

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Sebagai bukti penelitian yang akan dibuat, maka penelitian akan dibandingkan terhadap penelitian sejenis yang pernah dilakukan. Adapun Penelitian sebelumnya yang penulis angkat yaitu :

Tabel 1. Keaslian Penelitian

N O	Tahap Periode Jurnal	Penulis	Judul	Kesimpulan Hasil
1.	Jurnal Mantik Penusa Vol 15 No 1 Juni 2014 ISSN 2088- 3943	Hengki Tomando Sihotang	SISTEM PAKAR MENDIAGNOS A PENYAKIT KOLESTEROL PADA REMAJA DENGAN METODE CERTAINTY FACTOR (CF)	Membantu pekerjaan medis dalam pencegahan penyakit kolestrol pada remaja dengan cara melakukan diagnosa awal pada gejala-gejala nya.

			BERBASIS WEB	
2.	Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK) Volume (2) No. 1 Juli 2017 ISSN: 2527- 5771/EISSN: 2549-7839	Nelly Astuti Hasibuan	SISTEM PAKAR MENDIAGNOS A PENYAKIT KAKI GAJAH MENGGUNAK AN METODE CERTAINTY FACTOR	Mendeteksi penyakit kaki gajah melalui mikroskopis darah yang ditularkan cacing filarial yang berasal dari jenis nyamuk tertentu.
3	Prosiding Seminar Ilmu Komputer dan Teknologi Informasi Vol. 2, No. 1, Maret 2017 e-ISSN 2540-7902 dan	Fersalina Indah Mevung, dkk.	DIAGNOSIS PENYAKIT KEJIWAAN MENGUNA KAN METODE CERTAINTY FACTOR	Dengan terciptanya Sistem pakar ini membantu masyarakat dalam penanganan 4 penyakit gangguan jiwa secara cepat dan mudah baik dari segi waktu maupun biaya yang

	p-ISSN 2541-366X			mungkin akan lebih hemat.
4	Jurnal TECHNO Nusa Mandiri Vol. XIII, No. 2 September 2016 P-ISSN: 1978-2136 E-ISSN: 2527-676X	Laila Septiana	PERANCANGAN SISTEM PAKAR DIAGNOSA PENYAKIT ISPA DENGAN METODE CERTAINTY FACTOR BERBASIS ANDROID	Sistem pakar ini sangat membantu masyarakat untuk mengetahui jenis gangguan pernapasan yang dialami oleh si penderita dengan mudah dan mengerti untuk penanganannya.
5	Jurnal Sains, Teknologi dan Industri, Vol. 11, No. 2, Juni 2014, pp. 305 – 313 ISSN	Fitri Wulandari, Ihsan Yuliandri	DIAGNOSA GANGGUAN GIZI MENGGUNAKAN METODE	Aplikasi ini sangat membantu masyarakat dalam mengenali penyakit akibat gangguan gizi buruk berdasarkan gejala

	1693-2390 print/ISSN 2407-0939 online		CERTAINTY FACTOR	yang dialami oleh sipenderita serta merekomendasikan asupan gizi yang tepat.
6	Jurnal TIMES, Vol. IV No 2 : 35-39 , 2015 ISSN : 2337 – 3601	Swono Sibagariang	SISTEM PAKAR DIAGNOSA PENYAKIT SAPI DENGAN METODE CERTAINTY FACTOR BERBASIS ANDROID	Mendeteksi penyakit sapi yang menular tanpa bersentuhan dengan dokter hewan.

7.	<p>Jurnal Teknik A Vol 6 No 2 September 2014 ISSN No. 2085- 0859</p>	<p>Siti Mujilahwati</p>	<p>DIAGNOSA PENYAKIT TANAMAN HIAS MENGUNAKAN METODE CERTAINTY FACTOR BERBASIS WEB</p>	<p>Membantu pembudi daya tanaman hias mengatasi kesulitan dalam mendiagnosa penyakit tanaman dikarenakan penyakit tanaman yang sangat bervariasi agar tidak terjadi kesalahan dalam memberikan solusinya.</p>
8.	<p>JIMP -Jurnal Informatika Merdeka Pasuruan Vol. 2, No.1 Agustus 2016 ISSN :2503- 1945</p>	<p>Mohamad Hadi dkk</p>	<p>PERANCANG AN SISTEM PAKAR DIAGNOSA PENYAKIT AYAM DENGAN METODE FORWARD</p>	<p>Aplikasi ini menghasilkan keluaran berupa kemungkinan penyakit ginjal yang diderita berdasarkan gejala yang dirasakan oleh masyarakat.</p>

			CHAINING	
--	--	--	----------	--

Pada penelitian ini penulis ingin merancang sebuah aplikasi Sistem Pakar mendiagnosa gangguan psikosomatis pada kaum muda. Dimana aplikasi ini dapat membantu setiap orang yang mengalami gangguan psikosomatis khususnya pada kaum muda yang lebih dominan.

II.2. Pengertian Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut (Suwono Sibagariang : 2015 : hal 1).

Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para

ahli di bidangnya(Siti Mujilahwati : 2014 : hal 1).

II.2.1. Konsep Dasar Sistem Pakar

- 1) Pakar adalah orang yang memiliki pengetahuan, khusus, pendapat pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.
- 2) *Knowledge engineer* (Perekayasa Sistem) adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan *counter example* dan menerangkan kesulitan-kesulitan konseptual.
- 3) Pemakai, sistem pakar memiliki beberapa pemakai, yaitu : pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambahkan basis pengetahuan, dan pakar(Sumber : Laila : 2016 : hal 2).

II.2.2. Ciri-ciri Sistem Pakar

Beberapa ciri-ciri sistem pakar adalah :

- 1) Memiliki fasilitas informasi yang handal.
- 2) Mudah dimodifikasi.
- 3) Dapat digunakan dalam berbagai jenis komputer.
- 4) Memiliki kemampuan untuk belajar beradaptasi(Sumber : Laila : 2016 : hal 2)

II.2.3. Tujuan Sistem Pakar

Tujuan dari sebuah Sistem Pakar adalah mentransfer kepakaran yang dimiliki seorang pakar kedalam komputer dan kemudian kepada orang lain (*non expert*).

Aktivitas pemindahan kepakaran adalah :

- a) *Knowledge Acquisition* (dari pakar atau sumber lain)
- b) *Knowledge Representation* (ke dalam komputer)
- c) *Knowledge Inferencing*
- d) *Knowledge Transferring* (Sumber : Laila : 2016 : hal 2).

II.2.4. Keuntungan Sistem Pakar

Beberapa keuntungan sistem pakar antara lain :

- 1) Membuat seorang yang awam dapat bekerja seperti para ahli.
- 2) Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
- 3) Meningkatkan output dan produktivitas.
- 4) Meningkatkan kualitas.
- 5) Mampu mengambil dan melestarikan keahlian para pakar terutama yang termasuk dalam keahlian langka.
- 6) Mampu beroperasi dalam lingkungan yang berbahaya.

- 7) Handal (Reliability).
- 8) Sebagai media pelengkap dalam pelatihan.
- 9) Meningkatkan kapabilitas dalam penyelesaian masalah.
- 10) Menghemat waktu dalam pengambilan keputusan
- 11) Meningkatkan kapabilitas sistem komputer
- 12) Bisa Melakukan proses secara berulang secara otomatis (Sumber : Laila septiana : 2016 : hal 2).

II.3. Metode Certainty Factor

Faktor kepastian (certainty factor) diusulkan oleh Shortliffe dan Buchanan pada 1975 untuk mengakomodasikan ketidakpastian pemikiran (inexact reasoning) seorang pakar. Seorang pakar, (misalnya dokter) sering kali menganalisis informasi yang ada dengan ungkapan seperti misalnya : mungkin, kemungkinan, besar, hampir pasti. Untuk mengakomodasi hal ini dengan menggunakan certainty factor (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang dihadapi. Secara umum, rule direpresentasikan dalam bentuk sebagai berikut :

<pre>IF E1 [AND / OR] E2 [AND / OR] . . . En THEN H (CF=CFi)</pre> (1)
--	----------

Dimana :

$E_1 \dots E_n$: fakta-fakta (evidence) yang ada

H : hipotesa atau konklusi yang dihasilkan

CF : tingkat keyakinan (certainty factor) terjadinya hipotesa akibat adanya fakta-fakta.

Metode certainty factor ini hanya bisa mengolah 2 bobot dalam sekali perhitungan. Untuk bobot yang lebih dari 2 banyaknya, untuk melakukan perhitungan tidak terjadi masalah apabila bobot yang dihitung teracak, artinya tidak ada aturan untuk mengkombinasikan bobotnya, karena untuk kombinasi seperti apapun hasilnya akan tetap sama. Misalnya, untuk mengetahui apakah seorang pasien tersebut menderita penyakit jantung atau tidak, dilihat dari hasil perhitungan bobot setelah semua keluhan-keluhan diinputkan dan semua bobot dihitung dengan menggunakan metode certainty factor. Pasien yang divonis mengidap penyakit jantung adalah pasien yang memiliki bobot mendekati +1 dengan keluhan-keluhan yang dimiliki mengarah kepada penyakit jantung. Sedangkan pasien yang mempunyai bobot mendekati -1 adalah pasien yang dianggap tidak mengidap penyakit jantung, serta pasien yang memiliki bobot sama dengan 0 diagnosisnya tidak diketahui atau unknown atau bisa disebut dengan netral (Sumber : Nelly Astuti Hasibuan dkk : 2017 : hal 3-5).

II.3.1. Model Perhitungan Certainty Factor

Dalam mengekspresikan derajat kepastian, *certainty factor* untuk mengansumsikan derajat kepastian seorang pakar terhadap suatu data. Konsep ini kemudian diformulasikan dalam rumusan dasar sebagai berikut :

$$CF[H,E] = MB[H,E] - MD[H,E]$$

$$CF[H,E] = CF[H] * CF[E]$$

$$Cfcombine CF[H,E]_{1,2} + CF[H,E]_2 * [1 - CF[H,E]_1]$$

$$Cfcombine CF[H,E]_{old,3} = CF[H,E]_{old} + CF[H,E]_3 * [1 - CF[H,E]_{old}]$$

Contoh kasus :

Kaidah produksi atau aturan *rule* yang berkaitan dengan kaki gajah adalah sebagai berikut :

Kaidah

IF Demam berulang-ulang selama 3-5 hari

AND Pembengkakan kelenjar getah bening

AND Panas dan sakit radang saluran kelenjar getah bening

AND Filarial abses

AND Pembesaran tungkai

AND Pembesaran lengan

AND Pembesaran buah dada

AND Pembesaran buah zakar

THEN Filariasis atau Kaki gajah

Langkah pertama, pakar menentukan CF untuk masing-masing gejala sebagai berikut:

CFpakar (Demam berulang-ulang selama 3-5 hari)	= 0.4
CFpakar (Pembengkakan kelenjar getah bening)	= 0.6
CFpakar (Panas dan sakit radang saluran kelenjar getah bening)	= 0.6
CFpakar (Filarial abses)	= 0.6
CFpakar (Pembesaran tungkai)	= 0.8
CFpakar (Pembesaran lengan)	= 0.8
CFpakar (Pembesaran buah dada)	= 0.8
CFpakar (Pembesaran buah zakar)	= 0.8

Misalkan *user* memilih jawaban sebagai berikut :

Demam berulang-ulang selama 3-5 hari	= 0.2
Pembengkakan kelenjar getah bening	= 0.4
Panas dan sakit radang saluran kelenjar getah bening	= 0.4
Filarial abses	= 0.4
Pembesaran tungkai	= 0.6
Pembesaran lengan	= 0.6
Pembesaran buah dada	= 0.6
Pembesaran buah zakar	= 0.6

Kaidah tersebut kemudian dihitung nilai Cfpakarnya dengan mengalikan C_{fuser} dengan CF menjadi : $CF_{[H,E]1} = CF_{[H]1} * CF_{[E]1}$

$$=0.4*0.2$$

$$=0.08$$

$$CF[H,E]2 = CF[H]2 * CF[E]2$$

$$=0.6*0.4$$

$$=0.24$$

$$CF[H,E]3 = CF[H]3 * CF[E]3$$

$$=0.6*0.4$$

$$=0.24$$

$$CF[H,E]4 = CF[H]4 * CF[E]4$$

$$=0.6*0.4$$

$$=0.24$$

$$CF[H,E]5 = CF[H]5 * CF[E]5$$

$$=0.8*0.6$$

$$=0.48$$

$$CF[H,E]6 = CF[H]6 * CF[E]6$$

$$=0.8*0.6$$

$$=0.48$$

$$CF[H,E]7 = CF[H]7 * CF[E]7$$

$$=0.8*0.6$$

$$=0.48$$

$$CF[H,E]8 = CF[H]8 * CF[E]8$$

$$=0.8*0.6$$

$$=0.48$$

Langkah yang terakhir adalah mengkombinasikan nilai CF dari kaidah. Berikut adalah kombinasi CF[H,E] dengan CF[H,E] :

CFcombine CF[H,E]1,2

$$= CF[H,E]1 + CF[H,E]2 * (1 - CF[H,E]1)$$

$$= 0.08 + 0.24 (1-0.08)$$

$$= 0.08 + 0.24 (0.92)$$

$$= 0.08 + 0.22$$

$$=0.3_{old1}$$

CFcombine CF[H,E] old1

$$= CF[H,E] old1 + CF[H,E]3 * (1 - CF[H,E] old1)$$

$$= 0.3 + 0.24 (1-0.3)$$

$$=0.3 + 0.24 (0.7)$$

$$=0.3 + 0.16$$

$$=0.46_{old2}$$

CFcombine CF[H,E]old2,4

$$= CF[H,E] old2 + CF[H,E]4 * (1 - CF[H,E] old2)$$

$$= 0.46 + 0.24 (1-0.46)$$

$$= 0.46 + 0.24 (0.54)$$

$$=0.46 + 0.12$$

$$= 0.58_{old3}$$

CFcombine CF[H,E]old3,5

$$= \text{CF[H,E] old3} + \text{CF[H,E]5} * (1 - \text{CF[H,E] old3})$$

$$= 0.58 + 0.48 (1-0.58)$$

$$= 0.58 + 0.48 (0.42)$$

$$= 0.58 + 0.20$$

$$= 0.78 \text{ old4}$$

CFcombine CF[H,E]old4,6

$$= \text{CF[H,E] old4} + \text{CF[H,E]6} * (1 - \text{CF[H,E] old4})$$

$$= 0.78 + 0.48 (1-0.78)$$

$$= 0.78 + 0.48 (0.22)$$

$$= 0.78 + 0.10$$

$$= 0.88 \text{ old5}$$

CFcombine CF[H,E]old5,7

$$= \text{CF[H,E] old5} + \text{CF[H,E]7} * (1 - \text{CF[H,E] old5})$$

$$= 0.88 + 0.48 (1-0.88)$$

$$= 0.88 + 0.48 (0.12)$$

$$= 0.88 + 0.05$$

$$= 0.93 \text{ old6}$$

CFcombine CF[H,E]old6,8

$$= \text{CF[H,E] old6} + \text{CF[H,E]8} * (1 - \text{CF[H,E] old6})$$

$$= 0.93 + 0.48 (1-0.93)$$

$$= 0.93 + 0.48 (0.07)$$

$$= 0.93 + 0.03$$

$$= 0.96$$

$$CF[H,E] \times 100 = 0.96 \times 100 = 96\%$$

Dengan demikian dapat dikatakan bahwa perhitungan *Certainty factor* kaki gajah memiliki persentase tingkat keyakinan 96% (Sumber : Nelly Astuti Hasibuan : 2017 : hal 5-7).

Table 1 Nilai *Certainty Factor*

Kondisi Tidak Pasti (<i>Uncertain Term</i>)	CF
Pasti Tidak (<i>Definitely not</i>)	-1.0
Hampir Pasti Tidak (<i>Almost Certainly Not</i>)	-0.8
Kemungkinan Besar (<i>Probably Not</i>)	-0.6
Kemungkinan Tidak (<i>Maybe Not</i>)	-0.4
Tidak Tahu (<i>Unknown</i>)	-0.2 to 0.2
Kemungkinan (<i>Maybe</i>)	0.4 0.6 0.8
Kemungkinan Besar (<i>Probably</i>)	1.0
Hampir Pasti (<i>Almost Certainly</i>)	
Pasti (<i>Definitely</i>)	

(Sumber : Fersalina Indah Mevung:2017:hal 2)

II.4. PHP



Gambar 1. Logo PHP

(Sumber : Dewi Maharani : 2017 : hal 3)

PHP: *Perl Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan kedalam HTML. PHP adalah bahasa pemrograman server side yang sudah banyak digunakan pada saat ini, terutama untuk pembuatan website dinamis. Untuk hal-hal tertentu dalam pembuatan web, bahasa pemrograman PHP memang diperlukan, misalnya saja untuk memproses data yang dikirimkan oleh pengunjung web.

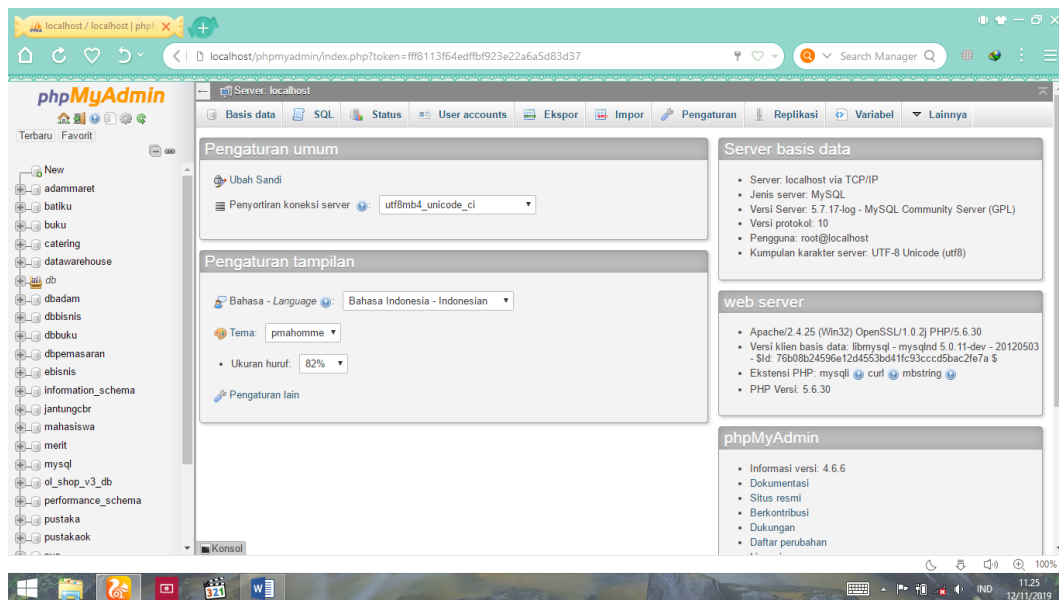
II.4.1 Kelebihan PHP

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.

2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintahperintah sistem (Dewi Maharani : 2017 : hal 2).

II.5. MySQL



Gambar 2 Tampilan MySQL

(Sumber : Dewi Maharani : 2017 : hal 3)

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakannya, tapi tidak boleh dijadikan produk turunan yang bersifat Closed Source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structur Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Sebagai database server, MySQL dapat dikatakan lebih unggul dibanding database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lipat lebih cepat dari PostgreSQL dan Lima kali lebih cepat dibanding Interbase.

II.5.1 Keistimewaan MySQL

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL.

1. Portability MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, dan lain-lain.

2. Open Source MySQL didistribusikan secara Open Source, sehingga dapat digunakan secara bebas.
3. Multi-user MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan.
4. Performance Tuning MySQL memiliki kecepatan yang tinggi dalam menangani query, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu (Dewi Maharani : 2017 : hal 3).

II.6. UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek. *Unified Modeling Language (UML)* bukanlah merupakan bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman, sehingga memungkinkan melakukan pemetaan (mapping) langsung dari model-model yang dibuat dengan *Unified Modeling Language (UML)* dengan bahasa-bahasa pemrograman berorientasi obyek, seperti *Java*.

UML tersusun atas sejumlah elemen grafis membentuk 9 diagram-diagram. Dalam penelitian ini melakukan desain hanya 4 diagram yaitu *Use Case Diagram*,

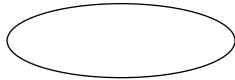
Activity Diagram, Class Diagram dan Sequence Diagram (Sumber : Yuli Syafitri : 2016 : hal 2).

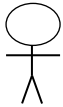




II.6.1. Use Case Diagram

Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem ini berinteraksi dengan dunia luar, misalnya menyusun sebuah daftar layanan kesehatan. Use case diagram dapat digunakan untuk memperoleh kebutuhan sistem dan memahami bagaimana sistem seharusnya bekerja. Komponen yang terdapat pada sebuah use case diagram terdiri dari [5]:

- a. Actor : pengguna perangkat lunak aplikasi, bisa berupa manusia, perangkat keras atau sistem informasi yang lain. Actor dapat memasukan informasi ke dalam sistem, menerima informasi dari sistem, atau keduanya.
- b. Use case : perilaku atau apa yang dikerjakan pengguna sistem aplikasi, termasuk interaksi antar actor dengan perangkat lunak aplikasi tersebut.

Table 2 **Simbol Use case**

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>Use case</i> .

	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use case</i>, tetapi tidak memiliki control terhadap <i>Use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>Use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>Use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</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>
	<p><i>Extend</i>, merupakan perluasan dari <i>Use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Yuli Syafitri : 2016 : hal 2)

II.6.2 Class Diagram

Class diagram adalah *visualisasi* kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Diagram ini memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain (dalam logical

view) dari suatu sistem. Kelas memiliki 3 area utama yaitu : nama, atribut, dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut fungsinya adalah untuk menunjukkan karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek (Sumber : Yuli Syafitri : 2016 : hal 3).

Table 3 Multiplicity Class Diagram

Multiplicity	Penjelasan
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




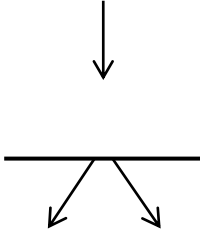
(Sumber : Yuli Syafitri : 2016 : hal 3)

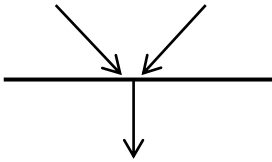
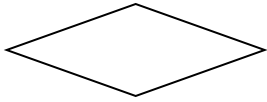

II.6.3. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana suatu aktivitas berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa kegiatan. Sebuah

aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas (Sumber : Yuli Syafitri : 2016 : hal 3).

Table 4 Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.

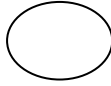
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true, false</i> .
	<i>Swimlane</i> , pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

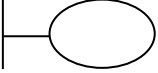

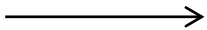
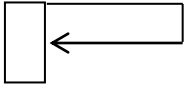


(Sumber : Yuli Syafitri : 2016 : hal 3)

II.6.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek berupa pesan (message) yang digambarkan terhadap waktu. Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses (Sumber : Yuli Syafitri : 2016 : hal 3).

Table 5 Simbol Sequence Diagram

Gambar	Keterangan
	<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.
	<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.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Yuli Syafitri : 2016 : hal 3)

II.7. Website



Gambar 3 Website Sistem Pakar

(Sumber : Indra Griha Tofik Isa : 2017 : hal 4)

Website adalah kumpulan kumpulan halaman web yang di dalamnya terdapat sebuah domain mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan. Jadi bisa dikatakan bahwa pengertian website adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan melalui jaringan-jaringan halaman. Hubungan antara satu halaman website dengan halaman website lainnya disebut hyperlink, sedangkan teks yang dijadikan media penghubung disebut hypertext (Sumber : Indra Griha Tofik Isa : 2017 : hal 4).

II.8. Web Mobile



Gambar 4. Web Mobile

(Sumber : Joni Karman : 2018 : hal 3)

Web mobile merupakan web atau halaman website internet yang dapat digunakan atau diakses pada perangkat mobile. Agar website yang telah anda buat dapat diakses pada berbagai perangkat, baik perangkat komputer secara umum (seperti PC/laptop) atau perangkat mobile (smartphone) maka diperlukan teknik tersendiri. Anda perlu membuat dua versi website sehingga ketika pengguna mengakses halaman pertama atau halaman index maka diberikan skrip yang akan memeriksa perangkat apa yang mengakses. Skrip yang digunakan untuk mendeteksi tersebut bisa menggunakan bahasa PHP atau JavaScript. Ketika dideteksi yang mengakses adalah perangkat mobile maka akan diarahkan ke halaman mobile,

sedangkan jika yang mengakses sebuah desktop PC/laptop maka akan diarahkan ke halaman website pada umumnya.

II.9. Psikosomatis

II.9.1 Ragam Koping Pada Remaja Saat Mengalami Psikosomatis

Perkembangan yang dialami remaja dengan berbagai dinamikanya, menuntut remaja mampu beradaptasi lebih cepat, namun bagi beberapa orang perubahan ini memberikan pengaruh yang tidak mudah dilalui. Perubahan ini pun bahkan mengakibatkan kondisi fisik dengan psikosomatis. Psikosomatis merupakan penyakit fisik yang gejalanya disebabkan oleh proses mental seseorang. Jika dalam pemeriksaan medis tidak ditemukan penyebab fisik atas gejala-gejala yang muncul, atau gejala fisik ini sering muncul akibat dari kondisi emosional, seperti kemarahan, depresi, dan rasa bersalah.

Ketika remaja mengalami keluhan gejala fisik namun gejala tersebut dapat menghilang sendiri atau paling tidak berkurang rasanya dengan tanpa ditangani secara medis dengan melakukan suatu hal. Kegiatan hal yang sama kemudian dilakukan remaja secara terus menerus demi proses mengenal dirinya. Berbagai cara menghilangkan keluhan pun dilakukan, proses pengkondisian diri yang terus saja berubah. Aktivitas pengkondisian