

BAB II

LANDASAN TEORI

II.1. Sistem

II.1.1. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisan sistem, yaitu kelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan.

Teori sistem melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep atau bidang kajian ilmiah ini terutama berkaitan dengan upaya-upaya untuk menerapkan berbagai disiplin ilmu yaitu ilmu perilaku, fisika, biologi dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomatis tugas-tugas yang dilakukan oleh manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan dan lain sebagainya. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*inputan*), pengolahan (*processing*) dan keluaran (*output*). (Tata Sutabri, 2012 : 2-3)

II.1.2. Pengertian Sistem

Mempelajari suatu sistem akan lebih mengena bila mengetahui terlebih dahulu apakah sistem itu. Pengertian tentang sistem pertama kali dapat diperoleh

dari definisi sistem itu sendiri. Jika kita perhatikan dengan seksama, diri kita juga terdiri dari berbagai sistem yang berfungsi untuk mengantar kita pada tujuan hidup kita.

Gordon B. Davis dalam bukunya menyatakan bahwa sistem bisa berupa abstrak atau fisik. Sistem yang abstrak adalah susunan gagasan-gagasan atau konsepsi yang teratur yang saling bergantung. Misalnya, sistem teologi adalah susunan yang teratur dari gagasan-gagasan tentang Tuhan, manusia, dan lain sebagainya. Sedangkan sistem yang bersifat fisik adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan. (Tata Sutabri, 2012 : 4-6).

II.1.3. Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. (Tata Sutabri, 2012 ; 13-14).

Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan

mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan Sistem (*Boundary*)

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

Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk

subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem tersebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh di dalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah signal yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, di mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolahan Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi, Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

II.2. Sistem Pendukung Keputusan

Michael S.Scott Morton mendefenisikan Sistem Pendukung Keputusan sebagai system berbasis komputer interaktif, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur. Pada dasarnya SPK dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pengambilan keputusan, sampai mengevaluasi pemilihan alternative (Hermawan Ariyanto, 2013).

Suatu SPK memiliki tiga subsistem utama yang menentukan kapabilitas teknik SPK tersebut, yaitu subsistem manajemen basis data, subsistem manajemen berbasis model, dan subsistem perangkat lunak penyelenggara dialog.

1. Subsistem manajemen basis data

Subsistem ini berfungsi untuk menyimpan data-data yang dihasilkan dari *internal*, *eksternal*, dan data *privat*.

2. Subsistem manajemen berbasis model

Subsistem ini berfungsi untuk menyederhakan permasalahan, sehingga masalah lebih mudah dipahami.

3. Subsistem perangkat lunak

Subsistem ini berfungsi untuk berkomunikasi antara pengguna dengan sistem.

II.2.1. Jenis Sistem Pendukung Keputusan

Keputusan-keputusan yang dibuat pada dasarnya dikelompokkan dalam 2 jenis, antara lain:

1. Keputusan Terprogram

Keputusan ini bersifat berulang dan rutin, sedemikian hingga suatu prosedur pasti telah dibuat menanganinya sehingga keputusan tersebut tidak perlu diperlakukan *de novo* (sebagai sesuatu yang baru) tiap kali terjadi.

2. Keputusan Tak Terprogram

Keputusan ini bersifat baru, tidak terstruktur jarang konsekuen. Tidak ada metode yang pasti untuk menangani masalah ini karena belum ada sebelumnya atau karena sifat dan struktur persisnya tak terlihat atau rumit atau karena begitu pentingnya sehingga memerlukan perlakuan yang sangat khusus. (Rika Yunitarini, 2013 : 46).

II.3. Metode AHP (*Analityc Hierarchy Process*)

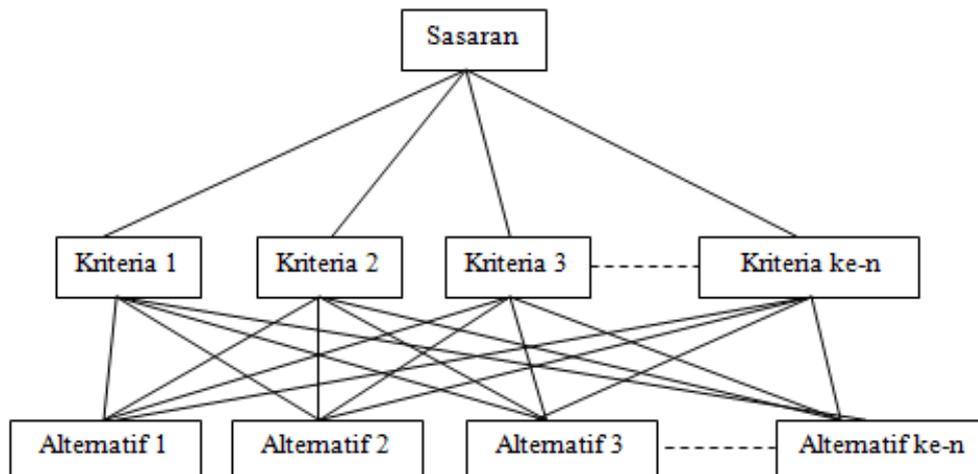
Pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. Peralatan utama AHP adalah sebuah hierarki fungsional dengan input utamanya persepsi manusia. Keberadaan hierarki memungkinkan dipecahnya masalah kompleks atau tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi suatu bentuk hierarki. AHP memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga mudah dipahami oleh semua pihak yang terlibat dalam pengambilan keputusan. (Syahrani Dhimas Prabowo, Eko Budi Setiawan, 2013 : 29).

II.3.1. Prinsip Dasar AHP (*Analityc Hierarchy Process*)

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah (Syahrani Dhimas Prabowo, Eko Budi Setiawan, 2013 : 29 - 30):

1. Membuat hierarki

Sistem yang kompleks bisa diatasi dengan memecahnya menjadi elemen - elemen pendukung, menyusun elemen secara hierarki dan menggabungkannya atau mensintesisnya.



Gambar II.1. Struktur Hirarki AHP

(Sumber : Syahrani Dhimas Prabowo, Eko Budi Setiawan, 2013)

2. Penilaian kriteria dan alternatif

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan, untuk berbagai, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat.

3. *Synthesis of priority* (menentukan prioritas)

Untuk setiap kriteria dan alternative, perlu dilakukan perbandingan berpasangan (*Pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternative kriteria bisa disesuaikan dengan dengan judgement yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.

4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna, pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antarobjek yang didasarkan pada kriteria tertentu.

II.3.2. Prosedur AHP (*Analityc Hierarchy Process*)

Pada dasarnya, prosedur atau langka-langka metode AHP meliputi (Syahrani Dhimas Prabowo, Eko Budi Setiawan , 2013 : 29) :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.

2. Menentukan prioritas elemen

Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.

3. Sintesis

Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah :

- a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
- b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
- c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4. Mengukur konsistensi

Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena pengguna tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi

yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :

- a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
- b. Jumlahkan setiap baris.
- c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
- d. Jumlahkan hasil bagi di atas dengan banyaknya elemen yang ada, hasilnya disebut λ maks.

5. Hitung Consistency Index (CI) dengan rumus :

$$CI = (\lambda_{maks} - n) / n \quad (1)$$

Dimana n = banyaknya elemen

6. Hitung Rasio Konsistensi atau Consistency Ratio (CR) dengan rumus :

$$CR = CI / IR \quad (2)$$

Dimana CR = Consistency Ratio

CI = Consistency Index

IR = Index Random Consistency

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data judgement harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar. Berdasarkan perhitungan Thomas L. dengan menggunakan 500 sampel diperoleh nilai rata-rata indeks random (RI) untuk setiap ordo matrik tertentu.

Tabel II.1. Daftar Indeks Random Konsistensi

Ukuran Matriks	Nilai IR
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

Sumber : Syahrani Dhimas Prabowo, Eko Budi Setiawan (2013 : 29)

II.4. Pengertian Basis Data

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi. Di dalam basis data, semua data diintegrasikan dengan departemen dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data dapat juga dideskripsikan sebagai kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam file-file terpisah ke dalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut (Indrajani, 2015 : 70).

II.4.1. Perencanaan Basis Data

Merupakan aktivitas manajemen untuk merealisasikan tahapan *Database Application Lifecycle* secara efektif dan efisien. Perencanaan basis data mencakup cara pengumpulan data, format data, dokumentasi yang diperlakukan, cara membuat desain, dan implementasi. Perencanaan basis data terintegrasi dengan keseluruhan strategi sistem informasi organisasi. (Indrajani, 2015 : 70).

Terdapat 3 hal yang berkaitan dengan strategi sistem informasi, yaitu :

1. Identifikasi rencana dan sasaran dari organisasi termasuk mengenai sistem informasi yang dibutuhkan.

2. Evaluasi sistem informasi yang ada untuk menetapkan kelebihan dan kekurangan yang dimiliki oleh sistem tersebut.
3. Penaksiran kesempatan teknik informatika yang mungkin memberikan keuntungan kompetitif. (Indrajani, 2015 : 70).

II.5. ERD

Entity Relational (ER) Modeling adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut dengan entitas dan hubungan antara entitas-entitas tersebut yang digambarkan dalam suatu model. Karena terdapat keterbatasan pada ER Model, maka terdapat pengembangan penambahan konsep semantik pada ER yang disebut *Enhanced Entity Relational (EER) Model*. (Indrajani : 2015 ; 17).

Relasi antara dua file atau dua tabel dapat dikategorikan menjadi 3 macam. Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan atribut yang ada dalam satu atau dua file. (Tata Sutabri : 2012 ; 144).

Adapun relasi yang terjadi di antara dua himpunan entitas dapat berupa sebagai berikut:

1. *One to one relationship* dua file

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Seperti pada pelajaran privat di mana satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru.

2. *One to many relationship* dua file

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak lawan satu. Seperti pada sistem pengajaran di sekolah dasar, di mana satu guru mengajar banyak siswa dan siswa hanya diajar oleh satu guru.

3. *Many to many relationship* dua file

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi, di mana satu dosen mengajar banyak mahasiswa dan mahasiswa diajar oleh banyak dosen.

II.6. Kamus data

Kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi. Kamus data terdapat pada tahapan analisis dan perancangan. Pada tahap analisis, kamus data berfungsi untuk mendefinisikan data yang mengalir pada sistem. Sedangkan pada tahap perancangan, kamus data ini digunakan untuk merancang masukan dan keluaran seperti laporan serta basis data. Pada DFD aliran data memiliki sifat global, sedangkan pada kamus data dibuat berdasarkan aliran data yang terdapat pada DFD. (Indrajani : 2015 ; 30).

Sumber kamus data yaitu :

1. Data Store (file-file)
2. Data Flow (aliran data)

3. Data Element yang dinyatakan dalam spesifikasi data dan berasal dari file.

Tabel II.2 Notasi Kamus Data

Notasi	Keterangan
=	Is Composed Of
+	And
()	Optional (May be present or absent)
{ }	Iteration
[]	Select one of several alternative choices
**	Comment
@	Identifier (key field) for a store
	Separates alternative choices in the construct

(Sumber : Indrajani ; 31)

Contoh kamus data, antara lain:

name = courtesy-title + first-name +(middle-name) + last-name

courtesy-title = [Mr. | Miss | Mrs. | Ms. | Dr. | Profesor]

first-name = {legal-character}

middle-name = {legal-character}

last-name = {legal-character}

legal-character = [A-Z|a-z|0-9|'|-'|]

II.7. Normalisasi

Normalisasi adalah suatu teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji

hubungan, yaitu *functional dependencies* antara atribut. Pengertian lainnya adalah suatu teknik yang menghasilkan sekumpulan hubungan dengan sifat-sifat yang diinginkan dan memenuhi kebutuhan pada perusahaan. (Indrajani, 2015 : 7).

Proses normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Proses ini selalu diuji pada beberapa kondisi. Apakah ada kesulitan pada saat menambah (*insert*), menghapus (*delete*), mengubah (*update*), atau membaca (*retrieve*) pada suatu database. Bila ada kesulitan pada pengujian tersebut maka relasi dapat dipecah dalam beberapa tabel lagi. (Tata Sutabri, 2012 : 138).

Adapun bentuk-bentuk normalisasi menurut (Indrajani, 2015 : 9-10) adalah sebagai berikut:

1. Unnormalized Form (UNF)

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang *unnormalized*, yaitu dengan memindahkan data dari sumber informasi.

Contoh: nota penjualan yang disimpan ke dalam format tabel dengan baris dan kolom.

2. First Normal Form (1NF)

Merupakan sebuah relasi di mana setiap baris dan kolom berisikan satu dan hanya satu nilai.

Proses UNF ke 1NF

- a. Tentukan satu atau sekumpulan atribut sebagai kunci untuk tabel *unnormalized*.

b. Identifikasikan grup yang berulang dalam tabel *unnormalized* yang berulang untuk kunci atribut.

3. Second Normal Form (2NF)

Berdasarkan pada konsep *full functional dependency*, yaitu A dan B merupakan atribut sebuah relasi. B dikatakan *fully dependent* terhadap A jika B *functional dependent* pada A tetapi tidak pada propersubset dari A. 2NF merupakan sebuah relasi dalam 1NF dan setiap atribut *non-primary-key* bersifat *fully functionally dependent* pada *primary key*.

1NF ke 2NF

- a. Identifikasikan *primary key* untuk relasi 1NF.
- b. Identifikasikan *functional dependencies* dalam relasi.
- c. Jika terdapat *partial dependencies* terhadap *primary key*, maka hapus dengan menempatkan dalam relasi yang baru bersama salinan determinannya.

4. Third Normal Form (3NF)

Berdasarkan pada konsep *transitive dependency*, yaitu suatu kondisi di mana A, B, dan C merupakan atribut sebuah relasi, maka A – B dan B – C, maka *transitively dependent* pada A melalui B. 3NF adalah sebuah relasi dalam 1NF dan 2NF, di mana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* pada *primary key*.

2NF ke 3NF

- a. Identifikasi *primary key* dalam relasi 2NF.

- b. Identifikasi *functional dependencies* dalam relasi.
- c. Jika terdapat *transitive dependencies* terhadap *primary key* hapus dengan menempatkannya dalam relasi yang baru bersama dengan salinan determinannya.

5. Boyce-code Normal Form (BCNF)

Berdasarkan pada *functional dependencies* yang dimasukkan dalam hitungan seluruh *candidate key* dalam suatu relasi. Bagaimanapun BCNF juga memiliki batasan-batasan tambahan disamakan dengan defenisi umum dari 3NF. Suatu relasi dikatakan BCNF, jika dan hanya jika setiap determinan merupakan *candidate key*. Perbedaan antara 3NF dan BCNF yaitu untuk *functional dependent* $A - B$, 3NF memungkinkan *dependency* ini dalam suatu relasi jika adalah atribut *primary key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menetapkan dengan jelas bahwa untuk *dependency* ini agar ditetapkan dalam relasi A, maka A harus merupakan *candidate key*. Setiap relasi dalam BCNF juga merupakan 3NF, tetapi relasi dalam 3NF belum tentu termasuk ke dalam BCNF. Dalam BCNF kesalahan jarang sekali terjadi, Kesalahan dapat terjadi pada relasi yang (Indrajani, 2015 : 9-10):

- a. Terdiri atas 2 atau lebih *composite candidate key*.
- b. *Candidate key overlap*, sedikitnya satu atribut.

II.8. *SQL Server 2008 R2*

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi setiap bahasa ini memiliki bentuk standar yang ditetapkan oleh ANSI. (*Jurnal Sistem Informasi*, Vol. 6, No.2, Adelia, Jimmy Setiawan, September 2011 : 115).

1. *Data Definition Language* (DDL), digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data. Secara umum DDL yang digunakan adalah:
 - a) *CREATE* untuk membuat objek baru.
 - b) *USE* untuk menggunakan objek.
 - c) *ALTER* untuk mengubah objek yang sudah ada.
 - d) *DROP* untuk menghapus objek.
2. *Data Manipulation Language* (DML), digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah-perintah yang umum dilakukan adalah:
 - a) *SELECT* untuk menampilkan data.

- b) *INSERT* untuk menambahkan data baru.
- c) *UPDATE* untuk mengubah data yang sudah ada.
- d) *DELETE* untuk menghapus data.

II.9. Microsoft Visual Basic 2010

Pada akhir tahun 1999, Teknologi .NET diumumkan. Microsoft memosisikan teknologi tersebut sebagai *platform* untuk membangun XML Web *Services*. XML Web *services* memungkinkan aplikasi tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apapun melalui internet. Visual Basic.NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal .NET Framework. (Priyanto Hidayatullah, 2012 : 5).

II.10. Unified Modeling Language (UML)

UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan sebuah sistem pengembangan software berbasis *object oriented*. UML sendiri juga memberikan standar penulisan sebuah system blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik. (Jurnal Sisfotek Global, Rachmat Agusli, ISSN : 2088 – 1762 Vol. 5 No. 1 / Maret 2015).

UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen- komponen yang diperlukan dalam sistem *software*. Notasi standar yang disediakan UML bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis yaitu , diagram *use case*, diagram *sequence*, dan diagram *class*. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML cocok untuk penulisan piranti lunak dalam bahasa pemrograman yang berorientasi objek. (Jurnal Ilmiah komputer dan Informatika, Volume 1 No 1, Ari Wiwekananda, Agustus, 2012).

II.10.1. Diagram *Unified Modeling Language* (UML)

Diagram-diagram yang termasuk di dalam UML, yaitu (Indrajani, 2015 : 45 – 52) :

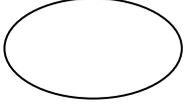
1. Use Case Diagram

Merupakan suatu diagram yang merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan.

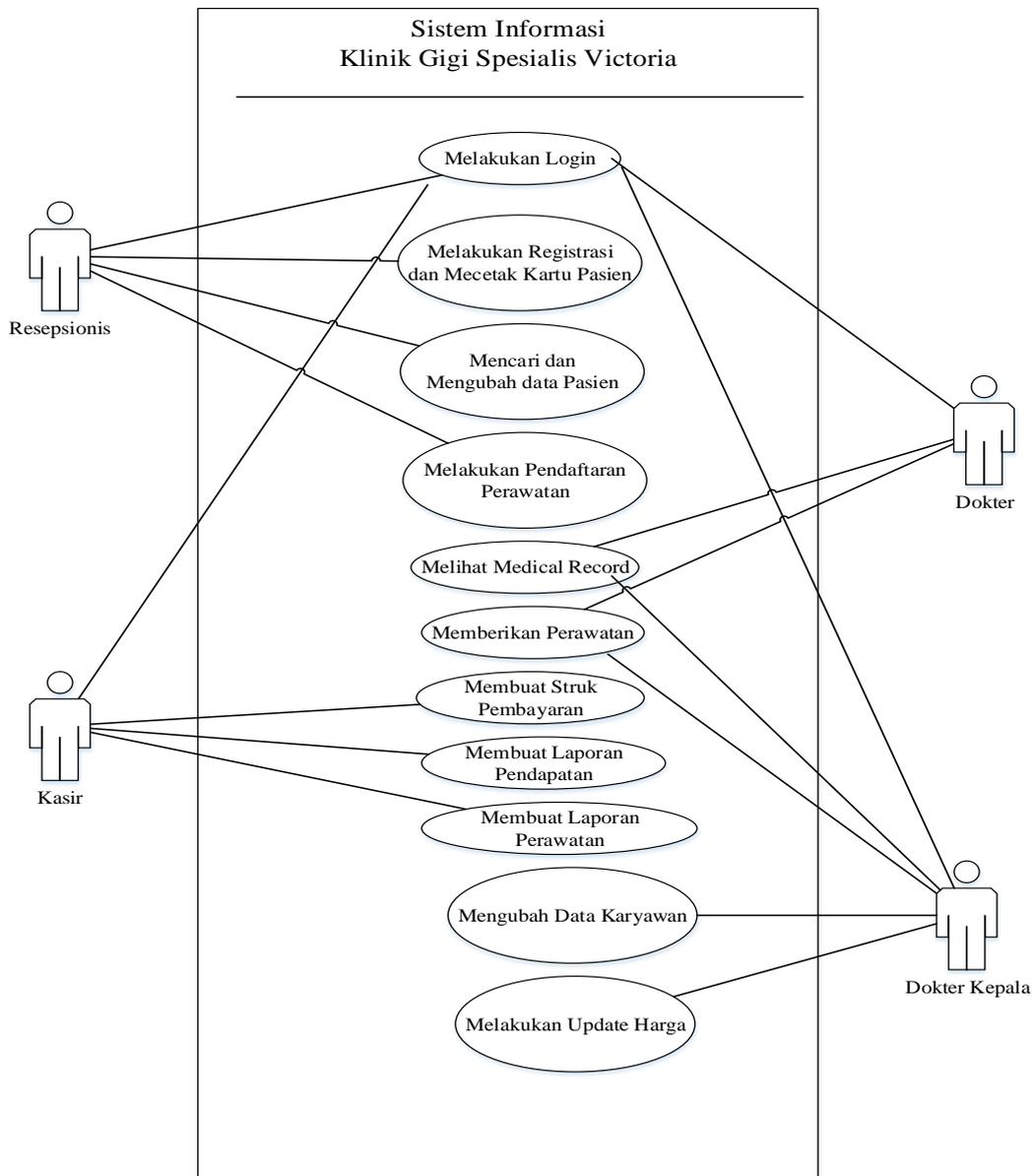
Use case diagram dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan dari suatu sistem. Jadi, dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan

sesuatu, bagaimana cara actor akan menggunakan sistem, serta apa saja yang dapat dilakukan terhadap suatu sistem.

Tabel II.3. Komponen *Use Case Diagram*

	<p><i>System Boudary</i> menggambarkan batasan antara sistem dengan actor.</p>
	<p>Simbol ini menggambarkan interaksi antara actor dengan <i>software</i> aplikasi tersebut.</p>
	<p>Actor menggambarkan pengguna dari sistem, dapat berupa manusia atau sistem terotomatisasi lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.</p>
	<p>Menggambarkan hubungan antar actor dan <i>use case</i>.</p>

(Sumber : Indrajani, 2015 : 46)

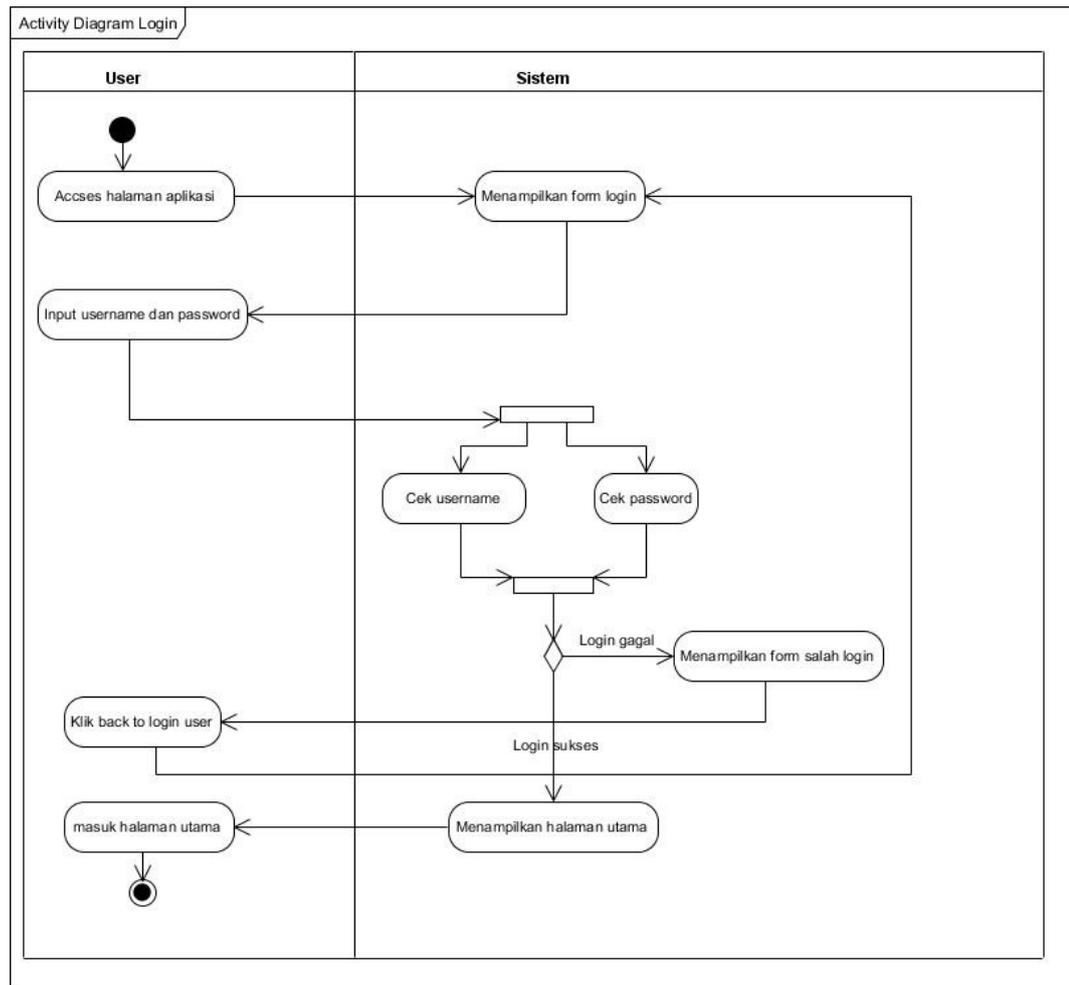


Gambar II.2. Use Case Diagram
(Sumber : Indrajani., 2015 : 45)

2. Activity Diagram

Digunaka untuk menganalisis *behavior* dengan *use case* yang lebih kompleks dan menunjukkan interaksi-interaksi di antara mereka satu sama lain. *Activity diagram* sebenarnya memiliki kesamaan dengan

statechart diagram dalam hal menggambarkan aliran data pada model bisnis, tetapi *activity* diagram biasanya digunakan untuk menggambarkan aktivitas bisnis yang lebih kompleks, dimana digambarkan hubungan antar satu *use case* dengan *use case* lainnya.

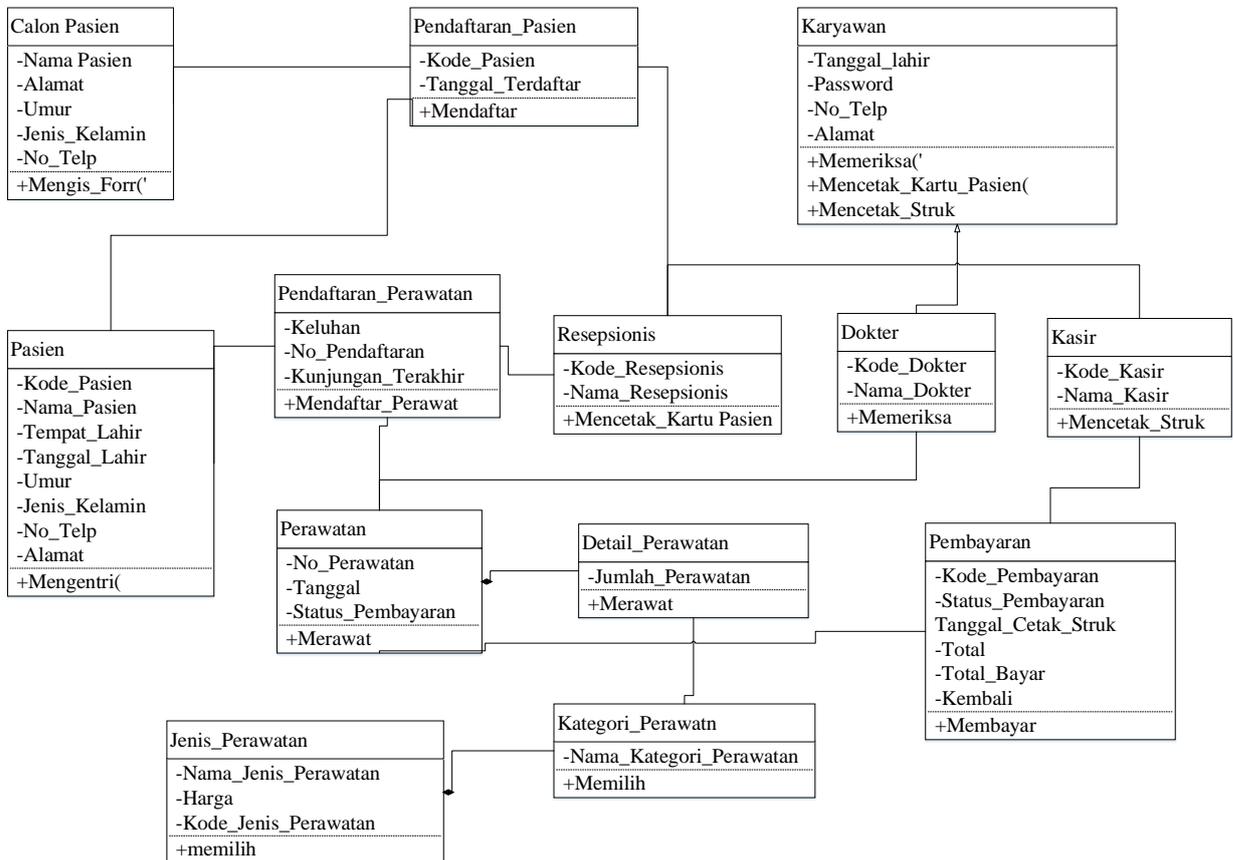


Gambar II.3. Activity Diagram
(Sumber : Rosana Junita Sirait, dkk., 2015)

3. Class Diagram

Diagram ini digunakan untuk menggambarkan perbedaan yang mendasar antara *class-class*, hubungan antar-*class*, dan di mana sub-

sistem *class* tersebut. Pada *class* diagram terdapat *mana class*, *attributes*, *operations*, serta *association* (hubungan antar-*class*).

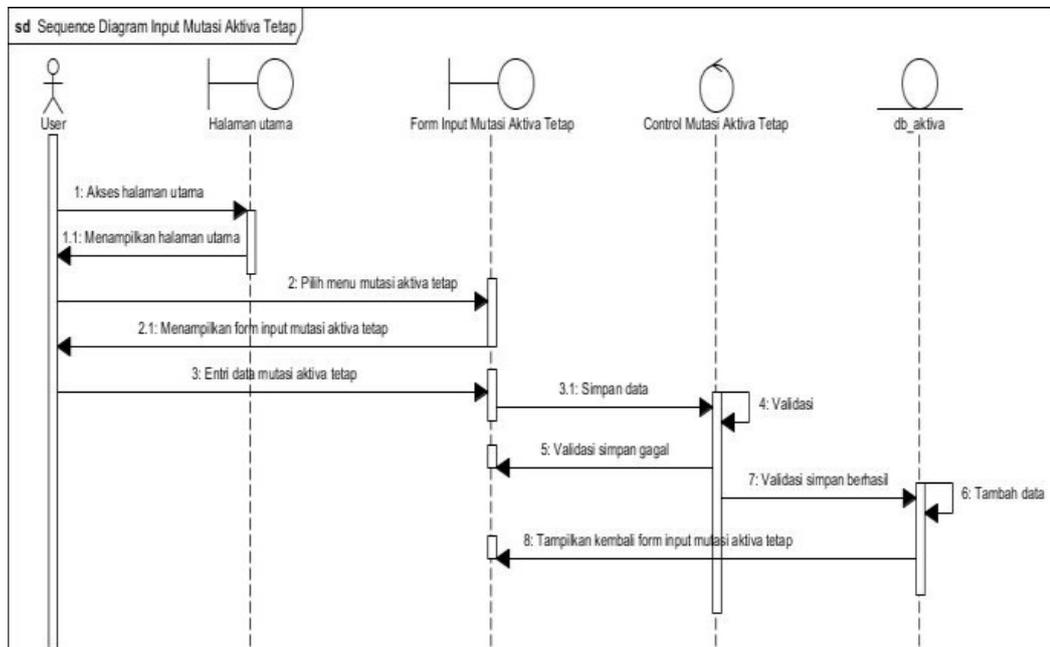


Gambar II.4. Class Diagram
(Sumber : Indrajani., 2015 : 50)

4. Sequence Diagram

Merupakan suatu diagram interaksi yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi (*particular interaction*) dan pesan yang ditukar dalam urutan waktu. *Sequence* diagram merupakan peralatan untuk interaksi berkomunikasi diagram.

Sebuah interaksi didesain antara objek atau sistem yan berpartisipasi dalam sebuah kolaborasi.



Gambar II.5. Sequence Diagram
(Sumber : Rosana Junita Sirait, dkk., 2015)