



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

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi

Metode penelitian yang digunakan dalam mengimplementasikan *keystroke dynamic* dengan jaringan saraf tiruan menggunakan *particle swarm optimization* pada aplikasi untuk mengidentifikasi *user* adalah sebagai berikut.

a. Studi Pustaka

Pada tahap ini, akan dilakukan studi pustaka dengan teori-teori terkait penelitian, bahasa pemrograman yang akan digunakan, algoritma yang digunakan, dan desain aplikasi yang akan dibangun pada penelitian kali ini.

b. Perancangan

Pada perancangan penelitian ini akan dibuat arsitektur jaringan saraf tiruan untuk menunjukkan detail dari masing-masing *layer* pada jaringan saraf tiruan, *flowchart* untuk menjelaskan alur program, *Data Flow Diagram* untuk menjelaskan aliran data yang saling terkait pada sistem, struktur tabel *database* yang akan menjelaskan tabel-tabel terkait sistem, *entity relationship diagram* yang akan menjelaskan hubungan antar tabel dan rancangan antarmuka aplikasi.

c. Implementasi

Pada tahapan implementasi dilakukan penulisan program sesuai dengan rancangan yang dibuat sebelumnya. Aplikasi dibuat menggunakan bahasa pemrograman PHP dengan Javascript dan C#. Aplikasi yang dibuat dapat digunakan untuk mengidentifikasi *user* berdasarkan *input* yang diberikan oleh *user* dan memberikan hasilnya berupa cocok atau tidaknya user.

d. Uji Coba dan Evaluasi

Pada tahapan uji coba, sistem akan diuji oleh *user* dengan memasukkan input dan sistem akan mengeluarkan *output*. Jika hasil *output* tidak sesuai, maka akan dilakukan perbaikan lebih lanjut hingga hasil yang memuaskan.

e. Penulisan Skripsi

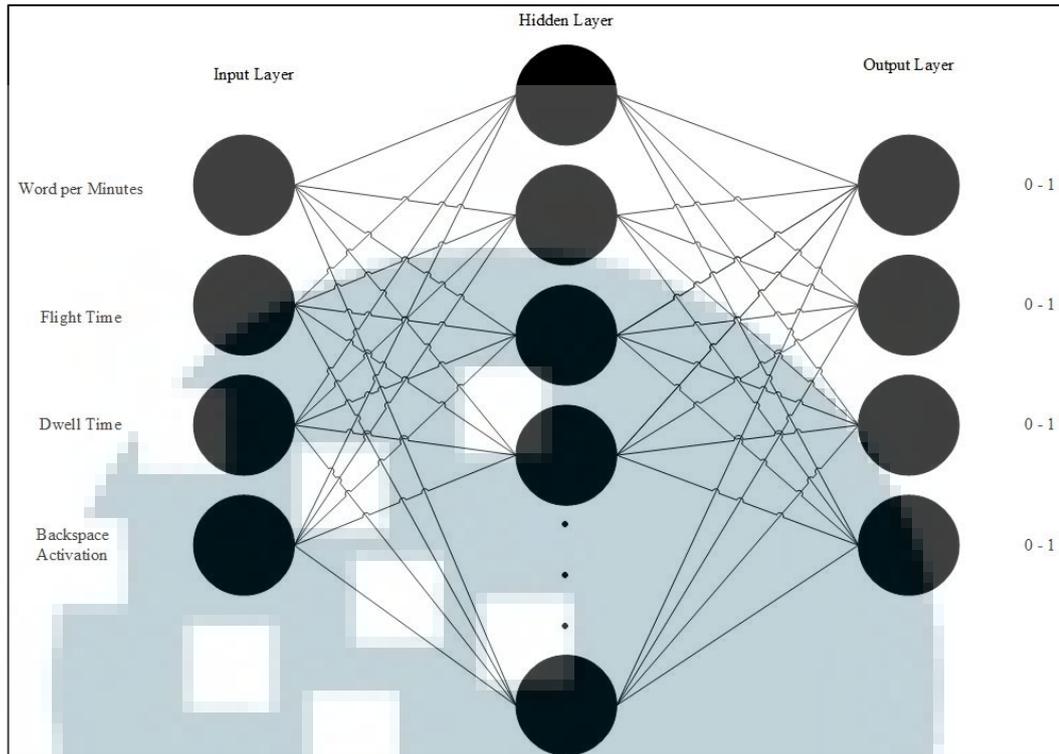
Tahap ini dilakukan penulisan skripsi sebagai bentuk laporan penelitian dan dokumentasi terhadap sistem yang dibuat.

3.2 Perancangan Sistem

Perancangan aplikasi yang dibuat pada penelitian ini adalah arsitektur jaringan saraf tiruan yang akan menjelaskan jumlah neuron pada masing-masing *layer* di jaringan saraf tiruan, *Data Flow Diagram* untuk menggambarkan aliran data dari satu proses ke proses lain, *sitemap* untuk menjelaskan gambaran umum pada *web* aplikasi yang dibangun, *flowchart* untuk menggambarkan alur proses aplikasi, struktur tabel akan menjelaskan secara detail setiap tabel yang digunakan di aplikasi ini, *Entity Relationship Diagram* untuk menjelaskan hubungan antar tabel dan rancangan antarmuka untuk aplikasi yang akan dibangun.

3.2.1 Arsitektur Jaringan Saraf Tiruan

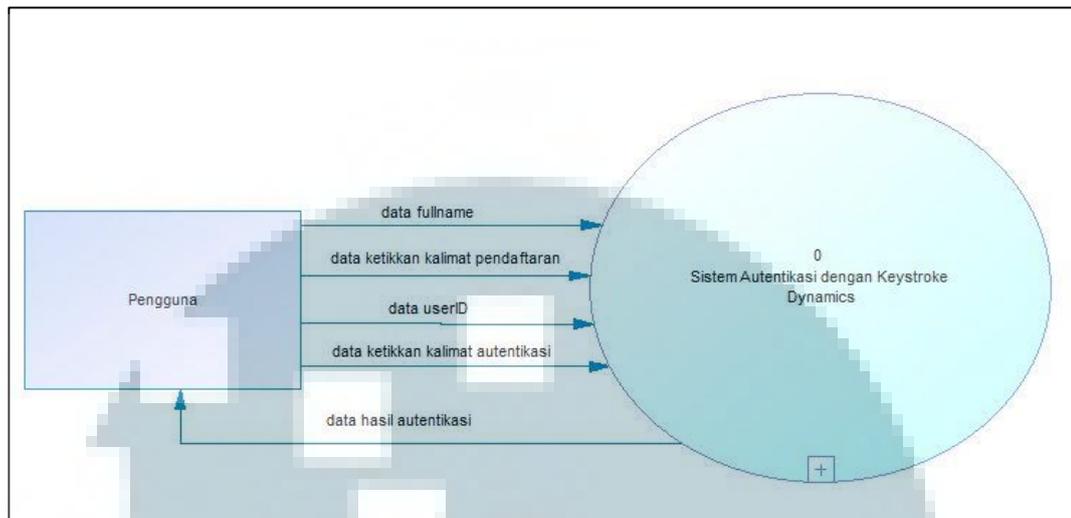
Jaringan saraf tiruan yang akan dibangun pada penelitian ini memiliki tiga *layer* yang terdiri dari satu *input layer*, satu *hidden layer*, dan satu *output layer*. Gambar 3.1 menunjukkan jumlah *neuron* dari masing-masing *layer*.



Gambar 3.1 Perancangan Arsitektur Jaringan Saraf Tiruan

Neuron yang dimiliki oleh *input layer* berjumlah empat yang terdiri dari *Word per Minutes*, *Flight Time*, *Dwell Time*, dan *Backspace Activation*. Keempat masukan tersebut didapat dari olahan data *keystroke* yang selanjutnya digunakan dalam jaringan saraf tiruan. Pada *hidden layer*, jumlah *neuron* akan ditentukan pada proses uji coba dengan minimal *neuron* satu buah, sedangkan pada *output layer*, *neuron* berjumlah empat yang masing-masing *neuron* akan memiliki nilai antara nol (0) hingga satu (1).

3.2.2 Data Flow Diagram

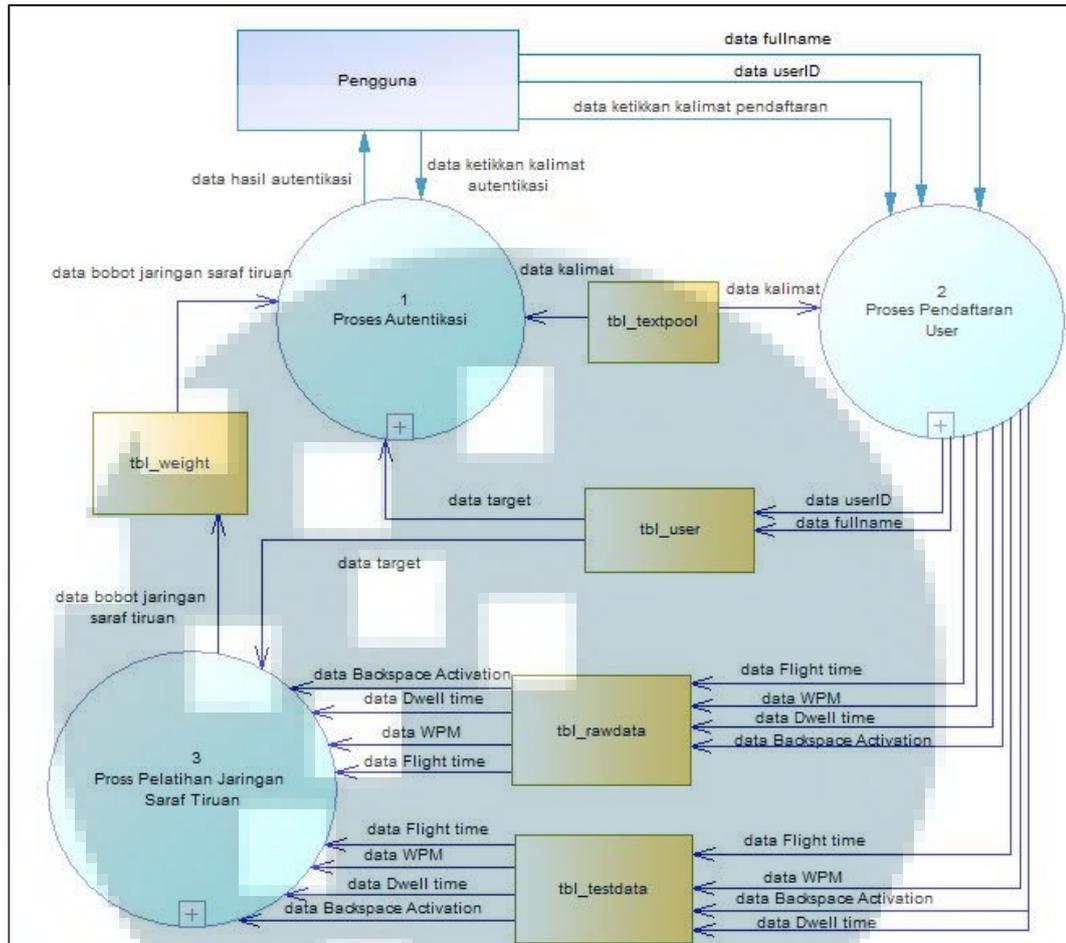


Gambar 3.2 *Context Diagram*

Gambar 3.2 menunjukkan *context diagram* untuk sistem autentikasi dengan *keystroke dynamic*. Terdapat entitas pengguna yang akan menggunakan sistem tersebut. Pengguna akan memasukkan data berupa *fullname*, *userID*, ketikkan kalimat autentikasi, dan ketikkan kalimat pendaftaran. *Fullname* dan *userID* berupa string yang akan digunakan untuk proses pendaftaran dan autentikasi, sedangkan ketikkan kalimat merupakan data yang akan diolah oleh sistem pendaftaran *user* dan sistem autentikasi untuk memperoleh data *WPM*, *flight time*, *dwel time*, dan *backspace activation* yang akan digunakan oleh proses pelatihan jaringan saraf tiruan. Entitas pengguna akan menerima data hasil proses autentikasi berupa *string* yang akan digunakan untuk menentukan bahwa proses autentikasi sudah selesai atau belum.

Pada level 1 dari *data flow diagram*, terdapat tiga proses, yaitu proses pelatihan jaringan saraf tiruan, proses pendaftaran *user*, dan proses autentikasi.

Gambar 3.3 menunjukkan bagaimana aliran data yang terdapat pada level 1 dari *data flow diagram*.

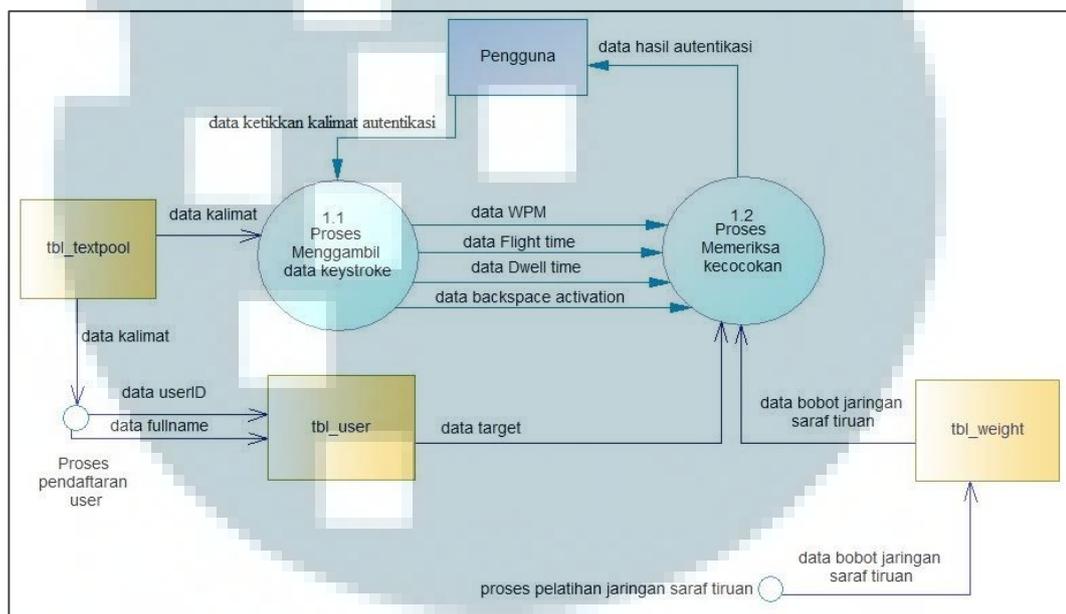


Gambar 3.3 Data Flow Diagram Level 1

Berdasarkan Gambar 3.3, data *fullname*, data *userID*, data ketikkan kalimat pendaftaran, dan data ketikkan kalimat autentikasi digunakan pada proses pendaftaran user dan proses autentikasi. Pada *data flow diagram* level 1 terdapat 5 tabel yang digunakan, yaitu *tbl_textpool* yang memiliki data kalimat untuk digunakan pada proses autentikasi dan proses pendaftaran user, *tbl_user* yang berisi data *userID* dan data *fullname* dari proses pendaftaran user dan data target yang akan digunakan untuk proses autentikasi dan proses pelatihan jaringan saraf tiruan, *tbl_rawdata* yang mendapatkan data *flight time*, data WPM, data *dwell time*, dan data *backspace activation* dari proses pendaftaran yang selanjutnya data tersebut digunakan pada proses pelatihan jaringan saraf tiruan, *tbl_testdata* mendapatkan

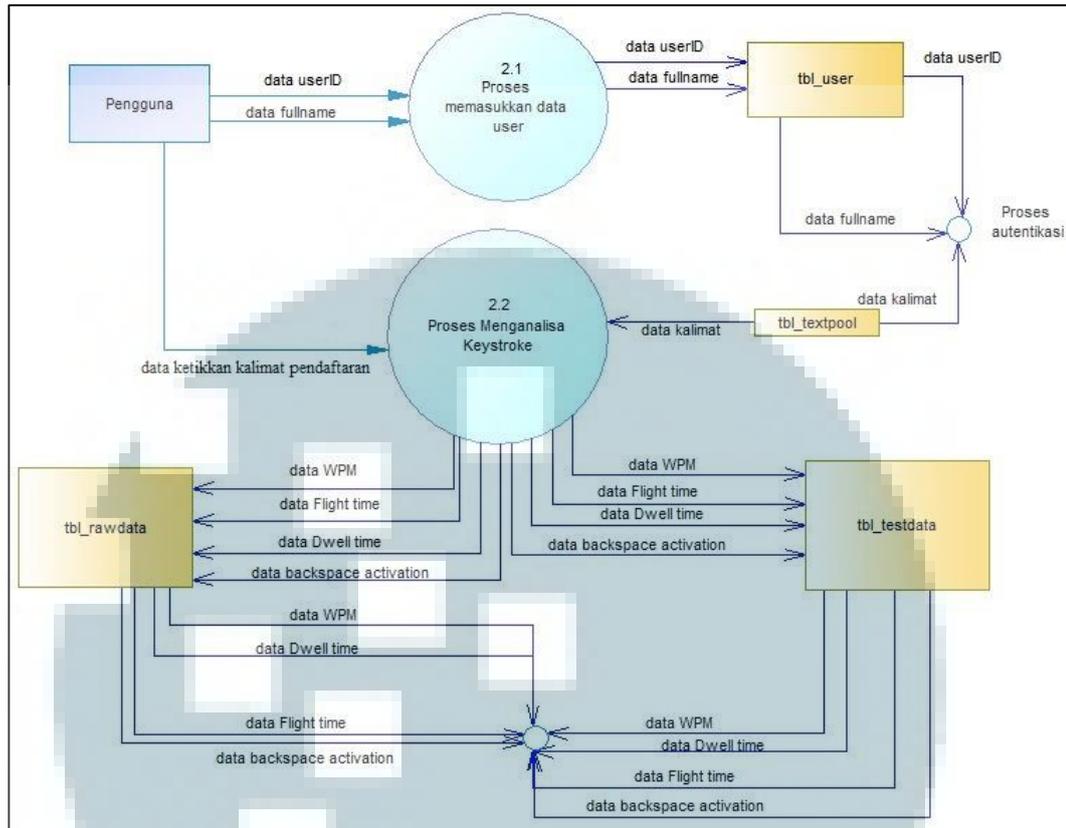
data yang sama dengan `tbl_rawdata` yang selanjutnya digunakan pada proses pelatihan jaringan saraf tiruan, `tbl_weight` yang memiliki data masukkan dari proses pelatihan jaringan saraf tiruan berupa data bobot jaringan saraf tiruan untuk selanjutnya digunakan pada proses autentikasi.

Proses autentikasi akan menggunakan data target dari `tbl_user`, data kalimat dari `tbl_textpool`, dan data bobot jaringan saraf tiruan dari `tbl_weight` untuk digunakan pada proses autentikasi yang akan digambarkan pada Gambar 3.4.



Gambar 3.4 *Data Flow Diagram* Level 2 Proses Autentikasi

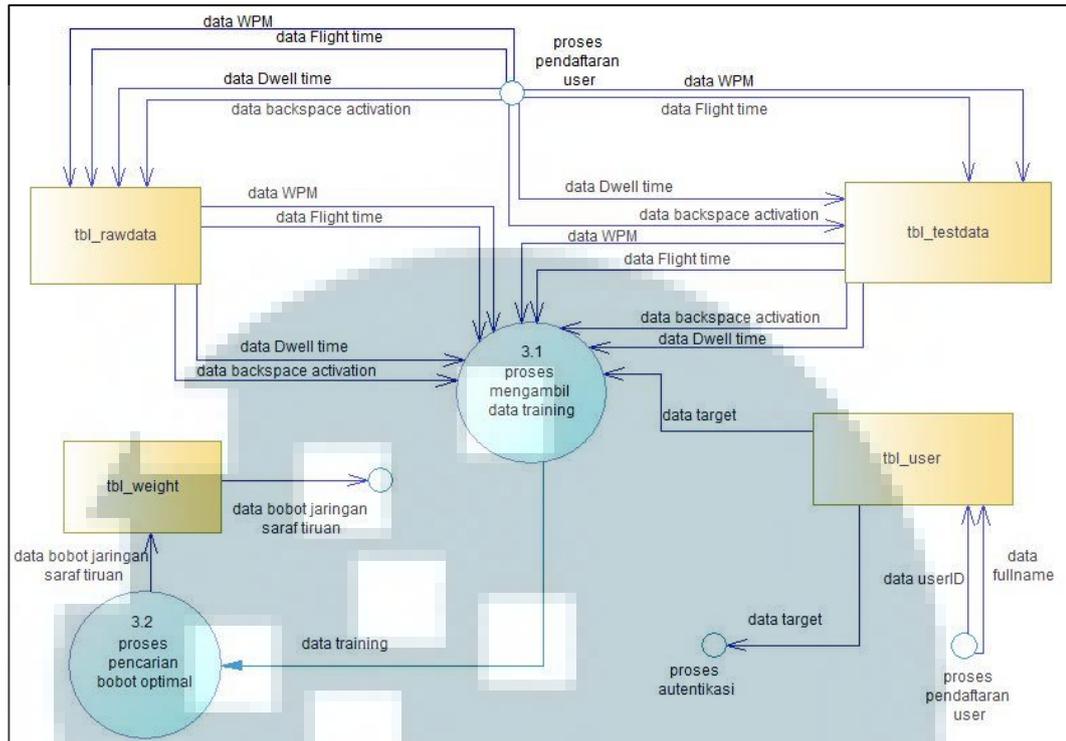
Proses menganalisis *keystroke* akan menerima data ketikkan kalimat autentikasi dari entitas pengguna yang selanjutnya akan diolah menjadi data WPM, data *Flight time*, data *Dwell time*, dan data *backspace activation* yang selanjutnya akan diolah pada proses memeriksa kecocokan. Gambar 3.1 menunjukkan *Data Flow Diagram* level 2 untuk proses pendaftaran *user*.



Gambar 3.5 Data Flow Diagram Level 2 Proses Pendaftaran User

Data *userID* dan data *fullname* yang dimasukkan oleh entitas pengguna diolah pada proses memasukkan data *user* kemudian proses tersebut memasukkan data ke *tbl_user*. Entitas pengguna juga memasukkan data ketikkan kalimat pendaftaran yang akan diolah oleh proses menganalisis *keystroke* untuk menghasilkan data *WPM*, data *Flight time*, data *Dwell time*, dan data *backspace activation*. Masing-masing data tersebut berjenis *float*.

Pada data flow diagram level 2 proses pelatihan jaringan saraf tiruan, terdapat proses mengambil data training dari *tbl_rawdata*, *tbl_testdata*, dan *tbl_user*. Data *training* tersebut meliputi data *WPM*, data *Flight time*, data *Dwell time*, dan data *Backspace activation*, dan data target. Gambar 3.6 menunjukkan aliran data yang terdapat pada *data flow diagram* level 2 proses pelatihan jaringan saraf tiruan.

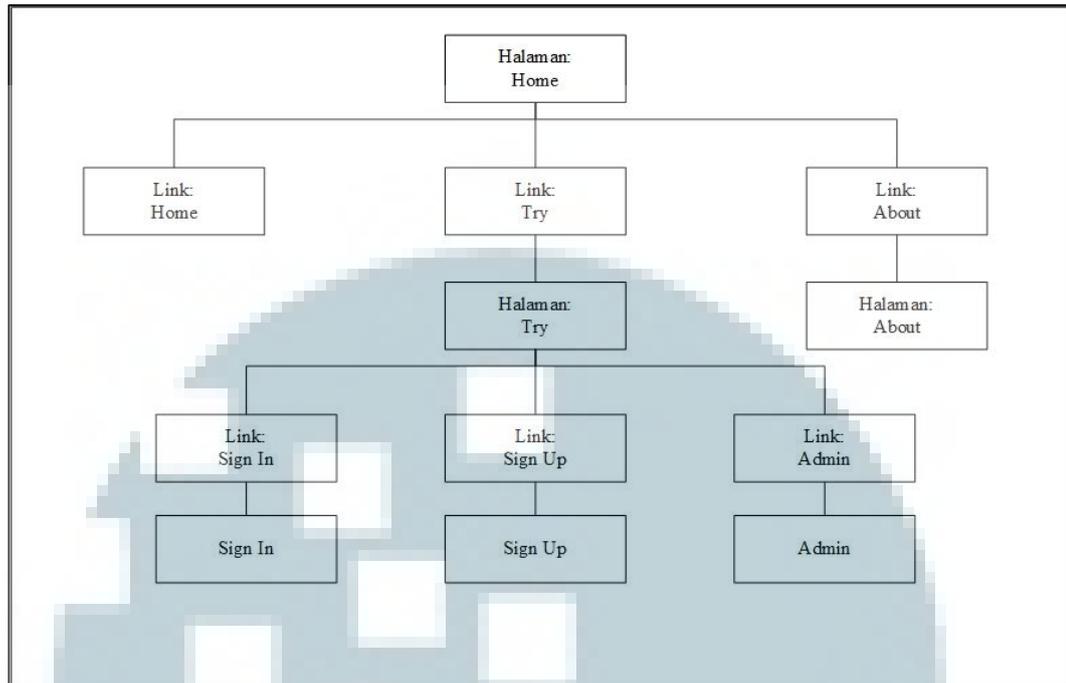


Gambar 3.6 *Data Flow Diagram Level 2* Proses Pelatihan Jaringan Saraf Tiruan

Setelah proses mengambil data *training* selesai, kemudian proses tersebut mengirim data *training* ke proses pencarian bobot optimal untuk selanjutnya diproses hingga menghasilkan data bobot jaringan saraf tiruan yang akan disimpan pada *tbl_weight* untuk digunakan pada proses autentikasi.

3.2.3 Sitemap

Gambar 3.7 menunjukkan *sitemap* pada *website* yang dibangun. Ketika pengguna membuka *website* tersebut, halaman yang pertama kali muncul adalah halaman *Home*. Pada halaman *Home*, terdapat tiga *link* yang terdiri dari *link Home*, *link Try*, dan *link About*.



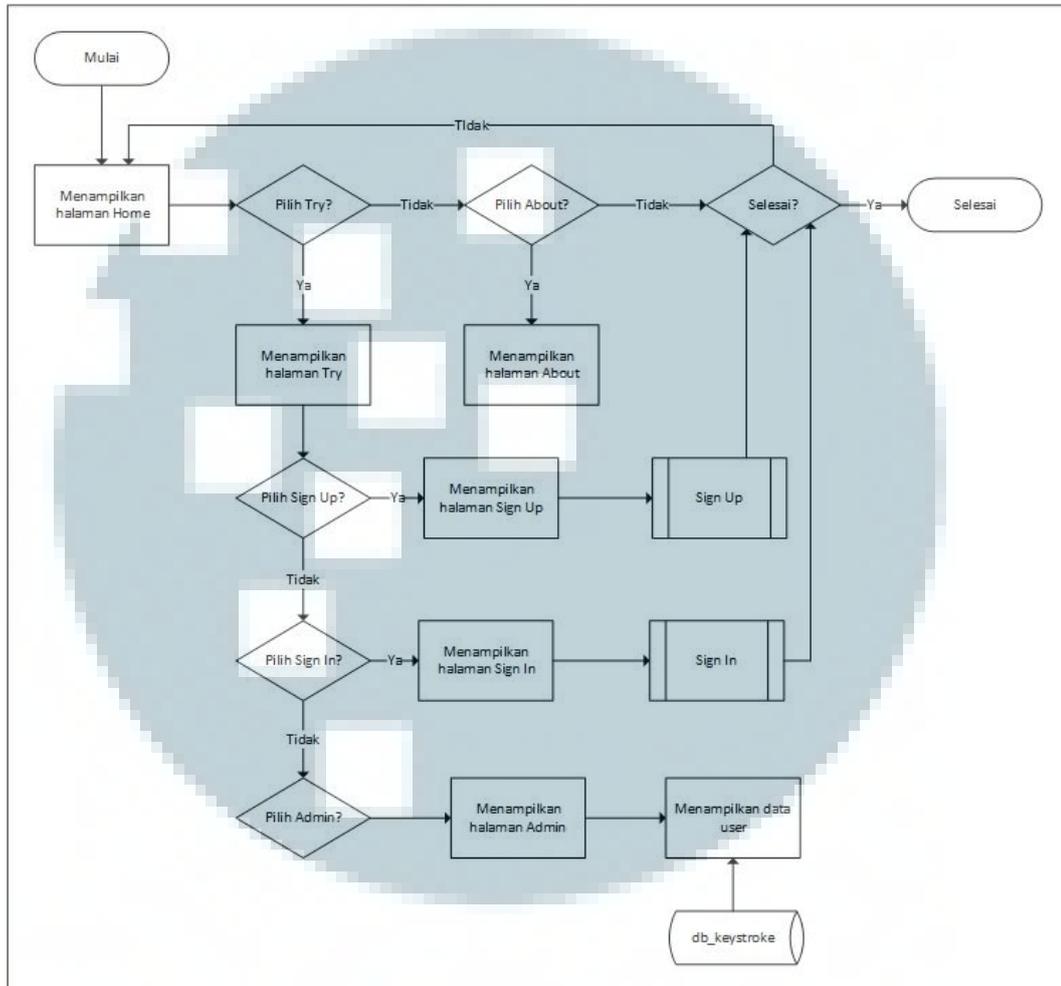
Gambar 3.7 Sitemap Website Autentikasi Menggunakan Keystroke

Jika pengguna menekan *link Home*, maka pengguna akan kembali ke halaman *Home*. Ketika pengguna menekan *link Try*, maka pengguna akan diarahkan ke halaman *Try* yang berisi *link* untuk *Sign Up*, *link* untuk *Sign In*, dan *link Admin*, sedangkan pada *link About*, pengguna akan diarahkan pada halaman *about* yang berisi informasi singkat tentang aplikasi dan pembuatnya. Baik *link Home*, *Try*, maupun *About*, ketiga *link* tersebut berada di setiap halaman yang dimiliki pada *website* ini.

Pada *link sign in*, *link sign up*, dan *link admin*, pengguna akan tetap pada halaman *Try*. Namun, konten pada halaman *Try* disesuaikan dengan *link* yang dipilih oleh pengguna.

3.2.4 Flowchart

Gambar 3.8 menunjukkan *flowchart* dari keseluruhan sistem autentikasi menggunakan *keystroke dynamic*.

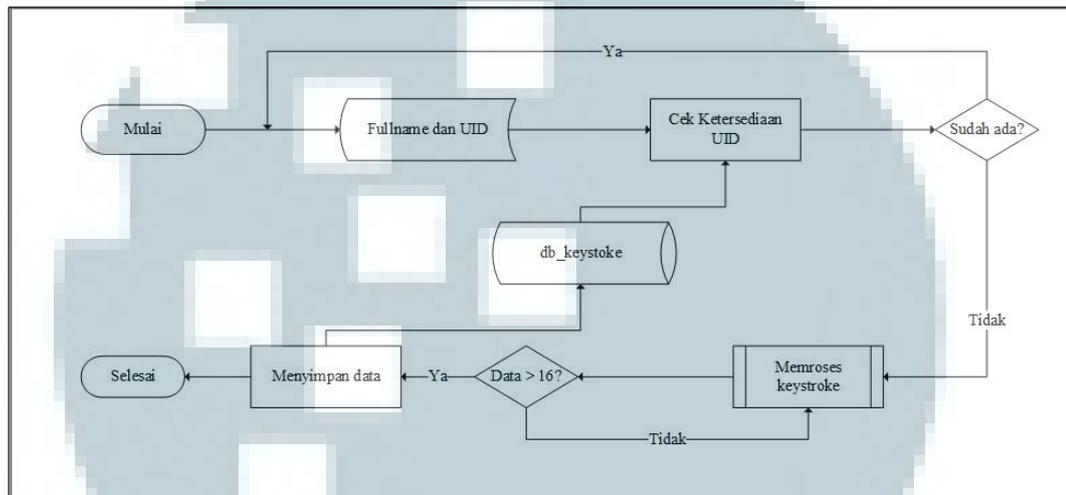


Gambar 3.8 *Flowchart* sistem

Sistem dimulai dengan menampilkan halaman Home, pengguna akan memilih halaman mana yang akan dikunjungi selanjutnya. Terdapat pilihan apakah pengguna akan mengunjungi halaman Try atau halaman About. Pada halaman Try, pengguna dapat memilih halaman selanjutnya seperti halaman Sign up, halaman Sign in, dan halaman Admin. Jika pengguna memilih halaman Sign in, maka akan dilakukan proses Sign up. Jika pengguna memilih proses Sign up, selanjutnya

dilakukan proses Sign in. Sedangkan halaman Admin akan menampilkan daftar *user* yang sudah terdaftar.

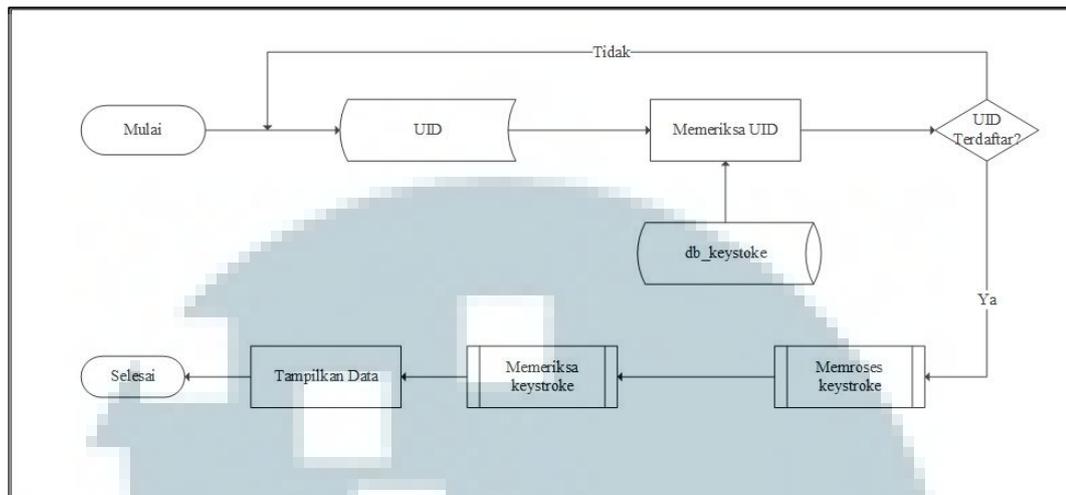
Gambar 3.9 menunjukkan *flowchart* pada bagian *sign up*. Ketika pengguna menekan tombol *sign in*, pengguna diminta untuk memasukkan nama lengkap dan *userID* (UID).



Gambar 3.9 *Flowchart Sign Up*

Setelah menerima *fullname* dan *UID*, selanjutnya akan dilakukan cek ketersediaan UID di *database* apakah sudah pernah terdaftar atau belum, jika sudah, pengguna akan diminta memasukkan UID yang lain. Setelah dipastikan UID tersedia, proses selanjutnya adalah memproses *keystroke* hingga data *keystroke* terkumpul 15 data. Selanjutnya data tersebut disimpan ke *database* untuk selanjutnya digunakan pada proses lain.

Gambar 3.10 menunjukkan *flowchart* untuk bagian *Sign In*. ketika pengguna menekan tombol *Sign In*, pengguna akan memasukkan UID yang sebelumnya sudah didaftarkan. Selanjutnya sistem akan memeriksa UID tersebut, jika belum terdaftar maka pengguna akan diminta untuk memasukkannya lagi, jika sudah maka akan dilanjutkan ke bagian memproses *keystroke*.

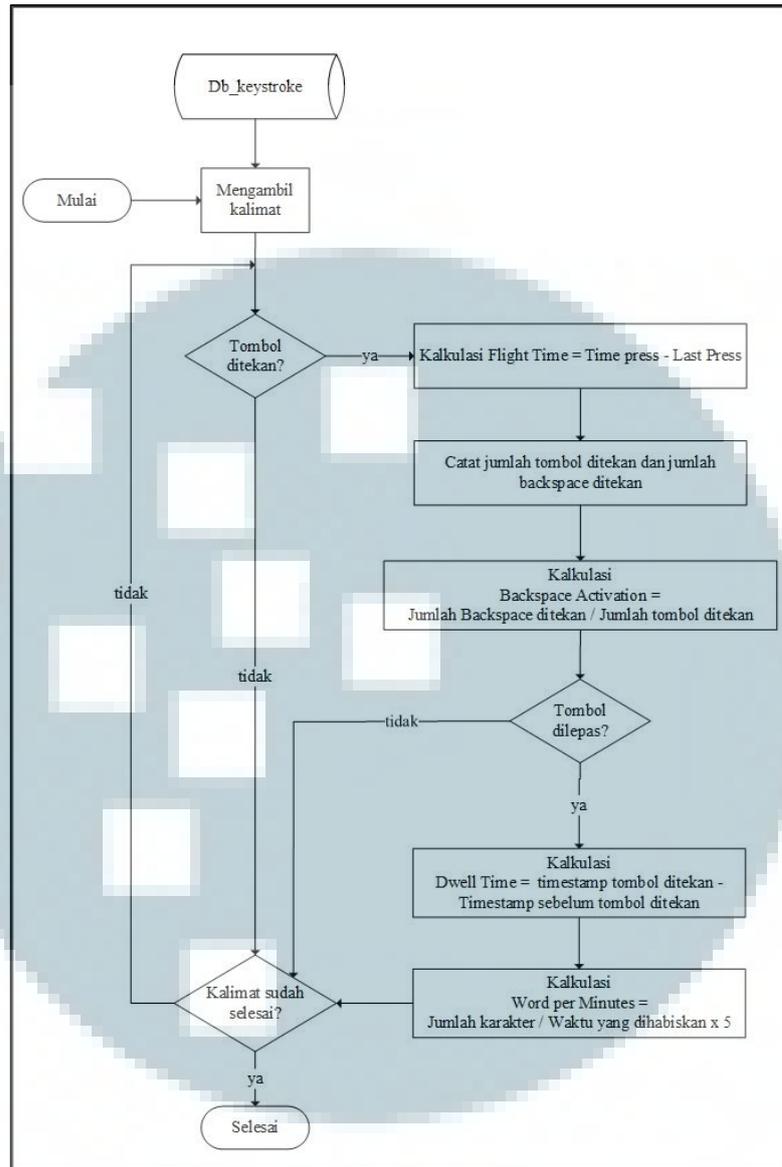


Gambar 3.10 Flowchart Sign In

Setelah memproses *keystroke* selesai, dilanjutkan ke subproses memeriksa *keystroke* yang akan mengembalikan data untuk selanjutnya ditampilkan kepada pengguna.

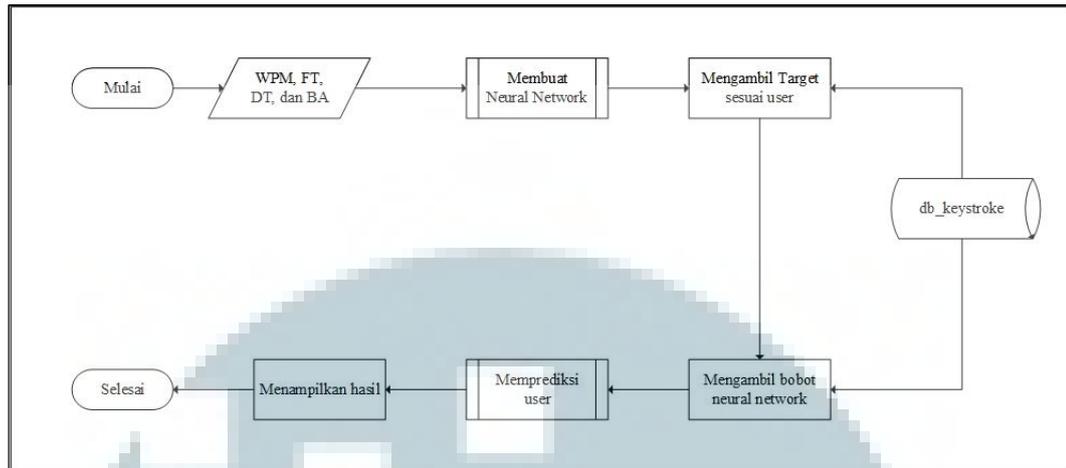
Gambar 3.11 menunjukkan subproses Memproses *keystroke*. Dimulai dari proses mengambil kalimat dari *database* untuk ditampilkan, kemudian pengguna akan mengetikkan kalimat tersebut. Selanjutnya akan dilakukan pengecekan apakah pengguna menekan tombol atau tidak, jika ditekan, maka akan dikalkulasi *Flight time*. Selanjutnya akan dicari jumlah tombol yang ditekan seluruhnya dan jumlah tombol *backspace* yang ditekan. Proses selanjutnya adalah menghitung *backspace activation*.

Selanjutnya akan diperiksa apakah tombol dilepas oleh pengguna atau tidak, jika tombol ditekan, maka akan dikalkulasi *Dwell time* dan selanjutnya menuju proses penghitungan *Word per Minutes*. Hal di atas akan terus diulangi hingga kalimat sudah selesai diketik seluruhnya.



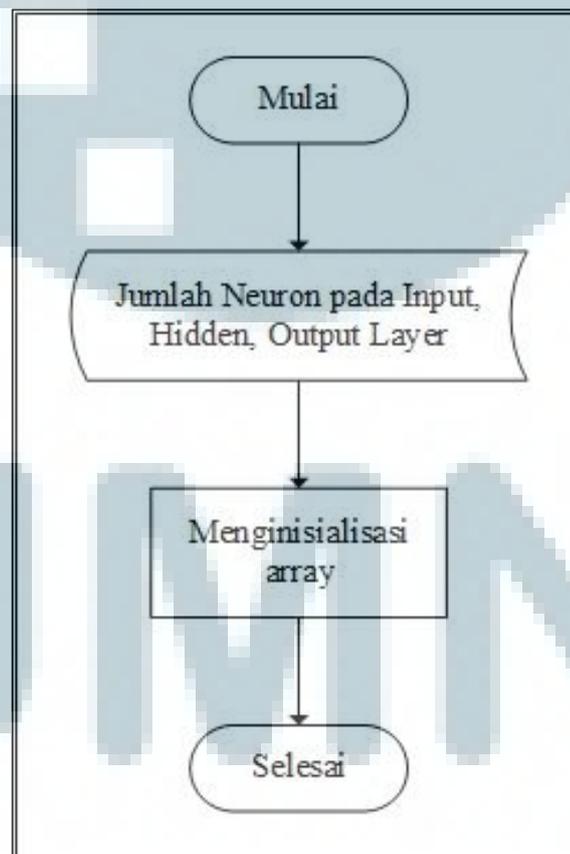
Gambar 3.11 *Flowchart Memproses Keystroke*

Gambar 3.12 menunjukkan proses memeriksa *keystroke*. Proses tersebut menggunakan data *Word per Minutes (WPM)*, *Dwell time*, *Flight time*, dan *Backspace Activation (BA)*. Selanjutnya akan dibuat *Neural Network* pada subproses membuat *neural network*. Kemudian dilanjutkan dengan mengambil target sesuai dengan UID dan mengambil bobot yang ada pada *database*. Kemudian sistem akan dilanjutkan dengan memprediksi berdasarkan data yang ada dan mengembalikan data tersebut.

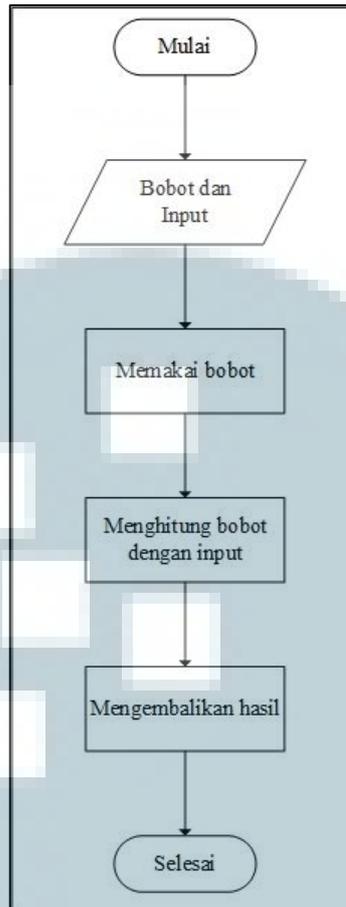


Gambar 3.12 *Flowchart* Memeriksa Keystroke

Pada proses membuat *neural network* yang ditunjukkan pada Gambar 3.13, akan dilakukan inialisasi *array* sesuai dengan jumlah *neuron* pada *layer input*, *hidden*, dan *output*.



Gambar 3.13 *Flowchart* Membuat Neural Network

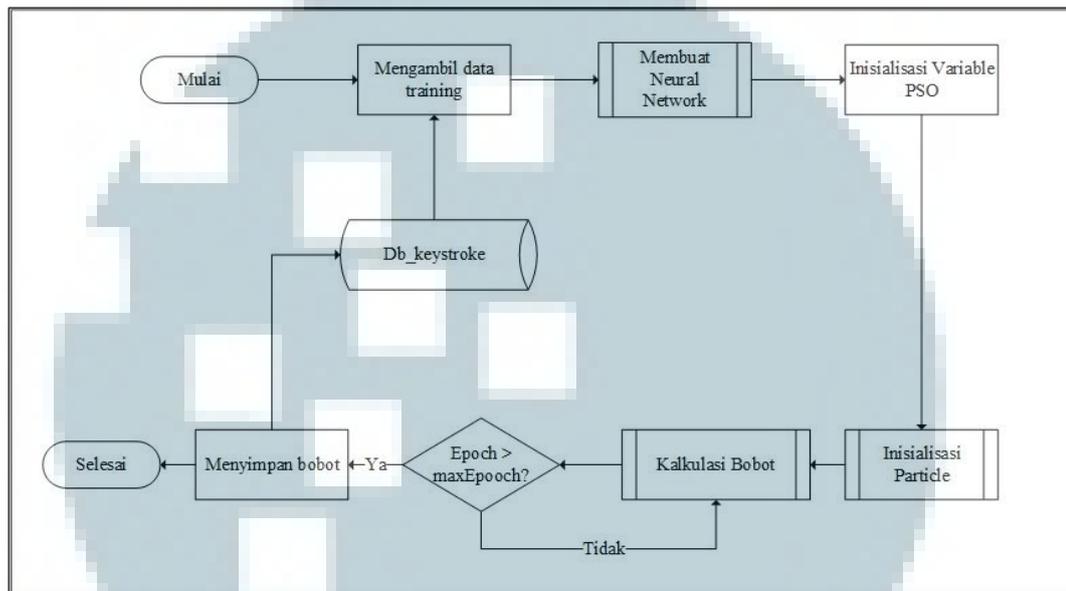


Gambar 3.14 *Flowchart Memprediksi User*

Gambar 3.14 menunjukkan *flowchart* memprediksi *user* dengan menggunakan data bobot dan *input*, kemudian dilanjutkan pada proses memakai bobot tersebut pada *neural network* yang sudah dibuat pada proses sebelumnya. Pada proses selanjutnya, dilakukan penghitungan bobot dengan *input* dan mengembalikan hasil perhitungan ke proses selanjutnya.

Gambar 3.15 menunjukkan *flowchart Training* pada sistem ini. Dimulai dari mengambil data *training* pada *database*, kemudian membuat *neural network*, proses selanjutnya adalah menginisialisasi *variable* untuk *Particle Swarm Optimization*. Pada subproses inisialisasi *particle*, akan dibuatkan kawan *particle* sesuai dengan jumlah yang dimasukkan pada proses inisialisasi *variable*.

Selanjutnya pada subproses kalkulasi bobot, akan dilakukan penghitungan bobot yang optimal sesuai dengan kriteria yang ditentukan. Hal tersebut akan dilakukan hingga maksimum *epoch*/jumlah iterasi terpenuhi dan menyimpan bobot tersebut ke *database*.



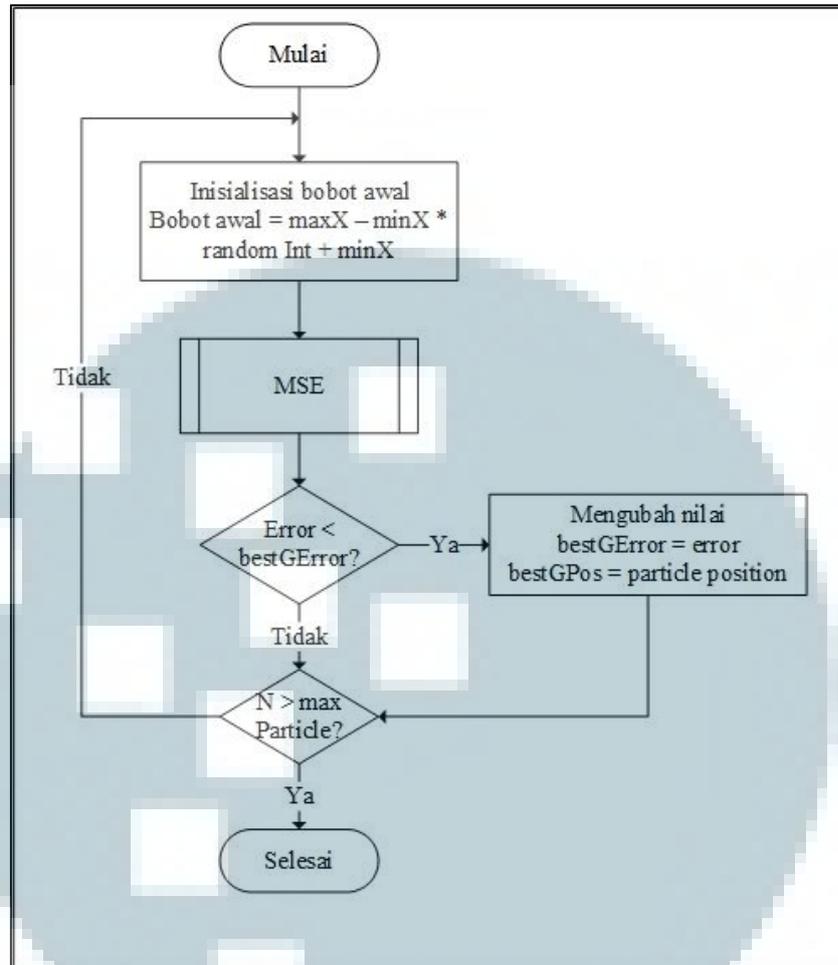
Gambar 3.15 *Flowchart Training*

Gambar 3.16 menunjukkan *flowchart* Inisialisasi *Particle*. Proses dimulai dengan menginisialisasi bobot awal dengan perhitungan sesuai dengan Rumus 3.1.

$$\text{Bobot awal} = \text{maxX} - \text{minX} * \text{random Int} + \text{minx}$$

... Rumus 3.1

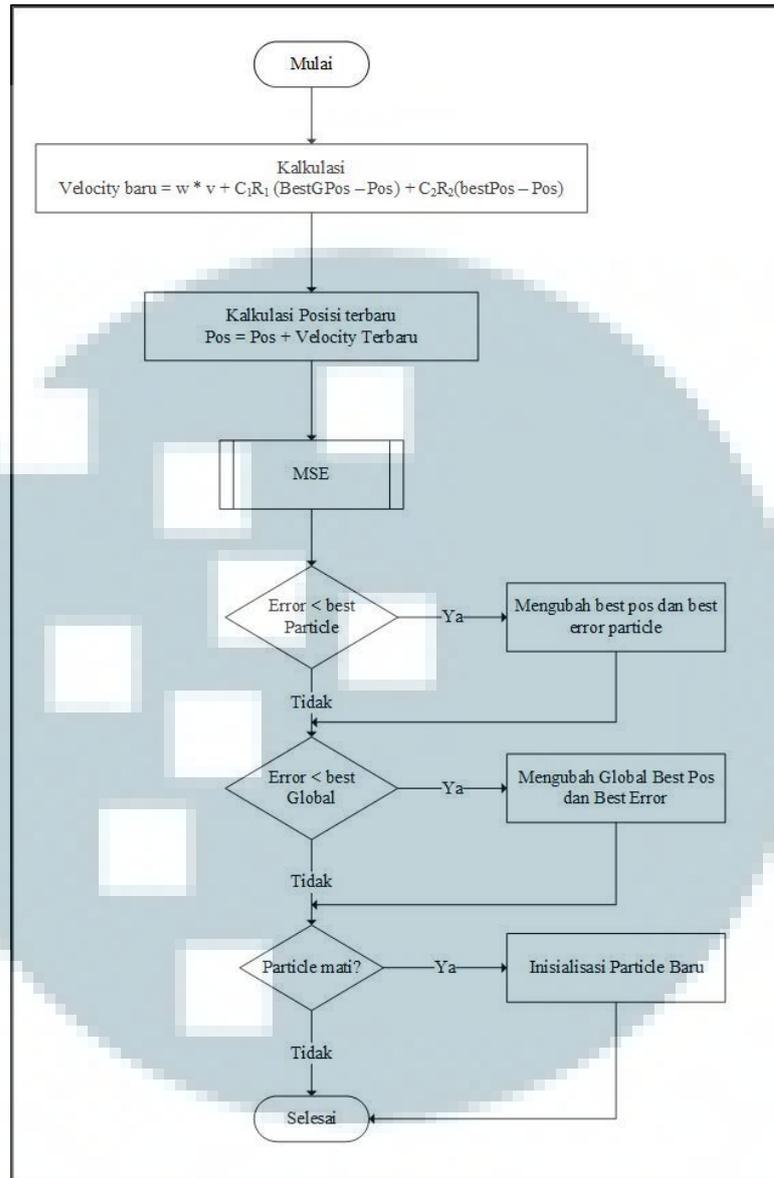
Nilai *maxX* dan *minX* merupakan nilai yang ditentukan menjadi batasan bobot yang akan dimasukan, pada algoritma *Particle Swarm Optimization*, *maxX* dan *minX* dikenal sebagai *search space*.



Gambar 3.16 *Flowchart* Inisialisasi *Particle*

Pada proses selanjutnya, akan dilakukan perhitungan *Mean Square Error* (MSE) untuk mencari selisih antara nilai target dan nilai yang dihasilkan oleh perhitungan bobot. Selanjutnya akan dilakukan pengecekan apakah nilai *error* yang dihasilkan lebih kecil dari nilai *error* terbaik pada seluruh kawan *particle*, jika nilai tersebut lebih kecil, maka nilai *error* terbaik akan diubah menjadi nilai *error particle* tersebut dan posisi terbaik seluruh kawan *particle* akan diubah menjadi nilai posisi *particle* tersebut. Hal tersebut diulangi sesuai dengan jumlah *particle* yang sudah ditentukan sebelumnya.

Gambar 3.17 menunjukkan *flowchart* Kalkulasi Bobot. Proses tersebut mencari bobot optimal untuk *Neural Network* dan akan menyimpannya di *database*.



Gambar 3.17 *Flowchart* Kalkulasi Bobot

Proses dimulai dengan menentukan *velocity* atau arah perpindahan dari *particle* ke posisi yang baru dengan perhitungan seperti pada Rumus 3.2.

$$\text{new velocity} = w \times v + c_1 r_1 (\text{BestGPos} - \text{Pos}) + c_2 r_2 (\text{bestPos} - \text{Pos})$$

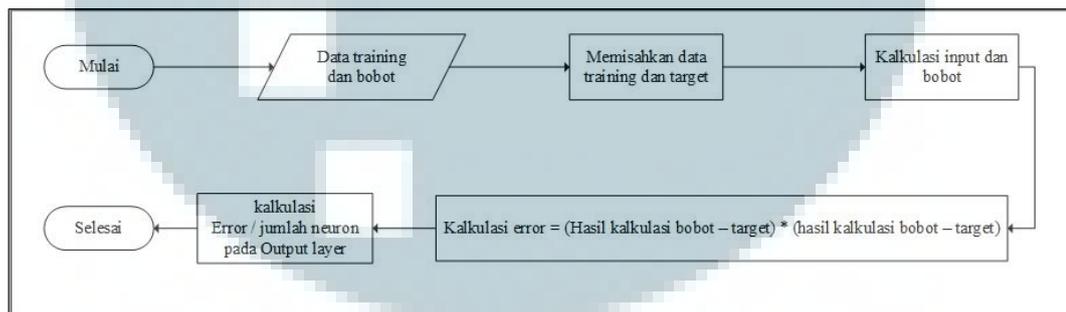
... Rumus 3.2

Selanjutnya adalah proses memindahkan *particle* ke posisi yang baru dengan perhitungan seperti pada Rumus 3.3.

$$\text{Posisi} = \text{posisi} + \text{new velocity}$$

... Rumus 3.3

Pada proses selanjutnya, akan dilakukan perhitungan *error* pada subproses MSE. Setelah nilai *error* terbaru didapatkan, dilakukan pengecekan apakah nilai *error* tersebut lebih kecil dari nilai *error* terbaik yang *particle* miliki, jika ya, maka posisi tersebut dimasukkan ke dalam nilai posisi terbaik yang dimiliki oleh *particle*. Selanjutnya, apakah nilai *error* tersebut lebih kecil dari nilai *error* terbaik yang dimiliki oleh kawanan *particle*, jika ya, maka nilai posisi tersebut dimasukkan ke dalam nilai posisi terbaik dari kawanan *particle*. *Particle* juga memiliki kemungkinan hilang dari kawanan, jika *particle* hilang, maka akan dibuatkan *particle* baru dengan posisi yang random.



Gambar 3.18 *Flowchart Mean Square Error (MSE)*

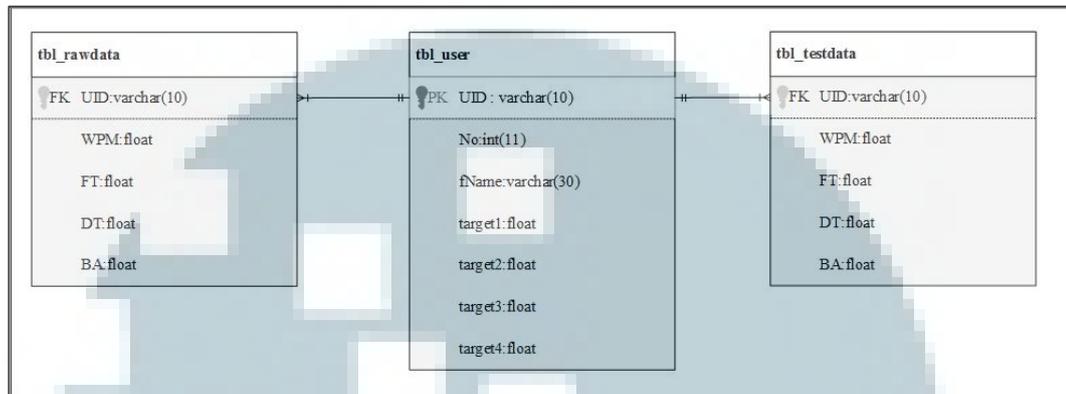
Gambar 3.18 menunjukkan *flowchart Mean Square Error*. Pada subproses MSE, dilakukan perhitungan selisih antara hasil kalkulasi bobot dengan target yang seharusnya dihasilkan. Proses dimulai dengan *input* data *training* dan bobot, selanjutnya dipisahkan antara *input* dan target pada data *training* untuk selanjutnya dihitung melalui proses kalkulasi *input* dan bobot. Kalkulasi *error* berdasarkan Rumus 3.4.

$$\text{error} = (\text{hasil kalkulasi bobot} - \text{target}) \times (\text{hasil kalkulasi bobot} - \text{target})$$

... Rumus 3.4

Selanjutnya *error* tersebut dijumlahkan dan dibagi dengan jumlah *neuron* pada *output layer*.

3.2.5 Entity Relationship Diagram



Gambar 3.19 Entity Relationship Diagram Sistem Autentikasi dengan *Keystroke*

Gambar 3.19 menunjukkan tabel yang memiliki hubungan pada sistem autentikasi dengan *keystroke*. Tabel *tbl_user* memiliki *primary key* *UID*, sedangkan *tbl_rawdata* dan *tbl_testdata* memiliki *foreign key* *UID* yang sama dengan *UID* pada *tbl_user*. Hubungan antara *tbl_user* dan *tbl_rawdata* adalah *one-to-many*, yang berarti satu *user* dapat memiliki data lebih dari satu pada *tbl_rawdata* begitu pula pada *tbl_testdata* memiliki hubungan *one-to-many* dengan *tbl_user*, yang berarti satu *user* dapat memiliki data pada *tbl_testdata* lebih dari satu.

3.2.6 Struktur Tabel

Aplikasi yang dibuat menggunakan *database* *MySQL* dalam penyimpanan data dengan struktur tabel sebagai berikut.

A. Tabel *tbl_user*

Fungsi : Tabel ini digunakan untuk menyimpan data user.

Primary Key : *UID*

Tabel 3.1 Struktur Tabel tbl_user

Nama Kolom	Tipe	Panjang	Keterangan
No	Integer	11	Nomor urut <i>user</i>
UID	Varchar	10	<i>User ID</i> yang digunakan untuk <i>sign in</i>
fName	Varchar	30	Nama lengkap
Target1	Integer	11	Nilai target <i>binary</i> ke-1
Target2	Integer	11	Nilai target <i>binary</i> ke-2
Target3	Integer	11	Nilai target <i>binary</i> ke-3
Target4	Integer	11	Nilai target <i>binary</i> ke-4

B. Tabel tbl_textpool

Fungsi : Tabel ini berisi kalimat yang digunakan dalam proses autentikasi dan pendaftaran user.

Primary Key : textID

Tabel 3.2 Struktur Tabel tbl_textpool

Nama Kolom	Tipe	Panjang	Keterangan
textID	Integer	11	Nomor unik setiap <i>text</i>
text	text	255	Kalimat yang digunakan untuk proses autentikasi dan pendaftaran <i>user</i>

C. Tabel tbl_weight

Fungsi : Tabel ini digunakan untuk menyimpan bobot jaringan saraf tiruan.

Tabel 3.3 Struktur Tabel tbl_weight

Nama Kolom	Tipe	Panjang	Keterangan
colNo	Integer	11	Nomor urut nilai bobot
wVal	double		Nilai bobot

D. Tabel tbl_rawdata

Fungsi : Tabel ini digunakan untuk menyimpan data *keystroke* dari proses pendaftaran yang akan digunakan pada proses training.

Foreign Key : UID

Tabel 3.4 Struktur Tabel tbl_rawdata

Nama Kolom	Tipe	Panjang	Keterangan
UID	Varchar	10	User ID yang digunakan untuk <i>sign in</i>
WPM	float		Nilai <i>Word per Minutes</i>
FT	float		Nilai <i>Flight time</i>
DT	float		Nilai <i>Dwell time</i>
BA	float		Nilai <i>Backspace activation</i>

E. Tabel tbl_testdata

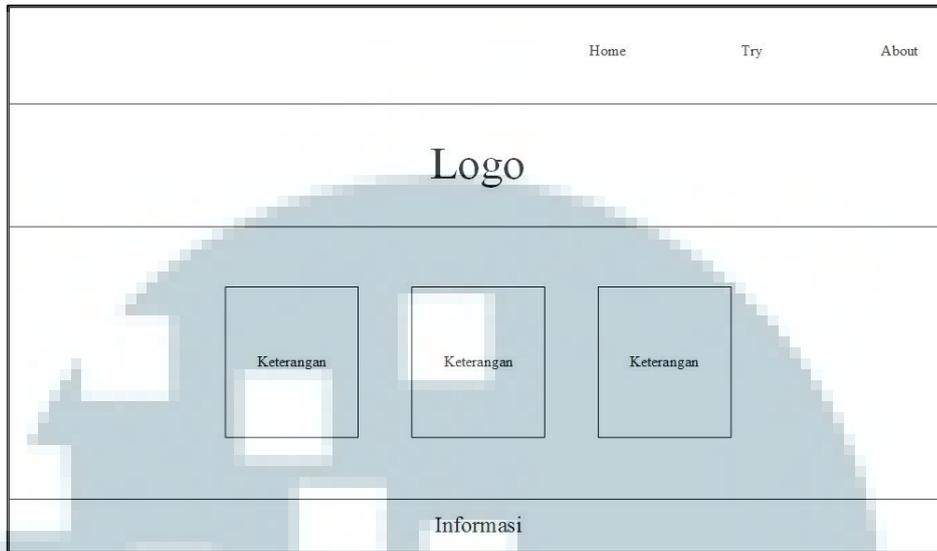
Fungsi : Tabel ini digunakan untuk menyimpan data *keystroke* dari proses pendaftaran yang digunakan untuk validasi bobot.

Foreign Key : UID

Tabel 3.5 Struktur Tabel tbl_testdata

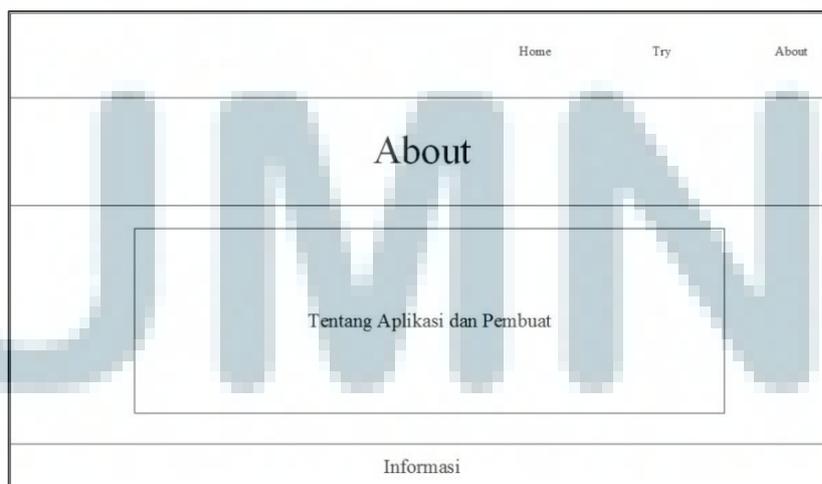
Nama Kolom	Tipe	Panjang	Keterangan
UID	Varchar	10	User ID yang digunakan untuk <i>sign in</i>
WPM	float		Nilai bobot
FT	float		Nilai <i>Flight time</i>
DT	float		Nilai <i>Dwell time</i>
BA	float		Nilai <i>Backspace activation</i>

3.2.7 Rancangan Antarmuka



Gambar 3.20 Rancangan Antarmuka Halaman Depan

Gambar 3.20 menunjukkan rancangan antarmuka halaman depan. Halaman depan terdiri dari keterangan singkat bagaimana *keystroke* bekerja. Pada bagian atas halaman depan terdapat *link* untuk menuju halaman *Home*, halaman *Try*, dan halaman *About*. Ketiga *link* tersebut selalu berada di setiap halaman yang akan buat. Pada bagian bawah halaman juga terdapat informasi mengenai pembuat *website* yang selalu muncul di berbagai halaman.



Gambar 3.21 Rancangan Antarmuka Halaman *About*

Gambar 3.21 menunjukkan rancangan antarmuka halaman *About* yang berisi informasi mengenai aplikasi, informasi mengenai pembuat aplikasi, dan tujuan dari dibuatnya aplikasi tersebut. Sama halnya seperti halaman *Home*, halaman tersebut memiliki tiga *link* di bagian atas dan informasi di bagian bawah halaman.

Pada link *Try*, terdapat tiga tombol seperti yang ditunjukkan pada Gambar 3.22. Ketiga tombol tersebut memiliki fungsi sesuai dengan tulisan yang berada pada tombol tersebut.



Gambar 3.22 Rancangan Antarmuka Halaman *Try*

Ketika *user* menekan tombol *Sign In*, *user* tetap berada pada halaman *Try*. Namun, konten pada halaman *Try* berganti menjadi seperti yang ditunjukkan pada Gambar 3.23.

Home	Try	About
<h2>Sign In</h2>		
<input type="text" value="User ID"/>		
<input type="button" value="Masuk"/>		
Informasi		

Gambar 3.23 Rancangan Antarmuka Halaman *Sign In*

Halaman *Try* dengan konten *Sign in* terdiri dari *textbox* untuk memasukkan *user ID* dan tombol masuk.

Home	Try	About
<h2>Sign In</h2>		
<input type="text" value="Keterangan Cocok/Tidak Cocok"/>		
<input type="button" value="Coba lagi"/>		
Informasi		

Gambar 3.24 Rancangan Antarmuka Keterangan *Sign In*

Gambar 3.24 menunjukkan rancangan antarmuka keterangan *sign in*. pada halaman tersebut terdapat keterangan mengenai cocok atau tidaknya seorang *user* terhadap pola *keystroke* yang dilakukan dan juga terdapat tombol coba lagi untuk mengulang proses autentikasi.

Gambar 3.25 Rancangan Antarmuka Halaman Pengambilan Data *keystroke*

Gambar 3.25 menunjukkan rancangan antarmuka halaman untuk pengambilan data *keystroke*. Halaman tersebut terdiri dari bagian kalimat yang harus diketikkan oleh *user* dan *textarea* yang menjadi tempat *user* mengetik. Halaman tersebut juga memiliki tombol kalimat selanjutnya untuk menuju kalimat lainnya.

Gambar 3.26 Rancangan Antarmuka Halaman *Sign Up*

Berdasarkan Gambar 3.26, rancangan antarmuka halaman *sign up* terdiri dari dua *textbox* untuk mengisi nama lengkap dan *user ID*. Halaman tersebut

juga memiliki tombol mulai untuk menuju halaman pengambilan data *keystroke*.

Bagian *Admin* ditunjukkan pada Gambar 3.27. Halaman tersebut menampilkan daftar dari *user* yang sudah terdaftar pada aplikasi beserta target dari masing-masing *user*.



Gambar 3.27 Rancangan Antarmuka Halaman Admin

UMMN