



**SISTEM PAKAR DIAGNOSA PENYAKIT DEMAM MENGGUNAKAN
FORWARD CHAINING BERBASIS MOBILE**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir
Memperoleh Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : ARMAN AHMADI
N.P.M : 1414370036
PROGRAM STUDI : SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

ARMAN AHMADI

SISTEM PAKAR DIAGNOSA PENYAKIT DEMAM BERDARAH DENGAN METODE FORWARD CHAINING BERBASIS MOBILE

Penyakit Demam Berdarah Dengue (DBD) pertama kali dilaporkan di Surabaya pada tahun 1968. Penyakit DBD ini terus mengalami peningkatan dan menyebar bertambah luas. Seluruh wilayah Indonesia mempunyai risiko terjangkit penyakit ini. Hampir sepanjang tahun penyakit DBD ini selalu ditemukan di seluruh Indonesia terutama pada awal musim penghujan. Sebagian besar wilayah tropis dan subtropis sering ditemukan penyakit Demam Berdarah Dengue (DBD), terutama Asia Tenggara, Amerika Tengah, dan Karibia. Manusia adalah host DBD serta virus dengue dengan kategori famili Flaviridae dan genus Flavivirus sebagai agent. Media perantara virus dengue menuju host manusia yaitu melalui vektor nyamuk yang sudah terinfeksi, antara lain nyamuk *Aedes aegypti* dan *Aedes albopictus*. *World Health Organization* atau WHO menyimpulkan bahwa insiden terjadinya DBD di dunia mengalami perkembangan yang sangat pesat karena diperkirakan 390 juta terinfeksi oleh virus dengue per tahun. Dengan situasi tersebut. Penulis ingin membuat sistem yang dapat membantu masyarakat dalam mendiagnosa penyakit demam berdarah yang diderita oleh seseorang. Perancangan aplikasi yang akan dibuat berbentuk aplikasi mobile yang dapat digunakan oleh semua orang dan dapat dipasang pada semua smartphone yang menggunakan sistem operasi android. Aplikasi yang akan dibangun juga menggunakan metode *forward chaining* dalam mendiagnosa gejala yang terjadi pada penyakit demam berdarah. Aplikasi diagnosa penyakit demam berdarah ini dirancang semudah mungkin agar pengguna yang masih awam dapat menggunakan aplikasi secara mudah. Sehingga pengguna yang akan menggunakannya dapat dengan cepat mendeteksi jenis penyakit demam berdarah yang diderita dan memberikan penanganan kepada jenis penyakit tersebut.

Kata kunci : *Android, Demam Berdarah, Forward Chaining, Sistem pakar.*

DAFTAR ISI

Halaman

| | |
|------------------------------------|-------------|
| LEMBAR JUDUL | |
| LEMBAR PENGESAHAN | |
| ABSTRAK | |
| KATA PENGANTAR | i |
| DAFTAR ISI | iv |
| DAFTAR GAMBAR | viii |
| DAFTAR TABEL | x |
| BAB I PENDAHULUAN | 1 |
| 1. Latar Belakang Masalah | 1 |
| 2. Rumusan Masalah | 2 |
| 3. Batasan Masalah | 3 |
| 4. Tujuan Penelitian | 3 |
| 5. Manfaat Penelitian | 4 |
| 6. Metode Penelitian | 4 |
| a. Metode Pengumpulan Data | 4 |
| b. Metode Perancangan Sistem | 5 |
| 7. Sistematika Penulisan | 7 |
| BAB II LANDASAN TEORI | 9 |
| 1. Pengertian Sistem | 9 |
| 2. Karakteristik Sistem | 11 |
| 3. Pengertian Sistem Pakar | 13 |

| | |
|--|-----------|
| 4. Metode Forward Chaining | 16 |
| 5. Penyakit Demam | 17 |
| 6. Penyakit Demam Berdarah | 19 |
| a. Demam Berdarah Di Indonesia..... | 20 |
| 7. Flowmap | 22 |
| 8. Unified Modelling Language (UML) | 25 |
| a. PengenalanUML | 25 |
| b. Use Case Diagram | 25 |
| c. Activity Diagram | 27 |
| d. Sequence Diagram | 28 |
| e. Class Diagram | 29 |
| 9. Aplikasi Mobile | 30 |
| 10. Definisi Visual Studio 2015 | 31 |
| 11. Android..... | 32 |
| 12. Pengertian Database | 34 |
| 13. Perancangan Basis Data (Database Design)..... | 36 |
| 14. Database SQLite | 37 |
| BAB III ANALISIS DAN PERANCANGAN SISTEM | 39 |
| 1. Analisis Sistem | 39 |
| 2. Fungsional Aiplikasi | 41 |
| 3. Analisis Kebutuhan | 42 |
| 4. Rule Aplikasi Diagnosa Penyakit Demam Berdarah..... | 44 |
| 5. Perancangan Sistem | 50 |

| | |
|--|-----------|
| a. Use Case Diagram..... | 50 |
| b. Use Case..... | 50 |
| c. Skenario Use Case..... | 51 |
| 6. Activity Diagram | 53 |
| 7. Diagram Sequence..... | 56 |
| 8. Perancangan Aplikasi | 58 |
| 9. Perancangan Class Diagram..... | 62 |
| 10. Perancangan Database | 63 |
| a. Perancangan Tabel..... | 63 |
| BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM | 64 |
| 1. Implementasi Sistem Yang Digunakan | 64 |
| a. Spesifikasi Perangkat Keras..... | 64 |
| b. Spesifikasi Perangkat Lunak..... | 64 |
| 2. Tampilan Aplikasi Diagnosa Penyakit Demam Berdarah | 65 |
| a. Tampilan Menu Utama..... | 65 |
| b. Tampilan Demam Berdarah..... | 66 |
| c. Tampilan Diagnosa | 67 |
| d. Tampilan Hasil Diagnosa | 68 |
| e. Tampilan tentang | 69 |
| 3. Pengujian Aplikasi Diagnosa Penyakit Demam Berdarah | 70 |
| 4. Evaluasi | 72 |
| BAB V PENUTUP | 74 |
| 1. Kesimpulan | 74 |

2. Saran 74

DAFTAR PUSTAKA

LAMPIRAN

BAB I

PENDAHULUAN

1. Latar Belakang Masalah

Penyakit Demam Berdarah *Dengue* (DBD) pertama kali dilaporkan di Surabaya pada tahun 1968. Penyakit DBD ini terus mengalami peningkatan dan menyebar bertambah luas. Seluruh wilayah Indonesia mempunyai risiko terjangkit penyakit ini. Hampir sepanjang tahun penyakit DBD ini selalu ditemukan di seluruh Indonesia terutama pada awal musim penghujan (Sucipto, 2015).

Sebagian besar wilayah tropis dan subtropis sering ditemukan penyakit Demam Berdarah *Dengue* (DBD), terutama Asia Tenggara, Amerika Tengah, dan Karibia. Manusia adalah host DBD serta virus *dengue* dengan kategori famili *Flaviridae* dan *genus Flavivirus* sebagai agent. Media perantara virus dengue menuju host manusia yaitu melalui vektor nyamuk yang sudah terinfeksi, antara lain nyamuk *Aedes aegypti* dan *Aedes albopictus*. *World Health Organization* atau WHO menyimpulkan bahwa insiden terjadinya DBD di dunia mengalami perkembangan yang sangat pesat karena diperkirakan 390 juta terinfeksi oleh virus *dengue* per tahun. Kasus di Amerika, Asia Tenggara, dan Pasifik Barat diperkirakan lebih dari 3,2 juta terjangkit DBD pada tahun 2015 (Bestari, 2018).

Situasi tersebut dapat dihindari jika masyarakat memiliki sedikit pengetahuan tentang kesehatan. Penulis ingin membuat sistem yang dapat membantu masyarakat dalam mendiagnosa penyakit demam berdarah yang diderita oleh seseorang. Untuk itu penulis membuat suatu aplikasi yang dapat

mengetahui jenis penyakit demam berdarah yang diderita menggunakan metode *forward chaining* agar dapat memberikan hasil jenis penyakit demam berdarah yang diderita. Sistem akan memberikan daftar gejala-gejala sampai bisa mengidentifikasi suatu objek berdasarkan jawaban yang diterimanya. sistem pakar ini dapat menjelaskan dan mengdiagnosa apakah seseorang terdiagnosa penyakit Demam berdarah (klasik), Demam berdarah *dengue (hemoragik)* atau *Sindrom Syok Dengue*.

Forward Chaining merupakan fakta untuk mendapatkan kesimpulan (*conclusion*) dari fakta tersebut. Penalaran ini berdasarkan fakta yang ada (*data driven*), metode ini adalah kebalikan metode *Backward Chaining*, dimana metode ini dijalankan dengan mengumpulkan fakta-fakta yang ada untuk menarik kesimpulan. Dengan kata lain, prosesnya dimulai dari fakta-fakta yang ada melalui proses *interface fact* (penalaran fakta-fakta) menuju suatu *goal* (suatu tujuan). Berdasarkan latar belakang diatas maka penulis tertarik untuk memilih judul “**Sistem Pakar Diagnosa Penyakit Demam Berdarah Dengan Metode *Forward Chaining* Berbasis *Mobile***”.

2. Rumusan Masalah

Adapun permasalahan yang dihadapi dalam perancangan diagnosa penyakit demam berdarah ini adalah sebagai berikut:

- a. Bagaimana cara mengetahui apakah seseorang sedang mengalami demam berdarah dengan menggunakan metode *forward chaining*?

- b. Bagaimana membuat aplikasi diagnosa penyakit demam yang mudah digunakan oleh pengguna yang masih awam secara cepat menggunakan aplikasi *mobile* ?

3. Batasan Masalah

Dalam perancangan aplikasi diagnosa penyakit demam berdarah ini, penulis membatasi masalah sebagai berikut :

- a. Perancangan aplikasi yang penulis rancang ini hanya pada bagian diagnosa jenis penyakit demam berdarah seperti Demam berdarah (klasik), Demam berdarah *dengue* (*hemoragik*) dan *Sindrom Syok Dengue* yang diderita oleh seseorang.
- b. Bahasa pemrograman yang digunakan dalam perancangan aplikasi diagnosa penyakit demam ini adalah *C#* dan *database SQLite* dengan menggunakan metode *forward chaining*.

4. Tujuan Penelitian

Tujuan yang ingin dicapai penulis dalam merancang aplikasi diagnosa penyakit demam berdarah ini adalah:

- a. Agar mengetahui bagaimana membuat suatu sistem pakar dengan menggunakan metode *forward chaining*.
- b. Mempermudah bagi masyarakat agar mengetahui jenis penyakit demam berdarah yang diderita.

5. Manfaat Penelitian

Merancang aplikasi diagnosa penyakit demam berdarah ini bermanfaat bagi penulis dan masyarakat luas antara lain :

- a. Dapat memberikan solusi atas penyakit demam berdarah yang diderita oleh seseorang dengan menggunakan aplikasi yang penulis rancang.
- b. Memberikan kemudahan dan cepat dalam mendeteksi penyakit demam berdarah yang diderita.

6. Metode Penelitian

a. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode deskriptif. Adapun teknik pengumpulan data dilakukan dengan cara sebagai berikut:

1. Studi Literatur

Pengumpulan data dengan cara mengumpulkan *literature*, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

2. Studi Lapangan

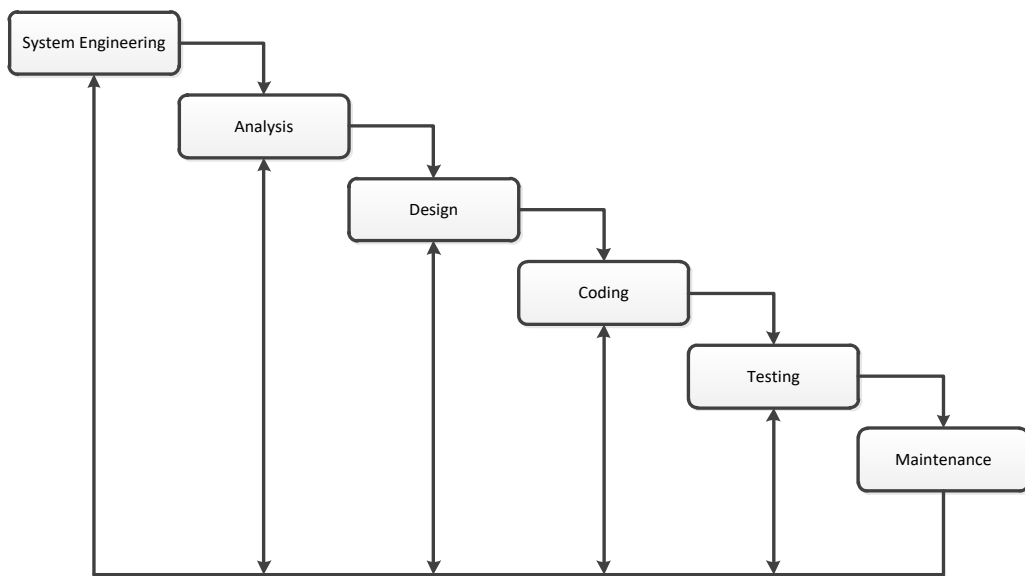
a) Wawancara (*Interview*)

Yaitu memperoleh data dengan mengadakan dengan tanya jawab kepada penderita yang pernah terserang penyakit demam berdarah.

b. Metode Perancangan Sistem

Metodelogi yang digunakan untuk membangun sistem ini adalah Model *Waterfall*. Model ini terdiri dari beberapa tahapan, yaitu: *System Engineering, Analysis, Design, Coding, Testing* dan *Maintenance*.

Paradigma *Waterfall* dapat dilihat pada gambar berikut:



Gambar 1 Paradigma *Waterfall* (*Classic Life Cycle*)

(Sumber: Indrawaty, 2012)

Penjelasan Metodelogi *Waterfall*:

1. *System Engineering* adalah Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan sebagainya.
2. *Analisis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan *software*. Dalam hal ini dilakukan

dengan menganalisa dokumen-dokumen yang digunakan dalam informasi data.

3. *Design* adalah tahap penterjemah dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
4. *Coding* adalah tahap penterjemah data/pemecahan masalah *software* yang telah dirancang dalam bahasa pemrograman yang telah ditentukan dan digunakan dalam pembuatan sistem menggunakan *software Visual Studio 2017* dengan bahasa pemrograman *Android C#* dan *SQLite Database*.
5. *Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak untuk sistem informasi pengolahan data-data kemudian dilanjutkan dengan pengujian terhadap modul-modul dan terakhir pada tampilan antar muka untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan input yang diberikan hasilnya sesuai dengan yang diinginkan.
6. *Implementation* adalah menguji *software* aplikasi yang telah dibuat dan dirancang dengan membuat aplikasi *software* menjadi *exe*, dan diimplementasikan sesuai pembahasan tersebut.
7. *Maintenance* adalah perangkat lunak yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai. Pemeliharaan dapat dilakukan jika ada permintaan tambahan fungsi sesuai dengan

keinginan pemakai ataupun adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.

7. Sistematika Penulisan

Adapun struktur penulisan pada masing-masing bab dalam laporan skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Membahas Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian, Metodologi Penelitian dan Sistematika Penulisan.

BAB II LANDASAN TEORI

Memaparkan teori-teori yang didapat dari sumber-sumber yang relevan untuk digunakan sebagai referensi yang berhubungan dengan objek penelitian.

BAB III ANALISA DAN PERANCANGAN SISTEM

Menjelaskan tentang gambaran sistem serta deskripsi dari hasil analisis sistem yang akan dijadikan sebagai petunjuk untuk perancangan sistem selanjutnya dengan alat bantu perancangan sistem yaitu UML (*Unified Modeling Language*).

BAB IV IMPLEMENTASI SISTEM DAN PENGUJIAN

Bab ini menguraikan implementasi sistem yang telah dibuat, komponen-komponen kebutuhan sistem serta kelemahan dan kelebihan sistem yang telah dibuat.

BAB V PENUTUP

Mengemukakan kesimpulan yang diambil dari hasil penelitian dan perancangan sistem, serta saran-saran untuk pengembangan sistem selanjutnya, agar dapat dilakukan perbaikan dimasa yang akan datang.

BAB II

LANDASAN TEORI

1. Pengertian Sistem

Sistem adalah kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai tujuan tertentu (Yesputra, 2015).

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggerakya yaitu rakyat yang berada dinegara tersebut.

Kata "sistem" sering digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.



Gambar 2. Model Sistem Sederhana

Sumber : Al-Bahra Bin Ladjamudin, 2013

Sistem dapat didefinisikan sebagai suatu komponen atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain dan terpadu (Ladjamudin, 2013). Menurut John Burch dan Gary sistem informasi memiliki komponen-komponen yang saling terintegrasi membentuk satu kesatuan dalam mencapai sasaran sistem, yaitu :

- a) Blok masukan dalam sebuah sistem informasi meliputi metode-metode dan media untuk menangkap data yang akan dimasukkan, dan dapat berupa dokumen dokumen dasar.
- b) Blok model (*Model Blok*) yaitu blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang berfungsi memanipulasi data untuk keluaran tertentu.
- c) Blok keluaran (*Output Block*) yaitu blok berupa data-data keluaran seperti dokumen output dan informasi yang berkualitas.
- d) Blok teknologi (*Technologi Block*) yaitu blok yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem secara keseluruhan.
- e) Blok Basis Data (*Database Block*) Merupakan kumpulan data yang berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan perangkat lunak untuk memanipulasinya.
- f) Blok kendali (*Controls Block*) Meliputi masalah pengendalian terhadap operasional sistem yang berfungsi mencegah dan menangani kesalahan/kegagalan sistem.

2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

a. Komponen-komponen sistem atau elemen sistem dapat berupa :

- 1) Elemen-elemen yang lebih kecil yang disebut *subsistem*, misalkan sistem komputer terdiri dari *subsistem* perangkat keras, perangkat lunak dan manusia.
- 2) Elemen-elemen yang lebih besar yang disebut *supra sistem*. Misalkan bila perangkat keras adalah sistem yang memiliki subsistem CPU, perangkat I/O dan memori, maka sistem perangkat keras adalah sistem yang dapat menjalankan komputer.

b. Batas sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan luar sistem

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar

yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

d. Penghubung

Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu *subsistem* ke *subsistem* lainnya. *Output* dari satu *subsistem* akan menjadi *input* untuk *subsistem* yang lainnya dengan melalui penghubung.

e. Masukkan

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah *maintanance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran

Merupakan hasil dari energi yang diolah oleh sistem. Meliputi keluaran yang berguna, contohnya Informasi yang dikeluarkan oleh komputer, dan keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

g. Pengolah

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan

menjadi keluaran. Contoh CPU pada Komputer, Bagian Produksi yang mengubah bahan baku menjadi barang jadi, Bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

h. Sasaran atau tujuan

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi yang dibutuhkan dan output yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

3. Pengertian Sistem Pakar

Sistem pakar (*expert system*) berasal dari istilah sistem pakar berbasis pengetahuan. Sistem pakar adalah suatu sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia. Sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Sistem pakar merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem ini bekerja untuk mengadopsi pengetahuan manusia ke komputer yang menggabungkan dasar pengetahuan untuk menggantikan seorang pakar dalam menyelesaikan suatu masalah. (Harto, 2013)

Sistem pakar berasal dari istilah *knowledge base expert system*. Sistem pakar adalah suatu sistem yang dirancang agar dapat menyelesaikan suatu

permasalahan tertentu dengan meniru kerja dari para ahli dalam menjawab pertanyaan dan memecahkan suatu masalah. Dengan sistem pakar ini orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli sistem pakar ini juga membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

Sistem pakar merupakan cabang dari AI (*Artificial Inteligent*) yang membuat ekstensi khusus untuk spesialisasi pengetahuan guna memecahkan suatu permasalahan pada *Human Expert* (ahli manusia). *Human Expert* (ahli manusia) merupakan seseorang yang ahli dalam suatu bidang ilmu pengetahuan tertentu, ini berarti bahwa expert memiliki suatu pengetahuan atau skill khusus yang dimiliki oleh orang lain. *Expert* dapat memecahkan suatu permasalahan yang tidak dapat dipecahkan oleh orang lain dengan cara efisien (Nirmala, 2014).

Pengetahuan di dalam *expert system* berasal dari orang atau *knowledge* yang berasal dari buku-buku referensi, surat kabar atau karya ilmiah orang lain, pengetahuan manusia ke dalam komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Atau dengan kata lain sistem pakar adalah sistem yang didesain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah seperti yang dilakukan oleh para pakar dalam hal ini adalah dokter (Nirmala, 2014).

Proses inferensi dilakukan dalam suatu modul yang disebut *inference engine* (mesin inferensi). Ketika representasi pengetahuan pada bagian *knowledge base* (dasar pengetahuan) telah lengkap, atau paling tidak telah berada pada level cukup akurat, maka referensi pengetahuan tersebut telah siap digunakan.

Sedangkan *inferensi engine* (dasar pengetahuan) merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning* (pemikiran).

Terdapat dua metode umum penalaran yang dapat digunakan apabila pengetahuan dipresentasikan untuk mengikuti aturan-aturan sistem pakar yaitu metode *forward chaining* dan metode *backward chaining*.

Adapun komponen yang membentuk suatu sistem pakar adalah sebagai berikut : (Fadhilah, 2012)

a. Basis Pengetahuan (*Knowledge Base*)

Jika proses akuisisi data telah selesai dilakukan, maka data-data tersebut harus direpresentasikan menjadi basis pengetahuan dan basis aturan yang selanjutnya dikumpulkan, dikodekan dan digambarkan dalam bentuk rancangan lain menjadi bentuk yang sistematis.

b. Basis Data (*data base*)

basis data (database) adalah Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

c. Mesin Inferensi (*Inferensi Engineer*)

Mekanisme inferensi adalah bagian dari sistem pakar yang melakukan penalaran atau pelacakan dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi mekanisme inferensi menguji aturan satu demi satu sampai kondisi aturan itu benar.

Ada dua teknik utama Mesin inferensi adalah program komputer yang

memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan memformulasikan kesimpulan.

d. Antar Muka Pemakai (*User Interface*)

Antar muka pemakai memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi dan memberikan tuntunan penggunaan sistem secara menyeluruh langkah demi langkah sehingga pemakai mengerti apa yang harus dilakukan terhadap sistem.

4. Metode *Forward Chaining*

Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju (Verina, 2015)

Forward Chaining adalah tehnik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta – fakta tersebut dengan bagian IF dari *rules IF – THEN*. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam *database*.

Langkah – langkah dalam membuat sistem pakar dengan menggunakan metode forward chaining yaitu : (Irmayani, 2011)

- a. Pendefinisian masalah dimulai dengan pemilihan domain masalah dan akuisi pengetahuan.
- b. Pendefinisian data input, yaitu sistem forward chaining memerlukan data awal untuk memulai inferensi.
- c. Pendefinisian struktur pengendalian data, yaitu aplikasi yang kompleks memerlukan aturan tambahan untuk membantu pengaktifan aturan.
- d. Penulisan kode awal, yaitu untuk menentukan efektifitas pengetahuan sistem dalam struktur aturan yang baik.
- e. Pengujian sistem, yaitu dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.
- f. Perancangan antarmuka, dibuat bersamaan dengan pembuatan basis knowledge.
- g. Pengembangan sistem, yaitu meliputi penambahan antar muka dan pengetahuan sesuai dengan prototipe sistem.
- h. Evaluasi sistem, pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

5. Penyakit Demam

Definisi demam adalah keadaan suhu tubuh di atas suhu normal, yaitu suhu tubuh di atas 38° Celsius. Suhu tubuh adalah suhu visera, hati, otak, yang

dapat diukur lewat oral, rektal, dan aksila. 1,2,3 Cara pengukuran suhu menentukan tinggi rendahnya suhu tubuh. Pengukuran suhu melalui mulut dilakukan dengan mengambil suhu pada mulut (mengulum termometer dilakukan pada anak yang sudah kooperatif), hasilnya hampir sama dengan suhu dubur, namun bisa lebih rendah bila frekuensi napas cepat. Pengukuran suhu melalui dubur (rektal) dilakukan pada anak di bawah 2 tahun. Termometer masuk ke dalam dubur sedalam 2-3 cm dan kedua pantat dikatupkan, pengukuran dilakukan selama 3 menit. Suhu yang terukur adalah suhu tubuh yang mendekati suhu yang sesungguhnya (core temperature). Dikatakan demam bila suhu di atas 38° Celsius. (Ismoedijanto, 2000)

Pengukuran suhu melalui ketiak (*axilar*) hanya dapat dilakukan pada anak besar mempunyai daerah aksila cukup lebar, pada anak kecil ketiaknya sempit sehingga terpengaruh suhu luar. Pastikan puncak ujung termometer tepat pada tengah aksila dan pengukuran dilakukan selama 5 menit. Hasil pengukuran aksila akan lebih rendah 0,5-1,00 C dibandingkan dengan hasil pengukuran melalui dubur. Pengukuran suhu dengan cara meraba kulit, daerah yang diraba adalah daerah yang pembuluh darahnya banyak seperti di daerah pipi, dahi, tengkuk. Meskipun cara ini kurang akurat (tergantung kondisi tangan ibu), namun perabaan ibu cukup bisa dipercaya dan digunakan sebagai tanda demam pada program MTBS.

6. Penyakit Demam Berdarah

Penyakit Demam Berdarah Dengue (DBD) merupakan salah satu penyakit menular yang berbahaya dapat menimbulkan kematian dalam waktu singkat dan sering menimbulkan wabah. Penyakit demam berdarah sampai saat ini masih merupakan masalah di banyak negara terutama di negara-negara berkembang. WHO memperkirakan terdapat 50 juta kasus DBD terjadi setiap tahunnya, sebanyak 500.000 kasus membutuhkan perawatan rumah sakit dan sedikitnya terdapat 22.000 kematian. (Baharuddin, 2018)

Penyakit Demam Berdarah *Dengue* adalah penyakit yang disebabkan oleh virus *Dengue* ditularkan kepada manusia melalui gigitan nyamuk *Aedes Aegypti* dan *Aedes Albopictus*. Di Indonesia merupakan wilayah endemis dengan sebaran di seluruh wilayah tanah air. Gejala yang akan muncul seperti ditandai dengan demam mendadak, sakit kepala, nyeri belakang bola mata, mual dan manifestasi perdarahan seperti mimisan atau gusi berdarah serta adanya kemerahan di bagian permukaan tubuh pada penderita. (Kemkes, 2018)

Pada umumnya penderita DBD (Demam Berdarah *Dengue*) akan mengalami fase demam selama 2-7 hari, fase pertama: 1-3 hari ini penderita akan merasakan demam yang cukup tinggi 40°C, kemudian pada fase ke-dua penderita mengalami fase kritis pada hari ke 4-5, pada fase ini penderita akan mengalami turunnya demam hingga 37°C dan penderita akan merasa dapat melakukan aktivitas kembali (merasa sembuh kembali) pada fase ini jika tidak mendapatkan pengobatan yang adekuat dapat terjadi keadaan fatal, akan terjadi penurunan trombosit secara drastis akibat pemecahan pembuluh darah (perdarahan). Di fase

yang ketiga ini akan terjadi pada hari ke 6-7 ini, penderita akan merasakan demam kembali, fase ini dinamakan fase pemulihan, di fase inilah trombosit akan perlahan naik kembali normal kembali.

a. Demam Berdarah Di Indonesia

Demam Berdarah Dengue di Indonesia DEMAM berdarah dengue (DBD) atau dengue haemorrhagic fever (DHF) bukan penyakit baru di Indonesia. Tahun 1969, kasus pertama DBD dilaporkan di Jakarta (Kho Lin Keng, dkk). Jauh hari sebelum itu, penyakit dengue, cikal bakal munculnya penyakit DBD, sudah dikenal di Indonesia sejak tahun 1779 David Bylon, Batavia). Namun, wabah DBD bukan dimulai di Indonesia, melainkan di Yunani, Amerika Serikat, Australia, dan Jepang, yang terjadi pada sekitar tahun 1920. (Nadesul, 2016)

Di Indonesia sendiri, selain di Iran, Malaysia, Singapura, dan Vietnam, penyakit dengue, atau serangan oleh virus penyebab DBD untuk yang pertama kali, hanya bersifat endemik. Sampai sekarang DBD senantiasa hadir di Indonesia dari musim ke musim. Tahun 2007 di bulan April ini wabah atau KLB (Keadaan Luar Biasa) DBD dinyatakan melanda DKI Jakarta. Virus dengue penyebab DBD memerlukan bantuan nyamuk untuk berpindah ke tubuh manusia. Nyamuknya sendiri harus jenis nyamuk belang-belang hitam-putih Aedes, dan bukan oleh jenis nyamuk lainnya. Nyamuk rumah, nyamuk malaria, dan jenis nyamuk lainnya tidak dapat membawa virus dengue, sehingga bukan nyamuk penularnya. (Nadesul, 2016)

Kita mengenal jenis nyamuk *Aedes aegypti*, nyamuk yang gemar hidup di dalam rumah, dan ada juga *Aedes albopictus*, nyamuk belang hitam-putih juga yang lebih menyukai tinggal di kebun sekitar rumah. Dua-duanya bisa menjadi pembawa virus dengue, atau disebut vector. Untuk Indonesia, *Aedes th* sering sebagai pembawa virus dengunya *gegy* dibanding *Aedes albopictus*. Berbeda dengan *Aedes albopictus* si nyamuk kebun, nyamuk *Aedes aegypti* lebih menyukai tinggal di ruangan rumah yang sejuk, lembab, dan gelap. Hinggapnya bukan di dinding melainkan di barang-barang yang bergelantungan di kamar. Sedang nyamuk *Aedes albopictus* lebih menyukai berada di semak-kebun sekitar rumah. Nyamuk demam berdarah bukan tergolong rakus. Ia hanya menggigit pada jam-jam tertentu saja. Itu pun hanya nyamuk betina yang menggigit. Darah manusia dibutuhkannya untuk bertelur. Jam praktik nyamuk *Aedes* pagi hari pukul 06.00-08.00, dan sore hari pukul 15.00-17.00. Di luar jam praktiknya nyamuk betina hinggap di air jernih tergenang untuk bertelur.

Umur nyamuk *Aedes* hanya sepuluh hari, paling lama dua-tiga minggu. Bertelur 200-400 butir. Perindukannya bukan di air kotor seperti nyamuk lain, melainkan di air jernih. Bukan pula sembarang air jernih, tetapi air jernih yang tergenang tak terusik. Biasanya di air dalam wadah (barang bekas berisi air hujan di pekarangan, talang air, ceruk pohon, atau wadah penyimpan air bersih di dalam rumah, seperti tempayan, gentong, jambangan bunga, baki penampung air di alas kulkas). Jarak terbang nyamuk *Aedes* bisa mencapai seratus meter. Maka, luas penyemprotan (fogging) apabila sudah terjangkit kasus DBD, dilakukan sejauh radius seratus meter dari lokasi pasien DBD. Pamong yang mendapat laporan dari

warga, wajib melaporkan kasus DBD ke dinas kesehatan untuk mendapatkan penyemprotan. Virus dengue berukuran mikron saja.

Kita mengenal ada empat tipe. Ada yang ganas, ada pula yang lebih jinak. Tanpa bantuan nyamuk, virus dengue tak perlu ditakuti. Setelah berhasil memasuki tubuh manusia, virus hanya bertahan hidup tak lebih dari duabelas hari. Sesudah itu mati sendiri. Dari mana virus dengue datang? Virus dengue berasal dari tubuh pasien yang sedang terserang virus dengue. Kemudian apabila ada nyamuk *Aedes* yang dua jenis itu menggigit tubuh si pengidap virus, virus akan bersiklus hidup di dalam tubuh nyamuk. Nyamuk yang di dalam tubuhnya sudah bervirus lalu memindahkannya ke tubuh orang sehat setelah menggigitnya. Begitu seterusnya terjadi.

Virus dengue berpindah dan berpindah lagi ke banyak tubuh sehat lainnya melalui gigitan nyamuk bervirus. Hanya nyamuk *Aedes* bervirus saja yang bisa menularkan penyakit DBD. Nyamuk *Aedes* yang di tubuhnya tidak membawa virus, bukan nyamuk penular DBD Indonesia sebetulnya sudah sejak dulu memiliki kedua jenis nyamuk *Aedes* tersebut. Namun, oleh karena baru belakangan ada virus dengue yang dibawa masuk ke sini, maka penyakit DBD baru berjangkit empat dasawarsa lalu.

7. Flowmap

Flowmap mempunyai fungsi untuk mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis komputer) dan aliran data (dalam bentuk dokumen keluaran dan masukan). Pengertian *Flowmap* adalah campuran

peta dan *flowchart*, yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan.

Flowmap menolong analis dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowmap* dapat dikatakan sebuah aliran data berbentuk dokumen atau formulir didalam suatu sistem informasi yang merupakan suatu aktivitas yang saling terkait dalam hubungannya dengan kebutuhan data dan informasi. Proses aliran dokumen ini dapat terjadi dengan entitas di luar sistem.

Bila seorang analis dan *programmer* akan membuat *flowmap*, ada beberapa petunjuk yang harus diperhatikan, seperti:

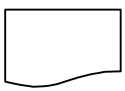

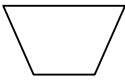
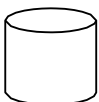
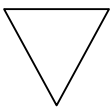

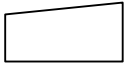
- a. *Flowmap* digambarkan dari halaman atas ke bawah dan kiri ke kanan.
- b. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat di mengerti oleh pembacanya.
- c. Kapan aktivitas dimulai dan berakhir harus di tentukan secara jelas.
- d. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
- e. Lingkup dan *range* dari aktifitas yang sedang di gambarkan harus di telusuri dengan hati-hati.
- f. Gunakan simbol-simbol *Flowmap* yang standar

Kegunaan dari *Flowmap* ini adalah :

- a. Menggambarkan aktivitas apa saja yang sedang berjalan
- b. Menjabarkan aliran dokumen yang terlihat

- c. Menjelaskan hubungan
- d. Hubungan data dan informasi dengan bagian
- e. Bagian dalam aktivitas tersebut. Mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis komputer) dan aliran data (dalam bentuk dokumen keluaran dan masukan)

Tabel 1. Simbol *Flowmap*

| Symbol | Nama Simbol | Keterangan |
|---|-----------------|---|
|  | Dokumen | Digunakan untuk menggambarkan semua jenis dokumen yang merupakan formulir yang digunakan untuk mengentry data keluarga. |
|  | Proses | Merupakan kegiatan proses dari operasi program komputer |
|  | Proses manual | Merupakan proses manual pada flowmap |
|  | File Harddisk | Merupakan media penyimpanan dari proses entry data dan proses komputerisasi |
|  | Offline Storage | Merupakan tempat penyimpanan data berupa arsip |
|  | Garis Alir | Merupakan arus data |
|  | Keyboard | Merupakan proses penyimpanan menggunakan keyboard |

Sumber : Denis Hadi, 2015

8. *Unified Modeling Language (UML)*

a) *Pengenalan UML*


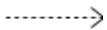

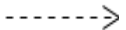



Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa\is dan desain yang berisi sintak dalam memodelkan sistem secara visual (Haviluddin, 2011). Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram (DFD)* untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktur, kemudian juga ada *State Transition Diagram (STD)* yang digunakan untuk memodelkan *real time* (waktu nyata).




Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML)*.

b) *Use Case Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case* (Haviluddin, 2011).

Tabel 2. Simbol *Use Case Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|--|
| 1 |  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
| 3 |  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
| 4 |  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> . |
| 5 |  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |
| 6 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |

| | | | |
|----|---|----------------------|---|
| 8 |  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |






Sumber : (Gellysa Urva, 2015)

c) *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 3. Simbol *Activity Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|--------|------|------------|
| | | | |

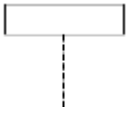
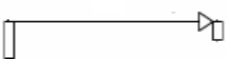
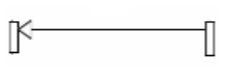
| | | | |
|---|---|---------------------|---|
| 1 |  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | <i>Action</i> | <i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | <i>Initial Node</i> | Bagaimana objek dibentuk atau diawali. |
| 4 |  | <i>Final Node</i> | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | <i>Fork Node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

Sumber : (Gellysa Urva, 2015)

d) *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 4. Simbol *Sequence Diagram*

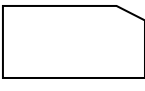
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------|--|
| 1 |  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi. |
| 2 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

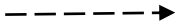

Sumber : (Gellysa Urva, 2015)

e) *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Tabel 5. Simbol *Class Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------|---|
| 1 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

| | | | |
|---|---|-------------------|---|
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya |
| 3 |  | <i>Extend</i> | Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan. |

Sumber : (Gellysa Urva, 2015)

9. Aplikasi *Mobile*

Aplikasi *Mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *Handphone*. Dengan menggunakan aplikasi *Mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya. Pemanfaatan aplikasi *Mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, karena dengan memanfaatkan adanya fitur game, music player, sampai video player membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun. (Kosidin, 2016) Dengan menggunakan aplikasi mobile, maka dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya. (Surahman, 2017)

Beberapa penelitian juga sudah banyak yang menggunakan aplikasi mobile, baik itu untuk hiburan, mempermudah dalam layanan komunikasi data, maupun sebagai pengendali alat kamera DSLR. Aplikasi mobile dibangun dengan

beberapa bahasa pemrograman mobile. Adapun contoh dari mobile programming untuk ponsel di antaranya adalah Javafx mobile, J2ME, C++, C#.NET dan Flash Lite. (Surahman, 2017)

10. Definisi *Visual Studio 2015*

Visual Studio 2015 adalah rilis pertama *Microsoft* yang besar sejak beralih ke pendekatan *open source* untuk .NET dan teknologi terkait. Termasuk *compiler Roslyn* baru untuk C# dan *Visual Basic*, *.NET Core Framework*, *ASP.NET* itu sendiri, dan banyak lagi. Hasilnya memungkinkan jangkauan yang lebih luas untuk aplikasi .NET, termasuk membangun dan penggelaran di *Mac*, *Linux*, dan *Windows*. *Microsoft* juga telah bekerja untuk mengintegrasikan Visual Studio dengan *framework* JavaScript sumber berbasis komunitas, manajer paket, dan perangkat UI. Model ASP.NET 5 menyederhanakan pengembangan web modern menggunakan kerangka kerja seperti *Bootstrap*, *AngularJS*, *Knockout*, *Gulp*, dan banyak lagi. (Lars, 2015)

Visual Studio 2015 mendukung model *Universal App* yang baru untuk dibangun di *Windows*. Aplikasi ini bisa ditulis sekali dan disesuaikan dengan desktop, tablet, dan telepon. Ini termasuk dukungan yang akan datang untuk pengembangan *Windows 10*. Pengembangan *Mobile cross-platform* juga didukung. *Microsoft* telah menyediakan template proyek untuk *Apache Cordova open-source*. Hal ini memungkinkan pengembang untuk membangun aplikasi *Mobile* yang berjalan di *iOS*, *Android*, dan *Windows Phone* menggunakan

teknologi web *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS), dan *JavaScript*.

Umumnya, programmer yang sudah berada di tingkat menengah akan tertarik untuk mempelajari bahasa pemrograman ini. Dibandingkan *Visual Basic*, *Visual C#* lebih sulit dicerna para pemula karena banyak syntax yang dikemas dalam simbol. Kemudian, untuk investasi jangka panjang, elajirlah Python Berbeda dengan kedua bahasa pemrograman sebelumnya, Visual Studio tidak mendukung Python secara penuh. Namun demikian, bukan berarti Anda tidak bisa memilih Visual Studio sebagai editor serta IDE yang mudah bagi bahasa pemrograman ini. Python dianggap investasi karena sanggup mengolah struktur data berukuran besar secara cepat dan mudah.

Di masa depan, struktur data merupakan salah satu tulang punggung artificial intelligence. Memang sangat sulit menghadirkan ketiga bahasa pemrograman itu dalam satu buku secara komprehensif karena keterbatasan jumlah halaman. Oleh karena itu, buku ini ditujukan bagi programmer pemula yang ingin mengetahui secara cepat dan praktis bagaimana memulai dari awal ketiga bahasa itu dipelajari. Dengan kata lain, pembahasan terkait ketiga bahasa pemrograman di atas dibatasi hanya pada struktur dasar. (Jubilee, 2017)

11. Android

Android merupakan salah satu *Mobile Operating System* atau sistem operasi *handphone* yang berupa software platform open source untuk *Mobile device*, yang mana *Mobile Operating System* yaitu sistem operasi yang dapat

mengontrol sistem dan kinerja barang elektronik berbasis *Mobile*, yang fungsinya sama seperti *Windows*, *Linux* dan *Mac OS X* pada *desktop PC* atau Notebook atau Laptop tetapi lebih sederhana. (Muharom, 2013)

Android merupakan sistem operasi yang berisi middleware serta aplikasi-aplikasi dasar. Basis sistem operasi *Android* yaitu kernel linux 2.6 yang telah diperbaharui untuk *Mobile device*. Pengembangan aplikasi *Android* menggunakan bahasa pemrograman java. Yang mana konsep-konsep pemrograman java berhubungan dengan Pemrograman Berbasis Objek (OOP)). Selain itu pula dalam pengembangan aplikasi *Android* membutuhkan software development kit (SDK) yang disediakan *Android*, SDK ini memberi jalan bagi programmer untuk mengakses *application programming interface* (API) pada *Android*.

Android memiliki beberapa fitur yang menarik bagi yang ingin mengembangkan aplikasi, diantaranya sebagai berikut : (Fahnun, 2013)

- a. *Application Framework* yang memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- b. *Dalvik Virtual Machine*, yaitu mesin virtual yang dioptimalkan untuk perangkat *Mobile*.
- c. *Graphic Library*, yang mendukung grafik 2D dan 3D berdasarkan OpenGL Library.
- d. *Media Supported*, yang mendukung beberapa media seperti: audio, video, dan berbagai format gambar(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).

- e. *Hardware Independent*, mendukung GSM, *Bluetooth*, EDGE, 3G, Wifi, kamera, GPS, kompas, dan *accelerometer*

12. Pengertian Database

Data Base Management System (DBMS) merupakan *software* yang akan menentukan data diorganisasikan, disimpan, diubah, diambil kembali, dan membaca data. DBMS merupakan antarmuka bagi pemakai dalam mengorganisasikan *database* yang disusunnya (Yulansari, 2013: 7).

Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis dapat diartikan sebagai markas atau gudang dimana tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Data Base Management System (DBMS) adalah merupakan suatu sistem *software* yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. *Database* sendiri adalah sekumpulan data yang berhubungan secara logika dan memiliki beberapa arti yang saling berpautan.

Sistem Basis Data merupakan suatu sistem menyusun dan mengelola *record-record* menggunakan *computer* untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga

mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Salah satu cara menyajikan data untuk mempermudah modifikasi adalah dengan cara pemodelan data. Model yang akan dipergunakan pada pelatihan ini adalah *Entity Relationship Model*.

Berdasarkan Orientasi pemakainnya DBMS dikelompokkan dalam 2 kategori, yaitu:

- a. DBMS yang berorientasi untuk satu atau sedikit pemakai. Contoh: *MS-Access, dBase/Clipper, FoxBase, dan Borland-Paradox*.
- b. DBMS yang berorientasi untuk banyak pemakai. Contoh: *IBM-DB2, Borland-Interbase, Informix, Oracle, MS-SQL Server, MySQL*

Berdasarkan perkembangan teknologinya:

- a. DBMS Konvensional (*Legacy DBMS*)
- b. DBMS Berorientasi Objek (*Objek-Oriented DBMS/OODBMS*)
- c. DBMS Obejk Relasional (*Objek-Relational DBMS/ORDBMS*)
- d. DBMS untuk Web/Internet (*Internet DBMS*)

Komponen Sistem Basis Data terdiri dari 6 Komponen , yakni :

- a. *Hardware*

Biasanya berupa perangkat komputer standar, media penyimpan sekunder dan media komunikasi untuk sistem jaringan.

- b. *Operating System*

Yakni merupakan perangkat lunak yang memfungsikan, mengendalikan seluruh sumber daya dan melakukan operasi dasar dalam sistem komputer. Harus sesuai dengan DBMS yang digunakan.

c. *Database*

Yakni basis data yang mewakili sistem tertentu untuk dikelola. Sebuah sistem basis data bisa terdiri dari lebih dari satu basis data.

d. *DBMS (Database Management System)*

Perangkat lunak yang digunakan untuk mengelola basis data. Contoh kelas sederhana: *dBase, Foxbase, Rbase, MS. Access, MS. Foxpro, Borland Paradox*. Contoh kelas kompleks: *Borland-Interbase, MS. SQL Server, Oracle, Informix, Sybase*.

e. *User (Pengguna Sistem Basis Data)*

Orang-orang yang berinteraksi dengan sistem basis data, mulai dari yang merancang sampai yang menggunakan di tingkat akhir.

f. *Optional Software*

Perangkat lunak pelengkap yang mendukung dan bersifat opsional.

13. Perancangan Basis Data (*Database Design*)

Proses perancangan database merupakan bagian dari micro lifecycle. Sedangkan kegiatan-kegiatan yang terdapat di dalam proses tersebut diantaranya : pengumpulan data dan analisis, perancangan database secara konseptual, pemilihan DBMS, perancangan database secara logika (data model mapping),

perancangan database secara fisik, dan implementasi sistem database. (Abdillah, 2006)

Sedangkan kegiatan utama dalam perancangan suatu database adalah :

- a. perancangan basisdata secara konseptual (*conceptual scheme design*).
- b. perancangan basis data secara logika (*logical design*).
- c. perancangan basisdata secara fisik (*phisycal design*).

Tujuan perancangan basisdata :

- a. untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan user secara khusus dan aplikasi-aplikasinya.
- b. memudahkan pengertian struktur informasi.
- c. mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan (*response time, processing time, dan storage space*).

14. Database SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. (Maulana, 2017)

Tidak seperti pada paradigma *client-server* umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme

seperti ini tentunya membawa keuntungan karena dapat mereduksi overhead, latency times, dan secara keseluruhan lebih sederhana. Seluruh elemen basis data (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

SQLite merupakan paket perangkat lunak yang bersifat *public domain* yang menyediakan sistem manajemen basis data relasional atau RDBMS. Sistem basis data relasional digunakan untuk menyimpan *record* yang didefinisikan oleh pengguna pada ukuran tabel yang besar dan memproses perintah *query* yang kompleks dan menggabungkan data dari berbagai tabel untuk menghasilkan laporan dan rangkuman data. Kata '*Lite*' pada SQLite tidak menunjuk pada kemampuannya, melainkan menunjuk pada sifat dari SQLite, yaitu ringan ketika dihubungkan dengan kompleksitas pengaturan, *administrative overhead*, dan pemakaian sumber. (Noer, 2017)

BAB III

ANALISA DAN PERANCANGAN SISTEM

1. Analisis Sistem

a. Analisis masalah

Analisis masalah adalah penguraian dari suatu masalah yang utuh ke dalam bagian-bagian komponennya dengan maksud mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Penyakit Demam Berdarah Dengue (DBD) pertama kali dilaporkan di Surabaya pada tahun 1968. Penyakit DBD ini terus mengalami peningkatan dan menyebar bertambah luas. Seluruh wilayah Indonesia mempunyai risiko terjangkit penyakit ini. Hampir sepanjang tahun penyakit DBD ini selalu ditemukan di seluruh Indonesia terutama pada awal musim penghujan.

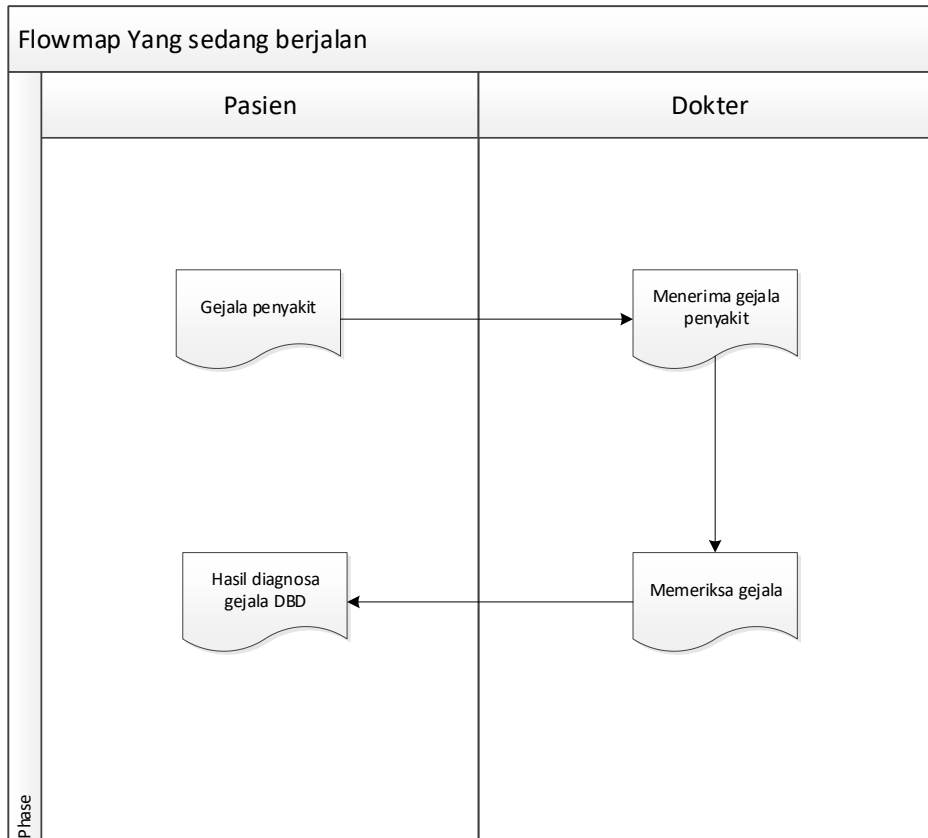
Sebagian besar wilayah tropis dan subtropis sering ditemukan penyakit Demam Berdarah *Dengue* (DBD), terutama Asia Tenggara, Amerika Tengah, dan Karibia. Manusia adalah host DBD serta virus dengue dengan kategori famili *Flaviridae* dan genus *Flavivirus* sebagai agent. Media perantara virus dengue menuju host manusia yaitu melalui vektor nyamuk yang sudah terinfeksi, antara lain nyamuk *Aedes aegypti* dan *Aedes albopictus*. *World Health Organization* atau WHO menyimpulkan bahwa insiden terjadinya DBD di dunia mengalami perkembangan yang sangat pesat karena diperkirakan 390 juta terinfeksi oleh

virus dengue per tahun. Kasus di Amerika, Asia Tenggara, dan Pasifik Barat diperkirakan lebih dari 3,2 juta terjangkit DBD pada tahun 2015.

Situasi tersebut dapat dihindari jika masyarakat memiliki sedikit pengetahuan tentang kesehatan. Penulis ingin membuat sistem yang dapat membantu masyarakat dalam mendiagnosa penyakit demam berdarah yang diderita oleh seseorang. Untuk itu penulis membuat suatu aplikasi yang dapat mengetahui jenis penyakit demam berdarah yang diderita menggunakan metode *forward chaining* agar dapat memberikan hasil jenis penyakit demam berdarah yang diderita. Sistem akan memberikan daftar gejala-gejala sampai bisa mengidentifikasi suatu objek berdasarkan jawaban yang diterimanya. sistem pakar ini dapat menjelaskan dan mengdiagnosa apakah seseorang terdiagnosa penyakit Demam berdarah (klasik), Demam berdarah dengue (*hemoragik*) atau *Sindrom Syok Dengue*.

Sistem yang akan diusulkan nantinya diharapkan bisa membantu pengguna dalam proses mendeteksi apakah terserang demam berdarah atau tidaknya. Proses-proses yang dilakukan nantinya dapat dilakukan tanpa memakan waktu dan uang yang dikeluarkan oleh pengguna.

b. Flowmap



Gambar 3. Flowmap Yang Sedang Berjalan

2. Fungsional Aplikasi

Melihat dari banyaknya masyarakat yang belum memahami tentang penyakit demam berdarah ini, maka dari itu perancangan aplikasi diagnosa penyakit demam berdarah memiliki fungsi sebagai berikut :

- a. Memberikan hasil dari diagnosa pertanyaan yang dijawab pengguna untuk mendeteksi pengguna apakah penyakit demam berdarah atau tidaknya.

- b. Aplikasi ini diharapkan memberikan manfaat bagi seluruh masyarakat luas agar dapat mendeteksi penyakit demam berdarah yang terjadi sebelum dikonsultasikan ke yang lebih ahli.

3. Analisis Kebutuhan

a. Analisis Kebutuhan Fungsional

Terdapat tiga alat penelitian dalam penyelesaian aplikasi diagnosa penyakit demam berdarah ini, yaitu :

1) *Smartphone Android*

Smartphone Android yang digunakan untuk menguji coba aplikasi ini, memiliki spesifikasi sebagai berikut :

- a) *CPU* : *Qualcomm MSM8909 1,6 GHz*
- b) *Memory Internal* : *1 GB RAM, 8 GB ROM*
- c) *Memory External* : *8 GB*
- d) *Operating System* : *Android OS, V5.0 (Lollipop)*
- e) *Tipe Layar* : *Corning Gorilla Glass 3 Multi Touch Screen*
- f) *Ukuran Layar* : *720 x 1280 pixel*

2) *Hardware*

Hardware yang digunakan memiliki spesifikasi sebagai berikut :

- a) *Processor* : *Intel® Core™ i3*
- b) *Memory* : *2 GB DDR 3*
- c) *Harddisk* : *500 GB*

- d) *Display* : 14 Inch WXGA (1366 x 768)
- e) *Sound Card* : Integrated
- f) *Video Type* : Intel®HD Graphics dan Nvidia G-Force GT 520M
- g) *Keyboard, Mouse, Speaker, Headset.*

3) *Software*

Software yang digunakan dalam pembuatan aplikasi diagnosa penyakit demam berdarah adalah :

- a) *Visual Studio .Net 2017*

b. Analisis Kebutuhan Non Fungsional (Untuk Pengguna)

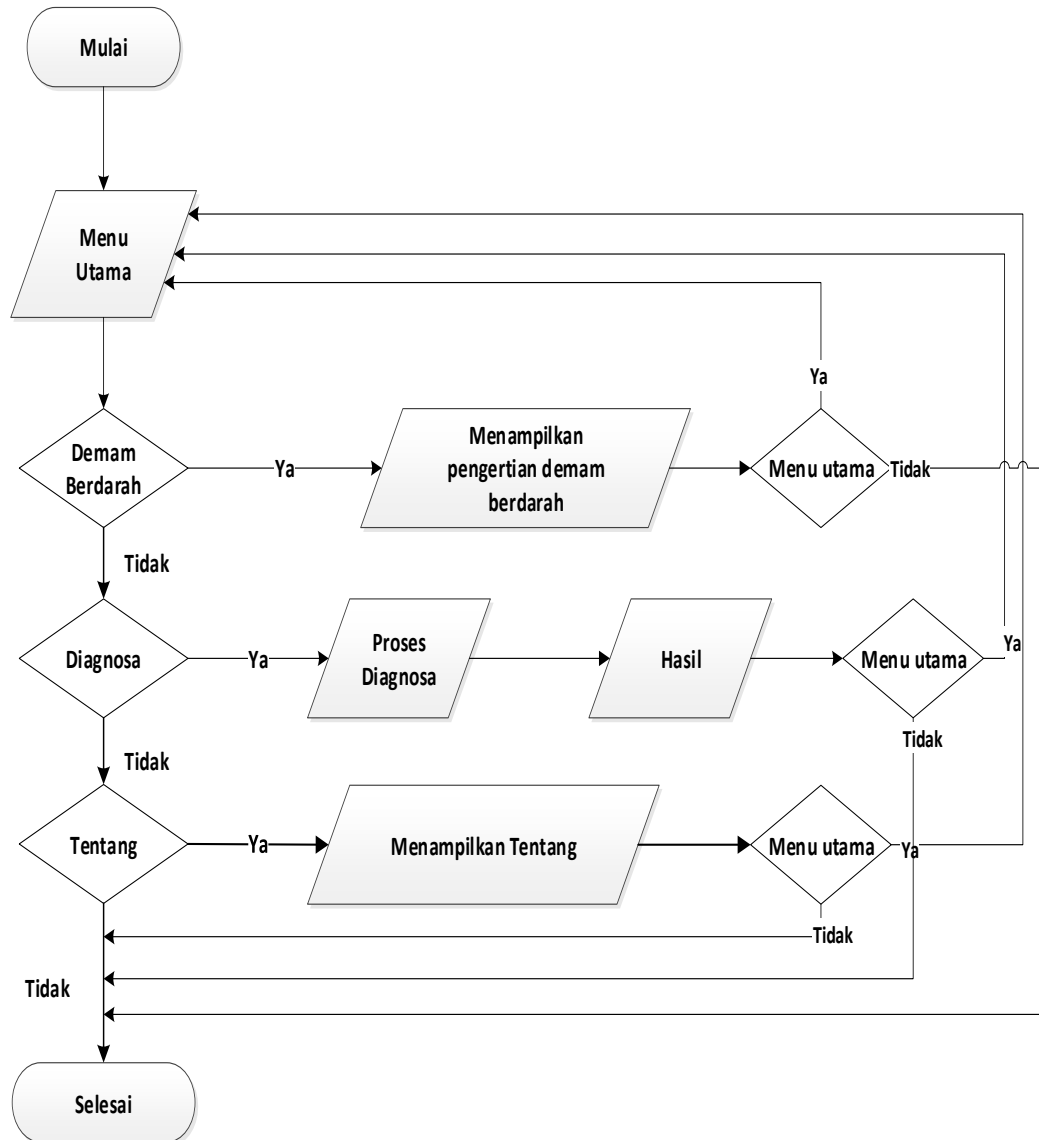
Spesifikasi minimum *hardware* yang dapat menjalankan aplikasi diagnosa penyakit demam berdarah ini adalah sebagai berikut :

1) *Smartphone Android*

Smartphone Android yang digunakan untuk menguji coba aplikasi ini, memiliki spesifikasi sebagai berikut :

- a) *CPU* : Dual-core 1.2 GHz Cortex-A9
- b) *Memory Internal* : 512 GB RAM, 4 GB ROM
- c) *Memory External* : 2 GB
- d) *Operating System* : Android OS, V5.0 (Lollipop)
- e) *Tipe Layar* : Corning Gorilla Glass 3 Multi Touch Screen
- f) *Ukuran Layar* : 840 x 480 pixel

Flowchart



Gambar 4. Flowchart Aplikasi Diagnosa Penyakit Demam Berdarah

4. Rule Aplikasi Diagnosa Penyakit Demam Berdarah

Berikut adalah deskripsi aturan pada aplikasi diagnosa penyakit demam berdarah:

- a. Gejala

Tabel 6. Data Gejala

| Kode | Gejala |
|-------------|---|
| G01 | Apakah anda mengalami demam? |
| G02 | Apakah anda merasa sakit kepala? |
| G03 | Apakah terasa nyeri pada otot dan tulang? |
| G04 | Apakah Mengalami mual-mual dan muntah? |
| G05 | Apakah ada rasa nyeri di belakang mata? |
| G06 | Apakah terjadi pendarahan kecil di kulit berupa bercak merah? |
| G07 | Apakah terjadi demam selama 2 sampai 7 hari? |
| G08 | Apakah mengalami mimisan atau pendarahan pada gusi? |
| G09 | Apakah terjadi pembesaran hati dan limpa? |
| G10 | Apakah terjadi peningkatan hematokrit? |
| G11 | Apakah anda sudah memeriksa trombosit ke dokter? |
| G12 | Apakah trombosit kurang dari 100.000? |
| G13 | Apakah kulit anda pucat? |
| G14 | Apakah tangan dan kaki menjadi dingin? |
| G15 | Apakah terjadi penurunan kesadaran? |
| G16 | Apakah ada tanda-tanda syok? |
| G17 | Apakah ada perubahan pada tekanan nadi? |

b. Hasil

Tabel 7. Data Hasil

| Kode | Hasil | Solusi |
|------|---|--|
| H01 | Demam Dengue (<i>Dengue Fever</i>) | Berikan anak banyak minum larutan oralit atau jus buah, air tajin, air sirup, susu, untuk mengganti cairan yang hilang akibat kebocoran plasma, demam, muntah/diare. |
| H02 | Demam Berdarah Dengue (<i>Dengue Hemorrhagic Fever</i>) | Banyak beristirahat. Minum banyak cairan untuk mencegah dehidrasi (terutama untuk mengganti cairan tubuh yang terbuang akibat gejala demam tinggi dan muntah-muntah). Mengonsumsi parasetamol dan acetaminophen untuk meredakan demam dan nyeri. Berhenti menjalani aktivitas untuk sementara |

| | | |
|-----|---|--|
| | | waktu sampai tubuh benar-benar pulih. |
| H03 | Sindroma Syok Dengue (<i>Dengue Shock Syndrome</i>) | disarankan minum sebanyak – banyaknya dan sesering mungkin serta memberikan antipiretika dan “surface cooling” |

c. Aturan

R1 = If G01 and G02 and G03 and G04 and G05 and G06 then H01

R2 = If G07 and G08 and G09 and G10 and G11 and G12 then H02

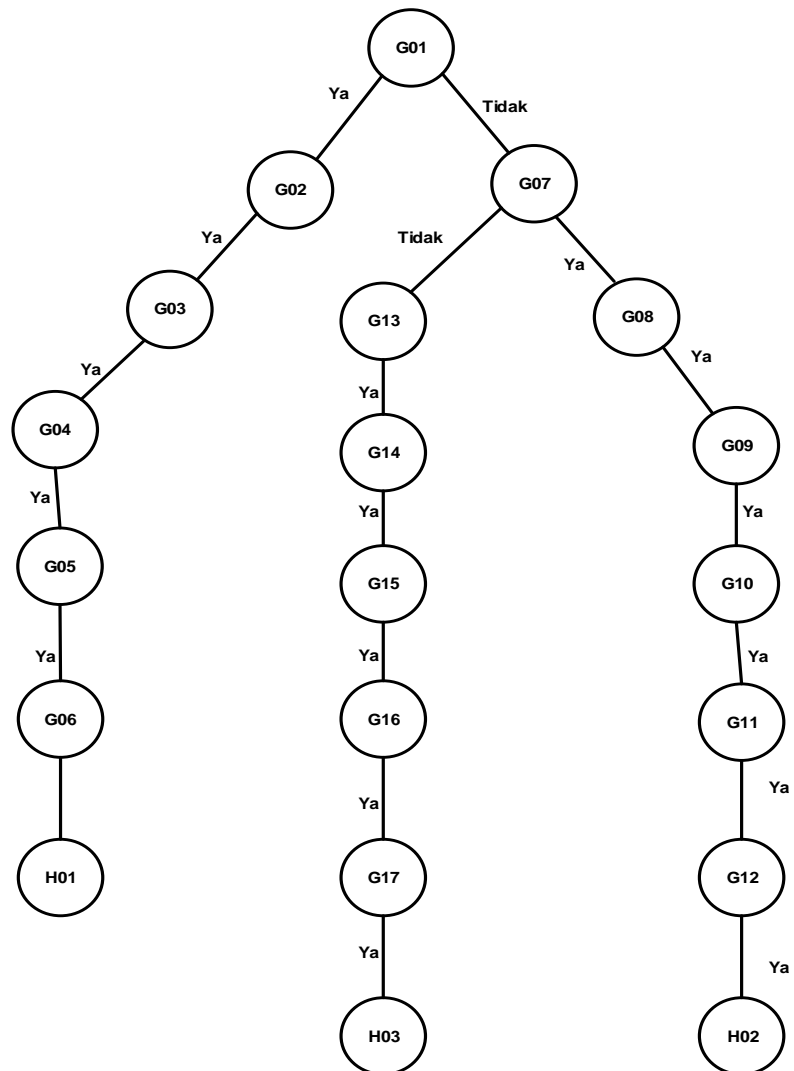
R3 = If G13 and G14 and G15 and G16 and G17 then H03

Tabel 8. Aturan

| Kode | Gejala | Kode | Hasil |
|------|---|------|--------------------------------------|
| G01 | Apakah anda mengalami demam? | H01 | Demam Dengue (<i>Dengue Fever</i>) |
| G02 | Apakah anda merasa sakit kepala? | | |
| G03 | Apakah terasa nyeri pada otot dan tulang? | | |
| G04 | Apakah Mengalami mual-mual dan muntah? | | |
| G05 | Apakah ada rasa nyeri di belakang mata? | | |
| G06 | Apakah terjadi pendarahan kecil di kulit | | |

| | | | |
|-----|---|-----|---|
| | berupa bercak merah? | | |
| G07 | Apakah terjadi demam selama 2 sampai 7 hari? | H02 | Demam Berdarah Dengue (<i>Dengue Hemorrhagic Fever</i>) |
| G08 | Apakah mengalami mimisan atau pendarahan pada gusi? | | |
| G09 | Apakah terjadi pembesaran hati dan limpa? | | |
| G10 | Apakah terjadi peningkatan hematokrit? | | |
| G11 | Apakah anda sudah memeriksa trombosit ke dokter? | | |
| G12 | Apakah trombosit kurang dari 100.000? | | |
| G13 | Apakah kulit anda pucat? | H03 | Sindroma Syok Dengue (<i>Dengue Shock Syndrome</i>) |
| G14 | Apakah tangan dan kaki menjadi dingin? | | |
| G15 | Apakah terjadi penurunan kesadaran? | | |
| G16 | Apakah ada tanda-tanda syok? | | |
| G17 | Apakah ada perubahan pada tekanan nadi? | | |

d. Pohon Keputusan

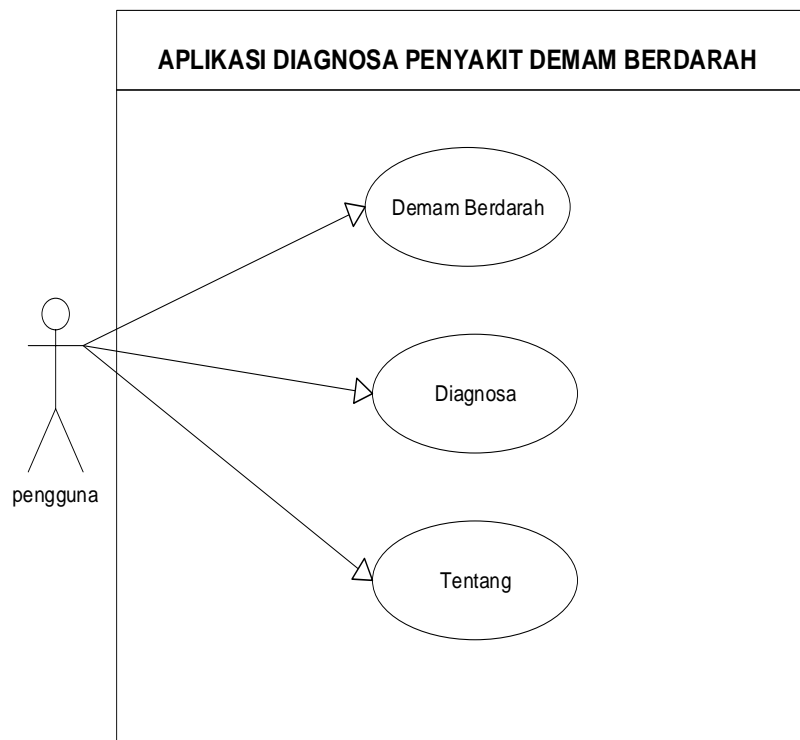


Gambar 5. Pohon Keputusan

5. Perancangan Sistem

a. Use Case Diagram

Untuk mendapatkan informasi dari sebuah sistem yang dibuat, maka penulis menggunakan *use case diagram*. Dengan diagram ini, proses yang terjadi pada sebuah aplikasi akan dapat diketahui. *Use case diagram* dari aplikasi diagnosa penyakit demam berdarah dapat dilihat dibawah ini :



Gambar 6. Use Case Diagram Aplikasi Diagnosa Penyakit Demam Berdarah

b. Use Case

1. Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor pada aplikasi diagnosa penyakit demam berdarah :

Tabel 9. Definisi Aktor

| Aktor | Deskripsi |
|----------|---|
| Pengguna | Orang yang menggunakan aplikasi diagnosa penyakit demam berdarah. |

2. Definisi *Use Case*

Berikut adalah deskripsi pendefinisian *Use case* pada aplikasi diagnosa penyakit demam berdarah :

Tabel 10. Definisi *Use Case*

| No | <i>Use case</i> | Deskripsi |
|----|-----------------|---|
| 1. | Demam Berdarah | Merupakan menu yang berisi tentang pengertian tentang penyakit demam berdarah yang sering menyerang. |
| 2. | Diagnosa | Merupakan menu yang berisi proses diagnosa pengguna apakah terserang penyakit demam berdarah atau tidak dengan pertanyaan-pertanyaan. |
| 3. | Tentang | Merupakan menu yang berisi tentang pembuat aplikasi diagnosa penyakit demam berdarah |

c. Skenario *Use case*

Berikut adalah skenario jalannya masing-masing *use case* yang telah didefinisikan sebelumnya :

i. Skenario *Use Case* Demam Berdarah

Nama *Use case* : Demam Berdarah

Skenario :

Tabel 11. Skenario *Use Case* Demam Berdarah

| Aksi Aktor | Reaksi Sistem |
|--------------------------------|---|
| Skenario Normal | |
| 1. Memilih menu demam berdarah | |
| | 2. Menampilkan pengertian dasar tentang penyakit demam berdarah |

ii. Skenario *Use case* Diagnosa

Nama *Use case* : Diagnosa

Skenario :

Tabel 12. Skenario *Use Case* Diagnosa

| Aksi Aktor | Reaksi Sistem |
|--------------------------|--|
| Skenario Normal | |
| 1. Memilih menu diagnosa | |
| | 2. Menampilkan menu diagnosa apakah pengguna penyakit demam berdarah atau tidak. |

iii. Skenario *Use case* Tentang

Nama *Use case* : Tentang

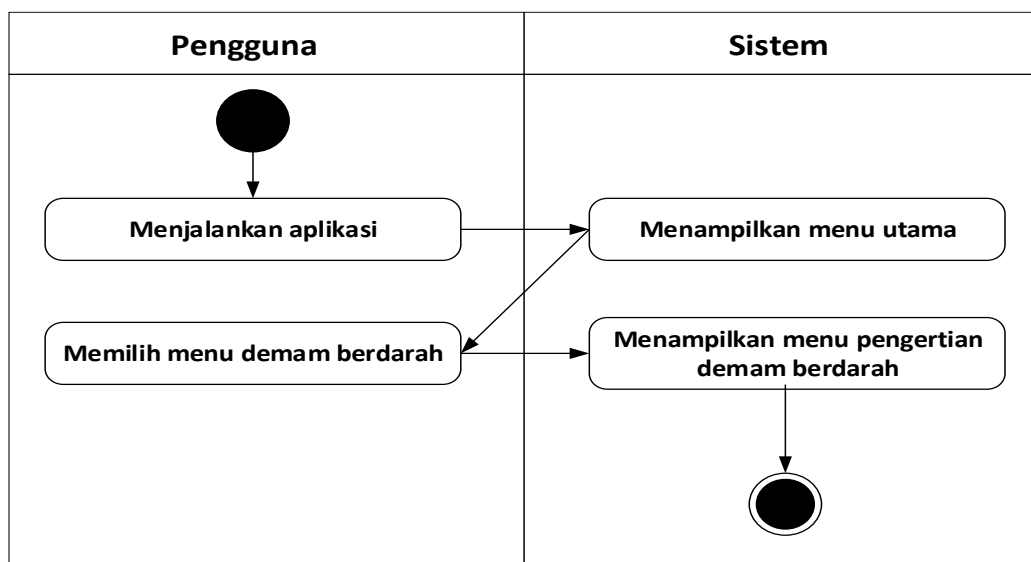
Skenario :

Tabel 13. Skenario *Use Case* Tentang

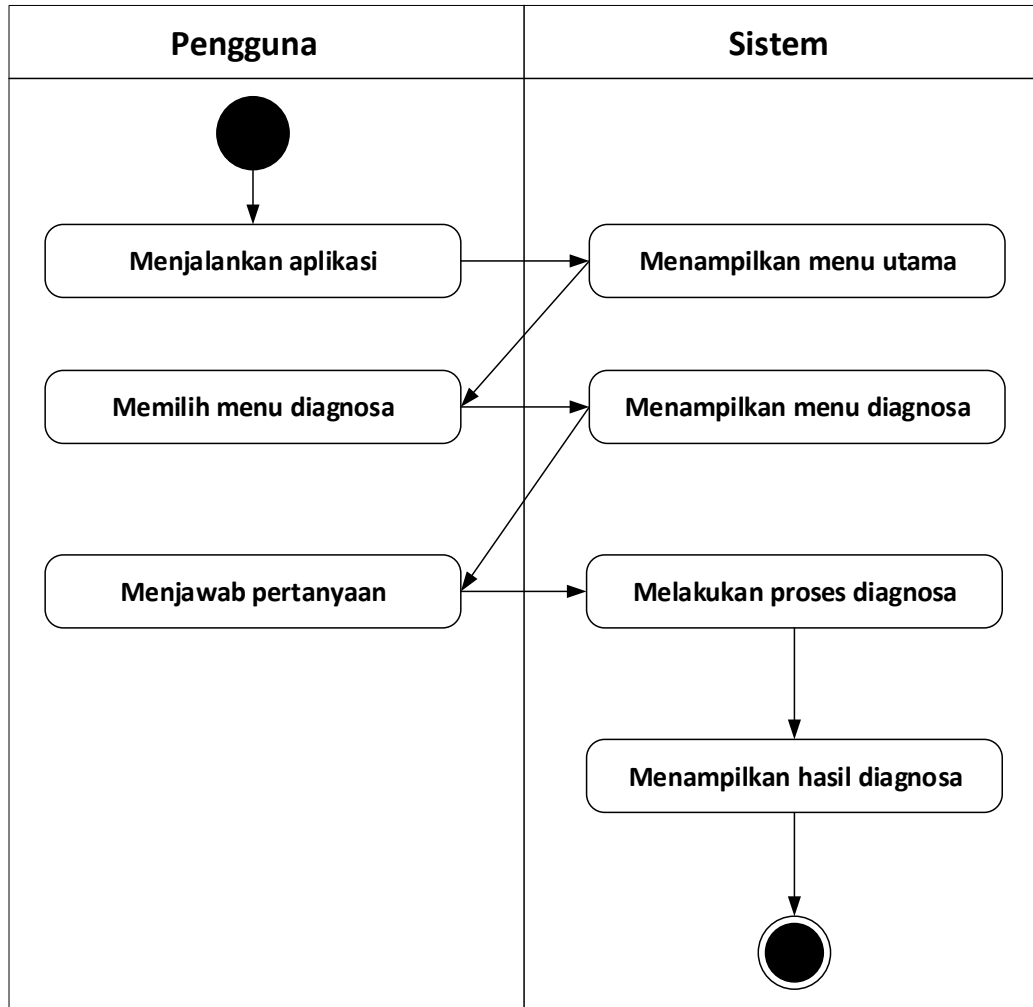
| Aksi Aktor | Reaksi Sistem |
|-------------------------|--|
| Skenario Normal | |
| 1. Memilih menu tentang | |
| | 2. Menampilkan menu yang berisi tentang pembuat aplikasi diagnosa penyakit demam berdarah. |

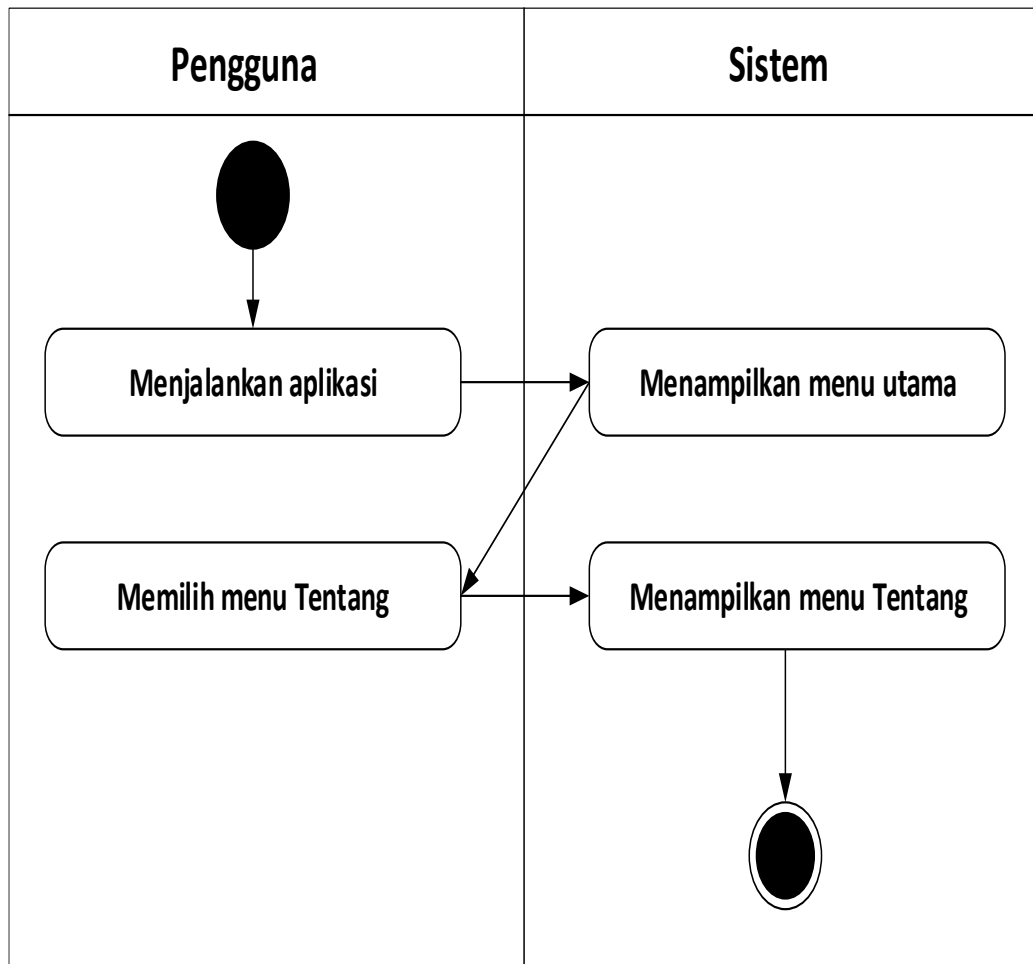
6. Activity Diagram

a. *Activity Diagram* Demam Berdarah



Gambar 7. Activity Diagram Demam Berdarah

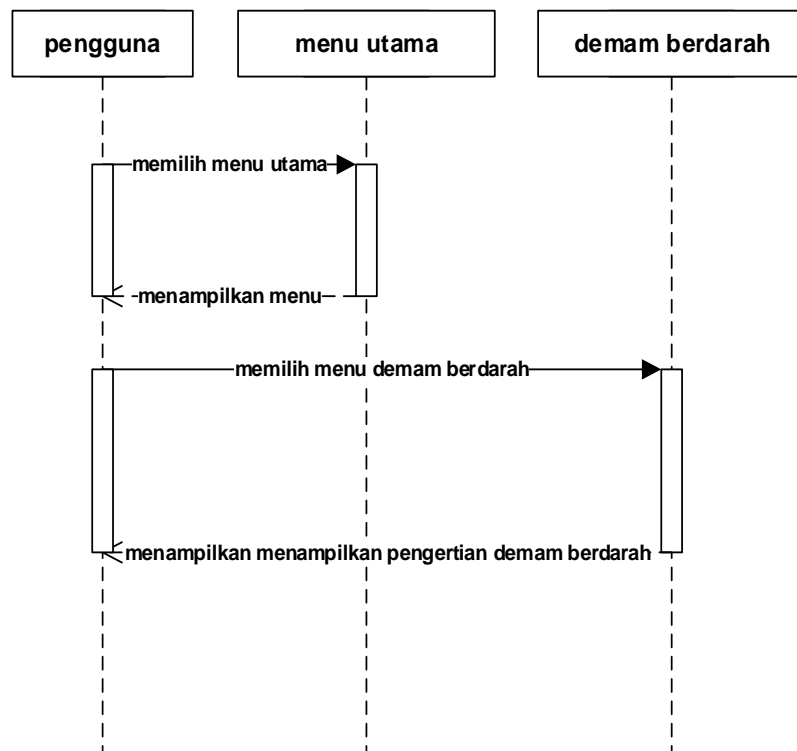
b. *Activity Diagram* DiagnosaGambar 8. *Activity Diagram* Diagnosa

c. *Activity Diagram Tentang***Gambar 9. Activity Diagram Tentang**

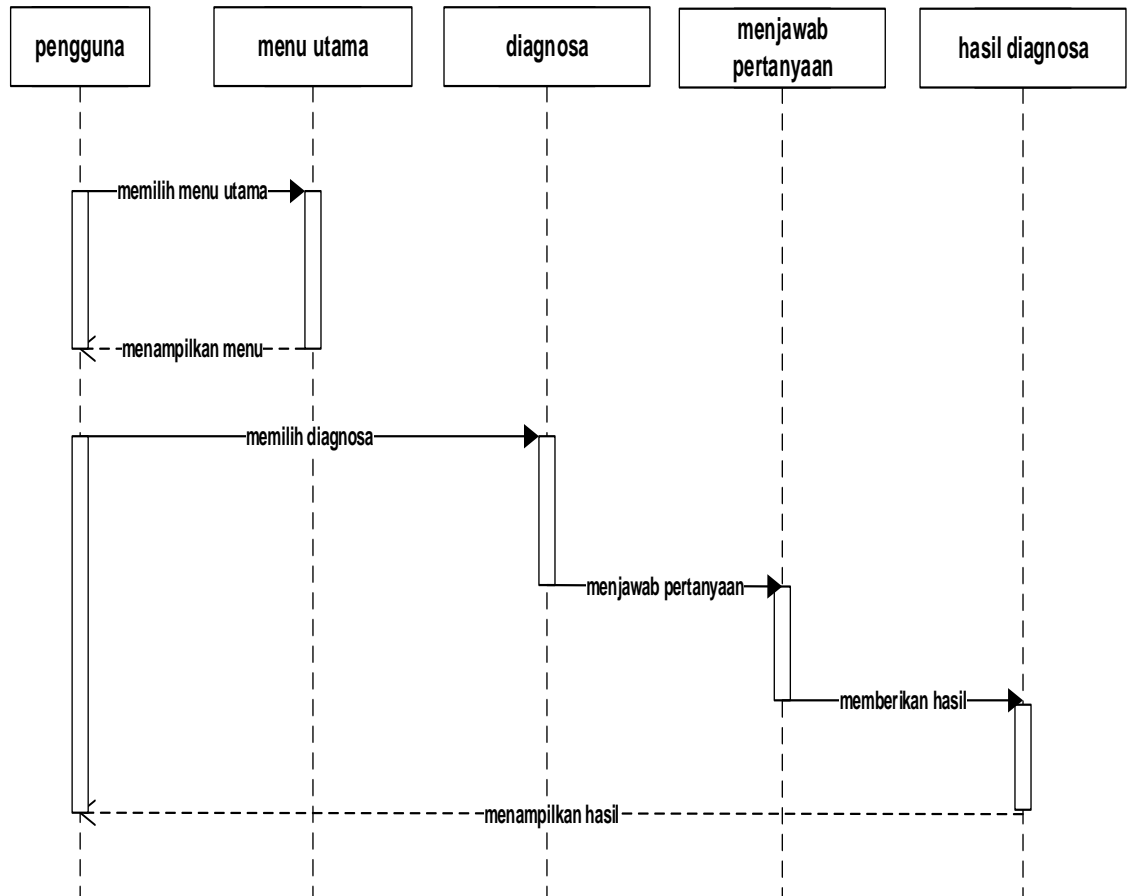
7. Diagram Sequence

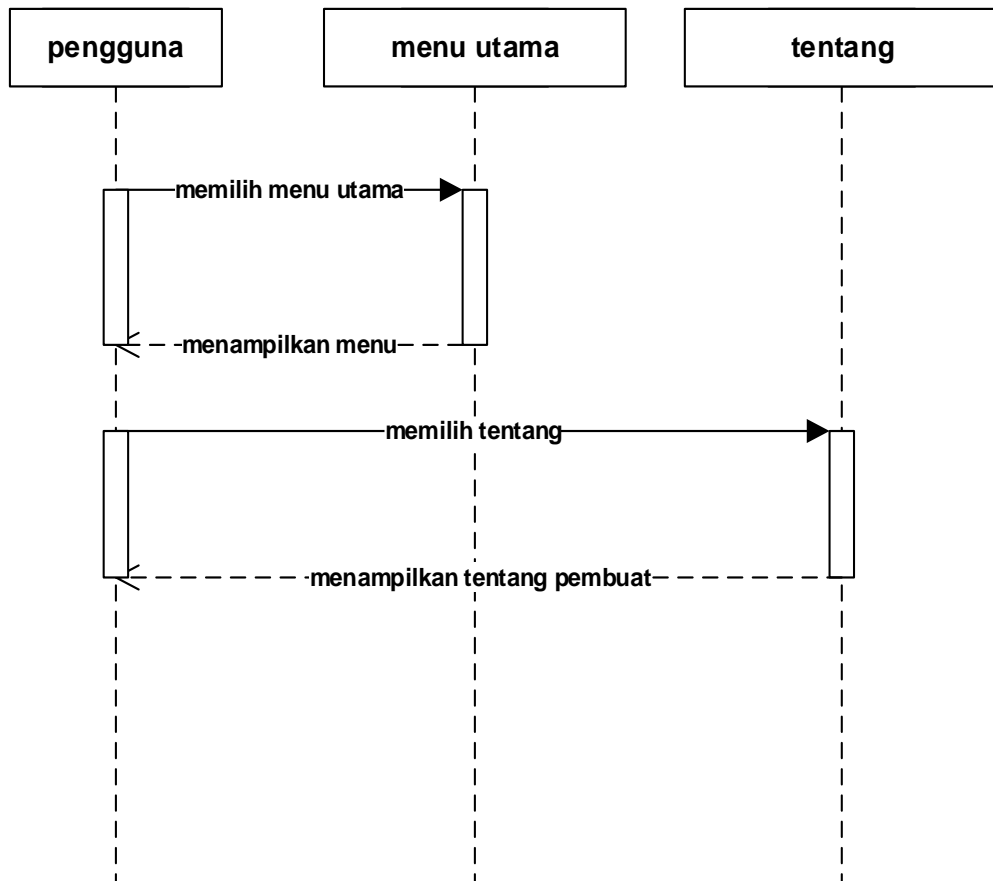
Berikut adalah *diagram sequence* aplikasi diagnosa penyakit demam berdarah:

a) *Diagram Sequence* Demam Berdarah



Gambar 10. *Diagram Sequence* Demam Berdarah

b) *Diagram Sequence Diagnosa***Gambar 11.** *Diagram Sequence Diagnosa*

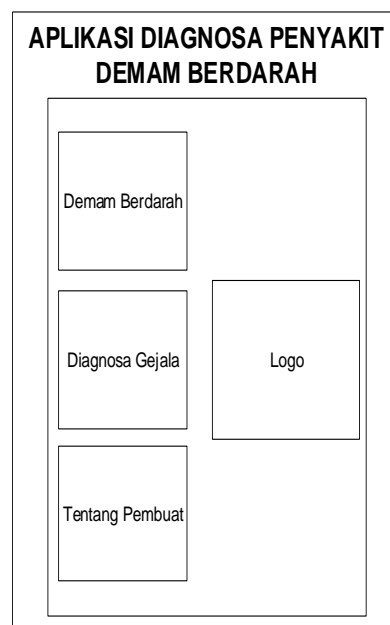
c) *Diagram Sequence Tentang***Gambar 12. *Diagram Sequence Tentang*****8. Perancangan Aplikasi**a. Perancangan Antarmuka (*User Interface*)

Perancangan antar muka adalah rancangan yang dilakukan untuk memberikan gambaran aplikasi yang akan ditampilkan secara sederhana kepada pengguna. Diharapkan pengguna yang menggunakan aplikasi ini dapat dengan

mudah mengerti fungsi dari tombol yang ada pada aplikasi. Dalam aplikasi diagnosa penyakit demam berdarah ini, terdapat beberapa bagian tampilan yang memiliki fungsi berbeda pada setiap tombolnya. Fungsi – fungsi dari tombol yang ada pada setiap bagian tampilan akan dijelaskan dan dapat dilihat pada gambar berikut :

1) Rancangan Tampilan Menu Utama

Rancangan tampilan menu utama adalah tampilan yang pertama kali ditampilkan dan memiliki beberapa fungsi untuk menghubungkan ke tampilan lainnya. Tampilan ini disebut dengan tampilan utama, yang dapat digunakan oleh pengguna untuk menuju ke tampilan yang diinginkannya dengan memilih menu yang ada pada tampilan. Menu utama memiliki teks judul dan 3 tombol.



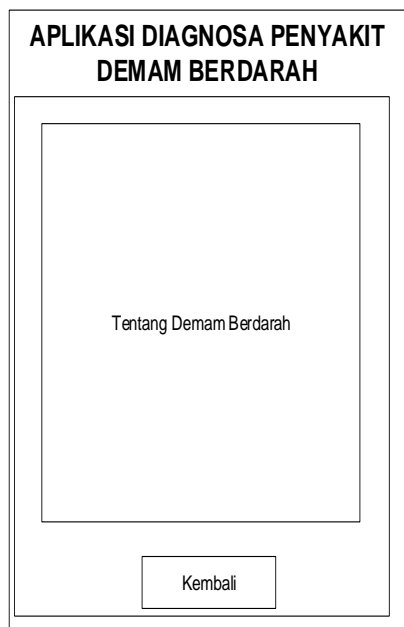
Gambar 13. Rancangan Tampilan Menu Utama

Berikut fungsi dari 3 tombol yang ada pada menu utama :

- a) Demam berdarah berfungsi untuk menuju ke tampilan menu pengertian dasar tentang penyakit demam berdarah.
- b) Diagnosa berfungsi untuk menuju ke tampilan proses diagnosa apakah pengguna terserang penyakit demam berdarah atau tidak.
- c) Tentang pembuat berfungsi untuk menuju ke tampilan tentang.

2) Rancangan Tampilan *Form* Demam Berdarah

Rancangan tampilan demam berdarah adalah tampilan yang berisikan pengertian dasar tentang penyakit demam berdarah. Tampilan demam berdarah memiliki teks dan 1 tombol.



Gambar 14 Rancangan Tampilan Demam Berdarah

3) Rancangan Tampilan Diagnosa

Rancangan tampilan diagnosa adalah tampilan yang berisikan pertanyaan yang harus dijawab oleh pengguna agar dapat mendiagnosa apakah pengguna penyakit demam berdarah atau tidak. diagnosa memiliki teks pertanyaan dan 2 tombol.

The image shows a wireframe for a diagnostic application titled "APLIKASI DIAGNOSA PENYAKIT DEMAM BERDARAH". The interface consists of a central text box containing the question "Pertanyaan tentang gejala yang dialami". Below this question are two buttons: "Ya" and "Tidak", which serve as response options for the user.

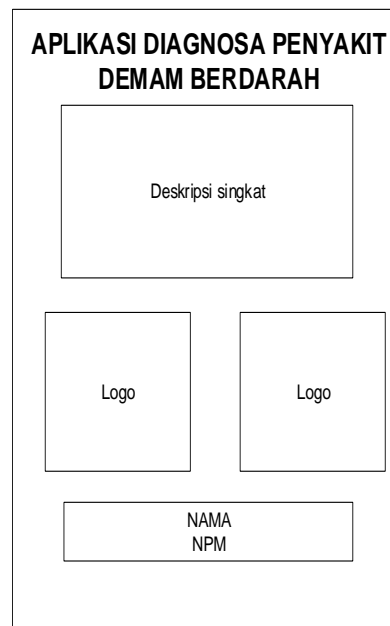
Gambar 15 Rancangan Tampilan Diagnosa

Berikut fungsi dari 2 tombol yang ada pada diagnosa :

- a. Tombol Ya berfungsi untuk memilih pertanyaan apakah benar.
- b. Tombol Tidak berfungsi untuk memilih pertanyaan apakah tidak benar.

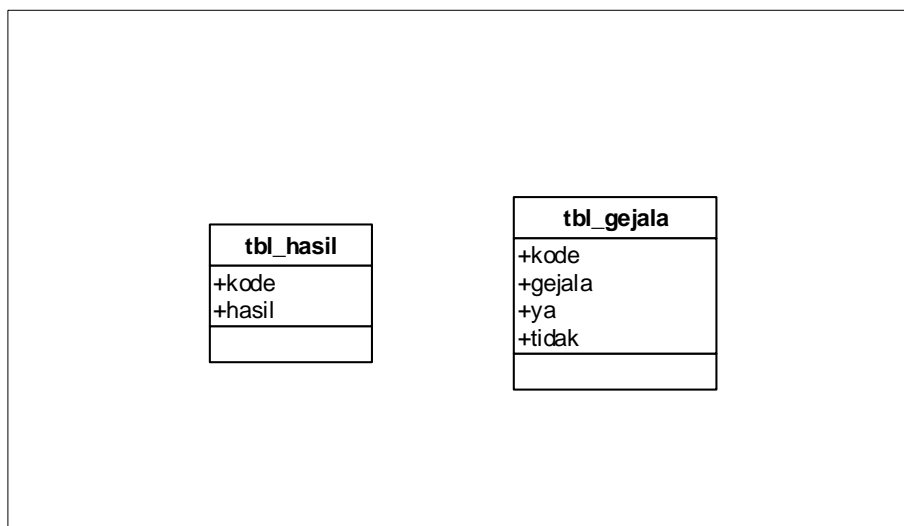
4) Rancangan Tampilan Menu Tentang

Rancangan ini adalah rancangan yang menampilkan informasi dari si pembuat aplikasi diagnosa penyakit demam berdarah.



Gambar 16. Rancangan Tampilan Menu Tentang

9. Perancangan *Class Diagram*



Gambar 17. Class Diagram Aplikasi Diagnosa Penyakit Demam Berdarah

10. Perancangan *Database*

a. Perancangan Tabel

Struktur file digunakan dalam perancangan sistem untuk menentukan nilai atau tipe data suatu atribut pada file yang terdapat pada database. Pada tahapan perancangan struktur file untuk mempermudah dalam mengetahui suatu nilai atau tipe data yang ada pada file penyimpanan ini akan dijelaskan mengenai perancangan basis data yang akan digunakan. Penyusunan table ini pada dasarnya digunakan untuk memudahkan dalam pemasukan dengan penyimpanan data yang sesuai dengan kelompok dari data atau informasi tersebut.

1. Tabel Gejala

Tabel 14. Gejala

| Nama field | Type data | Size | Keterangan |
|-------------------|------------------|-------------|-------------------|
| Kode | Varchar | 10 | Primary Key |
| Gejala | Varchar | 255 | - |
| Ya | Varchar | 100 | - |
| Tidak | Varchar | 20 | - |

2. Tabel Hasil

Tabel 15. Hasil

| Nama_field | Type data | Size | Keterangan |
|-------------------|------------------|-------------|-------------------|
| Kode | Varchar | 10 | Primary key |
| Hasil | Varchar | 100 | Foreign Key |

BAB IV

IMPLEMENTASI SISTEM

1. Implementasi Sistem Yang Digunakan

Tahapan implementasi yang dilakukan untuk menyelesaikan perancangan sistem pakar diagnosa penyakit demam berdarah ini diperlukan informasi mengenai penyediaan perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

Berikut disediakan perangkat keras dan perangkat lunak yang dibutuhkan.

a. Spesifikasi Perangkat Keras

Aplikasi sistem pakar diagnosa penyakit demam berdarah ini, telah diuji pada smartphone dengan spesifikasi perangkat keras sebagai berikut:

- 1) *CPU* : *Qualcomm MSM8909 1,2 GHz, GPU*
- 2) *Memory Internal* : 1 GB RAM, 8 GB ROM
- 3) *Memory External* : 8 GB
- 4) *Operating System* : Android OS, 5.0 (Lollipop)
- 5) Tipe Layar : *Corning Gorilla Glass 3 Multi Touch Screen*
- 6) Ukuran Layar : 480 x 840 *pixel*

b. Spesifikasi Perangkat Lunak

Aplikasi ini dijalankan pada perangkat lunak dengan spesifikasi sebagai berikut:

- 1) Sistem Operasi : *Android OS, 5.0 (Lollipop)*

2. Tampilan Aplikasi Sistem Pakar Diagnosa Penyakit Demam berdarah

Tampilan aplikasi sistem pakar diagnosa penyakit demam berdarah ini terdiri dari tampilan menu utama, demam berdarah, diagnosa dan data user. Menu utama berisi menu - menu aplikasi yaitu menu utama, demam berdarah, diagnosa, dan tentang.

Adapun tampilan menu-menu aplikasi sistem pakar diagnosa penyakit demam berdarah adalah sebagai berikut :

a. Tampilan Menu Utama

Tampilan menu utama terdiri dari demam berdarah, diagnosa, dan tentang.



Gambar 18. Tampilan Menu Utama

Berikut fungsi dari 3 tombol yang ada pada menu utama :

- 1) Tombol diagnosa berfungsi untuk melakukan sistem pakar diagnosa penyakit demam berdarah.
- 2) Tombol pengertian demam berdarah berfungsi untuk masuk ke menu pengertian tentang demam berdarah untuk pemahaman lebih lanjut.
- 3) Tombol tentang berfungsi untuk masuk ke menu tentang pembuat aplikasi diagnosa penyakit demam berdarah.

b. Tampilan Demam Berdarah

Tampilan demam berdarah memiliki fungsi untuk melihat tentang penyakit demam berdarah untuk memberikan pengetahuan tentang penyakit tersebut.



Gambar 19. Tampilan Pengertian Demam Berdarah

c. Tampilan Diagnosa

Tampilan diagnosa memiliki fungsi untuk diagnosa gejala yang terjadi pada pengguna dengan memberikan pertanyaan dan user yang menggunakannya harus menjawab pertanyaan yang diberikan sehingga aplikasi akan memberikan hasil dari diagnosa penyakit yang menyerang pengguna tersebut.



Gambar 20. Tampilan Diagnosa

d. Tampilan Hasil Diagnosa

Tampilan hasil diagnosa memiliki fungsi sebagai memberikan hasil diagnosa pengguna berdasarkan gejala pertanyaan yang diberikan oleh aplikasi.



Gambar 21. Tampilan Hasil Diagnosa

e. Tampilan Tentang

Tampilan tentang berfungsi untuk melihat profil pembuat aplikasi diagnosa penyakit demam berdarah ini.



Gambar 22. Tampilan Tentang

3. Pengujian Aplikasi Sistem Pakar Diagnosa Penyakit Demam Berdarah

Pengujian aplikasi sistem pakar diagnosa penyakit demam berdarah digunakan untuk menguji sistem pada salah satu menu dimana data yang digunakan adalah proses menjawab pertanyaan seputar gejala yang terjadi pada pengguna tersebut. Cara menggunakan aplikasi sistem pakar diagnosa penyakit demam berdarah adalah sebagai berikut :

- a. Langkah awalnya pengguna menjalankan aplikasi dan akan ditampilkan menu utama aplikasi.

- b. Kemudian pengguna memilih menu diagnosa yang ada pada aplikasi.
- c. Lalu akan tampil menu tentang pertanyaan gejala seputar gejala yang terjadi pada pengguna tersebut.



Gambar 23. Diagnosa

- d. Setelah pengguna selesai dalam menjawab pertanyaan yang diberikan oleh aplikasi, maka akan ditampilkan tampilan baru yaitu tampilan hasil.



Gambar 24. Hasil Diagnosa

4. Evaluasi

Aplikasi sistem pakar diagnosa penyakit demam berdarah memiliki kelebihan dan kekurangan sebagai berikut :

a. Kelebihan

- 1) Memberikan solusi pada orang yang menderita penyakit demam berdarah agar lebih cepat dalam penanganannya.

2) Penggunaan aplikasi tidak membutuhkan akses jaringan internet untuk dapat menjalankannya.

b. Kelemahan

- 1) Aplikasi hanya dapat mendeteksi penyakit demam berdarah yang umum.
- 2) Tidak dapat mengupdate database pada aplikasi karena database sudah menjadi satu pada aplikasi yang telah dirancang.

BAB V

PENUTUP

1. Kesimpulan

Berdasarkan hasil pembuatan aplikasi sistem pakar diagnosa penyakit demam berdarah, maka didapat beberapa kesimpulan seperti berikut:

- a. Aplikasi dapat memberikan hasil penyakit demam berdarah yang diderita pengguna dengan cara menjawab pertanyaan yang diberikan oleh aplikasi secara mudah.
- b. Aplikasi dapat dijalankan dengan mudah menggunakan smartphome android yang banyak dipakai oleh masyarakat luas.

2. Saran

Berikut adalah saran dari penulis agar aplikasi sistem pakar diagnosa penyakit demam berdarah dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

- a. Penyakit demam berdarah yang terdapat pada aplikasi adalah penyakit demam berdarah yang dapat dideteksi dengan menggunakan gejala diagnosa pertanyaan yang diberikan oleh aplikasi.
- b. Aplikasi yang dirancang masih menggunakan metode forward chaining dalam mendiagnosa penyakit demam berdarah yang diderita sehingga tidak dapat menampilkan nilai dari hasil gejala yang dialami.

DAFTAR PUSTAKA

- Abdillah, Andretti, 2006, Perancangan Basisdata Sistem Informasi Penggajian (Studi Kasus pada Universitas 'XYZ'), Palembang.
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Baharuddin, 2018, Efektivitas Ekstrak Dahan Kelor Terhadap Mortalitas *Larva Aedes aegypti*.
- Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).
- Denis Hadi, 2015, Aplikasi Penilaian Kinerja Guru Berdasarkan Feedback Dari Siswa Berbasis Web (Studi Kasus : Smp Negeri 2 Dayeuhkolot).
- Fadhilah, 2012, Perancangan Aplikasi Sistem Pakar Penyakit Kulit Pada Anak Dengan Metode *Expert System Development Life Cycle*, Garut.
- Fanun, 2013, Informasi Kampus Berbasis Web Pada Android, Yogyakarta.
- Fuad, R. N., & Winata, H. N. (2017). Aplikasi Keamanan File Audio Wav (Waveform) Dengan Terapan Algoritma RSA. InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan, 1(2), 113-119.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), 1(1).
- Harto, 2013; Perancangan Sistem Pakar Untuk Mengidentifikasi Penyakit Pada Tanaman Semangka Dengan Menggunakan Metode *Certainty Factor*, Medan.
- Haviluddin, 2011; Memahami Penggunaan UML (*Unified Modelling Language*), Samarinda.
- Indrawaty, 2011, Implementasi Metode Simple Additive Weighting Pada Sistem Pengambilan Keputusan Sertifikasi Guru, Bandung.

- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- Irmayani, 2015, Sistem Pakar Penelusuran Kecerdasan Pada Anak Dengan Menggunakan Metode *Forward Chaining*, Medan.
- Ismoedijanto, 2000, Demam pada Anak, Surabaya.
- Jubilee Enterprise, 2017, Belajar VB, Visual C#, Dan Python Menggunakan Visual Studio, Jakarta, Penerbit : PT Elex Media Komputindo.
- Kementerian Kesehatan Republik Indonesia, 2018, Demam Berdarah Dengue (DBD), www.kemkes.go.id.
- Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). Implementasi Augmented Reality Sebagai Media Promosi Penjualan Rumah. In Seminar Nasional Royal (Senar) (Vol. 1, No. 1, pp. 429-434).
- Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. *Jurnal Teknik dan Informatika*, 5(2), 13-19.
- Ladjamudin, A.-B. Bin. (2013). *Analisis Dan Desain Sistem Informasi*.
- Lars, 2015, *Microsoft Visual Studio 2015 Unleashed, 3rd Edition, United States Of America*.
- Maulana, 2017, Pengembangan Aplikasi Android Untuk Studi Bahasa Carakan Madura, Surabaya
- Nadesul, Hendrawan, 2016, Demam Berdarah Dan Virus Zika, Jakarta. Penerbit :
- Nirmala, 2014, Sistem Pakar Diagnosa Awal Penyakit Kulit Pada Sapi Bali dengan Menggunakan Metode *Forward chaining* dan *Certainty Factor*, Denpasar.
- Noer, 2017, Aplikasi Perpustakaan Smk Siliwangi Ams Banjarsari Berbasis Android.
PT. Kompas Media Nusantara.
- Rahim, R. (2018, October). A Novelty Once Methode Power System Policies Based On SCS (Solar Cell System). In International Conference of ASEAN Prespective and Policy (ICAP) (Vol. 1, No. 1, pp. 195-198).
- Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., ... & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for

- Selecting Best Employees. In Journal of Physics: Conference Series (Vol. 1028, No. 1, p. 012052). IOP Publishing.
- Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. *Int. J. Sci. Res. Sci. Technol*, 3(6), 470-473.
- Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.
- Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.
- Sitorus, Z. (2018). Kebutuhan Web Service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas. *Jurnal Teknik dan Informatika*, 5(2), 87-90.
- Sitorus, Z., Saputra, K, S., Sulistianingsih, I. (2018) C4.5 Algorithm Modeling For Decision Tree Classification Process Against Status UKM.
- Surahman, 2017, Aplikasi Mobile Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan, Bandung.
- Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 100-109.
- Urva, 2015, pemodelan UML E-Marketing Minyak Goreng, Kisaran.
- Verina, 2015; Penerapan Metode *Forward Chaining* untuk Mendeteksi Penyakit THT.
- Yesputra, R. (2015). Sistem Informasi Pengolahan Data Periklanan Menggunakan Microsoft Visual Basic 6 . 0, 180–185.
- Yulansari, 2013, Sistem Informasi Pengolahan Data Iuran Badan Pembantu Penyelenggaraan Pendidikan Sekolah Menengah Kejuruan Negeri 2 Donorojo.