# BAB II LANDASAN TEORI

## 2.1 Tinjauan Studi

Ada beberapa penelitian yang menggunakan *Naive Bayes* sebagai metode untuk yang digunakan untuk merancang system komputerisasi diantaranya :

1. Berdasarkan penelitian yang dilakukan Tita Tjayati, 2014 mengenai Analisis Perbandingan Metode Certainty Factor Dan Naïve Bayesian Dalam Mendeteksi Kemungkinan Anak Terkena Disleksia, hasil yang didapat dalam penelitian ini diperoleh tingkat akurasi rat-rata 58% untuk metode Certainty Factor dan rata-rata akurasi data untuk metode Naïve Bayesian adalah 93%. Hasil tersebut sudah menunjukkan metode mana yang lebih baik dalam mendeteksi kemungkinan anak terkena disleksia, yakni metode Naïve Bayesian karena selisih nya jauh lebih baik dibandingkan metode Certainty Factor ini disebabkan oleh parameter pada metode Certainty Factor yaitu nilai MB (Meansure of Believe) dan nilai MD (Meansure of Disbelieve) yang berikan langsung dari seorang pakar. Dan kedua metode tersebut dapat diterapkan dalam sistem pendeteksi disleksia sesuai cara dan perhitungan kedua metode tersebut.
2. Penelitian yang dilakukan oleh (Bustami : 2014) yang berjudul penerapan algoritma *Naive Bayes* untuk mengklasifikasi data nasabah asuransi. Penelitian ini membahas tentang penerapan metode *Naive Bayes* dalam mengelompokkan data nasabah untuk asuransi. Hasil dari penelitian ini menunjukkan bahwa akurasi hasil klasifikasi dengna *Naive Bayes* sangat akurat.
3. Penelitian yang dilakukan oleh (Bayu Hendra Setiawan dan Edy Mulyanto :2014) yang berjudul Sistem Pakar Pengenalan Gejala Dini Penyakit Epilepsy menggunakan Metode Naive Bayes, Penelitian ini berhasil membangun sebuah sistem pakar untuk diagnosa penyakit epilepsi beserta cara penanganannya. Penyakit epilepsi memiliki 24 jenis penyakit dengan jumlah gejala sebanyak 103 gejala. Metode inferensi yang digunakan adalah Naïve Bayes. Metode ini dipilih untuk mengatasi masalah ketidakpastian dalam proses pelacakan. Sistem pakar yang dibangun berbasis web agar memudahkan dalam distribusi sistem. Kinerja sistem diuji dengan membandingkan data rekam medis sebanyak 20 data dengan hasil keluaran sistem. Dari pengujian tersebut dihasilkan 85% data rekam medis hasil diagnosanya cocok dengan keluaran sistem, sehingga dapat disimpulkan sistem ini layak untuk digunakan.

## 2.2 Penyakit Kaki Gajah (Filariasis)

Penyakit Kaki Gajah (Filariasis atau Elephantiasis) adalah golongan penyakit menular yang disebabkan oleh cacing Filaria yang ditularkan melalui berbagai jenis nyamuk. Setelah tergigit nyamuk, parasit (larva) akan menjalar dan ketika sampai pada jaringan sistem lympa maka berkembanglah menjadi penyakit tersebut. Penyakit ini bersifat menahun (kronis) dan bila tidak mendapatkan pengobatan, dapat menimbulkan cacat menetap berupa pembesaran kaki, lengan dan alat kelamin baik perempuan maupun laki-laki. Penyakit Kaki Gajah bukanlah penyakit yang mematikan, namun demikian bagi penderita mungkin menjadi sesuatu yang dirasakan memalukan bahkan dapat mengganggu aktifitas sehari-hari.Penyakit Kaki Gajah umumnya banyak terdapat pada wilayah tropis. Menurut info dari WHO, urutan negara yang terdapat penderita mengalami penyakit kaki gajah adalah Asia Selatan (India dan Bangladesh), Afrika, Pasifik dan Amerika. Belakangan banyak pula terjadi di negara Thailand dan Indonesia (Asia Tenggara)(Sastaviyana Yuliangga,2009).

### Penularan Penyakit Kaki Gajah

Penyakit ini ditularkan melalui nyamuk yang menghisap darah seseorang yang telah tertular sebelumnya.Darah yang terinfeksi dan mengandung larva dan akan ditularkan ke orang lain pada saat nyamuk yang terinfeksi menggigit atau menghisap darah orang tersebut. Tidak seperti Malaria dan Demam berdarah,*Filariasis* dapat ditularkan oleh 23 spesies nyamuk dari *genus Anopheles*, *Culex*, *Mansonia*, *Aedes*&*amp*;*Armigeres*. Karena inilah, *Filariasis* dapat menular dengan sangat cepat(Sastaviyana Yuliangga, 2009).

### 2.2.2 Tanda dan Gejala Penyakit Kaki Gajah

Seseorang yang terinfeksi penyakit kaki gajah umumnya terjadi pada usia kanak-kanak, dimana dalam waktu yang cukup lama (bertahun-tahun) mulai dirasakan perkembangannya(Sastaviyana Yuliangga, 2009).

Adapun gejala akut yang dapat terjadi antara lain :

1. Demam berulang-ulang selama 3-5 hari, demam dapat hilang bila istirahat dan muncul lagi setelah bekerja berat.
2. Pembengkakan kelenjar getah bening (tanpa ada luka) didaerah lipatan paha, ketiak (*lymphadenitis*) yangtampak kemerahan, panas dan sakit.
3. Radang saluran kelenjar getah bening yang terasa panas dan sakit yang menjalar dari pangkal kaki ataupangkal lengan kearah ujung (*retrograde lymphangitis*)
4. *Filarial abses* akibat seringnya menderita pembengkakan kelenjar getah bening, dapat pecah dan mengeluarkannanah serta darah
5. Pembesaran tungkai, lengan, buah dada, buah zakar yang terlihat agak kemerahan dan terasa panas (*earlylymphodema*)

Sedangkan gejala kronisdari penyakit kaki gajah yaitu: berupa pembesaran yang menetap (*elephantiasis*) pada tungkai, lengan, buah dada, buah zakar (*elephantiasis skroti*).

### 2.2.3 Pencegahan Penyakit Kaki Gajah

Bagi penderita penyakit gajah diharapkan kesadarannya untuk memeriksakan kedokter dan mendapatkan penanganan obat-obatan sehingga tidak menyebarkan penularan kepada masyarakat lainnya. Untuk itulah perlu adanya pendidikan dan pengenalan penyakit kepada penderita dan warga sekitarnya.Pemberantasan nyamuk diwilayah masing-masing sangatlah penting untuk memutus mata rantai penularan penyakit ini. Menjaga kebersihan lingkungan merupakan hal terpenting untuk mencegah terjadinya perkembangan nyamuk diwilayah tersebut(Sastaviyana Yuliangga,2009).

## 2.4 Algoritma *Naive Bayes*

Algoritma *Naive Bayes* merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. *Naive Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris *Thomas Bayes*, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai *Teorema Bayes*. Teorema tersebut dikombinasikan dengan *Naive* dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi *Naive Baye/s* diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak adahubungannya dengan ciri dari kelas lainnya.

Persamaan dari teorema *Bayes* adalah :

Keterangan :

X : Data dengan class yang belum diketahui

H : Hipotesis data merupakan suatu class spesifik

P(H|X) : Probabilitas hipotesis berdasar kondisi (posterioriprobability)

P(H) : Probabilitas hipotesis (prior probability)

P(X|H) : Probabilitas berdasarkan kondisi pada hipotesis

P(X) : Probabilitas

Untuk menjelaskan teorema *Naive Bayes*, perlu diketahui bahwa prosesklasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apayang cocok bagi sampel yang dianalisis tersebut. Karena itu, teorema *bayes* di atas disesuaikan sebagai berikut :



Dimana Variabel *C* merepresentasikan kelas, sementara variabel *F1 ... Fn* merepresentasikan karakteristik petunjuk yang dibutuhkan untukmelakukan klasifikasi. Maka rumus tersebut menjelaskan bahwapeluang masuknya sampel karakteristik tertentu dalam kelas *C*(*Posterior*) adalah peluang munculnya kelas *C* (sebelum masuknyasampel tersebut, seringkali disebut *prior*), dikali dengan peluangkemunculan karakteristik karakteristik sampel pada kelas *C* (disebutjuga *likelihood*), dibagi dengan peluang kemunculan karakteristikkarakteristik sampel secara global ( disebut juga *evidence*). Karena itu,rumus diatas dapat pula ditulis secara sederhana sebagai berikut :



Nilai *Evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampelakan diklasifikasikan.

## 2.5 Siklus Pengembangan Sistem

Kebutuhan Sistem

Desain Sistem

Implementasi kurang lengkap/ ada permintaan baru

***Desain Sistem***

Perancangan Konseptual

Perancangan Fisik

***Operasi dan Pemeliharaan***

***Analisis Sistem***

Studi Kelayakan

Analisis Kebutuhan

***Implementasi Sistem***

Pemrograman dan pengujian konversi

Perubahan

Lingkup/Kebutuhan

Kesalahan atau masalah

yang tidak memungkinkan

implementasi dilaksanakan

Sistem siap

beroperasi mandiri

**Gambar 2. 1**Siklus Pengembangan System

### 2.5.1Analisis Sistem

Menurut Kusrini (2010), tahapan analisis system dimulai karena adanya permintaan terhadap system baru. Permintaan bisa datang dari seorang pimpinan/manager di luar departemen sistem informasi yang melihat adanya masalah atau menemukan adanya peluang baru. Namun, adakalanya inisiatif pengembangan sistem baru berasal dari bagian yang bertanggung jawab terhadap pengembangan sistem informasi. Tujuan utama dari analisis sistem adalah menentukan hal-hal secara detail yang akan dikerjakan oleh sistem yang diusulkan.

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem,adalah sebagai berikut :

1. *Identify*, mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Massalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuku dipecahkan. Tahap identifikasi masalah sangat penting karena akan menentukan keberhasilan pada langkah-langkah selanjutnya.
2. *Understand*, adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.
3. *Analyze*, menganalisis sistem tanpa *report.*
4. *Report*, yaitu membuat laporan hasil analisis. Tujuan utama dari pembuatan laporan hasil analisis yaitu pelaporan bahwa analisis telah selesai dilakukan.

### 2.5.2 Desain Sistem

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*systems design*). (Jugiyanto,2010)

Desain sistem dapat diartikan sebagai berikut ini:

1. Tahap setelah analisis dari siklus pengembangan sistem;
2. Pendefinisian dari kebutuhan-kebutuhan fungsional;
3. Persiapan untuk rancang bangun implementasi;
4. Menggambarkan bagaimana suatu sistem dibentuk;
5. Yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi;
6. Termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

Tahap desain sistem mempunyai dua tujuan utama, yaitu:

1. Untuk memenuhi kebutuhan kepada pemakai sistem
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya.

Desain sistem dapat dibagi dalam dua bagian yaitu desain sistem secara umum (*General systems design*) dan desain sistem secara terinci ( *detailed systems design*).

* + - 1. **Desain Sistem Secara Umum**

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada *user* tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analis sistem untuk mengidentifikasikan komponen-komponen sistem informasi yang akan didesain secara rinci oleh pemrogram komputer dan ahli teknik lainnya.

Pada tahap ini, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user*. Komponen sistem informasi yang didesain adalah model, output, input, database, teknologi dan kontrol.

* + - 1. **Desain Sistem secara Rinci (*Detailed System Design)***
* **Desain Input Terinci**

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil dari transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukkan. Desain input terinci dimulai dari desain dokumen dasar sebagai penangkap input yang pertama kali. Jika dokumen dasar tidak didesain dengan baik, kemungkinan input yang tercatat dapat salah bahkan kurang.

Fungsi dokumen dasar dalam penanganan arus data :

1. Dapat menunjukkan macam dari data yang harus dikumpulkan dan ditangkap
2. Data dapat dicatat dengan jelas, konsisten dan akurat
3. Dapat mendorong lengkapnya data, disebabkan data yang dibutuhkan disebutkan satu persatu di dalam dokumen dasarnya.

* **Desain Output Terinci**

Desain output terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk output-output dari sistem yang baru. Desain Output Terinci terbagi atas dua, yaitu desain output berbentuk laporan di media kertas dan desain output dalam bentuk dialog di layar terminal.

* 1. Desain output dalam bentuk laporan

Desain ini dimaksudkan untuk menghasilkan output dalam bentuk laporan di media kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

* 1. Desain output dalam bentuk dialog layar terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai sistem (*user*) dengan komputer. Percakapan ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan output informasi kepada *user*, atau keduanya

Beberapa stategi dalam membuat layar dialog terminal :

1. Dialog pertanyaan / jawaban
2. Menu

Menu banyak digunakan karena merupakan jalur pemakai yang mudah dipahami dan mudah digunakan. Menu berisi beberapa alternatif atau *option* atau pilihan yang disajikan kepada user. Pilihan menu akan lebih baik bila dikelompokkan sesuai fungsinya.

* **Desain Database Terinci**

Basis data *(database)* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan database dalam sistem informasi disebut *database system.*

Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam didalam suatu organisasi. Dengan sistem basis data ini tiap-tiap orang atau bagian dapat memandang database dari beberapa sudut pandang yang berbeda. Bagian kredit dapat memandangnya sebagai data piutang, bagian penjualan dapat memandangnya sebagai data penjualan, bagian personalia dapat memandangnya sebagai data karyawan, bagian gudang dapat memandangnya sebagai data persediaan. Semuanya terintegrasi dalam sebuah data yang umum. Berbeda dengan sistem pengolahan data tradisional, sumber data ditangani sendiri-sendiri untuk tiap aplikasinya. Pada tahap ini, desain database dimaksudkan untuk mendefinisikan isi atau struktur dari tiap-tiap file yang telah diidentifikasikan di desain secara umum.

* **Desain Teknologi**

Tahap desain teknologi terbagi atas dua, yaitu desain teknologi secara umum dan terinci. Pada tahap ini kita menentukan teknologi yang akan dipergunakan dalam menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi yang dimaksud meliputi:

1. Perangkat keras (*hardware*), yang terdiri dari alat masukan, alat  
   pemroses, alat output dan simpanan luar
2. Perangkat lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (O*perating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak (*application software*).
3. Sumber daya manusia (*brainware*), misalnya operator komputer, pemrogram, spesialis telekomunikasi, sistem analis dan lain sebagainya

Desain teknologi sangat diperlukan pada tahap implementasi dan pengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

* **Desain Model**

Tahap desain model terbagi menjadi dua, yaitu desain model secara umum dan terinci. Tahap desain model secara umum berupa desain sistem secara fisik dan logika. Desain fisik dapat digambarkan dengan bagan alir sistem dan bagan alir dokumen, dan desain secara logika digambarkan dengan diagram arus data (DAD). Pada tahap desain model terinci, model akan mendefinisikan secara rinci urut-urutan langkah dari masing-masing proses yang digambarkan di DAD. Urut-urutan langkah proses ini diwakili oleh suatu program komputer.

## Teknik Pengujian Sistem

### 2.6.1 White Box

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Pengujian sistem/ perangkat lunak memiliki sejumlah aturan yang berfungsi sebagai sasaran pengujian, diantaranya adalah sebagai berikut :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Pengujian *White Box,* adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh *test case*. Dengan menggunakan metode *white box*, perekayasa sistem dapat melakukan *test case* yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah dugunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true* dan *false,* mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka, dan menggunakan struktur data internal untuk menjamin validatasnya. Pengujian *Basis Path* adalah teknik pengujian *white box* yang diusulkan pertama kali oleh Tom McCabe. Metode *basis path* ini memenugkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi. (Roger S. Pressman, 2002:536)

9

10

6

7

8

1

2

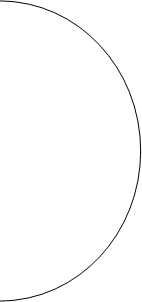
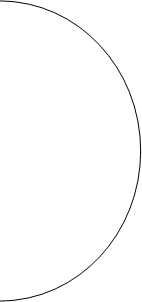
3

5

4

**Gambar 2. 2**Contoh Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan pertama keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. Edge harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural. (Sumber: Roger S. Pressman, 2002. 536)



R3

R2

R1

R4

**Gambar 2. 3**Contoh Grafik Alir

*Kompleksitas siklomatis* adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metriks ini digunakan dalam konteks metode pengujian *basis path*, maka nilai yang terhitung untuk kompleksitas siklomatis menentukan jumlah *jalur independen*. Jalur independen adalah jalur yang melalui program yang mengintroduksi sedikitnya satu rangkaian statemen proses baru atau suatu kondisi baru. Bila dinyatakan dengan terminologi grafik alir, jalur independen harus bergerak sepanjang paling tidak satu edge yang tidak dilewatkan sebelum jalur tersebut ditentukan. Sebagai contoh, serangkaian jalur independen untuk grafik alir yang ditunjukkan pada gambar 2.12. adalah :

Jalur 1 : 1 – 11

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan di atas terdiri dari sebuah *basis set* untuk grafik alir pada gambar 2.12. Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi kompleksitas siklomatis memberikan jawaban. Fondasi *kompleksitas siklomatis* adalah teori grafik, dan memberi kita metriks perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis.
2. Kompleksitas siklomatis, *V(G),* untuk grafik alir *G* ditentukan sebagai *V(G)* = *E – N* + 2 di mana *E* adalah jumlah edge grafik alir dan *N* adalah jumlah simpul grafik alir.
3. Kompleksitas siklomatis, *V(G),* untuk grafik alir G juga ditentukan sebagai *V(G)* = P + 1, dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir *G*.

Pada gambar 2.12. grafik alir, kompleksitas siklomatis dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region.
2. *V(G)* = 11 edge – 9 simpul + 2 = 4.
3. *V(G)* = 3 simpul yang diperkirakan + 1 =4.

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.12 adalah 4. Yang lebih penting, nilai untuk *V(G)* memberi kita batas atas untuk jumlah jalur independen yang membentuk *basis set*, dan implikasinya, batas atasjumlah pengujian yang harus didesain dan dieksekusi untuk menjamin semua statemen program.

### 2.6.2 Black Box

*Black box aproach* adalah Suatu sistem dimana *input* dan *output*nya dapat didefinisikan tetapiprosesnya tidak diketahui atau tidak terdefinisi.Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani sedangkan pihak luar hanya mengetahui masukan dan hasilnya). Sistem ini terdapat pada subsistem tingkat terendah.

Metode ujicoba *black box* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *black box* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *black box* bukan merupakan alternatif dari ujicoba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*. Ujicoba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database eksternal*
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Tidak seperti metode *white box* yang dilaksanakan diawal proses, ujicoba *black box* diaplikasikan dibeberapa tahapan berikutnya. Karena ujicoba *black box*dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji ?
2. Jenis *input* seperti apa yang akan menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan-batasan kelas data diisolasi ?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem ?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan ujicoba *black box*, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari uji kasus tambahan harus didesain untuk mencapai ujicoba yang cukup beralasan.
2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, dari pada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik.

## 2.7 Perangkat Lunak Pendukung

Perangkat lunak pendukung yang digunakan penulis dalam membangun sistem ini yaitu PHP dan MySQL, seperti pada tabel di bawah ini :

**Tabel 2. 1**Perangkat Lunak Pendukung

|  |  |  |
| --- | --- | --- |
| **NO** | **PERANGKAT LUNAK PENDUKUNG** | |
| 1 | PHP | Sebuah bahasa *scripting* yang terpasang pada HTML. Yang bertujun untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. |
| 2 | MySQL | Salah satu pengolah database yang menggunakan SQL (*Strukture Query Language*) sebagai bahan dasar untuk mengakses databasenya. Yang memiliki keuntungan seperti *open source* dan memiliki kemampuan menampung kapasitas yang besar. |

## 2.8 Implementasi Sistem

Sistem telah dianalisa dan didesain secara rinci dan teknologi telah diseleksi dan dipilih. Tiba saatnya sekarang system untuk di implementasikan (diterapkan).Tahap implementasi system merupakan tahap meletakkan system supaya siap untuk dioperasikan.Tahap implementasi system dapat terdiri dari langkah-langkah sebagai berikut :

1. Menerapkan Rencana Implementasi. Rencana implementasi merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan terutama untuk mengatur biaya dan waktu yang dibutuhkan selama tahap implementasi.

2. Melakukan kegiatan implementasi. Kegiatan implementasi dilakukan dengan dasar kegiatan yang telah direncanakan dalam rencana implementasi. Kegiatan-kegiatan yang dapat dilakukan dalam tahap ini adalah sebagai berikut :

a. Pemilihan dan pelatihan personil. Telah diketahui bahwa manusia merupakan faktor yang perlu dipertimbangkan dalam system informasi. Jika system informasi ingin sukses, maka personil-personil yang terlihat harus diberi pengertian dan pengetahuan yang cukup tentang system informasi dan posisi serta tugas mereka nanti.

b. Persiapan tempat dan instalasi perangkat keras dan perangkat lunak. Jika peralatan baru akan dimiliki, maka tempat atau ruangan untuk peralatan ini perlu dipersiapkan terlebih dahulu. Keamanan fisik dari tempat ini perlu juga dipertimbangkan. Sistem komputer yang besar membutuhkan tempat dengan lingkungan yang lebih, harus dipertimbangkan. Langkah selanjutnya setelah persiapan fisik tempat adalah menginstalasi perangkat keras yang sudah dikirim dan menginstalasi perangkat lunak yang sudah ada.

c. Pemrograman dan pengetesan sistem. Pemrograman merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemrogram harus berdasarkan dokumentasi yang disediakan oleh analis system hasil dari desain system secara rinci. Sebelum program diterapkan, maka terlebih dahulu program bebas dari kesalahan-kesalahan. Oleh sebab itu, program harus diuji untuk menemukan kesalahan-kesalahan yang mungkin dapat terjadi. Program ditest untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai.

d. Pengetesan sistem. Pengetesan sistem biasanya dilakukan setelah pengetesan program. Pengetesan sistem dilakukan untuk memeriksa kekompakan antara komponen sistem yang diimplementasikan.Tujuan utama dari pengetesan sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.

## 2.9 Kerangka Pemikiran

**Masalah**

1. Bagaimana cara merekayasa” Sistem Cerdas Diagnosa Penyakit Kaki Gajah Menggunakan Metode Naïve Bayes”?
2. Bagaimana hasil penerapanSistem mengenai ” Sistem Cerdas Diagnosa Penyakit Kaki Gajah Menggunakan Metode Naïve Bayes”?

**Analisis Sistem**

* Sistem Berjalan
* Sistem Diusulkan

**Desain Sistem**

* Desain Model
* Desain User Interface
  + Desain Output
  + Desain Input
* Desain Database
* Desain Teknologi

**Pembangunan Sistem**

* PHP(Hypertext Preeprocessor)
* MySQL(My Strukture Query langunge)

**Pengujian Sistem**

* White Box
* Black Box

**Tujuan**

1. Untuk mengetahui rekayasa ” Sistem Cerdas Diagnosa Penyakit Kaki Gajah Menggunakan Metode Naïve Bayes”
2. Untuk mengetahui hasil penerapan Sistem informasi mengenai ” Sistem Cerdas Diagnosa Penyakit Kaki Gajah Menggunakan Metode Naïve Bayes”

**Solusi**

Sistem Cerdas Diagnosa Penyakit Kaki Gajah Menggunakan Metode Naïve Bayes

**Implementasi Sistem**

Puskesmas Bulango

**Peluang**

Kebutuhan masyarakat akan dokter yang dapat mendiagnosa penyakit kaki gajah.

**Gambar 2. 4**Kerangka Pikir