BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Muhammad Yuki Saputra dan dkk, 2019. Pada sistem pakar ini terdapat fakta-fakta atau gejala-gejala mengenai Kelainan pada sistem pencernaan anak yang digunakan sebagai pengetahuan dasar terhadap pendiagnosaan penyakit Kelainan pada sistem pencernaan anak. Dimana pencegahan tersebut tergantung pada tingkat persentase diagnosa Kelainan pada sistem pencernaan anak yang dihasilkan.

Melda Vandy Septiandika, 2014. Kematian akibat campak sering terjadi pada anak dengan malnutrisi terutama di negara berkembang. Kelompok yang paling rentan untuk terkena penyakit ini adalah bayi dan anak-anak yang belum pernah mendapatkan imunisasi campak. Penyakit ini juga merupakan salah satu penyebab utama tingginya angka kesakitan dan angka kematian pada bayi dan anak-anak.

Reza Setiawan,(2018). Untuk membantu seseorang mendiagnosa dini penyakit dini yang mungkin diderita salah satunya caranya adalah menggunakan sistem pakar. Sistem ini diharapkan dapat memberikan diagnosa awal terkait penyakit yang diderita oleh seseorang sehingga dapat ditangani secara dini. Pada penelitian ini dibangun sistem pakar dengan menggunakan metode *Dempster-Shafer* berbasis web yang dapat mendiagnosa penyakit campak berdasarkan gejala-gejala yang dialami.

II.2. Sistem

Sistem umumnya diartikan sebagai satu kesatuan yang utuh. Menurut Mulyanto (2009), sistem adalah kumpulan dari sub-sub system baik abstrak maupun fisik yang saling terintegrasi dan salin berkolaborasi untuk mencapai suatu tujuan tertentu. Al Fatta (2013) mendefenisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain. Dari definisi tersebut dapat disimpulkan bahwa sistem adalah kumpulan dari elemen-elemen yang saling berinteraksi dan saling berhubungan satu sama lainnya untuk mencapai suatu tujuan. (Muhdar Abdurahman: 2018)

II.3. Informasi

Informasi merupakan data atau fakta yang telah diproses sedemikian rupa, sehingga berubah bentuknya menjadi informasi. Di samping itu informasi dapat mengurangi ketidakpastian serta mempunyai nilai dalam keputusan karena dengan adanya informasi kita dapat memilih tindakan-tindakan dengan resiko yang paling kecil. Untuk menghasilkan kebijaksanaan dan keputusan yang baik diperlukan pengolahan data menjadi informasi yang relevan dengan masalah perusahaan yang sedang dihadapi. Dengan demikian data itu merupakan bahan mentah yang harus diproses lebih dahulu baru kemudian dapat digunakan.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Informasi juga disebut data yang diproses atau data yang memiliki arti. Informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuann seseorang yang menggunakan. Para pembuat keputusan memahami bahawa informasi menjadi faktor kritis. Informasi dapat berupa data mentah, data tersusun, kapasitas, sebuah saliran informasi dan sebagainya. (Astia Firman dan dkk, 2016 : 29).

II.4. Sistem Informasi

Sistem Informasi adalah serangkaian *hardware, software,* data, manusia, dan prosedur yang bekerja bersama untuk memproduksi informasi. Prosedur adalah suatu instruksi atau serangkaian instruksi, yang diikuti oleh *users* untuk menyelesaikan kegiatannya. Kegunaan umum dari sistem informasi termasuk kategori dari sistem informasi yang dapat digunakan oleh hampir semua bagian di dalam perusahaan. Sistem informasi terintegrasi digunakan oleh banyak bagian dan memfasilitasi berbagi informasi *(information Sharing*) dan komunikasi dalam perusahaan (Mas Ayoe Elhias, 2016 : 3).

II.5. Sistem Pakar

Sistem Pakar (*Expert System*) merupakan solusi AI bagi masalah pemrograman pintar (*Intelligent*). Profesor Edward Feigenbaum dari *Stanford University* yang merupakan *pionir* dalam teknologi sistem pakar mendefenisikan Sistem Pakar sebagai sebuah program komputer pintar (*intelligent*

computer program) yang memanfaatkan pengetahuan (*knowledge*) dan prosedur inferensi (*inference procedure*) untuk memecahkan masalah yang cukup sulit hingga membutuhkan keahlian khusus dari manusia.

Dengan kata lain, Sistem Pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem Pakar memanfaatkan secara maksimal pengetahuan khusus layaknya seorang pakar untuk memecahkan masalah.

Pakar atau ahli (*expert*) didefenisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang tidak mampu dipecahkan kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih efisien namun bukan berarti lebih murah. Pengetahuan yang dimuat ke dalam Sistem Pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasikan lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli. Istilah Sistem Pakar (*expert system*). Sering disinonimkan dengan sistem berbasis pengetahuan (*knowledge-based system*) atau Sistem Pakar berbasis pengetahuan (*knowledge based expert system*). (B. Herawan Hayadi, 2018:2).

II.5.1. Manfaat Sistem Pakar

Sistem pakar sangat popular karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya :

1. Meningkatnya produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seseorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Memudahkan akses pengetahuan seorang pakar.
6. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
7. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar. (B. Herawan Hayadi, 2018:2).

II.5.2. Ciri–Ciri Sistem Pakar

Menurut B. Herawan Hayadi (2018), ciri–ciri sistem pakar adalah sebagai berikut :

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data yang tidak pasti
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah atau *rule* tertentu.

5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Pengetahuan dan mekanisme *inferensi* jelas terpisah.
7. Keluarannya bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara searah oleh pemakai

II.5.3. Konsep Umum Sistem Pakar

Sistem pakar terdiri dari beberapa konsep yang harus dimiliki. Konsep dasar dari suatu sistem pakar adalah sebagai berikut :

1. Keahlian

   Adalah suatu pengetahuan khusus yang diperoleh dari latihan, belajar dan pengetahuan. Pengetahuan dapat berupa fakta, teori, aturan, strategi global untuk memecahkan masalah.

2. Ahli (*Expert*)

   Melibatkan kegiatan mengenali dan memformulasikan permasalahan, memecahkan masalah secara cepat dan tepat, menerangkan pemecahannya, belajar dari pengalaman, merestrukturisasi pengetahuan, memecahkan aturan serta menentukan relevansi.

3. Mentransfer Keahlian (*Transfering Expertise*)

   Adalah proses pentransferan keahlian dari seorang pakar kedalam komputer agar dapat digunakan oleh orang lain yang bukan pakar. Pengetahuan tersebut ditempatkan kedalam sebuah komponen yang dinamakan basis pengetahuan.

4. Menyimpulkan Aturan (*Inferencing Rule*)

Merupakan kemampuan komputer yang telah diprogram, penyimpulan ini dilakukan oleh mesin *inferensi* yang meliputi prosedur tentang penyelesaian masalah.

5. Peraturan (*Rule)*

   Diperlukan karena mayoritas dari sistem pakar bersifat *rule-based systems,* yang berarti pengetahuan disimpan dalam bentuk peraturan.

6. Kemampuan menjelaskan (*Explanation Capability*)

   Adalah karakteristik dari sistem pakar yang memiliki kemampuan menjelaskan atau memberi saran mengapa tindakan dianjurkan atau tidak dianjurkan. (B. Herawan Hayadi, 2018: 5).

II.6. Penyakit Campak

Penyakit campak adalah salah satu penyakit infeksi penyebab kematian bayi diseluruh dunia yang meningkat setiap tahun. Penyakit ini diakibatkan oleh virus campak, komplikasi penyakit campak antara lain radang selaput otak (meningitis), radang paru – paru, infeksi telinga. Pada tahun 2012 di Indonesia terjadi 15.987 kasus campak, 4 diantaranya mengalami kematian, sedangkan di Jawa Tengah terjadi 490 kasus campak. Lebih dari 95 % kematian akibat campak terjadi di negara – negara berpenghasilan penduduk rendah dengan infrastruktur kesehatan lemah. Penyakit campak masih merupakan masalah kesehatan di Indonesia dalam upaya menurunkan angka kesakitan dan angka kematian. Indonesia telah melaksanakan berbagai upaya antara lain dengan program reduksi campak. Dalam rangka percepatan reduksi campak, maka dilakukan pemberian

imunisasi campak dosis tambahan pada kelompok usia yang beresiko tinggi secara lebih luas berupa pelaksanaan crash program campak pada anak usia 6 – 59 bulan, mereka yang rentan terhadap campak yaitu anak diatas satu tahun, anak tidak mendapatkan imunisasi, serta remaja dan dewasa muda yang belum mendapatkan imunisasi kedua. (Riyani Wulan Sari dan dkk,2018)

II.7. *Metode Dempster Shafer*

Metode Dempster-Shafer pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan model ketidakpastian dengan range probabilities dari pada sebagai probabilitastunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster itu pada sebuah buku yang berjudul Mathematical Theory Of Evident. Dempster-Shafer Theory Of Evidence, menunjukkan suatu cara untuk memberikan bobot kenyakinan sesuai fakta yang dikumpulkan. Pada teori ini dapat membedakan ketidakpastian dan ketidaktahuan. Teori Dempster-Shafer adalah representasi, kombinasi dan propogasi ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang secara instutitif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat. Secara umum teori Dempster-Shafer ditulis dalam suatu interval: [Belief,Plausibility]. Belief (Bel) adalah ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada evidence, dan jika bernilai 1 menunjukkan adanya kepastian. Plausibility (Pls) akan mengurangi tingkat kepastian dari evidence. Plausibility bernilai 0 sampai 1. Jika yakin akan X’, maka dapat dikatakan bahwa Bel(X’) = 1, sehingga rumus di atas nilai dari Pls(X) = 0.

Belief (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 (nol) maka mengindikasikan bahwa tidak ada *evidence,* dan jika bernilai 1 menunjukkan adanya kepastian. Menurut Giarratano dan Riley fungsi *belief* dapat diformulasikan sebagai:

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \quad \text{.....(2.1)}$$

sedangkan *Plausibility* (Pls) dinotasikan sebagai :

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{Y \subseteq X'} m(X') \quad \text{.....(2.2)}$$

dimana:

Bel(X) = *Belief* (X) Pls(X) = *Plausibility* (X)

m(X) = *mass function* dari (X) m(Y) = *mass function* dari (Y)

Plausibility juga bernilai 0 sampai 1, jika kita yakin akan X' maka dapat dikatakan *Belief* (X') = 1 sehingga dari rumus di atas nilai Pls (X) = 0. Beberapa kemungkinan range

Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* yang dinotasikan dengan Θ. FOD ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* (Adrian O'neill, 2000), dimana :

$$\Theta = \{\theta 1, \theta 2, ...\theta n\} \quad \text{.....(2.3)}$$

dimana:

$\Theta =$ FOD atau *environment*

$\theta 1....\theta n =$ elemen/unsur bagian dalam *environment*

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan P(Θ), setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m = P(\Theta) \longrightarrow [0,1]$$

sehingga dapat dirumuskan:

$$\sum_{X \in P(\Theta)} m(X) = 1 \approx \sum_{X \in P(\theta)} m(X) = 1 \quad \text{.....(2.4)}$$

dengan $P(\Theta)$ = *power set* dan m(X) = *mass function* dari (X). (Mikha Dayan Sinaga, 2016)

II.8. Database

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan dan instansi (Gellysa Urva, Helmi Fauzi Siregar : 2015).

Database diartikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil dan dicari secara tepat. Selain berisi data, *database* juga berisi metadata. Metadata adalah data yang menjelaskan tentang struktur data itu sendiri. Sebagai contoh, dapat memperoleh informasi tentang nama-nama kolom dan tipe data yang ada pada sebuah tabel. Data nama kolom dan tipe yang ditampilkan tersebut disebut metadada (Dwiny Meidelfi, dkk, 2018).

II.9. *Hypertext Preprocessor (PHP)*

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi *HTML*. Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. *PHP* merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*.

Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

Adapun pengertian lain *PHP* adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasiskan kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode *HTML*". Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode *PHP* untuk mengirimkan permintaan ke server. Sistem kerja dari *PHP* diawali dengan permintaan yangberasal dari halaman *website* oleh *browser*. Berdasarkan *URL* atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*.

Selanjutnya *webserver* akan mencarikan berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode *HTML* dan menampilkannya. Pada prinsipnya sama dengan memanggil kode *HTML*, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe *file* yang diminta user. Jika tipe *file* yang diminta adalah *PHP*, maka akan memeriksa isi script dari halaman *PHP* tersebut. Apabila dalam *file* tersebut tidak mengandung script *PHP*, permintaan user akan langsung ditampilkan ke *browser*, namun jika dalam *file* tersebut mengandung script *PHP*, maka proses akan dilanjutkan ke modul *PHP* sebagai mesin yang menerjemahkan script-script *PHP* dan mengolah *script* tersebut, sehingga dapat dikonversikan ke kode-kode *HTML* lalu ditampilkan ke *browser user*. (Astria Firman, dkk, 2016)

II.10. MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data *SQL* (DBMS) yang multithread, dan multi-user. *MySQL* adalah implementasi dari system manajemen basisdata relasional (RDBMS). *MySQL* dibuah oleh *TcX* dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. (Adis Lena Kusuma Ratna, 2014)

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu *SQL* (*Structured Query Language*). *SQL* adalah s/ebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, *MySQL* dapat dikatakan lebih unggul dibandingkan database server lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan

query *MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*. (Adis Lena Kusuma Ratna, 2014)

II.11. Normalisasi

Menurut Indrajani (2015 : 11) Normalisasi adalah suatu teknik dengan pendekatan bottom-up yang digunakan untuk membantu mengidentifikasikan hubungan. Dimulai dari menguji hubungan, yaitu functional dependecies antara atribut. Tujuan utama normalisasi adalah mengidentifikasikan kesesuain hubungan yang mendukung data untuk memenuhi kebutuhan data perusahaan. Terdapat lima bentuk normal yang biasa digunakan yaitu :

1. *First Normal Form (1 NF)*

   Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal.

2. *Second Normal Form (2 NF)*

   Untuk menjadikan tabel normal tingkatke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *fungsional* tergantung pada semua atribut pembentuk *primary key*.

3. *Third Normal Form (3 NF)*

   Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung pada *primary key*.

4. *Fourth Normal Form* (4 NF)

   Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalue dependency.*

5. *Fifth Normal Form* (5 NF)

   Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasiakan berbentuk normal kelimaji kata belter sebut dapat dipecah atau diproyeksikan menjadi beberapa table dan dari proyeksi-proyek siitu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula.

II.12. *Unified Modeling Language (UML)*

Menurut Windu Gata (2015) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

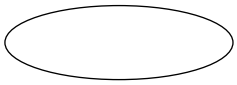
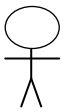
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut(Urva. Dkk, 2015) :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Adapun simbol-simbol yang digunakan dalam *use case* diagram terdapat pada tabel II.1

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki

	control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

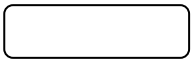
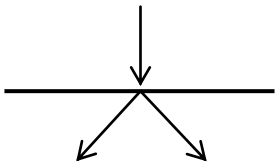
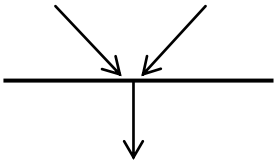
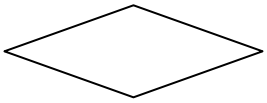
(Sumber :Urva, dkk; 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Adapun simbol-simbol yang terdapat pada activity diagram terdapat pada tabel II.2:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan

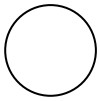
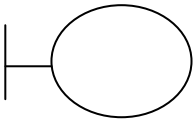
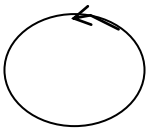
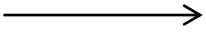
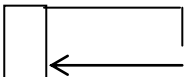
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber :Urva, dkk; 2015)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class,* berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class,* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber :Urva, dkk; 2015)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class*

diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility,* tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel dibawah ini :

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	**Penjelasan**
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber :Urva, dkk; 2015)