

## BAB II

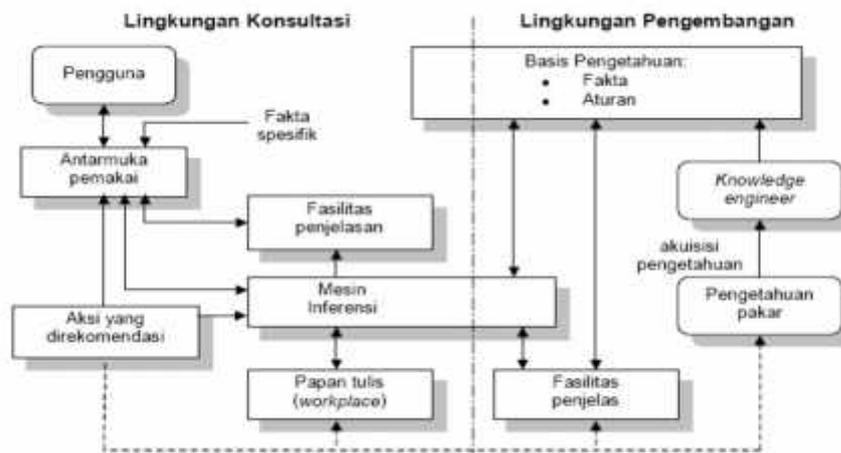
### LANDASAN TEORI

#### 2.1 Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*Development environment*) dan lingkungan konsultasi (*Consultation environment*) (Turban, 2005).

1. Lingkungan pengembangan yang digunakan dalam sistem pakar untuk membangun komponen-komponennya dan menempatkan pengetahuan dalam basisnya.
2. Lingkungan konsultasi yang digunakan oleh pemakai untuk mendapatkan informasi atau pengetahuan dari pakar.

Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam Gambar 2.1 di bawah ini :



Gambar 2. 1 Arsitektur Sistem Pakar

Dari penjelasan tersebut, dapat disimpulkan bahwa sistem pakar mempunyai dua bagian pokok didalamnya. Yaitu pengembangan dan konsultasi, pengembangan ditujukan pada bagian dimana sistem pakar itu dibuat dari mulai membangun komponen sampai dengan proses mentransfer pengetahuan pakar ke sistem pakar tersebut. Sedangkan bagian konsultasi yaitu bagian dimana sistem pakar digunakan oleh pengguna.

Sistem pakar yang akan dibangun berkaitan dengan diagnosa tingkatan stres pada remaja.

### **2.1.1 Tahapan Pembuatan Sistem Pakar**

Ada beberapa tahapan dalam pembuatan sistem pakar, seperti yang telah di jelaskan oleh (Kusemadewi, 2003) antara lain :

#### **1. Identifikasi Masalah**

Merupakan tahapan untuk mengidentifikasi permasalahan yang dikaji dalam hal ini ditentukan batasan masalah yang dikaji, pakar yang akan terlibat & tujuan yang akan dicapai. Sebaiknya permasalahan yang dikaji bersifat sempit tetapi kompleks (mendalam), pada tahap ini *knowledge engineer* dapat menentukan masalah secara umum memilih pakar dahulu kemudian berdiskusi dengan pakar untuk menentukan batasan masalah yang dikaji, pakar yang dipilih yaitu dokter kejiwaan yang terdapat pada RS Jiwa Bandar Lampung.

#### **2. Konseptualisasi**

Merupakan tahapan dimana *knowledge engineer* & pakar menentukan konsep yang akan dikembangkan menjadi sistem pakar. Dari konsep tersebut dirinci seluruh unsur-unsur yang terlibat dan dikaji hubungan antar unsur serta mekanisme pengendalian yang diperlukan untuk mencapai solusi.

### 3. Formalisasi

Merupakan hubungan antara unsur-unsur digambarkan dalam bentuk formal yang biasa digunakan oleh sistem pakar, struktur data dan bentuk inferensi ditentukan. Pada tahap ini juga *knowledge engineer* menentukan alat pembangun sistem pakar yang akan digunakan.

### 4. Implementasi

Tahap ini dimasuki setelah formalisasi hubungan antara unsur-unsur telah dilakukan secara lengkap dan alat pembangun yang sesuai telah dipilih. Dalam tahap ini *knowledge engineer* menterjemahkan bentuk hubungan antar unsur kedalam bahasa komputer.

### 5. Pengujian

Merupakan tahap akhir terhadap sistem pakar yang telah dibangun. Perbaikan selalu di lakukan setiap tahapan agar diperoleh sistem yang lengkap dan akurat. Pada pengujian, ini dilakukan menggunakan pengujian *blackbox* yang mana pengujian dilakukan menitik beratkan pada test stres program dan secara fungsionalitas pada *interface*.

## 2.2 Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Ada 2 bentuk pendekatan basis pengetahuan yang umum digunakan (Arhami, 2005) yaitu :

#### a. Penalaran berbasis aturan (*Rule Based Reasoning*)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk IF-THEN. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah pencapaian solusi).

b. Penalaran berbasis kasus (*Case Based Reasoning*)

Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan. ( Arhami, 2005).

Yang digunakan pada pembuatan sistem pakar ini yaitu penalaran berbasis kasus. Sebagai contoh dalam database telah dilakukan perekaman terhadap beberapa kasus yang telah terjadi (ini disebut dengan kasus lama). Kemudian ketika dilakukan perekaman data pada kasus baru, maka kasus baru akan dibandingkan nilai kedekatannya dengan kasus lama (nilai *similarity*). Dari beberapa perbandingan dengan kasus lama, nilai *similarity* yang paling besar yang akan menjadi hasil dari kasus baru.

### **2.3 Case Based Reasoning (CBR)**

Pada skripsi ini, pembuatan sistem pakar menggunakan metode *case base reasoning*. Sistem penalaran komputer berbasis kasus (*case base reasoning/CBR*) merupakan sistem yang bertujuan untuk menyelesaikan suatu kasus baru dengan cara mengadaptasi solusi-solusi yang terdapat kasus-kasus sebelumnya yang mirip dengan kasus baru tersebut (Andriana, 2008).

Keuntungan sistem penalaran komputer berbasis kasus (Andriana, 2008) yaitu :

- a) Mengurangi dampak penambahan informasi pengetahuan, karena tidak memerlukan pemahaman bagaimana menyelesaikan masalah.
- b) Tidak memerlukan suatu model yang eksplisit dan pengetahuan didapat dengan cara mengumpulkan kejadian-kejadian yang telah terjadi.
- c) Kemampuan untuk belajar dengan menambahkan kasus baru seiring waktu tanpa perlu menambahkan aturan baru atau mengubah yang sudah ada.
- d) Kemampuan untuk mendukung justifikasi dengan menawarkan kasus yang sudah ada sebelumnya lebih diutamakan.

### 2.3.1 Tahapan *Case Base Reasoning* (CBR)

*Case Based Reasoning* (CBR) menggunakan pendekatan kecerdasan buatan (*Artificial Intelligent*) yang menitik beratkan pemecahan masalah dengan didasarkan pada knowledge dari kasus-kasus sebelumnya. (Andriana, 2008).

Secara umum, metode ini terdiri dari 4 langkah, yaitu :

#### 1) *Retrieve*

Pada saat terjadi permasalahan baru, pertama-tama sistem akan melakukan proses *Retrieve*. Proses ini akan melakukan dua langkah pemrosesan, yaitu pengenalan masalah dan pencarian persamaan masalah tingkat stress kasus lama yang tersimpan pada database.

#### 2) *Reuse*

Proses ini sistem akan menggunakan informasi kasus stress sebelumnya yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru dan menggunakan kembali informasi dan pengetahuan dalam kasus tersebut untuk mengatasi masalah. Pada proses *Reuse* akan menyalin, menyeleksi, dan melengkapi informasi yang akan digunakan.

#### 3) *Revise* (meninjau ulang solusi yang diajukan).

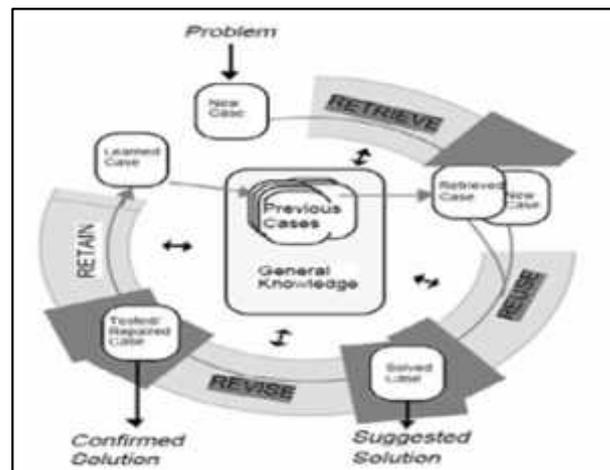
Proses ini informasi tersebut akan dikalkulasi, dievaluasi, dan diperbaiki kembali untuk mengatasi kesalahan-kesalahan yang terjadi pada permasalahan baru. Pada *revise*, proses hasil kalkulasi akan dilakukan evaluasi secara langsung dengan cara hasil kalkulasi nilai *similarity* dari kasus pasien baru ditampilkan kembali oleh sistem.

#### 4) *Retain*

Proses ini akan mengindeks, mengintegrasikan, dan mengekstrak solusi yang baru. Selanjutnya solusi baru itu akan disimpan ke dalam *knowledge-base* untuk menyelesaikan permasalahan yang akan datang. Tentunya permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan dengannya. Setelah dilakukan *retain*, data kasus baru pada pendiagnosa pasien yang tersimpan

maka akan menjadi pembanding (sebagai kasus lama) di proses diagnosa berikutnya.

Langkah- langkah tersebut dilihat pada Gambar 2.2 (Adriana,2008) di bawah ini :



Gambar 2. 2 Siklus Case Base Reasoning

### 2.3.2 Algoritma Nearest Neighbor

Dalam metode CBR yang akan dirancang pada skripsi ini, proses *retrieve* (penelusuran kasus) pada sistem penalaran komputer berbasis kasus untuk pendeteksi tingkat stress menggunakan *algoritma nearest neighbor*. *algoritma nearest neighbor* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur-fitur yang ada.

Adapun rumus untuk melakukan perhitungan kedekatan antara dua kasus adalah sebagai berikut :

Nilai similarity setiap variabel (Kusrini dan Lutfi, 2009)

$$\text{Similarity (T,S)} = \frac{\sum_{i=1}^n f(T_i, S_i) \times W_i}{W_i}$$

Keterangan :

T = kasus baru

S = kasus yang ada dalam penyimpanan

n = jumlah atribut dalam setiap kasus

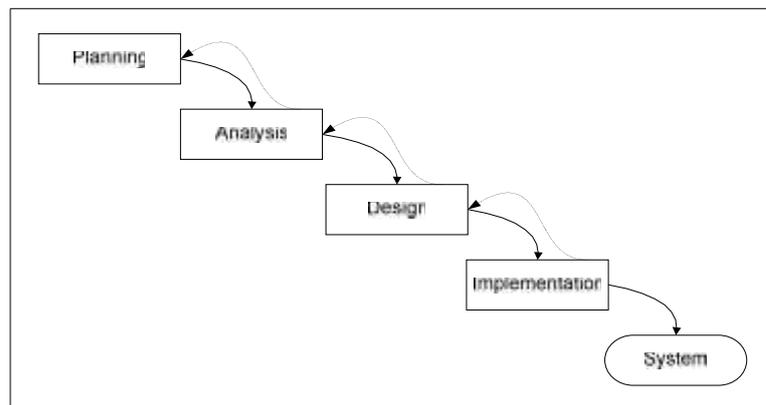
i = atribut individu antara 1 s.d. n

$f$  = fungsi similarity untuk fitur I dalam kasus T dan kasus S

W = bobot yang diberikan pada atribut ke-i

## 2.4 Metode Pengembangan Sistem

Pada metode penelitian ini dilakukan rekayasa perangkat lunak yang digunakan adalah model *Waterfall* seperti pada gambar di bawah ini:



Gambar 2. 3 Metode Pengembangan Model Waterfall

Keterangan:

### 1. *Planning* (Perencanaan)

Tahap perencanaan merupakan proses penting untuk mengetahui mengapa sistem harus dibuat dan menentukan bagaimana cara membangun sistem tersebut.

Sistem harus dibuat agar memudahkan dalam menganalisis tingkat stres belajar pada siswa SMA. Hal ini diperlukan karena pada masa remaja tingkat stres sangat rentan terjadi.

2. *Analysis (Analisis)*

Analisis sistem dilakukan untuk memberikan jawaban pertanyaan siapa yang akan menggunakan sistem. Apa yang akan dilakukan oleh sistem, dimana dan kapan sistem tersebut digunakan. Pada tahap ini pembuat sistem akan melakukan observasi dan pengamatan terhadap sistem yang lama, kemudian mengidentifikasi, memanfaatkan dan mengembangkan peluang, dan membangun konsep untuk sebuah sistem baru.

3. *Design (Perancangan)*

Tahap perancangan dilakukan untuk menetapkan bagaimana sistem akan dioperasikan. Hal ini berkaitan dengan menentukan perangkat keras, perangkat lunak, jaringan, tampilan program, form dan laporan yang akan dipakai. Selain itu perlu juga menspesifikasi program, database dan file yang dibutuhkan. Sistem pakar dibangun menggunakan penalaran berbasis kasus, dibuat dengan bahasa pemrograman Delphi 7 dan Ms. Access sebagai databasenya.

4. *Implementation*

Merupakan tahap berikutnya untuk menerjemahkan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang telah ditentukan. Semua tahap ini desain perangkat lunak sebagai sebuah program lengkap atau unit program.

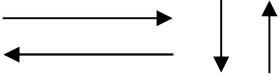
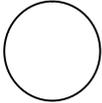
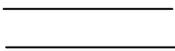
5. *System*

Tahapan ini, merupakan hasil sistem yang telah dibuat dalam bentuk perangkat lunak yang telah dipasang dan digunakan, termasuk didalamnya proses pemeliharaan dan perbaikan kesalahan. Perangkat lunak yang telah selesai dibuat dapat mengalami perubahan-perubahan atau penambahan sesuai dengan permintaan user atau perubahan sistem.

## 2.5 Data Flow Diagram (DFD)

Dalam merancang alur data pada skripsi ini, penulis menggunakan *data flow diagram*.

*Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan proses tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Jogiyanto, 2005).

Simbol	Keterangan
<p>(<i>external entity</i>)</p> 	Merupakan sumber atau tujuan dari aliran data dari atau ke sistem.
<p>Arus data (<i>data flow</i>)</p> 	Menggambarkan arus data.
<p>Proses (<i>process</i>)</p> 	Merupakan kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.
<p>Simpanan data (<i>data storage</i>)</p> 	Merupakan komponen yang berfungsi untuk menyimpan data atau file.

Tabel 2. 1 Simbol untuk DFD

## 2.6 Basis Data (*Database*)

Penyimpanan data pada sistem pakar yang dibahas pada skripsi ini yaitu *database*. Adapun *database* yang akan digunakan adalah *MS. Access 2007*.

*Database* adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu (Kristanto, 2003). Sekumpulan tanda-tanda yang berbentuk kesatuan tersendiri .

Istilah-istilah yang digunakan dalam basis data :

- 1) File : merupakan kumpulan dari atribut record-record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda-beda dalam data value-nya.
- 2) Record : merupakan kumpulan dari elemen-elemen yang saling berhubungan atau berkaitan menginformasikan tentang entry secara lengkap.
- 3) Field : merupakan, merupakan bagian terkecil dari record dan bentuknya unik dijadikan field kunci yang dapat mewakili record-nya.

Entity : merupakan tempat kejadian atau konsep yang informasikan direkam.