

BAB II

LANDASAN TEORI

2.1. Penelitian Sejenis

2.2. Sistem Pakar

Sistem pakar atau sistem pakar adalah sistem yang mencoba menyuntikkan pengetahuan manusia ke dalam komputer untuk memungkinkan komputer memecahkan masalah seperti yang biasa dilakukan para pakar, dan sistem pakar yang baik dirancang untuk memecahkan masalah tertentu dengan meniru pekerjaan Para ahli (Kusumadewi, 2003). Sistem pakar adalah paket perangkat lunak atau paket program komputer yang dirancang untuk memberikan saran dan bantuan dalam memecahkan masalah dalam bidang khusus tertentu seperti sains, teknik, matematika, kedokteran, pendidikan, dll. Sistem pakar merupakan bagian dari kecerdasan buatan (Kusumadewi, 2003).

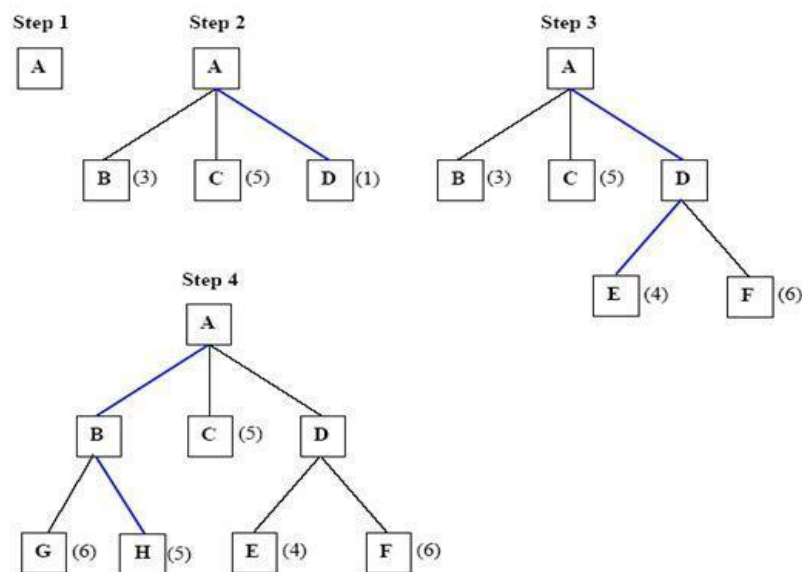
Sebuah sistem pakar biasanya didesain untuk memilih beberapa fitur, diantaranya (Kusumadewi, 2003):

- a. Kinerja tinggi. Sistem pakar harus mampu bereaksi terhadap semua kondisi di lapangan dengan baik atau lebih baik dari pakar itu sendiri.
- b. Waktu Respons yang Wajar. Sistem pakar harus dapat merespon dalam waktu yang wajar atau lebih cepat dibandingkan dengan apa yang dibutuhkan pakar untuk mengambil keputusan.
- c. Keandalan yang baik. Seharusnya tidak ada masalah dengan sistem pakar atau sistem tidak digunakan.
- d. Dapat dimengerti. Sistem pakar harus mampu menjelaskan hasil inferensi bertahap.
- e. Meningkatkan ketersediaan keahlian.
- f. Kurangi pengeluaran.
- g. mengurangi kerusakan. Sistem pakar dapat digunakan di lingkungan yang dapat membahayakan manusia.
- h. Orang mungkin pensiun, berhenti bekerja, atau mungkin mati. Tapi sistem pakar masih ada dan masih bisa membantu.

- i. Sistem pakar dapat terdiri dari beberapa pakar. Ini bisa lebih baik dari sekedar ahli karena tingkat keahlian meningkat.
- j. Sistem pakar dapat menjelaskan secara detail mengapa sebuah kesimpulan ditampilkan. Berbeda dengan orang yang mungkin terlalu lelah atau malas atau tidak mampu menjelaskannya.
- k. Respon cepat. Dalam beberapa keadaan darurat yang membutuhkan respons manusia yang lebih cepat, Anda dapat mengandalkan sistem pakar.
- l. Sistem pakar baik hati, tidak emosional, dan selalu memberikan jawaban yang sempurna. Ini sangat penting dalam situasi darurat ketika seorang ahli tidak tersedia saat dibutuhkan.

2.3. Best First Search

Pencarian pertama terbaik adalah algoritma yang memeriksa grafik dengan memperluas node yang paling menjanjikan atau node yang dipilih sesuai dengan aturan yang ditentukan. Node mewakili area pencarian. Best first search juga merupakan kombinasi dari metode depth-first search (DFS) dan breadth-first search (BFS), dimana pencarian dapat dilakukan pada node dengan level yang lebih rendah selama node tersebut memiliki nilai heuristik yang lebih baik.



Gambar 2. 1 Struktur Metode BFS

Diperlukan dua daftar untuk mengimplementasikan algoritma pencarian ini, yaitu:

OPEN untuk mengelola node yang sudah dibuat tetapi belum dievaluasi, dan CLOSE untuk mengelola node yang sudah dibuat dan dievaluasi. Algoritma lengkapnya adalah sebagai berikut.

1. OPEN berisi keadaan awal dan CLOSE kosong.
2. Ulangi sampai target ditemukan atau sampai tidak TERBUKA.
 - a. Ambil node terbaik yang dalam keadaan Open.
 - b. Jika simpulnya sama dengan target, maka sukses
 - c. Jika tidak, setel node ke CLOSED.
 - d. Buat semua aksesori untuk node ini.
 - e. Untuk setiap pengikut, lakukan hal berikut:
3. Jika follower belum pernah dinaikkan, evaluasi follower, tambahkan ke OPEN dan catat induknya.

Jika turunan sudah dibuat, ubah jalur induknya jika jalur melalui induk tersebut lebih baik daripada jalur induk sebelumnya. Kemudian perbarui pembayaran pengikut dan node anak lainnya.

2.4. Saluran Pernafasan

Penyakit pernapasan adalah penyakit pada saluran pernapasan atas atau bawah yang biasanya menular dan, tergantung pada agen penyebab, faktor lingkungan, dan faktor yang, dapat menyebabkan spektrum penyakit yang luas, mulai dari penyakit tanpa gejala atau infeksi ringan hingga penyakit berat dan fatal. Namun, pedoman ini mendefinisikannya sebagai penyakit pernapasan yang disebabkan oleh penyakit menular yang dapat ditularkan dari satu orang ke orang lain. Gejala biasanya mulai dengan cepat, mulai dari beberapa jam hingga beberapa hari. Gejalanya meliputi demam, batuk dan sering sakit tenggorokan, rhinitis (pilek), sesak napas, mengi atau kesulitan bernapas. Contoh patogen penyebab ISPA yang dicantumkan dalam pedoman ini adalah rhinovirus, respiratory syncytial virus, parainfluenza virus, Severe Acute Respiratory Syndrome Associated Coronavirus (SARS-CoV), dan virus influenza. (Patricia,

Harriman, Carmem, Wing-Hong dan Yuguo, 2007), riwayat infeksi pernafasan berulang:

Infeksi saluran pernafasan adalah infeksi pada saluran udara, hidung, sinus, tenggorokan atau laring. Infeksi saluran pernafasan merupakan penyakit yang paling banyak menyerang anak-anak. Penyakit pernafasan pada bayi dan anak dapat menyebabkan kecacatan pada usia dewasa bila dikaitkan dengan adanya PPOK 2 (Oemiati, 2013). Penyakit pernafasan adalah gangguan kompleks dan heterogen yang memiliki banyak penyebab dan dapat mempengaruhi titik manapun di sepanjang saluran udara. Penyakit pernafasan merupakan salah satu alasan utama pasien mengunjungi fasilitas kesehatan (Handayani, 2010)...

2.5. *Unified Modeling Language*

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh model meta yang solid yang dapat digunakan untuk menggambarkan dan merancang sistem berbasis komputer, terutama yang dibuat menggunakan bahasa pemrograman yang berorientasi objek. (Fowler, 2005)

Menurut (Windu Gata & Grace, 2013), UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menentukan, dan membuat perangkat lunak. UML adalah metodologi untuk mengembangkan sistem berorientasi objek dan alat untuk mendukung pengembangan sistem.

Berdasarkan oleh (Nugroho, 2009), UML merupakan kolaborasi metodologi yang membawa semua method menjadi satu kesatuan, Teknik pembuatan sebuah objek, dimana biasanya digunakan oleh teknisi atau perancang aplikasi untuk mendiskripsi kan dan mengalisa bagaimana sebuah sistem berjalan sebelum dibuat..

2.5.1. *Use Case Diagram*

Istilah "penggunaan kasus" mengacu pada teknik yang digunakan dalam pengembangan perangkat keras komputer dan pembuatan sistem informasi untuk mendokumentasikan persyaratan fungsional yang diperlukan dari suatu sistem. Kasus penggunaan menggambarkan interaksi antara "aktor," atau

"inisiator," dan sistem itu sendiri. Dalam Sistem Yang Ada, Kasus Gunakan Direpresentasikan Gangan Urutan Langkah-Langkah Sederhana.

Perilaku sistem adalah bagaimana sistem bekerja dan bereaksi. Perilaku sistem ini direkam dalam use case. The use case adalah bentuk dari dua buah entitas dimana ada aktor dan fungsi yang dapat dilakukan aktor tersebut. Manfaat dari kasus penggunaan adalah:

Digunakan untuk berkomunikasi dengan pengguna akhir dan pakar domain. Digunakan untuk mengidentifikasi siapa yang berinteraksi dengan sistem dan apa yang Anda ingin system lakukan.

Dari segi fungsi, fungsi use case adalah sebagai berikut:

Use case adalah interaksi atau dialog antara sistem dan aktor, termasuk pertukaran pesan dan aksi dari system.


Use case diprakarsai oleh aktor dan dapat mencakup peran aktor lain. Kasus penggunaan harus memberikan nilai minimum untuk satu aktor.



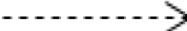





Kasus penggunaan dapat memiliki ekstensi yang menentukan tindakan spesifik dalam suatu interaksi, atau kasus penggunaan lain dapat ditambahkan.

Kelas use case memiliki objek use case yang disebut skenario. Skenario mewakili serangkaian pesan dan tindakan.

Untuk mendukung pembuatan use case, berikut penjelasan komponen-komponen yang termasuk dalam use case:

Tabel 2. 1 Komponen Use Case

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan seseorang yang berinteraksi dengan sistem, yang hanya dapat memasukkan data dan menerima data dari sistem dan tidak mengontrol use case. Dan biasanya sang aktor difoto dengan figur stickman.

2		<i>Use case</i>	Deskripsi fungsional dari sistem yang akan diluncurkan sehingga pengguna dapat lebih memahami cara menggunakan sistem.
3		<i>Association</i>	Tautan antara elemen yang menghubungkan aktor dan use case.
4		<i>Generalization</i>	Suatu elemen yang menjadi spesialisasi dari elemen lain. elemen yang entah bagaimana bergantung pada elemen lain.
5		<i>Dependency</i>	Menunjukkan bahwa use case adalah sumber eksplisit.
6		<i>Include</i>	Menunjukkan bahwa use case target memperluas perilaku use case pada titik tertentu.
7		<i>Extend</i>	Menentukan paket yang berisi sistem terbatas.
8		<i>System</i>	Interaksi aturan dan elemen lain yang bekerja bersama untuk menciptakan perilaku lebih besar daripada jumlah elemen. Elemen fisik yang ada saat aplikasi dijalankan dan mewakili sumber daya komputasi.
9		<i>Collaboration</i>	Menggambarkan seseorang yang berinteraksi dengan sistem, yang hanya dapat memasukkan data dan menerima data dari sistem dan tidak mengontrol use case. Dan biasanya sang aktor difoto dengan figur stickman.
10		<i>Note</i>	Deskripsi fungsional dari sistem yang akan diluncurkan sehingga pengguna dapat lebih memahami cara menggunakan sistem.

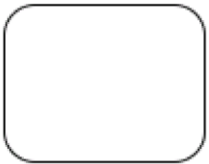




2.5.2. The Activity Diagram

Diagram aktivitas adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam artian, diagram hampir identik dengan aliran data, tetapi perbedaan prinsip antara diagram aktivitas mendukung perilaku yang lebih paralel (Rijal, 2010).

(Sukamto & Shalahuddin, 2013), “Diagram bawah atau diagram aktivitas menggambarkan alur kerja atau pengoperasian suatu sistem atau proses bisnis atau menu perangkat lunak”.

Diagram aktivitas adalah diagram yang menggambarkan operasi sistem. Diagram ini menunjukkan dari awal hingga akhir langkah-langkah apa saja yang terjadi dalam proses kerja sistem yang akan dibangun. Oleh karena itu, penggunaan bagan fungsional dapat mempermudah proses produksi sistem.

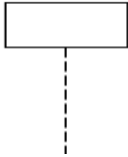
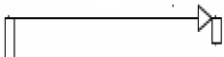

Tabel 2. 2 Komponen *Activity Diagram*

No.	Gambar	Nama	Keterangan
1		<i>Activity</i>	Show how each UI module communicates with the others.
2		<i>Action</i>	Operation completion status as determined by the system.
3		<i>Initial Node</i>	Method of production or object genetics.
4		<i>Activity Final Node</i>	How things are made and destroyed.
5		<i>Fork Node</i>	Converging beams of light form many beams at a same location.

2.5.3. The Sequence Diagram

Diagram urutan adalah representasi grafis dari perilaku algoritma, seperti tampilan sejumlah objek contoh dan pesan yang melewati objek -objek tersebut dalam kasus penggunaan tertentu (Rijal, 2010).

Tabel 2. 3 Komponen *Sequences Diagram*

No.	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objects that interact with one another are called "entities" or "antarmuka."
2		<i>Message</i>	Definition of inter-object communication; includes details on ongoing operations.
3		<i>Message</i>	Object-to-object communication definition: data about ongoing operations

2.5.4. The Class Diagram

Menurut (Sukamto & Shalahuddin, 2013), "diagram kelas" (juga dikenal sebagai "diagram kelas") menggambarkan arsitektur sistem dengan menguraikan kelas yang digunakan untuk membangunnya. Sebuah sekolah dapat didefinisikan oleh karakteristiknya dan metode operasinya.

Atribut adalah jenis variabel yang dimiliki kelompok tertentu.

Metode dan prosedur adalah kemampuan yang dimiliki kelompok tertentu.

Jenis -jenis kelas berikut harus ditampilkan pada diagram kelas yang baik untuk mendapatkan hasil yang diinginkan dalam hal Tata Letak Struktur Kelas.


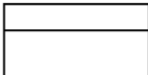




a) Kelompok inti fungsi utama dari gugus node pertama sistem.

Grup Desain Sistem (B) (Tampilan). grup yang bertanggung jawab untuk mendefinisikan dan mengatur antarmuka pengguna.

Gunakan definisi kasus berfungsi sebagai dasar untuk Kelas (pengontrol). Kelompok yang menangani proses bisnis dari komputer kadang -kadang disebut sebagai "tim manajemen proses" (atau "tim proses"), dan mereka bertanggung jawab untuk menentukan fungsi apa yang harus dimasukkan berdasarkan definisi kasus penggunaan.

Definisi data yaitu yang digunakan untuk mendapatkan Kelas (model). Modul Penyimpanan dan Pengambilan Informasi.

Tabel 2. 4 Komponen *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Nary Association</i>	Hindari penggabungan dengan lebih dari dua objek.
2		<i>Class</i>	Kumpulan objek yang memiliki atribut dan fungsi yang sama.
3		<i>Association</i>	Asosiasi antar kelas yang memiliki arti umum biasanya melibatkan himpunan.
4		<i>Composition</i>	Ketika sebuah kelas tidak dapat berdiri sendiri dan harus menjadi bagian dari kelas lain, kelas tersebut memiliki hubungan komposisi dengan kelas yang bergantung padanya.
5		<i>Dependency</i>	Hubungan antar kelas menurut ketergantungan antar kelas. Hubungan antar kelas dengan makna universal
6		<i>Aggregation</i>	Hindari penggabungan dengan lebih dari dua objek.

2.6. Rumah Sakit Umum Daerah Srengat Blitar

RSUD Srengat Blitar menganut konsep ekologis atau standar ekologis rumah sakit (Kalel C) dan menyediakan empat layanan utama: pediatri, ginekologi, kedokteran internal, dan perawatan khusus untuk orang sakit. Rumah sakit ini akan memiliki 100 tempat tidur tersebar di empat lantai. Pasien yang membutuhkan perawatan intensif, perawatan darurat, operasi, perawatan intensif, laboratorium, dan area khusus lainnya akan dirawat di rumah sakit. Ada sejumlah bangunan keagamaan di daerah tersebut, termasuk masjid, tempat ibadah, kuil, dan rumah tradisional. Moto perusahaan adalah "kebahagiaan dan keamanan Anda adalah prioritas utama kami."