



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODE DAN PERANCANGAN APLIKASI

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini antara lain sebagai berikut.

1. Identifikasi Masalah

Mengidentifikasi gejala pada pasien yang kemungkinan terkena penyakit tifus dengan bantuan dari pakar.

2. Studi Literatur

Melakukan studi terhadap contoh dan referensi yang berhubungan dengan penelitian ini, seperti penelitian yang menggunakan metode yang sama yang sudah dilakukan sebelumnya, jurnal, buku, *website*, dan lain-lain.

3. Pengumpulan Data

Pengumpulan data akan dilakukan dengan metode wawancara dengan pakar.

4. Perancangan dan Analisis Program

Perancangan dibuat dengan membuat diagram alir sistem, data flow diagram, dan *design interface*.

5. Pembangunan Aplikasi

Aplikasi berbasis *web* ini dibangun menggunakan Notepad++ dengan bahasa pemrograman PHP, HTML dan *database* MySQL.

## 6. Analisis dan Pengambilan Kesimpulan

Akan dilakukan uji coba atau *testing* terhadap aplikasi bertujuan untuk mencari bug atau kekurangan dari program dan akan mencari beberapa sukarelawan untuk berperan sebagai pengguna dari uji coba tersebut.

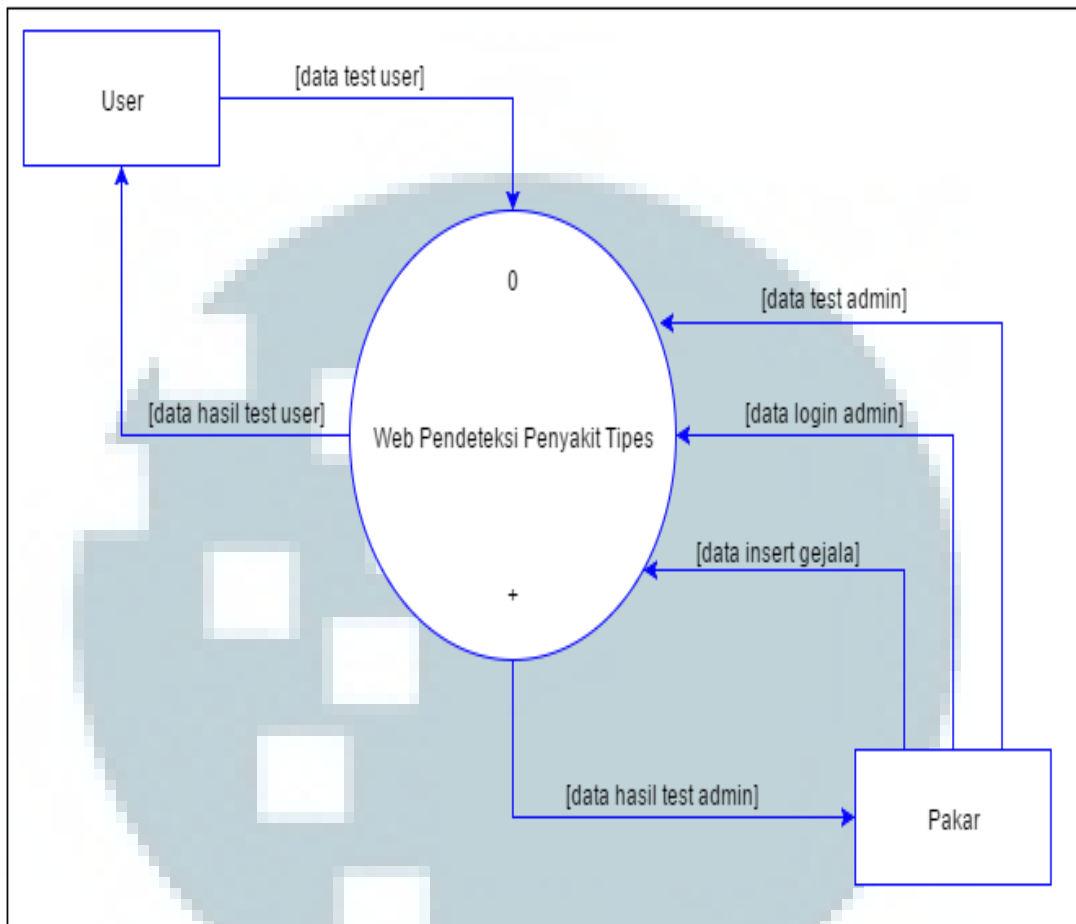
### 3.2 Perancangan Aplikasi

Dalam membuat aplikasi ini, digunakanlah diagram alir (*Flowchart Diagram*), *Entity Relationship Diagram*, dan *Data Flow Diagram*. Diagram alir adalah suatu tipe diagram yang melambangkan suatu proses menggunakan simbol-simbol yang berbeda yang mengandung informasi mengenai langkah-langkah dari suatu prosedur (Janssen, 2010). *Entity Relationship Diagram* adalah suatu model data yang secara grafis mengilustrasikan entitas dari suatu sistem informasi dan hubungan antar entitas tersebut, sementara *Data Flow Diagram* adalah representasi dari alur dan pertukaran informasi di dalam sistem (Janssen, 2010).

Rancangan pada aplikasi sistem pakar untuk mendeteksi penyakit tifus berbasis *web* sebagai berikut.

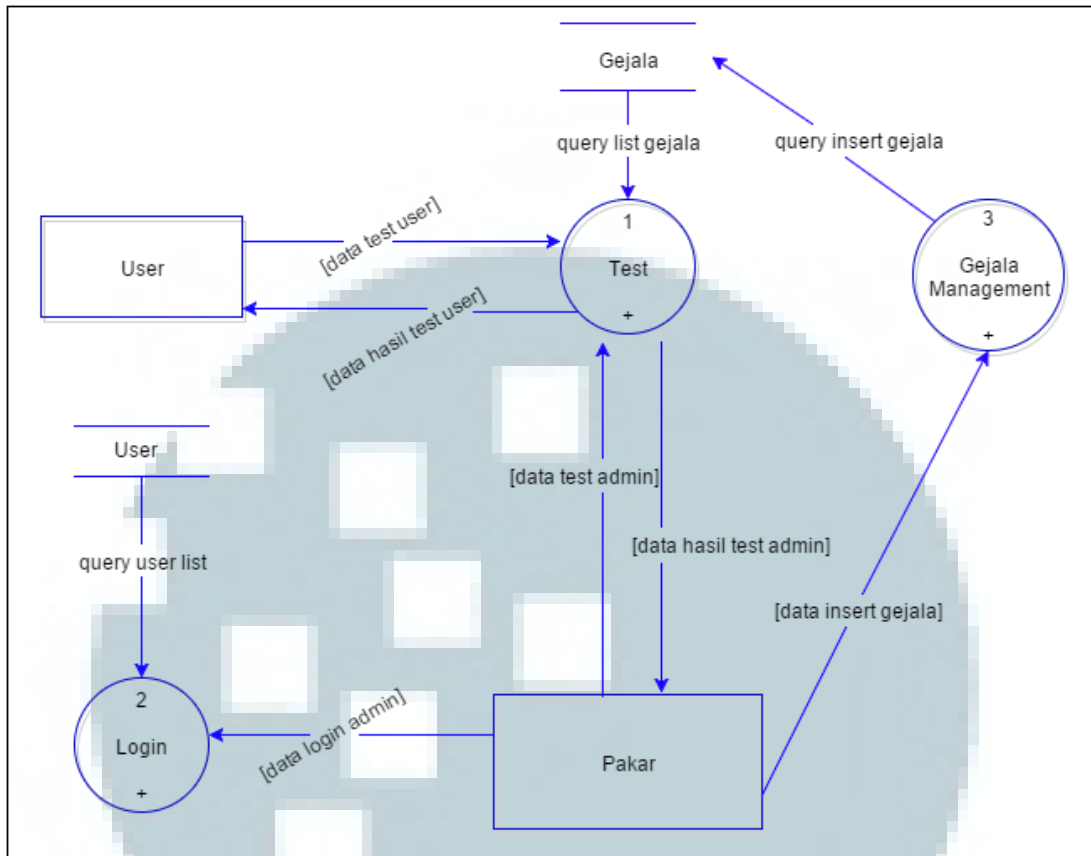
1. *Data flow diagram*
2. Diagram alir
3. Struktur tabel
4. *Design interface*

### 3.2.1 Data Flow Diagram



Gambar 3.1 Context Diagram

*Context diagram* untuk aplikasi ini terdiri dari dua entitas eksternal, yaitu pakar dan *user*. Berdasarkan pada gambar 3.1, *context diagram* ini dipecah menjadi tiga proses pada DFD *Level 1*, yaitu proses *test*, proses *login*, dan proses penambahan gejala. Entitas *user* hanya dapat melakukan *test* yaitu mendeteksi penyakit dan mendapatkan data hasil *test* yaitu hasil dari perhitungan algoritma berdasarkan gejala yang dipilih. Pakar dapat *login*, menambah gejala beserta bobot gejala dan melakukan *test* mendeteksi penyakit.



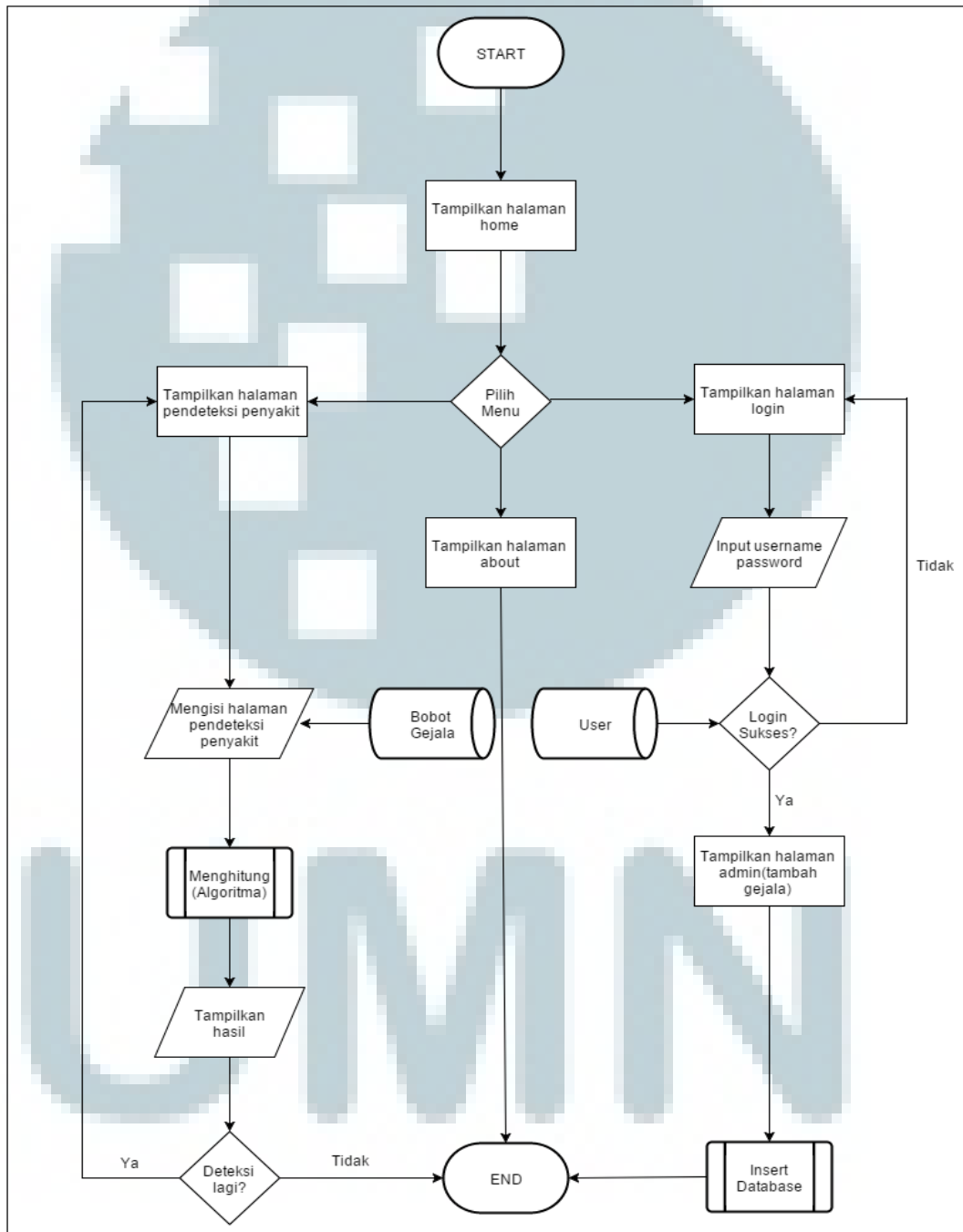
Gambar 3.2 DFD Level 1

DFD Level 1 menjelaskan lebih detail dari *context diagram*. Dari data keluar masuk dan *query user list* yang keluar dari tabel user ketika pakar melakukan *login*. Pada proses ini pakar memberikan data *login admin* saja. Dari tabel gejala memberikan *query list* gejala pada proses *test* ketika *user* melakukan deteksi penyakit, *user* memberikan *data test user* dan mendapatkan data hasil *test user*. Dan juga dari tabel gejala memberikan *query list* gejala pada proses *test* ketika pakar melakukan deteksi penyakit, *admin* memberikan data *test admin* dan mendapatkan data hasil *test admin*. Dan pada proses gejala *management*, pakar memberikan data *insert* gejala, lalu *query insert* gejala tersebut masuk ke tabel gejala.

### 3.2.2 Diagram Alir

Diagram alir pada aplikasi ini dibagi menjadi tiga bagian, yaitu diagram alir sistem secara umum, diagram alir untuk insert database, dan diagram alir untuk menghitung(algoritma).

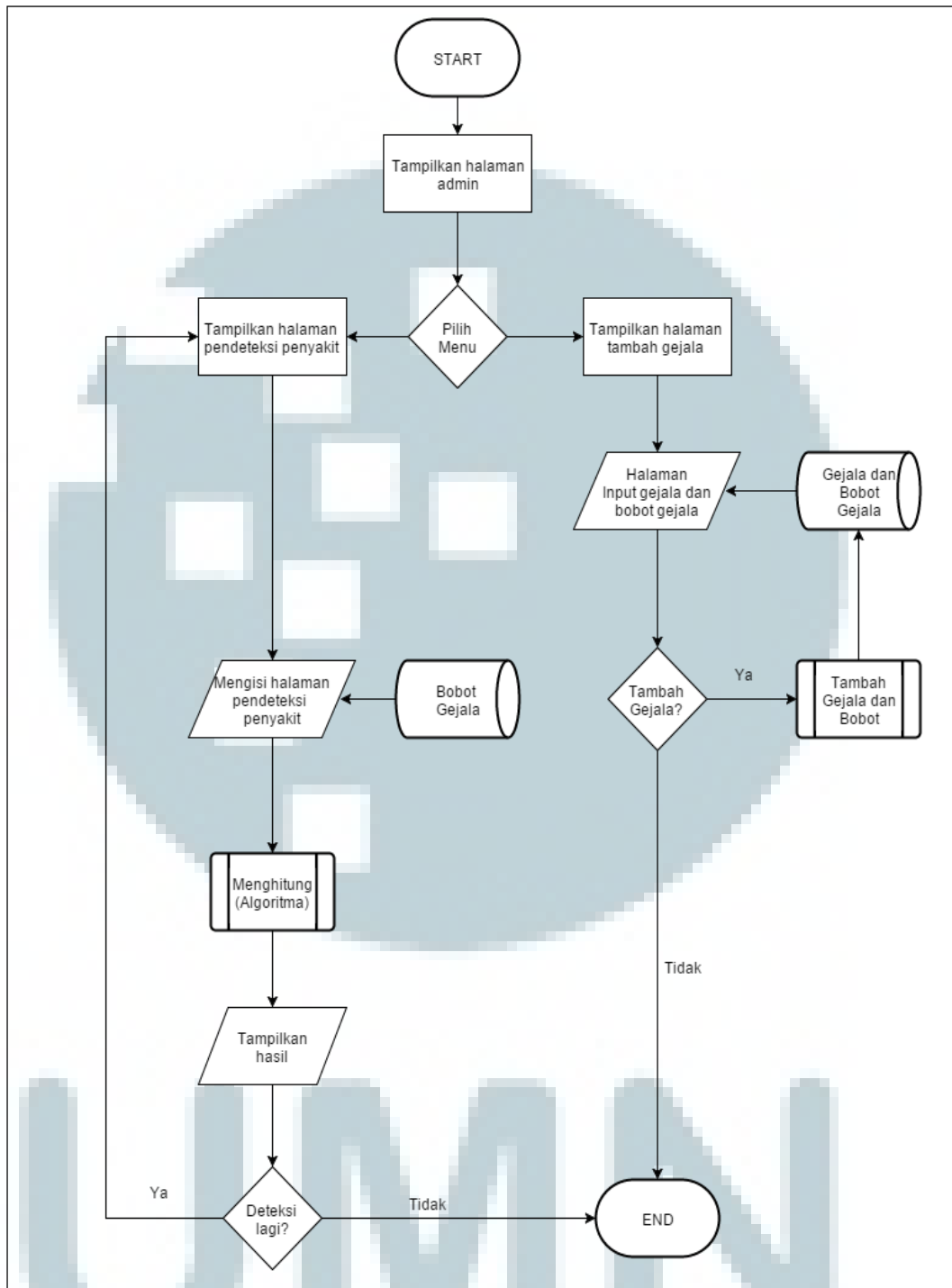
#### A. Diagram Alir Sistem



Gambar 3.3 Diagram Alir Sistem

Gambar 3.3 menggambarkan diagram alir dari aplikasi yang digunakan untuk mendeteksi penyakit tifus dengan algoritma *certainty factor*. Mula-mula proses dimulai dengan menampilkan halaman utama. Pada halaman utama ini terdiri dari 4 menu yaitu *Home*, *Deteksi Penyakit*, *About* dan *Login*. Menu *home* adalah halaman utama yang menampilkan gambar saja. Menu *Deteksi penyakit* adalah halaman untuk *user* untuk mengisi gejala-gejala. Gejala tersebut ditampilkan dari *database* yang sudah dibuat yaitu tabel gejala. *User* hanya tinggal memilih jawaban yang ditampilkan yaitu ya atau tidak. Ketika *user* selesai mengisi halaman deteksi penyakit tersebut, *user* memilih tombol *submit* yang bertujuan untuk mendapatkan hasil. Dalam proses mendapatkan hasil tersebut, pertanyaan gejala yang telah dipilih memiliki bobot yang diberikan oleh pakar. Bobot-bobot tersebut dihitung menggunakan algoritma *certainty factor*. Hasil perhitungan algoritma tersebut akan keluar menjadi presentase di halaman hasil. Setelah selesai, *user* dapat melakukan deteksi lagi dengan memilih halaman deteksi penyakit lagi atau dapat mengakses halaman lain. Berikutnya menu *about* adalah halaman yang berisi pengetahuan yang simpel tentang apakah penyakit tifus itu beserta pembuat website. Setelah itu menu *login* adalah menu khusus untuk *admin*/pakar yang memiliki id untuk mengakses halaman *admin*. Jika *username* dan *password* yang dimasukan salah, maka halaman *admin* tidak dapat diakses dan harus mengisi *username* dan *password* lagi. Jika *login* sukses maka akan tampil halaman *admin*. Pada halaman *admin*, pakar dapat menambahkan gejala jika ada gejala baru yang berhubungan dengan penyakit tifus beserta bobot gejala tersebut. Gejala-gejala beserta bobot yang pakar masukan tersebut akan disimpan di *database*.

**B. Diagram Alir *Insert Database***

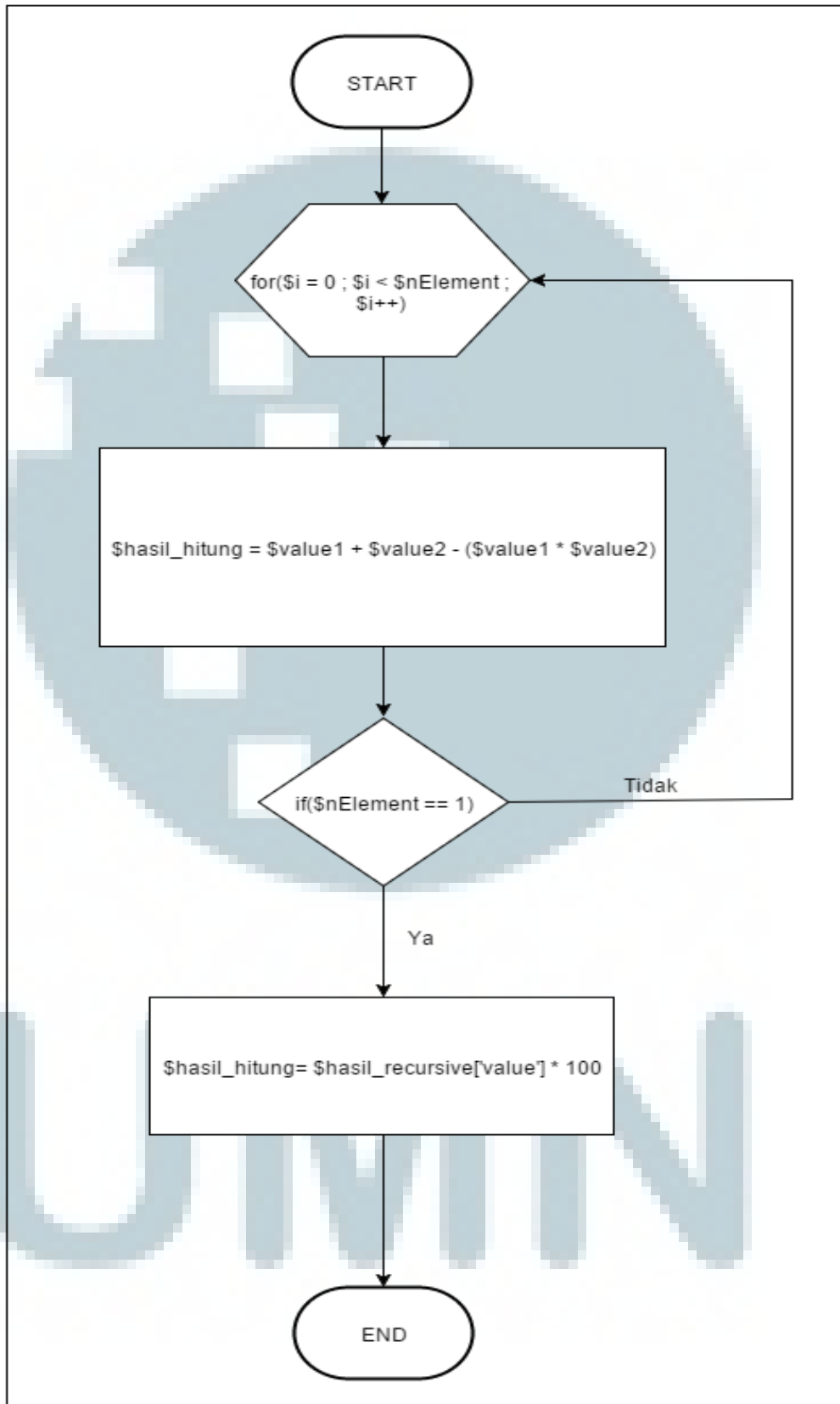


Gambar 3.4 Diagram Alir *Insert Database*



Gambar 3.4 menggambarkan diagram alir dari aplikasi yang digunakan untuk mendeteksi penyakit tifus dengan algoritma *certainty factor*. Mula-mula proses dimulai dengan menampilkan halaman *admin*. Pada halaman *admin* ini terdiri dari 3 menu, yaitu tambah gejala, deteksi penyakit, dan *logout*. Menu tambah gejala ini adalah menu dimana pakar dapat menambahkan gejala jika ada gejala baru yang berhubungan dengan penyakit tifus beserta besar bobot gejala tersebut. Gejala-gejala beserta bobot yang pakar masukan tersebut akan disimpan di *database*. Jika pakar ingin menambahkan gejala lagi, pakar dapat memilih lagi halaman tambah gejala. Jika tidak pakar dapat memilih halaman deteksi penyakit untuk melakukan *test* setelah menambahkan gejala atau keluar dari halaman *admin* yaitu dengan memilih menu *logout*. Pendeteksi penyakit di halaman *admin* ini sama dengan halaman deteksi penyakit di halaman *user* yaitu halaman untuk pakar yang biasanya untuk mengecek atau melakukan *test* setelah menambahkan gejala. Halaman untuk mendeteksi penyakit, halaman ini awal-awal menampilkan gejala. Gejala tersebut ditampilkan dari *database* yang sudah dibuat yaitu tabel gejala. Pakar hanya tinggal memilih jawaban yang ditampilkan yaitu ya atau tidak. Ketika pakar selesai mengisi halaman deteksi penyakit tersebut, pakar memilih tombol *submit* yang bertujuan untuk mendapatkan hasil. Dalam proses mendapatkan hasil tersebut, pertanyaan gejala yang telah dipilih memiliki bobot yang diberikan oleh pakar. Bobot-bobot tersebut dihitung menggunakan algoritma *certainty factor*. Hasil perhitungan algoritma tersebut akan keluar menjadi presentase di halaman hasil. Setelah selesai, pakar dapat melakukan deteksi lagi dengan memilih halaman deteksi penyakit lagi atau dapat mengakses halaman lain.

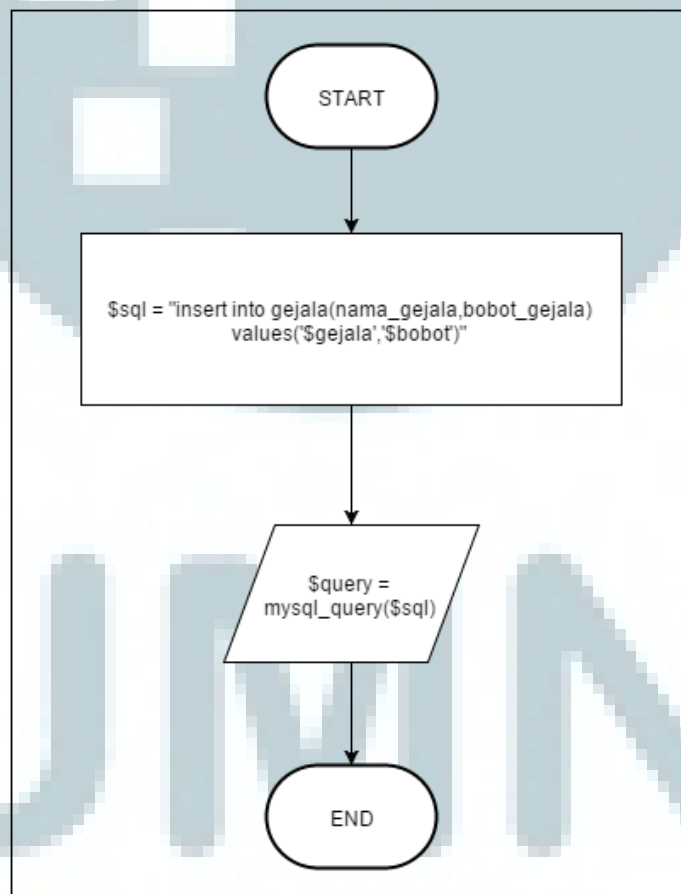
### C. Diagram Alir Menghitung(Algoritma)



Gambar 3.5 Diagram Alir Menghitung(Algoritma)

Gambar 3.5 menggambarkan diagram alir dari aplikasi yang digunakan untuk mendeteksi penyakit tifus dengan algoritma *certainty factor*. Pada diagram alir ini terdapat proses menghitung bobot gejala dengan menggunakan algoritma *certainty factor* dengan bobot-bobot yang diberikan oleh pakar. Algoritma *certainty factor* ini menghitung per 2 bobot gejala sampai hasil tersisa 1 yang disebut hasil akhir. Karena bobot gejala yang dimasukan pakar pasti bernilai positif karena tidak mungkin pakar menambahkan gejala yang tidak berhubungan dengan penyakit tifus. Hasil akhir tersebut akan ditampilkan pada halaman hasil akhir.

#### D. Diagram Alir Tambah Gejala dan Bobot



Gambar 3.6 Diagram Alir Tambah Gejala dan Bobot

Gambar 3.6 menggambarkan diagram alir pada saat gejala dan bobot ditambahkan ke *database*.

### 3.2.3 Struktur Tabel

1. Nama Tabel : *User*

Fungsi : menyimpan data-data pengguna untuk login

*Primary Key* : *id\_user*

*Foreign Key* : -

Tabel 3.1 Tabel *User*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	<i>id_user</i>	int	11	ID User
2	<i>username</i>	varchar	50	Username User
3	<i>password</i>	varchar	50	Password User
4	<i>status</i>	int	1	Status User

Pada tabel *user* ini disimpan data-data pengguna untuk login yaitu *admin / pakar*. *Primary key* tabel *user* ini adalah *id\_user*. *id\_user* merupakan *id user*, *username* adalah *username user*, *password* adalah *password user*, dan *status* adalah *status user*. *Username* dan *password* dibutuhkan oleh pengguna yang sudah terdaftar untuk melakukan *login* untuk mengakses halaman *admin*.

2. Nama Table : *Gejala*

Fungsi : menyimpan data-data gejala beserta bobot

*Primary Key* : *id\_gejala*

*Foreign Key* : -

Tabel 3.2 Tabel Gejala

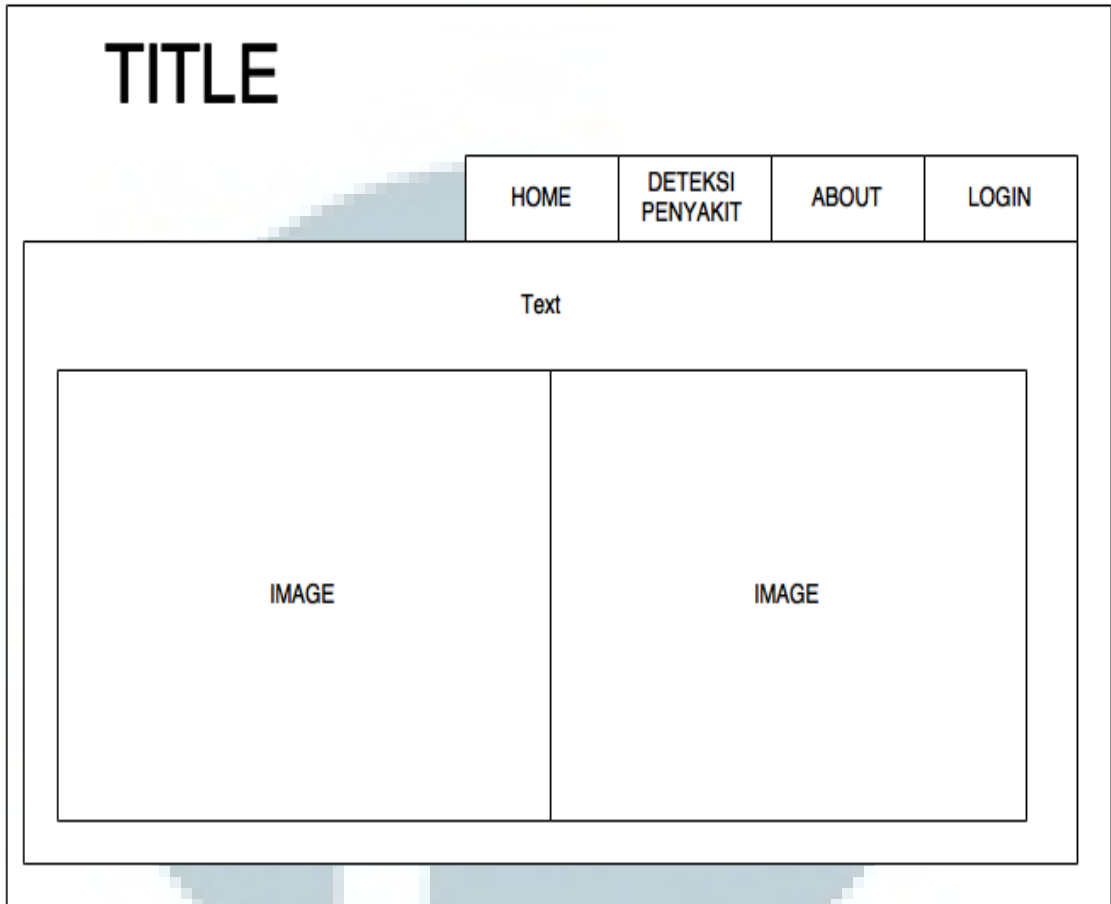
No	Nama Field	Tipe Data	Panjang	Keterangan
1	Id_gejala	int	30	ID Gejala
2	nama_gejala	varchar	150	Nama Gejala
3	bobot_gejala	float	-	Bobot Gejala

Pada tabel gejala ini disimpan data-data gejala beserta bobotnya. *Primary key* dari tabel gejala ini yaitu *id\_gejala*. *Id\_gejala* adalah *id* gejala, *nama\_gejala* adalah nama gejala, dan *bobot\_gejala* adalah bobot gejala. Nama gejala ditampilkan dalam bentuk pertanyaan pada halaman deteksi penyakit. Bobot gejala dihitung pada proses perhitungan menggunakan algoritma *certainty factor*.

### 3.2.4 Design Interface

Untuk memudahkan *user* dalam menggunakan aplikasi, diperlukan tampilan yang mudah dimengerti dan digunakan. Oleh sebab itu, diperlukan *design interface* yang simpel dan mudah dimengerti oleh *user*. Dalam merancang *user interface* ini diperlukan *design* yang berguna dalam pembuatan *design* aplikasi. *Design user interface* ini disesuaikan dengan rancangan sistem. *Design* ini terdiri dari *design* untuk halaman *home*, deteksi penyakit, *about*, dan *login*.

## 1. Halaman *Home*



Gambar 3.7 Antarmuka Halaman Home

Halaman *home* dapat mengakses beberapa bagian yaitu bagian deteksi penyakit, bagian *about*, dan bagian *login*. Menu akan selalu mengikuti pengguna di semua bagian halaman tersebut. Di halaman *home* ini hanya terdapat judul dan gambar saja. Pengguna biasa hanya dapat mengakses bagian *home*, bagian deteksi penyakit, dan bagian *about* saja. Bagian *login* hanya dapat diakses oleh pakar dengan *username* yang sudah didaftarkan. Selanjutnya akan dijelaskan ketika pengguna memilih halaman deteksi penyakit.

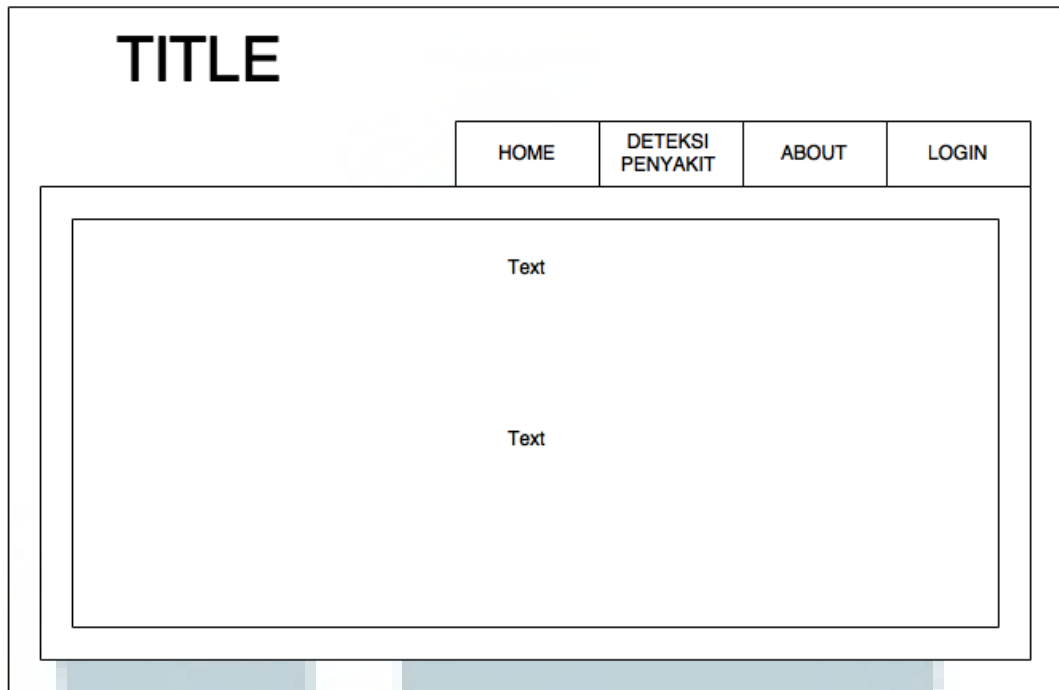
## 2. Halaman Deteksi Penyakit

The image shows a web page layout for disease detection. At the top left, the word "TITLE" is displayed in a large, bold, black font. To the right of the title is a horizontal navigation menu with four buttons: "HOME", "DETEKSI PENYAKIT", "ABOUT", and "LOGIN". Below the navigation menu is a large rectangular frame containing the main content area. Inside this frame, there are two lines of "Text" centered horizontally, followed by a "Test Question" centered horizontally. In the bottom-left corner of the main content area, there is a rectangular button labeled "Submit".

Gambar 3.8 Antarmuka Halaman Deteksi Penyakit

Pada halaman deteksi penyakit ini akan menampilkan daftar pertanyaan yang akan menentukan hasil prediksi sistem. Semua pernyataan ditampilkan sekaligus dalam satu halaman. Dalam menjawab pertanyaan-pertanyaan tersebut, pengguna cukup memilih pilihan ya atau tidak yang telah disediakan. Setelah selesai menjawab semua pertanyaan yang diberikan. Pilih tombol submit untuk melihat hasil deteksi penyakit dari gejala yang pengguna pilih. Dari halaman pendeteksi penyakit ini, dapat mengakses halaman *home*, halaman *about*, dan halaman *login*. Selanjutnya akan dijelaskan halaman *about*.

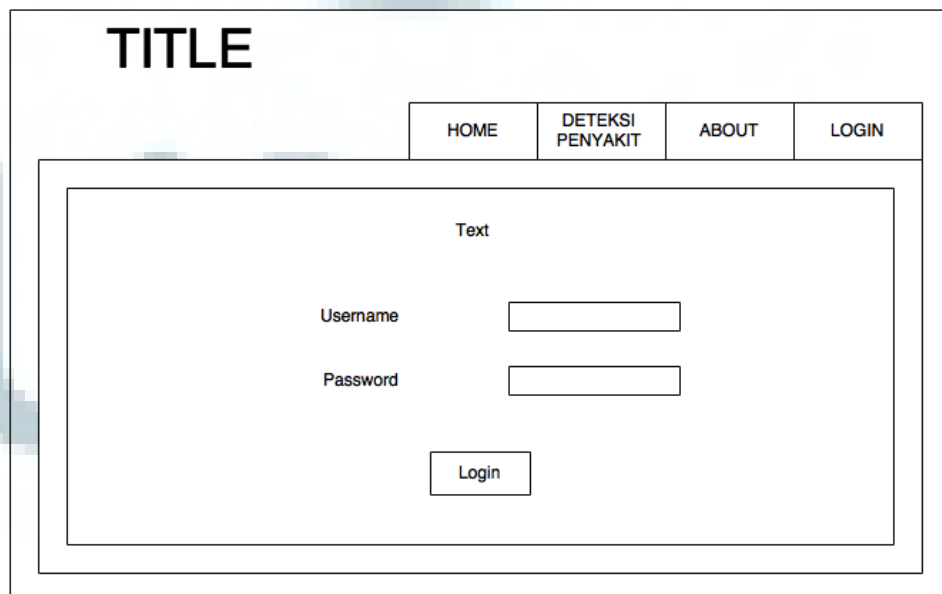
### 3. Halaman *About*



Gambar 3.9 Antarmuka Halaman About

Halaman *about* menjelaskan secara singkat apakah penyakit tifus itu dan identitas pembuat *website*. Selanjutnya akan dijelaskan halaman *login*.

### 4. Halaman *Login*



Gambar 3.10 Antarmuka Halaman Login



Halaman *login* hanya untuk *admin* yang memiliki *username* yang sudah terdaftar untuk mengakses halaman *admin*. Pengguna yang tidak terdaftar tidak bisa mengakses halaman *admin*.

