

BAB II

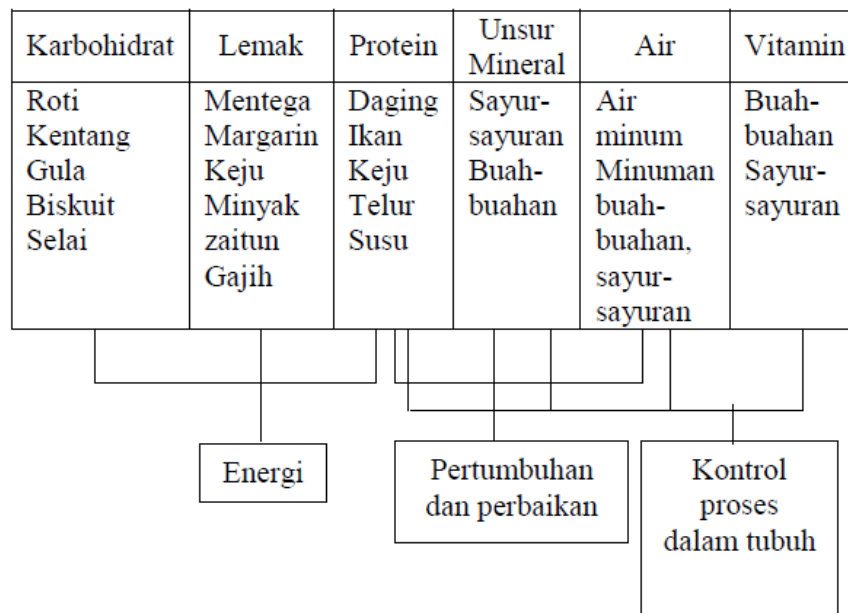
LANDASAN TEORI

2.1. Makanan Sehat

Makanan yang sehat yaitu makanan yang higienis dan bergizi. Makanan yang higienis adalah makanan yang tidak mengandung kuman penyakit dan tidak mengandung racun yang dapat membahayakan kesehatan. Bahan makanan yang akan kita makan harus mengandung komposisi gizi yang lengkap, yaitu terdiri atas karbohidrat, lemak, protein, vitamin, mineral, dan air. Di Indonesia komposisi tersebut dikenal dengan nama makanan “4 sehat 5 sempurna” (Hanifa, N. dan Luthfeni, 2006).

Zat gizi merupakan unsur yang terkandung dalam makanan yang dapat memberikan manfaat bagi kesehatan manusia. Masing-masing bahan makanan yang dikonsumsi memiliki kandungan gizi yang berbeda. Makanan yang satu dengan makanan yang lainnya memiliki kandungan zat gizi yang berbeda-beda. Perbedaan tersebut dapat berupa jenis zat gizi yang terkandung dalam makanan, maupun jumlah dari masing-masing zat gizi. Setiap zat gizi memiliki fungsi yang spesifik. Masing-masing zat gizi tidak dapat berdiri sendiri dalam membangun tubuh dan dalam menjalankan proses metabolisme. Namun berbagai zat gizi memiliki fungsi yang berbeda.

Menurut Almatsier (2011), zat gizi adalah ikatan kimia yang diperlukan oleh tubuh untuk melakukan fungsinya yaitu karbohidrat, lemak, dan protein berfungsi sebagai sumber energi atau penghasil energi yang bermanfaat untuk menggerakkan tubuh dan proses metabolisme di dalam tubuh, zat gizi yang berfungsi sebagai pembentuk sel-sel pada jaringan tubuh manusia dan memelihara jaringan tersebut, serta mengatur proses-proses kehidupan merupakan fungsi dari kelompok zat gizi seperti protein, lemak, mineral, vitamin dan air (Marmi. 2013). Hubungan antara nutrisi, fungsi mereka dalam tubuh, dan makanan penting yang memasoknya, ditunjukkan pada Gambar 2.1.



Gambar 2.1. Diagram Menunjukkan Fungsi Mereka dan Makanan Representatif Tempat Mereka Biasa Ditemukan

2.2. Gizi

Gizi yang terkandung dalam makanan sangat penting untuk diperhatikan. Salah satu upaya perbaikan status gizi masyarakat yang mempunyai daya ungkit tinggi dan berkelanjutan adalah pendidikan dan penyuluhan gizi yang bertujuan untuk merubah perilaku kearah pola hidup sehat dan sadar gizi. Dengan pola hidup sehat dan sadar gizi, masyarakat akan terbiasa dengan mengonsumsi aneka ragam pangan, berperilaku hidup bersih, melakukan aktivitas fisik, dan mempertahankan berat badan normal. Apabila semua perilaku gizi seimbang tersebut diterapkan, maka masyarakat akan terhindar dari kekurangan atau kelebihan gizi dan turut mengurangi berbagai penyakit infeksi.

Kebutuhan asupan gizi setiap orang berbeda-beda, tergantung dari jenis kelamin, usia, kapasitas aktifitas keseharian, dan lainnya. Berikut tabel angka kecukupan gizi yang telah ditetapkan oleh pemerintah (Pemerintah Indonesia, 2013):

Tabel 2.1. Angka Kecukupan Gizi (perorang perhari)

Kelompok Umur	BB	TB	Energi	Protein	Lemak (gr)			Karbohidrat	Serat	Air
	(kg)	(cm)	(kkal)	(gr)	Total	n-6	n-3	(gr)	(gr)	(mL)
Bayi/Anak										
0 – 6 bulan	6	61	550	12	34	4.4	0.5	58	0	-
7 – 11 bulan	9	71	725	18	35	4.4	0.5	82	10	800
1-3 tahun	13	91	1125	26	44	7	0.7	155	16	1200
4-6 tahun	19	112	1600	35	62	10	0.9	220	22	1500
7-9 tahun	27	130	1850	49	72	10	0.9	254	26	1900
Laki-laki										
10-12 tahun	34	142	2100	56	70	12	1.2	289	30	1800
13-15 tahun	46	158	2475	72	83	16	1.6	340	35	2000
16-18 tahun	56	165	2675	66	89	16	1.6	368	37	2200
19-29 tahun	60	168	2725	62	91	17	1.6	375	38	2500
30-49 tahun	62	168	2625	65	73	17	1.6	394	38	1600
50-64 tahun	62	168	2325	65	65	14	1.6	349	33	2600
65-80 tahun	60	168	1900	62	53	14	1.6	309	27	1900
80+ tahun	58	168	1525	60	42	14	1.6	248	22	1600
Perempuan										
10-12 tahun	36	145	2000	60	67	10	1	275	28	1800
13-15 tahun	46	155	2125	69	71	11	1.1	292	30	200
16-18 tahun	50	158	2125	59	71	11	1.1	292	30	2100
19-29 tahun	54	159	2250	56	75	12	1.1	309	32	2300
30-49 tahun	55	159	2150	57	60	12	1.1	323	30	2300
50-64 tahun	55	159	1900	57	53	11	1.1	285	28	2300
65-80 tahun	54	159	1550	56	43	11	1.1	252	22	1600
80+ tahun	53	159	1425	55	11	11	1.1	232	20	1500
Hamil (+an)										
Trimester 1			+180	+20	+6	+2	+0.3	+25	+3	+300
Trimester 2			+300	+20	+10	+2	+0.3	+40	+4	+300
Trimester 3			+300	+20	+10	+2	+0.3	+40	+4	+300
Menyusui (+an)										
6 bln pertama			+300	+20	+11	+2	+0.2	+45	+5	+800
6 bln kedua			+400	+20	+13	+2	+0.2	+55	+6	+650

2.3. Penyakit

Penyakit adalah suatu keadaan tidak normal dari tubuh atau pikiran yang menyebabkan ketidaknyamanan, disfungsi, atau kesukaran terhadap orang yang

dipengaruhinya. Beberapa penyakit seringkali diderita oleh masyarakat Indonesia, antara lain diabetes, kolesterol, hipertensi, asam urat, dan penyakit kulit.

2.3.1. Diabetes

Diabetes mellitus merupakan penyakit metabolik yang berlangsung kronik dimana penderita diabetes tidak bisa memproduksi insulin dalam jumlah yang cukup atau tubuh tidak mampu menggunakan insulin secara efektif sehingga terjadilah kelebihan gula di dalam darah dan baru dirasakan setelah terjadi komplikasi lanjut pada organ tubuh (Khairani, 2012).

Klasifikasi etologi kelainan glikemia diabetes mellitus sebagai berikut (Azrimaidaliza, 2011):

1. Tipe 1

Ditandai dengan kegagalan produksi insulin yang parsial atau total oleh sel-sel B pankreas. Faktor penyebab masih belum dimengerti dengan jelas tetapi beberapa virus tertentu, penyakit autoimun dan faktor-faktor genetik mungkin turut berperan.

2. Tipe 2

Ditandai dengan resistensi insulin ketika hormon insulin diproduksi dengan jumlah yang tidak memadai atau dengan bentuk yang tidak efektif. Ada korelasi genetik yang kuat pada tipe diabetes ini dan proses terjadinya berkaitan erat dengan obesitas. Anak dengan diabetes tipe 2 dilaporkan memiliki riwayat penyakit kardiovaskular dalam keluarga dan atau sindrom metabolik.

3. Tipe Spesifik

Berupa defek genetik pada fungsi sel-B, defek genetik pada kerja insulin, penyakit pada kelenjar eksokrin pankreas, endokrinopati, ditimbulkan oleh obat-obatan atau zat kimia, infeksi, bentuk *immune-mediated* diabetes yang langka. Kadang-kadang sindrom genetik lain yang disertai diabetes.

4. Diabetes Gestasional

Bentuk diabetes yang terjadi selama kehamilan. Kebanyakan, tapi tidak semuanya, akan sembuh setelah melahirkan.

2.3.1.1. Bahan Anjuran dan Pantangan bagi Penderita Diabetes

Terdapat bahan-bahan masakan yang dianjurkan dan tidak dianjurkan bagi penderita diabetes, antara lain (Kevin Adrian, 2018):

Tabel 2.2. Bahan Anjuran dan Pantangan bagi Penderita Diabetes

Bahan Anjuran	Bahan Pantangan
Beras Merah	Beras Putih
Ubi	Daging Tinggi Lemak
Oatmeal	Kulit Ayam
Daging Tanpa Lemak	Susu Tinggi Lemak
Ayam Tanpa Kulit	Tepung Terigu
Brokoli	Roti Tawar Putih
Bayam	Pizza Beku
Makarel	Sayur Kalengan Tinggi Garam
Yoghurt	Buah Tinggi Gula
Tuna	
Salmon	
Sarden	
Buah Rendah Gula	

2.3.2. Kolesterol

Kolesterol merupakan salah satu dari lemak atau senyawa lipid yang sejak awal diyakini sebagai faktor pencetus PJK dan banyak penelitian membuktikan hal tersebut. Berdasarkan laporan WHO tahun 2002 tercatat sebanyak 4,4 juta kematian akibat hiperkolesterol. Kadar kolesterol yang tinggi di dalam tubuh akibat dari mengonsumsi makanan yang dapat menaikkan kadar kolesterol akan menyebabkan penyempitan dan pengerasan pembuluh darah atau disebut juga aterosklerosis (Sari, 2010).

Orang yang mempunyai berat badan lebih seringkali mempunyai kadar kolesterol darah yang lebih tinggi bila dibandingkan dengan orang yang berat badannya normal. Peningkatan kolesterol darah juga dapat disebabkan oleh kenaikan kolesterol yang terdapat pada *verylow-density lipoprotein* dan *low-density lipoprotein* sekunder karena peningkatan *trigliserida* yang besar dalam

sirkulasi apabila terjadi penumpukan lemak berlebihan di dalam tubuh (Listiyana, 2013).

2.3.2.1. Bahan Anjuran dan Pantangan bagi Penderita Kolestrol

Terdapat bahan-bahan masakan yang dianjurkan (Kevin Adrian, 2018) dan tidak dianjurkan (Irene Anindyaputri, 2017) bagi penderita kolestrol, antara lain:

Tabel 2.3. Bahan Anjuran dan Pantangan bagi Penderita Kolestrol

Bahan Anjuran	Bahan Pantangan
Alpukat	Ampela
Apel	Bahan Tinggi Lemak
Bunga Matahari	Hati
Beras Cokelat	Jantung
Brokoli	Kepiting
Strawberry	Kulit Ayam
Bunga Matahari	Margarin Tinggi Lemak
Jagung	Minyak Kelapa
Jelai	Otak
Kacang-Kacangan	Santan
Kedelai	Telur
Minyak Sayur	Udang
Minyak Zaitun	Usus
Ubi	
Oatmeal	
Pir	
Salmon	
Sarden	
Ubi	
Terong	
Tuna	
Tongkol	
Bahan Rendah Atau Tanpa Lemak	

2.3.3. Hipertensi

Hipertensi atau tekanan darah tinggi merupakan pembunuh diam-diam karena pada sebagian besar kasus tidak menunjukkan gejala apapun. Hipertensi

merupakan salah satu faktor resiko utama yang menyebabkan serangan jantung dan stroke, yang menyerang sebagian besar penduduk dunia. Hipertensi adalah suatu keadaan dimana dijumpai tekanan darah 140/90 mmHg atau lebih untuk usia 13-50 tahun dan tekanan darah mencapai 160/95 mmHg untuk usia di atas 50 tahun. Pengukuran tekanan darah minimal sebanyak dua kali untuk lebih memastikan keadaan tersebut. Hipertensi dapat dikelompokkan menjadi dua jenis, yaitu: Hipertensi primer atau essensial dan hipertensi sekunder. Hipertensi primer adalah hipertensi yang tidak atau belum diketahui penyebabnya. Hipertensi primer menyebabkan perubahan pada jantung dan pembuluh darah. Sedangkan hipertensi sekunder adalah hipertensi yang disebabkan atau sebagai akibat dari adanya penyakit lain dan biasanya penyebabnya sudah diketahui, seperti penyakit ginjal dan kelainan hormonal atau pemakaian obat tertentu (Saputra, 2013).

Penyakit hipertensi menimbulkan kecacatan permanen, kematian mendadak dan yang berakibat sangat fatal. Untuk meningkatkan kualitas hidup agar tidak menimbulkan masalah di masyarakat perlu upaya pencegahan dan penanggulangan hipertensi dimulai dengan meningkatkan kesadaran masyarakat dan perubahan pola hidup ke arah yang lebih sehat (Herwati, 2014).

2.3.3.1. Bahan Anjuran dan Pantangan bagi Penderita Hipertensi

Terdapat bahan-bahan masakan yang dianjurkan (Novita Joseph, 2018) dan tidak dianjurkan (Tempo, 2017) bagi penderita hipertensi, antara lain:

Tabel 2.4. Bahan Anjuran dan Pantangan bagi Penderita Hipertensi

Bahan Anjuran	Bahan Pantangan
Bahan Rendah Garam	Bahan Tinggi Garam
Yoghurt	Alkohol
Susu Skim	Bacon
Bayam	Acar
Bit	Daging Babi
Brokoli	Daging Kambing
Buncis	Daging Sapi
Cokelat	Donat
Daun Singkong	Sosis

Flaxseed	Nugget
Kacang Lima	Soda
Kacang Panjang	
Kangkung	
Kembang Kol	
Kentang	
Kol	
Oatmeal	
Pisang	
Salmon	
Sawi Hijau	

2.3.4. Asam Urat

Penyakit gout adalah salah satu tipe dari arthristis (rematik) yang disebabkan terlalu banyaknya atau tidak normalnya kadar asam urat di dalam tubuh karena tubuh tidak bisa mengsekresikan asam urat secara normal/seimbang. Asam urat merupakan asam yang berbentuk kristal yang merupakan hasil akhir dari metabolisme purin, dimana purin merupakan salah satu komponen asam nukleat yang terdapat pada inti sel tubuh. Purin bisa didapatkan pada semua makanan yang berasal dari tanaman sayur, buah, kacang-kacangan dan makanan yang bersumber dari hewan seperti udang, cumi, kerang, kepiting dan ikan teri (Arjani, 2018).

Penyakit asam urat berkaitan dengan pola asupan makanan, sehingga salah satu cara pencegahan dengan mengontrol pola asupan makanan. Jika tidak mengontrol pola asupan, kadar asam urat dalam darah akan berlebihan dan menimbulkan penumpukan kristal asam urat yang apabila terbentuk pada cairan sendi, maka akan terjadi penyakit asam urat (Budiono, 2016).

2.3.4.1. Bahan Anjuran dan Pantangan bagi Penderita Asam Urat

Terdapat bahan-bahan masakan yang dianjurkan dan tidak dianjurkan bagi penderita asam urat, antara lain (Dokter Sehat, 2018):

Tabel 2.5. Bahan Anjuran dan Pantangan bagi Penderita Asam Urat

Bahan Anjuran	Bahan Pantangan
Kentang	Bayam
Yoghurt	Asparagus
Jeruk	Kembang Kol
Pepaya	Jamur
Strawberry	Udang
Buah Naga	Kepiting
Belimbing Wuluh	Remis
Jahe	Tiram
Labu Kuning	Cumi-Cumi
Sawi Hijau	Daging Sapi
Sawi Putih	Daging Kambing
Tomat	Daging Babi
Singkong	Mangga
Ubi	Durian
	Nangka
	Nanas

2.3.5. Penyakit Kulit

Penyakit kulit merupakan salah satu penyakit yang masih sangat dominan terjadi dan menjadi masalah kesehatan masyarakat Indonesia. Data Profil Kesehatan Indonesia 2008 menunjukkan bahwa distribusi pasien rawat jalan menurut *International Classification of Diseases-10* (ICD-10) di rumah sakit di Indonesia tahun 2008 dengan golongan sebab sakit “Penyakit Kulit dan Jaringan Subkutan” terdapat sebanyak 64.557 pasien baru (Depkes, 2009). Hal ini juga dibuktikan dari data Profil Kesehatan Indonesia 2010 yang menunjukkan bahwa penyakit kulit dan jaringan subkutan menjadi peringkat ketiga dari 10 penyakit terbanyak pada pasien rawat jalan di rumah sakit se-Indonesia berdasarkan jumlah kunjungan yaitu sebanyak 192.414 kunjungan dan 122.076 kunjungan diantaranya merupakan kasus baru (Kemenkes, 2011).

Penyakit kulit memiliki beberapa macam jenis dibedakan dari penyebabnya, yaitu (Lina Ulin Miranti Safrina, 2008):

1. Disebabkan infeksi oleh bakteri :

a. Bisul (*furunkulosis*)

Bisul (*furunkulosis*) merupakan infeksi yang disebabkan infeksi pada folikel rambut oleh bakteri *S aureus* . Gejala-gejalanya dapat ditunjukkan dengan timbulnya bintik merah dan meradang (*abses*) yang terasa nyeri pada tempat yang infeksi, yang merupakan pusat fluktuasi yang merupakan puncak pustule. Begitu inti bagian tengah nekrosis hancur, maka secara bertahap rasa sakit akan hilang.

Pengobatan untuk mengatasi gejala tersebut dapat menggunakan pengobatan antibakteri misalnya dengan menggunakan triklosan 2% dan pemberian flukloksasin.

b. Eritrasma

Eritrasma merupakan infeksi yang disebabkan infeksi oleh bakteri gram positif, *Corynebacterium Minitissimum*. Gejala-gejalanya dapat ditunjukkan dengan Eritrasma atau bercak berwarna merah sampai kecoklatan pada kulit yang timbul di daerah *intertrigones*a atau daerah lipatan seperti lipatan paha, dan daerah payudara.

Pengobatan untuk mengatasi gejala tersebut dengan pengobatan anti bakteri misalnya dengan *imidazol topical*, asam *fusidat topical* atau pemberian *eritromisin oral* selama 2 minggu.

2. Disebabkan infeksi oleh virus :

a. Kutil

Kutil merupakan neoplasma jinak epidermis yang disebabkan oleh virus dari kelompok *human papilloma virus* (HPV). Gejala-gejalanya dapat ditunjukkan dengan timbulnya kutil atau tonjolan seperti kembang kol yang terutama sering terdapat pada tangan, biasanya sering terdapat pada anak-anak dan usia dewasa. Kutil dapat menyebar berkelompok atau timbul di sekitar kuku.

Pengobatan untuk mengatasi gejala tersebut dapat diobati dengan obat yang dioles yang mengandung asam *salisilat* atau *glutardehid* atau dengan cara lain seperti dengan semprotan nitrogen cair.

b. Herpes Simplek

Herpes simplek disebabkan oleh *Herpes Virus Homonis* (HSV) berdasarkan antigennya ada dua tipe : tipe I merupakan tipe klasik yang berhubungan yang lebih di kenal dengan sariawan (*cold sore/stomatitis*) pada bibir dan muka.dan tipe II berhubungan dengan genital. Gejala herpes dapat ditunjukkan dengan adanya erosi yang terasa perih dan nyeri pada mukosa kulit maupun di bibir.

Pengobatan untuk mengatasi gejala tersebut dapat diobati dengan obat yang dioles atau topikal yang mengandung *asiklovir* 5% obat ini digunakan untuk mencegah replikasi virus, bukan untuk membunuhnya dan tidak untuk menyembuhkan.

3. Disebabkan infeksi oleh jamur :

Jamur biasanya menyebabkan penyakit pada manusia antara lain adalah dermatofit (*Dermatophyte* yang berarti tumbuhan jamur). Jamur serupa ragi *Candida albicans*, yang meyebabkan infeksi jamur. Penyakit akibat infeksi jamur diantaranya:

a. Kutu Air (*Tinea Pedis*)

Kutu air disebabkan oleh jenis jamur *Tricophyton* dan merupakan infeksi jamur *Dermatophytose* yang paling banyak timbul. Gejala-gejalanya dapat ditunjukkan dengan gatal diantara jari kaki kemudian terbentuk gelembung yang lalu pecah dan mengeluarkan cairan, kulit menjadi lunak dan terkelupas, sehingga membuka peluang bagi infeksi sekunder oleh kuman. Penyebaran melalui penggunaan bersama kamar mandi dan ruang ganti, adapun sumber infeksiya adalah serpihan kulit yang terkelupas.

Pengobatan untuk mengatasi gejala tersebut dengan krim *mikonazol* atau salep *whitifild* (*benzoate* 5%, *salisilat* 5% dalam *lanolin* dan *vaselin*) dapat juga digunakan *griseofulvin* atau *ketokonazole* untuk *peroral*.

b. Kuku Kapur (*Anychomycose*)

Yang bercirikan kuku menebal, kuku keras, dan mudah patah, berwarna keputih-putihan, dan ada kalanya tidak lurus, terlebih bila sirkulasi darah di jari-jari tidak baik. Pengobatan : tebinafin oral 1 kali sehari 250 mg.

c. Panu (*Pytiriasis Versicolor*)

Infeksi ini banyak terjadi di Indonesia dan daerah tropis lain infeksi ini berupa bercak-bercak putih dan kecoklatan atau kemerahan di tengkuk, dada punggung, dan lengan, terutama hipopikmen di muka merupakan masalah. Panu (*Tinea Versicolor*) merupakan infeksi jamur yang ditandai dengan adanya bercak putih di kulit, dengan berbagai ukuran dan warna, ditutupi sisik halus dengan disertai gatal-gatal (*Macula*), atau tanpa keluhan kosmetik. Penyebabnya adalah masalah *Malassezia Fufur* suatu jamur yang terdiri dari kelompok sel dengan *Hypen* pendek diatas kulit.

Pengobatan untuk mengatasi gejala tersebut dapat dilakukan dengan memoleskan bercak-bercak dengan larutan *salisilat* 5-10% dalam spritus atau menggunakan *ketokonazole* selama 2-3 minggu.

d. Ketombe (*Dandruff, Pityriasis Capitis*)

Ketombe bercirikan terlepasnya serpih-serpih berlebihan dari kulit kepala yang biasanya disertai gatal-gatal. Menurut perkiraan, penyerpihan meningkat disebabkan oleh jamur *Pityrosporum Ovale*.

Pengobatan untuk mengatasi gejala tersebut dapat dilakukan dengan shampo yang mengandung *selensulfida* 2,5%, dan *piroctone olamine* (*Oktopiroc*). Pada kasus yang hebat (*Eczema Seborosis*) dianjurkan menggunakan gel *ketokonazole* 2%.

e. Sariawan (*Candidiasis*)

Merupakan infeksi mulut dengan gejala luka perih dan bercak-bercak putih pada mukosa mulut serta lidah, yang dapat menjalar ke tenggorokan dan *oseophagus*. Ciri lain adalah berupa radang di sudut mulut (*cheilitis*).

Pengobatan untuk mengatasi gejala tersebut secara efektif dapat dilakukan dengan *flukonazol* secara *oral*, pilihan yang kedua *itraconazol* dan *ketokonazol oral*, dan pilihan yang ketiga adalah tablet isap *amfoterisin*.

f. Eksim (*Dermatitis*)

Kelainan eksim (*Dermatitis*) adalah suatu reaksi kulit yang ditandai dengan radang dan gatal, bisa disebabkan oleh alergi pada detergen atau sabun cuci lainnya.

Pengobatan untuk mengatasi gejala tersebut dapat menggunakan obat seperti *kortikosteroid topical* iclamol salep yang dioleskan 1-3 kali sehari.

2.3.5.1. Bahan Anjuran dan Pantangan bagi Penderita Gatal Kulit

Terdapat bahan-bahan masakan yang dianjurkan (Destriyana, 2015) dan tidak dianjurkan (Info Menarik, 2016) bagi penderita hipertensi, antara lain:

Tabel 2.6. Bahan Anjuran dan Pantangan bagi Penderita Penyakit Kulit

Bahan Anjuran	Bahan Pantangan
Apel	Udang
Semangka	Kepiting
Kunyit	Remis
Lemon	Tiram
Teh Hijau	Cumi-Cumi
Bawang Putih	Telur
Jahe	Laron
Yoghurt	Gembili
Ubi	Kacang-Kacangan
	Tepung Terigu
	Susu Sapi
	Daging Sapi
	Daging Kambing
	Daging Babi

2.4. Web

Web merupakan suatu ruang informasi di mana sumber-sumber daya yang berguna diidentifikasi oleh pengenalan global yang disebut *Uniform Resource Identifier* (URI). URL dapat diibaratkan suatu alamat, di mana alamat tersebut terdiri atas (Sutarman, 2007):

1. Protokol yang digunakan oleh suatu *browser* untuk mengambil informasi.
2. Nama komputer (*server*) di mana informasi tersebut berada.
3. Jalur atau *path* serta nama *file* dari suatu informasi.

Format umum dari URL adalah sebagai berikut:
 “protokol_transfer://nama_host/path/nama_file”.

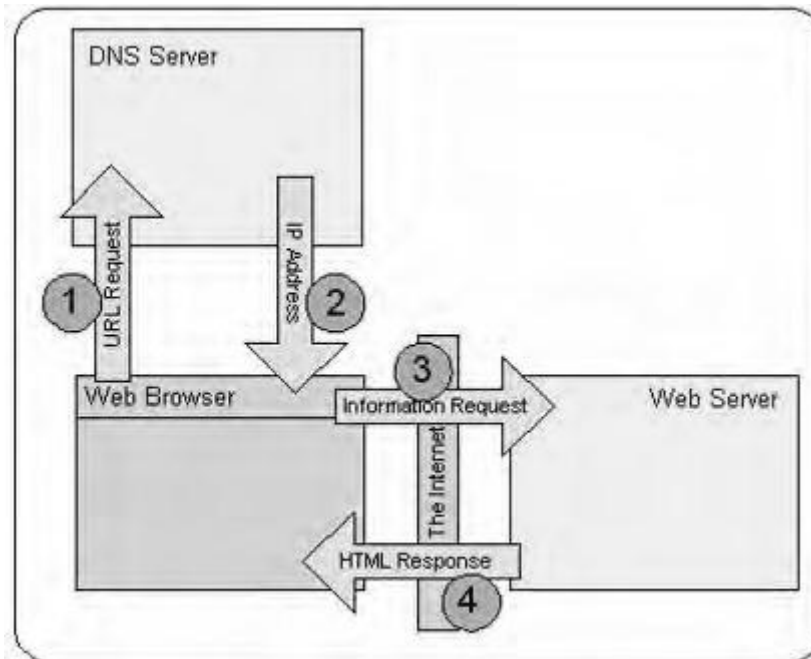
Contoh: <http://www.mine.com/e-jurnal/index.html>

Dari contoh tersebut dapat disimpulkan bahwa:

1. http adalah protokol yang digunakan.
2. www.mine.com adalah nama *host* atau *server* komputer di mana informasi yang dicari berada.
3. e-jurnal adalah jalur atau *path* dari informasi yang dicari.
4. index.html adalah nama *file* di mana informasi tersebut berada.

Sebuah halaman web diakses dengan menggunakan *web browser* dengan menuliskan URL-nya atau mengikuti *link* yang menuju kepadanya. *Uniform Resource Locator* (URL) akan menunjukkan lokasi dokumen yang dikelola oleh *web server*. URL diubah menjadi alamat IP *web server* yang bersangkutan. *Browser* kemudian mengirimkan request *Hypertext Transfer Protocol* (HTTP) ke *web server* dan akan menjawab dokumen yang diminta dalam format *Hypertext Markup Language* (HTML). HTTP adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan oleh *web server* dalam menyediakan dokumen yang diminta *web browser*. Protokol ini merupakan komunikasi jaringan komputer yang diatur dengan protokol yang memungkinkan beragam jaringan komputer untuk berkomunikasi. Protokol ini secara resmi dikenal dengan *Transmission Control Protocol* (TCP/IP) yang merupakan cara untuk mempaketkan sinyal elektronik

sehingga data tersebut dapat dikirim ke komputer lain. Gambar 2.2 menggambarkan cara kerja web yang diakses.



Gambar 2.2. Cara Kerja Web yang Diakses

Web dapat dikategorikan menjadi dua yaitu web statis dan web dinamis atau interaktif. Web statis adalah web yang menampilkan informasi-informasi yang sifatnya statis atau tetap, sedangkan web dinamis adalah web yang menampilkan informasi serta dapat berinteraksi dengan *user* yang sifatnya dinamis. Untuk membuat web dinamis dibutuhkan pemrograman web yang mempunyai dua kategori, yaitu (Sutarman, 2007):

1. *Server – side Programming*

Perintah-perintah program atau *script* dijalankan di *web server*, kemudian hasilnya dikirimkan ke *browser* dalam bentuk HTML.

2. *Client – side Programming*

Perintah program dijalankan di *web browser* sehingga ketika *client* meminta dokumen yang mengandung *script*, maka *script* tersebut akan *download* dari servernya kemudian dijalankan di *browser* yang bersangkutan.

Program web yang tergolong dalam *server side* seperti CGI/Perl, *Active Server Pages* (ASP), *Java Server Pages* (JSP), PHP dan *ColdFussion* (CFM). Sedangkan yang tergolong *client side* seperti Javascript, VBScript dan HTML. Teknologi server side yang dilakukan dalam aplikasi ini adalah PHP. PHP termasuk dalam produk *open source* sehingga *source code* dapat diubah dan didistribusikan secara bebas. PHP dapat berjalan diberbagai web server misalnya IIS, Apache dan PWS. Adapun kelebihan-kelebihan PHP adalah sebagai berikut:

1. PHP mudah dibuat dengan kecepatan akses tinggi.
2. PHP dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula.
3. PHP diterbitkan secara gratis.
4. PHP adalah termasuk bahasa yang *embedded* (bisa diletakkan dalam *tag* HTML).

2.5. Semantic Web

Semantic adalah cabang linguistik yang mempelajari makna / arti yang terkandung dalam bahasa, kode, atau jenis lain dari representasi. (Aris Kurniawan, 2019). *Semantic* biasanya dikontraskan dengan ekspresi makna sintaksis yaitu pembentukan simbol kompleks dari simbol yang lebih sederhana.

Semantic Web merupakan pengembangan dari *world wide web* dimana konten web yang ditampilkan tidak hanya dalam bahasa format manusia yang umum (*natural language*) tetapi juga dalam format yang dapat dibaca dan digunakan oleh mesin (*software*). *Semantic web* memiliki informasi yang dimiliki oleh mesin yang memiliki kecerdasan buatan sehingga mampu menemukan dan mengintegrasikan informasi dengan mudah. Dengan demikian fungsi web menjadi wadah bagi pertukaran data, informasi dan pengetahuan melalui kecerdasan buatan sehingga mengerti keinginan *user* dimana dapat diinstruksikan untuk mengambil informasi sesuai kriteria tertentu. Tujuan dari *semantic web* adalah mengatur informasi dan prosedur. Fundamental dalam pembangunan *semantic web* adalah kreasi dan *semantic metadata*. Metadata terdiri dari dua bagian, yaitu:

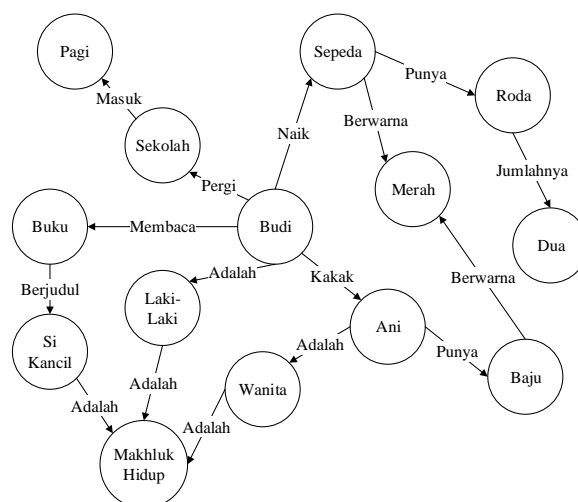
1. Penggambaran sebuah dokumen. Contohnya adalah halaman web atau bagian dari suatu dokumen seperti sebuah paragraf.
2. Penggambaran entitas didalam suatu dokumen. Contohnya adalah seseorang atau sebuah perusahaan.

Pada semua kasus, yang terpenting bahwa metadata adalah *semantic*, yang menggambarkan semua isi dari dokumen tersebut.

Saat membuat aplikasi *semantic*, sebenarnya terdapat dua variabel yang dibangun secara sekaligus. Variabel pertama adalah web terdiri dari *protocol* komunikasi dan format web. Terdapat standar *semantic web* yang direkomendasikan W3C seperti RDF, OWL dan SPARQL. Variabel lainnya adalah *semantic* merepresentasikan makna dari web data.

2.5.1. Jaringan *Semantic*

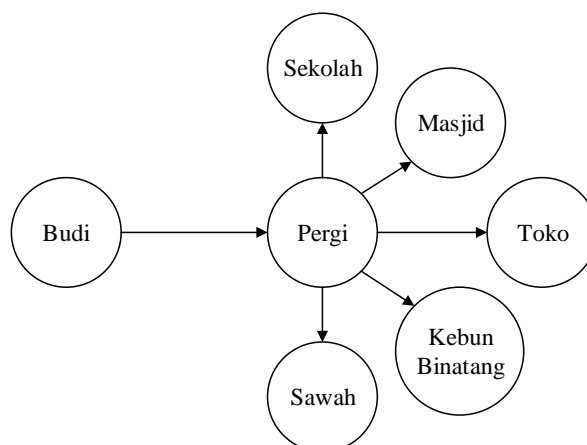
Jaringan *semantic* merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan *semantic* terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tersebut. Objek di sini bisa berupa benda atau peristiwa. Antara 2 objek dihubungkan oleh *arc* yang menunjukkan hubungan antar objek. Gambar 2.3 merupakan contoh representasi pengetahuan dengan menggunakan jaringan *semantic*.



Gambar 2.3. Contoh Jaringan *Semantic*

Salah satu kelebihan dari jaringan *semantic* adalah ‘bisa mewariskan’. Sebagai contoh, pada gambar 2.3 ada garis yang menghubungkan antara Budi dengan laki-laki, dan laki-laki ke makhluk hidup. Sehingga apabila ada pertanyaan: Apakah Budi makhluk hidup? Maka kita bisa merunut garis dari makhluk hidup, kemudiian ke laki-laki, dan akhirnya ke Budi. Sehingga terbukti bahwa Budi adalah makhluk hidup.

Sistem jaringan *semantic* ini selalu tergantung pada jenis masalah yang akan dipecahkan. Jika masalah itu bersifat umum, makan hanya memerlukan sedikit rincian. Jika ternyata masalah itu banyak melibatkan hal-hal lain, maka di dalam jaringan awalnya diperlukan rincian dari node awal Budi, apabila Budi hendak pergi ke berbagai tempat. Node Budi dihubungkan dengan node baru, yaitu pergi (Nuriana Ayuningtyas, 2009).



Gambar 2.4. Perluasan Jaringan *Semantic*

2.5.2. Komponen-Komponen dalam *Semantic Web*

Pembuatan *semantic web* dimungkinkan dengan adanya sekumpulan standar yang dikoordinasi oleh *World Wide Web Consortium* (W3C). Standar yang paling penting dalam pembangunan *semantic web* adalah XML, XML Schema, RDF, RDF Schema dan OWL. Komponen – komponen dalam *semantic web* ini yang memungkinkan komunikasi dan interaksi pada level mesin (Niko Ibrahim, 2007).

1. XML dan XML *Schema*

Extensible Markup Language (XML) merupakan bahasa *markup* yang didesain untuk menjadi sarana yang mudah dalam mengirimkan dokumen melalui web. Berbeda dengan *Hypertext Markup Language* (HTML), XML memungkinkan penggunaannya untuk mendefinisikan *custom tag*. XML Schema merupakan bahasa yang digunakan untuk mendefinisikan sekumpulan aturan (*schema*) yang harus dipatuhi oleh dokumen XML. Struktur dari dokumen XML yang dibuat harus sesuai dengan *schema* yang telah didefinisikan tersebut.

2. RDF dan RDF *Schema*

Resource Description Framework (RDF) adalah spesifikasi yang dibuat oleh W3C sebagai metode umum untuk memodelkan informasi dengan menggunakan sekumpulan format sintak. Ide dasar dari RDF adalah bagaimana kita dapat membuat pertanyaan mengenai sebuah *resource* web dalam bentuk ekspresi subjek (S), predikat (P), objek (O). Dalam terminologi RDF, SPO ini seringkali disebut dengan istilah N-triple. Subjek mengacu pada *resource* yang ingin dideskripsikan. Predikat merupakan komposisi yang menerangkan sudut pandang dari subjek yang dijelaskan objek, sementara subjek dan objek merupakan entitas. Objek di dalam RDF dapat menjadi subjek yang diterangkan oleh objek lainnya. Dengan inilah objek dapat berupa masukkan yang dapat diterangkan secara jelas dan detail, sesuai dengan keinginan pengguna yang memberikan masukkan. Dengan menggunakan RDF, *website* dapat menyimpan dan melakukan pertukaran informasi antar web. RDF telah digunakan pada aplikasi-aplikasi berikut:

1. RDF *Site Summary* (RSS)

RSS memberikan informasi *update* sebuah *website* tanpa pengunjung perlu mengunjungi *website* tersebut.

2. *Friend of a Friend* (FOAO)

Didesain untuk mendeskripsikan orang-orang, ketertarikan dan hubungan mereka.

3. *Semantically-Interlinked Online Communities* (SIOC)

Menerangkan komunitas *online* dan menciptakan koneksi antara diskusi berbasis internet seperti *message board*, *blog* maupun *mailing list*. *RDF schema* dapat dipandang sebagai kamus data atau *vocabulary* untuk mendeskripsikan *properties* dan *classes* dari *resources* RDF.

3. *Ontology Web Language* (OWL)

OWL adalah suatu bahasa yang dapat digunakan oleh aplikasi-aplikasi yang bukan sekedar menampilkan informasi tersebut pada manusia melainkan juga yang perlu memproses isi informasi. *Ontology* sendiri dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah tersebut dengan cara yang lebih mudah atau dengan pengertian lain adalah representasi istilah beserta hubungannya. Ketika informasi yang ada dalam dokumen perlu untuk diproses oleh aplikasi atau mesin, OWL dapat digunakan untuk merepresentasikan makna suatu istilah secara eksplisit sekaligus hubungan antara istilah-istilah tersebut.

Dengan menggunakan OWL, kita dapat menambah *vocabulary* tambahan disamping *semantic* formal yang telah dibuat sebelumnya menggunakan XML, RDF dan RDF Schema. Hal ini sangat membantu penginterpretasian mesin yang lebih baik terhadap isi web. Untuk mendeskripsikan *properties* dan *classes*, OWL menambahkan *vocabulary* seperti:

1. “*among others*”
2. Relasi antar *classes* (misalnya : “*disjointness*”)
3. Kardinalitas
4. Kesamaan (*equality*)
5. *Characteristic property*
6. *Enumerated classes*

OWL memiliki tiga *sub-language* yaitu:

1. OWL Lite

Mendukung pengguna yang memerlukan klasifikasi hirarki dan dalam batasan yang sederhana.

2. OWL DL

Mendukung konstruksi seluruh OWL, tetapi hanya dapat digunakan pada batasan tertentu.

3. OWL Full

Diperuntukkan bagi pengguna yang menginginkan maksimum penggunaan dan kebebasan sintaksis.

2.5.3. Keuntungan *Semantic*

Keuntungan yang dimiliki oleh *semantic web* adalah sebagai berikut:

1. Waktu yang diperlukan untuk mendapatkan informasi yang dicari lebih singkat.
2. Pekerjaan pencarian yang dilakukan manusia dapat digantikan oleh mesin.

2.6. *Ontology*

Dalam literatur kecerdasan buatan terdapat beberapa pengertian *ontology*. *Ontology* adalah istilah yang dipinjam dari filosofi yang mengacu kepada ilmu untuk menggambarkan jenis-jenis entitas di dunia dan bagaimana mereka berhubungan. Menurut Barnaras pada proyek Kactus (Nuriana Ayuningtyas, 2009) memberikan definisi *ontology* yaitu: “Penjelasan secara eksplisit dari konsep terhadap representasi pengetahuan pada *knowledge base*”. Proyek Sensus (Nuriana Ayuningtyas, 2009) juga memberikan definisi: “Sebuah *ontology* adalah sebuah struktur hirarki dari istilah untuk menjelaskan sebuah domain yang dapat digunakan sebagai landasan untuk sebuah *knowledge base*” (Nuriana Ayuningtyas, 2009). Pengertian lain mengemukakan *ontology* adalah sebuah uraian formal yang menjelaskan tentang sebuah konsep dalam suatu domain tertentu (*classes*, terkadang disebut *concepts*), properti dari setiap konsep yang menjelaskan bermacam-macam fitur dan atribut sebuah *concepts* (*slots*, terkadang disebut *roles* atau *properties*) dan batasan pada *slots* (*facets*, terkadang disebut *role restriction*). Sebuah *ontology* bersama dengan seperangkat *instances* (menyatakan objek pada suatu domain) dari *class* membentuk sebuah *knowledge base*. Pendapat lain mengatakan bahwa *ontology* terbentuk oleh 4 tuple (C,R,I,A),

C adalah *concept*, R adalah *relation*, I adalah *instance*, dan A adalah *axiom*. *Axiom* digunakan untuk menyediakan informasi mengenai *class* dan *properties*, sebagai contoh batasan pada *properties*.

2.6.1. Alasan Menggunakan *Ontology*

Terdapat beberapa alasan untuk menggunakan *ontology*, yaitu (Nuriana Ayuningtyas, 2009):

1. Menjelaskan suatu domain secara eksplisit.
Memberikan struktur hirarki dari istilah untuk menjelaskan sebuah domain dan bagaimana mereka berhubungan.
2. Berbagi pemahaman dari informasi yang terstruktur.
Sebagai contoh beberapa *website* yang berbeda mempunyai informasi medis atau menyediakan servis medis *e-commerce*. Jika *website* tersebut dipakai bersama dan dipublikasikan dengan dasar *ontology* yang sama maka perangkat lunak dapat mengekstrak dan mengumpulkan informasi dari *site* yang berbeda. Perangkat lunak tersebut dapat menggunakan kumpulan informasi tersebut untuk menjawab permintaan *user* atau sebagai *data input* untuk aplikasi lainnya.
3. Penggunaan ulang domain pengetahuan.
Apabila ingin membangun *ontology* yang luas dapat mengintegrasikan dengan beberapa *ontology* yang sudah ada.

2.6.2. Komponen-Komponen dalam *Ontology*

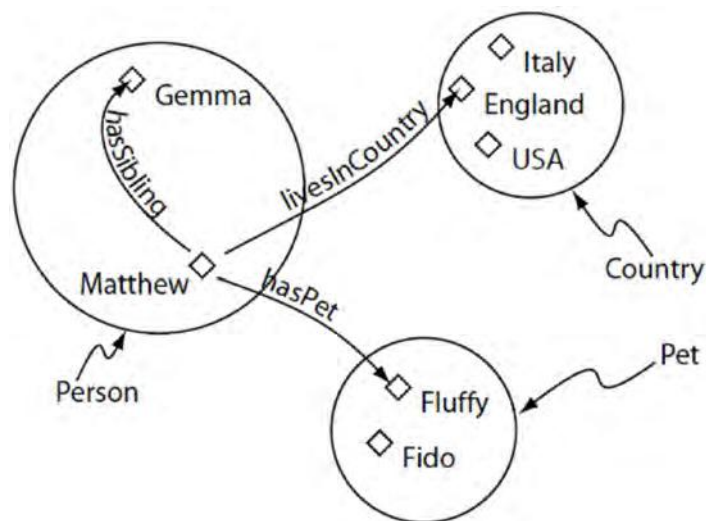
Ontology memiliki beberapa komponen yang dapat menjelaskan *ontology* tersebut, yaitu (Nuriana Ayuningtyas, 2009):

1. *Instance* atau *individual* digunakan untuk merepresentasikan elemen pada suatu domain. Contoh *instance* dapat dilihat pada Gambar 2.5.



Gambar 2.5. Representasi *instance* atau *individual*

2. *Class* merupakan titik pusat *ontology*. *Class* menjelaskan sebuah konsep dalam suatu domain yang terdiri dari beberapa *instance* atau *individual*. *Class* juga dikenal sebagai *concept*, *object* dan *categories*. Sebuah *class* memiliki *subclasses* yang menyatakan *concept* yang lebih spesifik dari *superclass*. Contoh *class* dapat dilihat pada Gambar 2.6.

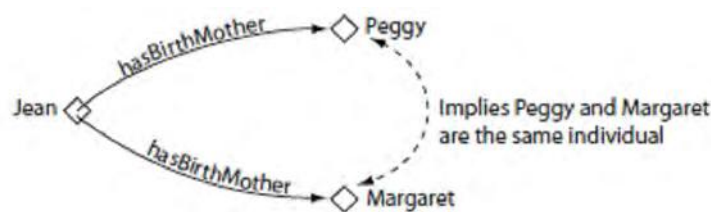


Gambar 2.6. Representasi *class* yang terdiri dari *instance*

3. *Properties* atau *slot*. *Properties* atau *slot* terdiri dari dua jenis, yaitu *object properties* dan *datatype properties*. *Object properties* akan menghubungkan *instance* dengan *instance* sedangkan *datatype properties* akan menghubungkan *instance* dengan *datatype value* seperti *text*, *string*, atau *number*. Pada *object properties*, terdapat beberapa jenis *properties*, yaitu:

a. *Functional Properties.*

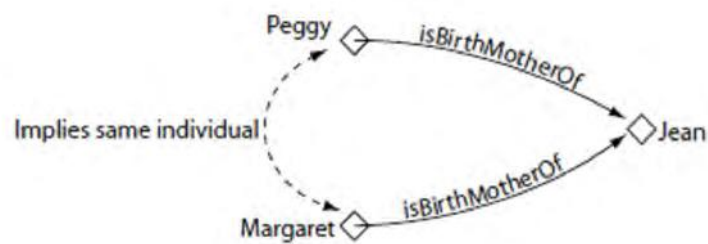
Functional properties adalah sebuah individu yang berhubungan hanya dengan satu individu. *Functional property* disebut juga sebagai *single valued property* atau *feature*. Sebagai contoh pada Gambar 2.7 menunjukkan individu Jean *hasBirthMother* Peggy dan individu Jean *hasBirthMother* Margaret, *hasBirthMother* merupakan *functional property*, dapat disimpulkan bahwa Peggy dan Margaret adalah individu yang sama karena Jean hanya memiliki satu *BirthMother*.



Gambar 2.7. Contoh *Functional Properties*

b. *Inverse Functional Properties.*

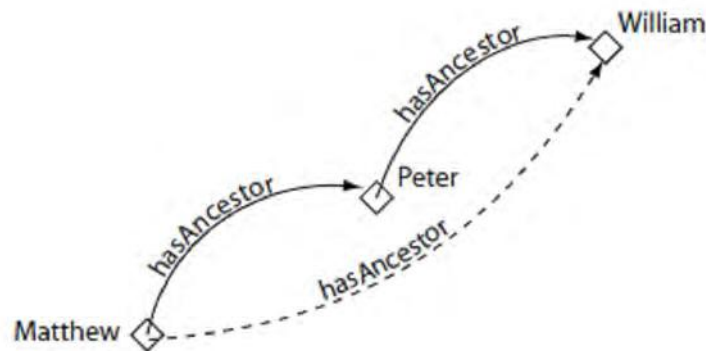
Jika sebuah properti adalah *inverse functional*, maka *inverse property* tersebut adalah *functional*, sebuah individu berhubungan hanya dengan satu individu. Sebagai contoh pada Gambar 2.8 menunjukkan *inverse functional property isBirthMotherOf* yang merupakan *inverse property* dari *hasBirthMother*, *hasBirthMother* adalah *functional*, *isBirthMother* adalah *inverse functional*. Jika dikatakan bahwa Peggy merupakan *birthmother* dari Jean, dapat dikatakan juga bahwa Margaret merupakan *birthmother* dari Jean. Sehingga dapat disimpulkan bahwa Peggy dan Margaret adalah individu yang sama.



Gambar 2.8. Contoh *Inverse Functional Properties*

c. *Transitive Properties.*

Jika sebuah properti *transitive*, properti menghubungkan individu A dengan individu B serta menghubungkan individu B dengan individu C, maka dapat disimpulkan bahwa individu A berhubungan dengan individu C melalui properti P. Sebagai contoh pada Gambar 2.9 menunjukkan *transitive property hasAncestor*. Jika individu Matthew mempunyai *ancestor* Peter, Peter mempunyai *ancestor* William, maka dapat disimpulkan bahwa Mathew mempunyai *ancestor* William.

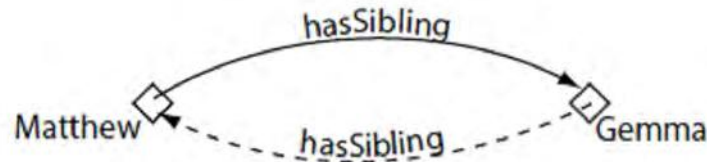


Gambar 2.9. Contoh *Transitive Properties*

d. *Symetric Properties.*

Properti P adalah *symetric* jika properti tersebut menghubungkan A ke individu B kemudian menghubungkan B ke individu A. Sebagai contoh pada Gambar 2.10, individu Mathew berhubungan dengan individu Gema melalui *hasSibling property*. Dengan kata lain, jika

Mathew bersaudara dengan Gema maka dapat disimpulkan bahwa Gema bersaudara dengan Mathew.



Gambar 2.10. Contoh *Symetric Properties*

2.6.3. Pembangunan *Ontology*

Terdapat beberapa tahapan yang dilakukan dalam pengembangan *ontology* yaitu (Nuriana Ayuningtyas, 2009):

1. Menentukan domain dan batasan *ontology*.

Dalam mengembangkan *ontology* dimulai dengan mendefinisikan domain dan batasan dengan menjawab pertanyaan berikut:

- a. Domain apa yang akan melingkupi *ontology*?
- b. Mengapa *ontology* digunakan?
- c. Apa jenis pertanyaan terhadap informasi dalam *ontology* sehingga perlu menyediakan jawaban?

Salah satu cara dalam menentukan batasan dalam *ontology* adalah dengan membuat daftar pertanyaan yang harus dapat dijawab oleh *knowledge base* atau yang biasa disebut *competency questions*.

2. Mempertimbangkan penggunaan *ontology* yang sudah ada.

Ontology yang sudah ada dapat diperhalus dan diperluas untuk domain dan *task* yang akan dibuat. Penggunaan *ontology* yang sudah ada merupakan persyaratan apabila sistem yang akan dibuat akan berinteraksi dengan aplikasi lainnya yang telah dilakukan pada suatu *ontology* atau perbendaharaan kata yang dikontrol. Banyak *ontology* yang telah tersedia dan dapat dimasukkan dalam pengembangan *ontology* yang dilakukan.

3. Menentukan istilah penting dalam *ontology*. Menulis daftar istilah yang akan dijelaskan ke *user*. Terdapat beberapa pertanyaan yang dapat membantu penentuan istilah, yaitu:
 - a. Istilah apa saja yang akan diperbincangkan?
 - b. Apa yang akan menjadi jawaban mengenai istilah-istilah tersebut?
 - c. Properti apa saja yang dimiliki istilah tersebut?

Sebagai contoh pada *ontology* minuman anggur, istilah yang penting meliputi minuman anggur, anggur, tempat membuat anggur, warna minuman anggur, bentuk, cita rasa dan kadar gula. Pada dasarnya, hal ini dibutuhkan untuk mendapat daftar istilah yang menyeluruh tanpa khawatir tumpang tindih antara *concept*, hubungan antara istilah dan properti dari *concept* atau *concept* tersebut termasuk *class* atau *slot*.

4. Mendefinisikan *class ontology* dan menyusun *class* dalam hirarki taksonomi (*subclass–superclass*).

Terdapat beberapa pendekatan dalam pengembangan hirarki *class* (Nuriana Ayuningtyas, 2009), yaitu :

- a. Proses pengembangan ***top-down*** dimulai dengan mendefinisikan *concept* umum dalam domain dilanjutkan dengan *concept* yang lebih spesifik.
 - b. Proses pengembangan ***bottom-up*** dimulai dengan mendefinisikan *class* yang paling spesifik kemudian dikelompokkan menjadi *class* dengan konsep yang lebih umum.
 - c. Proses pengembangan ***combination*** adalah sebuah kombinasi antara pendekatan *top-down* dan *bottom-up*. Mendefinisikan konsep yang menonjol terlebih dahulu kemudian menggeneralisasi dan mengkhususkan konsep tersebut.
5. Mendefinisikan *slot* atau *properties* dan menjabarkan nilai dari *slot* tersebut.
 6. Mendefinisikan *facets* pada *slots*.

Slot dapat memiliki *facet* yang berbeda dalam menggambarkan tipe nilai, nilai tersebut dapat berupa *cardinality* (sejumlah nilai) dan fitur lainnya. Berikut merupakan beberapa *facet* yang umum digunakan:

a. *Slot cardinality*.

Slot cardinality mendefinisikan sejumlah nilai yang dimiliki oleh *slot*. Beberapa sistem hanya membedakan *single cardinality* (mempunyai satu nilai) dengan *multiple cardinality* (mempunyai beberapa nilai).

b. *Slot value type*.

Sebuah tipe *facet* menggambarkan tipe dari nilai yang dapat mengisi *slot*. Berikut merupakan tipe nilai yang umum digunakan:

- 1) *String*, tipe nilai yang paling sederhana yang biasa digunakan untuk *slot*.
- 2) *Number*, terkadang tipe nilai yang lebih spesifik seperti *Float* dan *Integer* yang digunakan dalam penggambaran *slot* dengan nilai numerik.
- 3) Boolean slot mempunyai *yes-no flag*.
- 4) *Enumerated slot* merinci daftar dari spesifik nilai yang diperbolehkan untuk *slot*.
- 5) *Instance* membuat hubungan antara individual.

7. Membuat *instances*.

Langkah terakhir adalah membuat *individual* atau *instances* pada *class* dalam suatu hirarki. Dalam mendefinisikan *individual* atau *instances* pada *class* harus diperhatikan persyaratan berikut:

- a. Pemilihan *class*.
- b. Pembuatan *individual* atau *instances* pada *class* tersebut. *Knowledge base* selanjutnya dapat dibuat dengan mendefinisikan *individual* atau *instances* dari *class* yang terisi pada nilai spesifik *slot* dan *slot* batasan tambahan (*facets*).
- c. Mengisi nilai *slot*.

2.7. *Tool Ontology*

Ontology dapat dibuat dan dijalankan menggunakan beberapa *tool*, yaitu *protege* dan *jena fuseki*. *Protege* berguna untuk membuat model *ontology*, sedangkan *jena fuseki* untuk menjalankan model *ontology* yang telah dibuat.

2.7.1. Protege

Protege merupakan tool *ontology* dengan *platform open source* untuk membangun domain model dan aplikasi *knowledge based*. Protege mengimplementasikan struktur pemodelan *knowledge* dan dapat memvisualisasikan hasil *ontology* dalam berbagai format. Sebuah *ontology* menggambarkan *concept* dan hubungan-hubungan yang penting dalam domain yang khusus, yang menyediakan kosa kata dalam domain tersebut. Dalam beberapa tahun terakhir, *ontology* telah diadopsi dalam bisnis dan komunitas ilmiah seperti *scientific knowledge portal*, manajemen informasi, integrasi informasi, *electronic commerce* dan *semantic web service*. Pada Protege terdapat dua cara dalam pemodelan *ontology*, yaitu:

1. Protege Frame editor *User* dapat membangun *ontology* dalam *frame-based* dengan kesepakatan dengan *Open Knowledge Base Connectivity Protocol* (OKBC). Pada model ini, sebuah *ontology* terdiri dari seperangkat *class* yang terorganisir pada suatu hirarki yang merepresentasikan sebuah domain, seperangkat slot yang berhubungan dengan *class* serta *instance* dari tiap *class* tersebut.
2. Protege OWL Protege-OWL editor merupakan kelanjutan dari Protege yang mendukung *Ontology Web Language* (OWL). OWL merupakan pengembangan mutakhir standar bahasa *ontology* yang disahkan oleh *World Wide Web Consortium* (W3C) untuk mempopulerkan *semantic web vision*. Protege-OWL editor memungkinkan *user* untuk :
 - a. Mengambil dan menyimpan OWL dan RDF *ontology*.
 - b. Mengubah dan memvisualisaikan *class*, *properties* dan *Semantic Web Rule Language* (SWRL).
 - c. Menjabarkan karakteristik *class* secara logis sebagai ekspresi OWL.
 - d. Mengeksekusi penalaran seperti *description logic classifier*.
 - e. Mengubah OWL individual untuk *semantic web*.

Protege-OWL berhubungan secara erat dengan Jena dan mempunyai *open source Java API* untuk pengembangan *Semantic Web Service*. Jena merupakan *framework* berbasis Java untuk mengkonstruksi aplikasi

Semantic web. Framework ini menyediakan lingkungan pemrograman RDF, RDF Schema OWL dan SPARQL.

Protege-OWL *Application Programming Interface* (API) adalah sebuah *Java Library open source* untuk *Ontology Web Language* (OWL) dan RDF. API menyediakan metode untuk mengambil dan menyimpan *file* OWL, menanyakan OWL data model, menjalankan penalaran berdasarkan *Description Logic Engines* dan sebagai *Graphical User Interface*. API didesain untuk dapat digunakan dalam dua konteks, yaitu:

- a. Pengembangan komponen yang dieksekusi dalam Protege-OWL *Editor User Interface*.
- b. Pengembangan aplikasi *stand-alone* (contoh : aplikasi Swing, Servlet dan plug-in Eclipse).

2.7.2. Jena Fuseki

Jena adalah sebuah *framework* Java untuk membangun aplikasi web *Semantic*. Jena menyediakan perpustakaan Java yang luas untuk membantu *developer* mengembangkan kode yang menangani RDF, RDFS, RDFa, OWL dan SPARQL sesuai dengan rekomendasi dari W3C yang diterbitkan. Jena juga menyertakan *rule-based inference engine* untuk melakukan penalaran berdasarkan *ontology* OWL dan RDFS, serta berbagai strategi penyimpanan dalam penyimpanan tiga kali lipat RDF pada memori atau *disk*.

Jena awalnya dikembangkan oleh para peneliti di HP Labs, dimulai di Bristol, Inggris, pada tahun 2000. Jena selalu menjadi proyek *open-source*, dan telah banyak digunakan dalam berbagai aplikasi web *Semantic* dan aplikasi demo. Pada tahun 2009, HP memutuskan untuk memfokuskan kembali aktivitas pengembangan dari dukungan langsung pengembang Jena, meskipun tetap mendukung tujuan proyek. Akhirnya, tim proyek berhasil mengajukan permohonan agar Jena diakuisisi oleh Apache Software Foundation pada November 2010 (Nuriana Ayuningtyas, 2009).