BAB III
OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Dalam penelitian ini, penulis menjelaskan tentang sejarah Poliklinik PT Len Industri (Persero), visi dan misi Poliklinik PT Len Industri (Persero), struktur organisasi dan deskripsi tugas semua bagian dalam organisasi.

3.1.1. Sejarah Singkat Poliklinik PT Len Industri (Persero)

Poliklinik PT. Len Industri berdiri sejak tahun 2000, poliklinik ini didirikan untuk melayani kebutuhan karyawan perusahaan akan adanya pelayanan medis didalam lingkungan perusahaan. Seluruh pasien dari poliklinik ini adalah karyawan PT Len Industri. Setiap harinya paling sedikit ada satu pasien yang mengunjungi poliklinik ini, poliklinik ini memiliki seorang assisten dokter yang merangkap sebagai petugas rekam medis dan poliklinik ini hanya memiliki satu orang dokter dan satu orang resepsionis yang dirangkap oleh pelaksana harian. Kebanyakan pasien yang mengunjungi poliklinik ini bukan hanya untuk berobat tetapi juga untuk melakukan konsultasi medis. Jam kerja dokter di poliklinik ini tidak setiap hari tetapi disesuaikan dengan waktu kerja dokter tersebut di tempat lain.

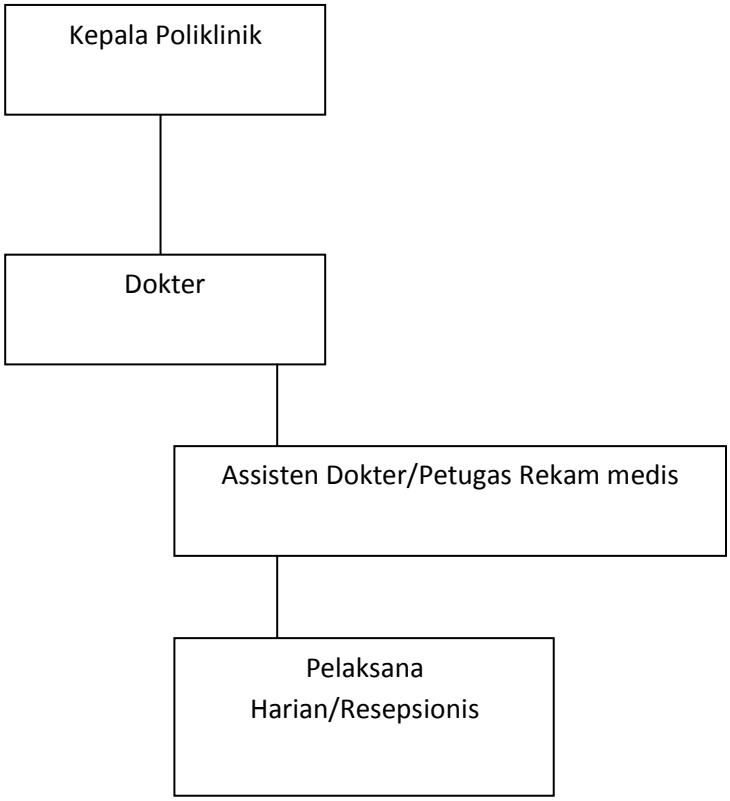
3.1.2. Visi dan Misi Poliklinik PT Len Industri (Persero)

Visi Poliklinik PT.Len Industri adalah untuk menjadikan perusahaan yang kuat melalui karyawan yang sehat

Misi Poliklinik PT.Len Industri adalah untuk memberikan pelayanan kesehatan yang baik kepada setiap insan Len

3.1.3. Struktur Organisasi Poliklinik PT Len Industri (Persero)

Secara umum poliklinik PT.Len Industri memiliki struktur organisasi yang sama seperti pada poliklinik lainnya.



Gambar 3.1 Struktur Organisasi Poliklinik PT Len Industri (Persero)

3.1.4. Deskripsi Tugas

1. Kepala Poliklinik

Tugas Kepala Poliklinik adalah untuk memimpin dan mengawasi semua fungsi poliklinik termasuk dengan kebutuhan - kebutuhannya. Dikarenakan poliklinik ini melekat didalam perusahaan maka fungsi kepala poliklinik berada di dalam kendali Sistem Manajemen.

2. Pelaksana Harian/Resepsionis

Tugas dari pelaksana harian adalah untuk melakukan pekerjaan harian poliklinik PT.Len Industri serta melakukan tugas penerimaan tamu dan atau pasien baru.

3. Dokter

Tugas dari dokter di poliklinik ini adalah memberikan pelayanan kesehatan kepada pasien sesuai dengan jadwal praktek yang dimilikinya.

4. Assisten Dokter

Tugas dari Assisten Dokter atau petugas rekam medis adalah melakukan tugas pokok dan fungsi dari assisten dokter yaitu menjadi repsesionis dan melakukan assistensi secara menyeluruh terhadap tugas dokter. Selain itu petugas ini juga bertugas sebagai petugas rekam medis dan administrasi di poliklinik ini.

3.2. Metode Penelitian

Metode penelitian merupakan cara penelitian yang digunakan untuk mendapatkan data agar mencapai tujuan tertentu.

Menurut Sugiyono (2009:3) Metode penelitian diartikan sebagai cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu

3.2.1. Desain Penelitian

Metode penelitian yang penulis gunakan adalah metode deskriptif dan metode *action* dengan studi kasus pada poliklinik PT Len Industri (Persero).

Metode penelitian deskriptif *(descriptive research)* bertujuan mendeskripsikan suatu keadaan atau fenomena apa adanya. Dalam studi ini peneliti tidak melakukan manipulasi semuakegiatan atau peristiwa berjalan seperti keadaan yang sebenarnya. Waktu yang dibutuhkan untuk melakukan penelitian ini dapat disesuaikan dengan kebutuhan, waktu penelitian bisa menjadi sangat panjang tetapi dapat pula singkat.

Metode penelitian action *(action research)* merupakan bentuk penelitian terapan bertujuan untuk mencari cara yang paling efektif untuk menghasilkan perubahan pada suatu lingkungan yang terkendali sebagian atau sepenuhnya dikendalikan. Hasil akhir dari penelitian action adalah untuk menciptakan situasi, melakukan perubahan dan memonitoring hasil dari perubahan tersebut.

3.2.2. Jenis dan Metode Pengumpulan Data

Jenis data yang digunakan dalam penelitian ini adalah data kualitatif (data yang tidak dinyatakan dalam bentuk angka) dan data kuantitatif (data yang dinyatakan dalam bentuk angka). Data kualitatif diambil dari penjelasan atau keterangan baik secara lisan maupun tulisan. Data kuantitaif diambil dari survei yang dilakukan pada penyedia jasa kesehatan di bandung.

1) Data subjek, yang dibutuhkan dalam penulisan ini berupa opini dan tanggapan yang didapat oleh penulis dari hasil wawancara.

2) Data dokumenter, dalam penelitian ini berupa dokumen. Misalnya sejarah Dinas/Instansi, struktur organisasi, SOP yang berjalan, dan contoh rekam medis saat ini.

3.2.2.1. Sumber Data Primer

a. Wawancara

Teknik pengumpulan data dan informasi dengan melakukan tanya jawab dengan bagian-bagian yang terkait dengan objek penelitian yaitu divisi dari suatu instansi penyedia jasa kesehatan yang menangani rekam medis.

Menurut Guritno, Suryo et al (2011:33) Wawancara adalah suatu cara pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Ada beberapa faktor yang akan mempengaruhi arus informasi dalam wawancara, yaitu pewawancara, responden, pedoman wawancara dan situasi wawancara

b. Observasi

Yaitu pengumpulan data melalui pengamatan secara langsung terhadap pelaksanaan rekam medis yang dilakukan oleh suatu instansi penyedia jasa kesehatan. Menurut Guritno, Suryo et al (2011:64) observasi adalah melakukan pengamatan secara langsung ke objek penelitian untuk melihat dari dekat kegiatan yang dilakukan. Pengamatan merupakan suatu teknik atau cara mengumpulkan data dengan jalan mengadakan terjun langsung saat kegiatan sedang berlangsung.

3.2.2.2. Sumber Data Sekunder

Sumber data ini berasalkan dokumen dokumen yang dimiliki oleh poliklinik PT.Len Industri dan juga bersedia diserahkan untuk kepentingan penelitian. Data yang digunakan sebagai sumber data sekunder didapatkan berdasarkan teknik dokumentasi. Teknik dokumentasi adalah teknik pengumpulan data dengan cara mengumpulkan dokumen- dokumen yang terkait dengan objek penelitian. Kemudian dilakukan analisis terhadap dokumen dokumen tersebut sehingga data - data yang dibutuhkan sistem dapat diperoleh.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Metode pendekatan dan pengembangan sistem berguna untuk memenuhi kebutuhan pengembangan sistem sehingga sistem yang nanti nya dihasilkan dapat sesuai dengan kebutuhan.

3.2.3.1. Metode Pendekatan Sistem

Metode beorientasi objek mulai berkembang sekitar tahu 1980an ketika Grady Booch mempublikasikan papernya mengenai bagaimana melakukan perancangan untuk bahasa ADA namun memberi judul paper tersebut dengan *Object-Oriented Desig*n. Dari situ pengembangan berbasis objek terus berkembang hingga sekarang ini.Pada tahun 1991 Peter Coad dan Yourdon memperkenalkan metode berorientasi objek yang lebih sederhana dibandingkan Booch.Metode ini menjadi cepat populer karena mendukung layanan-layanan yang terdapat pada C++. Pada waktu itu C++ merupakan bahasa pemrograman berorientasi objek yang paling populer.

Pada tahun 1991 *Object Modelling Technique (OMT)* dikenalkan oleh Rumbaugh.Pendekatan yang digunakan tidak jauh berbeda dengan pendekatan yang digunakan Coad Yourdon namun dengan notasi yang berbeda. OMT tidak hanya sepenuhnya berbasis pada *data driven* tapi juga memisahkan proses dari data dengan penggunan *data flow diagram* yang terpisah dengan diagram kelas. OMT juga menggunakan notasi *state transition diagram* untuk memodelkan aspek dinamis sistem. Baru pada tahun 1994 muncul use case dan object oriented software engineering oleh Ivar Jacobson.

Pada tahun 1994 itu juga yaitu bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, mempelopori usaha untuk penyatuan notasi pendekatan berorientasi objek. Pada tahun 1995 dihasilkan *draft* pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – http://www.omg.org).Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003.

Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999.Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

Pengembangan menggunakan metode ini menitik beratkan kepada object, dengan melihat suatu sistem terdiri dari objek yang saling berhubungan. Setiap objek mempunyai atribut dan metode masing-masing.Tahap-tahap metodologi berdasarkan *Sistem Development Life Cycle* (*SDLC*) digunakan dengan memperhatikan karakteristik khusus berorientasi objek Sutopo (2002:35) yaitu Analisis, Desain dan Implementasi.

Metodologi Perancangan Berorientasi Objek

a. Encapsulation

Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

b. Inheritance

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya.Inheritance mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di anatara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya. Kelas Objek dapat didefinisikan

atribut dan service dari kelas Objek lainnya.Inheritance menggambarkan generalisasi sebuah kelas.

c. Polymorphism

Polimorfisme yaitu konsep yang menyatakan bahwa seuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama. Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Objek.

Adapun *The Object Management Group (OMG)* yaitu suatu konsorsium pengembang perangkat lunak menguraikan ciri-ciri utama kunci analisis dan perancangan berorientasi objek yang membedakan dari metodologi lainnya yaitu;

1. *Abstraksi* yaitu mendifinisikan suatu keterhubungan diantara suatu kelompok tipe objek dimana objek tersebut merepresentasikan suatu rangkaian karakteristik yang dipakai bersama dengan tipe objek lain.

2. *Enkapsulasi* yaitu mengimplikasikan pengemasan operasi dan data secara bersama-sama pada suatu tipe objek dimana hanya data tersebut yang dapat diakses melalui antarmukanya.

3. *Reuse* atau penggunaan ulang yaitu suatu kemampuan untuk menggunakan ulang tipe objek selama perancangan suatu sistem dan kelas-kelas objek dalam suatu implementasi suatu sistem.

4. Spesialisasi yaitu ketika suatu objek menurunkan operasi, tipe atribut dan tipe keterhubungan dari satu atau lebih supertipe.
5. Komunikasi objek yaitu dalam sistem orientasi objek, menggunakan bentuk dari satu objek mengirim permintaan ke objek lain.
6. Polymorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku yang berbeda. Polymorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

Kelebihan

1. Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalampembangunan sistem
2. Dibandingkan dengan SSAD, waktu pengembangan, level organisasi, ketangguhan,dan penggunaan kembali (reuse) kode program lebih tinggi dibandingkan dengan metode OOAD Sommerville(2003:69).
3. Tidak ada pemisahan antara fase desain dan analisis, sehingga meningkatkan komunikasi antara user dan developer dari awal hingga akhir pembangunan sistem.
4. Analis dan programmer tidak dibatasi dengan batasan implementasi sistem, jadi desain dapat diformliasikan yang dapat dikonfirmasi dengan berbagai lingkungan eksekusi.
5. Relasi obyek dengan entitas (thing) umumnya dapat di mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam sistem. Hal ini memudahkan dalam mehami desain Sommerville (2003:72).

6. Memungkinkan adanya perubahan dan kepercayaan diri yang tinggi terhadap kebernaran software yang membantu untuk mengurangi resiko pada pembangunan sistem yang kompleks Booch (2005:44).

7. *Encapsuliation data* dan *method*, memungkinkan penggunaan kembali pada proyek lain, hal ini akan memperingan proses desain, pemrograman dan reduksi harga.

8. OOP memungkinkan adanya standarisasi obyek yang akan memudahkan memahami desain dan mengurangi resiko pelaksanaan proyek.

9. Dekomposisi obyek, memungkinkan seorang analis untuk memcah masalah menjadi pecahan-pecahan masalah dan bagian-bagian yang dimanage secara terpisah. Kode program dapat dikerjakan bersama-sama. Metode ini memungkinkan pembangunan software dengan cepat, sehingga dapat segera masuk ke pasaran dan kompetitif. Sistem yang dihasilkan sangat fleksibel dan mudah dalam pemeliharaan.

3.2.3.2. Metode Pengembangan Sistem

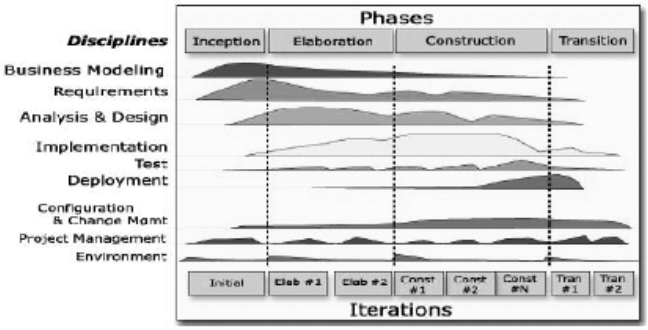
Rational Unified Process (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *usecase driven* dan pendekatan iteratif untuk siklus pengembangan perankat lunak.Gambar dibawah menunjukkan secara keseluruhan arsitektur yang dimiliki RUP.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

Dimensi pertama digambarkan secara horizontal.Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*.

Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Management, Project Management, Environtment.*



Gambar 3.2 Arsitektur RationalUnified Process
Sumber : Wikipedia.id.

Pada penggunaan kedua standar tersebut diatas yang beorientasi obyek (*object oriented*) memiliki manfaat yakni:

1. Improve Productivity

Standar ini dapat memanfaatkan kembali komponen – komponen yang telah tersedia/dibuat sehingga dapat meningkatkan produktifitas.

2. Deliver High Quality System

Kualitas sistem informasi dapat ditingkatkan sebagai sistem yang dibuat pada komponen – komponen yang telah diuji *(well-tested dan well-proven)* sehingga dapat mempercepat *delivery* sistem informasi yang dibuat dengan kualitas yang tinggi.

3. Lower Maintenance Cost

Standar ini dapat membantu untuk menyakinkan dampak perubahan yang terlokalisasi dan masalah dapat dengan mudah terdeteksi sehingga hasilnya biaya pemeliharaan dapat dioptimalkan atau lebih rendah dengan pengembangan informasi tanpa standar yang jelas.

4. Facilitate Reuse

Standar ini memiliki kemampuan yang mengembangkan komponen – komponen yang dapat digunakan kembali untuk pengembangan aplikasi yang lainnya

5. Manage Complexity

Standar ini mudah untuk mengatur dan memonitor semua proses dari semua tahapan yang ada sehingga suatu pengembangan sistem informasi yang amat kompleks dapat dilakukan dengan aman dan sesuai dengan harapan semua manajer proyek IT/IS yakni *deliver good quality software within cost and schedule time and the users accepted.*

Fase RUP

1. Inception/insepsi

    a. Menentukan Ruang lingkup proyek

    b. Membuat ‘Business Case’

c. Menjawab pertanyaan “apakah yang dikerjakan dapat menciptakan ‘good business sense’ sehingga proyek dapat dilanjutkan

2. Elaboration/elaborasi

a. Menganalisa berbagai persyaratan dan resiko

b. Menetapkan ‘base line’

c. Merencanakan fase berikutnya yaitu construction

3. Construction/konstruksi

a. Melakukan sederetan iterasi

b. Pada setiap iterasi akan melibatkan proses berikut: analisa desain,implementasi dan testing

4. Transition/transisi

a. Membuat apa yang sudah dimodelkan menjadi suatu produk jadi

b. Dalam fase ini dilakukan:

Beta dan performance testing

a. Membuat dokumentasi tambahan seperti; training, user guides dan sales kit

b. Membuat rencana peluncuran produk ke komunitas pengguna

Peran Use Case Pada Setiap Fase

1. Inception

a. Menolong mengembangkan scope proyek

b. Menolong menetapkan penjadwalan dan anggaran

2. Elaboration

a) Menolong dalam melakukan analisa resiko

b) Menolong mempersiapkan fase berikutnya yaitu konstruksi

3. Construction

a) Melakukan sederetan iterasi

b) Pada setiap iterasi akan akan melibatkan proses berikut: analisa desain, implementasi dan testing

4. Transistion

a) Membuat apa yang sudah dimodelkan menjadi suatu produk jadi

b) Dalam fase ini dilakukan:

a) Beta dan performance testing

b) Membuat dokumentasi tambahan seperti; training, user guides dan sales kit

c) Membuat rencana peluncuran produk ke komunitas pengguna

Penerapan Tahapan Metodologi Pengembagan Perangkat Lunak dengan Menggunakan RUP (Contoh Kasus)

Metodologi *Rational Unified Process* (RUP). Metode RUP merupakan metode pengembangan kegiatan yang berorientasi pada proses. Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak yaitu:

1. Inception

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*).Pada akhir fase ini, prototipe perangkat lunak versi *Alpha* harus sudah dirilis.

2. Elaboration

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilisan prototipe versi *Betha* dari perangkat lunak.

3. Construction

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini.Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

4. Transition

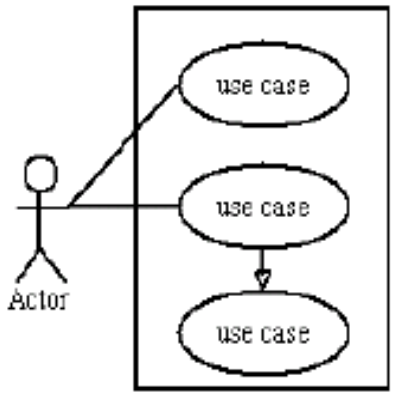
Instalasi ,*deployment* dan sosialisasi perangkat lunak dilakukan pada tahap ini.

3.2.3.3. Alat Bantu Analisis dan Perancangan

1.Use Case Diagram

a. Definisi Usecase Diagram

Usecase diagram adalah model fungsional sebuah sistem yang menggunakan *actor* dan *usecase*. *Usecase* adalah layanan (*services)* atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya Henderi et al (2008:36). *Usecase* adalah suatupola atau gambaran yang menunjukan kelakukan atau kebiasaan sistem. Setiap *Usecase* adalah suatu urut-urutan (*sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah *actor* dan sistem dalam bentuk sebuah dialog Henderi (2008:39). *Usecase* Diagram dibuat untuk memvisualisasikan/ menggambarkan hubungan antara *Actor* dan *Usecase*. *Usecase* diagram mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.



Gambar 3.3.Simbol dan Notasi Dasar Use Case Diagram
Sumber : Diktat Kuliah Uml Stimik Amikom

b. Sistem

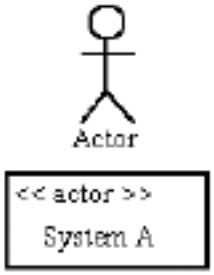
Gambar batasan *(boundries)* sebuah sistem menggunakan empat persegi panjang yang berisi usecase - usecase. Tempatkan actor-actor yang terlibat pada setiap usecase pada bagian luar *boundaries* sistem



Gambar 3.4. Use Case
Sumber : Diktat Kuliah Uml Stimik Amikom

c. Usecase

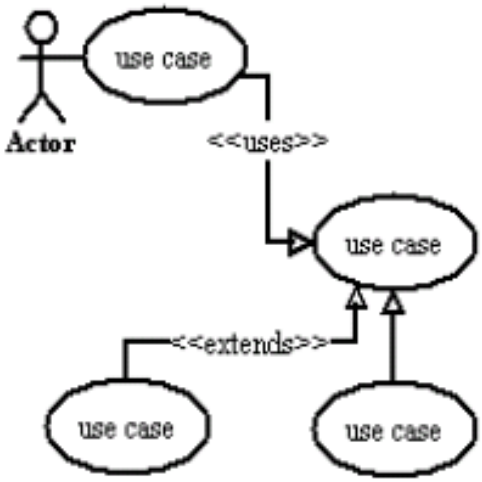
Gambar usecases menggunakan lingkaran berbentuk bulat telur *(ovals)* Beri nama *ovals* tersebut dengan kata kerja *(verbs)* yang menggambarkan fungsi –fungsi sistem.



Gambar 3.5.Actor
Sumber : Diktat Kuliah Uml Stimik Amikom

d. Actors

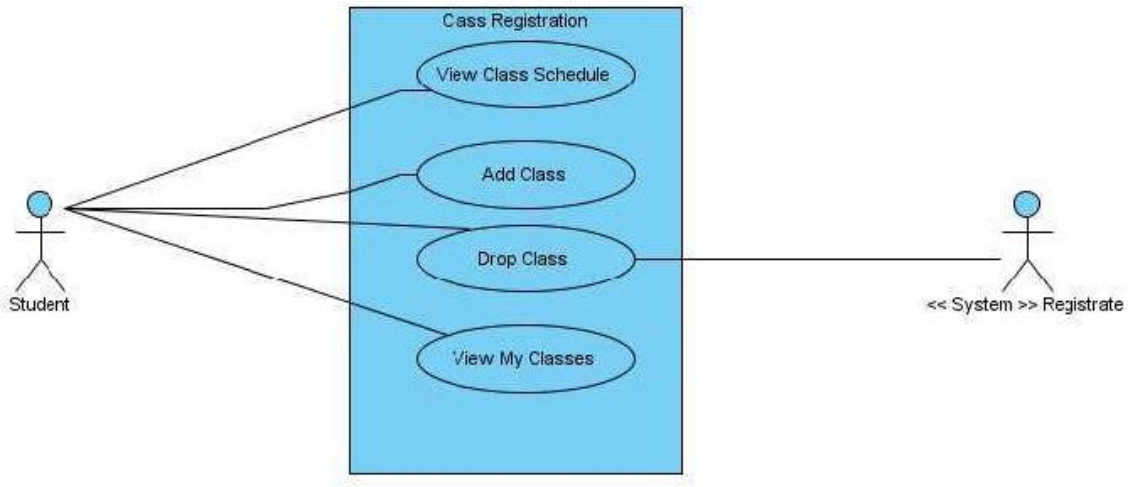
Actors adalah para pengguna *(users)* dari sebuah sistem. Kadangkala sebuah sistem adalah merupakan *actors* bagi sistem yang lain, beri nama *actors* sistem tersebut dengan *streotipe* (bentuk klise/tiruan) *actor* (gambar 5.3). *Actor* adalah seseorang atau sesuatu yang harus berinteraksi dengan sistem atau sistem yang dibangun/dikembangkan.



Gambar 3.6.Relationship
Sumber : Diktat Kuliah Uml Stimik Amikom

e. Relationships

Ilustrasikan atau gambaran relasi/hubungan antara sebuah *actor* dan *usecase* dengan sebuah garis sederhana. Untuk relasi-relasi antara *usecases*, gunakan tanda anak panah-anak panah pada *"uses"*yang lainnya atau gunakan *"extends"*. Suatu relasi*"uses"* mengindikasikan bahwa ada *usecase* yang dibutuhkan oleh *usecase* yang lain untuk melakukansebuah permintaan *(task).* Sementara suatu relasi*"extends"* mengindikasikan beberapa alternatif opsi(pilihan) tertentu pada tingkatan yang lebih bawah yang ada pada *usecase.*



Gambar 3.7.**Contoh Usecase Diagram**
Sumber : Diktat Kuliah Uml Stimik Amikom

Berdasarkan gambar 3.6*Use Case Diagram Class Registration* di atas diketahui bahwa sistem pendaftaran kelas mempunyai empat *use case* yang menunjukan fungsi sistem. Sementara itu actor atau para pengguna (*users*) yang terlibat pada sistem *class registration* initerdiri dari dua yaitu pelajar (*student*) dan sistem registrasi sebagai sistem yang lain.

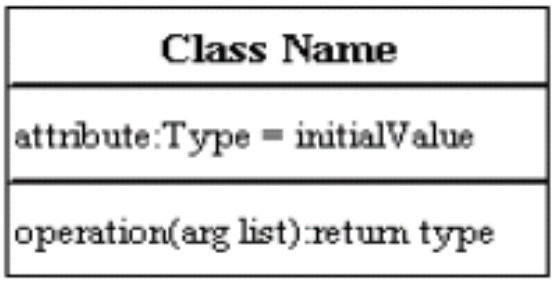
2. Class Diagram

a. Definisi Class Diagram

Class adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, nbehavior umum, relasi umum, dan semantic/kata yang umum. Class-class ditentukan/ditemukan dengan cara memeriksa objek-objek dalam sequence diagram dan collaboration diagram. Sebuah class digambarkan seperti sebuah bujur sangkar dengan tigabagian ruangan. Class sebaiknya diberi nama menggunakan kata benda sesuai dengan domain/bagian/kelompoknya Whitten L. Jeffery (2004:38).

Class Diagram adalah diagram yang menunjukan class-class yang ada dari sebuahsistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML Henderi (2008:89). Sementara menurut Whitten L. Jeffery et al (2004:432) class diagram adalah gambar grafis mengenaistruktur objek statis dari suatu sistem, menunjukan class-class objek yang menyusun sebuahsistem dan juga hubungan antara class objek tersebut.

Elemen-eleman class diagram dalam pemodelan UML terdiri dari: *Class-class*, struktur class, sifat class (class *behavior*), perkumpulan/gabungan (*association*),6pengumpulan/kesatuan (*agregation*), ketergantungan (d*ependency*), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan *role name* (peranan/tugas nama).



Gambar 3.8.Contoh Class Diagram
Sumber : Diktat Kuliah Uml Stimik Amikom

b. Classes

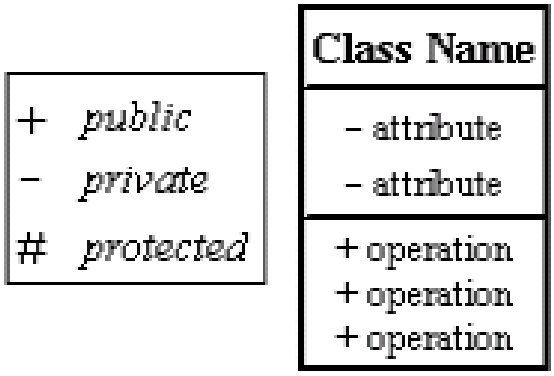
Ilustrasikan/gambarkan classes dengan bentuk empat persegi panjang yang dibagi kedalam ruang-ruang terpisah *(compartments).* Nama class ditempatkan pada bagian pertama (rata tengah, di-*bold*, dan Huruf besar), daftar atribut diletakan pada bagian kedua, dan tuliskan operasi-operasi pada *class* dibagian ketiga



Gambar 3.9.ContohActive Class
Sumber : Diktat Kuliah Uml Stimik Amikom

c. Active Classes

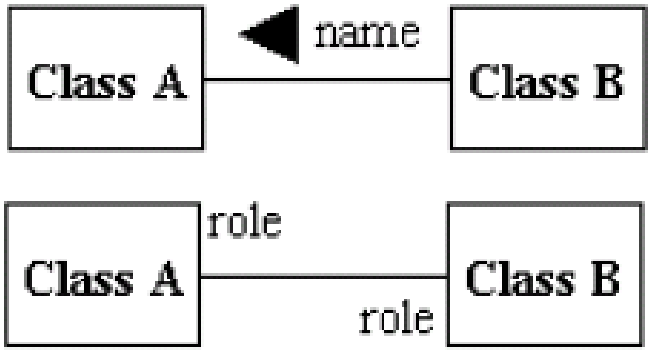
Active classes adalah class yang memulai dan mengontrol aliran/arus/arah aktifitas. Sementara *passive class* menyimpan data dan melayani *(menserve)* class yang lain. Gambarkan *active class* pada sebuah bolder tebal dibagian tengah.



Gambar 3.10.ContohVisibility
Sumber : Diktat Kuliah Uml Stimik Amikom

d. Visibility

Gunakan penilai *visibility* ("penglihatan") untuk menandakan siapa yang bisa mengakses informasi – informasi yang diisi kedalam sebuah *class*. *Private visibiliy* artinya informasi yang ada dalam sebuah *class* disembunyikan/dipartisi dari pihak luar. *Public visibility* mengijinkan semua *class* yang lainnya untuk melihat nilai informasi. *Protected visibility* mengijinkan *class-class* yang ada yang merupakan turunannya untuk mengakses informasi yang ada didalamnya karena mereka merupakan *class* turunan dari class induknya/*inherited*.

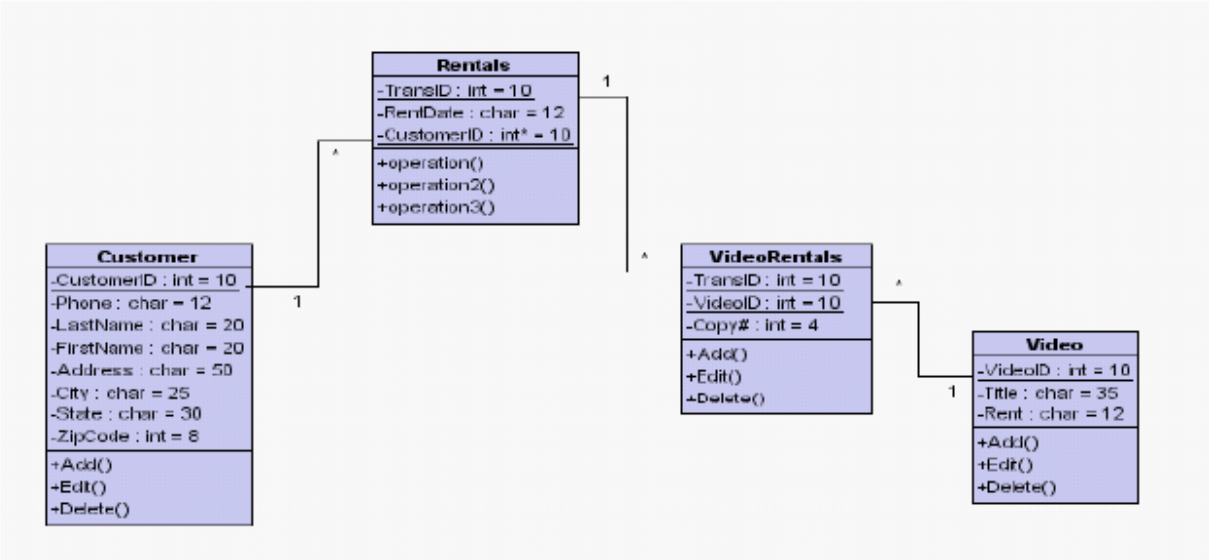


Gambar 3.11.Contoh Associations
Sumber : Diktat Kuliah Uml Stimik Amikom

e. Associations

Associations adalah representasi/gambaran relasi statis diantara *class-class*. Tempatkan nama *associations* pada bagian atas, di, atau dibawah garis *associations.* Gunakan tanda anak panah yang berisi sebuah kata yang mengindikasikan relasi secara langsung. Letakan *role* (aturan/ ketentuan) pada bagian akhir *associations.* Aturan merepresentasikan arah bagi kedua kelas untuk saling berhubungan satu sama lain (gambar 3.8). Pada umumnya sebuah ”nama” tidak digunakan untuk menunjukan *role* sebuah class.

Dalam *associations* terdapat *roles* (aturan/ketentuan) yang disebut dengan *multiplicity*. *Multiplicity* adalah jumlah kejadian minimum dan maksimum dari satu*object/class* untuk satu kejadian tunggal dari *object/class* yang terkait Whetten L. Jeffery et al(2004:414).



Gambar 3.12.ContohClass Diagram
Sumber : Diktat Kuliah Uml Stimik Amikom

3 Sequence Diagram

a. Definisi Sequence Diagram

Sequence Diagram adalah suatu diagram yang memperlihatkan/menampilkaninteraksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa ”pesan/*message*” Henderi (2008:84). Sequence diagram dapat juga didefinisikan sebagaisuatu diagram yang menggambarkan interaksi-interaksi yang ada antar *class* dalam suatu hubungan perubahan dengan sebuah pesan pada akhir waktu Miller Randy. *Sequence Diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan *output* tertentu. *Sequence Diagram* diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan Henderi (2008:88).

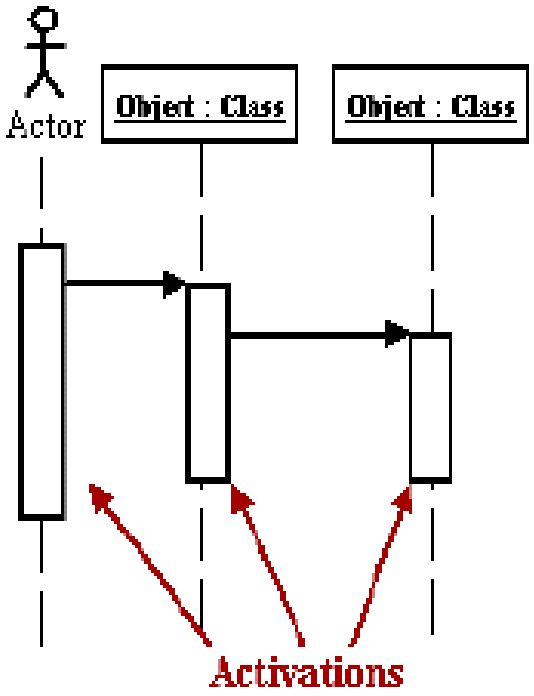


Gambar 3.13.Contoh Simbol dan Notasi Dasar Sequence Diagram

Sumber : Diktat Kuliah Uml Stimik Amikom

b. Class roles

Class roles menggambarkan *way* (jalan) sebuah objek akan menunjukan reaksi (berkelakuan) dalam sebuah keadaan (konteks). Gunakan simbol object UML untuk menggambarkan aturan-aturan *class*, tetapi bukan daftar attribute-atribute objek

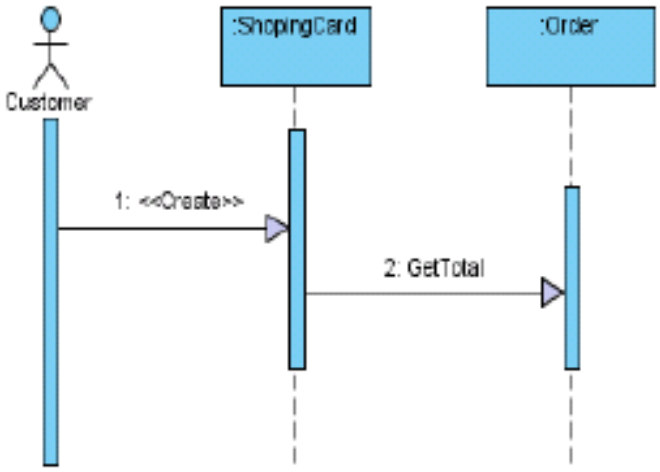


Gambar 3.14.Contoh Activations

Sumber : Diktat Kuliah Uml Stimik Amikom

c. Activation

Activation boxes merepresentasikan waktu yangdibutuhkan oleh sebuah objek untuk melaksanakan sebuah tugas/perintah secaralengkap.

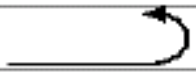
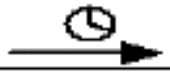


Gambar 3.15.ContohMessage

Sumber : Diktat Kuliah Uml Stimik Amikom

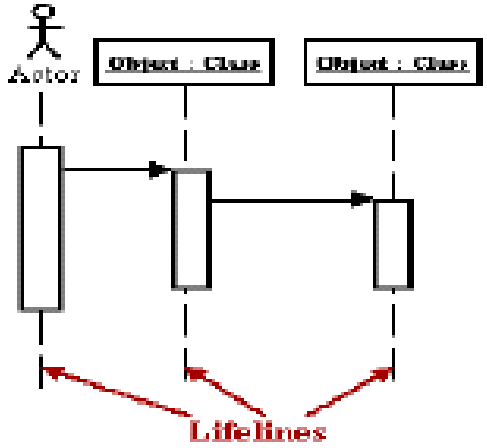
d. Messages

Message adalah anak panah yang merepresentasikan komunikasi antara objek.*Message* berguna untuk mengirimkan perintah kepada *LifeLine*. *Message* bisa berasal dari *actor* kepada *LifeLine* atau dari *LifeLine* kepada *LifeLine* yang lain. Gunakan setengah garis anak panah untuk merepresentasikan pesan-pesan *asyncronous*. Pesan *asyncronous* dikirim dari sebuah objek yang tidak akan menunggu respon dari penerima sebelum melanjutkan perintahnya

Arrow	Message type
	Simple
	Synchronous
	Asynchronous
	Balking
	Time out

Gambar 3.16.Contoh Simbol dan Notasi Dasar Message

Sumber : Diktat Kuliah Uml Stimik Amikom

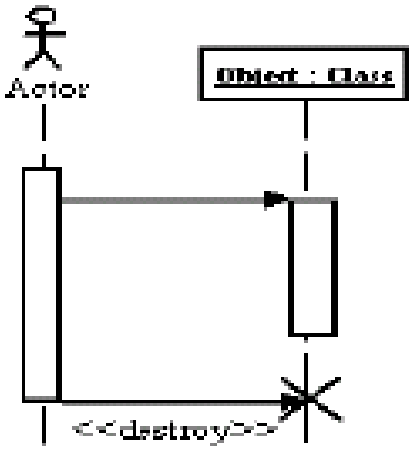


Gambar 3.17.ContohLifelines

Sumber : Diktat Kuliah Uml Stimik Amikom

e. Lifelines

Lifeline adalah tanda garis pisah yang mengindikasikan kehadiran objek pada saat terakhir/akhir waktu.

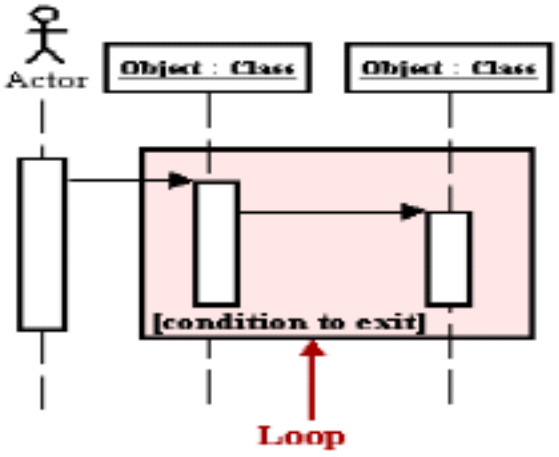


Gambar 3.18.Contoh Destroying Objects

Sumber : Diktat Kuliah Uml Stimik Amikom

f. Destroying Objects

Objek dapat di-akhiri secara lebih cepat menggunakan sebuah garis anah panah yang diberi label *"<<destroy>>>"* dan diujungnya diberi kesebuah label X.

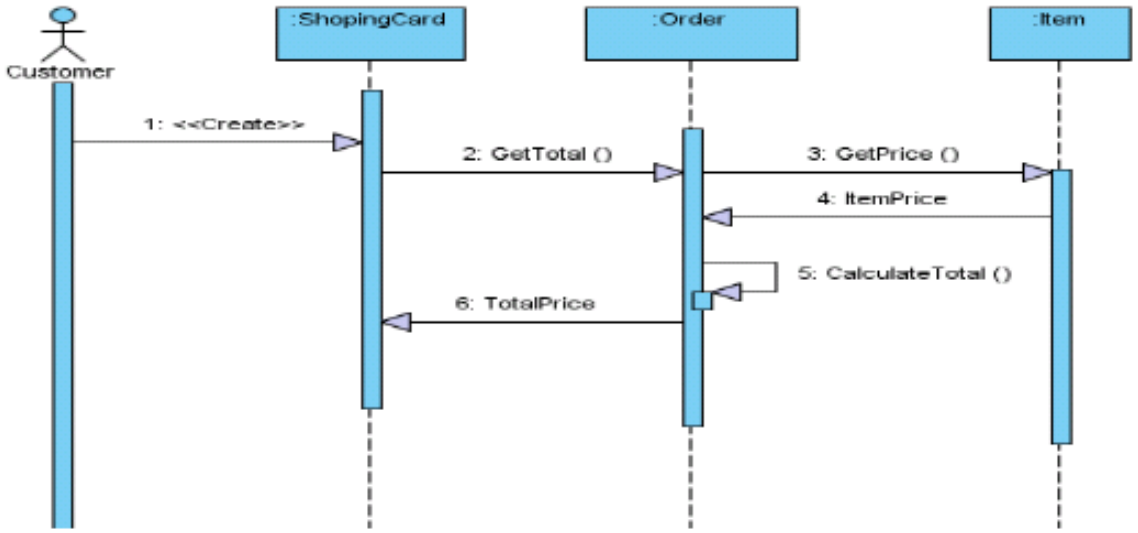


Gambar 3.19.ContohLoop

Sumber : Diktat Kuliah Uml Stimik Amikom

g. Loops

Suatu pengulangan atau *loop* dalam sebuah*sequence diagram* digambarkan seperti sebuah empat persegi panjang. Letakan kondisi untuk keluar pada pengulangan di bagian kiri bawah di dalam dalam kurang empat persegi panjang.



Gambar 3.20.Contoh Contoh Sequence Diagram

Sumber : Diktat Kuliah Uml Stimik Amikom

Berdasarkan gambar 3.17 diatas diketahui bahwa *sequence diagram shoping card* melibatkan *actor customer* dan tiga *object class* yaitu *shoping card*, order dan *item*. *Sequence diagram* ini diawali dengan kegiatan *Acto*r mengirimkan perintah *(message)* kepada *LifeLine Shoping Card*, dan selanjutnya *LifeLine* mengirimkan message kepada *LifeLine* yang lainnya, termasuk diantaranya ada satu *LifeLine* yang ditujukan kepada *LifeLine* itu *sendiri (message to self)* yang terjadi pada *LifeLine Order*.

4. Activity Diagram

a. Definisi Activity Diagram

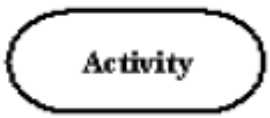
Activity diagram adalah diagram yang menggambarkan sifat dinamis secara alamiahsebuah sistem dalam bentuk model aliran dan kontrol dari aktivitas ke aktivitas lainnya. Sebuah aktivitas merepresentasikan suatu operasi pada beberapa class dalam sistem yang menghasilkan suatu perubahan keadaan *(state)* dari sistem tersebut. Secara khusus, *activity diagram* biasa digunakan untuk memodelkan diagram alir sebuah sistem kerja (*workflow*) atau proses bisnis (prosedur bisnis) dan operasi-operasi secara internal.Sementara menurut Whitten L. Jeffery el al (2004: 428), *activity diagram* adalah sebuah diagram yang dapat digunakan untuk menggambarkan secara grafisaliran proses bisnis, langkah-langkah sebuah *use case* atau logika *behavior* (metode) object. Karena sebuah *activity diagra*m adalah bentuk khusus dari *statechart diagram*, maka *activity diagram* kadang kala digunakan untuk memodelkan suatu kebiasaan sesuai dengan ketentuan/kaedah bisnis. Untuk dapat membangun activity diagram yang baik, berikut proses yang unggul untuk membangun *activity diagram*:

1. Tambahkan poin awal dan akhir pada sebuah usecase

2. Tambahkan sebuah kegiatan untuk tiap langkah utama pada use case (atau tiap langkah utama setiap pelaku yang menginisialisasi)

3. Tambahkan transisi dari setiap kegiatan ke kegiatan lain, poin keputusan, atau poin akhir

5. Tambahkan bar sinkronisasi di mana kegiatan dilakukan secara paralel.

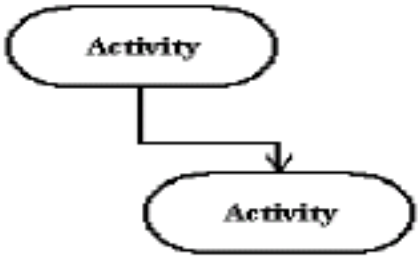


Gambar 3.21.ContohActivity

Sumber : Diktat Kuliah Uml Stimik Amikom

a. Action states

Action state adalah representasi/gambaran dari aksi yang tidak bisa diganggu oleh aksi yang berasal dari objek-objek. *Action state* digambarkan dalam bentuk empat persegi panjang yang pada sudut-sudutnya melingkar.

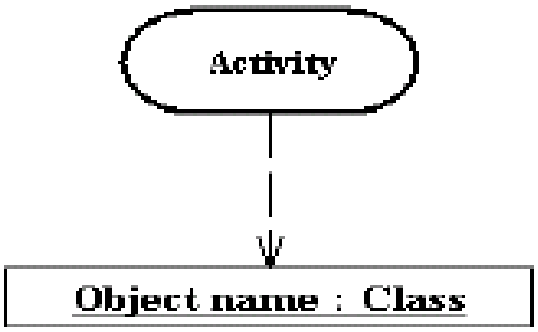


Gambar 3.22.Contoh Action States

Sumber : Diktat Kuliah Uml Stimik Amikom

b. Action Flow

Action digambarkan dalam bentuk anak panah yang mengilustrasikan relasi antara *action* pada *state.*



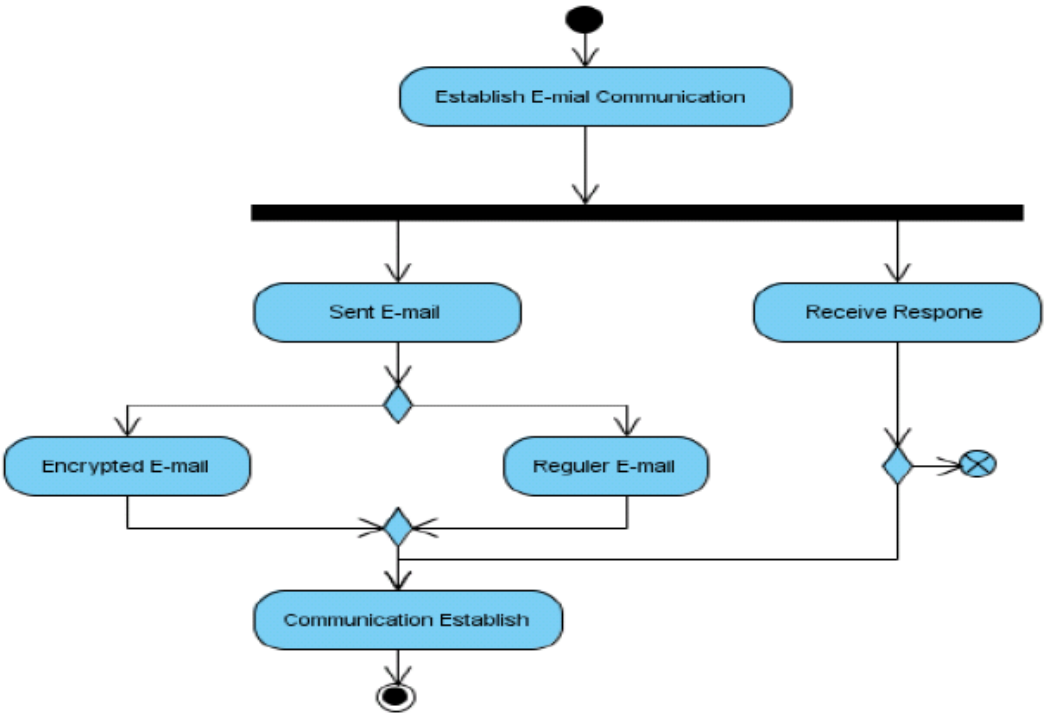
Gambar 3.23.**ContohAction Flow**

Sumber : Diktat Kuliah Uml Stimik Amikom

c. Object Flow

Object flow menunjuk kepada kegiatan penciptaan *(create)*dan memodifikasi oleh objek melalui *activities*. Sebuah tanda panah objek *flow* dari suatu aksi kepada suatu objek berarti bahwa aksi tersebut meng-*create* atau mempengaruhi objek tersebut. Sementara suatu tanda panahdari objek kepada suatu aksi mengindikasikan bahwa aksi *state* tersebut menggunakan objek.

Contoh Activity Diagram



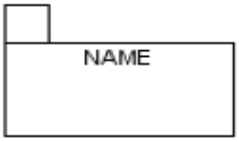
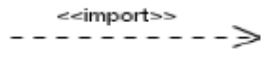
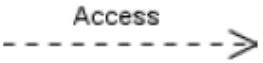
Gambar 3.24.ContohActivity Diagram

Sumber : Diktat Kuliah Uml Stimik Amikom

5. Package Diagram

a. Definisi Package Diagram

Package (paket) adalah sekelompok elemen-elemen model. Sebuah paket dapat berisi elemen-elemen model yang berlainan, termasuk paket-paket untuk menciptakan/menggambarkan sifat hirarki. Sebuah paket diberi nama yang menggambarkan isinya. Karena itu, *package* dapat digunakan untuk beberapa keperluan/kebutuhan. Berikut ini adalah simbol dan konsep dasar *package diagram.*

Contruct	Deskripsi	Lambang
Package	Sekelompok elemen-elemen model	
Import	Suatu dependency yang mengindikasikan isi tujuan paket secara umum yang ditambahkan kedalam sumber paket	
Access	Suatu dependency yang mengindikasikan isi tujuan paket secara umum yang bisa digunakan pada nama sumber paket	

Gambar 3.25.Jenis Gambar Activity Diagram
Sumber : Diktat Kuliah Uml Stimik Amikom

6. **Diagram kolaborasi/Collaboration Diagram:** serupa dengan diagram rangkaian/sekuensi, tetapi tidak fokus pada timing atau sekuensi pesan. Diagram ini justru menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan.

7. ***Diagram penguraian/Deployment:*** digunakan untuk mendeskripsikan arsitektur fisik dalam istilah ”node” untuk hardware dan software dalam sistem. Diagram ini menggambarkan konfigurasi komponen-komponen software *real-time*, prosesor, dan peralatan yang membentuk arsitektur sistem.

3.2.4. Pengujian Software

Metode pengujian sistem adalah beberapa *test case* harus dilaksanakan dengan beberapa perbedaaan strategi transaksi, query atau jalur navigasi yangmewakili penggunaan sistem yang tipikal, kritis atau abnormal. Isu kuncinya adalah pada pemilihan sekelompok *test case* yang cocok.

a. Black Box testing

Terfokus pada apakah unit program memenuhi kebutuhan *(requirement)* yang disebutkan dalam spesifikasi. Pada black box testing, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan. Jika ada yang tidak sesuai dengan outputnya maka untuk menyelesaikannnya diteruskan pada *white box testing*.

3.3 Analisis Sistem yang Sedang Berjalan

Analisis Sistem yang berjalan pada Poliklinik PT.Len Industri (Persero) ini bertujuan untuk mengetahui secara detail mengenai cara kerja dan proses bisnis dari sistem dan masalah yang dihadapi sistem agar dapat dijadikan landasan usulan perancangan analisis sistem yang sedang berjalan yang berdasarkan urutan proses kerja yang ada pada Poliklinik PT.Len Industri (Persero).

3.3.1 Analisis Prosedur yang Sedang Berjalan

Analisis prosedur yang sedang berjalan menguraikan secara sistematis aktifitas - aktifitas yang terjadi dalam proses bisnis Poliklinik PTLen Industri (Persero). Untuk menentukan kebutuhan dari pengguna yaitu dengan cara menganalisis sistem yang sedang berjalan, kemudian menguraikannya menjadi suatu bentuk kebutuhan-kebutuhan. Adapun hasil dari proses analisis sistem informasi pengelolaan data obat yang sedang berjalan adalah sebagai berikut :

1. **Proses Pengelolaan pendaftaran pasien baru** :

Proses Pengelolaan pendaftaran pasien baru dilaksanakan ketika ada pasien yang datang untuk berobat ke poliklinik, tetapi pasien tersebut belum pernah datang sebelumnya atau

merupakan pasien baru. Pada proses ini dimulai dengan pasien memberikan data – datanya kepada resepsionis untuk kemudian dibuatkan kartu berobat.

2. Proses Pemeriksaan Pasien:

Proses Pemeriksaan Pasien dilaksanakan ketika pasien yang sudah terdaftar didalam poliklinik datang dan ingin berobat ke poliklinik. Pasien tersebut akan langsung menuju ke assisten dokter untuk mendapatkan nomor antrian. Pada akhir proses ini dokter melakukan diagnosa atas keluhan-keluhan pasien dan memberikan diagnosa nya mengenai penyakit yang mungkin dialami oleh pasien.

3. Proses Pengelolaan Data Resep :

Proses Pengelolaan Data Resep, proses ini hanya bisa dilakukan setelah dokter melakukan proses diagnosa dokter. Pada proses ini dokter menuliskan obat-obatan yang harus dikonsumsi dan dibeli oleh pasien dalam jangka waktu tertentu di apotik.

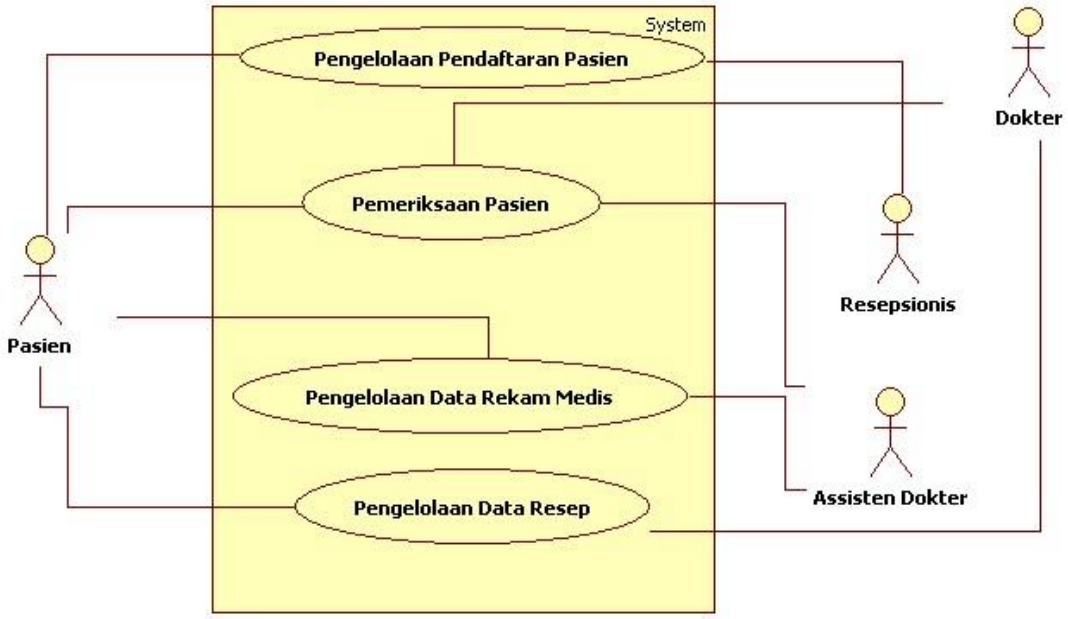
4. Proses Pengelolaan Data Rekam Medis :

Proses Pengelolaan Data Rekam Medis pada proses ini assisten dokter melakukan pengadministrasian dan pengelolaaan arsip rekam medis.

3.3.1.1 *Usecase Diagram*

Use Case merupakan perilaku *software* aplikasi dimana proses tersebut menggambarkan suatu sistem. Sehingga yang mengunakan sistem akan mudah mengerti mengenai kegunaan sistem yang dibangun.

Usecase diagram menggambarkan siapa saja aktor yang melakukan prosedur dalam sistem serta fungsi – fungsi yang terlibat dalam transformasi. Berikut ini adalah *diagram use case* yang menggambarkan proses utama dari sistem yang sedang berjalan, yaitu :



Gambar 3.26 Usecase Diagram Sistem Berjalan

3.3.1.2.Definisi Aktor dan Deskripsinya

Aktor merupakan orang, proses atau *system* lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri Rosa AS-M.Shalahuddin (2011:131).Berikut ini adalah *actor* yang ada pada proses utama dari sistem yang sedang berjalan, yaitu :

Tabel 3.1

Aktor dan Deskripsinya

No	Aktor	Deskripsi
1	Dokter	Pihak yang melakukan diagnosa kepada pasien dan membukukan informasi rekam medis.
2	Assisten Dokter	Pihak yang mengurus segala hal mengenai administrasi dokter
3	Resepsionis	Pihak yang mengurus mengenai proses tunggu pemeriksaan
4	Pasien	Pihak yang melakukan pemeriksaan medis

3.3.1.3.Definisi Use Case dan Deskripsinya

Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Rosa A.S-M.Shalahuddin (2011:131). Berikut ini adalah *usecase* yang ada pada proses utama dari sistem yang sedang berjalan, yaitu :

Tabel 3.2

Use Case dan Deskripsinya

No	Usecase	Deskripsi
1	**Pengelolaan pendaftaran Pasien**	**Merupakan proses dimana calon pasien baru melakukan pendaftaran untuk menjadi pasien poliklinik dan mengisi data demografi untuk pertama kali.**
2	**Pengelolaan Data Resep**	**Pengelolaan data resep merupakan proses dimana dokter menuliskan obat-obatan yang harus dikonsumsi oleh pasien.**
3	**Pemeriksaan pasien**	**Pemeriksaan pasien merupakan proses diagnosa dokter dan penulisan diagnosa dan keluhan pasien ke dalam *form* rekam medis oleh dokter.**
4	**Pengelolaan data rekam medis**	**Pengelolaan data rekam medis merupakan proses pengadministrasian dan pendokumentasian data rekam medis pasien.**

3.3.1.4.Skenario Use Case

Skenario *use case* menjelaskan aktor – aktor yang melakukan proses di dalam sistem, dan juga mendefinisikan respon yang diberikan oleh sistem tersebut terhadap proses yang dilakukan oleh aktor. Berikut ini adalah *usecase* yang ada pada proses utama dari sistem yang sedang berjalan, yaitu :

Tabel 3.3

Skenario Use Case Pengelolaan Pendaftaran Pasien yang sedang berjalan

Identifikasi	
Nama Use Case	Pengelolaan Pendaftaran Pasien
Aktor	Pasien
Worker	Resepsionis
Tujuan	Melakukan pendaftaran calon pasien baru dan memulakukan pengelolaan data pasien.
Skenario	
Aksi Aktor	**Reaksi Sistem**
	1. Resepsionis memberikan info pendaftaran pasien baru dan meminta data pasien baru.
2. Calon pasien poliklinik memberikan data- data pribadinya untuk di isikan kedalam formulir pendaftaran.	
	3. Resepsionis melakukan pengisian data – data pasien ke dalam formulir pendaftaran dan meminta *fotocopy* ktp, serta kartu id card pegawai.
4. Calon pasien menyerahkan *fotocopy* ktp dan *id card pegawai*.	
	5. Resepsionis menerima *fotocopy* ktp

	serta id card pegawai dan menyatukannya dengan formulir pendaftaran untuk kemudian meminta pasien mengecek apakah data tersebut sudah benar.
6. Calon pasien mengecek data tersebut dan memberikan verifikasi apakah data tersebut sudah benar atau belum	
	7. Resepsionis membenarkan apabila ada data yang salah.
	8. Resepsionis menyerahkan kartu berobat pasien.
9. Calon pasien menerima kartu berobat tersebut dan resmi menjadi pasien poliklinik.	

Tabel 3.4

Skenario Use Case Pengelolaan Data Resep yang sedang berjalan

Identifikasi	
Nama Use Case	Pengelolaan Data Resep
Aktor	Pasien
Worker	Dokter
Tujuan	Melakukan penulisan resep berdasarkan diagnosa yang dilakukan sebelumnya.
Skenario	
Aksi Aktor	**Reaksi Sistem**
	1. Dokter menuliskan resep berdasarkan diagnosa yang dilakukan.
	2. Dokter menjelaskan cara melakukan konsumsi resep yang akan diberikan.
3. Pasien menerima resep yang diberikan oleh dokter.	

Tabel 3.5

Skenario Use Case Pemeriksaan Pasien yang sedang berjalan

Identifikasi	
Nama Use Case	Pemeriksaan Pasien
Aktor	Pasien
Worker	Dokter, Assisten Dokter
Tujuan	Melakukan proses pemeriksaan pasien dan menuliskannya kedalam *form* rekam medis pasien.
Skenario	
Aksi Aktor	**Reaksi Sistem**
1. Pasien menyatakan keluhan – keluhan dan gejala yang dirasakan oleh pasien dan menyerahkan *form* rekam medisnya..	
	2. Dokter bertanya lebih lanjut mengeai gejala dan keluhan yang diutarakan oleh pasien.
3. Pasien menjawab pertanyaan dokter.	
	4. Dokter memberikan diagnosa awal dan prosedur yang harus dilakukan pasien untuk menjalani pengobatan.
5. Pasien menerima saran dokter dan menyatakan kesediaan nya untuk menjalani pengobatan.	
	6. Dokter menuliskan keluhan dan diagnosa yang diberikannya kepada pasien lalu menuliskannya kedalam *form* rekam medis pasien.
7. Pasien menerima *form* rekam medis tersebut.	
8. Pasien menyerahkan *form* rekam medisnya kepada assisten dokter.	
	9. Asisten dokter menerima *form* rekam medis dari pasien.

Tabel 3.6

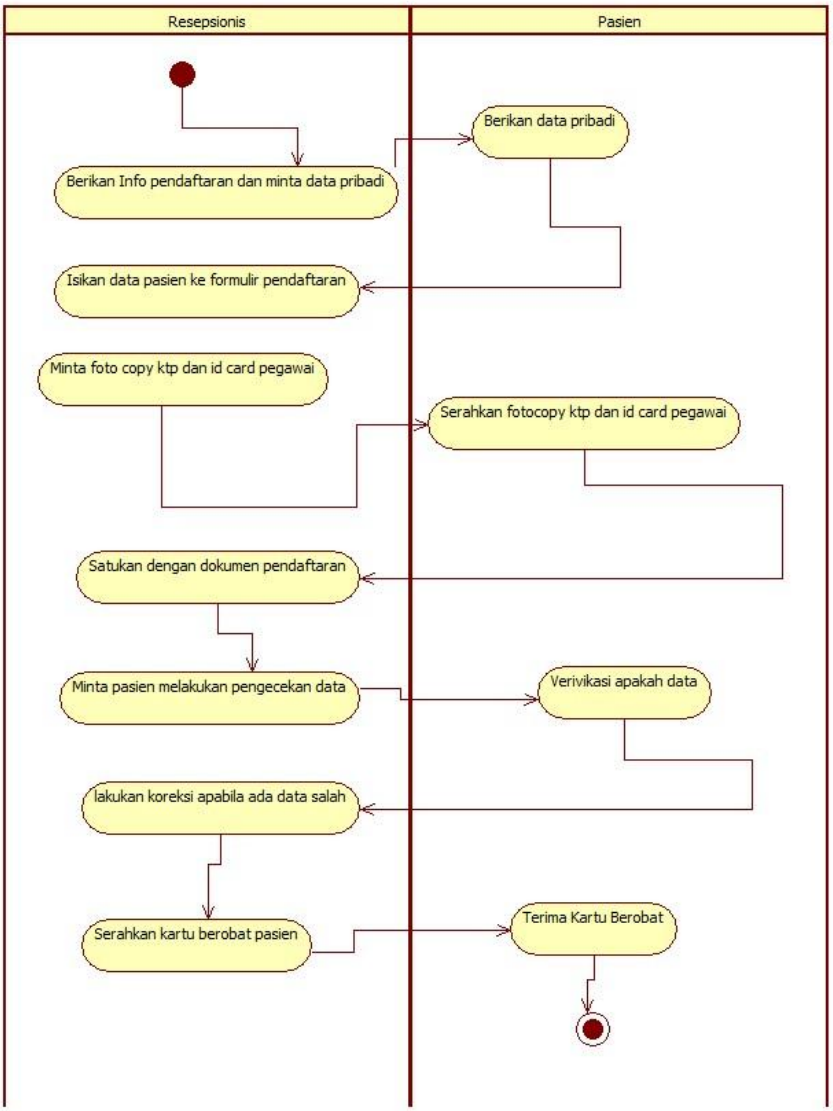
Skenario Use Case Pengelolaan data rekam medis yang sedang berjalan

Identifikasi	
Nama Use Case	Pengelolaan data rekam medis
Aktor	Pasien
Worker	Assisten Dokter

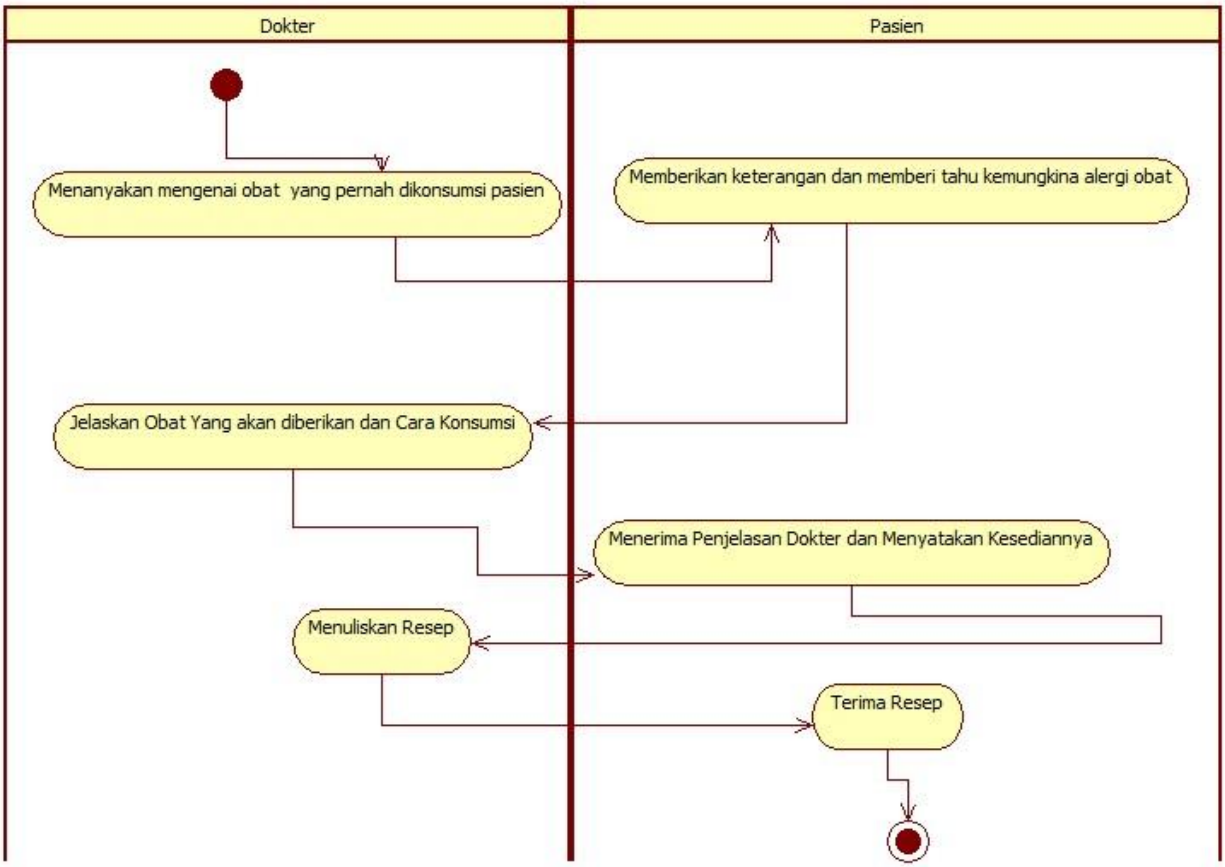
Tujuan	Melakukan proses adminitrasi pengelolaan rekam medis pasien.
Skenario	
Aksi Aktor	**Reaksi Sistem**
1. Pasien meminta arsip rekam medisnya untuk kebutuhan pemeriksaan	
	2. Apabila pasien tersebut baru akan diberikan *form* rekam medis baru yang masih kosong namun apabila pasien tersebut sudah memiliki data rekam medis maka akan diberikan data rekam medis yang sudah terisi.
3. Pasien mengembalikan data rekam medis yang digunakan untuk kebutuhan pemeriksaan setelah melakukan pemeriksaan.	
	4. Rekam medis pasien bersangkutan diarsipkan oleh assisten dokter.

3.3.1.5. Activity Diagram

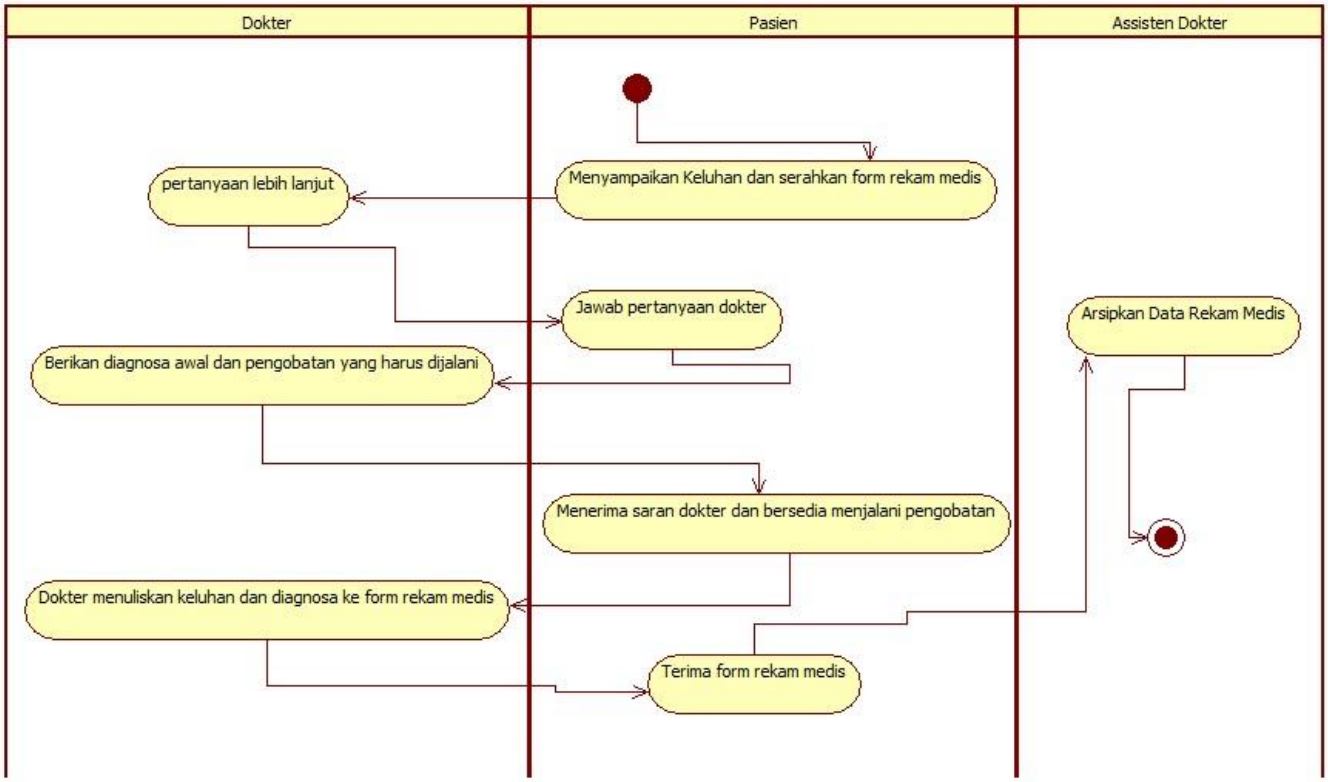
Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis, bukan apa yang dilakukan aktor Rosa A.S-M.Shalahuddin (2011:134). Berikut ini adalah *usecase* yang ada pada proses utama dari sistem yang sedang berjalan, yaitu :



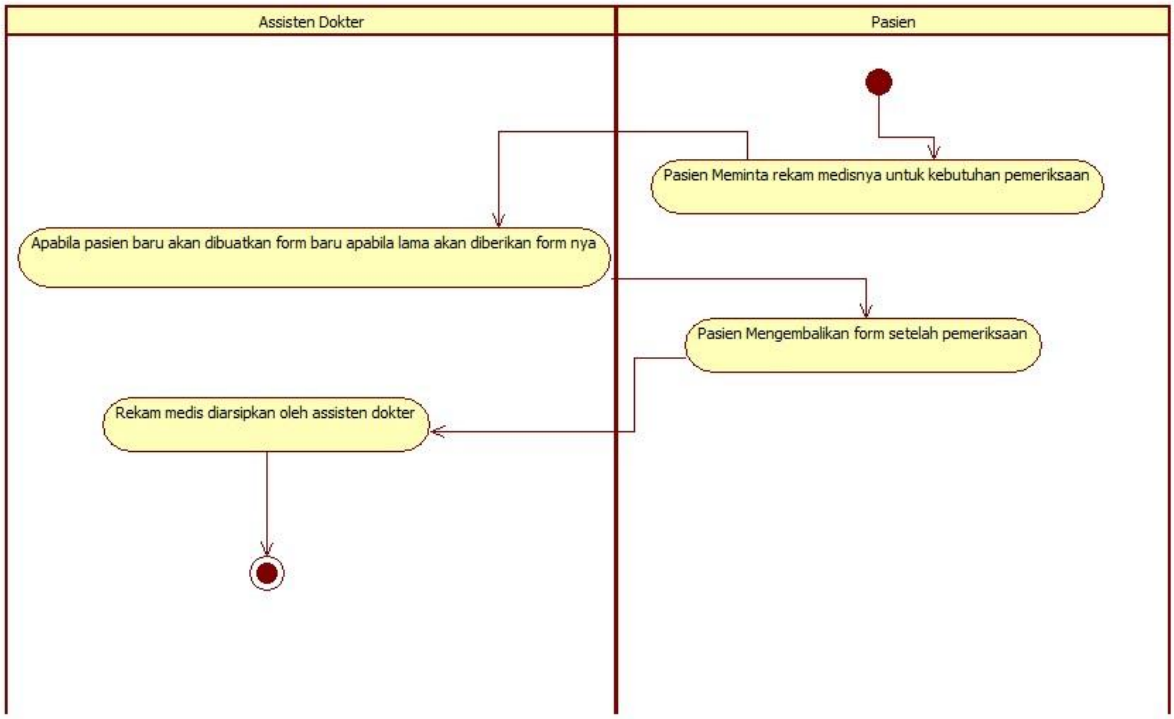
Gambar 3.27 Activity Diagram Pengelolaan Pendaftaran Pasien Yang Berjalan



Gambar 3.28 Activity Diagram Pengelolaan Data Resep Yang Berjalan



Gambar 3.29 Activity Diagram Pemeriksaan Pasien Yang Berjalan



Gambar 3.30 Activity Diagram Pengelolaan Data Rekam Medis Yang Berjalan

3.3.2. Evaluasi Sistem yang Sedang Berjalan

Melihat dari sistem yang sedang berjalan, bahwa kegiatan dari proses administrasi rekam medis,resep dan data demografi pasien masih memiliki banyak kekurangan antara lain sebagai berikut :

Tabel 3.7

Evaluasi Sistem yang sedang berjalan

No	Permasalahan	Aktor	Rencana Perancangan/ Solusi
1.	Dalam proses administrasi data pasien terutama pendaftaran pasien baru saat pengisian data pasien ke formulir pendaftaran masih manual dan tidak efisien.	Resepsionis *Activity-Pemeriksaan Pasien Berjalan*	Melakukan proses tersebut secara terkomputerisasi sehingga menjadi lebih cepat dan efisien.
2.	Sulitnya bagi dokter untuk membaca secara lengkap *form* rekam medis pasien untuk melakukan analisis dengan tepat dan cepat, karena harus membuka lembaran-lembaran *form* rekam medis pasien bersangkutan.	Dokter-*Activity Pemeriksaan Pasien*	Dengan adanya sistem informasi ini maka dokter mengoperasikan fungsi-fungsi yang ada di aplikasi melalui *mouse* dan *keyboard* sehingga menjadi lebih mudah
3.	Dalam Proses Pemeriksaan pasien masih harus membawa bawa *form* rekam medis nya dan	Pasien-*Activity Pemeriksaan Pasien*	Dengan adanya sistem informasi ini maka pasien tidak perlu membawa *form – form hardcopy*

	ini merupakan suatu hal yang mempersulit pasien		rekam medis nya.
4	Form rekam medis yang diterima oleh dokter diarsipkan langsung ke ruang arsip dan tidak adanya *backup*, ini akan membuat dalam penyajian kembali *form* rekam medis ini assisten dokter harus mencari *hardcopy* pasien bersangkutan dari ruang arsip dan hal tersebut memakan waktu.	Assisten Dokter-*Activity Pengelolaan Data Rekam Medis*	Dengan aplikasi ini assisten dokter akan memiliki *softcopy* rekam medis elektronik di dalam sistem sehiongga saat penyajian kembali akan lebih mudah. Kemudian apabila benar – benar diharuskan barulah assisten dokter mengambil *hardcopy* dari rekam medis pasien bersangkutan.
5	Apabila terjadi *force majeur* yang merusak dokumen hard copy rekam medis pasien seperti kecurian, kebakaran dan terkena air ataupun rusak nya kertas karena termakan oleh waktu maka assisten dokter akan kehilangan arsip berharga yang seharusnya menjadi tugas utama assisten dokter.	Assisten Dokter-*Activity Pemeriksaan Pasien*	Dengan adanya sistem informasi ini maka assisten dokter dapat memiliki backup dari rekam medis .

Evaluasi sistem ini dibuat untuk membantu meminimalisasi kekurangan dan kelemahan pada sistem yang sedang berjalan sehingga diharapkan dengan adanya perancangan sistem informasi rekam medis elektronik ini dapat memenuhi kebutuhan di Poliklinik PT.Len Industri (Persero).