

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Secara leksikal, sistem berarti susunan yang teratur dari pandangan, teori, asas dan sebagainya. Dengan kata lain, sistem adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antara bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interpenden satu sama lain. Selain itu dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan-perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dan elemen-elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan (Marimin ; 2008 : 1).

II.1.1. Karakteristik Sistem

Ada beberapa karakteristik sistem, yaitu :

a. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan.

b. Batasan masalah

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

c. Lingkungan luar sistem

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

d. Penghubung sistem

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau interface.

e. Masukan sistem

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan sinyal.

f. Keluaran sistem

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

g. Pengolah sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran sistem

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. (Tata Sutabri : 2007 ; 9-13).

II.2. Pengertian Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomis pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi tergantung pada tingkatannya (Agustinus ; 2012 : 1).

II.3. Pengertian Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan

penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38).

II.4. Akuntansi

Akuntansi merupakan bahasa bisnis. Sebagai bahasa bisnis akuntansi menyediakan cara untuk menyajikan dan meringkas kejadian-kejadian bisnis dalam bentuk informasi keuangan kepada pemakainya. Informasi akuntansi merupakan bagian terpenting dari seluruh informasi yang diperlukan oleh manajemen. Informasi akuntansi yang dihasilkan oleh suatu sistem dibedakan menjadi dua, yaitu informasi akuntansi keuangan dan informasi akuntansi manajemen.

Pemakai informasi akuntansi pun terdiri dari dua kelompok, yaitu pemakai eksternal dan pemakai internal. Yang dimaksud dengan pemakai eksternal mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat kerja dan masyarakat. Sedangkan pemakai internal adalah pihak manajer dari berbagai tingkatan dalam organisasi bersangkutan (Kusrini ; 2012 : 1).

II.5. Sistem Informasi Akuntansi

Informasi akuntansi diberikan sebagai alat atau sarana untuk membantu manajer menjalankan fungsi-fungsi manajemen sehingga tujuan organisasi dapat tercapai. Fungsi manajemen, seperti perencanaan (*planning*), pengorganisasian (*organizing*), pengarahan karyawan (*directing*) dan pengendalian (*controlling*) tidak dapat dilakukan tanpa informasi yang memadai. Informasi dalam sebuah organisasi merupakan perekat yang mengikat fungsi-fungsi manajemen dalam sebuah sistem sehingga memungkinkan organisasi bertindak koheren dan harmonis antar berbagai fungsi.

1. Perencanaan (*Planning*)

Tahap pertama dalam perencanaan adalah mengidentifikasi alternatif alternatif yang tersedia, kemudian memilih salah satu dari berbagai alternatif tersebut yang paling baik dan cocok dengan tujuan organisasi. Rencana manajemen biasanya diekspresikan dalam sebuah bentuk formal yang disebut dengan anggaran (*budgets*). Oleh karena itu, istilah penganggaran (*budgeting*) sering pula disebut sebagai proses perencanaan. Anggaran biasanya dibuat di bawah kendali dan koordinasi seorang manajer atau kepala bagian akuntansi. Umumnya anggaran dibuat setiap tahun dan mencerminkan rencana manajemen dalam format yang spesifik dan kuantitatif. Anggaran ini dibuat berdasarkan data yang dikumpulkan, dianalisis dan diikhtisarkan oleh akuntan manajemen.

2. Pengorganisasian Sumber-sumber Daya (*Organizing*)

Langkah selanjutnya adalah pengorganisasian sumber-sumber daya, dimana manajer menetapkan koordinasi antar bagian (*organizationchart*). Perkembangan Pengorganisasian hanya merupakan alat untuk mencapai tujuan yang telah ditentukan.

3. Pengarahan Karyawan (*Directing and Motivating*)

Setelah rencana kerja untuk periode mendatang disusun, manajer harus melakukan pemantauan terus menerus terhadap kegiatan sehari-hari dan menjaga agar organisasi dapat berfungsi sebagaimana mestinya. Untuk melakukan hal tersebut, maka manajer harus memiliki kemampuan untuk memotivasi dan mengarahkan karyawan secara efektif.

4. Pengendalian dan Pengawasan (*Controlling*)

Dalam melaksanakan tugas pengendalian, manajer sebenarnya mencari jaminan bahwa rencana yang telah disusun dalam bentuk anggaran sudah dilaksanakan sebagaimana mestinya (Suprihatmi : 2009 : 57).

II.6. Pengertian Bentuk Single Step

Laporan laba/rugi adalah laporan yang menunjukkan besarnya pendapatan dan beban pada akhir periode akuntansi. Penyusunan laporan laba/rugi yang bersumber dari kertas kerja, datanya diambil dari kolom perhitungan laba/rugi. Penyusunan laporan laba/rugi dapat dibedakan atas dua bentuk, yaitu bentuk langsung (*single step*) dan bentuk bertahap (*multiple step*). Laporan laba rugi *single step*, adalah laporan laba rugi yang mencatat seluruh

penghasilan terlebih dahulu kemudian dikurangi dengan seluruh beban (Ryan Ariefriansyah ; 2013 : 14).

Salon Cantik	
Laporan Laba Rugi	
31 Desember 2006	
Penghasilan	
Penghasilan Usaha:
Penjualan Jasa
Penghasilan di Luar Usaha :	
Laba Penjualan Peralatan
Total Penghasilan
Beban	
Beban Usaha:	
Beban gaji
Beban perlengkapan
Beban penyusutan peralatan
Beban listrik dan air
Beban sewa ruangan
Beban asuransi
Beban usaha lain-lain
Jumlah beban usaha:	
Beban bunga
Total beban
Laba Bersih sebelum pajak

Gambar II.1 Contoh Bentuk Single Step
(Sumber : Ryan Ariefriansyah, 2013 ;14)

II.7. 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.8. Pengertian 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.9. Pengertian 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 (*automatic teller machine*) bank karena bank telah 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.10. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer database bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna.

Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer di antara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat anda download pada alamat resminya <http://www.mysql.com>. MySQL sudah cukup lama dikembangkan, beberapa *fase* penting dalam pengembangan MySQL adalah sebagai berikut :

- a. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
- b. Versi *windows* dirilis pada 8 Januari 1998 untuk *windows 95* dan *windows NT*.

- c. Versi 3.23 : beta dari Juni 2000, dan dirilis pada Januari 2001.
- d. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (*unions*)
(Wahana ; 2010 : 5).

II.11. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata ; 2010 : 67).

II.12. 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.13. Teknik Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

II.13.1. Bentuk-bentuk Normalisasi

1. 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.

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

Definisi :

Sebuah table disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. 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.

3. 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.

4. 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 :

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

5. 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).

6. Boyce Code Normal Form (BCNF)

- a. Memenuhi 1st NF
- b. Relasi harus bergantung fungsi pada atribut superkey (Kusrini ; 2007 : 39-43).

II.14. UML (*Unified Modeling Language*)

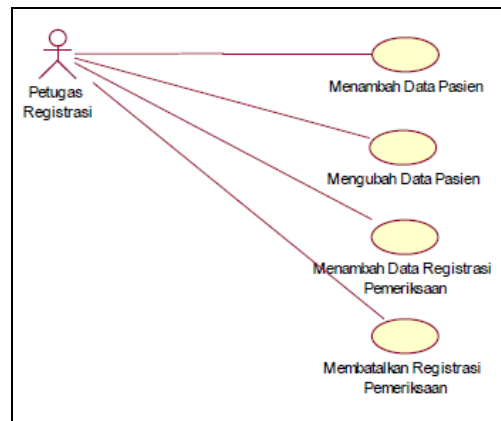
Menurut Haviluddin (2011 ; 3) *Unified Modelling Language* merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen.

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

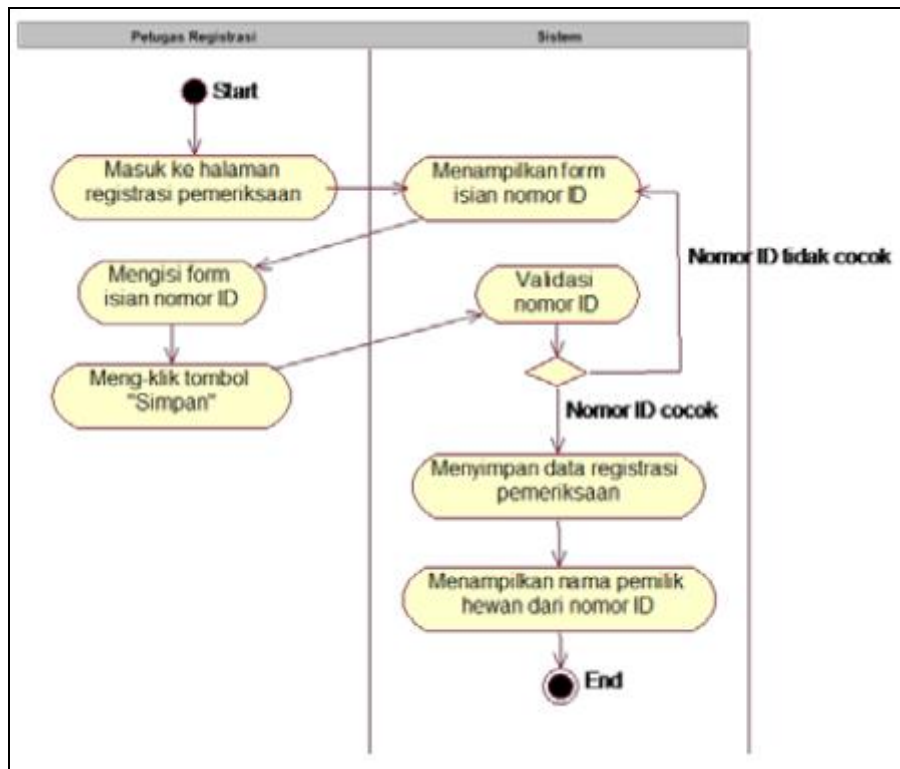
ditangani oleh sistem informasi rawat jalan. Contoh gambar *use case* untuk aktor petugas registrasi dan dokter pada rancangan sistem informasi rawat jalan yang akan dibuat dapat dilihat pada Gambar.



Gambar II.2 Contoh Usecase diagram
(Sumber : Haviluddin, 2011 ; 3)

2. Activity Diagram

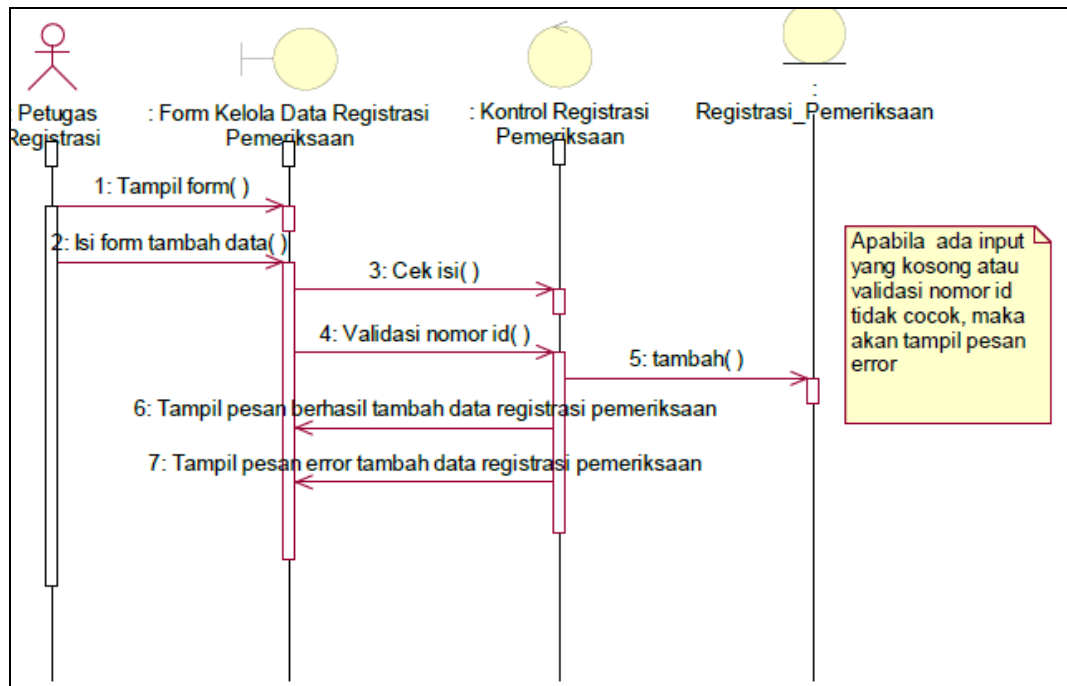
Activity diagram dibuat berdasarkan aliran dasar dan aliran alternatif pada skenario *use case diagram*. Pada *activity diagram* digambarkan interaksi antara aktor pada *use case diagram* dengan sistem. Contoh *activity diagram* yang dibuat berdasarkan skenario pada *use case* menambah data registrasi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



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

3. *Sequence diagram*

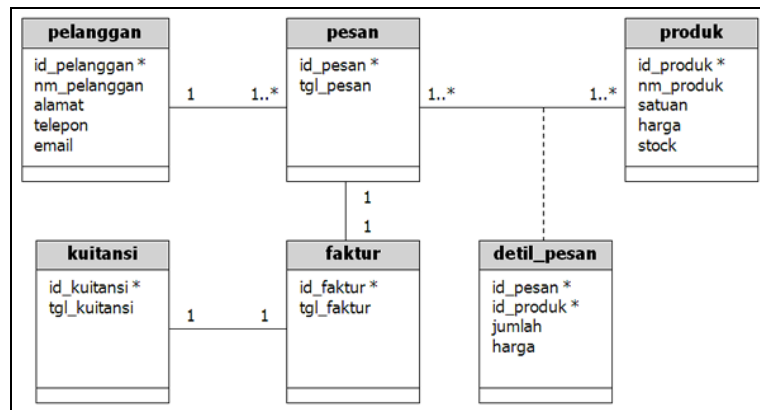
Sequence diagram dibuat berdasarkan *activity diagram* dan *class diagram*. *Sequence diagram* menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada *class diagram* dengan menggunakan operasi yang dimiliki kelas tersebut. Untuk aliran pesan, *sequence diagram* merujuk pada alur sistem *activity diagram* yang telah dibuat sebelumnya. Berikut adalah contoh *sequence diagram* menambah data registrasi pemeriksaan :



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

4. Class Diagram

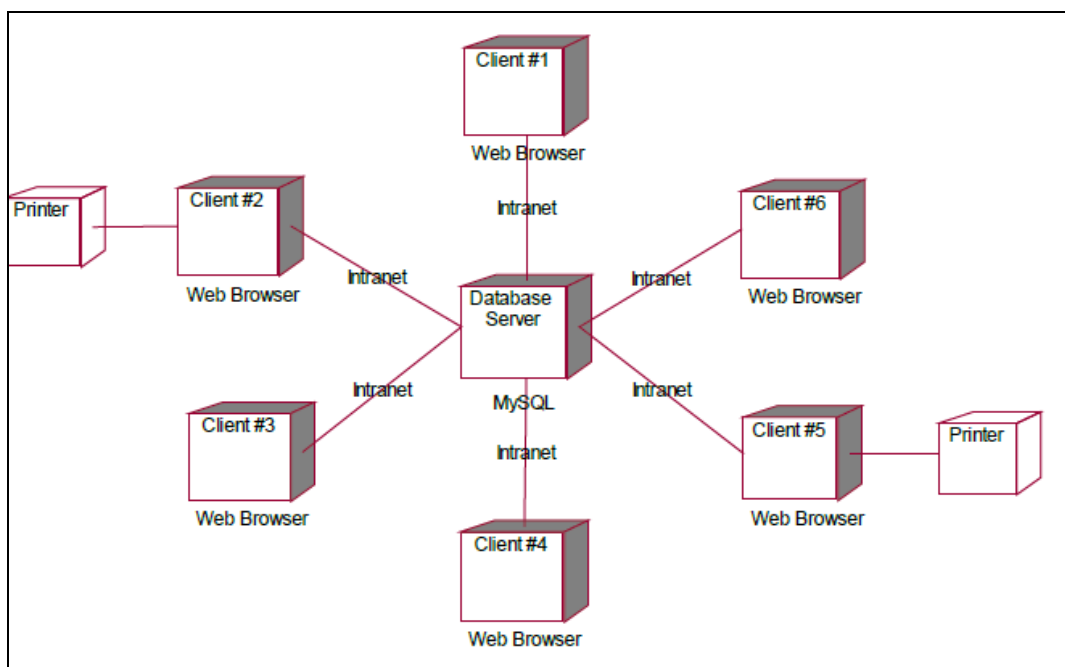
Class diagram dibuat berdasarkan *use case diagram* dan *activity diagram*. *Class diagram* dapat mendeskripsikan kelas-kelas yang digunakan dalam sistem informasi rawat jalan yang akan dibuat. Kelas dengan jenis *actor* diperoleh dari aktor-aktor yang digambarkan dalam *use case diagram*, sedangkan 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.5. Contoh Class Diagram
(Sumber : Haviluddin, 2011 ; 3)

5. Deployment Diagram

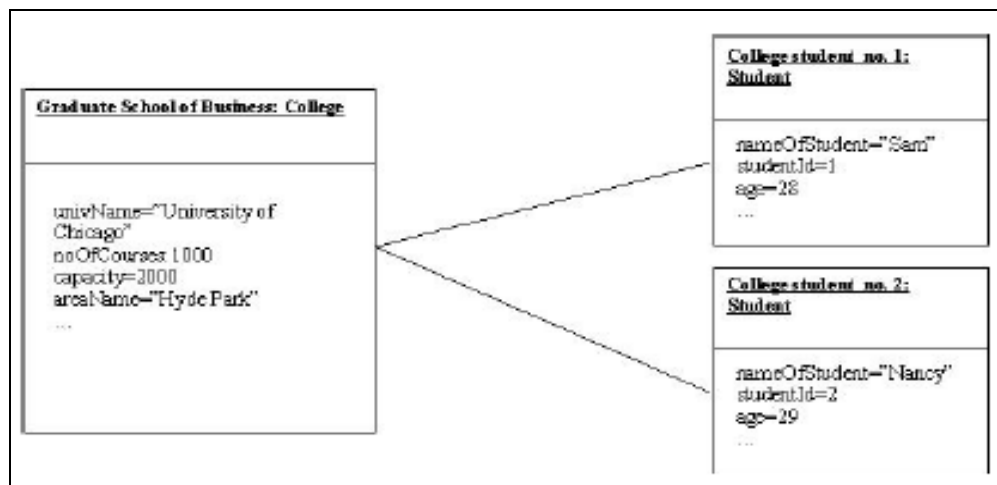
Deployment diagram dibuat untuk menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, dan keterhubungan antara komponen-komponen *hardware* tersebut.



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

6. Objek Diagram

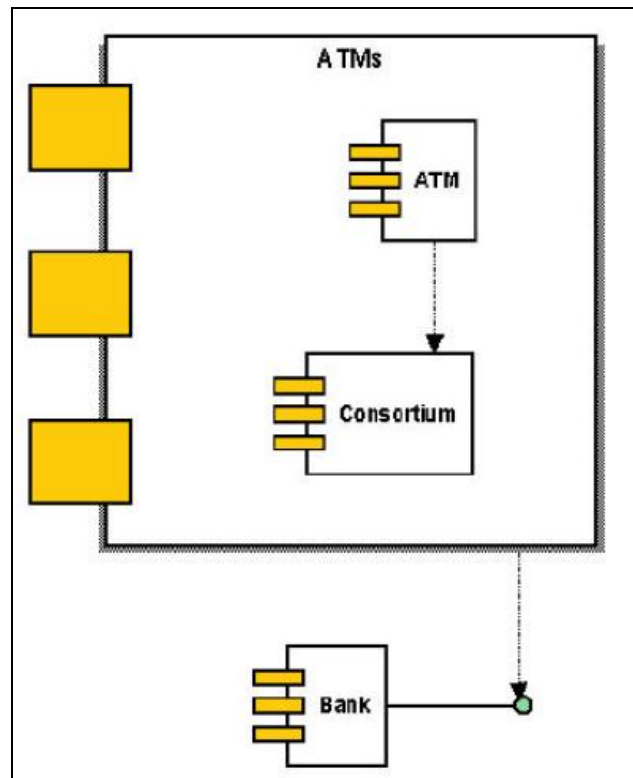
Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.



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

7. Component diagram

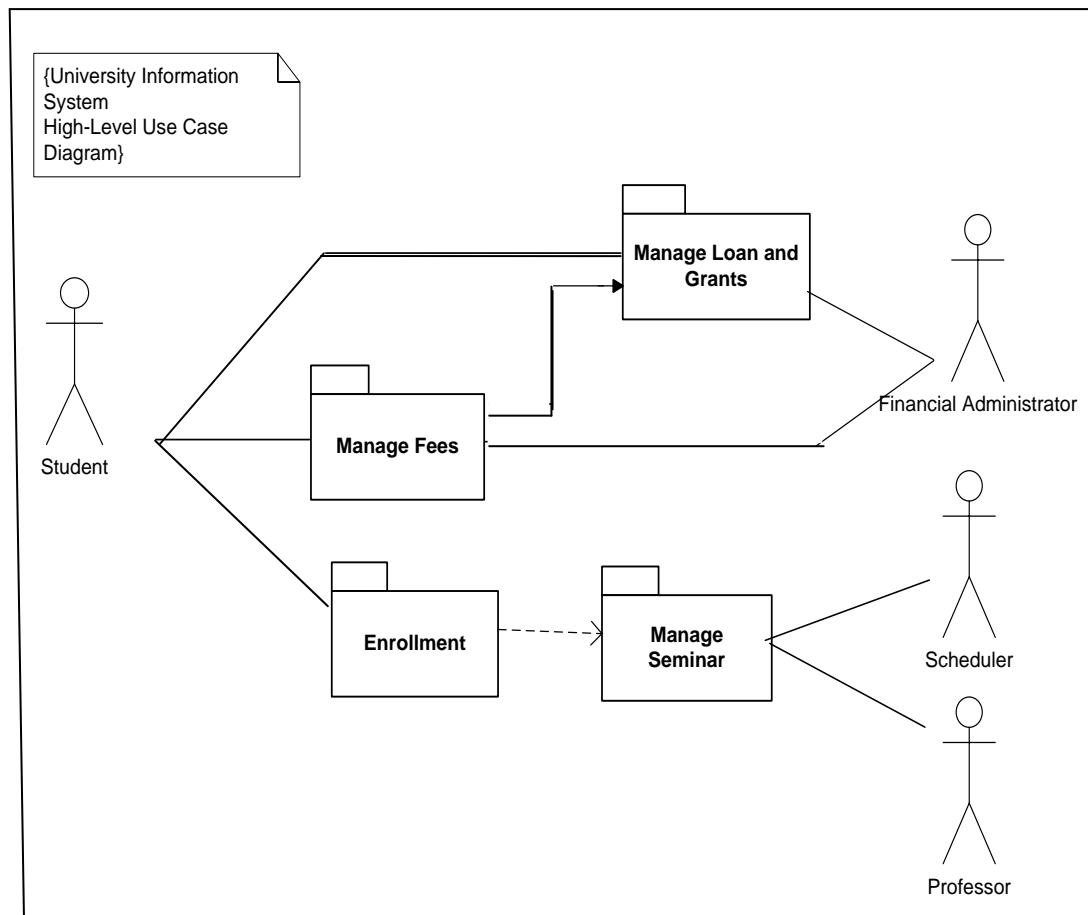
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.8. 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.9. Contoh Package Diagram

(Sumber : Havaluddin, 2011 ; 3)