

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Berikut adalah beberapa jurnal terdahulu dengan beberapa judul yang menggunakan penerapan metode *Dempster Shafer* dengan *Teorema Bayes* dapat dilihat dibawah ini :

1. Penelitian yang dilakukan Amanda Patria Putra (2015) dengan judul penelitian analisa perbandingan metode *Certainty Factor* dan *Dempster Shafer* pada sistem pakar diagnosa penyakit Diabetes Melitus. Penelitian sistem pakar diagnosa penyakit diabetes melitus telah beberapa kali dilakukan dengan dua metode berbeda yaitu *certainty factor* dan *dempster shafer*. Meski masing-masing peneliti menyatakan telah berhasil dalam jurnalnya, namun metode manakah sebenarnya yang lebih tepat digunakan pada diagnosa jenis penyakit diabetes melitus, yang mana gejala-gejalanya hampir sama. *Certainty factor* merupakan metode sistem pakar yang tujuannya untuk mengakomodasi ketidakpastian pemikiran seorang pakar dengan nilai kepastian, sedangkan *dempster shafer* merupakan teori pembuktian matematika berdasarkan nilai belief dan plausability. Untuk membandingkan keduanya, dibuatlah suatu *prototipe* sistem pakar dengan basis pengetahuan dan sampel 100 data pasien yang sama. Dari hasil analisa statistik, kesimpulannya, ada perbedaan hasil diagnosa antara kedua metode dan

metode *dempster shafer* lebih tepat digunakan pada sistem pakar diagnosa penyakit diabetes melitus.

2. Penelitian yang dilakukan Eli Rosmita Ritonga (2017) dengan judul penelitian sistem pakar diagnosa penyakit paru-paru pada anak dengan metode *Dempster-Shafer*. Sistem pakar merupakan program *Artificial Intelligence* yang menggabungkan basis pengetahuan dengan mesin inferensi. Ini merupakan bagian perangkat lunak spesialisasi tingkat tinggi, yang berusaha menduplikasi fungsi seorang pakar dalam satu bidang keahlian tertentu. Sistem pakar memprediksi penyakit paru pada anak ini bertujuan untuk mengetahui bagaimana cara mendiagnosa penyakit paru pada anak sehingga dapat dilakukan penanganan yang tepat sesuai dengan ciri-ciri penyakit yang terdapat pada anak tersebut. Didalam penerapan sistem pakar ini dibantu dengan metode *Dempster Shafer*. *Dempster Shafer* adalah suatu teori matematika untuk pembuktian berdasarkan *belief functions and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Atas dasar tersebut maka akan dibuat sistem yang dapat membantu *user* untuk dapat mendiagnosa penyakit paru pada anak sesuai dengan gejala-gejala yang terdapat pada anak yang terserang penyakit.
3. Penelitian yang dilakukan Syailendra Orthegea (2017) dengan judul penelitian Implementasi metode *Dempster-Shafer* untuk mendiagnosa penyakit tanaman padi. Penyakit tanaman padi seringkali mengakibatkan pertumbuhan tanaman

padi menjadi terganggu serta membuat produksi padi menjadi gagal. Timbulnya penyakit pada padi berasal dari bakteri, jamur, virus, dan selain itu kekurangan unsur hara juga termasuk penyakit. Penyakit pada tanaman padi juga mempunyai pengaruh yang besar terhadap pertumbuhan tanaman, misalkan tanaman berubah menjadi kerdil dan berubah warna, misalnya daun menguning atau mengering, serta dapat mengakibatkan tanaman mati. Dalam penelitian ini, sistem dikembangkan menggunakan metode *Dempster-Shafer* sebagai media diagnosa penyakit tanaman padi. Pengumpulan data dalam penelitian ini menggunakan studi literatur, metode wawancara dan metode observasi. Hasil dari penelitian ini adalah sebuah sistem untuk mendiagnosa penyakit tanaman padi menggunakan metode *Dempster-Shafer* yang memuat berbagai gejala, penyebab, solusi dan hasil diagnosis yang berdasarkan basis pengetahuan para pakar atau para ahli di bidang pertanian. Dari kasus uji yang telah dilakukan, hasil dari pengujian akurasi yaitu 90% yang menunjukkan bahwa aplikasi berfungsi dengan baik sesuai dengan metode *Dempster-Shafer*.

4. Penelitian yang dilakukan Hengki Tamando Sihotang (2018) dengan judul penelitian sistem pakar mendiagnosa penyakit *Herpes Zoster* dengan menggunakan metode *Teorema Bayes*. Penyakit *Herpes Zoster* adalah salah satu penyakit kulit yang sangat susah untuk diobati dan semua orang pasti bisa mengalaminya, karakteristik penyakit ini ditandai dengan adanya *vesikuler unilateral* yang berkelompok dengan nyeri yang ditandai dengan *radikuler* sekitar *dermatom*. Penelitian ini bertujuan untuk membangun

metode *teorema bayes* menggunakan *Visual Basic* 2008 sebagai alat bantu untuk mendiagnosa penyakit kulit. Dengan aplikasi ini tidak harus menunggu lama untuk mengantisipasi pengobatan secara cepat dan tepat. Cara menggunakan aplikasi ini yaitu admin menginputkan pertanyaan berupa gejala-gejalayang dialami oleh pasien, selanjutnya sistem akan mengolah semua jawaban pasien menggunakan metode *bayes* dan sistem akan mengeluarkan *output* berupa hasil diagnosa jenis penyakit. Sistem yang dibangun dapat membantu pasien dalam mengetahui jenis penyakit yang sedang diderita pasien dan sesuai dengan analisa pakar penyakit kulit.

5. Penelitian yang dilakukan Sari Murni (2019) dengan judul penelitian penerapan metode *Teorema Bayes* pada sistem pakar untuk mendiagnosa penyakit lambung. Penyakit lambung dapat dialami oleh siapapun, secara tiba-tiba lambung terasa sakit tidak menentu. Rasa sakit lambung dapat diatasi dengan minum obat. Tidak mudah mengenal sakit yang disebabkan gangguan pada lambung. Terkadang masuk angin yang berlebihan dan terus menerus dapat mengakibatkan gangguan pada lambung. Lambung merupakan organ pencernaan yang berbentuk seperti kantong dan terletak diperut kiri rongga perut di atas diafragma, terdiri dari Kardiak, Fundus, dan Pyorus. Menurut kepakaran penyakit lambung terdiri dari Penyakit Gastritis, Penyakit Dyspepsia, Penyakit GERD (*Gastroesophageal Reflux Disease*). Untuk diagnosa gejala sakit, melalui seorang pakar lambung akan diterapkan sistem pakar dengan metoda *Theorema Bayes*. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan

masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant*. Perhitungan metode *Bayes* dalam mendiagnosa penyakit lambung pada sistem pakar dirancang berdasarkan algoritma *Bayes* yaitu perhitungan sesuai dengan gejala-gejala penyakit yang diderita seseorang. Penyakit lambung diberi kode P01, P02, P03 ; gejala penyakit diberi kode G001 s.d G027. Dari perhitungan metode *Bayes* diperoleh Nilai Probabilitas dengan ada tidaknya penggunaan *Theorema Bayes* dalam setiap *range*. Sistem pakar juga memberi *advice* pengobatan medis berdasarkan diagnosa penyakit.

## **II.2. Konsep Dasar Sistem Pakar**

Konsep dasar sistem pakar mencakup beberapa persoalan mendasar, antara lain apa yang dimaksud dengan keahlian, siapa yang disebut pakarm bagaimana keahlian dapat ditransfer, dan bagaimana sistem bekerja, Keahlian sering dicapai dari pelatihan, membaca, dan mempratekkan, dan konsep dasar sistem pakar adalah yang mengandung keahlian, ahli, pengalihan keahlian, inferensi, dan aturan (Lukman Riyadi; 2016 : 30).

### **II.2.1. Sistem**

Secara etimologis, sistem informasi berasal dari kata “sistem” dan “informasi”. Sistem adalah seperangkat komponen yang saling berinteraksi, saling terkait, saling bergantung yang berfungsi secara keseluruhan untuk mencapai tujuan tertentu. Sistem yang efektif harus sinergis. Sistem biasanya beroperasi di lingkungan yang berada di luar dirinya sendiri. Sistem juga dapat didefinisikan sebagai kombinasi antara personil, bahan, fasilitas dan peralatan yang bekerja

sama untuk mengubah masukan, (input) menjadi keluaran (output) yang berarti dan dibutuhkan. Pada tingkat yang paling mendasar, informasi adalah propagasi sebab dan akibat dalam sebuah sistem. Informasi disampaikan baik sebagai isi pesan atau melalui pengamatan langsung atau tidak langsung terhadap apapun. Hal yang dirasakan dapat ditafsirkan sebagai pesan tersendiri, dan dalam pengertian itu, informasi selalu disampaikan sebagai isi pesan, oleh karena itu informasi adalah sebuah pesan yang memiliki berbagai pengertian yang berbeda tergantung pada konteksnya. Terdapat banyak sekali persepektif teori tentang sistem informasi. Perbedaan pandangan timbul oleh berbagai latar belakang dan sudut pandang orientasi disiplin ilmu. Sistem informasi menurut O'Brien & Marakas dalam dapat berupa kombinasi orang, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, dan kebijakan serta prosedur yang mengatur, mengambil, mengubah, dan menyebarkan informasi dalam suatu organisasi. Beheshtyan dalam, mengatakan bahwa sistem informasi meliputi seperangkat individu, data / informasi, metode, perangkat lunak, perangkat keras dan komunikasi yang aktif dalam organisasi untuk memberikan informasi yang berguna untuk mempercepat dan mempermudah kegiatan, menciptakan koordinasi dan pengendalian, membantu analisis masalah, mendukung pengambilan keputusan, serta mengurangi ketidakpastian dalam pengambilan keputusan, (Cosmas Eko Suharyanto; 2017 : 1-2).

### **II.2.2. Sistem Pakar**

Sistem pakar merupakan bagian dari artificial Intelligence (AI) yang cukup tua karena sistem ini dikembangkan pada tahun 1960. Sistem pakar yang muncul

pertama kali adalah General Purpose Problem Solver (GPS) yang dikembangkan Newel Simon. Istilah sistem pakar berasal dari istilah knowledge-based expert system. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar. “Sistem pakar adalah program komputer yang merepresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memeberikan saran”. “Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian di gunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia”. (Hengki Tamando Sihotang; 2018 : 34).

### **II.3. Mioma Uteri**

*Mioma uteri*, dikenal juga dengan sebutan *fibromioma*, *fibroid*, atau *leiomioma* merupakan *neoplasma* jinak yang berasal dari otot polos uterus dan jaringan ikat yang menumpangnya. *Mioma uteri* berbatas tegas, tidak berkapsul, dan berasal dari otot polos jaringan *fibrous* sehingga *mioma uteri* dapat berkonsistensi padat jika jaringan ikatnya dominan, dan berkonsistensi lunak jika otot rahimnya yang dominan. *Mioma uteri* adalah tumor jinak pada daerah rahim atau lebih tepatnya otot rahim dan jaringan ikat di sekitarnya. *Mioma* belum pernah ditemukan sebelum terjadinya *menarche*, sedangkan setelah *menopause*

hanya kira-kira 10% *mioma* yang masih tumbuh. Diperkirakan insiden *mioma uteri* sekitar 20%-30% dari seluruh wanita. Di Indonesia *mioma* ditemukan 2,39% - 11,7% pada semua penderita ginekologi yang dirawat. Wanita yang sering melahirkan, sedikit kemungkinannya untuk perkembangan *mioma* ini dibandingkan dengan wanita yang tak pernah hamil atau hanya satu kali hamil. Statistik menunjukkan 60% *mioma uteri* berkembang pada wanita yang tidak pernah hamil atau hanya hamil satu kali. Prevalensi meningkat apabila ditemukan riwayat keluarga, ras, kegemukan dan *nullipar*, (Islimsyaf Anwar; 2015 : 9-10).

#### **II.4. Metode *Dempster Shafer***

Teori *Dempster-Shafer* merupakan teori matematika dari bukti. Teori tersebut dapat memberikan sebuah cara untuk menggabungkan bukti dari beberapa sumber dan mendatangkan atau memberikan tingkat kepercayaan (direpresentasikan melalui fungsi kepercayaan) dimana mengambil dari seluruh bukti yang tersedia. Teori *Dempster-Shafer* pertama kali diperkenalkan oleh *Arthur P. Dempster and Glenn Shafer*, yang melakukan percobaan ketidak pastian dengan *range probabilities* daripada sebagai probabilitas tunggal. Kemudian pada tahun 1976 *Shafer* mempublikasikan teori *Dempster* pada buku yang berjudul *Mathematical Theory of Evident*. Secara umum Teori *Dempster-Shafer* ditulis dalam suatu interval, (Eli Rosmita Ritonga; 2017 : 40-41) :

*Belief* (Bel) adalah ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada



*evidence*, dan jika  $m$  bernilai 1 menunjukkan adanya kepastian. *Plausibility* (Pl) dinotasikan sebagai, (Eli Rosmita Ritonga; 2017 : 40-41) :

$$Pl(s) = 1 - Bel (-s) \dots \dots \dots [2]$$

*Plausability* akan mengurangi tingkat kepercayaan dari *evidence*. Pada teori *Dempster-Shafer* dikenal adanya *frame of discrement* yang dinotasikan dengan  $\theta$ . *Frame* ini merupakan semesta pembicaraan dari sekumpulan hipotesis. Tujuannya adalah mengaitkan ukuran kepercayaan elemen-elemen  $\theta$ . Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas ( $m$ ). Nilai  $m$  tidak hanya mendefinisikan elemen-elemen  $\theta$  saja, namun juga semua subsetnya. Sehingga jika  $\theta$  berisi  $n$  elemen, maka subset  $\theta$  adalah  $2^n$ . Jumlah semua  $m$  dalam subset  $\theta$  sama dengan 1. Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai  $m\{\theta\} = 1,0$ . Apabila diketahui  $X$  adalah subset dari  $\theta$ , dengan  $m_1$  sebagai fungsi densitasnya, dan  $Y$  juga merupakan subset dari  $\theta$  dengan  $m_2$  sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi  $m_1$  dan  $m_2$  sebagai  $m_3$ , yaitu, (Eli Rosmita Ritonga; 2017 : 40-41) :

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} M_1(X) \cdot M_2(Y)}{1 - \sum_{X \cap Y = \emptyset} M_1(X) \cdot M_2(Y)} \dots \dots \dots [3]$$

keterangan :

$m_1$  = densitas untuk gejala pertama

$m_2$  = densitas gejala kedua

$m_3$  = kombinasi dari kedua densitas diatas

$\theta$  = semesta pembicaraan dari sekumpulan hipotesis (X' dan Y')

X dan y = subset dari Z

$X'$  dan  $y'$  = subset dari  $\theta$ , (Eli Rosmita Ritonga; 2017 : 40-41)

#### II.4.1. Studi Kasus Metode *Dempster Shafer*

Pada studi kasus ini peneliti membahas Implementasi Metode Dempster-Shafer untuk Mendiagnosa Penyakit Tanaman Padi. Penyakit tanaman padi seringkali mengakibatkan pertumbuhan tanaman padi menjadi terganggu serta membuat produksi padi menjadi gagal. Timbulnya penyakit pada padi berasal dari bakteri, jamur, virus, dan selain itu kekurangan unsur hara juga termasuk penyakit. Penyakit pada tanaman padi juga mempunyai pengaruh yang besar terhadap pertumbuhan tanaman, misalkan tanaman berubah menjadi kerdil dan berubah warna, misalnya daun menguning atau mengering, serta dapat mengakibatkan tanaman mati.

1. Pada kasus ini akan diberikan contoh dengan memasukan 1 gejala. Perhitungan ini dimisalkan user memasukkan gejala bercak hitam pada pelepah daun.

Gejala 1 : Bercak hitam pada pelepah daun Dilakukan observasi Bercak hitam pada pelepah daun sebagai gejala dari penyakit dengan nilai  $m\{P3\} = 0.7$ ,  $m\{P4\} = 0.4$  untuk  $m1$  nilai densitas yang terpilih adalah yang tertinggi, maka :

$$m1\{P3\} = 0.7$$

$$m1\{\theta\} = 1 - 0.7 = 0.3$$

Dari perhitungan diatas dikarenakan gejala yang diambil hanya satu. Jadi hasil diagnosa dapat disimpulkan bahwa tanaman padi mengalami penyakit Busuk batang.

2. Pada kasus 2 (2 gejala) Pada kasus ini akan diberikan contoh dengan memasukan 2 gejala. Perhitungan ini dimisalkan user memasukkan gejala bercak hitam pada pelepah daun, dan daun berubah warna menjadi kuning/coklat/abu-abu.

Gejala 1 : Bercak hitam pada pelepah daun Dilakukan observasi Bercak hitam pada pelepah daun sebagai gejala dari penyakit dengan nilai  $m\{P3\} = 0.7$ ,  $m\{P4\} = 0.4$  untuk  $m1$  nilai densitas yang terpilih adalah yang tertinggi, maka :

$$m1\{P3,P4\} = 0.7$$

$$m1\{\theta\} = 1 - 0.7 = 0.3$$

Gejala 2 : Daun berubah warna menjadi kuning/coklat/abu-abu Kemudian dilakukan penambahan gejala daun berubah warna menjadi kuning/coklat/abuabu, setelah diobservasi gejala tersebut sebagai gejala dari penyakit dengan nilai densitas  $m\{P5\} = 0.7$ ,  $m\{P7\} = 0.8$  untuk  $m2$  nilai densitas yang terpilih adalah yang tertinggi, maka :

$$m2\{P5,P7\} = 0.8$$

$$m2\{\theta\} = 1 - 0.8 = 0.2$$

Maka dihitung nilai densitas baru untuk beberapa kombinasi dengan fungsi densitas  $m3$  dapat dilihat pada tabel dibawah ini :

**Tabel.II.1. Aturan Kombinasi Untuk  $m3$  Kasus 2**

M1	M2	
		$\{P5,P7\} = 0.8$
$\{P3,P4\} = 0.7$	$\{\theta\} = 0.56$	$\{P3,P4\} = 0.14$
$\{\theta\} = 0.3$	$\{P5,P7\} = 0.24$	$\{\theta\} = 0.06$

Sehingga dapat dihitung dengan Persamaan 9 :

$$m_3 \{P_5, P_7\} = 0.24 / (1 - 0.56) = 0.54$$

$$m_3 \{P_3, P_4\} = 0.14 / (1 - 0.56) = 0.31$$

$$m_3\{\theta\} = 0.06 / (1 - 0.56) = 0.14$$

Dari hasil perhitungan dengan metode Dempster-Shafer, nilai densitas paling tinggi adalah 0.54 dapat disimpulkan penyakit yang menyerang tanaman padi kemungkinan adalah penyakit Tungro, (Syailendra Orthege: 2017).

## II.5. Metode Teorema Bayes

Probabilitas *Bayes* merupakan salah satu cara yang baik untuk mengatasi ketidakpastian data dengan menggunakan *Formula Bayes* yang dinyatakan dengan rumus sebagai berikut :

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B)}$$

Dengan :  $p(A|B)$  : probabilitas A dan B terjadi bersama-sama

$p(B|A)$  : probabilitas B dan A terjadi bersama-sama.

$p(B)$  : probabilitas kejadian B Teori *Bayes* sudah dikenal dalam bidang kedokteran tetapi teori ini lebih banyak diterapkan dalam logika kedokteran modern. Teori ini lebih banyak diterapkan pada hal-hal yang berkenaan dengan probabilitas serta kemungkinan dari penyakit dan gejala-gejala yang berkaitan. *Teorema Bayes* merupakan metode yang baik didalam mesin pembelajaran berdasarkan data training, dengan menggunakan probabilitas bersyarat sebagai dasarnya. Metode *Bayes* juga merupakan suatu metode untuk menghasilkan estimasi parameter dengan menggabungkan informasi dari sampel dan informasi lain yang telah

tersedia sebelumnya. Keunggulan utama dalam penggunaan Metode *Bayes* adalah penyederhanaan dari cara klasik yang penuh dengan integral untuk memperoleh model marginal. Adapun bentuk dari *Teorema Bayes* untuk *evidence* tunggal E dan hipotesis ganda H1,H2,H3,..... Hn.

$$p(H_i|E) = \frac{p(H_i|E) \times p(H_i)}{\sum_{k=1}^n p(E|H_k) \times p(H_k)}$$

Keterangan :

$p(H_i |E)$  = Probabilitas hipotesis  $H_i$  terjadi jika *evidence* E terjadi

$p(E|H_i)$  = Probabilitas munculnya *evidence* E, jika hipotesis  $H_i$  terjadi

$p(H_i)$  = Probabilitas hipotesis  $H_i$  tanpa memandang *evidence* apapun.

n = Jumlah hipotesis yang mungkin.

Adapun bentuk dari *Teorema Bayes* untuk *evidence* ganda E1,E2,E3,..... En dan hipotesis ganda H1,H2,H3,..... Hn adalah sebagai berikut :

$$p(H_i|E_1, E_2, E_3 \dots E_m) = \frac{p(E_1|H_1) \times \dots \times p(E_m|H_1) \times p(H_i)}{\sum_{k=1}^n p(E_1|H_k) \times \dots \times p(E_m|H_k) \times p(H_k)}$$

, (Hengki Tamando Sihotang; 2018: 34).

### II.5.1. Studi Kasus Metode Teorema Bayes

Pada contoh kasus ini membahas sistem pakar mendiagnosa penyakit herpes zoster dengan menggunakan metode teorema bayes. Penyakit Herpes Zoster adalah salah satu penyakit kulit yang sangat susah untuk diobati dan semua orang pasti bisa mengalaminya, karakteristik penyakit ini ditandai dengan adanya vesikuler unilateral yang berkelompok dengan nyeri yang ditandai dengan radikuler sekitar dermatom. Contoh Kasus dilakukan pemeriksaan Gejala awal

Herpes Zoster melitus yang terkena penyakit dengan gejala – gejala yang muncul sebagai berikut :

$$G1 = 0.4 P(E|H1)$$

$$G2 = 0.4 P (E|H2)$$

$$G3 = 0.5 P(E|H3)$$

$$G4 = 0.5 P(E|H4)$$

$$G5 = 0.4 P(E|H5)$$

$$G6 = 0.3 P(E|H6)$$

Kemudian mencari nilai semesta dengan menjumlahkan dari hipotesa di atas :

$$\sum_{G10}^6 k - 1 = G1 + G4 + G6 + G7 + G8 + G10$$

$$= 0.4 + 0.4 + 0.5 + 0.5 + 0.4 + 0.3 = 2,5$$

Setelah hasil penjumlahan di atas diketahui, maka didapatkan rumus untuk menghitung nilai semesta adalah sabagai berikut :

$$P(H1) = \frac{H1}{\sum_{k-1}^9} = \frac{0.4}{2.5} = 0.16$$

$$P(H2) = \frac{H2}{\sum_{k-1}^6} = \frac{0.4}{2.5} = 0.16$$

$$P(H3) = \frac{H3}{\sum_{k-1}^6} = \frac{0.5}{2.5} = 0.2$$

$$P(H4) = \frac{H4}{\sum_{k-1}^6} = \frac{0.5}{2.5} = 0.2$$

$$P(H5) = \frac{H5}{\sum_{k-1}^6} = \frac{0.4}{2.5} = 0.16$$

$$P(H_6) = \frac{H_6}{\sum_{k=1}^6} = \frac{0.3}{2.5} = 0.12$$

Setelah nilai  $P(H_i)$  diketahui, probabilitas hipotesis  $H$  tanpa memandang *evidence* apapun, maka langkah selanjutnya adalah :

$$\sum_{k=1}^6 P(H_i) * P(E|H_i - n)$$

$$= (P(H_1) * P(E|H_1)) + (P(H_2) * P(E|H_2)) + (P(H_3) * P(E|H_3)) + (P(H_4) * P(E|H_4)) + (P(H_5) * P(E|H_5)) + (P(H_6) * P(E|H_6))$$

$$= (0.16 * 0.4) + (0.16 * 0.4) + (0.2 * 0.5) + (0.2 * 0.5) + (0.16 * 0.4) + (0.12 * 0.3)$$

$$= 0.064 + 0.064 + 0.1 + 0.1 + 0.064 + 0.036$$

$$= 0.428$$

Langkah selanjutnya ialah mencari nilai  $P(H_i|E)$  atau probabilitas hipotesis  $H_i$  benar jika diberikan *evidence*  $E$  :

$$P(1H | E) = \frac{0.4 * 0.064}{0.428} = 0.059813$$

$$P(2H | E) = \frac{0.4 * 0.064}{0.428} = 0.059813$$

$$P(3H | E) = \frac{0.5 * 0.1}{0.428} = 0.093458$$

$$P(4H | E) = \frac{0.5 * 0.1}{0.428} = 0.093458$$

$$P(5H | E) = \frac{0.4 * 0.064}{0.428} = 0.059813$$

$$P(6H | E) = \frac{0.3 * 0.036}{0.428} = 0.033645$$

Setelah seluruh nilai  $P(H_i|E)$  diketahui, maka jumlahkan seluruh nilai bayesnya dengan rumus sebagai berikut :

$$\begin{aligned}
\sum_{k=1}^n \text{bayes} &= \text{bayes1} + \text{bayes2} + \text{bayes3} + \text{bayes4} + \text{bayes5} + \text{bayes6} \\
&= (0.4 * 0.059813) + (0.4 * 0.059813) + (0.5 * 0.093458) + (0.5 * 0.093458) + \\
&\quad (0.4 * 0.059813) + (0.3 * 0.033645) \\
&= 0.059813 + 0.059812 + 0.11682 + 0,11682 + 0,059812 + 0,123360033 \\
&= 0.535359 \\
&= 0.535359 * 100\% = 43.5359 \%
\end{aligned}$$

Berdasarkan perhitungan dengan menggunakan metode Teorema Bayes, maka tingkat kemungkinan Gejala awal Herpes Zoster terkena penyakit 43,5359 %, (Hengki Tamando Sihotang; 2018: 34).

## II.6. Database

Basis data atau database adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam database komputer, dibutuhkan sistem manajemen basis data (database management system). Dalam pengembangan perangkat lunak tradisional yang memanfaatkan pemrosesan file, setiap kelompok pengguna menyimpan file-file-nya sendiri untuk menangani aplikasi pengolahan datanya masing-masing. Hal ini mengakibatkan adanya kerangkapan data atau disebut dengan redundancy. Redundansi dalam proses penyimpanan data yang terjadi berkali-kali dapat mengakibatkan beberapa masalah. Pertama, ada kebutuhan untuk melakukan pembaruan logis tunggal, misalnya seperti memasukkan data pada siswa baru beberapa kali: satu kali untuk setiap file tempat data siswa direkam. Hal ini menyebabkan duplikasi data. Kedua,



ruang penyimpanan terbuang ketika data yang sama disimpan berulang kali, dan masalah ini mungkin serius untuk database yang besar. Ketiga, file yang mewakili data yang sama mungkin menjadi tidak konsisten. Hal ini bisa terjadi karena update diaplikasikan pada beberapa file tapi tidak untuk file yang lain, (Cosmas Eko Suharyanto; 2017: 1).

## II.7. SQL

SQL atau disebut juga dengan SEQUEL (*Structured English Query Language*) merupakan bahasa pemrograman yang memiliki tujuan khusus dan dirancang untuk mengelola data dalam sistem manajemen *database* relasional (RDBMS-*Relational Database Management Systems*), atau untuk pengolahan aliran data dalam sistem manajemen basis data relasional. SQL memiliki tiga bagian utama yaitu bahasa pemrograman untuk mendefinisikan data (*Data Defenition Language-DDL*), untuk manipulasi dan akses data (*Data Manipulation Language-DML*) dan bagian yang digunakan untuk pengawasan/kontrol pemakai (*Data Control Language*). Bahasa SQL masih memiliki beberapa kekurangan diantaranya adalah ketiga bahasa tersebut harus terintergrasi menggunakan suatu bahasa pemrograman tertentu. Bahasa SQL atau SEQUEL dibangun atas dasar *Relational Algebra*. Secara luas pemakainya telah distandarkan dalam sebuah kerangka kerja yang terdaftar dalam *International Organization For Standardization* (ISO) pada tahun 1987, (Eko Darmanto; 2015: 406).

## II.8. My SQL


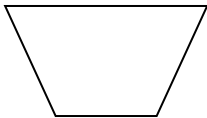

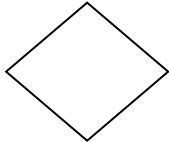
Salah satu contoh database management system adalah MySQL. MySQL adalah database open source terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan database terdepan untuk aplikasi berbasis web, yang digunakan oleh properti web profil tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi [9]. Kata “SQL” dari "MySQL" adalah singkatan dari "Structured Query Language". SQL adalah bahasa standar yang paling umum digunakan untuk mengakses database. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk men-generate laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL, (Cosmas Eko Suharyanto; 2017: 1).

## II.9. Flowchart


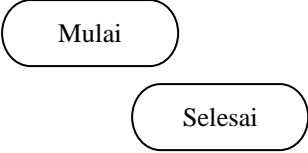

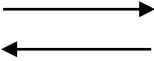
Flowchart Diagram alir bagan alir, atau bagan arus adalah sebuah jenis diagram yang mewakili algoritme, alir kerja atau proses yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutanya dihubungkan dengan panah. Saat ini aplikasi yang digunakan untuk menggambar diagram alir (flowchart) sudah banyak dibuat oleh para pengembang perangkat lunak. Pada awalnya perangkat lunak aplikasi dikelompokkan menjadi dua kelompok besar yaitu Software Komersial dan Software Berpemiliki, Seiring dengan bermunculannya beberapa perangkat- perangkat lunak maka istilah ini tidak

digunakan, saat ini perangkat lunak hanya dikelompokkan kedalam beberapa lisensi distribusi perangkat lunak. Open source, Perangkat lunak yang mengizinkan siapa pun untuk menggunakan, menyalin, dan mendistribusikan kembali, baik dimodifikasi atau pun tidak, secara gratis atau pun dengan biaya. Salah satu perangkat lunak pembuat diagram ini adalah DIA. Dikembangkan oleh Alexandar Larsson, didesain untuk merancang beberapa kebutuhan diantaranya flowchart, diagram jaringan, dan sebagainya. Flowchart dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan komputer atau keduanya. Flowchart adalah rangkaian simbol yang digunakan untuk mengkonstruksi. Simbol yang digunakan adalah, (Diaraya: 2017) :

**Tabel II.2. Simbol Flowchart**

Simbol	Keterangan
	<p><b>Dokumen:</b></p> <p>Simbol ini menyatakan masukan atau keluaran berasal dari dokumen dalam bentuk kertas yang dicetak.</p>
	<p><b>Proses Manual :</b></p> <p>Simbol ini menunjukkan pengolahan secara manual yang tidak dilakukan oleh komputer.</p>
	<p><b>Connector :</b></p> <p>Simbol ini menggambarkan penyambungan halaman yang terpisah.</p>
	<p><b>Decision :</b></p> <p>Simbol ini menyatakan pengambilan keputusan.</p>

Lanjutan Tabel II.2. Simbol Flowchart

	<p><b>Multi Documents :</b></p> <p>Simbol ini menggambarkan dokumen beserta rangkainya atau beberapa dokumen.</p>
	<p><b>Terminator :</b></p> <p>Simbol untuk pemulaan mulai atau selesai dari satu kegiatan..</p>
	<p><b>Processing :</b></p> <p>Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.</p>
	<p><b>Connecting :</b></p> <p>Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.</p>

(Sumber: Diaraya: 2017)

## II.10. UML (*Unified Modeling Language*)

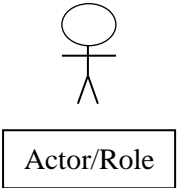

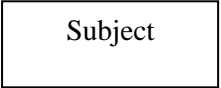
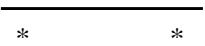
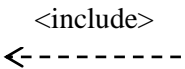
*Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek.

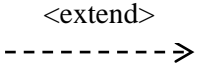

### 1. *Use case* Diagram

Elemen-elemen diagram use-case meliputi aktor, use case, batas subjek, dan seperangkat hubungan antara aktor, aktor dan kasus penggunaan, dan kasus penggunaan. Hubungan ini terdiri asosiasi, termasuk, memperluas, dan hubungan generalisasi. Masing-masing elemen ini dijelaskan selanjutnya. Aktor yang menempel pada diagram mewakili aktor. Seorang aktor bukanlah seorang namun

pengguna tertentu adalah peran yang dapat dimainkan pengguna saat berinteraksi dengan sistem. Sebuah aktor juga dapat mewakili sistem lain di mana sistem saat ini berinteraksi. Pada kasus ini, aktor secara opsional dapat diwakili oleh persegi panjang yang mengandung <<actor>> dan nama dari sistem. Pada dasarnya aktor mewakili unsur-unsur utama dalam lingkungan di mana sistem beroperasi. Aktor dapat memberikan input ke sistem, menerima output dari sistem, atau keduanya, yaitu, (Dennis, Alan; 2015) :

**Tabel II.1. Simbol Use Case**

<p>Apakah seseorang atau sistem yang memperoleh manfaat dari dan merupakan eksternal dari subjek, digambarkan sebagai figur tongkat (default) atau, jika aktor bukan manusia terlibat, persegi panjang dengan &lt;&lt;actor&gt;&gt; di dalamnya (alternatif), diberi label dengan perannya, apat dikaitkan dengan aktor lain menggunakan asosiasi spesialisasi / superclass, dilambangkan dengan panah dengan panah berongga, ditempatkan di luar batas subjek.</p>	
<p>Merupakan bagian utama dari fungsionalitas sistem, dapat memperpanjang use case lain, dapat menyertakan use case lain, ditempatkan di dalam batas sistem, diberi label dengan frase kata kerja deskriptif-kata benda.</p>	
<p>Termasuk nama subjek di dalam atau di atas, merupakan ruang lingkup subjek, mis, sistem atau individu proses bisnis.</p>	
<p>Menghubungkan aktor dengan use case yang digunakannya untuk berinteraksi</p>	
<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program</p>	

<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.	
Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem	

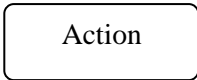



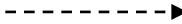



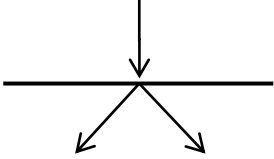
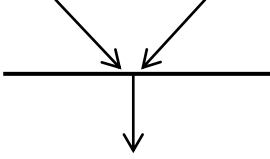
(Sumber : Dennis, Alan; 2015)

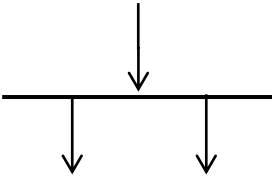
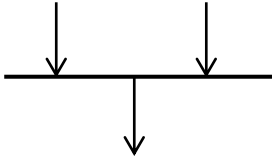
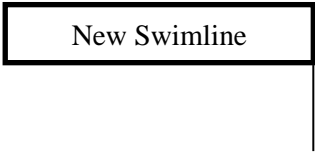
## 2. Diagram Aktivitas (*Activity Diagram*)

Model diagram aktivitas menggambarkan berbagai kegiatan yang, ketika digabungkan, mendukung proses bisnis. Proses bisnis biasanya melintasi departemen fungsional suatu produk baru melibatkan banyak kegiatan berbeda yang menggabungkan upaya banyak karyawan di banyak departemen. Dari perspektif berorientasi objek, proses ini memotong banyak objek. Banyak dari pendekatan pengembangan sistem berorientasi objek sebelumnya cenderung abaikan pemodelan proses bisnis. Namun, hari ini kami menyadari bahwa memodelkan proses bisnis itu sendiri adalah kegiatan yang sangat konstruktif yang dapat digunakan untuk memahami pertemuan tersebut persyaratan. Salah satu masalah potensial dalam membangun model proses bisnis, dari perspektif pengembangan sistem berorientasi objek, adalah bahwa mereka cenderung memperkuat pola pikir dekomposisi fungsional. Namun, selama mereka digunakan dengan benar, model proses bisnis adalah alat yang sangat kuat untuk mengkomunikasikan pemahaman analis saat ini dari persyaratan kepada pengguna. Martin Schedlbauer memberikan serangkaian praktik terbaik untuk diikuti saat memodelkan bisnis. Secara teknis, diagram aktivitas menggabungkan ide pemodelan proses dari banyak teknik yang berbeda, termasuk model acara,

statechart, dan Petri nets. Namun, diagram aktivitas UML memiliki lebih banyak kesamaan, yaitu, (Dennis, Alan; 2015) :

**Tabel II.2. Simbol Activity Diagram**

Digunakan untuk perilaku yang sederhana dan tidak dapat dikompromikan.	
Digunakan untuk mewakili serangkaian tindakan.	
Digunakan untuk mewakili objek yang terhubung ke serangkaian aliran objek.	
Menunjukkan urutan eksekusi.	
Memperlihatkan aliran suatu objek dari satu aktivitas (atau aksi) ke aktivitas lain (atau aksi)	
Menggambarkan awal dari serangkaian tindakan atau kegiatan.	
Digunakan untuk menghentikan semua aliran kontrol dan objek mengalir dalam suatu kegiatan (atau tindakan).	
Digunakan untuk menghentikan aliran kontrol tertentu atau aliran objek	
Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur, diberi label dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.	
Digunakan untuk menyatukan kembali jalur keputusan yang berbeda yang dibuat menggunakan	

Digunakan untuk membagi perilaku menjadi serangkaian kegiatan paralel atau bersamaan	
Digunakan untuk menyatukan kembali serangkaian kegiatan paralel atau bersamaan	
digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan aktivitas individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk menjalankan aktivitas (atau tindakan) Diberi label dengan nama individu atau objek yang bertanggung jawab	

(Sumber : Dennis, Alan; 2015)

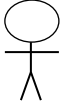
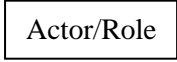
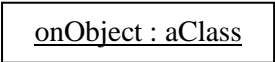
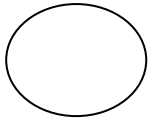
### 3. Diagram Urutan (*Sequence Diagram*)

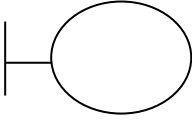
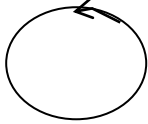



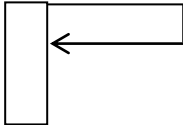
Sequence diagram adalah salah satu dari dua jenis diagram interaksi. Mereka mengilustrasikan objek yang berpartisipasi dalam use case dan pesan yang melewati mereka dari waktu ke waktu untuk satu penggunaan kasus. Diagram urutan adalah model dinamis yang menunjukkan urutan pesan yang eksplisit yang dilewatkan antar objek dalam interaksi yang ditentukan. Karena diagram urutan menekankan urutan waktu berdasarkan aktivitas yang terjadi di antara sekumpulan objek, mereka adalah sangat membantu untuk memahami spesifikasi waktu nyata dan kasus penggunaan kompleks. Diagram urutan dapat berupa diagram urutan umum yang menunjukkan semua skenario yang mungkin untuk kasus penggunaan, tetapi biasanya setiap analis mengembangkan satu set diagram



urutan contoh, masing-masing yang menggambarkan skenario tunggal dalam kasus penggunaan. Jika Anda tertarik untuk memahami aliran kontrol skenario berdasarkan waktu, Anda harus menggunakan diagram urutan untuk menggambarkan ini informasi. Diagram ini digunakan sepanjang fase analisis dan desain. Namun demikian diagram desain adalah implementasi yang sangat spesifik, seringkali termasuk objek database atau spesifikasi komponen antarmuka pengguna sebagai objek. Elemen-elemen diagram urutan menunjukkan diagram urutan contoh yang menggambarkan objek dan pesan untuk kasus penggunaan *Make Old Patient Appt*, yang menjelaskan proses oleh dimana pasien yang sudah ada membuat janji baru atau membatalkan atau menjadwal ulang janji, yaitu, (Dennis, Alan; 2015) :

**Tabel II.3. Simbol *Sequence Diagram***

<p>Adalah seseorang atau sistem yang mendapat manfaat dan bersifat eksternal dari sistem. Berpartisipasi dalam urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram. Digambarkan sebagai figur tongkat (default) atau, jika aktor bukan manusia terlibat, seperti sebuah persegi panjang dengan &lt;&lt;actor&gt;&gt; di dalamnya (alternatif).</p>	 
<p>Berpartisipasi dalam urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram</p>	
<p><i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>	

<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak</p>	
<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>	
<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>	
<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi</p>	
<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>	
<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>	

(Sumber : Dennis, Alan; 2015)


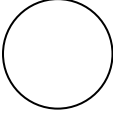




#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas

meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti, (Dennis, Alan; 2015)

:

**Tabel II.4. Tabel Simbol *Class Diagram***

Kelas/ <i>class</i> , kegunaannya untuk kelas pada struktur sistem	
Antarmuka/ <i>interface</i> , sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek	
Asosiasi/ <i>association</i> , asosiasi biasanya juga disertai dengan <i>multiplicity</i>	
Asosiasi berarah/ <i>directed association</i> , relasi antar kelas dengan makna kelas yang digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>	
Generalisasi, relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus)	
Agregasi/ <i>aggregation</i> , relasi antar kelas dengan makna semua bagian ( <i>whole part</i> )	

(Sumber : Dennis, Alan; 2015)

Fungsi dasar (mis., membuat instance baru), biasanya tidak ditampilkan secara eksplisit didiagram kelas, jadi biasanya kita tidak melihat metode konstruktor secara eksplisit pada diagram kelas. Operasi kueri membuat informasi tentang

keadaan suatu objek tersedia untuk lainnya objek, tetapi tidak mengubah objek dengan cara apa pun. Misalnya, hitung kunjungan terakhir () operasi yang menentukan kapan pasien terakhir kali mengunjungi kantor dokter akan menghasilkan objek sedang diakses oleh sistem, tetapi itu tidak akan membuat perubahan pada informasinya. Jika sebuah metode kueri hanya meminta informasi dari atribut di kelas (mis., nama pasien, alamat, telepon), maka itu tidak ditampilkan pada diagram karena kami menganggap bahwa semua objek memiliki operasi yang menghasilkan nilai atribut mereka. Operasi pembaruan mengubah nilai beberapa atau semua atribut objek, yang mungkin menghasilkan perubahan dalam status objek. Pertimbangkan untuk mengubah status pasien dari yang baru untuk saat ini dengan metode yang disebut perubahan status () atau mengaitkan pasien dengan tertentu janji dengan membuat janji (janji). Jika hasil operasi bisa mengubah keadaan objek, maka operasi harus secara eksplisit dimasukkan pada kelas diagram. Di sisi lain, jika operasi pembaruan adalah operasi penugasan sederhana, itu bisa dihilangkan dari diagram. Operasi destruktur hanya menghapus atau menghapus objek dari sistem. Misalnya, jika objek karyawan tidak lagi mewakili karyawan aktual yang terkait dengan perusahaan, karyawan dapat dihapus dari database karyawan, dan operasi destruktur akan digunakan untuk menerapkan perilaku ini. Namun, menghapus objek adalah salah satu dasarnya fungsi dan karenanya tidak akan dimasukkan pada diagram kelas, yaitu, (Dennis, Alan; 2015) :

**Tabel II.5. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

*(Sumber : Dennis, Alan; 2015)*