-pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi suatu perusahaan. Hasil dari proses akuntansi disebut dengan laporan keuangan. Informasi yang dihasilkan dari proses akuntansi tersebut harus dapat menjawab kebutuhan umum para pemakainya. Karena itu, laporan keuangan suatu badan usaha harus memiliki kualitas yang diperlukan oleh berbagai pihak yang membutuhkan informasi keuangan tersebut.

Akuntansi terutama mementingkan aktivitasnya pada mendesain sistem pencatatan, menyiapkan laporan keuangan berdasarkan data yang ada, dan menginterpretasikan laporan tersebut. Akuntan sering kali memeriksa pekerjaan para pemegang pembukuan. Di perusahaan besar, akuntan memiliki tanggung jawab dan otoritas yang lebih besar dibandingkan pemegang pembukuan. (Rudianto ; 2012 : 4).

II.2. Sistem Informasi Akuntansi

Sistem informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari sistem informasi akuntansi adalah :

1. Mendukung operasi sehari-hari.
 2. Mendukung pengambilan keputusan manajemen .
 3. Memenuhi kewajiban yang berhubungan dengan pertanggung-jawaban
- (Kusrini : 2007 : 10).

II.3. Penjualan

Penjualan merupakan tujuan utama dilakukannya kegiatan perusahaan. Perusahaan dalam menghasilkan barang/jasa, mempunyai tujuan akhir yaitu menjual barang/jasa tersebut kepada masyarakat. Oleh karena itu, penjualan memegang peranan penting bagi perusahaan agar produk yang dihasilkan oleh perusahaan dapat terjual dan memberikan penghasilan bagi perusahaan. Penjualan yang dilakukan perusahaan bertujuan untuk menjual barang/jasa yang diperlukan sebagai sumber pendapatan untuk menutup semua ongkos guna memperoleh laba.

Penjualan adalah pemindahan hak milik barang atau pemberian jasa yang dilakukan penjualan kepada pembeli dengan harga yang disepakati bersama dengan jumlah yang dibebankan kepada pelanggan dalam penjualan barang/jasa dalam suatu periode akuntansi (Freddy Rangkuti ; 2009 : 206).

II.4. Pembelian

Pembelian adalah jumlah barang yang dibeli pada suatu periode. Pembelian dapat dilakukan secara tunai maupun kredit, yang masing-masing akan mendapatkan perlakuan berbeda. Biasanya untuk pembelian tunai mendapatkan potongan tunai. Selain itu, beberapa pembelian kredit juga diberikan syarat 2/10,

n/30. Syarat 2/10 ini mempunyai arti apabila pembayaran dilakukan dalam tempo 10 hari sejak tanggal pembelian maka akan mendapatkan potongan sebesar 2%. Sementara n/30 berarti komitmen untuk melunasi pembayaran dalam tempo maksimal 30 hari.

Potongan pembelian biasa juga disebut potongan tunai pembelian. Potongan ini diberikan berupa pengurangan sejumlah harga dari harga pembelian. Seperti dijelaskan diatas, pemberian potongan ini dijelaskan melalui syarat pembayaran tertentu (Himayati ; 2008 : 25).

II.5. Metode FIFO

Metode alur / flow menunjukan cara suatu barang diambil dari inventori. Alur inventori dalam akunting mungkin tidak akan sama dengan alur dalam keadaan sebenarnya dari barang jadi. Dengan FIFO, biaya inventori diperhitungkan dalam barang yang siap dijual atau dikonsumsi yang sudah ada lebih lama dan hal itu berarti stock yang tersedia adalah pembelian yang paling lama atau paling dulu diproduksi dan unit yang digunakan akan dibebankan pada harga dari barang yang terlama.

Metode yang dapat digunakan ada 2, yaitu :

1. Perpetual

Dengan cara ini setiap keluar atau masuknya barang dapat diketahui dengan pasti dan terinci, karena selalu dicatat setiap jenis barang yang keluar atau masuk, serta biaya bahan yang dikeluarkan. Dengan cara ini dapat

memudahkan kita dalam melakukan pengecekan terhadap keluar masuknya barang.

2. Periodik

Dengan cara periodik persediaan barang dapat diketahui dalam satu periode tertentu, namun dengan cara ini keluar masuknya barang tidak dapat diketahui dengan rinci, karena dalam pencatatan hanya masuknya barang saja yang dicatat. Untuk mengetahui berapa biaya bahan baku yang dipakai dalam produksi harus dilakukan dengan cara menghitung sisa persediaan bahan baku yang masih ada digudang pada akhir periode akuntansi. Harga pokok persediaan awal ditambah harga pokok bahan baku yang dibeli dikurangi harga pokok persediaan bahan baku yang masih ada pada akhir periode adalah biaya-biaya bahan baku yang dipakai (Faisal Rahman ; 2013 : 3).

II.6. 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 *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. NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman *enterprise*, dan pemrograman perangkat mobile. Saat ini NetBeans telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15).

II.8. Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat

disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*.

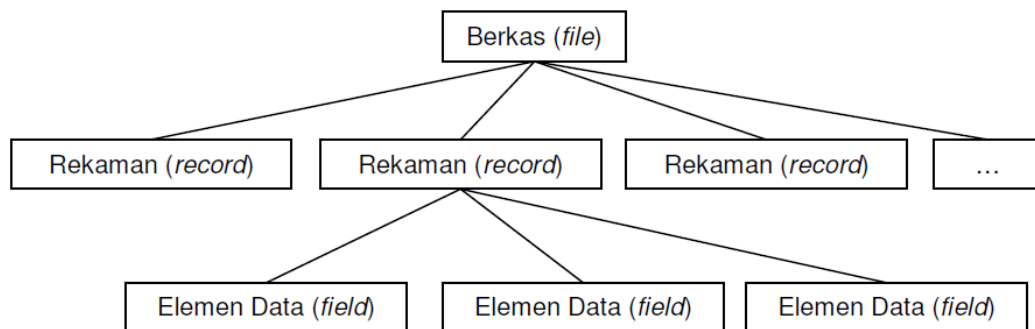
Pengaplikasian *database* dapat kita lihat dan rasakan dalam keseharian kita. *Database* ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (*anjungan tunai mandiri / automatic teller machine*) bank karena banktelah mempunyai *database* tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks *database* sebenarnya kita sudah melakukan perubahan (*update*) data pada *database* di bank. Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep *database*. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir(*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya (Agustinus Mujilan ; 2012 : 23).

II.8.1. Hierarki Database

Data diorganisasikan ke dalam bentuk elemen data (*field*), rekaman (*record*), dan berkas (*file*). Definisi dari ketiganya adalah sebagai berikut:

Elemen data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Misalnya data siswa terdiri dari NIS, Nama,

Alamat, Telepon atau Jenis Kelamin. Rekaman merupakan gabungan sejumlah elemen data yang saling terkait. Istilah lain dari rekaman adalah baris atau tupel. Berkas adalah himpunan seluruh rekaman yang bertipe sama (Haidar Dzacko ; 2007 : 1).



Gambar II.1. Hirarki Data
(Sumber : Haidar Dzacko ; 2007 : 1)

II.8.2. Model Database

Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu:

1. Model data konseptual (*high level*) menyajikan konsep tentang bagaimana *user* memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyajian data yaitu:
 - a. *Entity* (entitas) merupakan penyajian obyek, kejadian atau konsep dunia nyata yang keberadaannya secara eksplisit didefinisikan dan disimpan dalam basis data, contohnya Mahasiswa, Matakuliah, Dosen, Nilai dan lain sebagainya.

- b. *Attribute* (atribut) adalah keterangan-keterangan yang menjelaskan karakteristik dari suatu entitas seperti NIM, Nama, Fakultas, Jurusan untuk entitas Mahasiswa.
 - c. *Relationship* (hubungan) merupakan hubungan atau interaksi antara satu entitas dengan yang lainnya, misalnya entitas pelanggan berhubungan dengan entitas barang yang dibelinya.
2. Model data fiskal (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pencarian rekaman data lebih efisien.
 3. Model data implementasi (*representational*) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para *user* mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan oleh model hierarki, jaringan dan relasional (Haidar Dzacko ; 2007 : 3).

II.9. MySQL

Menurut Supardi (2007:97), perangkat lunak MySQL adalah perangkat lunak basis data *server* yang terkenal dan bersifat open-source dengan dukungan driver yang luas dari berbagai vendor. MySQL adalah seakuntansi implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

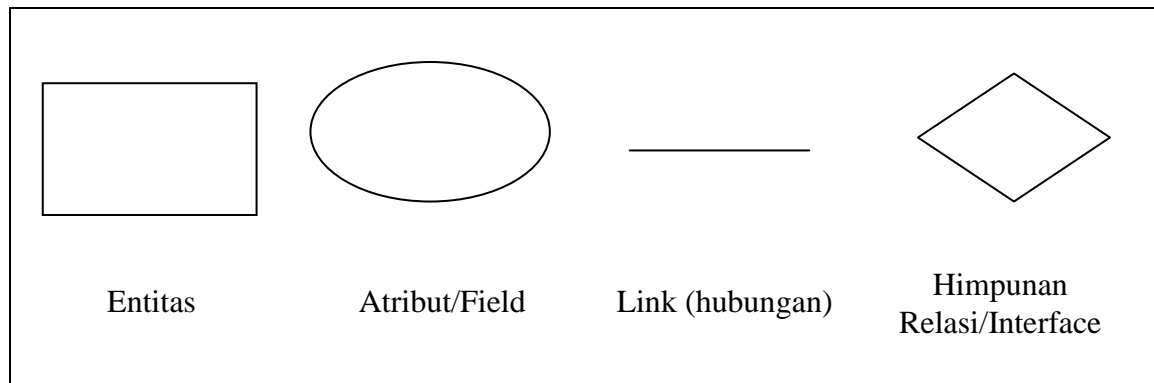
MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya. SQL (*Structured Query Language*). SQL adalah seakuntansi konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basisdata *non-transaksional*. Pada modus operasi *non-transaksional*, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak pengelola basis data kompetitor lainnya. Namun demikian pada modus *non-transaksional* tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus *non-transaksional* hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web*, CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus *non-transaksional*.

II.10. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini,

tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Yuniar Supardi ; 2010 : 448).



Gambar. II.2. Bentuk Simbol ERD
(Sumber : Yuniar Supardi ; 2010 : 448)

II.11. Kamus Data

Kamus data (*data dictionary*) mencakup definisi-definisi dari data yang disimpan di dalam basis data dan dikendalikan oleh sistem manajemen basis data. Figur 6.5 menunjukkan hanya satu tabel dalam basis data jadwal. Struktur basis data yang dimuat dalam kamus data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel, dan hal-hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai-nilai yang valid untuk data, dan karakteristik-karakteristik lainnya akan disimpan dalam kamus data. Perubahan-perubahan pada struktur data hanya dilakukan satu kali di dalam kamus data, program-program aplikasi yang mempergunakan data tidak akan ikut terpengaruh (Raymond McLeod ; 2008 : 171).

II.12. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu (Kusrini ; 2007 : 39).

II.12.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1st Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

c. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

d. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut superkey (Kusrini ; 2007 : 39).

II.13. UML (*Unified Modeling Language*)

Menurut Haviluddin (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.

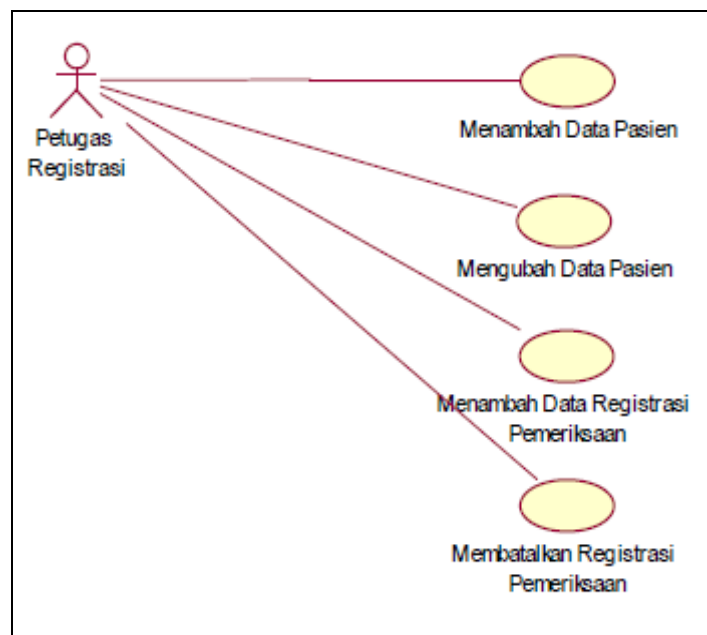
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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

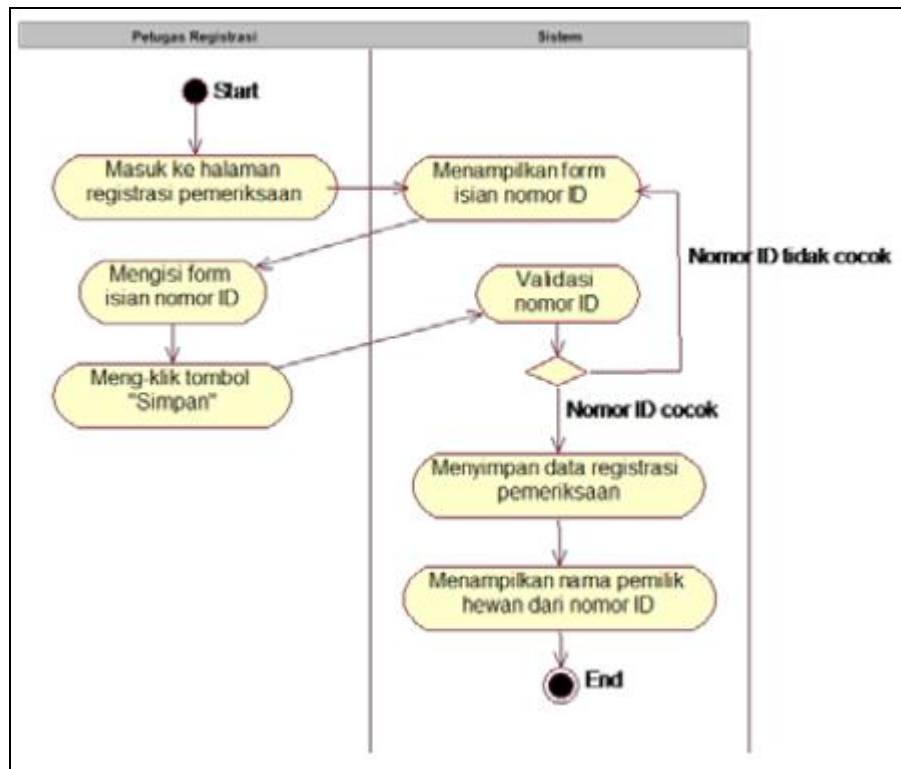
ditangani oleh sistem informasi rawat jalan. Contoh gambar *use case* untuk aktor petugas registrasi dan dokter pada rancangan sistem informasi rawat jalan yang akan dibuat dapat dilihat pada Gambar.



Gambar II.3. 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

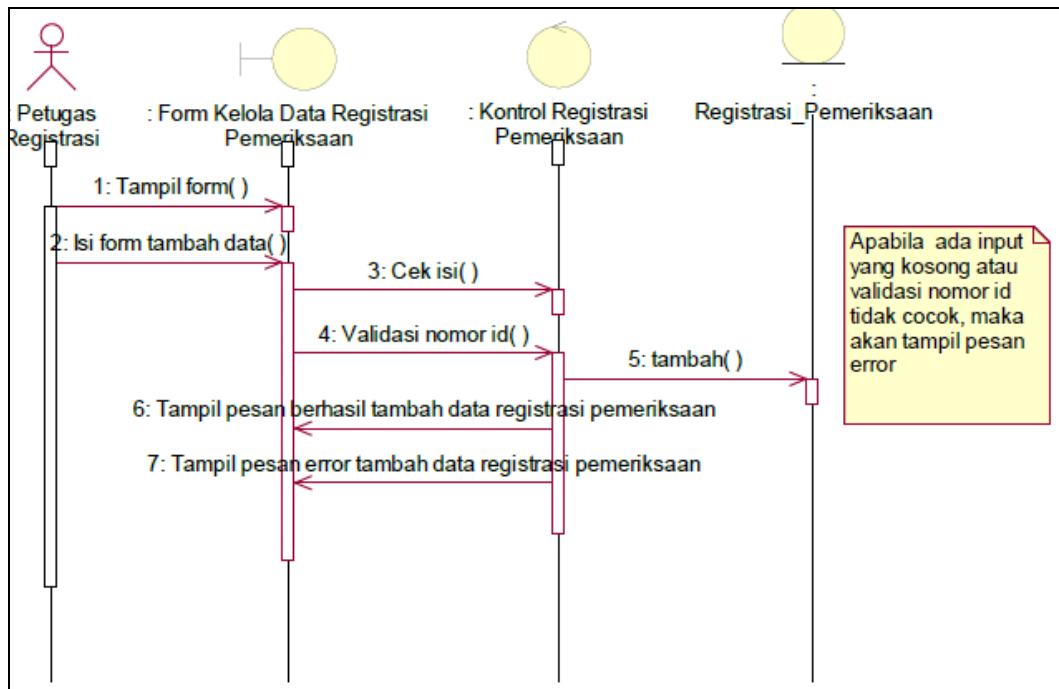


Gambar II.4. Contoh Activity Diagram
Sumber : Haviluddin ; 2011 : 3

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 :



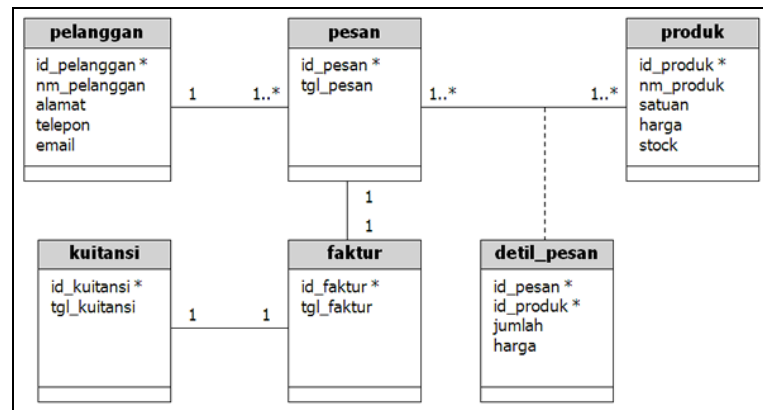
Gambar II.5. Contoh Sequence Diagram

Sumber : Haviluddin ; 2011 : 3

4. Class Diagram

Class diagram dibuat berdasarkan *use case diagram* dan *activity diagram*.

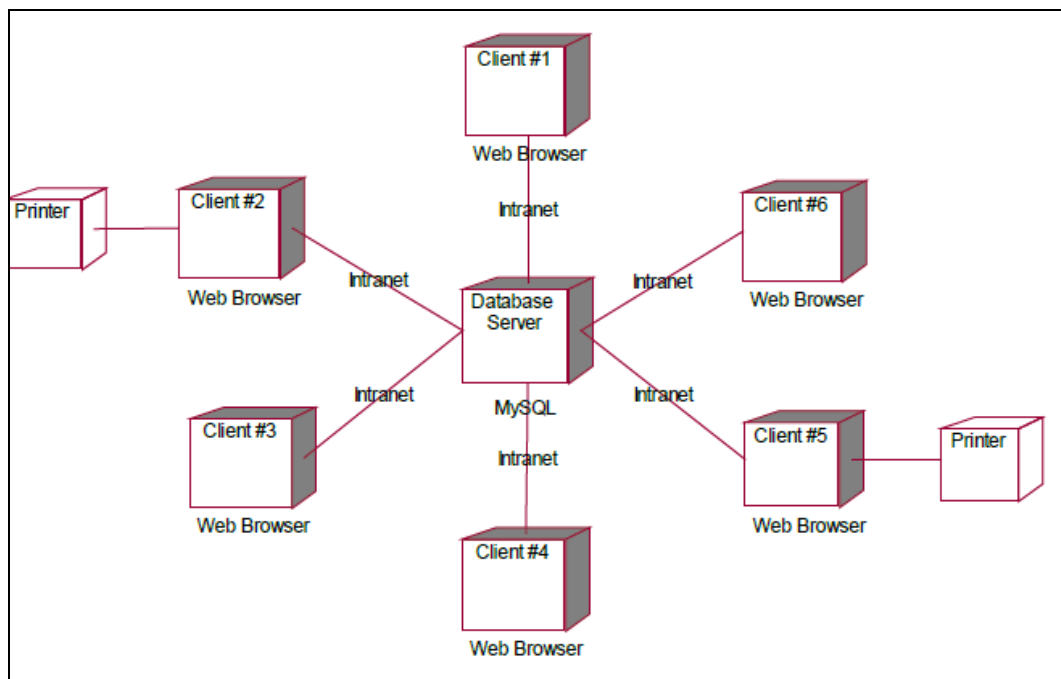
Class diagram dapat mendeskripsikan kelas-kelas yang digunakan dalam sistem informasi rawat jalan yang akan dibuat. Kelas dengan jenis *actor* diperoleh dari aktor-aktor yang digambarkan dalam *use case diagram*, sedangkan kelas dengan jenis lain seperti *entity*, *boundary*, dan *control* diperoleh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



Gambar II.6. Contoh Class Diagram
Sumber : Haviluddin ; 2011 : 3

5. Deployment Diagram

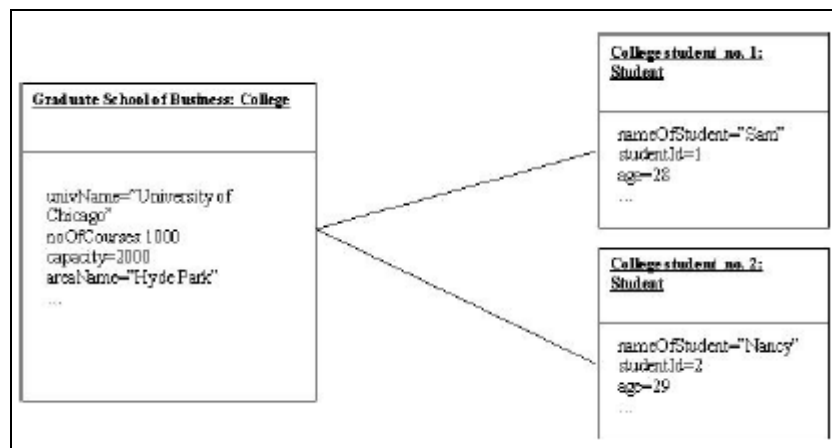
Deployment diagram dibuat untuk menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, dan keterhubungan antara komponen-komponen *hardware* tersebut.



Gambar II.7. Contoh Deployment Diagram
Sumber : Haviluddin ; 2011 : 3

6. Objek Diagram

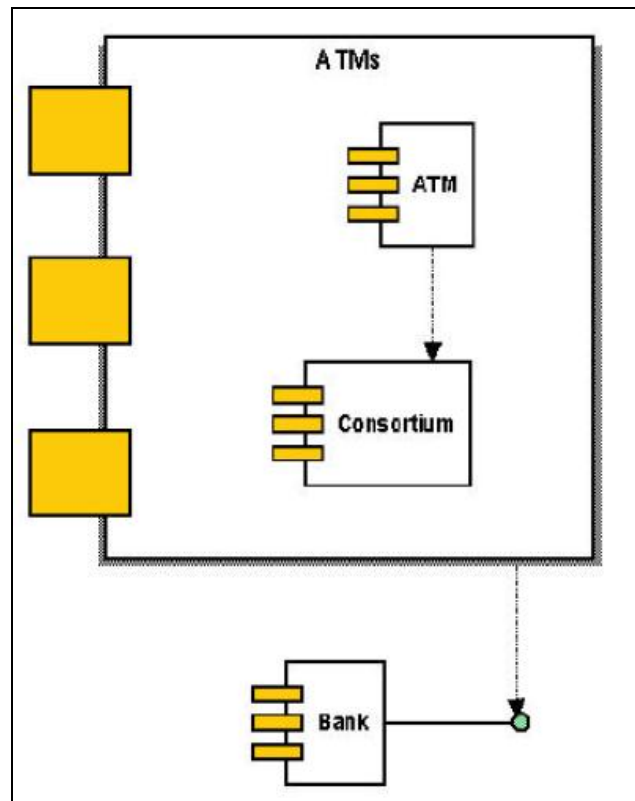
Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.



Gambar II.8. Contoh Objek Diagram
Sumber : Haviluddin ; 2011 : 3

7. Component diagram

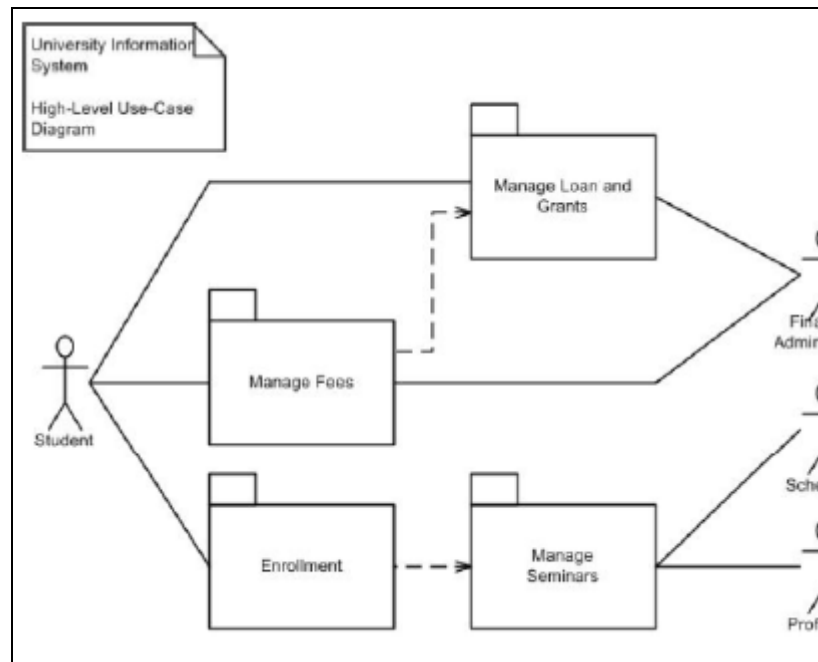
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.9. Contoh *Component Diagram*
Sumber : Haviluddin ; 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.10. Contoh Package Diagram
Sumber : Haviluddin ; 2011 : 3