

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Di dalam penulisan laporan ini, penulis melakukan perbandingan terhadap penelitian-penelitian terdahulu yang berhubungan dengan Poliklinik, dengan maksud meneliti kelebihan, kekurangan, dan menggali informasi yang bisa mendukung dilakukannya penulisan laporan ini. Berikut adalah referensi yang penulis jadikan sebagai rujukan :

1. Penelitian yang berjudul “RANCANG BANGUN APLIKASI REKAM MEDIS POLIKLINIK UNIVERSITAS TRILOGI”, tahun 2015, oleh Umar Al Faruq, Program Studi Sistem Informasi Fakultas Telematika, Universitas Trilogi Jakarta [1]. Jurnal ini berisi tentang penelitian dan perancangan aplikasi rekam medis pada poliklinik di Universitas Trilogi Jakarta menggunakan. Aplikasi ini yang akan menunjang prosedur-prosedur yang ada dalam sistem informasi Poliklinik di Universitas Trilogi Jakarta.
2. Penelitian yang berjudul “PERANCANGAN SISTEM INFORMASI POLIKLINIK”, tahun 2012, oleh R. Edith Indera Bagaskara, Wahyul Amien Syafei dan R. Rizal Isnanto, Teknik Elektro, Universitas Diponegoro Semarang [2]. Jurnal ini berisikan tentang perancangan dan implementasi Sistem Informasi Poliklinik berbasis komputer di Balai Pengobatan dan Rumah Bersalin MTA Solo.

Dari kedua referensi di atas, dapat diambil perbedaan yang signifikan pada metode pengembangan maupun hasil perancangannya. Program yang dihasilkan pun berbeda. Penelitian pertama menggunakan metode pengembangan SLDC atau *Waterfall* dan merancang dan membangun dari awal aplikasi rekam medis, sedangkan penelitian kedua menggunakan metode pengembangan berbasis struktur (tidak disebutkan namanya), aplikasi yang dihasilkan pun cakupannya luas yakni seluruh sistem informasi poliklinik (beberapa modul).

Adapun perbedaan kedua penelitian di atas dengan penelitian yang penulis lakukan, yakni laporan penelitian ini hanya tentang pengembangan aplikasi yang sudah ada, yaitu aplikasi Poliklinik, serta yang dikembangkan hanya pada prosedur *Medical Check Up* / MCU.

2.2. Dasar Teoritis

2.2.1. Pengertian Sistem

Sistem menurut McLeod yang dikutip (2010:34) dalam bukunya yang berjudul "*Management Information System*" adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan [3]. Sedangkan Menurut Satzinger, Jackson, dan Burd (2010:6) sistem merupakan sekumpulan komponen yang saling berhubungan dan bekerja bersama untuk mencapai suatu tujuan. Berdasarkan kedua pengertian di atas, maka dapat disimpulkan bahwa pengertian sebuah sistem adalah sekumpulan elemen yang terintegrasi dan bekerja bersama guna mencapai suatu tujuan tertentu [4].



Gambar 2.1 Contoh Model Sistem Sederhana

2.2.1.1. Elemen-elemen Sistem

Berikut ini elemen – elemen yang membentuk sebuah sistem, yaitu :

1. Tujuan

Setiap perancangan sistem mempunyai tujuan dan tujuan inilah yang membantu mengarahkan sistem menjadi berjalan dengan baik.

2. Masukkan (*Input*)

Masukkan sistem yaitu segala sesuatu yang masuk ke dalam sistem yang selanjutnya akan diproses oleh sistem.

3. Proses

Proses merupakan bagian yang melakukan perubahan dari masukkan menjadi keluaran yang berguna.

4. Keluaran (*Output*)

Keluaran ini merupakan hasil dari masukkan yang sudah diproses.

5. Batas

Batas sistem disini adalah pemisah antara sistem dan daerah luar sistem (lingkungan). Batasan sistem menentukan konfigurasi, ruang lingkup atau kemampuan sistem.

6. *Control* dan Umpan Balik (*Feed Back*)

Control ini diwujudkan dengan menggunakan *feedback* yang menerangkan keluaran, umpan balik ini juga digunakan untuk

mengendalikan baik masukan atau pun proses tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

7. Lingkungan

Lingkungan yaitu segala sesuatu yang berada di luar sistem. Lingkungan bisa berpengaruh terhadap operasi dalam arti bisa merugikan atau menguntungkan sistem itu sendiri.

2.2.1.2. Karakteristik Sistem

Sistem mempunyai karakteristik tertentu, antara lain :

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama membentuk suatu komponen sistem.

2. Batasan Sistem (*Boundary*)

Merupakan daerah – daerah yang membatasi suatu sistem dengan sistem lain atau dengan lingkungan kerjanya.

3. Sub Sistem (*Sub System*)

Bagian – bagian dari sistem yang beraktifitas dan berinteraksi satu sama lain untuk mencapai tujuan dan sasarnya masing – masing.

4. Lingkungan Luar Sistem (*Environment*)

Suatu sistem yang ada diluar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*Interface*)

Media penghubung antara suatu sub sistem dengan sub sistem lain.

6. Masukan Sistem (*Input*)

Energi yang masuk ke dalam sistem berupa perawatan dan sinyal. Masukkan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi.

7. Keluaran Sistem (*Output*)

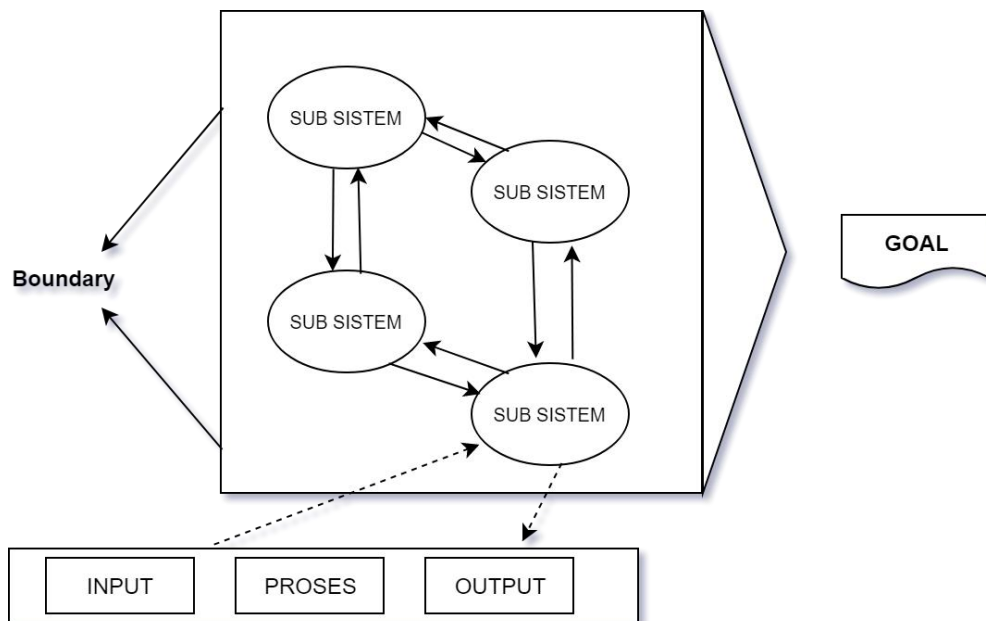
Hasil energi yang di olah dan diklasifikasi menjadi keluaran berguna dan sisa pembuangan.

8. Pengolahan Sistem (*Process*)

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

9. Sasaran Sistem (*Object*)

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.



Gambar 2.2 Contoh Karakteristik Sistem

2.2.1.3. Klasifikasi Sistem

Berikut ini adalah klasifikasi dari sistem:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berisi gagasan atau konsep, sedangkan sistem fisik yaitu sistem yang secara fisik dapat dilihat.

2. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik adalah suatu sistem yang operasinya dapat di prediksi secara tepat. Sedangkan sistem Probabilistik yaitu sistem yang tidak dapat diramal dengan pasti karena mengandung unsur probabilitas.

3. Sistem Tertutup dan Sistem Terbuka

Sistem tertutup adalah sistem yang tidak saling bertukar materi, informasi atau energi dengan lingkungan, dengan kata lain sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan. Sedangkan sistem

terbuka yaitu sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

4. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi karena alam. Sedangkan sistem buatan manusia yaitu sistem yang dibuat oleh manusia.

2.2.2. Pengertian Informasi

Informasi merupakan data yang berasal dari fakta yang tercatat dan selanjutnya dilakukan pengolahan (proses) menjadi bentuk yang berguna atau bermanfaat bagi pemakainya. Bentuk informasi yang kompleks dan terintegrasi dari hasil pengolahan sebuah *database* yang akan digunakan untuk proses pengambilan keputusan pada manajemen akan membentuk Sistem Informasi Manajemen.

Kualitas suatu informasi tergantung dalam hal berikut :

a. Relevan (*Relevancy*)

Informasi yang didapat harus bermanfaat bagi pemakainya.

b. Akurat (*Accuracy*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan serta harus jelas mencerminkan maksudnya.

c. Tepat Waktu (*Timriness*)

Informasi yang dihasilkan atau dibutuhkan harus tepat waktu.

d. Ekonomis (*Economic*)

Biaya sistem informasi akan mengalami kenaikan sesuai dengan berjalannya waktu.

e. Efisien (*Efficiency*)

Berapa banyak produksi meningkat karena tambahan unit sumber daya produksinya.

f. Dapat dipercaya (*Reliability*)

Sebuah indikator penting dari sistem informasi adalah dengan memperhatikan masalah reliabilitasnya.

2.2.3. Pengertian Sistem Informasi

Sistem Informasi adalah sekumpulan *Hardware*, *Software* dan *Brainware*, *procedure* atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan.

Keluaran suatu informasi dapat dikelompokkan ke dalam Lima jenis utama, yaitu :

1. Dokumen transaksi
2. Laporan yang direncanakan sebelumnya
3. Jawaban atas pertanyaan yang direncanakan sebelumnya
4. Laporan jawaban atas pertanyaan yang bersifat sementara
5. Integrasi antara mesin dan manusia

2.2.4. Pengertian Aplikasi

Aplikasi merupakan penerapan dari rancangan sistem untuk mengolah data yang menggunakan ketentuan bahasa pemrograman tertentu dan juga suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari *user*.

2.2.5. Pengertian Poliklinik

Poliklinik adalah salah satu unit pelayanan masyarakat yang bergerak pada bidang kesehatan. Sebuah poliklinik yang menawarkan fasilitas perawatan kesehatan yang dikhususkan untuk perawatan pasien rawat jalan. Tujuan utama dari Poliklinik adalah melayani para pasien yang akan berobat dan merupakan kegiatan utama, untuk itu penyediaan informasi yang dapat mendukung dan meningkatkan sistem sangatlah dibutuhkan [2].

2.2.6. Pengertian MCU

Medical check up merupakan pemeriksaan yang lebih difokuskan pada upaya pencegahan primer dan sekunder, yaitu mendeteksi berbagai faktor kesehatan secara menyeluruh yang dapat menimbulkan penyakit tertentu di kemudian hari. Harapan dari melakukan *medical check up* adalah mengetahui berbagai faktor risiko dengan melakukan perubahan-perubahan, misalnya mengubah kebiasaan merugikan tubuh dan mungkin juga bantuan obat-obatan (Cahyono, 2008:184)[5].

2.2.7. UML (*Unified Modelling Language*)

2.2.7.1. Pengenalan UML

UML (*Unified Modeling Language*) yang diperkenalkan pertama kali pada tahun 1990-an oleh Ivan Jacobson, serta Grady Booch. UML merupakan ‘bahasa’ untuk visualisasi, spesifikasi, konstruksi dan dokumentasi.

UML sebagai suatu cara untuk mengkomunikasikan idenya kepada para pembuat program serta calon pengguna sistem. Dalam rangka spesifikasi UML menyediakan model - model yang tepat serta lengkap. Secara khusus, UML

menspesifikasi langkah – langkah penting dalam pengambilan keputusan analisis dan tujuan penggunaan UML adalah sebagai berikut :

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. Menciptakan suatu bahasa pemodelan yang dapat di gunakan baik oleh manusia atau mesin.

UML terdiri dari 13 jenis diagram resmi, berikut 3 buah diagram yang akan digunakan dalam pemodelan penulisan ini, yaitu *Use Case Diagram*, *Skenario Use Case*, *Activity Diagram*.

2.2.7.2. *Use Case Diagram*

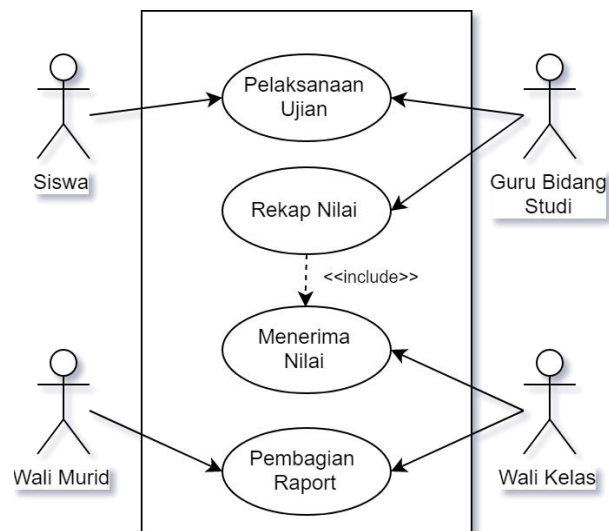
Use case diagram menggambarkan sejumlah eksternal actors dan hubungannya dengan *use case* yang diberikan oleh sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem. *Use case* adalah deskripsi fungsi yang disediakan sistem dalam bentuk teks sebagai dokumentasi dari *use case* simbol namun dapat juga dilakukan dalam *activity diagram*.

Dalam pemodelan sistem dengan UML, *actor* adalah seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan. *Use case* dan *actor* masing-masing tidak berdiri sendiri. mereka saling terhubung dengan apa yang dinamakan relasi.

Dalam hal ini, ada berbagai relasi yang dikenal dalam model UML, yaitu:

1. Relasi asosiasi, yaitu relasi yang terjadi antara *actor* dengan *use case* biasanya berupa asosiasi.

2. *Include Relationship*, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana kondisi ini sebuah *use case* adalah bagian *use case* lainnya.
3. *External Relationship*, yaitu kelakuan yang hanya terjadi dibawah kondisi tertentu. *Extend Relationship* memungkinkan suatu *use case* memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan *use case* lainnya.



Gambar 2.3 Contoh Use Case Diagram

2.2.7.3. Scenario Use Case

Setiap *use case* dilengkapi dengan skenario. Skenario *use case* adalah alur jalannya proses *use case* dari sisi aktor dan sistem. Skenario normal adalah skenario bila sistem berjalan normal tanpa terjadi kesalahan atau *error*. Sedangkan skenario alternatif adalah skenario bila sistem tidak berjalan normal, atau mengalami *error*.

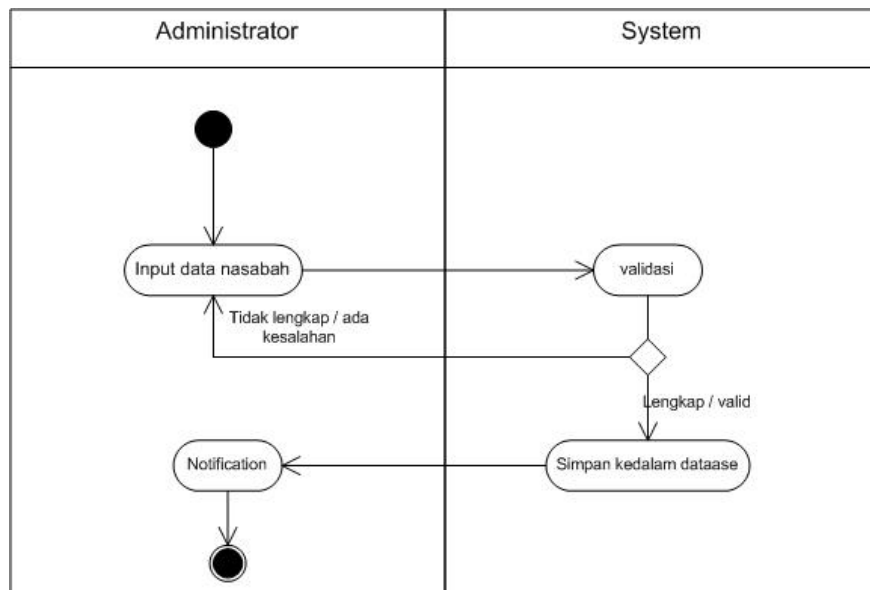
Nama Use Case : Absensi.			
Aktor : Mahasiswa.			
Deskripsi : Sebagai dokumentasi untuk mahasiswa yang sudah mengikuti atau memasuki perkuliahan.			
Skenario Utama			
Kondisi Awal : user (mahasiswa) sudah login.			
No	Aksi Aktor	No	Reaksi Sistem
1	Memilih menu absensi.	2	Menampilkan form absensi.
3	Input data absensi.	4	Menyimpan data absensi ke dalam database.
Kondisi Akhir : data absensi tersimpan ke dalam database.			

Gambar 2.4 Contoh Skenario Use Case

2.2.7.4. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas [7].



Gambar 2.5 Contoh Activity Diagram

2.3. Piranti Pendukung

Piranti pendukung untuk pengembangan aplikasi Poliklinik ini terbagi menjadi 3 bagian. Apabila terdapat istilah yang kurang dipahami, pembaca dapat membacanya melalui referensi-referensi seperti mesin telusur atau pun buku-buku yang berkaitan. :

a. Hardware

Hardware yang diperlukan yaitu :

1. Satu set *PC Server* yang, dengan minimum RAM 8 GB DDR4, *Hardisk* 1 TB, *Processor* Intel Xeon 2 Ghz.
2. Sebuah Switch
3. Satu set *PC User*, dengan minimum RAM 4 GB DDR4, *Hardisk* 512 GB, *Processor* Intel Core i3 2.0 Ghz
4. Kabel UTP
5. *Connector* RJ-45

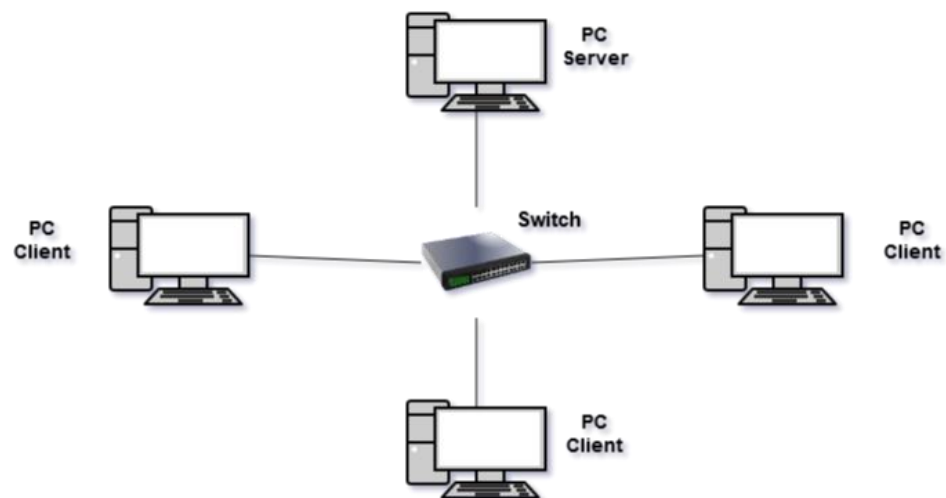
b. *Software*

Beberapa *software* yang perlu ter-*install* :

1. Windows 10 64 bit
2. PHP 5.x.x
3. Xampp Control 3.2.1
4. PostgreSQL
5. Oracle 10g
6. *Web Browser*
7. *Code Editor*
8. Zend Framework
9. PG Maestro

c. *Network*

Topologi jaringan yang digunakan yaitu berjenis Topologi *Star*. Dimana semua perangkat saling terhubung melalui perangkat switch, baik itu *PC Server* atau pun *PC User*.



Gambar 2.6 Topologi *star* yang digunakan