Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1 Penyakit Alergi

Alergi merupakan bentuk reaksi sistem kekebalan tubuh terhadap sesuatu yang dianggap berbahaya bagi tubuh manusia. Hal ini bisa berupa substansi yang masuk atau melakukan kontak dengan tubuh, baik secara langsung maupun tidak langsung. Orang yang memiliki alergi, memiliki sistem kekebalan tubuh yang bereaksi terhadap suatu substansi tertentu, yang umumnya tidak berbahaya terhadap lingkungan. Substansi ini disebut *allergen*. *Allergen* juga merupakan senyawa yang dapat menginduksi *Immunoglobulin E* (IgE). Biasanya, *allergen* melakukan kontak dengan manusia melalui paparan berupa proses penghisapan, proses menelan, kontak fisik, ataupun injeksi (Sibuea, 2013).

2.2 *Website*

Website merupakan sistem pelayanan informasi di internet yang berbasis grafis dan didasarkan pada konteks *hypertext*. Konsep ini sangat mirip dengan tulisan biasa, tetapi terdapat satu aspek yang penting, yaitu memungkinkan untuk konteks (*link*) didalam dokumen itu sendiri atau koneksi ke dokumen lain.

2.3 *Ontology*

Ontology terdiri dari istilah-istilah dasar, relasi antara istilah-istilah tersebut, dan aturan yang menggabungkan istilah tersebut. *Ontology* tersebut dapat dikatakan

Decision support...,Andre Halim,FTI UMN,2017

sebagai sebuah *knowledge* yang dapat di-*share* dan digunakan oleh beberapa aplikasi *software*. Oleh karena itu, *ontology* dapat menjadi sebuah spesifikasi eksplisit dari sebuah konsep, yang bertujuan untuk memberikan fasilitas *knowledge sharing* (Gruber, 1995).

Ontology merepresentasikan dunia nyata secara terstruktur dan sistematis. *Ontology* menyediakan sebuah model referensi untuk domain-domain *ontology* tersebut. Model referensi merupakan sekumpulan istilah yang dapat digunakan untuk menyederhanakan komunikasi antar *domain experts* dan meningkatkan pemahaman dan *knowledge sharing.* Berikut merupakan komponen-komponen dari *Ontology*.

1. *Instances*

   Instances adalah komponen dasar dari sebuah *ontology*. *Instance* dari sebuah *ontology* dapat berupa objek nyata seperti manusia, hewan, benda atau bisa juga berupa objek abstrak seperti angka dan huruf

2. *Class*

   Class menjelaskan konsep yang ada di dalam sebuah *domain*. Biasanya sebuah *class* merupakan kumpulan dari objek. Sebuah *class* juga memiliki turunan dari *class* itu yang mempresentasikan konsep yang lebih spesifik dari *class* atasnya.

3. *Attributes*

   Objek-objek yang berada dalam *ontology* dapat dideskripsikan dengan memberikan tambahan atribut ke dalam objek tersebut. Setiap atribut memiliki setidaknya sebuah nama dan nilai dari nama tersebut. Hal

Decision support...,Andre Halim,FTI UMN,2017

tersebut digunakan untuk menyimpan informasi yang spesifik tentang objek yang memiliki atribut tersebut.

4. *Relationship*

   Sebuah relationship menjadi penting dalam sebuah *ontology*. Hal tersebut dikarenakan dalam suatu *ontology*, relasi antar objek yang ada harus bisa mendeskripsikan keunggulan dari *ontology* yang berasal dari kemampuannya dalam mendeskripsikan sebuah relasi.

2.4 *Web Ontology Language (OWL)*

Web Ontology Language (OWL) merupakan suatu bahasa pemrograman *website* yang dapat digunakan oleh aplikasi-aplikasi yang memiliki fitur mencari, memproses, dan menampilkan informasi. OWL juga biasa digunakan untuk menggambarkan model data. Secara umum, OWL memiliki struktur yang sama dengan RDF. Hal-hal yang membedakan RDF dengan OWL terletak pada fitur *vocabulary* yang lebih kaya pada OWL. OWL dirancang untuk bisa dibaca oleh komputer, sehingga komputer dapat membaca dan mengerti hubungan antara data-data yang ada.

OWL memiliki tiga buah subbahasa sebagai berikut.

a. OWL *Lite*

   OWL *Lite* biasa digunakan oleh *user* yang membutuhkan suatu klasifikasi hierarki dan *constraint* yang sederhana.

b. OWL DL

Decision support...,Andre Halim,FTI UMN,2017

OWL DL biasa digunakan oleh *user* yang membutuhkan tingkat ekspresi maksimal dan semua kesimpulan yang dihasilkan dapat dihitung dalam waktu terbatas.

c. OWL *Full*

OWL *Full* biasa digunakan oleh *user* yang membutuhkan tingkat ekspresi maksimal dan kebebasan dalam penggunakan *syntax* dari RDF tanpa mempertimbangkan komputasi yang dibutuhkan.

```
<owl:Class rdf:ID="Agent">
  <rdfs:label>Agent</rdfs:label>
  <rdfs:subClassOf rdf:resource="&foaf;Agent"/>
  <rdfs:subClassOf rdf:resource="&loc;
   ThingHasLocationContext"/>
</owl:Class>

<owl:Class rdf:ID="Person">
  <rdfs:label>Person</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Agent"/>
</owl:Class>
```

Gambar 2.1 Contoh Program OWL

2.5 SPARQL

The Simple Protocol and RDF Query Language (SPARQL), merupakan bahasa pemrograman SQL, yang digunakan untuk mengambil data dari RFD. SPARQL bekerja dengan mengambil dari dari RDF *triples* dan *resources* dalam mencocokan bagian dari *query* dan mengembalikan hasil dari *query*. Bahasa ini khusus digunakan untuk pengembangan *Semantic Web*.

Query ini menggunakan URI untuk mengambil struktur dari RDF/OWL. Bahasa *query* ini hampir sama dengan *query* SQL pada *database* umum lainnya,

Decision support...,Andre Halim,FTI UMN,2017

tetapi SPARQL lebih sederhana dibanding SQL biasa. Berikut merupakan bagian-bagian yang digunakan dalam query SPARQL (Pollock, 2009).

1. PREFIX

   Pernyataan PREFIX merupakan sebuah metode yang digunakan sebagai petunjuk yang membawa informasi dalam sebuah halaman *website*. Pada dasarnya, PREFIX digunakan untuk menyingkat sebuah sumber data.

2. SELECT

   Pernyataan SELECT telah didefinisikan oleh sebuah daftar variabel yang dikembalikan sebagai hasil dari eksekusi *query*. Setiap variabel diawal dengan tanda tanya (?)

3. WHERE

   Pernyataan WHERE didefinisikan sebagai deretan yang dimiliki oleh setiap hasil *query* yang valid. Seluruh pola yang merepresentasikan suatu kalimat RDF harus sesuai dengan RDF *triples*, yaitu terdiri dari subjek, predikat, dan objek. Ketiga hal tersebut dapat direpresentasikan oleh URI.

4. OPTIONAL

   Pernyataan OPTIONAL digunakan untuk mengatasi ketidakcocokan struktur pola query dengan pola yang ada pada RDF.

5. FILTER

Decision support...,Andre Halim,FTI UMN,2017

Pernyataan FILTER digunakan untuk menyaring data sesuai dengan kebutuhan dari user, sehingga data yang ditampilkan hanya data yang memenuhi syarat dari FILTER

```
PREFIX fb:<http://rdf.freebase.com/ns/>

SELECT ?film ?when
WHERE
{
    ?film fb:film.film.initial_release_date ?when .
}
```

Gambar 2.2 Contoh *Query SPARQL*

2.6 Protégé

Protégé merupakan sebuah *software* yang bersifat gratis dan *open source*, yang bertindak sebagai sebuah sistem manajemen pengetahuan. Protégé memberikan *interface* yang berguna untuk mendefinisikan *ontology*.

Gambar 2.3 Logo Aplikasi Protégé 4.3

Plug-in architecture dari *tools* Protégé dapat disesuaikan untuk membangun aplikasi berbasis *ontology* yang sederhana maupun kompleks. Tim pengembang dapat mengintegrasikan *output* dari Protégé dengan sistem untuk membangun sebuah *intelligence* (http://protege.stanford.edu/).

Decision support...,Andre Halim,FTI UMN,2017

2.7 *Hypertext Preprocessor (PHP)*

Hypertext Preprocessor (PHP) merupakan sebuah bahasa pemrograman yang berbentuk *script*, yang dapat ditanamkan pada pemrograman HTML. Biasanya, PHP banyak digunakan untuk membuat program *website* dinamis. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda jika dibandingkan dengan pemrograman *client side* seperti JavaScript yang diproses pada *web browser*.

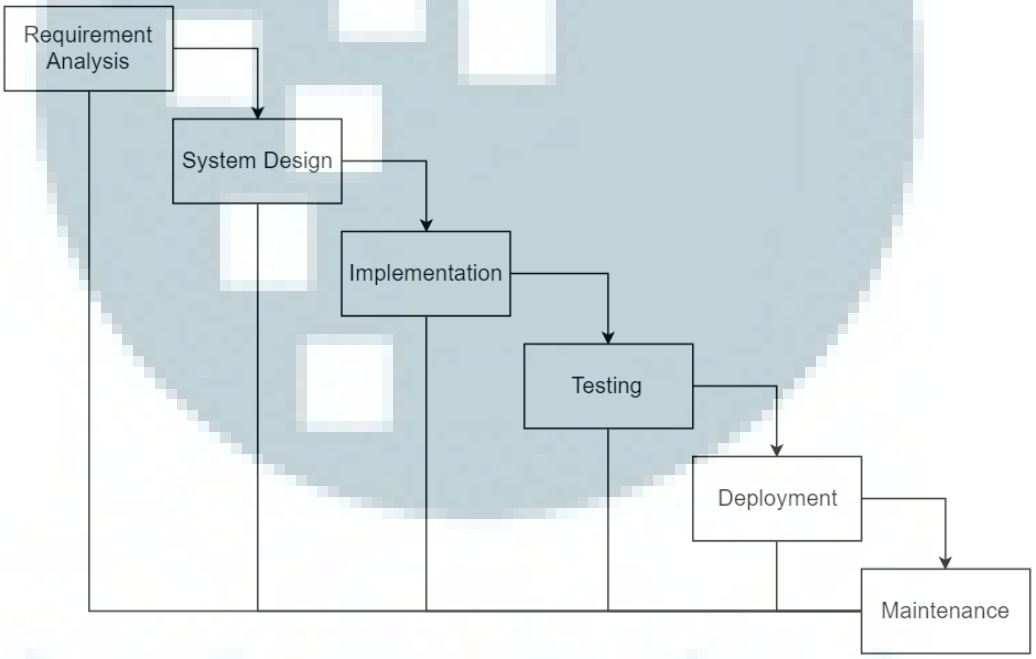
Perbedaan antara PHP dengan JavaScript adalah langkah-langkah dalam memuat halaman *web* dimana *embedded program* dijalankan. *Client-side language* seperti JavaScript dibaca dan dijalankan oleh *web browser*, setelah memuat halaman *web* dari server. Hal ini sangat berbeda dengan *server-side language* seperti PHP dimana program dijalankan oleh *web server* sebelum mengirimkan halaman *web* ke *browser*. Perbedaan lainnya adalah *client side language* memberikan control bagaimana reaksi halaman saat ditampilkan oleh *browser*, sedangkan *server-side language* mengizinkan untuk memuat halaman sebelum halaman tersebut dikirim ke *browser* (Yank, 2009).

PHP dikembangkan pertama kali oleh Rasmus Lerdorf pada tahun 1995. Pada awalnya PHP merupakan kepanjangan dari *Personal Home Page*. Akan tetapi, PHP berubah menjadi pemrograman *web* yang dapat tidak hanya untuk membuat halaman *web* yang sederhana, tetapi juga dapat digunakan untuk membuat *website* yang dapat digunakan oleh banyak pihak.

Decision support...,Andre Halim,FTI UMN,2017

2.8 *Waterfall Model*

Model *Waterfall* merupakan model *sequential*, yang cukup sering digunakan untuk proses pengembangan *software*. Model ini memiliki proses yang memiliki alur menurun seperti layaknya air terjun, dimulai dari tahap menentukan konsep, inisiasi, analisis, desain, konstruksi, pengujian, dan pelaksanaan dari *software* tersebut. Berikut adalah struktur dari model *Waterfall* (Poppendieck, 2003).



Gambar 2.4 Struktur Model *Waterfall*

1. *Requirement Anaylsis*

   Pada fase ini, dilakukan perancangan terhadap fungsi dan tujuan sistem yang akan dibuat. Seluruh kebutuhan sistem tersebut harus bisa dijelaskan secara terperinci, agar tahap selanjutnya dapat berjalan dengan lancar. Tahap ini juga merupakan tahap yang paling penting karena jika sudah

Decision support...,Andre Halim,FTI UMN,2017

ditetapkan, maka rancangan tidak dapat diubah tanpa kembali, tanpa mengubah proses selanjutnya.

2. *System Design*

   Pada fase ini, dilakukan desain sistem berdasarkan hasil analisis yang telah dilakukan pada fase sebelumnya. *Software* dan *hardware* yang sesuai dengan kebutuhan untuk dapat menyelesaikan tujuan sistem juga dianalisa pada fase ini.

3. *Implementation*

   Pada fase ini, desain yang sudah dibuat pada fase sebelumnya diterjemahkan ke dalam bahasa komputer melalui pemrograman. Pemrograman dilakukan hingga seluruh fungsi selesai dibuat dan dijalankan.

4. *Testing*

   Pada fase ini, program yang sebelumnya sudah dibuat diuji untuk mengetahui apakah masih ada *error* atau fungsi-fungsi yang belum berjalan dengan baik.

5. *Deployment*

   Pada fase ini, program yang sudah dibuat dan telah diuji dengan baik akan dipublikasikan kepada masyarakat.

6. *Maintenance*

   Setelah program dipublikasikan, jika ditemukan adanya beberapa masalah, maka akan dilakukan pemeliharaan untuk memperbaiki masalah-masalah yang ditemukan.

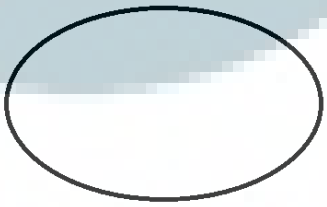
Decision support...,Andre Halim,FTI UMN,2017

2.9 *Flowchart*

Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* dapat membantu memecahkan masalah kedalam bagian-bagian yang lebih kecil dan menolong dalam menganalisa alternatif lainnya dalam sebuah proses operasi. *Flowchart* biasa digunakan untuk mempermudah penyelesaian suatu masalah, khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. Berikut adalah komponen-komponen umum dari *Flowchart* (Mahyusir, 1989).

2.9.1 *Terminal*

Terminal merupakan lambang yang digunakan untuk memulai dan mengakhiri sebuah *flowchart*, yang diisi dengan kata *START* atau *END*. Lambang *terminal* memiliki bentuk oval yang memanjang.



Gambar 2.5 Lambang *Terminal*

2.9.2 *Process*

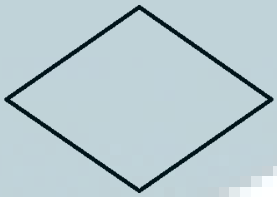
Process merupakan lambang yang digunakan untuk memulai sebuah kegiatan yang dilakukan. Lambang *process* menggunakan bentuk persegi panjang.

Decision support...,Andre Halim,FTI UMN,2017

Gambar 2.6 Lambang *Process*

2.9.3 *Decision*

Decision merupakan lambang *flowchart* yang digunakan jika terdapat kemungkinan yang mungkin terjadi. Biasanya digunakan jika ada pertanyaan yang membutuhkan jawaban ya / tidak, atau benar / salah. Lambang *decision* menggunakan bentuk belah ketupat.

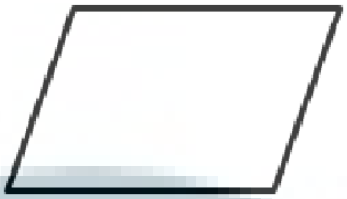


Gambar 2.7 Lambang *Decision*

2.9.4 *Input / Output*

Input / Output merupakan lambang yang digunakan pada saat dibutuhkan adanya data dimasukkan atau adanya hasil keluaran berupa data. Lambang *input* / *output* menggunakan bentuk jajar genjang.

Decision support...,Andre Halim,FTI UMN,2017



Gambar 2.8 Lambang *Input / Output*

2.9.5 *Flow Line*

Flow line merupakan panah yang menghubungkan setiap lambang-lambang yang digunakan pada *flowchart*. Panah itu menunjukkan jalur proses pada sebuah *flowchart*.

Gambar 2.9 Lambang *Flow Line*

2.10 *Decision Support System*

Decision Support System (DSS) adalah sebuah sistem yang digunakan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur namun tidak untuk menggantikan peran penilaian mereka. Berikut adalah karakteristik dari *Decision Support System* (Turban, Aronson, 2005).

1. Membantu pengambilan keputusan dalam memecahkan masalah, terutama pada situasi semi terstruktur dengan menyertakan penilaian manusia dan informasi terkomputerisasi
2. Memberi dukungan untuk semua level manajerial
3. Meningkatkan efektifitas pengambilan keputusan

Decision support...,Andre Halim,FTI UMN,2017

4. Memberi dukungan individu dan kelompok
5. Dapat diadaptasi dengan mudah, dalam hal menambahkan, menghapus, mengubah, atau menyusun kembali elemen-elemen dasar, dan dapat dimodifikasi untuk memecahkan masalah lainnya.

Komponen pada *Decision Support System* terdiri dari empat subsistem. Berikut adalah komponen dari *Decision Support System* (Turban, Aronson, 2005).

1. Manajemen Data

   Manajemen Data merupakan sistem data yang terorganisir dalam suatu *database*. Manajemen data ini diatur oleh *software* yang disebut dengan *Database Management System* (*DBMS*). Berikut adalah komponen dari manajemen data.

   a. *Database DSS*

      Database merupakan kumpulan data, yang memiliki hubungan-hubungan tersendiri, yang diatur supaya data tersebut dapat berkaitan dengan struktur dan kebutuhan dari sebuah perusahaan. *Database* pada sebuah DSS diambil dari sumber internal maupun eksternal, dari satu atau lebih *user*. Hasilnya ditempatkan pada sebuah tempat khusus di perusahaan.

   b. *Database Management System* (*DBMS*)

      DBMS merupakan sebuah *software* yang digunakan untuk mengelola *database*. Data dibuat, diakses, dan di-*update* oleh sebuah DBMS.

   c. *Data Directory*

Decision support...,Andre Halim,FTI UMN,2017

Data Directory merupakan merupakan sebuah katalog yang berisi tentang seluruh data yang ada di *database*. DD ini menyimpan seluruh informasi tentang *entity* dan *relationship* pada data.

d. *Query Facility*

Query Facility merupakan sebuah *query* yang berfungsi untuk memberikan kebutuhan untuk *user* dalam mengelola data, sehingga *user* dapat menemukan data yang dibutuhkannya.

2. Manajemen Model

Manajemen Model merupakan *software* yang memiliki empat fungsi, yaitu:

a. Sarana perancangan model
b. Sarana perancangan *output* dari model
c. Sarana memperbaharui dan mengubah model
d. Sarana manipulasi data

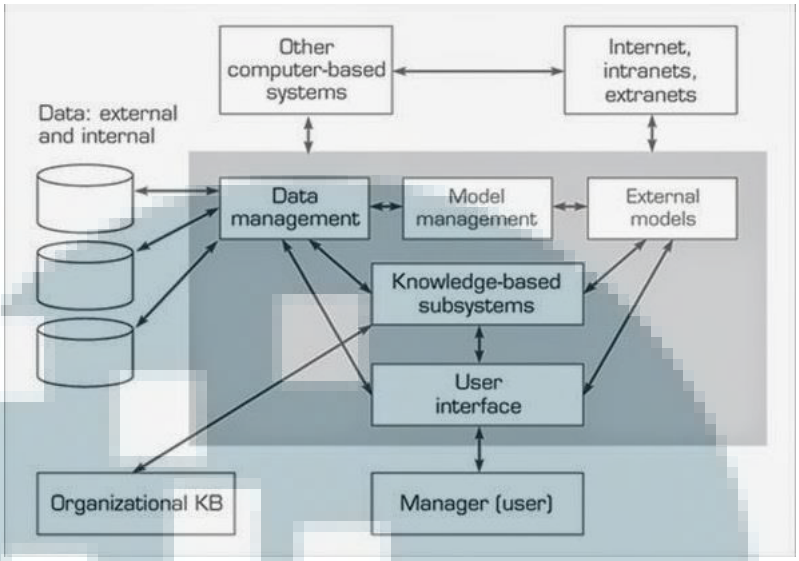
3. Subsistem Dialog

Subsistem Dialog merupakan sebuah istilah yang mencangkup seluruh aspek komunikasi antar *user*. Cangkupannya tidak hanya *hardware* dan *software*, tetapi juga faktor lainnya yang berkaitan dengan kemudahan penggunaan, akses, dan interaksi.

4. Manajemen *Knowledge*

Manajemen *Knowledge* merupakan sebuah subsistem yang mendukung subsistem lainnya, yang memberikan sumber dalam memperbesar pengetahuan para pengambil keputusan.

Decision support...,Andre Halim,FTI UMN,2017



Gambar 2.10 Struktur dari Decision Support System

Pada dasarnya, proses pengambilan keputusan dilakukan dengan tujuan untuk menghasilkan berbagai alternatif keputusan yang dapat diambil. DSS hanya sebagai alat bantu manajemen dalam pengambilan keputusan. Hal ini tidak dimaksudkan untuk menggantikan fungsi pengambilan keputusan dalam membuat keputusan, melainkan hanya sebagai alat bantu pengambilan keputusan dalam melaksanakan tugas, sehingga dapat dikatakan bahwa DSS memberikan manfaat bagi manajemen dalam hal meningkatkan efektifitas kerja, terutama dalam proses pengambilan keputusan (Enggar, 2013).

Adapun syarat-syarat yang harus dipenuhi untuk dapat membuat sebuah *Decision Support System*. Berikut adalah syarat dari *Decision Support System* (Bidgoli, 1989).

1. Memerlukan *hardware*, *software*, dan *user*.
2. Dirancang untuk mendukung sebuah pengambilan keputusan
3. Membantu *decision maker* dalam setiap level keputusan

Decision support...,Andre Halim,FTI UMN,2017