



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

3.1 Metode Penelitian

Adapun metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

a. Studi Literatur

Bertujuan mencari dan menemukan informasi-informasi yang berkaitan dengan topik yang dibahas dalam penelitian ini, baik mengenai pohon keputusan dan algoritma C4.5. Referensi yang digunakan bersumber dari buku, artikel, jurnal ilmiah, serta sumber lainnya.

b. Studi Kasus

Pencarian informasi adalah melakukan studi kasus dengan mengunjungi SMA Tarsisius Vireta secara langsung untuk mendapatkan data-data murid yang dibutuhkan serta nilai minimal yang dibutuhkan untuk memasuki jurusan IPA. Narasumber adalah Ibu Dwiyati M. T. Harbelubun, S.Pd selaku kepala bagian kurikulum.

c. Analisa Sistem

Analisis sistem merupakan tahap dimana semua kebutuhan yang diperlukan dalam pembangunan sistem dianalisis. Analisis tersebut berupa *platform* pengembangan sistem, basis data yang digunakan, dan perancangan sistemnya.

d. Perancangan Sistem

Perancangan yang dilakukan pada tahap ini adalah perancangan tampilan atau *user interface* aplikasi serta fitur-fitur yang akan terdapat dalam aplikasi ini.



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.

e. Pemrograman Sistem

Setelah proses-proses sebelumnya selesai dilakukan, maka dilakukan pemrograman sistem. Pemrograman sistem adalah realisasi atau pembuatan sistem yang telah dirancang sebelumnya.

f. Testing dan Evaluasi

Setelah proses pemrograman sistem selesai, maka proses selanjutnya yang dilakukan adalah uji coba terhadap aplikasi yang telah dibuat. Kemudian dilanjutkan dengan melakukan evaluasi atas hasil dari uji coba tersebut.

g. Penulisan Laporan

Penulisan laporan bertujuan sebagai dokumentasi dari sebuah penelitian yang telah dilakukan sehingga dapat memberikan informasi bagi penelitian sejenis.

3.2 Analisis Sistem

Sistem ini dirancang untuk memudahkan atau membantu *user* dalam memprediksi kelayakan masuk jurusan IPA dari kelas X ke kelas XI.

Fitur-fitur penting yang disertakan dalam program prediksi penjurusan siswa adalah seperti fitur input data yang diberikan, fitur proses data, fitur menampilkan hasil prediksi, dan fitur untuk mencetak hasil prediksi.

Dalam penerapannya, aplikasi ini memiliki tiga menu utama yang terdiri dari:

1. Menu input merupakan menu yang digunakan untuk meng-input data-data nilai *user* yang akan digunakan untuk menghasilkan *decision tree*.
2. Menu prediksi merupakan menu yang digunakan untuk memproses data-data yang telah di *input*. Di menu ini akan ditampilkan *output* prediksi kelayakan masuk jurusan yang dihasilkan oleh algoritma C4.5.

3. Menu *testing*, pada menu ini *user* dapat mencocok-an hasil prediksi yang dihasilkan program dengan data-data yang telah ada hasil penjurusannya. Digunakan untuk mendapatkan persentase *error* program.

3.3 Rancangan Aplikasi

Sebelum masuk ke dalam tahap pembuatan atau pemrograman, aplikasi harus dirancang terlebih dahulu. Pada aplikasi ini, hal yang dapat dilakukan oleh pengguna adalah:

- a. Meng-*input* data *user* yang akan diproses untuk *testing* dan prediksi.
- b. Membuat *decision tree* yang akan digunakan untuk prediksi.
- c. Melihat hasil *testing* dan hasil prediksi jurusan yang didapatkan *user*.
- d. Melakukan *crosscheck* untuk mendapatkan persentase kesuksesan program.
- e. Mencetak hasil prediksi yang telah dihasilkan program.
- f. Melihat halaman bantuan yang berisikan kegunaan tombol-tombol dalam program.

3.3.1 Data Flow Diagram

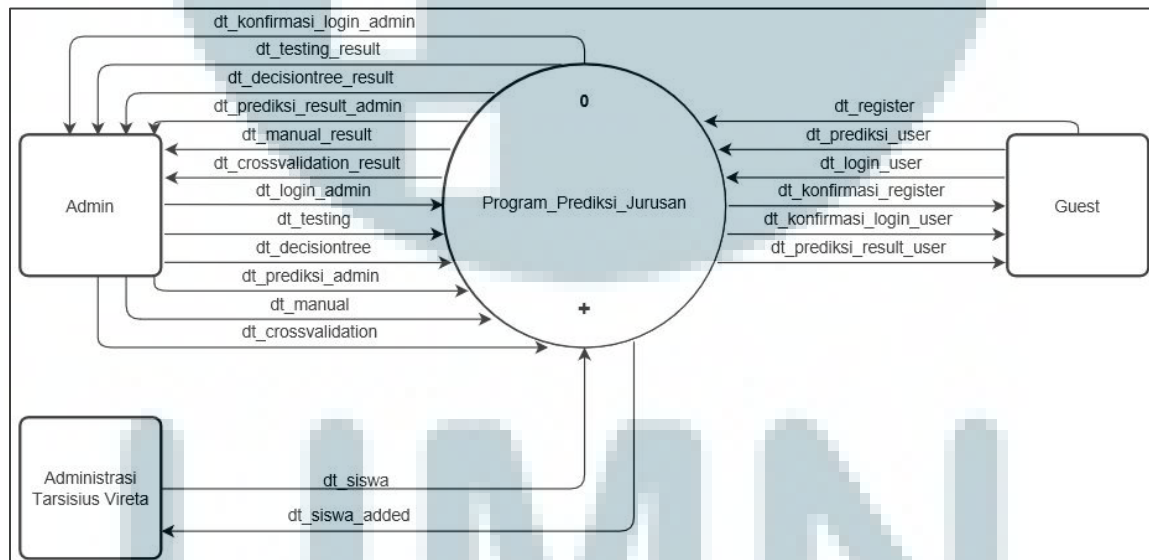
Data flow diagram adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Rubianto, 2013).

Pada *context diagram* yang ditunjukkan pada Gambar 3.1 terdapat tiga *entity*, yaitu “Admin”, “Administrasi Tarsisius Vireta”, dan “Guest”, dan satu proses yaitu “Program_Prediksi_Jurusan”.

Pada “Program_Prediksi_Jurusan” diterima enam data dari “Admin” berupa [dt_login_admin], [dt_testing], [dt_decisiontree], [dt_prediksi_admin], [dt_manual], dan [dt_crossvalidation]. “Program_Prediksi_Jurusan” juga menerima tiga data dari “Guest” berupa [dt_register], [dt_login_user], [dt_prediksi_user].

Sebaliknya “Program_Prediksi_Jurusan” mengirimkan enam data ke “Admin”, yaitu [dt_konfirmasi_login_admin], [dt_testing_result], [dt_decisiontree_result], [dt_prediksi_result_admin], [dt_manual_result], dan [dt_crossvalidation_result]. “Program_Prediksi_Jurusan” juga mengirimkan tiga data ke “Guest” yaitu [dt_konfirmasi_register], [dt_konfirmasi_login_user], [dt_prediksi_result_user].

Sementara “Program_Prediksi_Jurusan” mengirimkan satu data ke “Administrasi Tarsisius Vireta”, yaitu [dt_siswa_added] dan menerima satu data berupa [dt_siswa].



Gambar 3.1 Context Diagram Program

DFD Level satu yang terdapat pada Gambar 3.2 menunjukkan bahwa data yang dimasukkan *user* akan dilanjutkan ke dalam enam proses, yaitu “Proses_Login_Register”, “Proses_Testing”, “Proses_Decision_Tree”, “Proses_Prediksi”, “Proses_Crossvalidation”,

dan “Proses_Input_Data_Manual”. Proses_Login_Register menerima satu data dari *entity* “Admin” berupa [dt_login_admin] dan menerima dua data dari *entity* “Guest” berupa [dt_register] dan [dt_login_user], kemudian juga mengirimkan satu data ke *entity* “Admin” yaitu [dt_konfirmasi_login_admin] dan mengirimkan dua data ke *entity* “Guest” yaitu [dt_konfirmasi_register] dan [dt_konfirmasi_login_user], data-data ini akan disimpan di dalam tabel “Tabel_Login” untuk *user guest* dan “Tabel_Admin” untuk *user admin*.

Proses_Testing menerima satu data dari *entity* “Admin” berupa [dt_testing], sebaliknya juga mengirimkan satu data ke *entity* “Admin” yaitu [dt_testing_result]. Lalu proses ini juga menerima satu data dari *entity* “Administrasi Tarsisius Vireta” berupa [dt_siswa], serta mengirimkan satu data ke *entity* “Administrasi Tarsisius Vireta” yaitu [dt_siswa_added]. Proses ini tidak menerima dan mengirimkan data ke *entity* “Guest” dan data-data yang diterima dan dikirim disimpan dalam “Tabel_Testing” serta “Tabel_Training”.

Proses_Decision_Tree menerima satu data dari *entity* “Admin” berupa [dt_decisiontree] dan mengirimkan satu data ke *entity* “Admin” yaitu [dt_decisiontree_result]. Proses ini tidak menerima dan mengirimkan data ke *entity* “Guest” dan data-data yang diterima dan dikirim disimpan dalam “Tabel_Decision_Tree”.

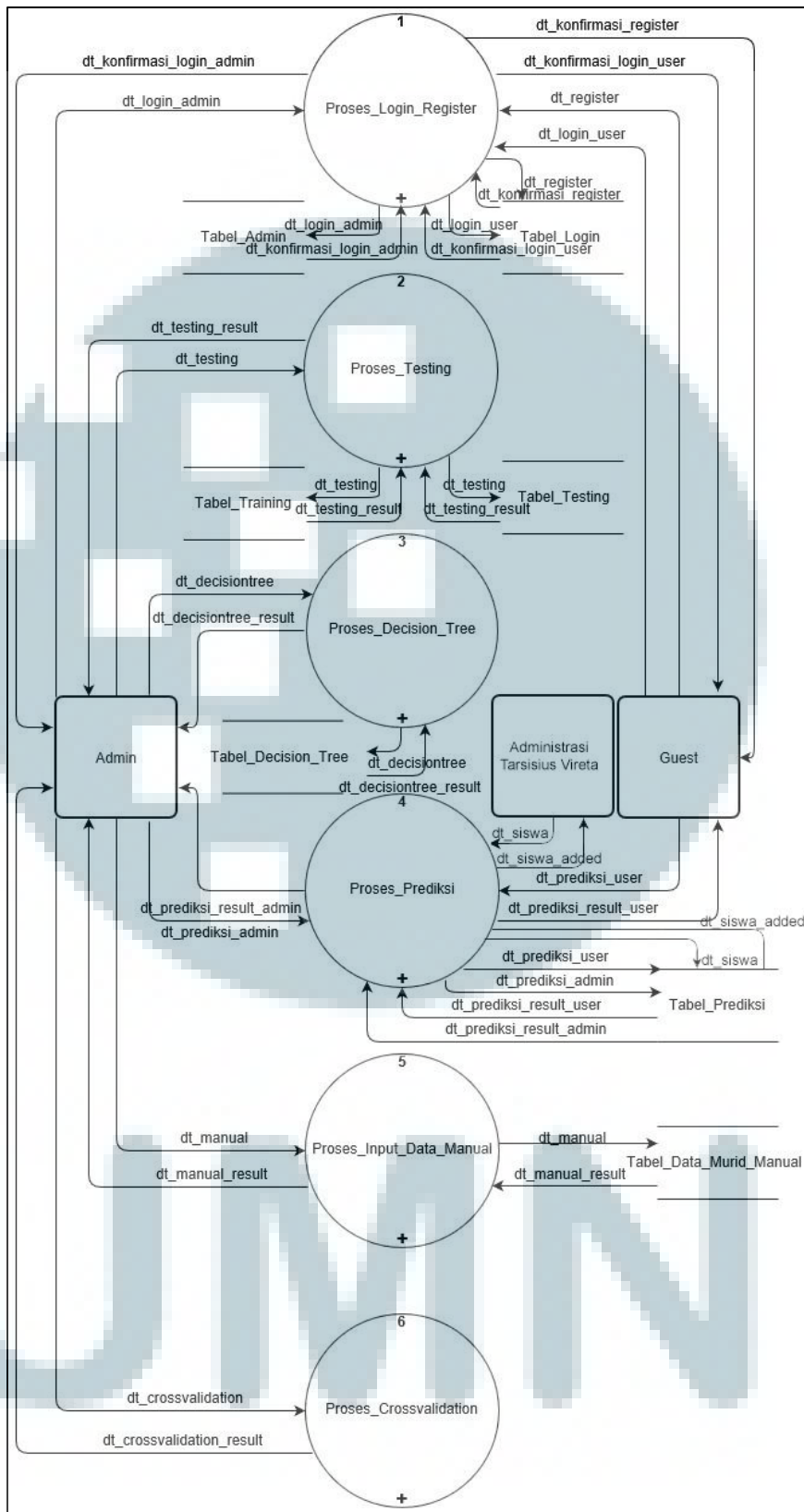
Proses_Prediksi menerima satu data dari *entity* “Admin” berupa [dt_prediksi_admin]. Proses ini juga menerima satu data dari *entity* “Guest” berupa [dt_prediksi_user]. Sebaliknya proses mengirimkan satu data ke *entity* “Admin” yaitu [dt_prediksi_result_admin], selain itu juga mengirimkan satu data ke *entity* “Guest” yaitu [dt_prediksi_result_user]. Semua data-data ini akan disimpan di dalam tabel “Tabel_Prediksi”.

Proses_Crossvalidation menerima satu data dari *entity* “Admin” berupa [dt_crossvalidation] dan mengirimkan satu data ke *entity* “Admin” yaitu [dt_crossvalidation_result]. Proses ini tidak menerima dan mengirimkan data ke *entity* “Guest”.

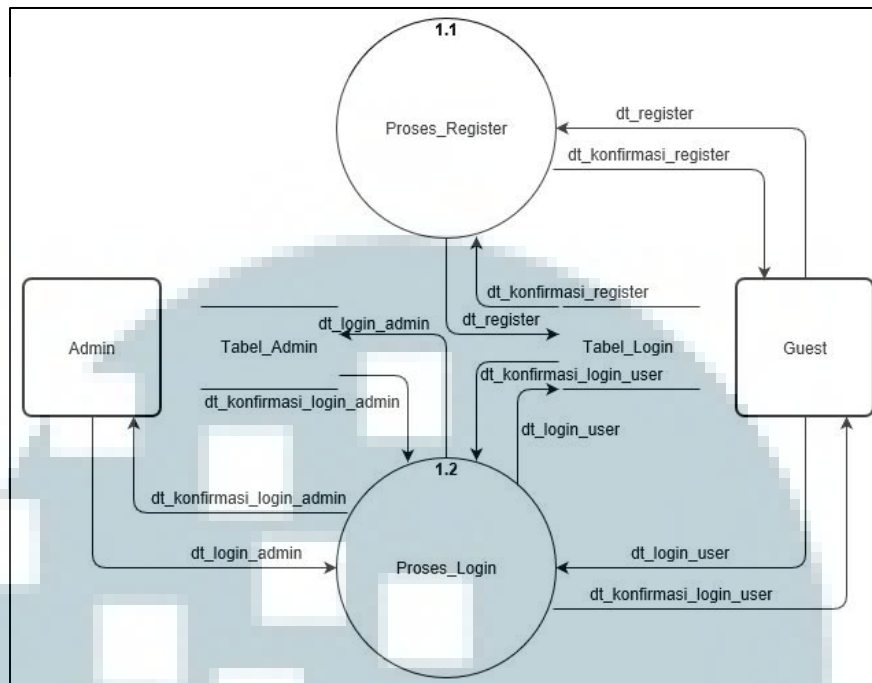
Proses_Input_Data_Manual menerima satu data dari *entity* “Admin” berupa [dt_manual], sebaliknya juga mengirimkan empat data ke *entity* “Admin” yaitu [dt_manual_result]. Proses ini tidak menerima dan mengirimkan data ke *entity* “Guest” dan data-data yang diterima dan dikirim disimpan dalam tabel “Tabel_Data_Murid_Manual”.

DFD level dua “Proses_Login_Register” menunjukkan bahwa data yang telah di-*input* pengguna akan dilanjutkan ke dalam dua proses, yaitu “Proses_Register” dan “Proses_Login”. Proses_Register menerima satu data dari *entity* “Guest” berupa [dt_register] dan mengirimkan satu data ke *entity* “Guest” yaitu [dt_konfirmasi_register], sedangkan Proses_Login menerima satu data dari *entity* “Admin” yaitu [dt_login_admin], kemudian juga mengirimkan satu data ke *entity* “Admin” yaitu [dt_konfirmasi_login_admin]. Proses Login juga menerima satu data dari *entity* “Guest” yaitu [dt_login_user], kemudian juga mengirimkan satu data ke *entity* “Admin” yaitu [dt_konfirmasi_login_user], data-data ini disimpan dalam “Tabel_Login” dan “Tabel_Admin”.

U
M
M
N

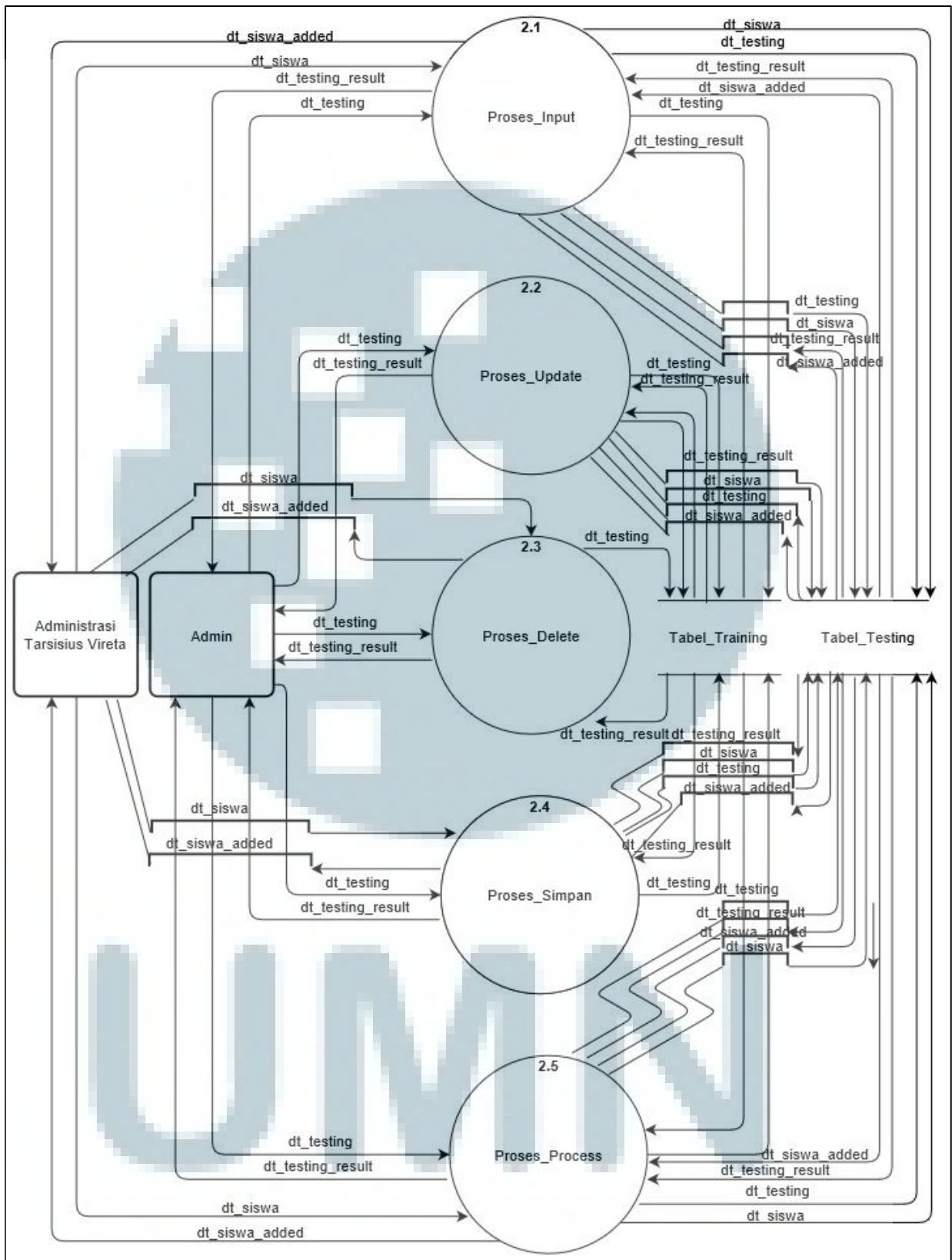


Gambar 3.2 DFD Level 1 Program



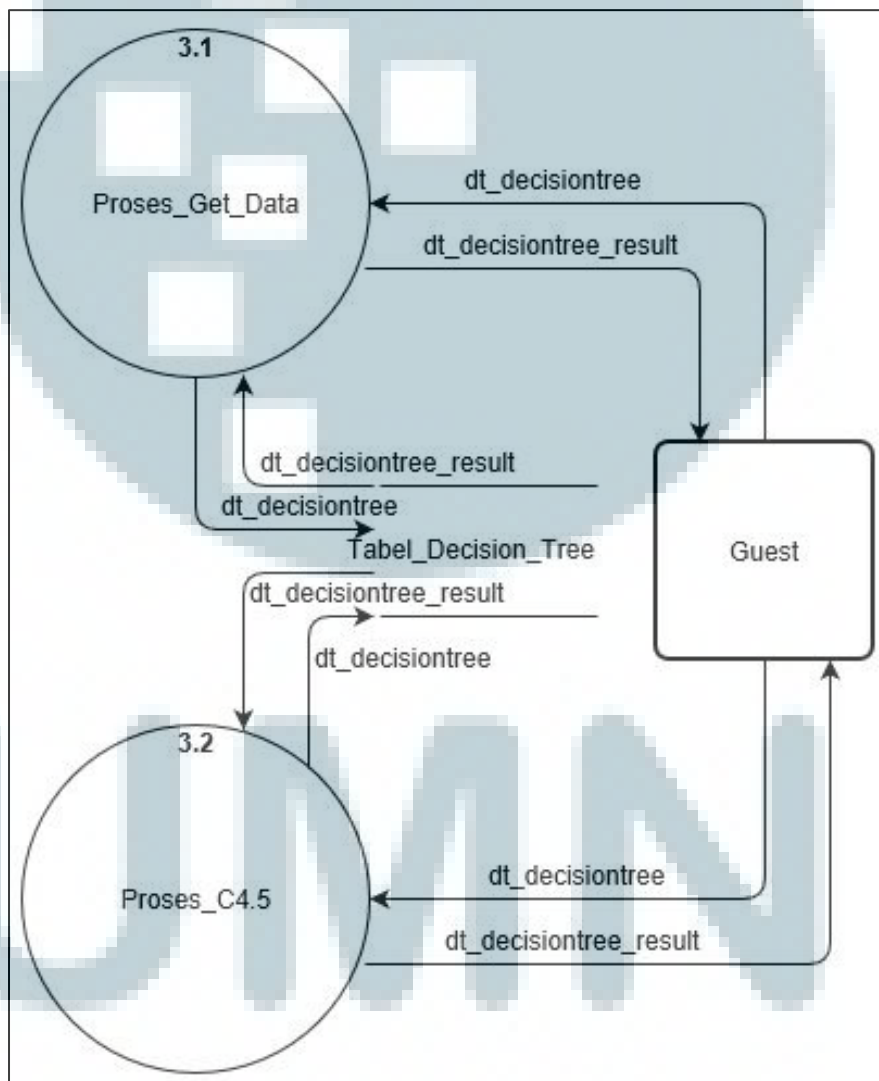
Gambar 3.3 DFD Level 2 Login Register

DFD level dua “Proses_Testing” menunjukkan bahwa data yang telah di-input pengguna akan dilanjutkan ke dalam lima proses yaitu “Proses_Input”, “Proses_Update”, “Proses_Delete”, “Proses_Simpan”, dan “Proses_Process”. Proses_Input menerima satu data dari *entity* “Admin” yaitu [dt_testing] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_testing_result]. Proses_Update menerima satu data dari *entity* “Admin” yaitu [dt_testing] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_testing_result]. Proses_Delete menerima satu data dari *entity* “Admin” yaitu [dt_testing] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_testing_result]. Proses_Simpan menerima satu data dari *entity* “Admin” yaitu [dt_testing] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_testing_result]. Proses_Process menerima satu data dari *entity* “Admin” yaitu [dt_testing] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_testing_result]. Semua data akan disimpan dalam “Tabel_Testing” serta “Tabel_Training”.



Gambar 3.4 DFD Level 2 Testing

DFD level dua “Proses_Decision_Tree” menunjukkan bahwa data yang di-input pengguna akan dilanjutkan ke dalam dua proses yaitu “Proses_Get_Data” dan “Proses_C4.5”. Proses_Get_Data menerima satu data dari entity “Admin” yaitu [dt_decisiontree] dan mengirimkan satu data ke entity “Admin” berupa [dt_decisiontree_result]. Proses_C4.5 menerima satu data dari entity “Admin” yaitu [dt_decisiontree] dan mengirimkan satu data ke entity “Admin” berupa [dt_decisiontree_result]. Semua data akan disimpan dalam “Tabel_Decision_Tree”.



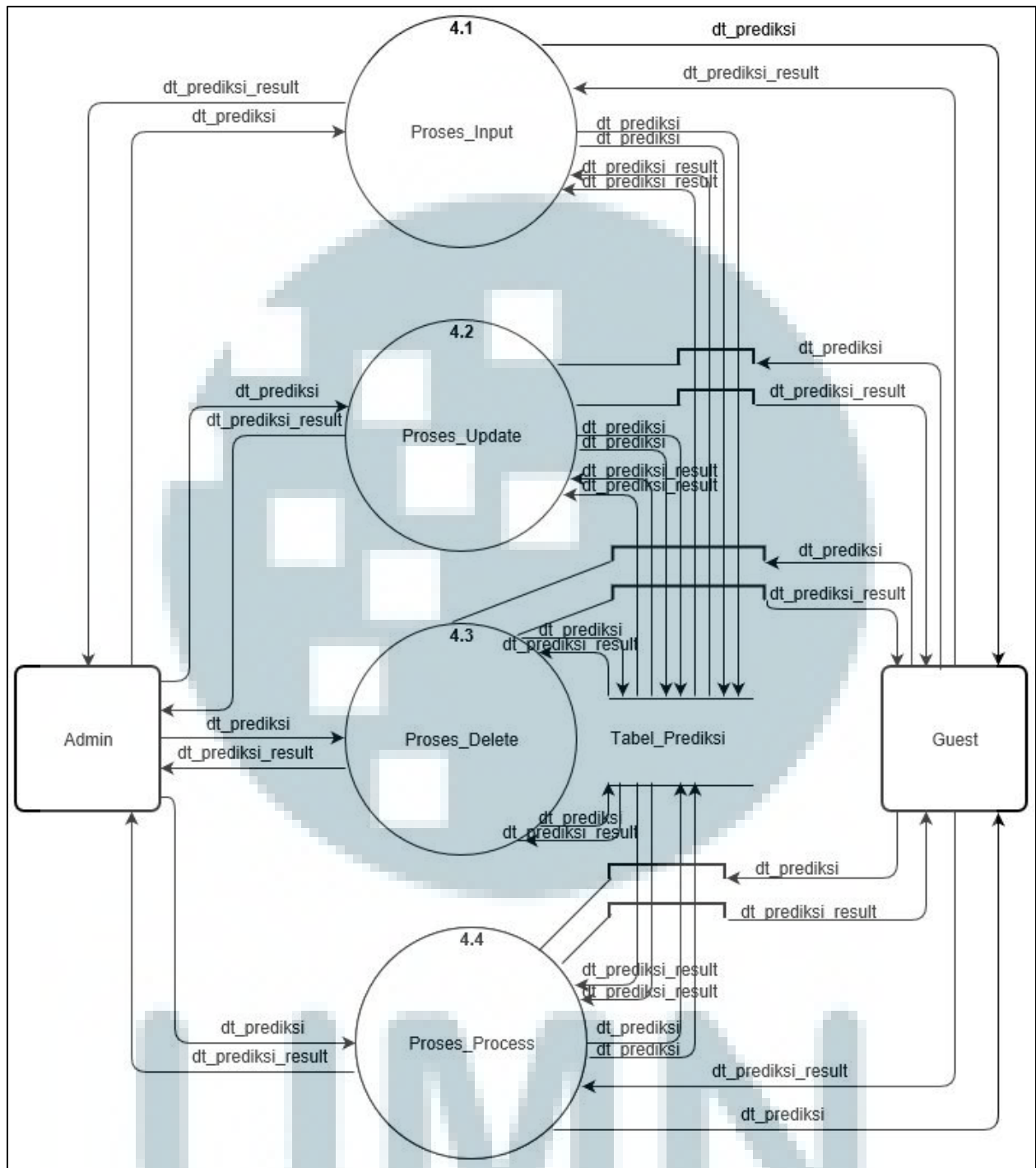
Gambar 3.5 DFD Level 2 Decision Tree

DFD level dua “Proses_Prediksi” menunjukkan bahwa data yang di-*input* pengguna akan dilanjutkan ke dalam empat proses yaitu “Proses_Input”, “Proses_Update”, “Proses_Delete”, dan “Proses_Process”. Proses_Input menerima satu data dari *entity* “Admin” yaitu [dt_prediksi_admin] dan satu data dari *entity* “Guest” yaitu [dt_prediksi_user], serta mengirimkan satu data ke *entity* “Admin” berupa [dt_prediksi_result_admin] dan satu data ke *entity* “User” berupa [dt_prediksi_result_user].

Proses_Update menerima satu data dari *entity* “Admin” yaitu [dt_prediksi_admin] dan satu data dari *entity* “Guest” yaitu [dt_prediksi_user] serta mengirimkan satu data ke *entity* “Admin” berupa [dt_prediksi_result_admin] dan satu data ke *entity* “User” berupa [dt_prediksi_result_user].

Proses_Delete menerima satu data dari *entity* “Admin” yaitu [dt_prediksi_admin] dan satu data dari *entity* “Guest” yaitu [dt_prediksi_user] serta mengirimkan satu data ke *entity* “Admin” berupa [dt_prediksi_result_admin] dan satu data ke *entity* “User” berupa [dt_prediksi_result_user].

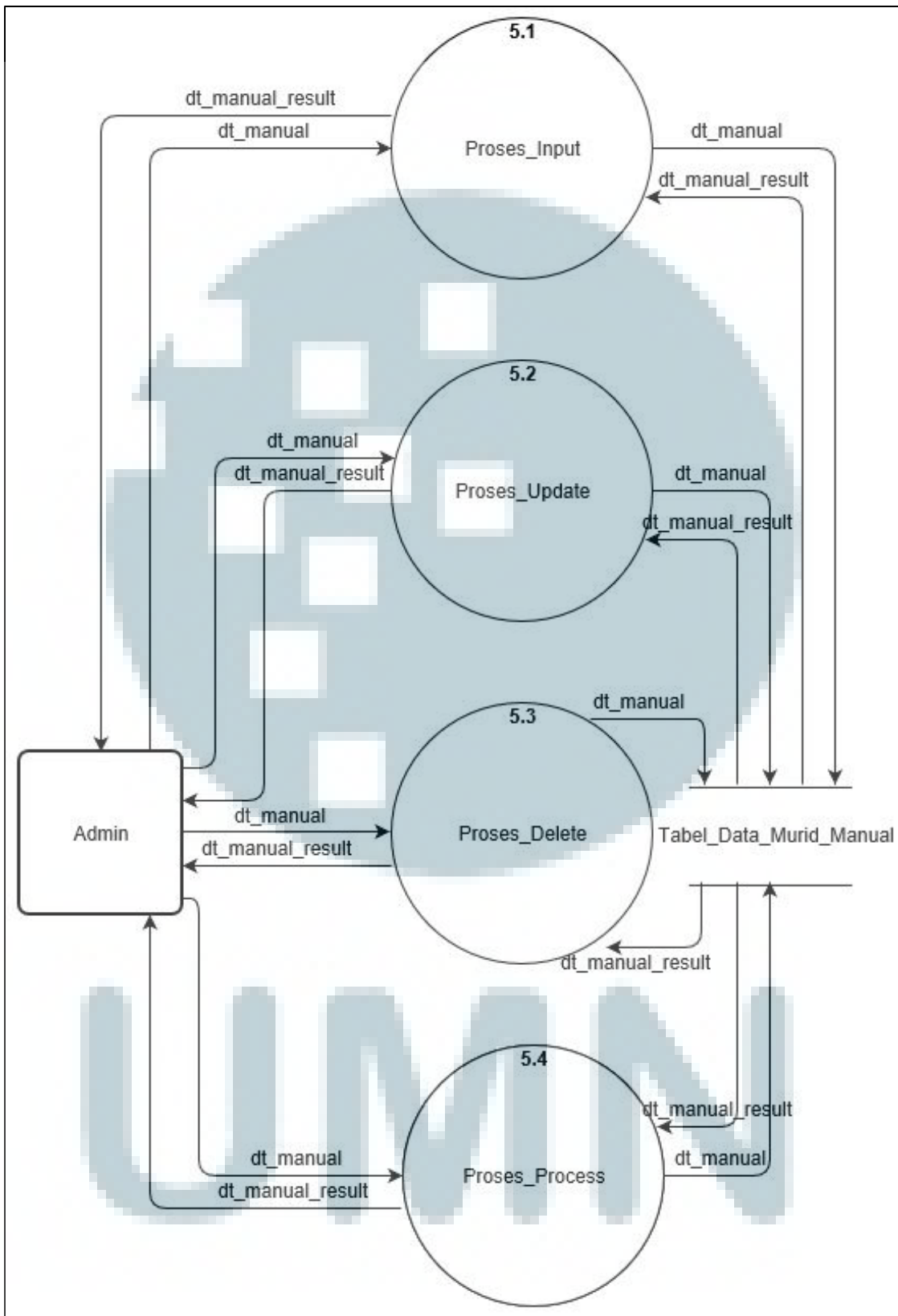
Proses_Process menerima satu data dari *entity* “Admin” yaitu [dt_prediksi_admin] dan satu data dari *entity* “Guest” yaitu [dt_prediksi_user] serta mengirimkan satu data ke *entity* “Admin” berupa [dt_prediksi_result_admin] dan satu data ke *entity* “User” berupa [dt_prediksi_result_user]. Semua data akan disimpan dalam tabel “Tabel_Prediksi”.



Gambar 3.6 DFD Level 2 Prediksi

DFD level dua “Proses_Input_Data_Manual” menunjukkan bahwa data yang di-*input* pengguna akan dilanjutkan ke dalam empat proses yaitu “Proses_Input”, “Proses_Update”, “Proses_Delete”, dan “Proses_Process”. Proses_Input menerima satu data dari *entity* “Admin” yaitu [dt_manual] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_manual_result]. Proses_Update menerima satu data dari *entity* “Admin” yaitu [dt_manual] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_manual_result]. Proses_Delete menerima satu data dari *entity* “Admin” yaitu [dt_manual] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_manual_result]. Proses_Process menerima satu data dari *entity* “Admin” yaitu [dt_manual] dan mengirimkan satu data ke *entity* “Admin” berupa [dt_manual_result]. Semua data akan disimpan dalam tabel “Tabel_Data_Murid_Manual”.

UMMN



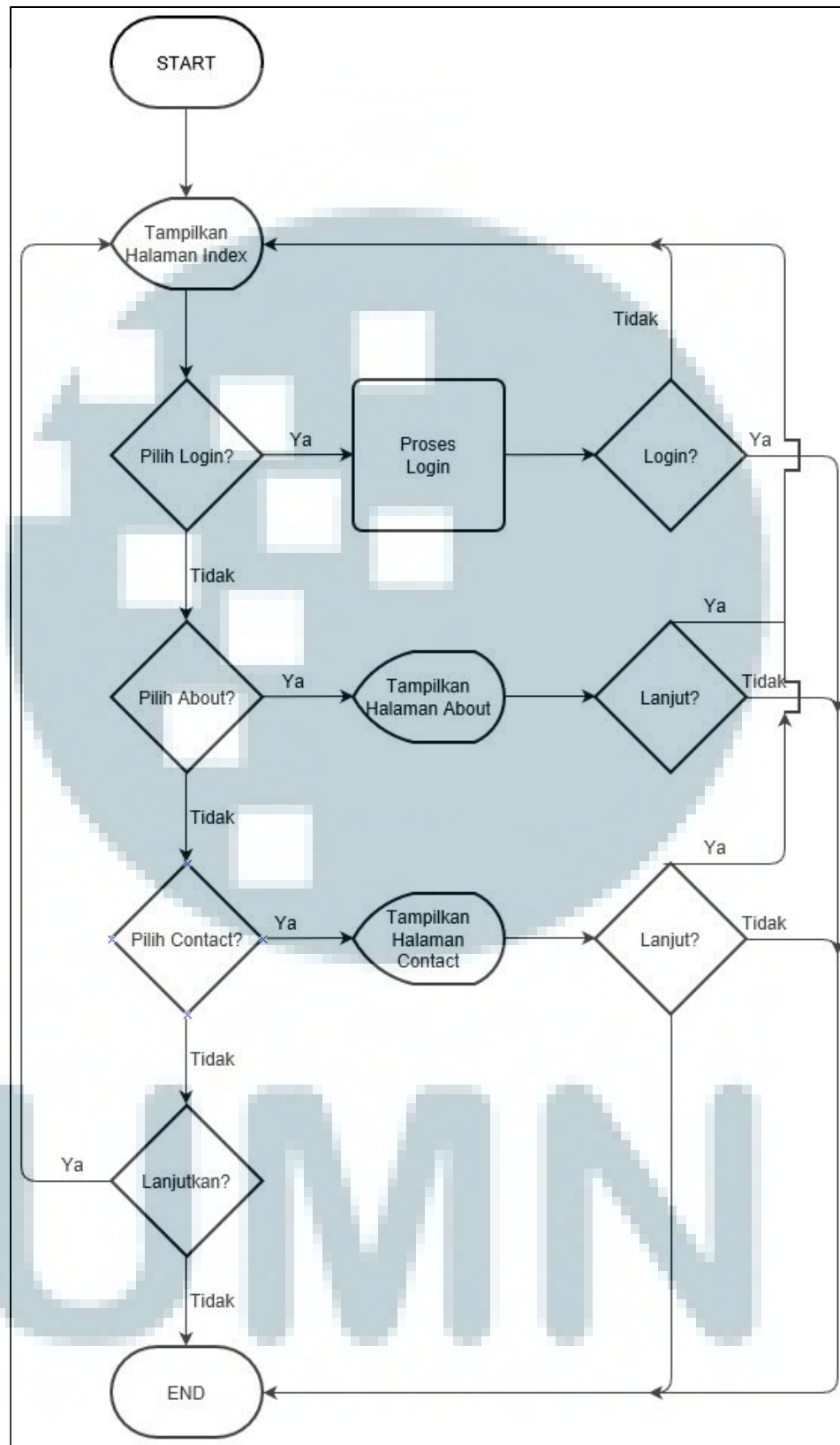
Gambar 3.7 DFD Level 2 Manual

3.3.2 Flowchart

Flowchart adalah cara penyajian visual aliran data melalui sistem informasi, operasi dilakukan dalam sistem dan urutan dimana mereka dilakukan. (Tanti, 2012).

Flowchart Index menunjukkan bahwa pertama *user* akan masuk ke “Halaman Index”, kemudian *user* diberikan pilihan, yaitu “Pilih *Login?*”, “Pilih *About?*”, dan “Pilih *Contact?*”. Apabila *user* memilih ya pada “Pilih *Login?*” maka akan ditampilkan “Proses Login” dan diberikan pilihan “*Login?*”, jika ya maka proses “End”. Jika tidak maka kembali ke “Halaman Index”, sedangkan jika pada proses pemilihan “Proses Login” dipilih tidak, maka dilanjutkan ke “Pilih *About?*”. Jika ya maka akan ditampilkan “Halaman *About?*” dan setelah proses selesai akan diberikan pilihan untuk “Lanjut?”, jika ya maka kembali ke “Halaman Index” dan jika tidak maka proses “End”, jika memilih tidak pada pemilihan “Pilih *About?*” maka dilanjutkan ke “Pilih *Contact?*”. Jika ya maka tampilkan “Halaman *Contact?*” dan setelah proses selesai akan diberikan pilihan untuk “Lanjut?”, jika ya maka kembali ke “Halaman Index” dan jika tidak maka proses “End”, jika memilih tidak pada pemilihan “Pilih *Contact?*” maka dilanjutkan ke proses pemilihan “Lanjutkan?”. Jika ya maka kembali ke “Halaman Index”, jika tidak maka proses akan “End”.

U
M
M
N

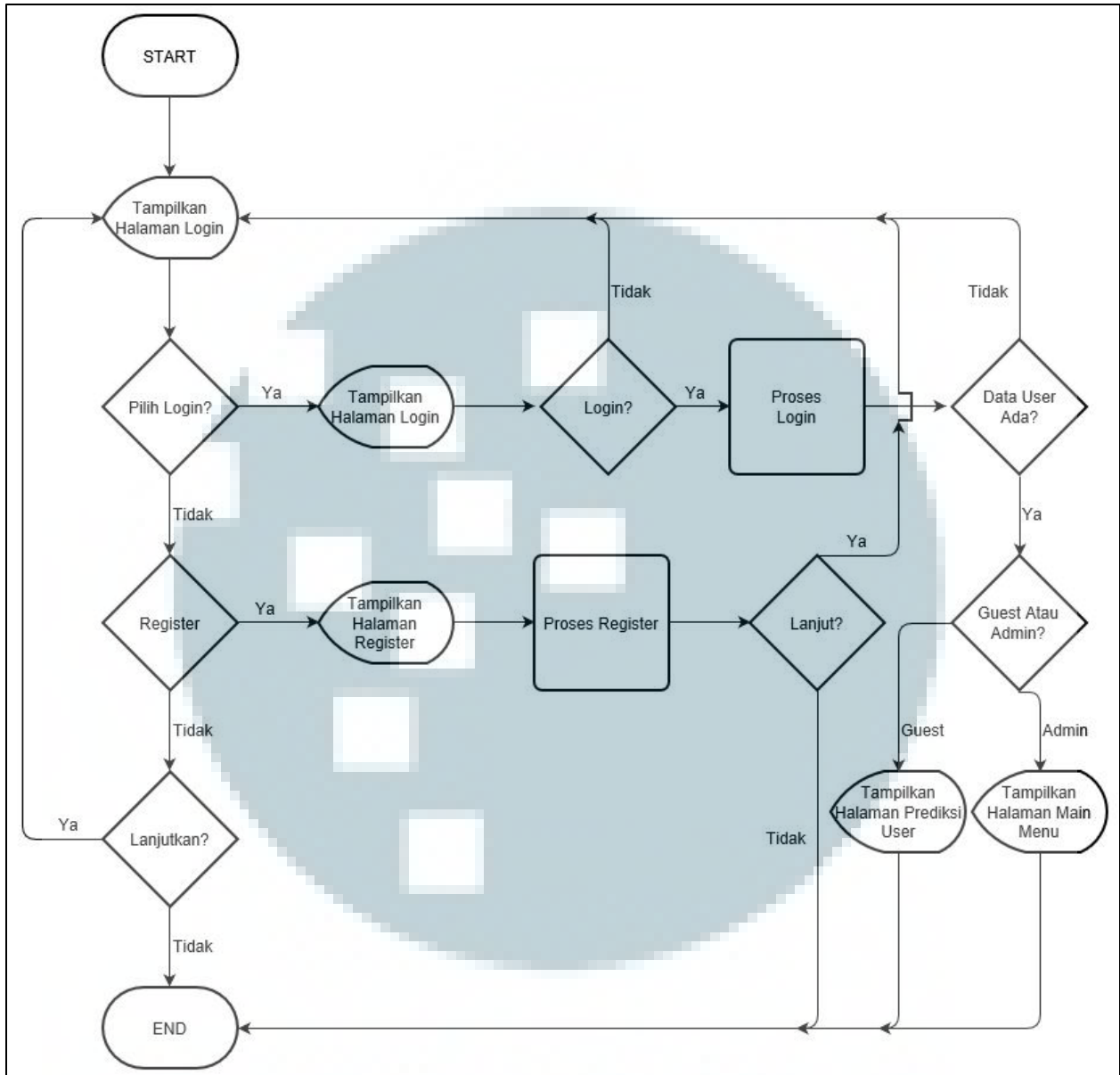


Gambar 3.8 *Flowchart Index*

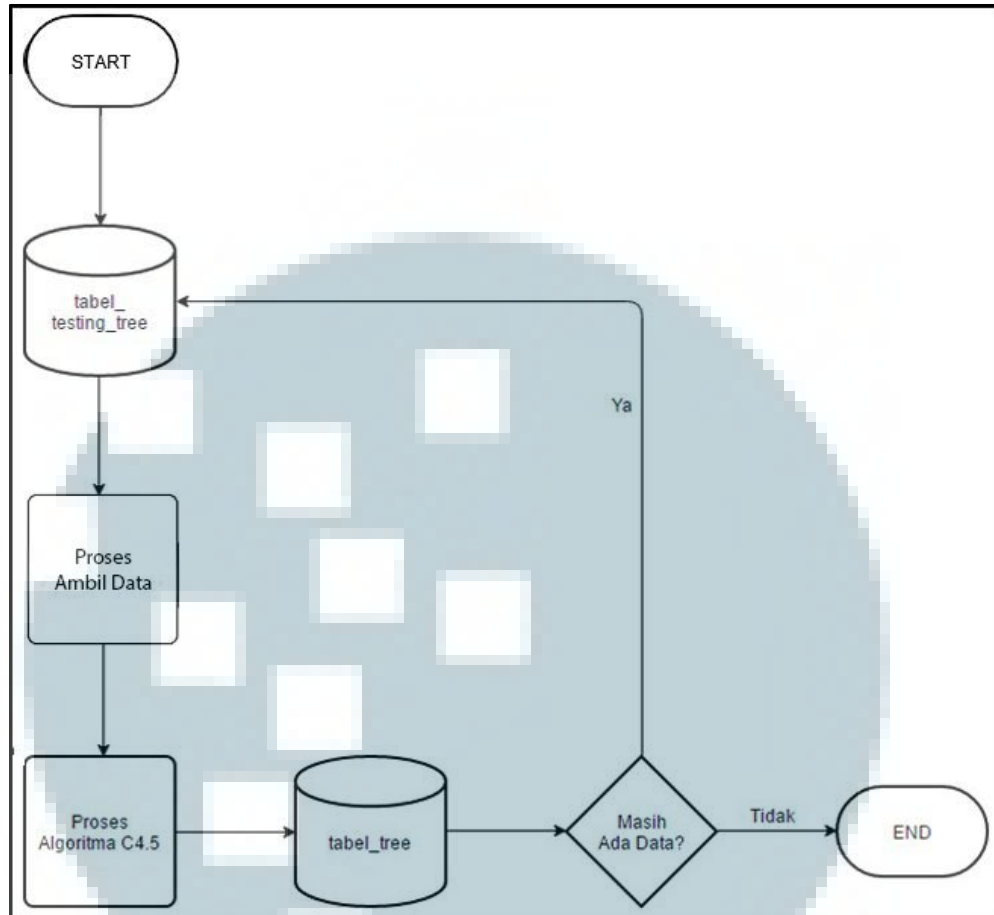
Flowchart Login menunjukkan bahwa setelah *user* masuk ke “Halaman Login” maka *user* dapat melakukan proses “Login”, disini data *user* akan dicek apakah sudah ada atau tidak, jika data *user* tidak ada maka *user* harus melakukan registrasi terlebih dahulu melalui proses “Registrasi *User*”, sedangkan apabila data *user* ada maka dilakukan proses pengecekan lagi, kali ini untuk mengetahui apakah *user login* sebagai *admin* atau *guest*. Apabila *user* masuk sebagai *guest* maka akan ditampilkan “Halaman Prediksi” versi *guest*., sedangkan jika pada saat proses “Login” memilih tidak maka proses akan “End”, sedangkan apabila *user* masuk sebagai *admin* maka akan ditampilkan “Halaman Utama”, kemudian proses “End”.

Flowchart testing menunjukkan agar aplikasi menghasilkan suatu “*Decision Tree*”, pertama akan menampilkan “Halaman *Testing*”. *User* kemudian dapat memilih untuk melanjutkan proses atau membatalkan proses. Jika lanjut maka *user* menginput “*Testing Data*”, jika tidak maka proses akan “*End*”. Kemudian data akan masuk ke “*Tabel_Testing*”, kemudian *user* dapat memilih untuk memulai proses “*Simpan Data*” Jika ya maka data yang disimpan di “*Tabel_Testing*” akan dimasukkan ke dalam “*Tabel_Training*”. Jika tidak maka kembali ke pemilihan “*Testing Data*”, kemudian setelah proses “*Simpan Data*” selesai, maka proses akan “*End*”.

U
M
M
N

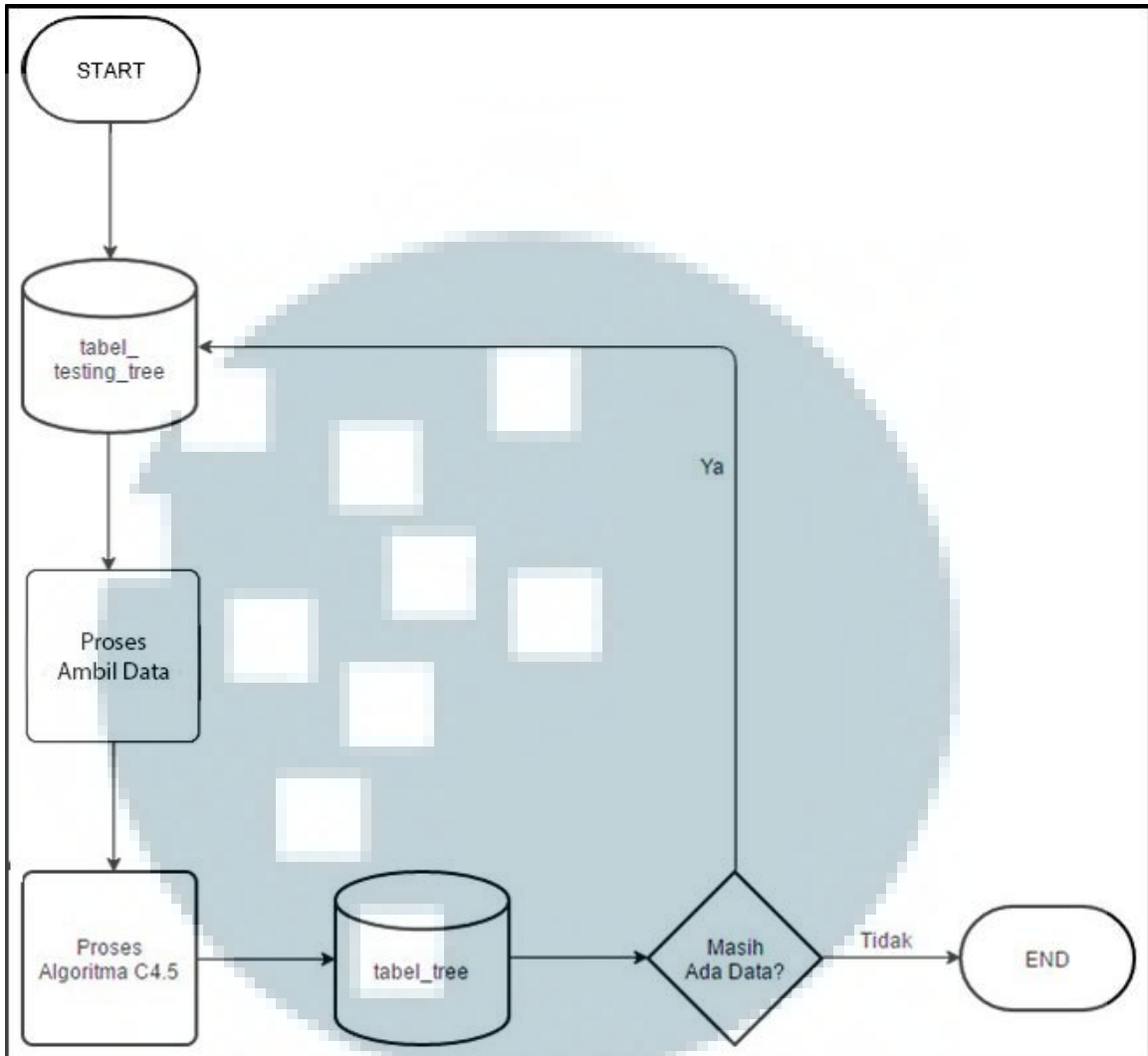


Gambar 3.9 Flowchart Login



Gambar 3.10 *Flowchart Testing*

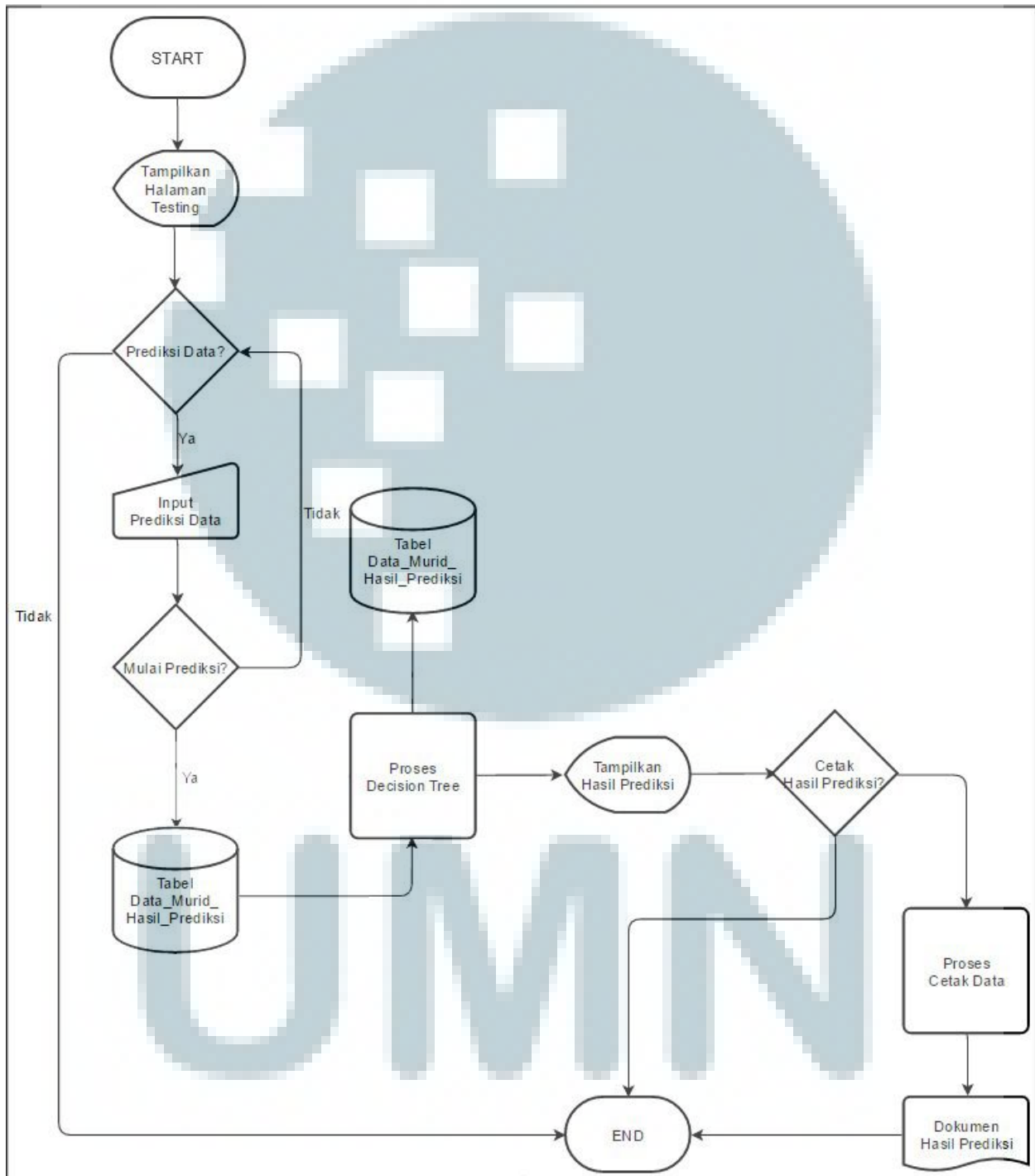
Flowchart training menunjukkan bahwa pertama program akan “START” kemudian dilakukan proses “Ambil Data” yang akan mengakses “Tabel_Training” selanjutnya data-data akan diproses oleh proses “Algoritma C4.5” untuk menghasilkan *decision tree*. Hasil dari proses akan disimpan di dalam tabel “Tabel_Decision_Tree”, kemudian akan dihadapkan pada pilihan “Masih Ada Data?”. Jika ya maka kembali lakukan proses “Ambil Data”, jika tidak maka proses akan “END”



Gambar 3.11 *Flowchart* Algoritma C4.5

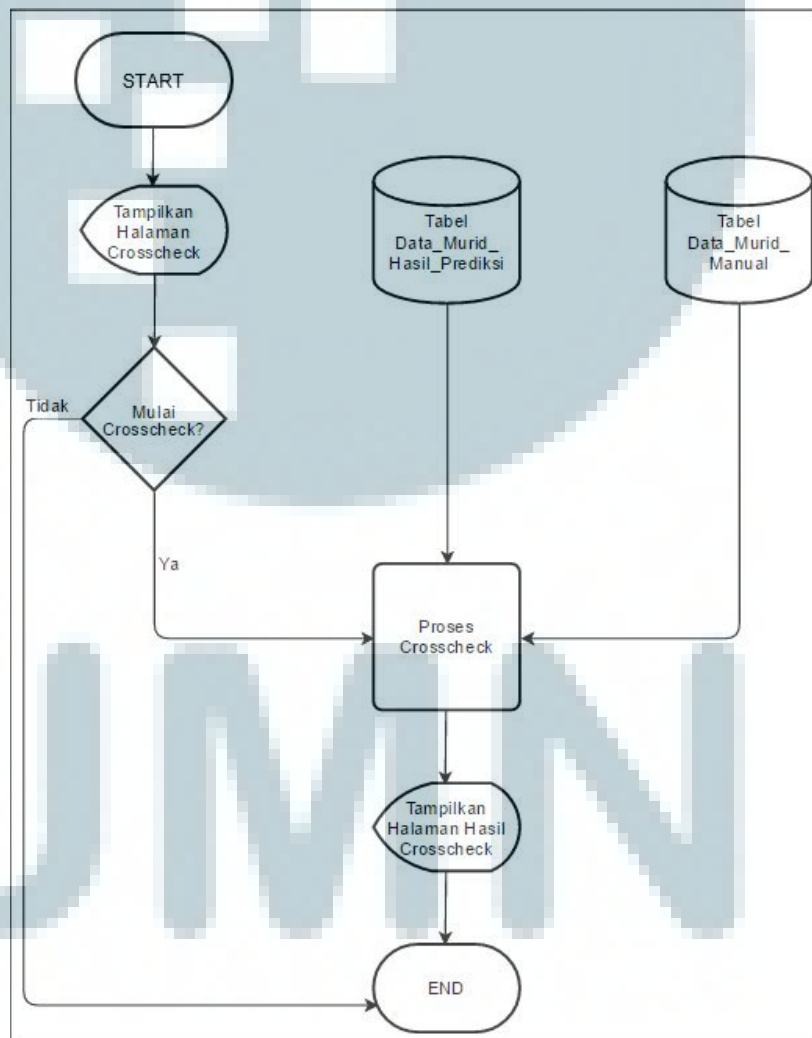
Flowchart prediksi menunjukkan agar bisa mendapatkan hasil prediksi data berupa jurusan siswa, dimulai dari menampilkan “Halaman Prediksi”. *User* kemudian dapat memilih untuk melanjutkan proses atau membatalkan proses. Jika lanjut maka *user* menginput “Prediksi Data”, jika tidak maka proses akan “End”. Data yang telah dimasukkan akan disimpan ke “Tabel_Prediksi”. Kemudian *user* dapat memilih untuk memulai prediksi, jika ya maka akan dilanjutkan dengan proses “Decision Tree” yang telah dihasilkan dari proses “Algoritma C4.5” sebelumnya. Jika tidak maka kembali ke

pemilihan “Prediksi Data”. Setelah proses “*Decision Tree*” selesai, maka data hasil proses akan dimasukkan ke “Tabel_Prediksi” dan akan menampilkan halaman “Hasil Prediksi” lalu proses akan “End”.



Gambar 3.12 *Flowchart* Prediksi

Flowchart cross validation menunjukkan agar bisa mendapatkan hasil *cross validation*, dimulai dari “Tampilkan Halaman *Cross validation*”, kemudian masuk proses “Mulai *Crossvalidation?*”, dimana data-data dari *database* yaitu “Tabel_Prediksi” dan “Tabel_Data_Murid_Manual” akan diambil untuk dilakukan proses “*Cross Validation*” yang kemudian menghasilkan *output* berupa persentase kesuksesan program, kemudian proses akan “End”.



Gambar 3.13 *Flowchart Cross Validation*

3.3.3 Struktur Tabel

Terdapat delapan tabel yang tidak berhubungan satu sama lain dengan tabel-tabel lainnya, yaitu “Tabel_Login”, “Tabel_Admin”, “Tabel_Testing”, “Tabel_Training”, “Tabel_Decision_Tree”, “Tabel_Prediksi”, dan “Tabel_Data_Murid_Manual”.

1. Nama Tabel: Tabel_Login

Fungsi: Menyimpan informasi dari data-data *user*.

Primary Key: ID

Foreign Key: -

Unique Key: email

Tabel 3.1 Struktur Tabel_Login

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	username	VARCHAR	50	<i>ID user</i>
3	email	VARCHAR	50	<i>Email user</i>
4	password	VARCHAR	50	<i>Password user</i>

2. Nama Tabel: Tabel_Admin

Fungsi: Menyimpan informasi dari data-data admin.

Primary Key: ID

Foreign Key: -

Unique Key: -

Tabel 3.2 Struktur Tabel_Admin

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	username	VARCHAR	50	ID admin
3	password	VARCHAR	50	<i>Password admin</i>

3. Nama Tabel: Tabel_Testing

Fungsi: Menyimpan data-data *user* yang akan dimasukan ke dalam Tabel_Training

Primary Key: ID

Foreign Key: -

Unique Key: nomor_induk

Tabel 3.3 Struktur Tabel_Testing

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	nama_user	INT	25	NIS Murid atau <i>user</i>
3	nomor_induk	VARCHAR	25	Nama Murid atau <i>user</i>
4	nilai_matematika	FLOAT	5	Nilai Matematika
5	nilai_fisika	FLOAT	5	Nilai Fisika
6	nilai_biologi	FLOAT	5	Nilai Kimia
7	nilai_kimia	FLOAT	5	Nilai Biologi
8	nilai_kognitif	FLOAT	5	Nilai Kognitif
9	nilai_psikomotorik	FLOAT	5	Nilai Psikomotorik
10	jurusan	VARCHAR	5	Jurusan

4. Nama Tabel: Tabel_Training

Fungsi: Menyimpan data-data *user* yang digunakan dalam proses pembuatan *decision tree*.

Primary Key: ID

Foreign Key: -

Unique Key: -

Tabel 3.4 Tabel_Training

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	nama_user	INT	25	NIS Murid atau <i>user</i>
3	nomor_induk	VARCHAR	25	Nama Murid atau <i>user</i>
4	nilai_matematika	FLOAT	5	Nilai Matematika
5	nilai_fisika	FLOAT	5	Nilai Fisika
6	nilai_biologi	FLOAT	5	Nilai Kimia
7	nilai_kimia	FLOAT	5	Nilai Biologi
8	nilai_kognitif	FLOAT	5	Nilai Kognitif
9	nilai_psikomotorik	FLOAT	5	Nilai Psikomotorik
10	jurusan	VARCHAR	5	Jurusan

5. Nama Tabel: Tabel_Decision_Tree

Fungsi: Menyimpan *decision tree* hasil proses algoritma C4.5.

Primary Key: ID

Foreign Key: -

Unique Key: gain_tree

Tabel 3.5 Struktur Tabel_Decision_Tree;

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	gain_tree	VARCHAR	25	Gain terendah
3	entropy_tree	VARCHAR	25	Entropy tertinggi
4	jurusan	VARCHAR	5	Jurusan hasil <i>decision tree</i>

6. Nama Tabel: Tabel_Prediksi

Fungsi: Menyimpan data-data *user* yang akan diprediksi, tabel ini juga digunakan untuk menampilkan prediksi jurusan yang didapat *user*.

Primary Key: ID

Foreign Key: -

Unique Key: nomor_induk

Tabel 3.6 Struktur Tabel_Prediksi

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	nama_user	INT	25	NIS Murid atau <i>user</i>
3	nomor_induk	VARCHAR	25	Nama Murid atau <i>user</i>
4	nilai_matematika	FLOAT	5	Nilai Matematika
5	nilai_fisika	FLOAT	5	Nilai Fisika
6	nilai_biologi	FLOAT	5	Nilai Kimia
7	nilai_kimia	FLOAT	5	Nilai Biologi
8	nilai_kognitif	FLOAT	5	Nilai Kognitif
9	nilai_psikomotorik	FLOAT	5	Nilai Psikomotorik
10	jurusan	VARCHAR	5	Jurusan yang didapat

7. Nama Tabel: Tabel_Data_Murid_Manual

Fungsi: Menyimpan informasi dari data-data *user*.

Primary Key: ID

Foreign Key: -

Unique Key: nomor_induk

Tabel 3.7 Struktur Tabel_Data_Murid_Manual

No	Nama Kolom	Data Type	Length	Keterangan
1	ID	INT		<i>Auto Increment</i>
2	nama_user	INT	25	NIS Murid atau <i>user</i>
3	nomor_induk	VARCHAR	25	Nama Murid atau <i>user</i>
4	nilai_matematika	FLOAT	5	Nilai Matematika
5	nilai_fisika	FLOAT	5	Nilai Fisika
6	nilai_biologi	FLOAT	5	Nilai Kimia
7	nilai_kimia	FLOAT	5	Nilai Biologi
8	nilai_kognitif	FLOAT	5	Nilai Kognitif
9	nilai_psikomotorik	FLOAT	5	Nilai Psikomotorik
10	jurusan	VARCHAR	5	Jurusan

3.3.4 Mock Ups

Sebelum pemrograman sistem dilakukan, *user interface* program harus dirancang terlebih dahulu. Hal ini dilakukan agar terdapat gambaran tampilan program secara keseluruhan, sehingga tampilan program dapat dibuat dengan rapi sesuai perancangan yang telah ditetapkan.

A. Halaman Login

Halaman *login* merupakan halaman dimana *user* memasukkan *username* dan *password* agar dapat masuk ke dalam menu utama program. Pada halaman *login* terdapat dua tombol yaitu *login*, dan *register*. Tombol *login* digunakan untuk melakukan pengecekan input *username* dan *password*. Tombol *register* digunakan untuk masuk ke halaman *register*, dimana *user* dapat mendaftarkan diri untuk kemudian mendapatkan akses masuk ke menu utama program.

The image shows a login form titled "PROGRAM PREDIKSI JURUSAN". It features two input fields: "Username" and "Password". Below the fields are two buttons: "LOGIN" and "REGISTER".

Gambar 3.14 *Mock Up* Halaman Login

B. Halaman Register

Halaman *register* merupakan halaman untuk mengisi data *user* baru yang ingin menggunakan program. Terdapat tiga *field* yang harus diisi yaitu *username*, *password*, dan *confirm password*. Tombol *register* digunakan untuk mengecek bahwa semua *field* terisi dengan benar, sedangkan tombol *back* digunakan untuk kembali ke halaman *login*.

The image shows a register form titled "PROGRAM PREDIKSI JURUSAN". It features a "BACK" button in the top left corner. Below it are three input fields: "Username", "Password", and "Confirm Password". At the bottom center is a "REGISTER" button.

Gambar 3.15 *Mock Up* Halaman Register

C. Halaman Utama

Halaman utama merupakan halaman pertama yang dijumpai *user* setelah *login*. Halaman ini berbeda satu sama lain untuk *user* yang mengakses program sebagai *guest* dan *user* yang mengakses sebagai *admin*. Untuk *user* admin terdapat enam tombol, yaitu tombol *testing*, tombol prediksi, tombol *crosscheck*, tombol *logoff*, tombol *about*, dan tombol *help*, sementara *user guest* hanya terdapat empat tombol, yaitu tombol prediksi, tombol *logoff*, tombol *about*, dan tombol *help*. Tombol *testing* digunakan untuk mengakses halaman *testing*, tombol prediksi untuk masuk halaman prediksi, tombol *logoff* untuk kembali ke halaman *login*, tombol *about* untuk menampilkan versi dan pembuat program, tombol *help* untuk masuk ke halaman bantuan.



Gambar 3.16 *Mock Up* Halaman Utama Admin



Gambar 3.17 *Mock Up* Halaman Utama User

D. Halaman Manual Input

Halaman *manual input* digunakan untuk memasukkan data-data murid secara manual untuk digunakan dalam proses perbandingan dengan data-data yang diperoleh dalam proses prediksi yang akan menghasilkan persentase kesuksesan program.



Gambar 3.18 *Mock Up* Halaman Manual Input

E. Halaman Testing

Halaman *testing* digunakan untuk memasukkan data-data murid ke database yang selanjutnya digunakan untuk mendapatkan *decision tree*. Terdapat empat tombol di halaman ini yaitu tombol *input* data, tombol *delete* data, tombol *testing*, dan tombol *back*. Tombol *input* digunakan untuk memasukkan data ke *database*, tombol *delete* digunakan untuk menghapus data dari *database*, tombol *testing* digunakan memulai proses *testing* dengan algoritma C4.5 untuk mendapatkan *decision tree*, dan tombol *back* untuk kembali ke halaman utama. Pada halaman ini juga terdapat satu *field* yang digunakan untuk menentukan nilai minimal untuk masuk jurusan IPA.

Sedangkan pada halaman *decision tree* terdapat dua tombol yaitu tombol *back* untuk kembali ke halaman *testing* dan tombol *save* untuk menyimpan hasil *decision tree* untuk digunakan dalam program prediksi jurusan.

BACK

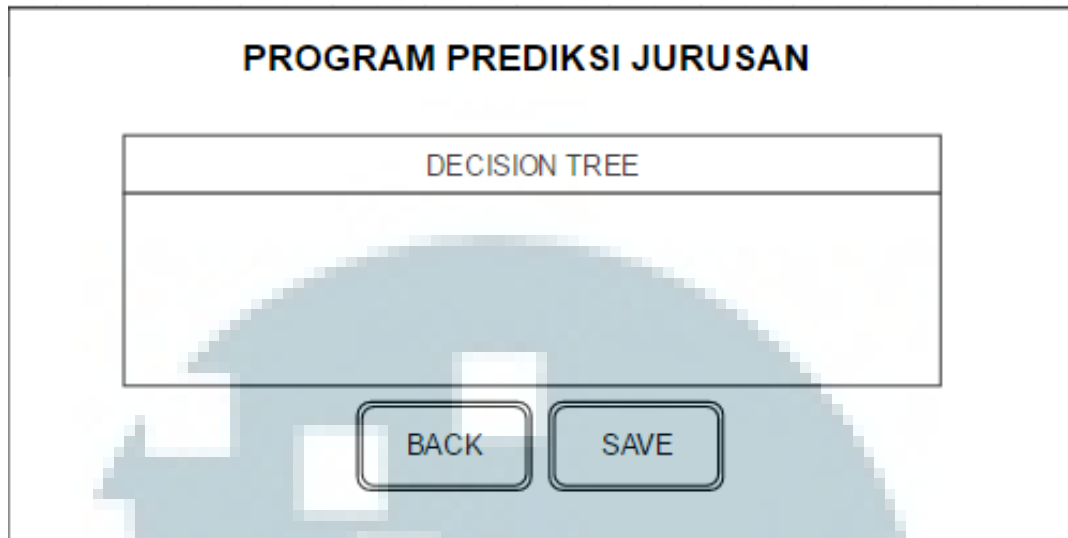
PROGRAM PREDIKSI JURUSAN

Minimal Nilai =

NIS	NAMA	MAT	FIS	BIO	KIM	BAKAT

INPUT DELETE TESTING

Gambar 3.19 *Mock Up* Halaman Testing



Gambar 3.20 *Mock Up* Halaman Decision Tree

F. Halaman Prediksi Jurusan

Pada halaman prediksi jurusan, *decision tree* yang telah didapatkan sebelumnya di halaman *testing* akan digunakan untuk memproses data-data yang dimasukkan *user* pada halaman *input* prediksi jurusan untuk hingga menghasilkan *output* berupa jurusan yang diprediksi sistem, apakah IPA atau IPS. Pada halaman *input* prediksi jurusan terdapat tiga tombol yaitu tombol *input*, tombol prediksi, dan tombol *back*. Tombol *input* digunakan untuk memasukkan data-data ke *database*, tombol prediksi digunakan untuk memulai proses prediksi jurusan, dan tombol *back* untuk kembali ke menu utama.

Sedangkan pada halaman *output* prediksi jurusan terdapat dua tombol, yaitu tombol *back* dan tombol cetak. Tombol *back* digunakan untuk kembali ke menu utama, sedangkan tombol cetak digunakan untuk mencetak data-data yang telah diproses.

PROGRAM PREDIKSI JURUSAN

NIS	NAMA	MAT	FIS	BIO	KIM	BAKAT

INPUT PREDIKSI

Gambar 3.21 *Mock Up* Halaman Input Prediksi Jurusan

PROGRAM PREDIKSI JURUSAN

NIS	NAMA	MAT	FIS	BIO	KIM	BAKAT	JURUSAN

BACK CETAK

Gambar 3.22 *Mock Up* Halaman Prediksi Jurusan

G. Halaman Cross Validation

Halaman *cross validation* adalah halaman yang digunakan untuk mencocokkan hasil prediksi jurusan yang didapatkan program dan data-data yang didapatkan secara manual yang akan digunakan untuk mendapatkan persentase kesuksesan program. Pada halaman ini terdapat dua tombol, yaitu tombol *back* dan tombol *cross validation*.

PROGRAM PREDIKSI JURUSAN

BACK

NIS	NAMA	MAT	FIS	BIO	KIM	BAKAT	JURUSAN

NIS	NAMA	MAT	FIS	BIO	KIM	BAKAT	JURUSAN

CROSSCHECK Hasil Crosscheck = %

Gambar 3.23 *Mock Up* Halaman Cross Validation

H. Halaman Help

Pada halaman *help*, *user* diberikan penjelasan mengenai tombol-tombol yang ada di program prediksi jurusan.

PROGRAM PREDIKSI JURUSAN

BACK

INPUT Digunakan untuk memasukan data ke database

DELETE Digunakan untuk menghapus data dari database

TESTING Digunakan untuk membuat decision tree

SAVE Digunakan untuk menyimpan decision tree untuk digunakan pada proses prediksi jurusan

Gambar 3.24 *Mock Up* Halaman Help

I. Halaman About

Pada halaman *about*, *user* dapat melihat nomor versi program dan identitas pihak pembuat program.



Gambar 3.25 *Mock Up* Halaman About

3.4 Program Prediksi Kelayakan Masuk Jurusan IPA

Dalam pembuatan program ini akan digunakan algoritma C4.5 dengan dua *class*, yaitu lulus dan gagal, untuk menghasilkan solusi dari jawaban yang diberikan *user* pada pertanyaan-pertanyaan yang akan ditampilkan kepada *user*. Program akan berbasis *web* sehingga akan ditulis dengan bahasa pemrograman *PHP* dibantu dengan *JavaScript*. Pembuatan *user interface* juga akan mengandalkan *CSS*. Studi kasus ke SMA Tarsisius Vireta Tangerang digunakan untuk memperoleh data-data murid yang selanjutnya digunakan sebagai pembandingan untuk mendapatkan persentase kesuksesan program serta evaluasi hasil penelitian.

Database yang akan digunakan adalah SQL dengan menggunakan *third-party software* yaitu *xampp* yang akan dihubungkan dengan program yang akan dibuat, *database* akan berisikan data-data berupa data-data murid seperti nama, nilai-nilai pelajaran yang penting untuk IPA seperti Matematika, Fisika, Biologi, dan Kimia beserta nilai Kognitif dan Psikomotorik.

Data-data yang akan dimasukkan ke dalam *database* untuk kemudian diproses berupa data-data nama murid atau *user*, nilai Kognitif, nilai Psikomotorik, nilai Matematika, nilai Fisika, nilai Biologi, dan nilai Kimia.

Untuk *user interface*, akan dimulai dengan halaman *login* dan *register*. Setelah *user* masuk, maka ditampilkan halaman utama yang disesuaikan dengan tipe *user*, apakah *guest* atau *admin*, lalu diberikan tombol untuk masuk ke halaman *testing* atau prediksi atau *crosscheck* untuk *admin*, sedangkan untuk *guest* hanya tombol untuk masuk halaman prediksi. Setelah masuk, program akan menampilkan tabel yang akan digunakan *user* untuk memasukkan data-data yang akan diproses, baik untuk *testing* maupun prediksi. Untuk *testing* diberikan tombol-tombol seperti *input data*, *update data*, *delete data*, *testing data*, dan *save data*. Berikut penjelasan singkat mengenai tombol-tombol tersebut:

1. Input Data

Tombol ini digunakan untuk memasukkan data-data murid ke dalam *database*. Data-data yang perlu dimasukkan berupa nama murid atau *user*, NIM murid atau *user*, nilai-nilai untuk mata pelajaran utama jurusan IPA, minat jurusan yang diinginkan murid, dan sebagainya.

Nilai-nilai yang diambil adalah nilai Kognitif, nilai Psikomotorik, nilai Matematika, nilai Fisika, nilai Kimia, dan nilai Biologi dengan perhitungan nilai 30% tugas, 30% ujian tengah semester, dan 40% ujian akhir semester. Nilai-nilai ini juga merupakan campuran dari nilai hasil perhitungan selama dua semester dibagi dua untuk mendapatkan rata-rata nilai.

1. Nama murid atau user
2. NIM murid atau user
3. Rata-rata nilai kognitif selama dua semester
4. Rata-rata nilai psikomotorik selama dua semester
5. Rata-rata nilai matematika selama dua semester
6. Rata-rata nilai fisika selama dua semester
7. Rata-rata nilai kimia selama dua semester
8. Rata-rata nilai biologi selama dua semester

Gambar 3.26 Variabel-variabel data program *testing*

2. Update Data

Tombol ini digunakan untuk mengubah atau memodifikasi data-data yang sudah ada sebelumnya di dalam *database*.

3. Delete Data

Tombol ini digunakan untuk menghapus data yang terdapat di dalam *database*.

4. Testing Data

Tombol ini digunakan untuk memulai proses pembuatan *decision tree* dengan memproses data-data yang telah dimasukkan untuk proses *testing*. Selanjutnya didapatkan nilai *gain* dan *entropy*, yang kemudian digunakan untuk menghasilkan *output* berupa *decision tree*.

5. Save Data

Tombol ini digunakan untuk menyimpan *decision tree* yang telah dihasilkan oleh proses *testing*.

Sedangkan untuk prediksi diberikan tombol-tombol seperti *input data*, *update data*, *delete data*, *prediksi data*, dan *cetak data*. Berikut penjelasan singkat mengenai tombol-tombol tersebut:

1. Input Data

Tombol ini digunakan untuk memasukkan data-data murid ke dalam *database*. Data-data yang perlu dimasukkan berupa nama murid atau *user*, NIM murid atau *user*, nilai-nilai untuk mata pelajaran utama jurusan IPA, minat jurusan yang diinginkan murid, dan sebagainya.

Nilai-nilai yang diambil adalah nilai Kognitif, nilai Psikomotorik, nilai Matematika, nilai Fisika, nilai Kimia, dan nilai Biologi dengan perhitungan nilai 30% tugas, 30% ujian tengah semester, dan 40% ujian akhir semester. Nilai-nilai ini juga merupakan campuran dari nilai hasil perhitungan selama dua semester dibagi dua untuk mendapatkan rata-rata nilai.

2. Update Data

Tombol ini digunakan untuk mengubah atau memodifikasi data-data yang sudah ada sebelumnya di dalam *database*.

3. Delete Data

Tombol ini digunakan untuk menghapus data yang terdapat di dalam *database*.

4. Prediksi Data

Tombol ini digunakan untuk memulai proses uji kelayakan dengan memproses data-data dalam *database* dengan menggunakan algoritma C4.5 yang telah diberikan dua *class* sebelumnya, yaitu lulus dan gagal, serta menggunakan *decision tree* yang telah dihasilkan dari proses *testing*. Data-data yang telah diproses akan menghasilkan solusi berupa data-data murid yang dinyatakan lulus uji kelayakan masuk jurusan IPA dan data-data murid yang dinyatakan gagal uji kelayakan masuk jurusan IPA.

5. Cetak Data

Tombol ini digunakan untuk mencetak hasil prediksi jurusan yang telah dihasilkan oleh program.

Sementara untuk *crosscheck* diberikan satu tombol, yaitu tombol *crosscheck* data dengan penjelasan singkat sebagai berikut:

1. Cross Validation Data

Tombol ini digunakan untuk melakukan proses *crosscheck* dengan membandingkan data-data hasil prediksi program dan data-data yang dimasukkan secara manual, untuk kemudian didapatkan persentase kesuksesan program.

1. Nama murid atau user
2. Rata-rata nilai matematika selama dua semester
3. Rata-rata nilai fisika selama dua semester
4. Rata-rata nilai kimia selama dua semester
5. Rata-rata nilai biologi selama dua semester
6. Rata-rata nilai bakat selama dua semester
7. Jurusan yang diperoleh

Gambar 3.27 Variabel-variabel data program *cross validation*