

## BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

### 2.1. Tinjauan Pustaka

Penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan telah disajikan dalam **Tabel 2.1** Tabel Literatur.

**Tabel 2.1** Tabel Literatur

No	Judul	Tahun	Penulis
1.	Sistem Pakar Diagnosa Penyakit Jantung Menggunakan Metode <i>Certainty Factor</i> Berbasis Web	2012	Jadiaman Parhusip dkk
2.	Sistem Pakar Diagnosa Penyakit Jantung dan Paru dengan <i>Fuzzy Logic</i> dan <i>Certainty Factor</i>	2014	Desak Putu Siska Dewi
3.	Perancangan Sistem Pakar Diagnosa Penyakit Jantung Dengan Metode <i>Forward Chaining</i>	2018	Fajar Agung Nugroho
4.	Sistem Pakar Diagnosa Penyakit Jantung Koroner Dengan Metode <i>Probabilistic Fuzzy Decision Tree</i>	2018	Wizra Aulia
5.	<i>Expert System to Diagnosing Heart Attack Disease Using Certainty Factor Method</i>	2019	Muhammad Iqbal dkk

### **2.1.1 Jadiaman Parhusip dkk (2012)**

Angka kematian para penderita penyakit jantung yang semakin meningkat, dikarenakan kurangnya pengetahuan tentang gejala awal penyakit jantung dan fasilitas kesehatan khususnya jantung di Indonesia masih sangat terbatas (Parhunsip, 2012). Sehingga dalam bidang kesehatan juga membutuhkan teknologi komputer. Salah satunya adalah digunakan untuk mendiagnosa penyakit jantung. Aplikasi sistem pakar untuk diagnosa penyakit jantung ini adalah suatu sistem yang terkomputerisasi untuk membantu dokter dan masyarakat dalam mendiagnosa penyakit jantung.

Pada penelitian ini akan dibuat sistem pakar menggunakan metode *certainty factor* (faktor kepastian) untuk mendiagnosa penyakit jantung pada manusia. Sistem ini dapat memberikan diagnosa awal penyakit jantung yang diderita oleh penderita, dari gejala-gejala yang dirasakan oleh penderita tanpa harus bertanya langsung ke pakar (Dokter). Rancangan aplikasi sistem pakar untuk mendiagnosa penyakit Jantung dibuat dengan aplikasi berbasis web, sehingga bisa diakses masyarakat secara luas, selain itu aplikasi ini dapat juga membantu paramedis untuk melakukan pengambilan keputusan dalam mendiagnosa penyakit jantung.

### **2.1.2. Desak Putu Siska Dewi (2014)**

Sistem pakar memungkinkan proses komunikasi antara dokter dan pasien dapat dilakukan tanpa harus melakukan tatap muka secara langsung. Pengguna akan diarahkan oleh sistem untuk menyampaikan keluhannya secara detail dan bertahap. Sistem pakar mampu memberikan penjelasan mengenai keterkaitan antara gejala-gejala yang disampaikan oleh pengguna. Bahkan, sistem pakar

mampu memberikan solusi terhadap permasalahan yang mengandung unsur ketidakpastian seperti adanya kemiripan gejala antara satu penyakit dengan penyakit lainnya.

Organ jantung dan paru adalah organ tubuh yang sangat vital bagi manusia. Gangguan pada jantung dan paru dapat menyebabkan dampak yang sangat besar bagi kesehatan manusia. Sistem pakar untuk mendiagnosa penyakit paru-paru dan jantung pada manusia dibuat karena banyak penyakit paru dan jantung lainnya diderita oleh masyarakat Indonesia. Metode yang digunakan dalam pembuatan sistem pakar ini adalah hasil pengkombinasian 2 metode, yaitu metode *certainty factor* (CF) dan *fuzzy logic*. Uji coba pada penelitian ini dilakukan terhadap pasien penyakit jantung dan paru-paru. Hasil akhir dari penyakit yang diderita kemudian didiskusikan kepada dokter spesialis penyakit dalam yang menghasilkan tingkat kemiripan sistem pakar yang telah dibuat dengan pakar sesungguhnya sebesar 94.61 % (Dewi, 2014).

### **2.1.3. Fajar Agung Nugroho (2018)**

Penyakit jantung merupakan salah satu penyebab kematian yang tinggi bersama stroke, kanker paruparu, kanker payudara, dan AIDS. Kebanyakan masyarakat awam sangat kurang memperhatikan kesehatan, terutama kesehatan jantung. Mereka enggan memeriksakan kesehatan jantungnya karena kurangnya pelayanan terhadap pasien, kurangnya tenaga medis khususnya dokter spesialis jantung serta jam kerja dokter yang terbatas (Nugroho, 2018). Berdasarkan latar belakang permasalahan di atas, maka diperlukan sebuah aplikasi komputer atau

program yang dapat menyimpan pengetahuan seorang pakar untuk melakukan diagnosa penyakit jantung dan memberikan hasil yang konsisten, cepat dan tepat.

Dalam penelitian ini dibuat sistem pakar yang mampu melakukan diagnosa serta memberikan solusi yang tepat dan cepat terhadap gejala-gejala penyakit jantung. Dengan sistem ini, diharapkan orang awam dapat mengetahui lebih dini gejala-gejala awal penyakit jantung sehingga dapat melakukan tindakan yang tepat untuk menghadapi penyakit tersebut tanpa bantuan dokter spesialis penyakit jantung. Penelitian ini akan merancang sistem pakar diagnosa penyakit jantung dengan menggunakan metode *forward chaining*. Sistem pakar yang akan dikembangkan menggunakan basis pengetahuan sehingga akan memberikan kemudahan apabila suatu saat terjadi penambahan aturan atau pengetahuan baru seiring dengan meningkatnya pengetahuan medis.

#### **2.1.4. Wizra Aulia (2018)**

Jantung merupakan salah satu organ yang memiliki peranan sangat penting bagi manusia. Jantunglah yang bertanggung jawab memompa darah dan mengalirkannya keseluruh tubuh. Penyakit jantung dan pembuluh darah merupakan penyebab kematian tertinggi di dunia maupun di Indonesia. Data dari *World Health Organization* , WHO (2007) menunjukkan bahwa sekitar 1.5 juta orang pertahun didunia meninggal karena penyakit jantung dan pembuluh darah. Penyakit jantung koroner merupakan salah satu penyakit penyebab kematian di Indonesia. Jika gejala penyakit jantung koroner dikenali sejak dini, tindakan antisipatif dapat dilakukan. Diagnosis adalah berdasarkan 6 gejala penyakit jantung koroner yaitu nyeri dada, tekanan darah tinggi, kolesterol, kadar gula

darah, hasil EKG dan jumlah denyut jantung. Metode yang digunakan adalah *Probabilistic Fuzzy Decision Tree* (PFDT) dengan algoritma *Probabilistic Fuzzy ID3*. Hasil akurasi sistem pakar diagnosis penyakit jantung koroner dengan metode PFDT mencapai 95% (Aulia, 2018).

#### **2.1.5. Muhammad Iqbal dkk (2019)**

Penyakit Jantung Koroner (PJK) merupakan salah satu penyakit yang menyebabkan kematian dan masih menjadi masalah kesehatan untuk negara maju dan berkembang. Diperkirakan karena pertumbuhannya yang cepat, hal itu menjadikannya penyebab kematian terbesar di tahun 2020. Penyebab penyakit jantung koroner tidak memandang umur dan jenis kelamin, dengan kejadian pada laki-laki lebih banyak dibandingkan pada wanita tetapi pada kejadian pada wanita akan meningkat setelah menopause (akhir dari siklus haid) sekitar usia 50 tahun. Hal ini disebabkan estrogen (senyawa steroid) memiliki efek perlindungan pada timbulnya aterosklerosis.

Dalam penelitian ini penulis ingin membuat suatu sistem yang dapat mendeteksi penyakit jantung oleh pengguna sebelum memastikan dirinya ke dokter untuk diperiksa. Sistem yang akan penulis rancang menggunakan metode *certainty factor* yang disediakan penilaian gejala diagnosis pengguna. Aplikasi ini dirancang dengan bantuan perangkat lunak Visual Studio dan MySQL untuk *database*. Hasil pengujian dalam penelitian ini menunjukkan nilai *certainty factor* dari masukan gejala yang mengarah kepada penyakit jantung koroner diperoleh sebesar 91,679058% (Iqbal et al., 2019).

## 2.2. Sistem Pakar

Salah satu bagian dari Artificial Intelligence (AI) yang akhir-akhir ini mengalami perkembangan pesat yaitu Sistem Pakar. Sistem pakar merupakan bagian dari kecerdasan buatan (artificial intelligence) yang merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaannya seperti dan sebaik yang dilakukan oleh manusia (Frans, 2015). Sistem pakar dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para pakar atau ahli. Pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli merupakan tujuan utama dari sistem pakar (Aulia, 2018). Dengan adanya sistem pakar ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli.

Sistem pakar (*expert system*) adalah paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti sains, rekayasa, matematika, kedokteran, pendidikan dan masih banyak lagi. Ada beberapa keunggulan dari sistem pakar, diantaranya:

1. Menghimpun data dalam jumlah yang sangat besar
2. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu
3. Mengerjakan perhitungan secara cepat dan tepat untuk mencari kembali data yang tersimpan dengan kecepatan tinggi.

Selain keuntungan-keuntungan tersebut, sistem pakar seperti halnya sistem lainnya juga memiliki kelemahan, diantaranya adalah :

1. Masalah dalam mendapatkan pengetahuan dimana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharannya.
3. Boleh jadi sistem tak dapat membuat keputusan.
4. Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan.

Kelemahan-kelemahan atau kekurangan dari sistem pakar tersebut bukanlah sama sekali tidak bisa diatasi, tetapi dengan terus melakukan perbaikan dan pengolahan berdasarkan pengalaman yang telah ada maka hal itu diyakini akan dapat diatasi, walaupun dalam waktu yang panjang dan terus-menerus.

### **2.3. *Certainty Factor***

*Certainty factor* merupakan metode yang mendefinisikan ukuran kepastian terhadap fakta atau aturan untuk menggambarkan keyakinan seorang pakar terhadap masalah yang sedang dihadapi (Parhusip, 2012). *Certainty Factor* merupakan bagian dari *certainty theory*, yang pertama kali dikenalkan oleh E. H. Shorliffe dan B. G. Buchanan dalam pembuatan MYCIN (merupakan aplikasi sistem pakar awal yang dirancang untuk mengidentifikasi infeksi didalam darah) mencatat bahwa pakar sering kali menganalisis informasi yang ada dengan

ungkapan seperti, misalnya: mungkin, kemungkinan besar, dan hampir pasti. Hal ini membuat tim MYCIN menggunakan *Certainty Factor* guna menggambarkan tingkat kepercayaan pakar terhadap masalah yang dihadapi.

MYCIN merupakan sistem pakar yang berbasis aturan yang mendiagnosis. MYCIN dapat mengenali sekitar 100 penyebab infeksi bakteri. Dengan demikian MYCIN dapat merekomendasikan resep obat yang efektif. Dalam tes terkontrol performanya dianggap dengan spesialis manusia. Metode pemrosesan dan tidak kepastian ini merupakan perintis dan menghasilkan pengaruh jangka panjang dalam pengembangan sistem pakar.

*Certainty Factor* menurut Giarrantano dan Riley dalam jurnal Parhusip (2012) didefinisikan sebagai berikut :

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

$$CF_{\text{gejala}} = CF_{\text{user}} * CF_{\text{pakar}}$$

Keterangan

CF = *Certainty Factor* (faktor kepastian) dalam hipotesis H yang dipengaruhi oleh fakta E.

MB= *Measure of Belief* (tingkat keyakinan), adalah ukuran kenaikan dari kepercayaan hipotesis H dipengaruhi oleh fakta E.

MD = *Measure of Disbelief* (tingkat tidak keyakinan), adalah kenaikan dari ketidakpercayaan hipotesis H terhadap E.

E = *Evidence* (peristiwa atau fakta)

H = Hipotesis (dugaan)

$$CF_{\text{combine}} = CF_1 + CF_2 * (1 - CF_1)$$

Dimana  $CF_1$  dan  $CF_2$  memiliki hipotesis yang sama:

$CF_1$  = nilai *Certainty Factor evidence* 1 terhadap hipotesis

$CF_2$  = nilai *Certainty Factor evidence* 2 terhadap hipotesis

**Tabel 2.2** Aturan nilai-nilai kepercayaan (Iqbal et al., 2019)

<i>Uncertain Term</i>	CF
<i>Definitely not</i> (Pasti tidak)	-1,0
<i>Almost certainly not</i> (Hampir pasti tidak)	-0,8
<i>Probably not</i> (Kemungkinan besar tidak)	-0,6
<i>Maybe not</i> (Mungkin tidak)	-0,4
<i>Unknown</i> (Tidak tahu)	-0,2 sampai 0,2
<i>Maybe</i> (Mungkin)	0,4
<i>Probably</i> (Kemungkinan besar)	0,6
<i>Almost certainly</i> (Hampir pasti)	0,8
<i>Definitely</i> (Pasti)	1,0

Terdapat kelebihan pada metode *Certainty Factor* yaitu:

1. Metode ini cocok dipakai pada sistem pakar yang mengandung ketidakpastian.
2. Metode ini dapat menjaga keakuratan data karena sistem perhitungan metode ini hanya dapat mengolah dua data dalam satu kali perhitungan.

Adapun kekurangan metode *Certainty Factor* yaitu:

1. Pemodelan ketidakpastian yang menggunakan perhitungan metode *Certainty Factor* biasanya masih diperdebatkan.

2. Metode Certainty Factor hanya dapat mengolah dua data saja, apabila ada data yang lebih dari dua maka perlu beberapa kali pengolahan data.

#### **2.4. PHP (Hypertext Preprocessor)**

PHP adalah bahasa pemrograman *web server-side* yang bersifat *open source*, PHP juga merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* ( *server side HTML embedded script*) (Wibowo, 2015). PHP digunakan untuk membuat halaman website menjadi dinamis, dinamis berarti *content* dalam website atau isi dari website tersebut dapat berubah-ubah setiap saat. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari website. Selanjutnya Rasmus merilis *sourcecode* tersebut untuk umum dan menamakannya PHP/FI.

Tahun 1997 PHP merilis PHP versi 2.0, pada versi ini PHP sudah terintegrasi dengan bahasa pemrograman C dan juga sudah dilengkapi dengan modul sehingga membuat kualitas kerja PHP menjadi lebih meningkat secara signifikan. Ditahun yang sama sebuah perusahaan *software* bernama Zend merilis ulang PHP versi ini dengan lebih baik, bersih dan cepat. Seiring perkembangan zaman ditahun 1994 PHP versi 4.0 mulai dirilis dan versi ini yang paling banyak digunakan pada saat awal abad ke-21 karena PHP versi ini sudah mampu membangun web kompleks dengan stabilitas kecepatan yang cukup tinggi. Ditahun 2004 perusahaan *software* Zend merilis PHP lagi dengan versi terbarunya yaitu versi 5.0 yang inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek atau yang sering disebut OOP kedalam PHP

untuk menjawab perkembangan bahasa pemrograman kearah paradigma berorientasi objek.

Bahasa pemrograman PHP sering digunakan karena PHP adalah bahasa yang bersifat *open source* yang memiliki kesederhanaan dan memiliki beberapa fitur yang berfungsi untuk menangani kebutuhan standar dalam pembuatan sebuah website. PHP juga merupakan bahasa *script* yang paling mudah dipahami karena tersedia dokumentasi yang cukup lengkap tentang fungsi dan contohnya. PHP juga dapat digunakan untuk berbagai sistem operasi antara lain : Unix, Macintosh dan juga windows. PHP dapat dijalankan secara *runtime* melalui *console* juga dapat menjalankan perintah-perintah sistem. *Open source* disini memiliki arti kode-kode PHP terbuka untuk umum dan tidak berbayar atas pembelian dari lisensi. *Web server* yang mendukung PHP dapat ditemukan dimana-mana, mulai dari Apache, IIS, Lighttpd hingga Xitami dengan konfigurasi yang relatif mudah. Selain itu PHP juga dilengkapi dengan berbagai macam pendukung lain seperti *support* langsung ke berbagai macam *database* yang populer seperti Oracle, MySQL dan lain-lain.

## **2.5. MySQL (*My Structure Query Language*)**

MySQL adalah sebuah perangkat lunak pembuat *database* yang bersifat terbuka atau *open source* dan berjalan disemua *platform* baik Linux maupun sistem operasi Windows, MySQL merupakan program pengakses *database* yang bersifat *network* sehingga dapat digunakan untuk aplikasi *multi-user* / banyak pengguna (Wibowo, 2015). MySQL adalah sebuah DBMS (*Database Management System*) menggunakan perintah SQL (*Structured Query Language*)

yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL dibagi menjadi dua lisensi, pertama adalah Free Software dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *Shareware* dimana perangkat lunak berpeilik memiliki batasan dalam penggunaannya. MySQL termasuk ke dalam RDBMS (*Relational Database Management System*). Sehingga, menggunakan tabel, kolom, baris, di dalam struktur *database*-nya. Jadi, dalam proses pengambilan data menggunakan metode *relational database*. Dan juga menjadi penghubung antara perangkat lunak dan *database server*.

Seiring dengan berjalannya waktu dan banyak *user* yang berminat menggunakan database ini, MySQL mulai merilis versi yang dapat diinstal pada hampir semua platform termasuk sistem operasi windows. Lisensi dari MySQL adalah freeware, yang artinya kita dapat mendownload dan menggunakannya tanpa harus membayar. Umumnya orang menyebut database seperti MySQL, PostgreSQL atau yang lainnya dengan sebutan database server saja. Beberapa juga menyebutnya dengan *database engine*, mesin database, SMD, DBMS, *Back End* atau program *database* saja.

Terdapat beberapa kelebihan pada MySQL, yaitu :

1. MySQL dapat berjalan dengan stabil di berbagai system operasi.
2. MySQL bersifat *open source* yang didistribusikan secara gratis di bawah lisensi GNU *General Public License* (GPL).
3. MySQL bersifat *multiuser*, MySQL mampu digunakan oleh beberapa *user* dalam waktu bersamaan tanpa adanya masalah.

4. MySQL memiliki kecepatan yang bagus dalam menangani perintah SQL (query) yang memiliki arti MySQL mampu lebih banyak memproses perintah SQL per satuan waktu.
5. Dalam keamanan MySQL juga lebih unggul karena MySQL memiliki beberapa lapisan *security* seperti level subnet mask, nama host, dan izin akses *user*. Dengan sistem perizinan yang mendetail serta adanya password yang terenkripsi.

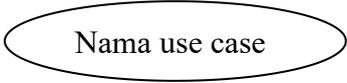
Adapun kekurangan MySQL yaitu :

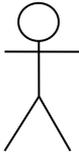
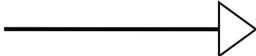
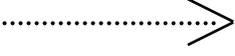
1. Sulit mengelola database dalam skala besar
2. Kurangnya *Technical Support*
3. Tidak cocok untuk game dan aplikasi *mobile*

## 2.6. Use Case Diagram

Menurut (Rossa & Shalahuddin, 2014), “*Use case* atau *use case diagram* adalah pemodelan untuk kelakuan (*behavior*) sistem yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi berupa simbol dan komponen pada *use case diagram*, dapat dilihat pada Tabel 2.3 berikut.

**Tabel 2.3** Notasi *Use Case Diagram* (Rossa & Shalahuddin, 2014)

Simbol	Deskripsi
	<p><i>Use case</i>, fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.</p>

	<b>Aktor / Actor</b> , orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	<b>Asosiasi / Association</b> , komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> / <i>use case</i> memiliki interaksi dengan aktor.
<<extend>> 	<b>Ekstensi/ Extend</b> , relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan.
	<b>Generalisasi / Generalization</b> , hubungan generaliasasi dan spesialisasi antara dua <i>use case</i> .
<<include>> 	<b>Menggunakan/ Include / uses</b> , <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

## 2.7. Pengujian

Metode pengujian Menurut Rosa & Shalahuddin (2018) : “Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi dan validasi. Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplemtasikan dengan benar sebuah fungsi spesifikasi“. Berikut ini terdapat beberapa pendekatan dalam pengujian diantaranya sebagai berikut:

## 1. Uji Akurasi

Pengujian akurasi sistem dilakukan untuk mengetahui seberapa akurat sistem yang dibuat. Pengujian dilakukan dengan membandingkan hasil diagnosis oleh pakar dengan hasil diagnosis oleh sistem dan kemudian akan dihitung persentase nilai keakuratan sistem. Dalam hal ini akan dilakukan pengujian terhadap beberapa kasus pengalaman pakar. Proses pengujian akan dilakukan dengan cara pakar melakukan diagnosis terlebih dahulu baru kemudian kasus diuji dengan sistem. Kemudian dilakukan perhitungan untuk menentukan akurasi sistem dengan menghitung rata-rata uji coba yang sesuai dengan hipotesa pakar kemudian dikalikan 100% agar dapat dilihat besarnya persentase akurasi sistem pakar.

## 2. *Black Box Testing* ( Pengujian Kotak Hitam )

*Black box testing* adalah *testing* yang dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang akan dites. *Black box testing* biasa disebut sebagai *behavioral testing*, *specification-based*, *input/output testing* atau *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui dengan menggunakan *black box testing* ini adalah:

- a. Fungsi yang hilang atau tidak benar.
- b. *Error* dari *interface*.
- c. *Error* dari struktur data atau akses *external database*.
- d. *Error* dari kinerja atau tingkah laku sistem.
- e. *Error* dari inisialisasi sistem.

Berikut terdapat beberapa kelebihan dalam penerapan pengujian dengan metode *black box*, yaitu :

1. Spesifikasi program dapat ditentukan di awal
2. Dapat digunakan untuk menilai konsistensi program
3. Uji kasus dilakukan secara cepat dan spesifik
4. Cocok dan efisien digunakan pada sistem atau segmen yang lebih besar
5. Tidak perlu melihat kode program secara detail

Adapun kekurangan dalam pengujian dengan metode *black box* adalah sebagai berikut :

1. Uji kasus sulit untuk dirancang jika persyaratan tidak jelas dan singkat
2. Tidak semua produk perangkat lunak dapat diuji
3. Uji *authoring* tidak efisien