

## **BAB II KAJIAN PUSTAKA**

Sistem pakar diagnosis penyakit pernapasan pada balita dibangun dengan merujuk pada beberapa teori. Teori yang dideskripsikan adalah sistem pakar, *Data Flow Diagram*, basis data, *PHP*, *MySQL*, *dreamweaver*, metode pengembangan sistem, metode uji coba, dan penyakit pernapasan pada balita.

### **A. Sistem Pakar**

#### **1. Kecerdasan Buatan**

Menurut H. A. Simon dalam Kusri (2006: 3), kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia. Menurut John McCarthy dalam Desiani & Arhami (2006: 3) kecerdasan buatan adalah ilmu untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Sesuai dengan definisi tersebut, maka ilmu kecerdasan buatan ini dapat diaplikasikan ke berbagai bidang seperti: Robotika (*Robotics*), Penglihatan Komputer (*Computer Vision*), Pengolahan Bahasa Alami (*Natural Language Processing*), Pengenalan Pola (*Pattern Recognition*), System Syaraf Buatan (*Artificial Neural System*), Pengenalan Suara (*Speech Recognition*) dan Sistem Pakar (*Expert System*).

Menurut Winston dan Prendergast dalam Kusri (2006:4), ada tiga tujuan kecerdasan buatan, yaitu: membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin lebih berguna. Kecerdasan adalah

kemampuan untuk belajar atau mengerti dari pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif.

## **2. Sistem Pakar**

### **a. Konsep dasar sistem pakar**

Sistem pakar adalah bagian dari kecerdasan buatan yang merupakan perangkat lunak untuk memecahkan masalah yang biasanya diselesaikan oleh seorang pakar atau ahli dan berisi aturan-aturan bagaimana memberlakukan informasi-informasi yang tersimpan, sehingga program dapat memberikan bantuan pengambilan keputusan mengenai suatu permasalahan tertentu, sebagaimana seorang pakar. Pakar atau ahli (*expert*) didefinisikan sebagai seseorang yang memiliki pengetahuan atau keahlian yang tidak dimiliki oleh kebanyakan orang (Rosnelly, 2012: 8). Sistem pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil keputusan yang diperoleh (Kusrini, 2008: 3).

### **b. Ciri-ciri sistem pakar**

Menurut Kusrini (2006: 14) sistem pakar yang baik haruslah memiliki ciri-ciri sebagai berikut:

- terbatas pada bidang yang spesifik,
- dapat memberikan penalaran untuk data–data yang tidak lengkap atau tidak pasti,

- dapat mengemukakan alasan yang diberikannya dengan bahasa yang dapat dipahami,
- bekerja berdasarkan kaidah atau aturan (*rule*) tertentu,
- dirancang dengan tujuan dapat dikembangkan secara bertahap,
- keluaran atau outputnya berisi anjuran atau nasihat,
- sistem dapat mengarahkan pengguna kepada *output*, tergantung dari dialog pengguna dan sistem,
- basis pengetahuan dan mekanisme inferensinya terpisah.

### **c. Kelebihan dan kekurangan sistem pakar**

Secara garis besar, banyak sekali keuntungan yang didapatkan dengan adanya sistem pakar, diantaranya sebagai berikut.

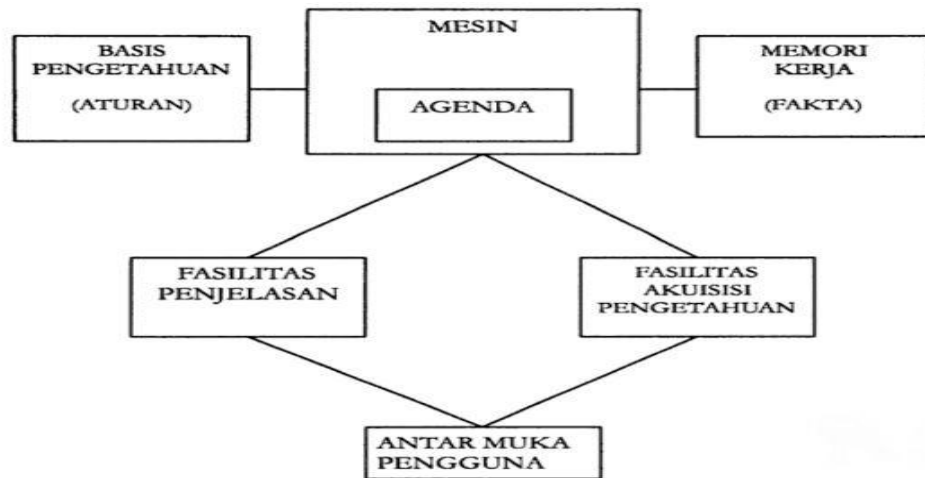
- Menghimpun data dalam jumlah yang sangat besar.
- Menyimpan data tersebut untuk jangka waktu yang panjang.
- Meningkatkan output dan produktivitas karena sistem pakar dapat bekerja lebih cepat dibandingkan manusia.
- Mempermudah pencarian pengetahuan dan nasihat yang diperlukan.
- Dapat bekerja dengan data yang kurang lengkap dan tidak pasti.
- Sistem pakar tidak dapat lelah dan bosan.
- Memberikan respons (jawaban) yang cepat.

Di samping memiliki kelebihan, sistem pakar juga memiliki kekurangan layaknya sistem lain (Desiani & Arhami, 2006 : 11), diantaranya adalah.

- Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu mudah diperoleh karena kadang kala pakar dari masalah tersebut tidak ada atau terkadang pendekatan yang dimiliki para ahli berbeda-beda.
- Membuat suatu sistem pakar yang benar-benar berkualitas sangatlah sulit dan memerlukan biaya yang tidak sedikit.
- Dapat jadi sistem pakar tidak dapat membuat keputusan
- Sistem pakar tidaklah 100% menguntungkan karena tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan.

#### d. Struktur sistem pakar

Struktur Sistem Pakar dapat dilihat pada Gambar 1.



**Gambar 1:** Struktur Sistem Pakar (sumber: Rosnelly, 2012)

Penjelasan dari komponen-komponen di atas adalah sebagai berikut.

##### 1. Basis Pengetahuan (*knowledge base*).

Basis pengetahuan merupakan komponen yang mengandung pengetahuan, pemahaman, formulasi, dan penyelesaian masalah.

Komponen sistem pakar terdiri dari dua elemen dasar, yakni fakta dan aturan. Fakta adalah informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan adalah informasi tentang bagaimana cara memperoleh fakta baru dari fakta yang telah diketahui (Desiani & Arhami, 2006: 234).

## 2. Mesin Inferensi (*Inference Engine*).

Mesin inferensi adalah bagian dari komputer yang bertindak sebagai otak dari sebuah sistem pakar. Komponen ini mengandung mekanisme fungsi berpikir dan pola penalaran. Metode ini terbagi menjadi dua, yaitu *forward chaining* (runut maju) dan *backward chaining* (runut balik).

### a. *Forward Chaining*

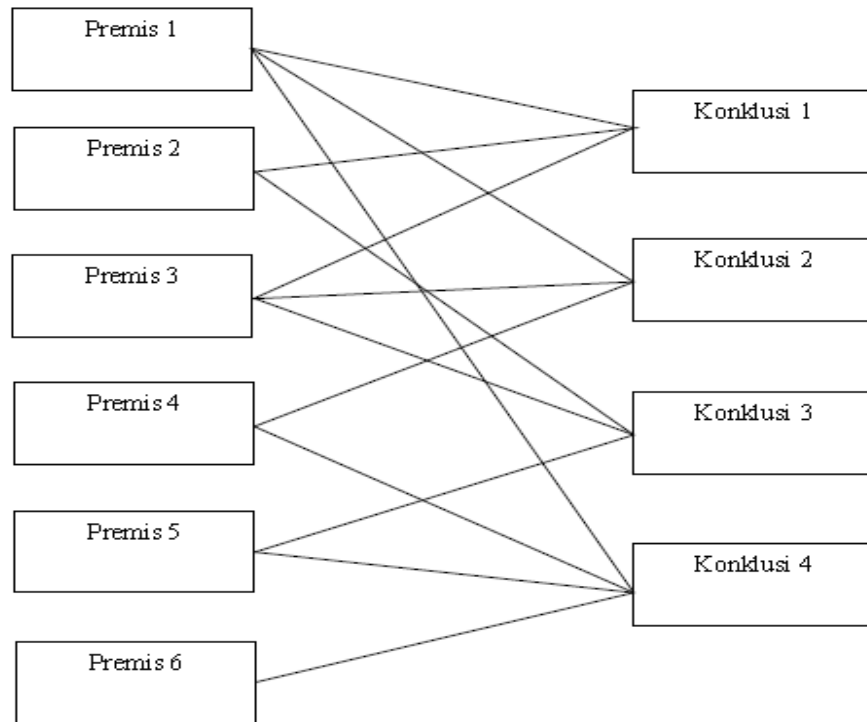
Pada metode ini, penalaran dimulai dari fakta-fakta yang ada (IF) kemudian bergerak maju melalui *premis-premis* untuk menuju kesimpulan (*THEN*). Metode runut maju ini digunakan untuk mendiagnosis suatu penyakit berdasarkan gejala-gejala yang ada. Cara kerja metode *forward chaining* yaitu seperti pada Gambar 2 di bawah ini.



**Gambar 2.** Cara Kerja Metode *Forward Chaining*

Gambar 3 adalah ilustrasi untuk mempermudah pemahaman tentang metode inferensi. Dalam penalaran ini, *user* diminta memasukkan premis-premis yang dialami. Seandainya *user* memilih Premis 1, Premis 2, dan Premis 3 maka aturan yang terpilih adalah aturan 1 dengan konklusinya adalah Konklusi 1. Seandainya *user* memilih Premis 1 dan Premis 6, maka

sistem akan mengarah pada aturan empat dengan konklusinya adalah Konklusi 4, tetapi karena aturan tersebut premisnya adalah Premis 1, Premis 4, Premis 5, dan Premis 6 maka premis-premis yang dipilih oleh *user* tidak cukup untuk mengambil kesimpulan Konklusi 4 sebagai konklusi terpilih.



**Gambar 3.** Graf Pengetahuan

b. *Backward Chaining*

Metode ini membantu dalam menangani permasalahan di mana konklusinya telah diketahui sebelumnya dan penyebab dari konklusi tersebut yang kemudian dicari. Penelusuran didasarkan pada suatu keyakinan bahwa ada kemungkinan konklusi dari daftar konklusi merupakan salah satu tujuan atau konklusi terpilih berdasarkan fakta yang diberikan *user*. Cara kerja metode ini dapat dilihat pada Gambar 4.



**Gambar 4.** Cara Kerja Metode *Backward Chaining*

Menggunakan Gambar 3, sistem dengan urutan tertentu akan mengambil sebuah konklusi sebagai calon konklusinya. Misal awalnya sistem akan mengambil hipotesis bahwa konklusinya adalah Konklusi 1. Untuk membuktikannya, sistem akan mencari premis-premis aturan yang mengandung Konklusi 1. Setelah itu sistem akan meminta umpan balik kepada *user* mengenai premis-premis yang ditemukan. Untuk Konklusi 1, premisnya adalah premis 1, premis 2, dan premis 3, maka sistem akan mencari tahu apakah *user* memilih premis-premis tersebut.

3. Memori Kerja (*Working Memory*).

Memori kerja adalah area penyimpanan fakta yang dihasilkan oleh mesin inferensi dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai *global database* dari fakta yang digunakan oleh aturan-aturan yang ada (Rika Rosnelly, 2012: 13).

4. Fasilitas Penjelasan (*Explanation Facility*).

Fasilitas penjelasan adalah komponen yang berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Menurut Turban dalam Desiani&Arhami (2006: 239), fasilitas penjelasan dapat menjelaskan bagaimana cara berpikir sistem dengan menjawab pertanyaan berikut:

- a) mengapa pertanyaan tertentu ditanyakan oleh sistem pakar?
- b) bagaimana kesimpulan tertentu dapat diperoleh?

- c) mengapa alternatif tertentu ditolak?
- d) apa rencana untuk memperoleh penyelesaian?

5. Fasilitas Akuisisi Pengetahuan (*Knowledge Acquisition Facility*).

Fasilitas ini merupakan suatu proses pengumpulan data-data pengetahuan suatu masalah dari pakar, meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan yang terdokumentasi ke program komputer yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

6. Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna (*user interface*) adalah komponen yang digunakan pengguna dan sistem untuk saling berkomunikasi. Antarmuka menerima informasi dari pengguna dan mengubahnya ke dalam bentuk yang dimengerti oleh sistem. Oleh sistem, informasi tersebut akan diolah kemudian diteruskan ke antarmuka. Oleh antarmuka, informasi tersebut diubah terlebih dulu menjadi informasi yang dapat dimengerti pengguna.

**3. Representasi Pengetahuan**

Pengetahuan merupakan intisari dari sebuah informasi, dapat berisi fakta, informasi, konsep, prosedur, model, dan heuristik yang dapat digunakan untuk menyelesaikan suatu persoalan. Pengetahuan diklasifikasikan menjadi tiga, yaitu: (a) pengetahuan *procedural* (*procedural knowledge*), adalah pengetahuan yang lebih menekankan pada bagaimana melakukan sesuatu; (b) pengetahuan deklaratif (*declarative knowledge*) yaitu menjawab pertanyaan



apakah sesuatu bernilai salah atau benar; dan (c) pengetahuan tacit (*tacit knowledge*), yaitu pengetahuan yang tidak dapat diungkapkan dengan bahasa.

Menurut Kusrini (2008: 6), representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah. Adapun karakteristik dari metode representasi pengetahuan adalah sebagai berikut:

- a. harus dapat diprogram dan hasilnya disimpan dalam memori,
- b. dirancang sedemikian sehingga isinya dapat digunakan untuk proses penalaran,
- c. model representasi pengetahuan merupakan sebuah struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencarian untuk aktivitas pencocokan pola.

Menurut Kusrini (2006: 24-25), representasi pengetahuan dapat dimodelkan menjadi:

- a. Logika (*logic*)

Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran. Dalam melakukan penalaran, komputer harus dapat menggunakan proses penalaran deduktif (penalaran umum ke khusus) dan proses penalaran induktif (penalaran khusus ke umum).

Berikut contoh penalaran deduktif:

*Premis mayor: Jika hari ini saya sakit, saya tidak akan berangkat sekolah.*

*Premis minor: Hari ini saya sakit.*

*Konklusi: Hari ini saya tidak berangkat sekolah.*

Berikut contoh penalaran induktif:

*Premis: Dioda yang salah menyebabkan alat elektronik rusak.*

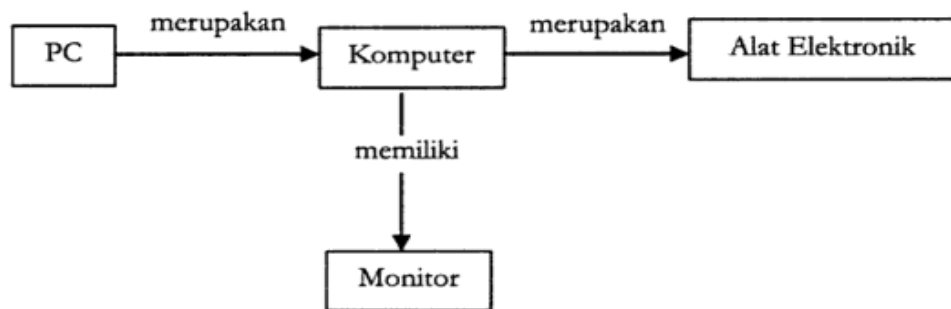
*Premis: Transistor rusak menyebabkan peralatan elektronik rusak.*

*Premis: Dioda dan transistor merupakan peralatan semikonduktor.*

*Konklusi: Peralatan semikonduktor yang rusak menyebabkan peralatan elektronik rusak.*

b. Jaringan semantik (*semantic nets*)

Jaringan semantik merupakan penggambaran grafis dari pengetahuan yang memperlihatkan hubungan antar obyek. Komponen dasar dari jaringan semantik (Gambar 5), yaitu simpul (*node*) dan penghubung (*arc/link*). *Node* menggambarkan objek, konsep, atau situasi sedangkan *arc/link* menggambarkan hubungan antar *node*. Contoh jaringan semantik:



**Gambar 5.** Representasi Jaringan Semantik

Gambar 5 merepresentasikan pernyataan bahwa semua PC merupakan komputer, semua komputer merupakan alat elektronik, dan

semua komputer memiliki monitor. Dari pernyataan tersebut dapat diketahui bahwa semua PC memiliki monitor dan hanya sebagian alat elektronik saja yang memiliki monitor.

c. *Object-Attribute-Value* (OAV)

*Object* dapat berupa fisik atau konsep. *Attribute* adalah karakteristik dari objek tersebut. *Value* adalah besaran/nilai/takaran spesifik dari *attribute* tersebut pada situasi tertentu, dapat berupa numerik, string atau konstan. Contoh representasi pengetahuan dengan menggunakan OAV:

**Tabel 1.** Representasi Pengetahuan dengan OAV

<b>Object</b>	<b>Attribute</b>	<b>Value</b>
Mangga	Warna	Hijau, Orange
Mangga	Berbiji	Tunggal
Mangga	Rasa	Asam, Manis
Pisang	Warna	Kuning, Hijau
Pisang	Bentuk	Lonjong

Pada Tabel 1, obyek yang dibahas adalah mangga. Mangga ini memiliki beberapa atribut (karakteristik dari obyek) salah satunya adalah atribut warna. Warna pada mangga memiliki *value* yakni hijau atau orange. Karena objek di sini memiliki beberapa atribut, maka objek ini disebut dengan OAV *multi-attribute*.

d. Bingkai (*frame*)

*Frame* berupa ruang-ruang (*slots*) yang berisi atribut untuk mendeskripsikan pengetahuan. Bingkai digunakan untuk merepresentasikan pengetahuan deklaratif. Contoh bingkai:

**Tabel 2.** Contoh Bingkai

<b>Ruang (slots)</b>	<b>Isi (filters)</b>
Nama	Flu
Gejala	1. Bersin 2. Pusing 3. Demam
Obat	1. Ultraflu 2. Mixagrip

e. Aturan produksi (*production rule*)

Aturan menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Aturan produksi ditulis dalam bentuk jika-maka (*if-then*). Aturan *if-then* mengandung anteseden (*antecedent*) dengan konsekuensi yang dihasilkannya. Anteseden mengacu kepada situasi yang terjadi sebelum konsekuensi dapat diamati.

Menurut Kusrini (2008: 7-8), aturan dalam kaidah produksi diklasifikasikan menjadi kaidah derajat pertama dan kaidah meta. Kaidah derajat pertama adalah aturan yang bagian konklusinya tidak menjadi premis bagi kaidah lain. Sebaliknya, kaidah meta merupakan kaidah yang berisi penjelasan bagi kaidah yang lain.

Contoh kaidah derajat pertama:

JIKA *Anemia*  
DAN *Batuk Kronis*  
MAKATBC


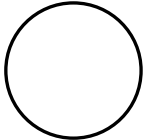

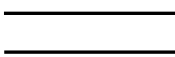
Contoh kaidah meta:

JIKA *Pusing*  
DAN *Cepat Lelah*

## B. Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) merupakan notasi-notasi yang digunakan untuk menggambarkan arus data sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data dan penyimpanan dari data tersebut (Jogiyanto Hartono, 2005: 701).

**Tabel 3.** Notasi Dasar DFD

Simbol	Keterangan
	Kesatuan Luar ( <i>External Entity</i> )
	Proses
	Arus Data
	Penyimpanan Data

Menurut Al-Bahra bin Ladjamuddin (2005: 67-75), elemen dasar dari DFD adalah sebagai berikut:

### 1. Kesatuan Luar (*External Entity*)

Kesatuan luar adalah sesuatu yang berada di luar sistem tetapi memberikan data ke dalam sistem atau sebaliknya. Kesatuan luar tidak termasuk bagian dari sistem. Pedoman kesatuan luar adalah sebagai berikut: nama kesatuan luar berupa kata benda dan kesatuan luar tidak boleh memiliki nama yang sama kecuali memang ada objeknya sama.

## 2. Proses (*Process*)

Proses merupakan kegiatan atau kerja yang dilakukan oleh sistem. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Pedoman pemberian nama proses adalah sebagai berikut:

- a. nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses,
- b. jangan menggunakan kata proses sebagai bagian dari nama suatu proses,
- c. tidak boleh ada beberapa proses yang memiliki nama yang sama,
- d. proses harus diberi nomor.

## 3. Simpanan Data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Pedoman pemberian nama simpanan data adalah sebagai berikut:

- a. nama harus mencerminkan simpanan data tersebut,
- b. bila namanya lebih dari satu kata, maka harus diberi tanda sambung.

## 4. Arus Data (*Data Flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Pedoman nama aliran data adalah sebagai berikut:

- a. nama aliran data yang terdiri dari beberapa kata dihubungkan dengan garis sambung,

- b. sedapatnya mungkin nama aliran data ditulis lengkap,
- c. tidak boleh ada aliran data dari kesatuan luar dan simpanan data atau sebaliknya, hubungan kesatuan luar dengan simpanan data harus melalui proses.

Menurut Agus Winarno (2007: 2) DFD dibagi menjadi 2 jenis.

a. Diagram Konteks (*Context Diagram*)

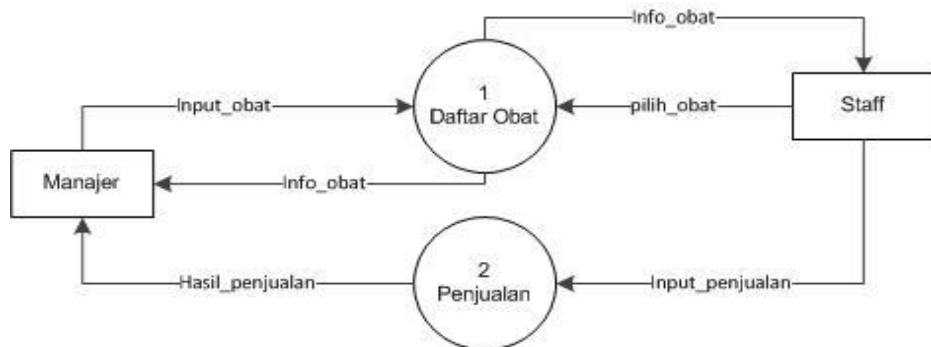
Diagram konteks merupakan DFD Level 0, yaitu diagram yang paling sederhana dari sebuah sistem informasi yang menggambarkan aliran data dari kesatuan luar ke dalam sistem dan sebaliknya.



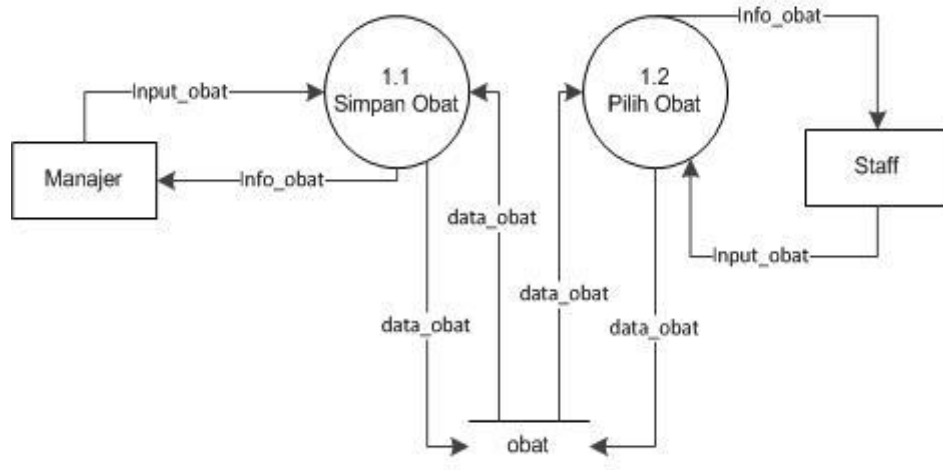
**Gambar 6.** Contoh Diagram Konteks

b. DFD Level n

DFD level n merupakan diagram yang digunakan untuk menggambarkan diagram hasil pengembangan dari diagram konteks ke dalam komponen yang lebih detail. Nilai n merupakan banyaknya angka/digit yang digunakan untuk penomoran proses yang ada.



**Gambar 7.** Contoh DFD Level 1



**Gambar 8.** Contoh DFD Level 2


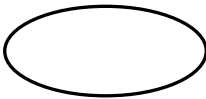
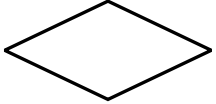

### C. Basis Data (*Database*)

Menurut Arief (2006: 33), *database* merupakan sekumpulan data yang saling terintegrasi satu sama lain dan terorganisasi berdasarkan sebuah skema atau struktur tertentu dan tersimpan pada sebuah *hardware* komputer. Di dalam sistem pakar, basis data merupakan tempat penyimpanan fakta-fakta kemudian ditambahkan dengan fakta baru yang diperoleh dari proses pelacakan oleh mesin inferensi. Selain itu basis data memiliki fungsi untuk mengelola data yang tersimpan di dalamnya seperti menambah, menghapus, melacak, dan lain sebagainya.

Model basis data relasional merupakan model basis data yang paling sederhana sehingga mudah untuk digunakan dan dipahami. Model basis data relasional ini umumnya dimodelkan menggunakan diagram relasi antar entitas yakni dengan *Diagram Entity-Relationship* (Diagram E-R).



**Tabel 4.** Notasi Dasar Diagram E-R (sumber: Yudi Priyadi, 2014: 20)

Simbol	Keterangan
	Persegi panjang, menunjukkan objek dasar
	Lingkaran/elips, menunjukkan atribut dari objek dasar
	Belah ketupat, menunjukkan relasi
	Garis, menunjukkan adanya relasi

Menurut Priyadi (2014: 21) penjelasan untuk notasi dasar pada Tabel 4 adalah sebagai berikut.

1. Entitas merupakan notasi untuk mewakili suatu objek dengan karakteristik yang sama, dilengkapi oleh atribut. Entitas biasanya berupa kata benda, pekerjaan, orang, dan tempat. Misalnya *user*, *admin*, dan *pakar* adalah contoh entitas.
2. Atribut adalah notasi yang menjelaskan karakteristik suatu entitas. Nama *user* atau asal *admin* merupakan contoh dari atribut. Atribut dapat berupa *key* yang bersifat unik seperti *primary key* dan *foreign key*.
  - a. Kunci primer (*primary key*)

Kunci primer adalah kunci yang dipilih sebagai kunci utamma dalam mengidentifikasi baris dalam tabel.
  - b. Kunci tamu (*foreign key*)

Kunci tamu adalah sebarang atribut yang menunjuk ke kunci primer pada tabel lain.

3. Relasi merupakan notasi dasar yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.
4. Garis penghubung merupakan notasi dasar yang digunakan untuk menunjukkan keterkaitan antara notasi-notasi yang digunakan dalam diagram E-R.

#### **D. PHP (*Personal Home Page tools*)**

PHP adalah bahasa pemrograman yang berjalan dalam sebuah web server dan berfungsi sebagai pengolah data. PHP termasuk dalam *HTML-embedded*, yang artinya kode PHP dapat disisipkan pada sebuah halaman HTML. Menurut Haryono (2004: 15), kode-kode bahasa PHP dalam penulisannya menyatu dengan tag-tag HTML dalam satu file. Kode PHP diletakkan antara tanda `<? atau <?php` dan diakhiri dengan tanda `?>` sebagai identitas bahasa pemrograman PHP. Akan tetapi, ada beberapa cara lagi untuk memberikan tanda bahwa kode yang ditulis adalah kode PHP, antara sebagai berikut.

1. Model javascript, php diawali dengan tag `<script language="php">` dan diakhiri dengan `</script>`.
2. Model ASP, penulisan kode php diawali dengan tag `<%` dan diakhiri dengan `%>`. Akan tetapi, cara ini berfungsi bila server diatur lebih dahulu sehingga modul php mengenalinya.

## E. MySQL

MySQL adalah salah satu jenis basis data *server* yang sangat terkenal disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses basis datanya. Selain itu MySQL bersifat gratis kecuali pada windows yang bersifat *shareware* atau perlu membayar setelah melakukan evaluasi dan memutuskan untuk digunakan pada keperluan produksi.

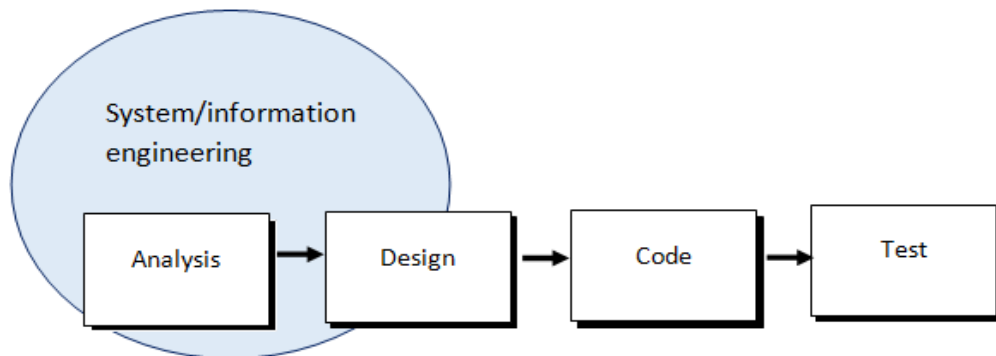
## F. Dreamweaver

Menurut M. Suyanto (2005: 244), Dreamweaver merupakan sebuah *software web design* yang menawarkan cara untuk mendesain website dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan memprogram. *Dreamweaver* memiliki satu jendela mini yang disebut HTML Source, tempat kode-kode HTML tertulis. Dreamweaver juga mampu mengenali *tag-tag* lain di luar HTML, seperti Cold Fussion dan ASP, serta mendukung *script-script* dinamik HTML dan CSS *style*. Selain itu, dreamweaver juga memberikan pilihan *workspace* baru yang terintegrasi, menyediakan lingkungan pengembangan yang lebih *familiar* dengan *dockable panel* yang lengkap dan jendela dokumen yang menyebut pemakai.

## G. Metode Pengembangan Sistem

Sistem pakar diagnosis penyakit pernapasan pada balita dibangun dengan sebuah metode analisis sistem aplikasi terstruktur yakni *Waterfall Model*. *Waterfall Model (Classic Life Cycle)* adalah model yang pertama kali

muncul yaitu sekitar tahun 1970. Model *Waterfall* sering dianggap kuno, tetapi paling sering digunakan dalam *Software Engineering* (SE).



**Gambar 9.** Model *Waterfall* Pressman

Pressman (2001: 28) menyatakan bahwa model *Waterfall* diuraikan dengan tahap-tahap sebagai berikut.

1. *Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan perancangan aplikasi.
2. *Design* adalah tahap penerjemah atau tahap perancangan dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
3. *Code* adalah tahap implementasi dari hasil aplikasi yang telah dirancang dalam bahasa pemrograman yang telah ditentukan dan digunakan dalam pembuatan aplikasi.
4. *Test* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian dilakukan agar fungsi-fungsi dalam aplikasi bebas dari *error*, dan hasilnya harus sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

## H. Metode Uji Coba

Pengujian merupakan metode yang dilakukan untuk menentukan kualitas perangkat lunak. Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126 yang dibuat oleh *International Organization for Standardization* (ISO) dan *International Electrotechnical Commission* (IEC). Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut:

1. *Functionality* (Fungsionalitas), yaitu kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Reliability* (Kehandalan), yaitu kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. *Usability* (Kebergunaan), yaitu kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
4. *Efficiency* (Efisiensi), yaitu kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
5. *Maintainability* (Pemeliharaan), yaitu kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.

6. *Portability* (Portabilitas), yaitu kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.

Masing-masing karakteristik kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa sub-karakteristik kualitas, sebagai berikut.

1. *Functionality* terdiri dari: *suitability* (kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna), *accuracy* (kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan), *security* (kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup/*hacker* maupun otorisasi dalam modifikasi data), *interoperability* (kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu), dan *compliance* (kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku).

2. *Reliability* terdiri dari: *maturity* (kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak), *fault tolerance* (kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak), dan *recoverability* (kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan).

3. *Usability* terdiri dari: *understandability* (kemampuan perangkat lunak dalam kemudahan untuk dipahami), *learnability* (kemampuan perangkat

lunak dalam kemudahan untuk dipelajari), *operability* (kemampuan perangkat lunak dalam kemudahan untuk dioperasikan), dan *attractiveness* (kemampuan perangkat lunak dalam menarik pengguna).

4. *Efficiency* terdiri dari: *time behavior* (kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya) dan *resource behavior* (kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan).
5. *Maintainability* terdiri dari: *analyzability* (kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan), *changeability* (kemampuan perangkat lunak untuk dimodifikasi tertentu), *stability* (kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak), dan *testability* (kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain).
6. *Portability* terdiri dari: *adaptability* (kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda), *instalability* (kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda), *coexistence* (kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya), dan *replaceability* (kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya).

Pengujian perangkat lunak dilakukan menggunakan kuesioner pengujian dengan skala jawaban Sangat Tidak Baik (STB), Tidak Baik (TB), Kurang Baik (KB), Baik (B), dan Sangat Baik (SB).

**Tabel 5.** Pedoman Penskoran

<b>Kategori</b>	<b>Skor</b>
Sangat Tidak Baik (STB)	1
Tidak Baik (TB)	2
Kurang Baik (KB)	3
Baik (B)	4
Sangat Baik (SB)	5

Konversi skor rata-rata menjadi nilai kualitatif menurut Saifuddin Azwar (2010: 163) disajikan berdasarkan Tabel 6:

**Tabel 6.** Klasifikasi Penilaian

<b>Rentang skor (i) kuantitatif</b>	<b>Kriteria Kualitatif</b>
$X > (\bar{x}_i + 1,50 SB_i)$	Sangat Baik
$(\bar{x}_i + SB_i) < X \leq (\bar{x}_i + 1,50 SB_i)$	Baik
$(\bar{x}_i - 0,5 SB_i) < X \leq (\bar{x}_i + SB_i)$	Cukup Baik
$(\bar{x}_i - 1,50 SB_i) < X \leq (\bar{x}_i - 0,5 SB_i)$	Sangat Kurang
$X \leq (\bar{x}_i - 1,50 SB_i)$	Sangat Kurang Baik

Keterangan:

skor maksimal ideal = skor tertinggi

skor minimal ideal = skor terendah

$X$  = rata-rata skor tiap butir

$\bar{x}_i$  = rata-rata ideal =  $\frac{1}{2}$ (skor maksimal ideal + skor minimal ideal)

$SB_i$  = simpangan baku ideal =  $\frac{1}{6}$  (skor maksimal ideal – skor minimal ideal)

Berdasarkan Tabel 6, maka didapat rentang skor kuesioner yang disajikan pada Tabel 7.



**Tabel 7.** Rentang Skor Kuesioner

<b>Rentang skor (i) kuantitatif</b>	<b>Kriteria Kualitatif</b>
$X > 4,00$	Sangat Baik
$3,67 < X \leq 4,00$	Baik
$2,66 < X \leq 3,67$	Cukup Baik
$1,99 < X \leq 2,66$	Sangat Kurang
$X < 1,99$	Sangat Kurang Baik

Untuk menentukan kualitas perangkat lunak dibutuhkan suatu pengujian. Pengujian merupakan metode yang dilakukan untuk menjelaskan tentang pengoperasian perangkat lunak yang terdiri dari perangkat pengujian, metode pengujian, dan pelaksanaan pengujian. Pengujian sistem menggunakan pengujian *Beta*, yaitu pengujian yang dilakukan pada satu atau lebih pengguna sistem pakar yang dibangun dan pengembang tidak terlibat. Rumus untuk mencari persentase jawaban pengguna adalah sebagai berikut:

$$Y = (P/Q) \times 100\%$$

Keterangan:

Y = Persentase jawaban responden tiap soal

P = Banyaknya jawaban responden tiap soal

Q = Total responden

## **I. Penyakit Pernapasan pada Balita**

Penyakit pernapasan pada balita yang akan digunakan sebagai basis pengetahuan untuk sistem pakar ini berupa tujuh jenis penyakit, yaitu: Batuk-Pilek (*Common Cold*), Sinusitis, Faringitis dan Tonsilofaringitis, Laringitis, Difteria, Bronkitis, dan Pneumonia. Data gejala dan penanganan/saran

diperoleh dari buku kedokteran dan dipadukan dengan pengetahuan dari dokter.

### 1. Batuk-Pilek (*Common Cold*)

Batuk-Pilek merupakan penyakit saluran pernapasan yang paling sering mengenai bayi dan anak. Bayi yang masih sangat muda akan sangat mudah tertular, karenanya perawat yang sedang batuk pilek tidak diperkenankan bekerja di ruangan bayi walaupun ia memakai masker, karena virus dapat menembusnya. Penularan juga masih tetap terjadi disebabkan seseorang yang pilek akan sering memegang hidungnya karena rasa gatal atau membuang ingusnya; jika tidak segera mencuci tangan ia akan menjadi sumber penularan (Ngastiyah, 2014: 31).

Gejala atau gambaran umum yang dialami pasien berupa batuk, pilek, bersin-bersin, pusing atau nyeri kepala atau kepala terasa berat, gangguan selera makan, terdapat cairan/lendir di tenggorok, hidung tersumbat, gelisah, dan bernapas melalui mulut.

Penanganan yang dapat dilakukan adalah membaringkan bayi tengkurap untuk pengeluaran sekret/lendir. Pada anak agak besar dapat diajarkan untuk mengeluarkan sekret/lendir/dahaknya sendiri. Jika batuk pilek lebih dari 2 hari belum sembuh, bawa anak ke dokter untuk mendapat pemeriksaan lanjutan.

### 2. Sinusitis

Menurut Ngastiyah (2014: 34), Sinusitis adalah radang sinus yang ada di sekitar hidung, dapat berupa Sinusitis Maksilaris atau Sinusitis

Frontalis (biasanya Sinusitis Maksilaris). Sinusitis dapat berlangsung akut maupun kronik, ia dapat mengenai anak yang sudah besar saat Sinusitis Pranasal sudah berkembang.

Gambaran umum yang biasa diderita pasien, yaitu: pilek, pusing atau nyeri kepala atau kepala terasa berat, serak, terdapat cairan/lendir di tenggorok, hidung tersumbat, bengkak kemerahan pada pipi yang dapat menjalar ke kelopak mata, nyeri di atas sinus/rongga berisi udara yang terdapat di sekitar pipi/hidung/mata, penciuman terganggu, tersumbatnya ostium (pembukaan ke dalam sinus untuk pertukaran udara dan lendir), dan bernapas melalui mulut.

Saran untuk orang tua adalah segera membawa anak ke dokter atau bagian THT untuk pemeriksaan lanjutan apakah perlu dilakukan pencucian sinus atau tidak.

### 3. Faringitis/Tonsilofaringitis

Radang faring pada bayi dan anak hampir selalu melibatkan organ sekitarnya, sehingga infeksi pada faring biasanya juga mengenai tonsil, sehingga disebut sebagai Tonsilofaringitis (Ngastiyah, 2014: 36).

Gambaran umum yang biasa dialami penderita, yaitu: demam/suhu tubuh lebih dari  $38^{\circ}$  Celcius, nyeri menelan, nyeri tenggorok, sakit telinga, bengkak kelenjar ludah/submandibula (di rahang bawah), dan mulut berbau.

Penanganan yang dapat dilakukan oleh orang tua adalah anjurkan anak untuk istirahat di tempat tidur sampai demam hilang dan berikan

makanan lunak. Jika sampai 2 hari demam tidak juga hilang dan kondisi anak tetap sama atau bahkan menurun, segera bawa ke dokter untuk pemeriksaan tenggorok apakah perlu tonsilektomi atau tidak.

#### 4. Laringitis

Penyebab Laringitis umumnya adalah *Streptococcus hemolyticus*, *Streptococcus viridians*, *pneumokokus*, *Staphylococcus hemolyticus*, dan *Haemophilus influenzae*. Proses radang pada laring dipermudah oleh trauma, bahan kimia, radiasi, alergi dan pemakaian suara berlebihan (Ngastiyah, 2014: 39).

Gejala yang biasa dialami oleh penderita Laringitis berupa batuk, pilek, sesak napas, demam/suhu tubuh lebih dari 38<sup>0</sup> Celcius, serak, nyeri menelan, napas berat, suara napas kasar, hilang suara, nyeri ulu hati, napas tersengal, dan gelisah.

Saran untuk orang tua terhadap balitanya yaitu anjurkan anak untuk istirahat bersuara dan bawa ke dokter atau bagian THT untuk pemeriksaan lanjutan, biasanya berupa laringoskopi direk.

#### 5. Difteria

Menurut Ngastiyah (2014: 40), penyakit Difteria adalah suatu infeksi akut yang mudah menular, dan yang sering diserang terutama saluran pernapasan bagian atas, dengan tanda khas timbulnya pseudomembran. Penyebab penyakit Difteria adalah kuman *Diphtheriae corynebacterium* bersifat gram positif dan polimorf, tidak bergerak, dan tidak membentuk spora.

Gambaran umum penderita Difteria, yaitu: pusing, sesak napas, demam, gangguan selera makan, serak, nyeri menelan, suara napas kasar, bengkak, pada leher, lesu, dan pucat.

Saran bagi orang tua adalah segera bawa anak ke dokter untuk melakukan pemeriksaan lanjutan, biasanya berupa pemeriksaan darah dan urin.

## 6. Bronkitis

Secara harfiah, Bronkitis adalah suatu penyakit yang ditandai oleh adanya inflamasi bronkus. Secara klinis para ahli mengartikan Bronkitis sebagai suatu penyakit atau gangguan respiratorik dengan batuk merupakan gejala yang utama dan dominan (Ngastiyah, 2014: 54).

Gejala yang biasa dialami pasien yaitu: batuk, sesak napas, terdapat cairan/ lendir di tenggorok, mengi, nyeri dada, dan suara napas kasar.

Penanganan yang dapat dilakukan orang tua adalah memberi anak minum yang banyak terutama sari buah-buahan, jangan biarkan anak merasa kepinginan, untuk meringankan gejala batuknya berikan minum hangat tidak manis; untuk anak yang agak besar beritahu supaya membuang dahaknya. Jika batuk tetap ada dan tidak ada perbaikan setelah 2 minggu berturut-turut dan/atau berulang paling sedikit 3x dalam 3 bulan, bawa anak ke dokter untuk pemeriksaan lanjutan

## 7. Pneumonia

Pneumonia adalah suatu radang paru yang disebabkan oleh bermacam-macam etiologi seperti bakteri, virus, jamur, dan benda asing.

Anak mengalami napas yang dangkal dan cepat, yaitu untuk usia kurang dari dua bulan napas lebih dari 60x/menit; dua bulan- satu tahun napas lebih dari 50x/menit; dan satu sampai lima tahun napas lebih dari 40x/menit (Kustantinah, 2007: 182).

Gambaran umum atau gejala Pneumonia biasanya adalah batuk, pusing, sesak napas, demam, nyeri sekitar hidung dan mulut, napas cuping hidung, napas cepat dan dangkal, serta lesu.

Saran bagi orang tua adalah segera bawa anak ke dokter untuk melakukan pemeriksaan lanjutan.