

BAB 3 METODOLOGI PENELITIAN

3.1 Analisis Kebutuhan

Dalam pembuatan suatu aplikasi, perlu untuk mengetahui hal-hal apa saja yang menjadi kebutuhan pengguna/*user*. Menyikapi hal ini, dilakukan analisis kebutuhan pengguna yang akan menjadi acuan dalam pembuatan sistem rekomendasi, sehingga dapat menghasilkan aplikasi yang sesuai dengan kebutuhan *user*. Dari hasil analisis, yang dibutuhkan adalah nilai para peserta didik Sekolah Dian Harapan Manado untuk menghasilkan rekomendasi mata pelajaran yang sesuai. Nilai-nilai tersebut akan digunakan dalam sistem sebagai acuan untuk mendapatkan hasil rekomendasi mata pelajaran. Tidak hanya melakukan analisis kebutuhan pengguna tetapi juga melakukan analisis kebutuhan untuk membangun *website*. Tahapan untuk membangun *website* adalah sebagai berikut:

1. *User Requirement* Tahapan pertama yang dilakukan yaitu menentukan *user requirement*, untuk mengetahui *user requirement* dilakukan proses wawancara kepada pihak sekolah Dian Harapan Manado. Proses wawancara dilakukan secara *online* dengan narasumber Christiwiranty V. R. Sianturi dari pihak Sekolah Dian Harapan Manado. Hasil dari proses wawancara, di simpulkan bahwa *user requirement* yang dibutuhkan adalah sistem yang dapat membantu siswa untuk melihat hasil rekomendasi mata pelajaran yang didapatkan oleh sekolah. Sistem yang dimaksud berisi informasi mengenai hasil rekomendasi mata pelajaran dan rekomendasi jurusan yang dapat menjadi bahan pertimbangan bagi siswa ketika ingin melanjutkan ke perguruan tinggi.
2. Studi Literatur Tahapan selanjutnya yang dilakukan adalah studi literatur, dalam tahapan ini akan dilakukan proses untuk mengumpulkan ilmu yang akan digunakan untuk rancang dan bangun sistem rekomendasi mata pelajaran. ilmu yang akan mengenai WASPAS akan dikumpulkan untuk pembuatan sistem dan *End User Computing Satisfaction* akan dikumpulkan untuk mengetahui kepuasan pengguna. Sumber literatur yang akan digunakan diperoleh dari jurnal atau artikel maupun sumber lainnya yang berasal dari internet.

3. Perancangan Sistem

Dalam tahapan perancangan sistem, dibagi menjadi dua bagian. Bagian yang pertama adalah tahapan perancangan visualisasi sistem rekomendasi, pada tahapan ini dibuat visualisasi alur dari sistem rekomendasi dengan menggunakan *flowchart*. Bagian kedua merupakan perancangan *mock-up website* sesuai dengan sistem rekomendasi mata pelajaran. Perancangan *mock-up* akan menjadi antarmuka dalam sistem yang dibuat menggunakan aplikasi Figma. Melalui proses perancangan sistem ini akan menghasilkan dasar acuan akan tahapan berikutnya, yaitu proses pembangunan sistem. Dengan adanya dasar acuan yang tepat, maka semakin mempermudah tahapan selanjutnya.

4. Pembangunan Sistem

Setelah dilakukan beberapa tahapan mulai dari analisis hingga perancangan sistem, proses pembangunan sistem dilakukan berdasarkan acuan yang dimiliki. Tahapan ini meliputi pembangunan sistem rekomendasi mata pelajaran dengan menggunakan ReactJS dan TailwindCSS untuk bagian *front-end* dan ExpressJS untuk bagian *back-end* serta untuk bagian *database* akan menggunakan PostgreSQL.

5. Pengujian

Tahap selanjutnya setelah membuat *website* yaitu melakukan *testing website*, metode yang digunakan untuk *testing website* adalah functional testing atau bisa disebut Black box. Proses yang dilakukan adalah menguji apakah setiap fitur yang telah dibuat berjalan dengan baik.

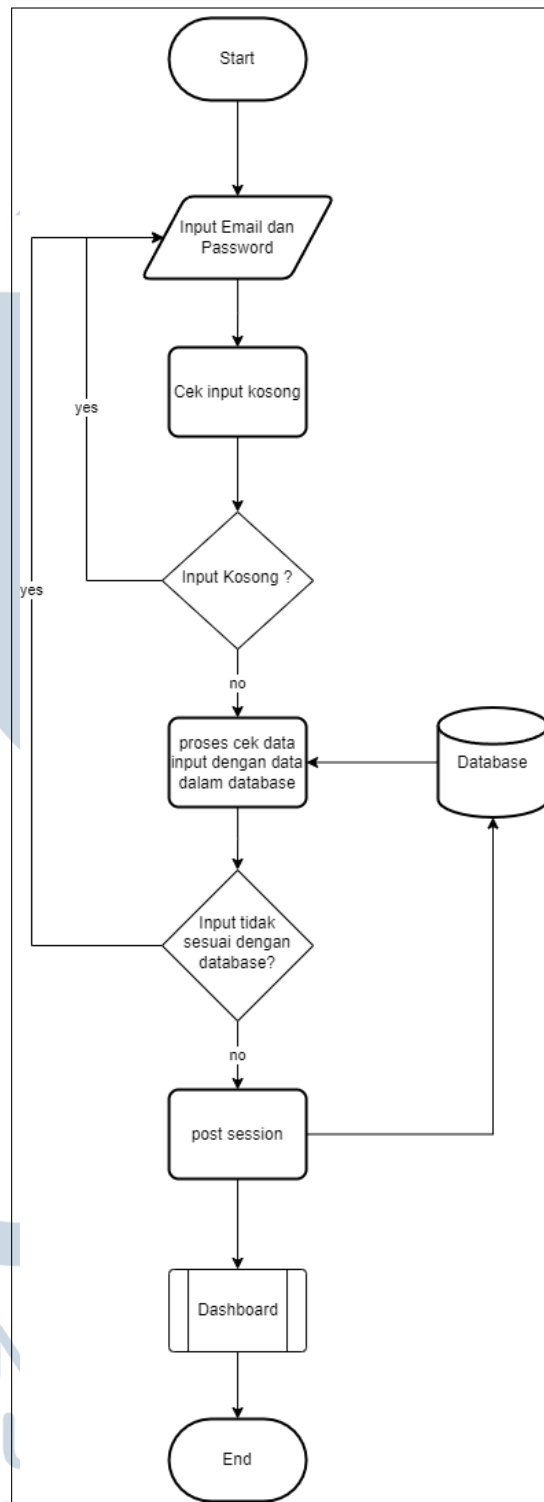
6. Evaluasi

Sistem yang telah dibangun kemudian diberikan kepada *user* untuk digunakan, setelah menggunakan sistem rekomendasi mata pelajaran maka *user* perlu memberikan umpan balik. Umpan balik ini berupa pengisian *Google Form* dengan pertanyaan menggunakan metode *End User Computing Satisfaction* dengan penilaian menggunakan Skala Likert. Hasil dari umpan balik menjadi bahan evaluasi terhadap sistem rekomendasi mata pelajaran yang ada, juga berkaitan dengan tingkat kepuasan pengguna terhadap sistem tersebut.

3.2 Flowchart Diagram

Untuk memvisualisasikan alur atau proses dari sebuah *website* dibutuhkan *flowchart*. Dalam bagian ini untuk memvisualisasikan proses dari *website* akan menggunakan *flowchart*. Proses yang akan divisualisasikan berupa proses *Login*, *Add Subject*, *Add Major*, *Add User* dan sistem rekomendasi.

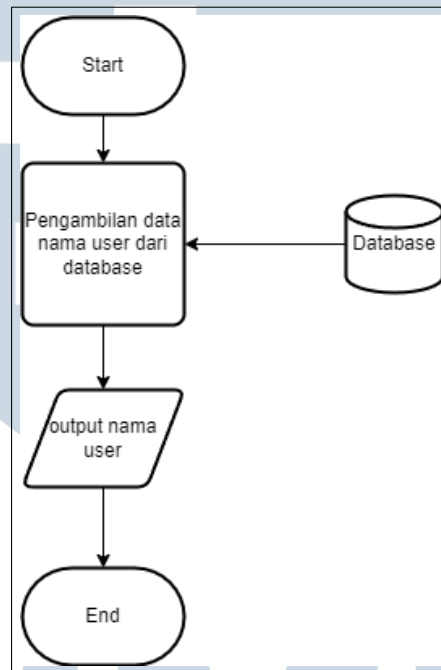




Gambar 3.1. Flowchart Login Admin

Pada Gambar 3.1 menjelaskan proses *login* yang dimana jika *User* berhasil login maka *User* akan dibawa ke Dashboard. Proses *login* dimulai dengan *User* melakukan *input Email* dan *Password*. Selanjutnya dilakukan pemeriksaan apakah

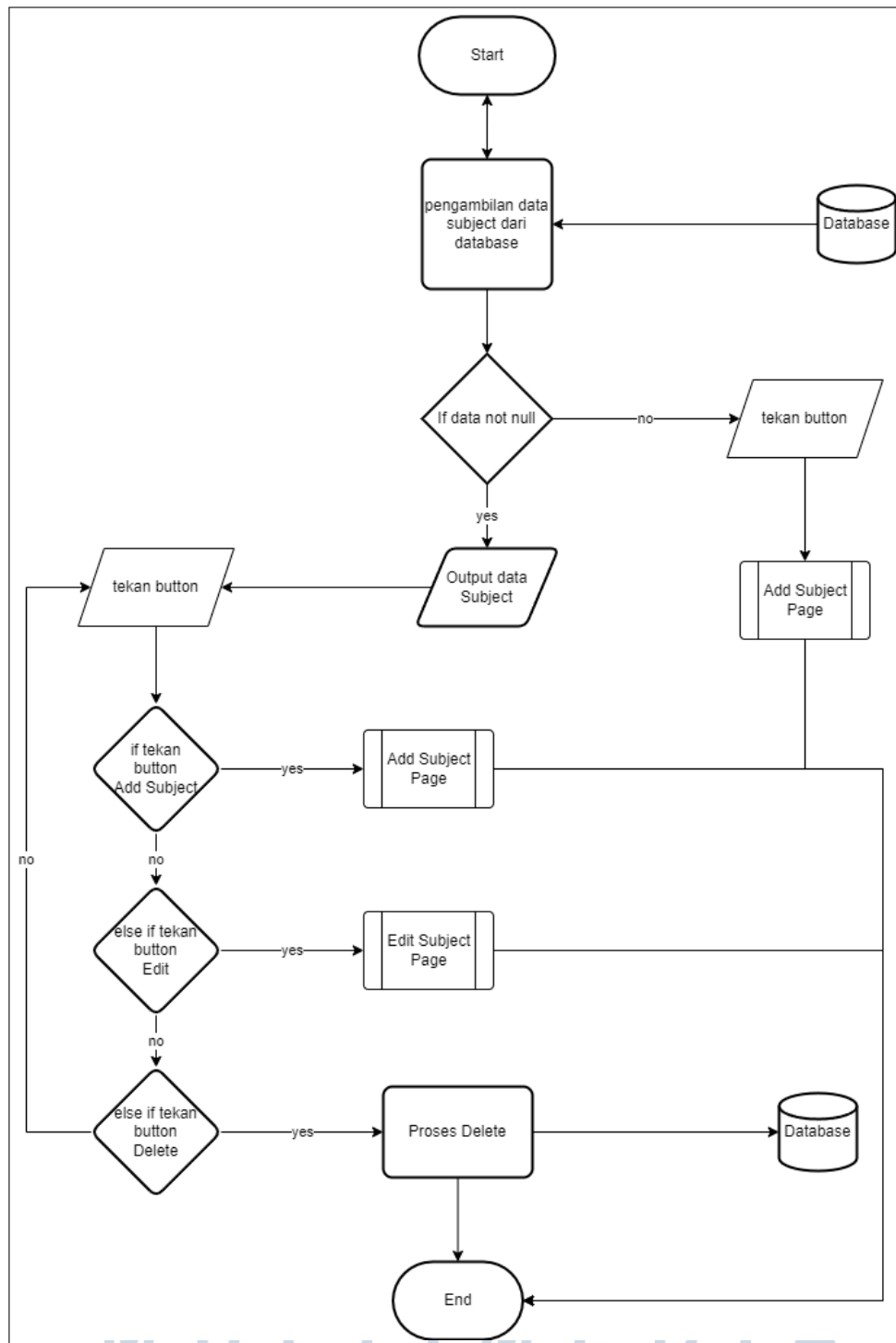
User melakukan *input* kosong, jika *input* *User* kosong maka *User* diwajibkan untuk mengisi setiap *field* yang ada. Selanjutnya melakukan proses pengecekan apakah *input* yang dimasukkan oleh *User* ada dalam *database* atau tidak, jika *input* tidak ada dalam *database* maka *User* wajib untuk mengisi kembali dengan *input* yang sesuai dengan data dalam *database*. Jika *User* berhasil melakukan *login* maka akan dilakukan penambahan *session* dalam tabel *session* dan *User* diarahkan ke *Dashboard*.



Gambar 3.2. Flowchart Dashboard

Dalam Gambar 3.2 dilakukan proses pengambilan nama *User* dari *database* sehingga bisa mengeluarkan *output* nama *User*. Kemudian proses dilanjutkan pada flowchart *Reference A*.

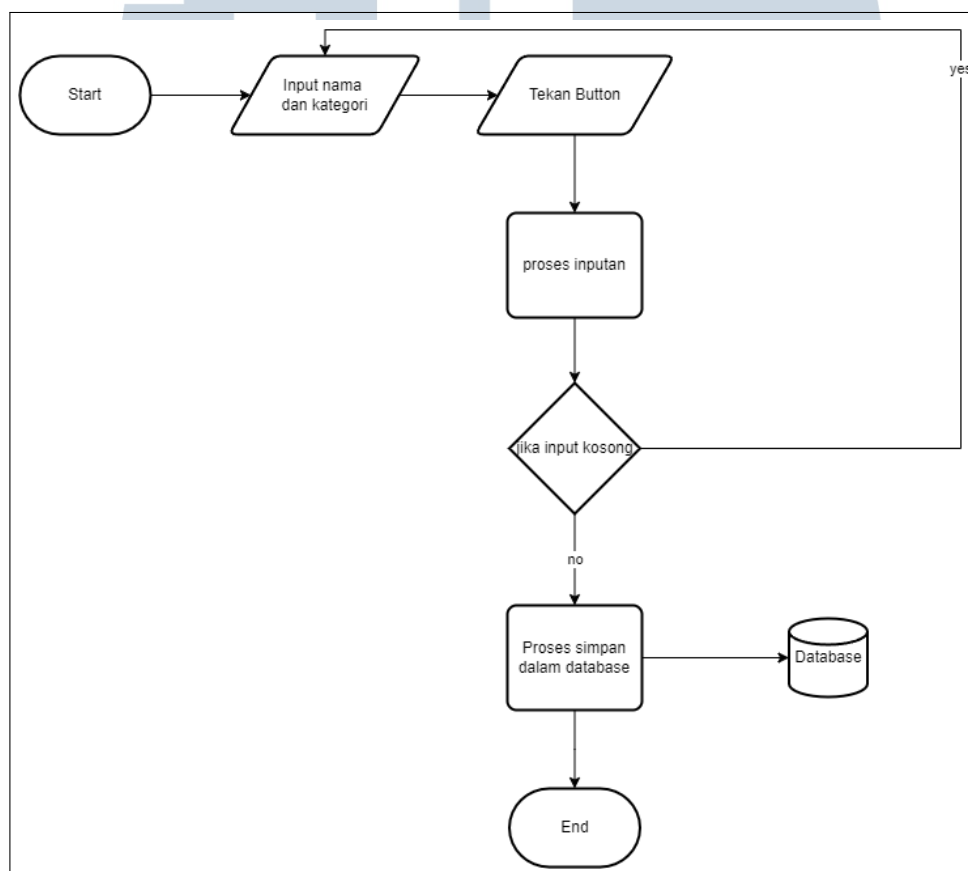
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3. Flowchart Subject Page

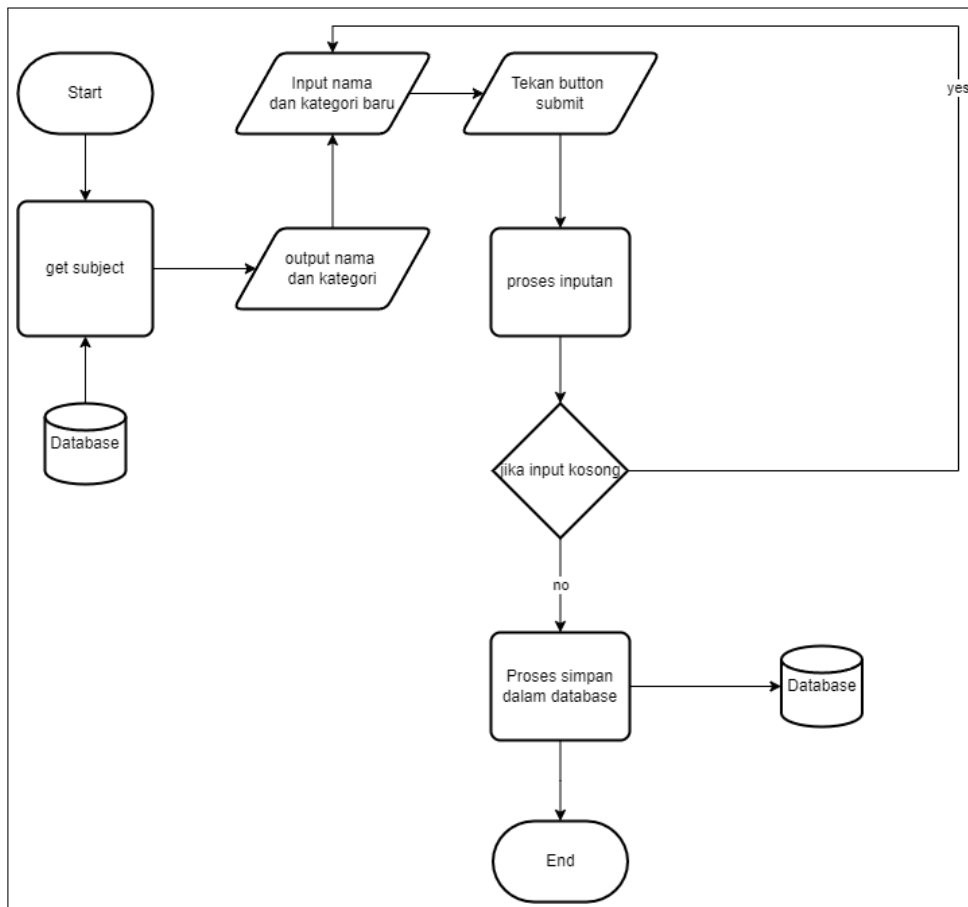
Pada Gambar 3.3 proses pertama yang dilakukan adalah mengambil data *subject* dalam *database*. Jika belum memiliki data *subject*, maka *User* bisa memilih untuk menekan tombol *Add Subject* yang akan diarahkan ke *flowchart Add Subject Page*. Jika tidak memilih untuk menekan tombol *Add Subject* proses

dilanjutkan pada flowchart *Reference A*. Jika sudah memiliki data *Subject* maka akan mengeluarkan *list data Subject*. Selanjutnya *User* bisa memilih untuk menekan beberapa tombol. Jika *User* menekan tombol *Add Subject* maka *User* akan diarahkan ke *flowchart Add Subject Page*. Jika *User* menekan tombol *Edit* maka akan diarahkan ke *flowchart Edit Subject Page*. Jika *User* memilih untuk menekan tombol *delete* maka akan melakukan proses *delete* data kemudian diarahkan ke *flowchart Subject Page*. jika tidak menekan tombol maka proses akan dilanjutkan pada *flowchart Reference A*.



Gambar 3.4. *Flowchart Add Subject Page*

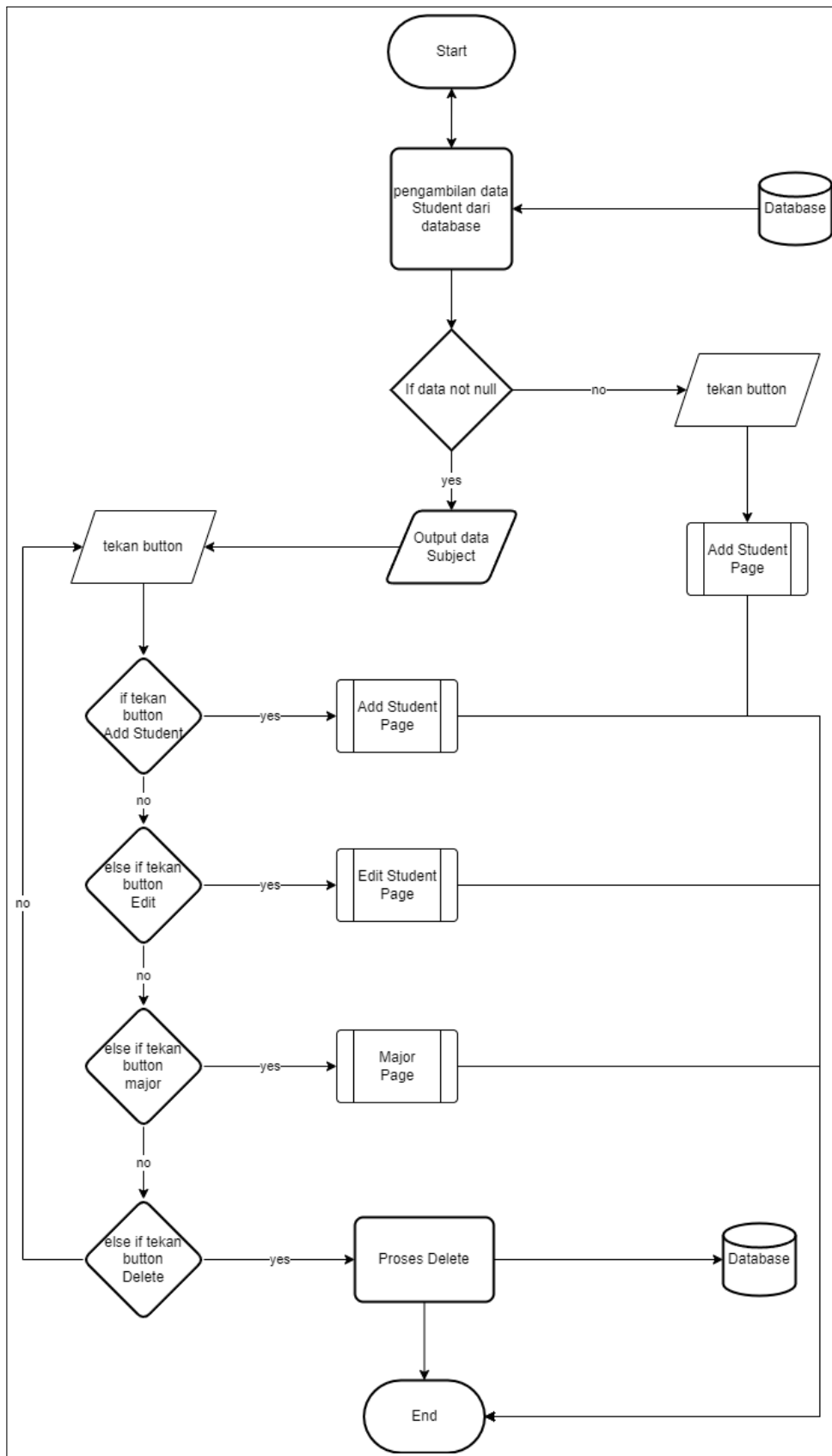
Pada Gambar 3.4 proses pertama adalah melakukan *input* nama dan category kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka *User* diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart Subject Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.



Gambar 3.5. Flowchart Subject Edit Page

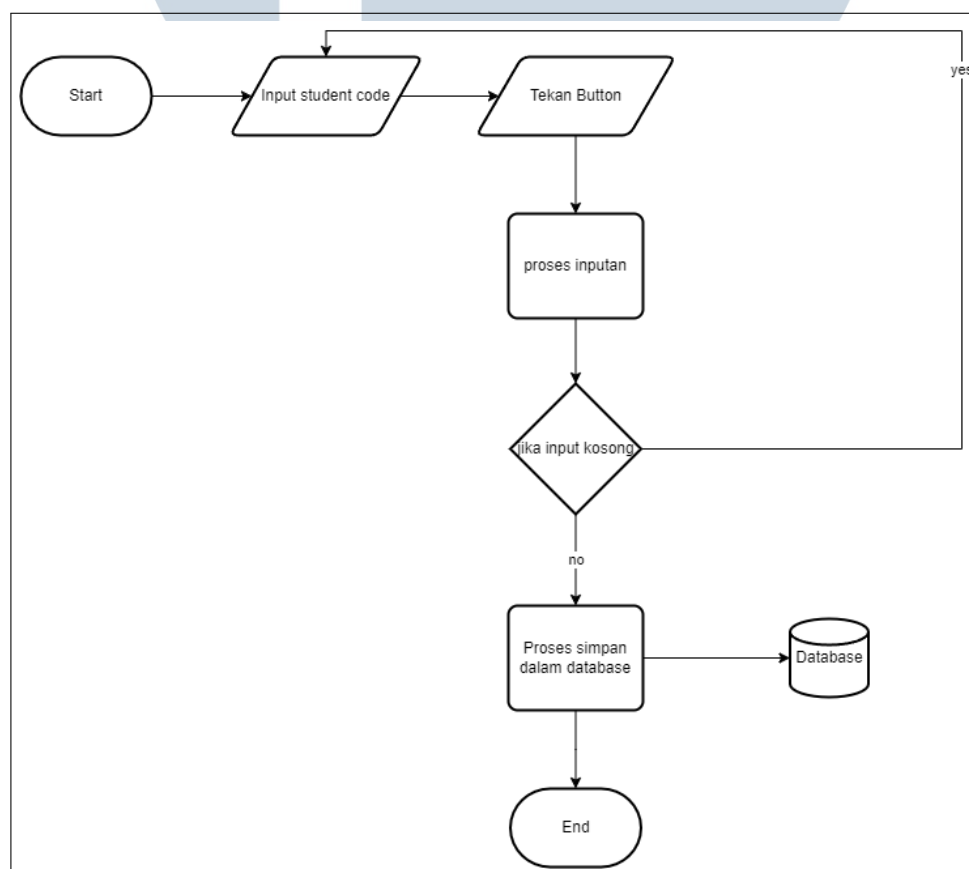
Pada Gambar 3.5 proses pertama adalah melakukan pengambilan data *subject* dari *database* selanjutnya *input* nama dan *category* baru kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka User diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart Subject Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Flowchart Student Page

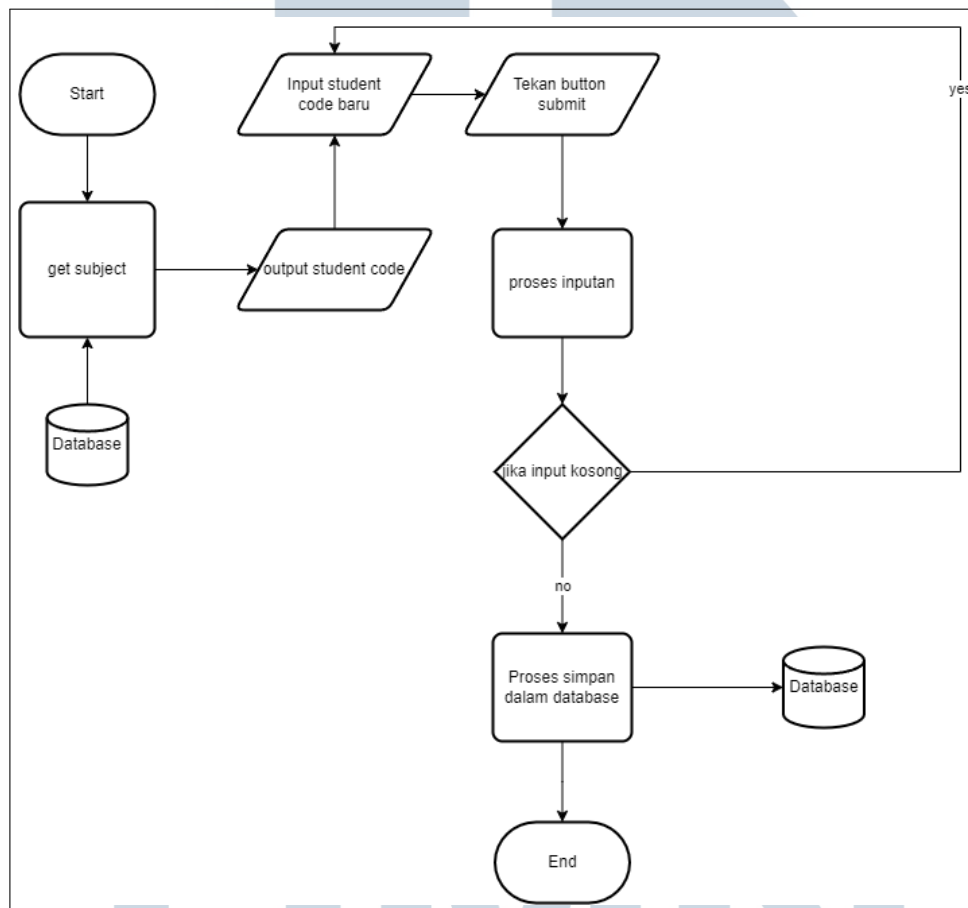
Pada Gambar 3.6 proses pertama yang dilakukan adalah mengambil data *student* dalam *database*. Jika belum memiliki data *student*, maka *User* bisa memilih untuk menekan tombol *Add Student* yang akan diarahkan ke *flowchart Add Student Page*. Jika tidak memilih untuk menekan tombol *Add Student* proses dilanjutkan pada *flowchart Reference A*. Jika sudah memiliki data *Subject* maka akan mengeluarkan *list data Subject*. Selanjutnya *User* bisa memilih untuk menekan beberapa tombol. Jika *User* menekan tombol *Add Student* maka *User* akan diarahkan ke *flowchart Add Student Page*. Jika *User* menekan tombol *Edit* maka akan diarahkan ke *flowchart Edit Student Page*. Jika *User* menekan tombol *majors* maka akan diarahkan ke *flowchart Major Page*. Jika *User* memilih untuk menekan tombol *delete* maka akan melakukan proses *delete* data kemudian diarahkan ke *flowchart Student Page*. jika tidak menekan tombol maka proses akan dilanjutkan pada *flowchart Reference A*.



Gambar 3.7. *Flowchart Add Student Page*

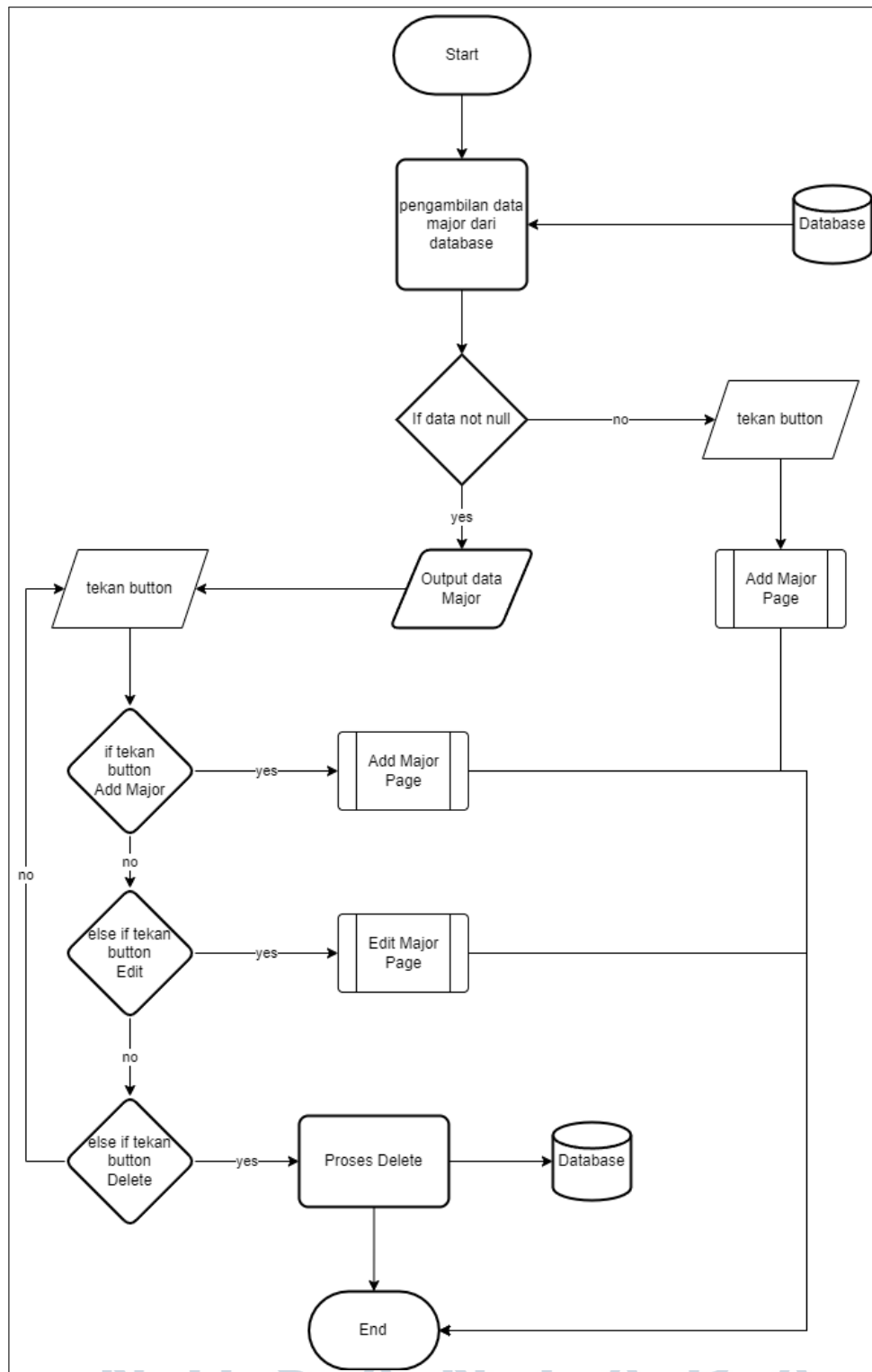
Pada Gambar 3.7 proses pertama adalah melakukan *input student code* kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka *User*

diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart Student Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.



Gambar 3.8. *Flowchart Student Edit Page*

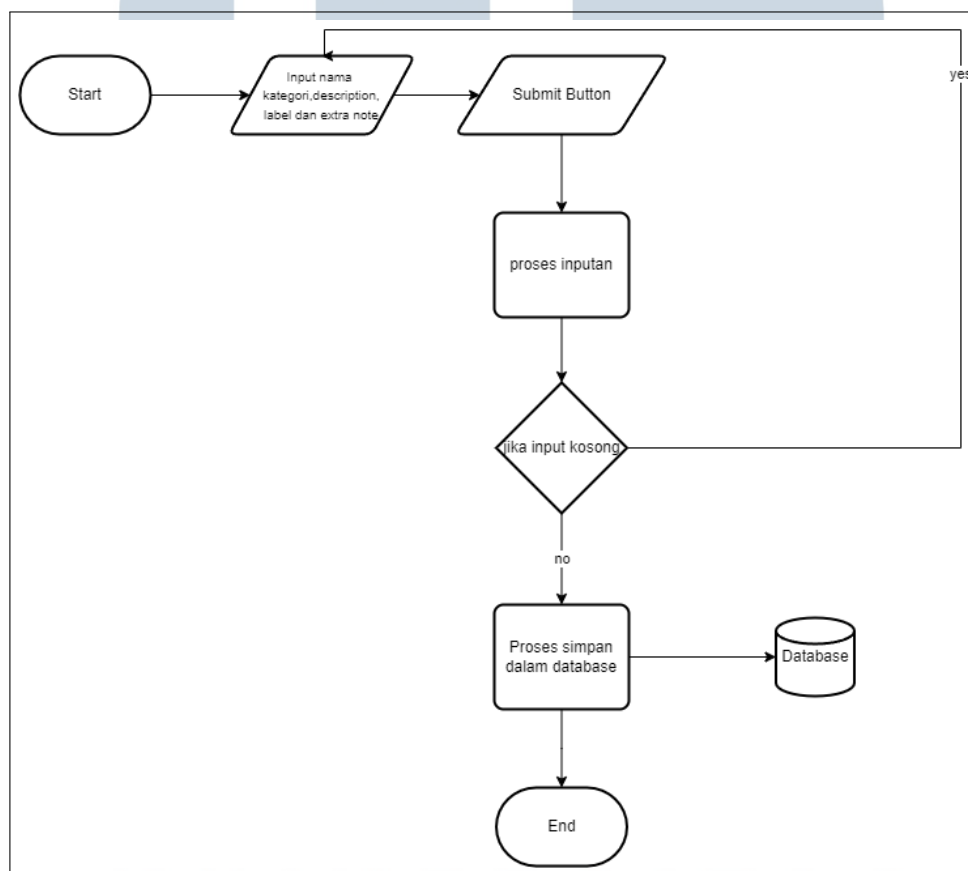
Pada Gambar 3.8 proses pertama adalah melakukan pengambilan data *student* dari *database* selanjutnya *input student code* baru kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka User diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart Student Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.



Gambar 3.9. Flowchart Major Page

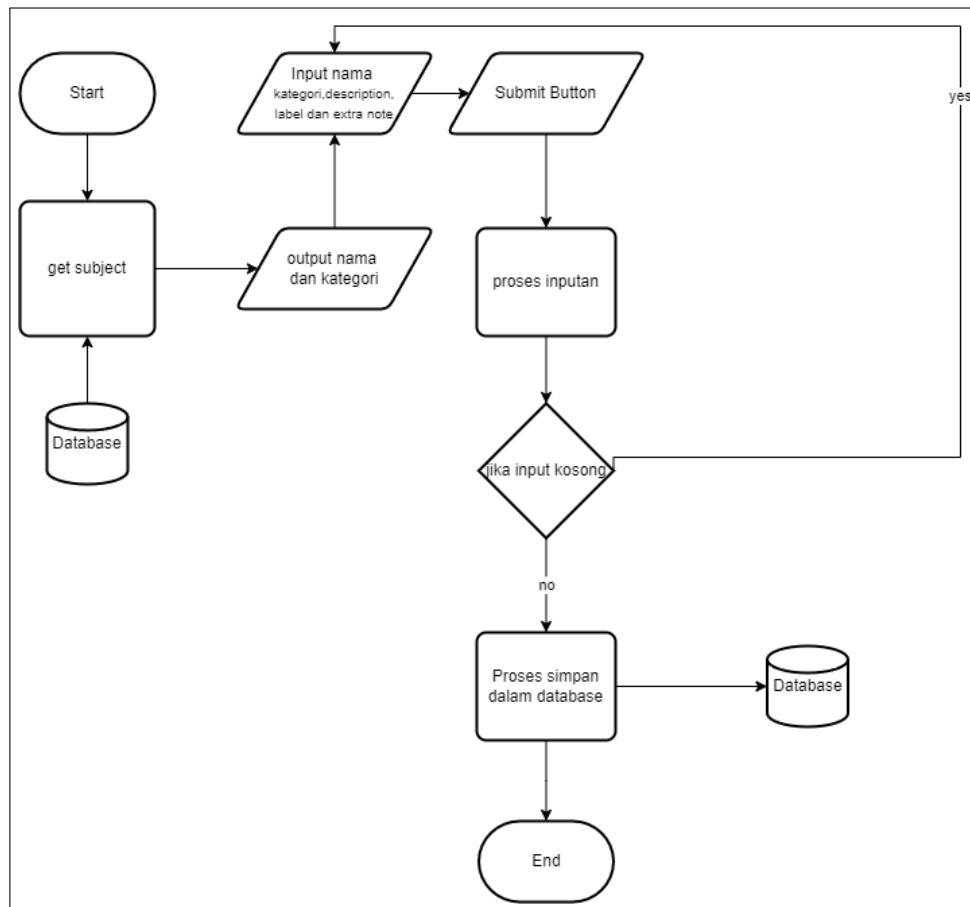
Pada Gambar 3.9 proses pertama yang dilakukan adalah mengambil data *Major* dalam *database*. Jika belum memiliki data *Major*, maka *User* bisa memilih

untuk menekan tombol *Add Major* yang akan diarahkan ke *flowchart Add Major Page*. Jika tidak memilih untuk menekan tombol *Add Major* proses dilanjutkan pada *flowchart Reference A*. Jika sudah memiliki data *Major* maka akan mengeluarkan *list data Major*. Selanjutnya *User* bisa memilih untuk menekan beberapa tombol. Jika *User* menekan tombol *Add Major* maka *User* akan diarahkan ke *flowchart Add Major Page*. Jika *User* menekan tombol *Edit* maka akan diarahkan ke *flowchart Edit Major Page*. Jika *User* memilih untuk menekan tombol *delete* maka akan melakukan proses *delete* data kemudian diarahkan ke *flowchart Major Page*. jika tidak menekan tombol maka proses akan dilanjutkan pada *flowchart Reference A*.



Gambar 3.10. *Flowchart Add Major Page*

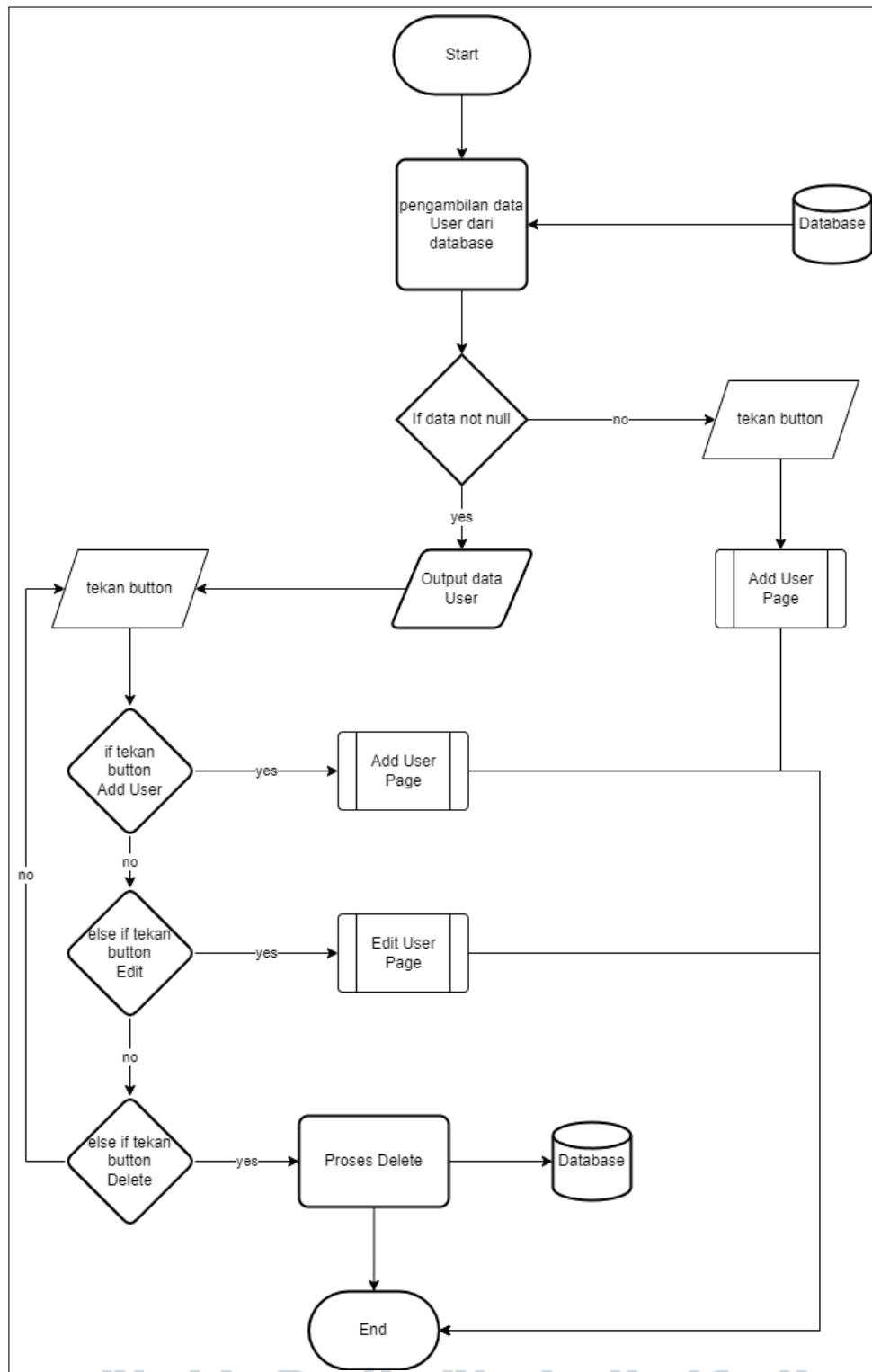
Pada Gambar 3.10 proses pertama adalah melakukan *input* nama dan *category* kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka *User* diwajibkan untuk memasukkan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart Major Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.



Gambar 3.11. Flowchart Edit Major Page

Pada Gambar 3.11 proses pertama adalah melakukan pengambilan data *major* dari *database* selanjutnya *input* nama dan category baru kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka User diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart Major Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.

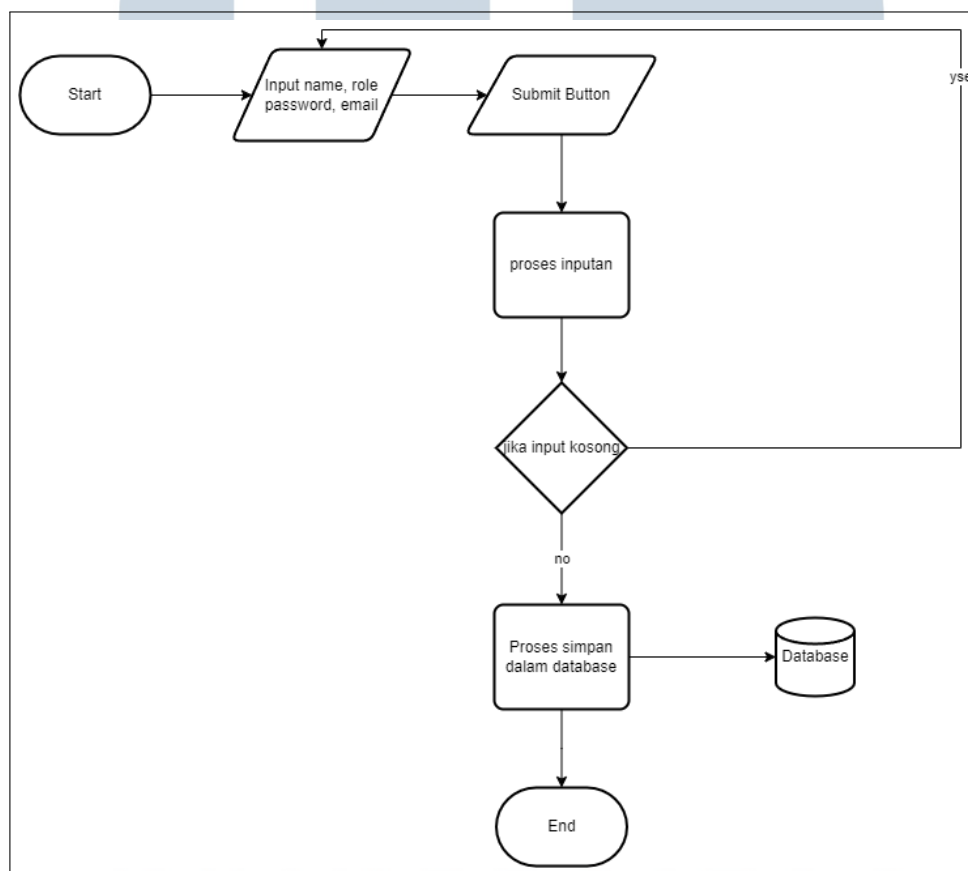
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.12. Flowchart User Page dan Delete User

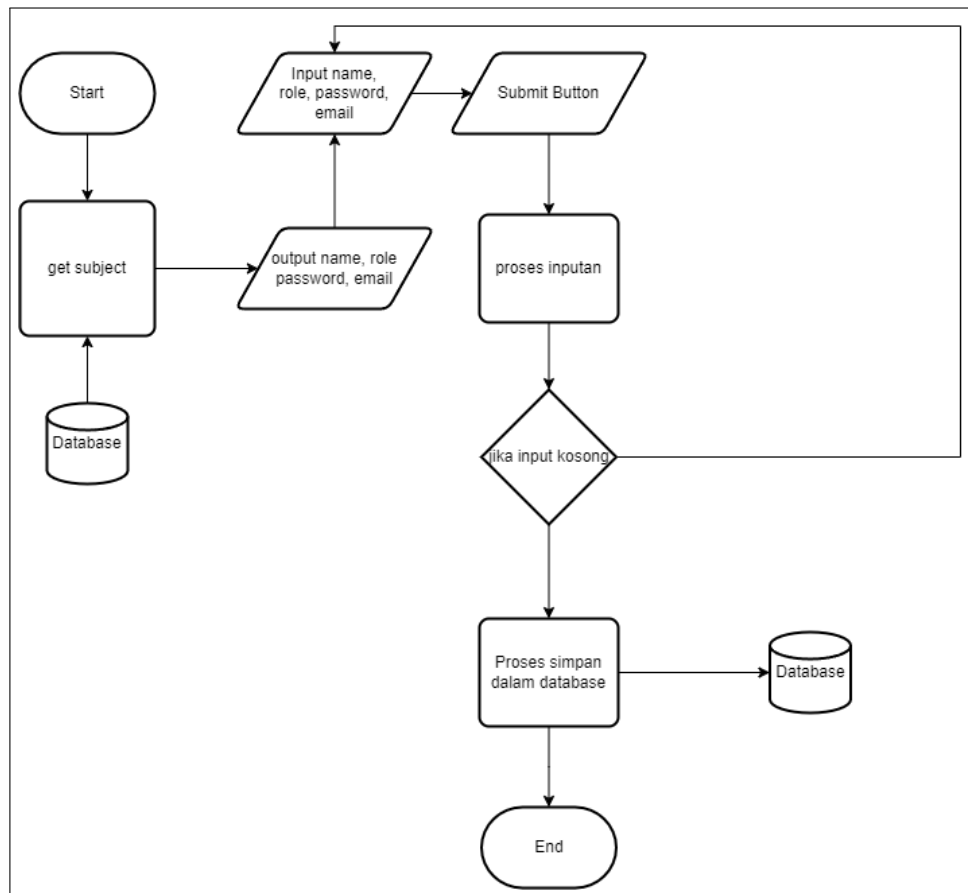
Pada Gambar 3.12 proses pertama yang dilakukan adalah mengambil data *User* dalam *database*. Jika belum memiliki data *User*, maka *User* bisa memilih

untuk menekan tombol *Add User* yang akan diarahkan ke *flowchart Add User Page*. Jika tidak memilih untuk menekan tombol *Add User* proses dilanjutkan pada *flowchart Reference A*. Jika sudah memiliki data *User* maka akan mengeluarkan *list data User*. Selanjutnya *User* bisa memilih untuk menekan beberapa tombol. Jika *User* menekan tombol *Add User* maka *User* akan diarahkan ke *flowchart Add User Page*. Jika *User* menekan tombol *Edit* maka akan diarahkan ke *flowchart Edit User Page*. Jika *User* memilih untuk menekan tombol *delete* maka akan melakukan proses *delete* data kemudian diarahkan ke *flowchart User Page*. jika tidak menekan tombol maka proses akan dilanjutkan pada *flowchart Reference A*.



Gambar 3.13. *Flowchart Add User Page*

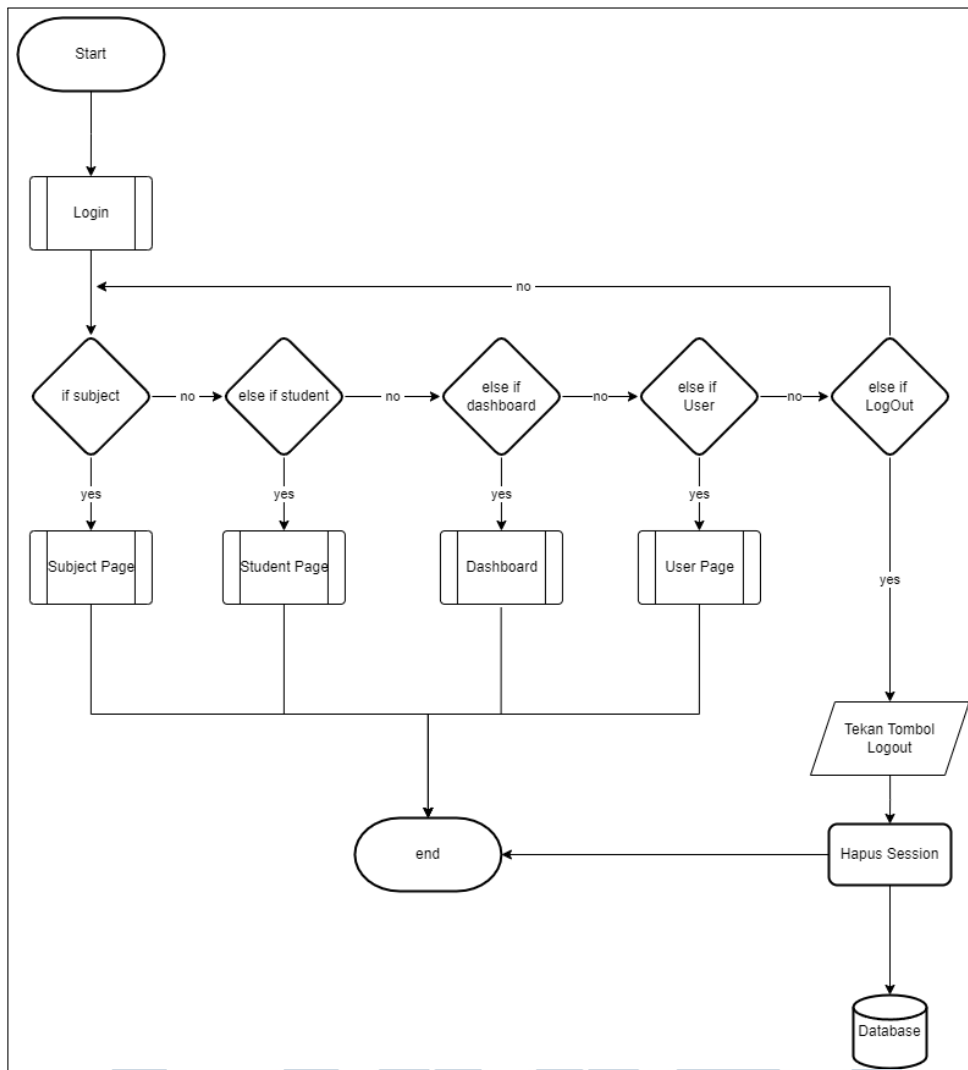
Pada Gambar 3.13 proses pertama adalah melakukan *input* nama, email, password, dan role kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka *User* diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart User Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.



Gambar 3.14. Flowchart Edit User Page dan Edit User

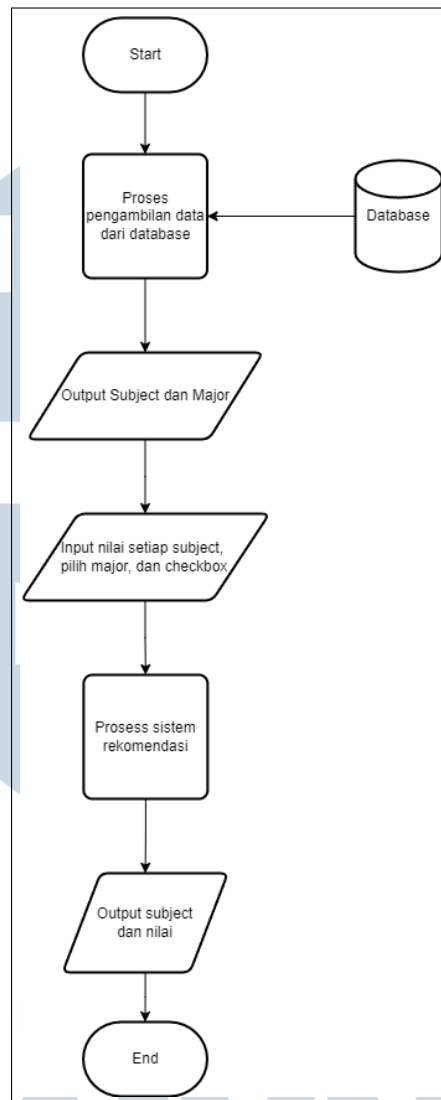
Pada Gambar 3.11 proses pertama adalah melakukan pengambilan data *user* dari *database* selanjutnya *input* nama, email, password, dan role baru kemudian memilih untuk menekan tombol. Jika memilih untuk menekan tombol *submit* maka akan memproses *input User*, jika *input User* kosong maka User diwajibkan untuk memasukan *input* pada setiap *field* yang tersedia. Jika *input User* tidak kosong maka akan melakukan proses simpan data *input User* ke dalam *database* kemudian diarahkan ke *flowchart User Page*. Jika tidak memilih untuk menekan tombol *submit* maka proses diarahkan ke *flowchart Reference A*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



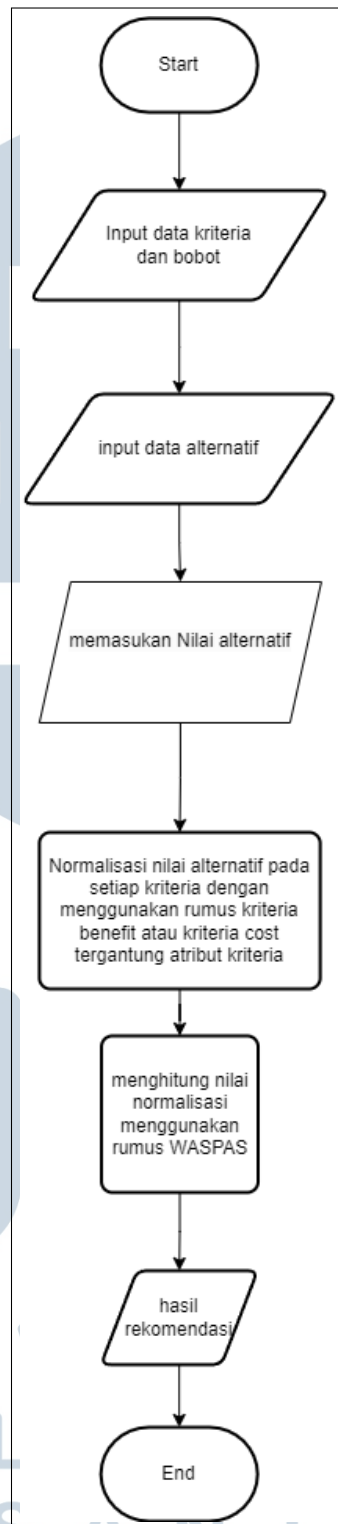
Gambar 3.15. *Flowchart Admin*

Pada Gambar 3.15 merupakan flowchart utama untuk admin. Untuk pertama *flowchart* dimulai dengan *login* ketika sudah login akan di arakan ke dashboard. Selanjutnya *user* bisa pilih menu, Jika menekan tombol *Subject* maka akan di arahkan ke *flowchart Subject Page*. Jika menekan tombol *Major* maka akan di arahkan ke *flowchart Major Page*. Jika menekan tombol *Dashboard* maka akan di arahkan ke *flowchart Dashboard*. jika menekan tombol *User* maka akan di arahkan ke *flowchart User Page*. Dan pilihan terakhir adalah menekan tombol *Logout* yang kemudian melakukan proses hapus *session* yang membuat User keluar dari *website*



Gambar 3.16. *Flowchart* Sistem Rekomendasi

Pada Gambar 3.16 proses pertama yang dilakukan adalah mengambil data *subject* dan *major* pada *database* yang dimana akan menghasilkan data *subject* dan *major* data-data tersebut akan digunakan oleh *user*. setelah itu *User* akan melakukan *input* seperti memilih *Major*, mengisi *value* dari setiap *subject* dan melakukan *checkbox*. setelah itu *user* menekan tombol *submit* sehingga melakukan proses sistem rekomendasi dan akan keluar urutan *subject* berdasarkan nilai yang di dapat.

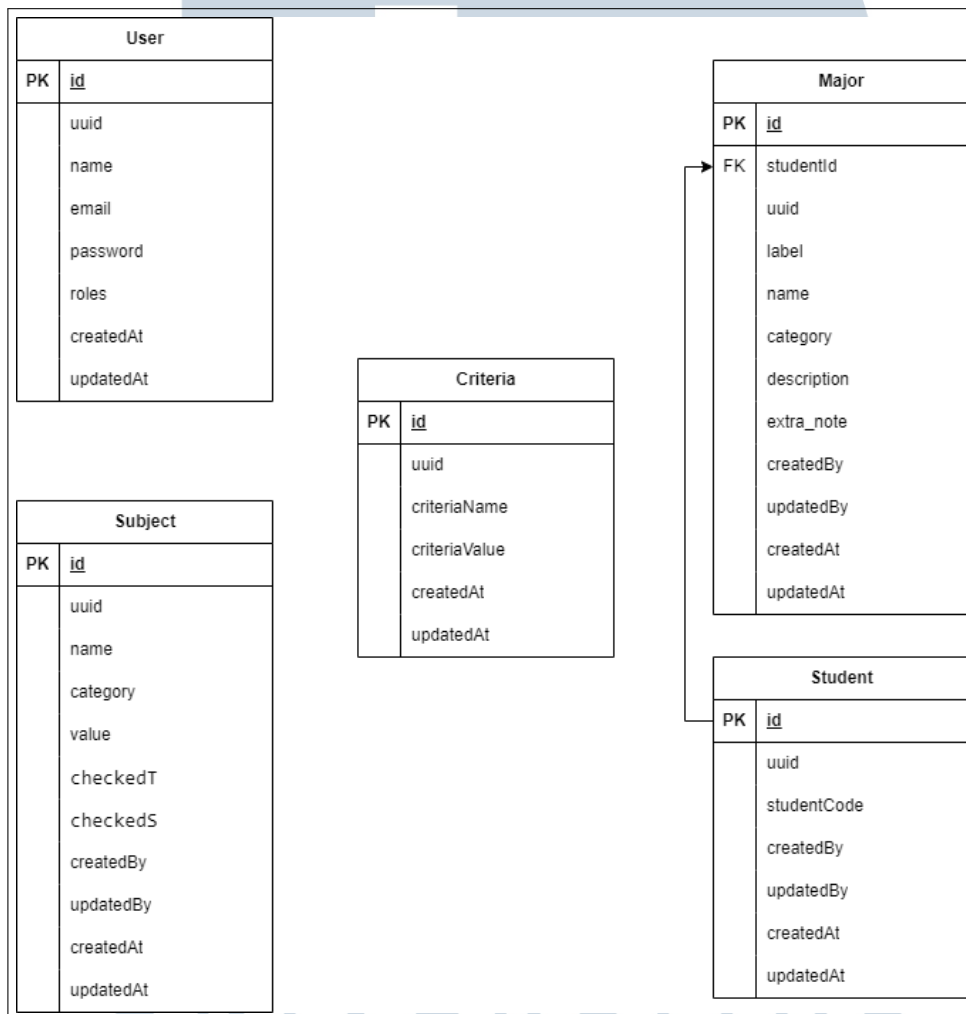


Gambar 3.17. Flowchart WASPAS

Pada Gambar 3.17 yang dilakukan pertama adalah *input* data kriteria dan bobot kemudian *input* data alternatif. Selanjutnya memasukan nilai alternatif

setelah memasukan nilai alternatif melakukan proses normalisasi nilai alternatif pada setiap kriteria dengan menggunakan rumus kriteria *benefit* atau kriteria *cost* tergantung atribut kriteria. Terakhir melakukan proses menghitung nilai normalisasi menggunakan rumus WASPAS dan akan menghasilkan nilai hasil rekomendasi.

3.3 Skema Database



Gambar 3.18. Database Schema

Pada Gambar 3.18 merupakan gambar schema *database* dalam sistem yang memiliki dua relasi yaitu, relasi *One to Many* antara tabel *Student* dan tabel *Major*.

3.4 Tabel Data

Tabel 3.1. Tabel User

Nama	Tipe Data	Constraint	Keterangan
id	int(11)	PRIMARY KEY	id user
uuid	Varchar(256)		uuid user
name	varchar(100)		name user
email	varchar(100)		email user
password	varchar(11)		password user
roles	varchar(11)		role user
createdAt	smalldatetime		data created at
updatedAt	smalldatetime		data update at

Pada Tabel 3.1 merupakan tabel yang akan menyimpan data pribadi user yang berelasi dengan tabel 3.2 dan tabel 3.4. Terdapat beberapa kolom dan jenis data sebagai berikut:

1. (id) tipe data *integer(11)*, data bersifat sebagai *primary key*, data akan menampung *id user*.
2. (uuid) tipe data *varchar(256)*, data akan menampung *uuid user*.
3. (name) tipe data *varchar(100)*, data akan menampung nama admin.
4. (email) tipe data *varchar(100)*, data akan menampung *email* dari admin.
5. (password) tipe data *varchar(11)*, data ini menampung *password* dari admin.
6. (roles) tipe data *varchar(20)*, data ini menampung *username* dari admin.
7. (createdAt) tipe data *smalldatetime*, data akan terisi ketika *User* menambahkan *user*.
8. (updatedAt) tipe data *smalldatetime*, data akan terisi ketika *User* merubah data *user*.

Tabel 3.2. Tabel Subject

Nama	Tipe Data	Constraint	Keterangan
id	int(11)	PRIMARY KEY	id user
uuid	Varchar(256)		uuid subject
name	varchar(100)		name subject
category	varchar(11)		category subject
value	int(11)		value subject
checkedT	float(11)		checkedT subject
checkedS	float(11)		checkedS subject
createdBy	Varchar(255)		data created By
updatedBy	Varchar(255)		data updated By
createdAt	smalldatetime		data created at
updatedAt	smalldatetime		data update at

Pada Tabel 3.2 terdapat beberapa kolom dan jenis data sebagai berikut:

1. (id) tipe data *integer(11)*, data bersifat sebagai *primary key*, data akan menampung *id subject*.
2. (uuid) tipe data *varchar(256)*, data akan menampung *uuid subject*.
3. (name) tipe data *varchar(100)*, data akan menampung nama *subject*.
4. (category) tipe data *varchar(11)*, data akan menampung *category* dari *subject*.
5. (value) tipe data *int(11)*, data ini menampung *value* dari *subject*.
6. (checkedT) tipe data *float(1)*, data ini menampung *checkedT* dari *subject*.
7. (checkedS) tipe data *float(1)*, data ini menampung *checkedS* dari *subject*.
8. (createdBy) tipe data *smalldatetime*, data menampung nama *User* menambahkan *subject*.
9. (updatedBy) tipe data *smalldatetime*, data menampung nama *User* merubah data *subject*.
10. (createdAt) tipe data *smalldatetime*, data akan terisi ketika *User* menambahkan *subject*.

- (updatedAt) tipe data *smalldatetime*, data akan terisi ketika *User* merubah data *subject*.

Tabel 3.3. Tabel Student

Nama	Tipe Data	Constraint	Keterangan
id	int(11)	PRIMARY KEY	id user
uuid	Varchar(255)		uuid subject
studentCode	Varchar(255)	UNIQUE	name subject
createdBy	Varchar(255)		data created By
updatedBy	Varchar(255)		data updated By
createdAt	smalldatetime		data created at
updatedAt	smalldatetime		data update at

Pada Tabel 3.3 terdapat beberapa kolom dan jenis data sebagai berikut:

- (id) tipe data *integer(11)*, data bersifat sebagai *primary key*, data akan menampung *id student*.
- (uuid) tipe data *varchar(256)*, data akan menampung *uuid student*.
- (studentCode) tipe data *varchar(100)*, data akan menampung *student code*.
- (createdBy) tipe data *varchar(255)*, data menampung nama *User* menambahkan *student*.
- (updatedBy) tipe data *varchar(255)*, data menampung nama *User* merubah data *student*.
- (createdAt) tipe data *smalldatetime*, data akan terisi ketika *User* menambahkan *student*.
- (updatedAt) tipe data *smalldatetime*, data akan terisi ketika *User* merubah data *student*.

Tabel 3.4. Caption

Nama	Tipe Data	Constraint	Keterangan
id	int(11)	PRIMARY KEY	id major
studentId	int	FOREIGN KEY	student id
uuid	Varchar(255)		uuid major
name	Varchar(255)		name major
label	Varchar(255)		label major
category	Varchar(255)		category major
description	text		description major
extra note	text		extra note major
createdBy	Varchar(255)		data created By
updatedBy	Varchar(255)		data updated By
updatedBy	Varchar(255)		data updated By
updatedAt	smalldatetime		data update at

Pada Tabel 3.4 terdapat beberapa kolom dan jenis data sebagai berikut:

1. (id) tipe data *integer(11)*, data bersifat sebagai *primary key*, data akan menampung *id major*.
2. (userId) tipe data *integer(11)*, data bersifat sebagai *foreign key*, data akan menampung *id user* yang menambah data *subject*.
3. (uuid) tipe data *varchar(255)*, data akan menampung *uuid major*.
4. (label) tipe data *varchar(255)*, data akan menampung nama *major*.
5. (name) tipe data *varchar(255)*, data akan menampung nama *major*.
6. (category) tipe data *varchar(255)*, data akan menampung *category* dari *major*.
7. (description) tipe data *text*, data akan menampung *description* dari *major*.
8. (extra note) tipe data *text*, data akan menampung *extra note* dari *major*.
9. (createdBy) tipe data *varchar(255)*, data menampung nama *User* menambahkan *major*.
10. (updatedBy) tipe data *varchar(255)*, data menampung nama *User* merubah data *major*.

11. (*createdAt*) tipe data *smalldatetime*, data akan terisi ketika *User* menambahkan *major*.
12. (*createdAt*) tipe data *smalldatetime*, data akan terisi ketika *User* merubah data *major*.

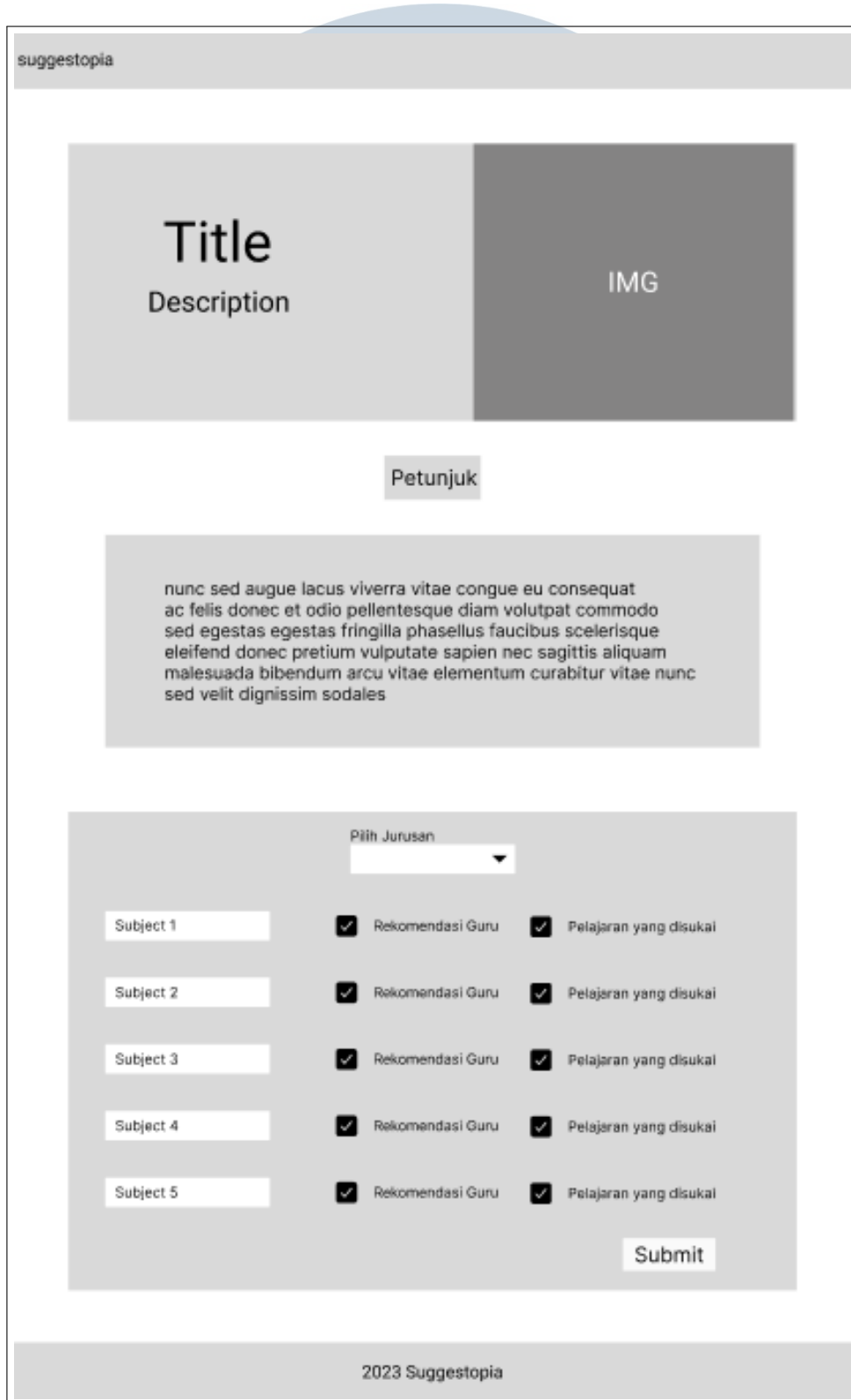
Tabel 3.5. Tabel Criteria

Nama	Tipe Data	Constraint	Keterangan
id	int(11)	PRIMARY KEY	id user
uuid	Varchar(255)		uuid kriteria
criteriaName	Varchar(255)		nama kriteria
criteriaValue	INT		nilai kriteria
createdAt	smalldatetime		data created at
updatedAt	smalldatetime		data update at

Pada Tabel ?? terdapat beberapa kolom dan jenis data sebagai berikut:

1. (*id*) tipe data *integer(11)*, data bersifat sebagai *primary key*, data akan menampung *id student*.
2. (*uuid*) tipe data *varchar(256)*, data akan menampung *uuid student*.
3. (*criteriaName*) tipe data *varchar(100)*, data akan menampung name kriteria.]
4. (*criteriaValue*) tipe data *varchar(100)*, data akan menampung nilai kriteria.]
5. (*createdAt*) tipe data *smalldatetime*, data akan terisi ketika pembuatan kriteria.
6. (*updatedAt*) tipe data *smalldatetime*, data akan terisi ketika ketika memperbaharui kriteria.

3.5 Wireframe Diagram



Gambar 3.19. Wireframe Home Page

Pada Gambar 3.19 merupakan gambar *wireframe* dari halaman utama website rekomendasi mata pelajaran. Dalam *wireframe* terdapat beberapa bagian mulai dari *header*, *body* dan *footer*. Pada bagian *header* terdapat *navbar* yang memiliki logo dan tulisan *website*, selanjutnya pada bagian *body* terdapat bagian *hero section* yang dimana menampilkan judul dan juga deskripsi singkat mengenai sistem serta sebuah *image*. Pada bagian *body* juga memiliki *collapse button* Petunjuk yang jika di tekan maka akan memunculkan bagian *card component* yang tertutup. Selanjutnya pada bagian *body* terdapat *form* rekomendasi yang dimana *form* tersebut akan diisi oleh User. Terakhir adalah bagian *footer*, pada bagian ini hanya menampilkan informasi singkat mengenai *website*



Gambar 3.20. *Wireframe Modal Result*

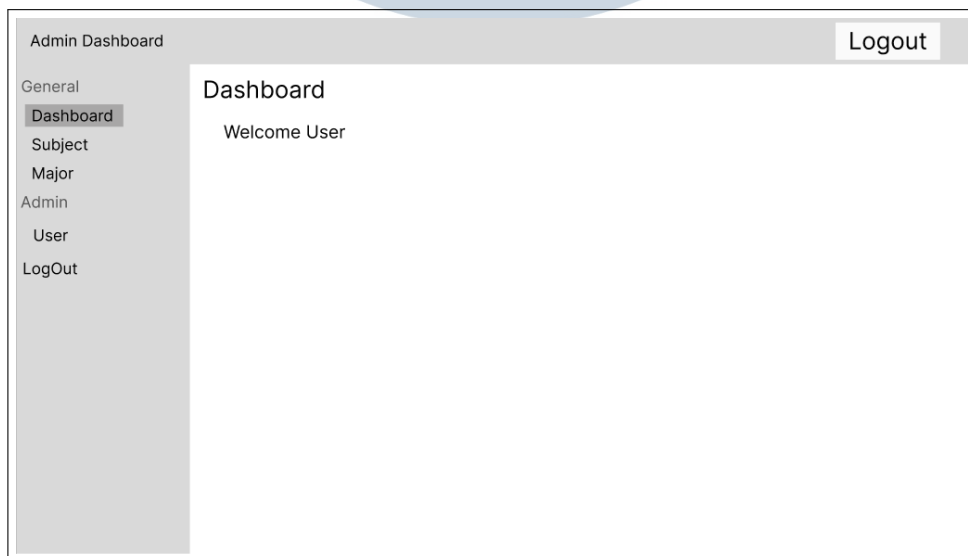
Pada Gambar 3.20 merupakan gambar *wireframe* yang menampilkan modal pada *website* yang berisi hasil rekomendasi. Hasilnya berupa urutan mata pelajaran

beserta nilai hasil rekomendasi.



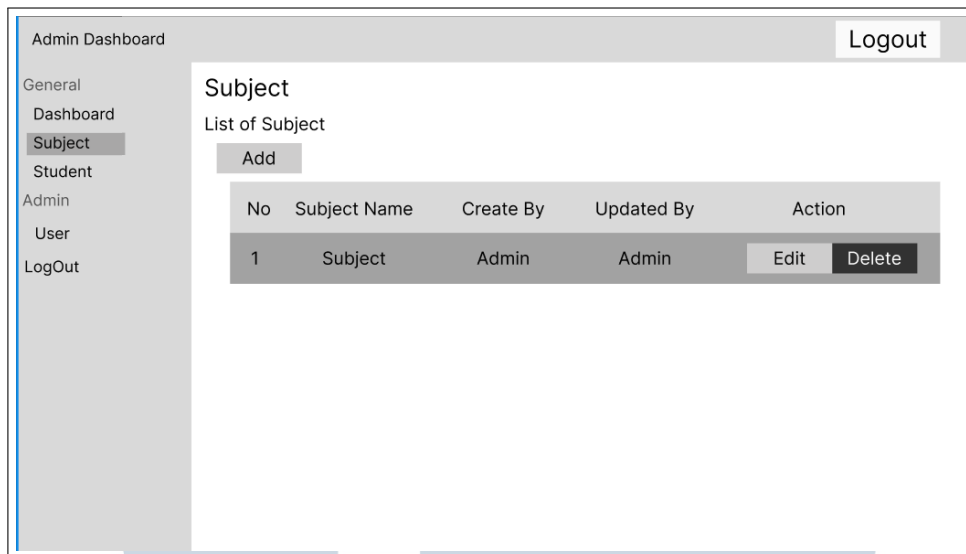
Gambar 3.21. Wireframe Login Page

Pada Gambar 3.21 merupakan gambar *wireframe* untuk tampilan login. Dalam terdapat *card form* yang berisi input *email* dan *password* serta tombol *login*



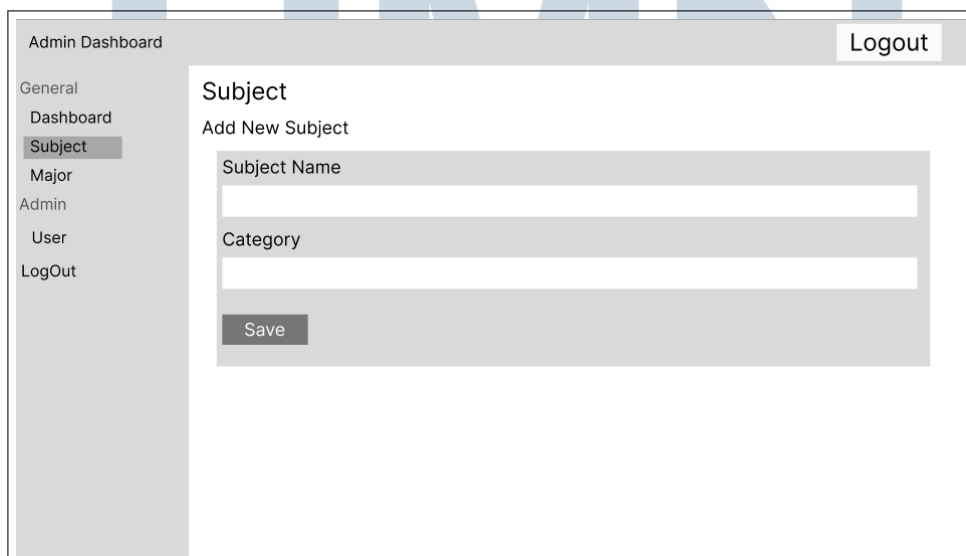
Gambar 3.22. Wireframe Dashboard Page

Pada Gambar 3.22 merupakan tampilan *wireframe* untuk halaman *Admin Dashboard*. Pada halaman ini terdapat sebuah tulisan *title* dan *welcome User*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.



Gambar 3.23. Wireframe Subject Page

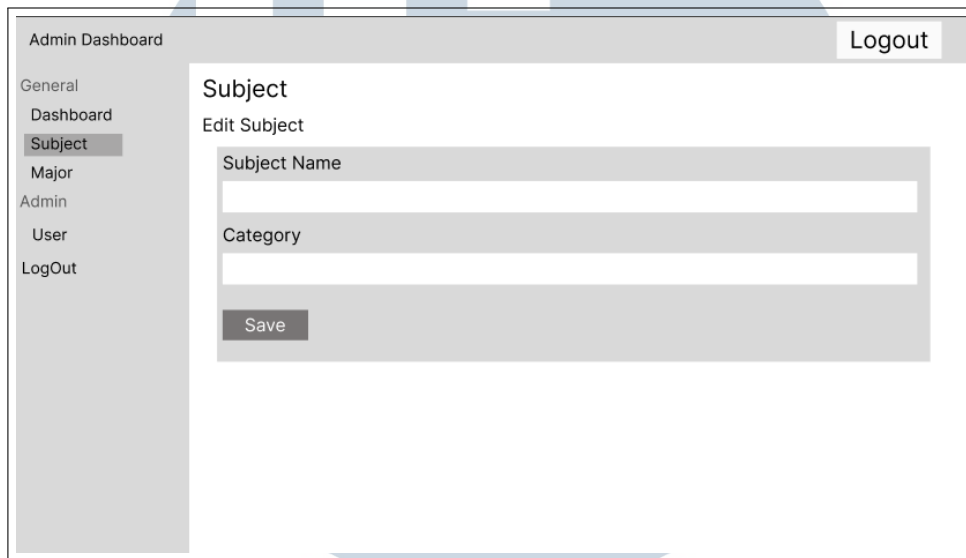
Pada Gambar 3.23 merupakan tampilan *wireframe* untuk halaman *Subject Page*. Halaman ini digunakan untuk menampilkan *list* data *subject* yang diambil dari *database*. Pada halaman ini bisa menekan tombol *Add* maka berpindah ke halaman *Add Subject Page*, jika menekan tombol *Edit* maka akan berpindah ke halaman *Edit Subject Page*, dan ketika menekan tombol *Delete* maka akan melakukan proses *delete* data. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.



Gambar 3.24. Wireframe Add Subject Page

Pada Gambar 3.24 merupakan tampilan *wireframe* untuk halaman *Add*

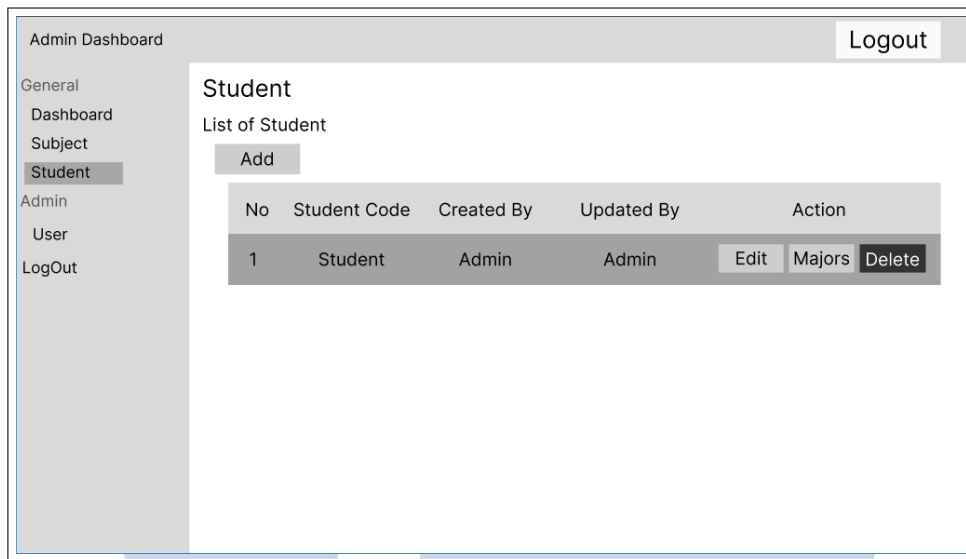
Subject. Pada halaman ini terdapat sebuah form untuk menyimpan data kedalam *database*. *Input* yang disediakan berupa *input text* dan *select*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.



Gambar 3.25. Wireframe Edit Subject Page

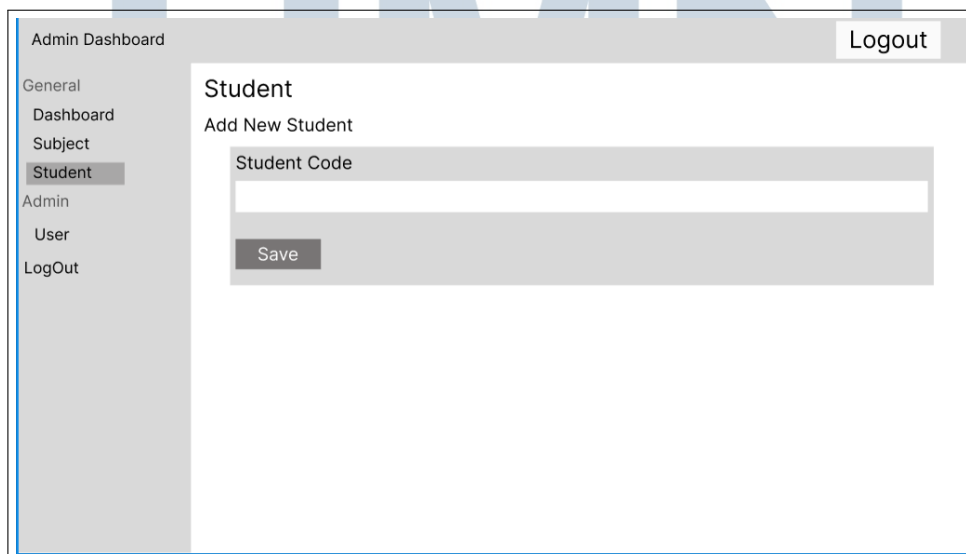
Pada Gambar 3.25 merupakan tampilan *wireframe* untuk halaman *Edit Subject*. Pada halaman ini terdapat sebuah form untuk mengedit data yang sudah ada dan menyimpan data yang telah diedit kedalam *database*. *Input* yang disediakan berupa *input text* dan *select*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.26. Wireframe Subject Page

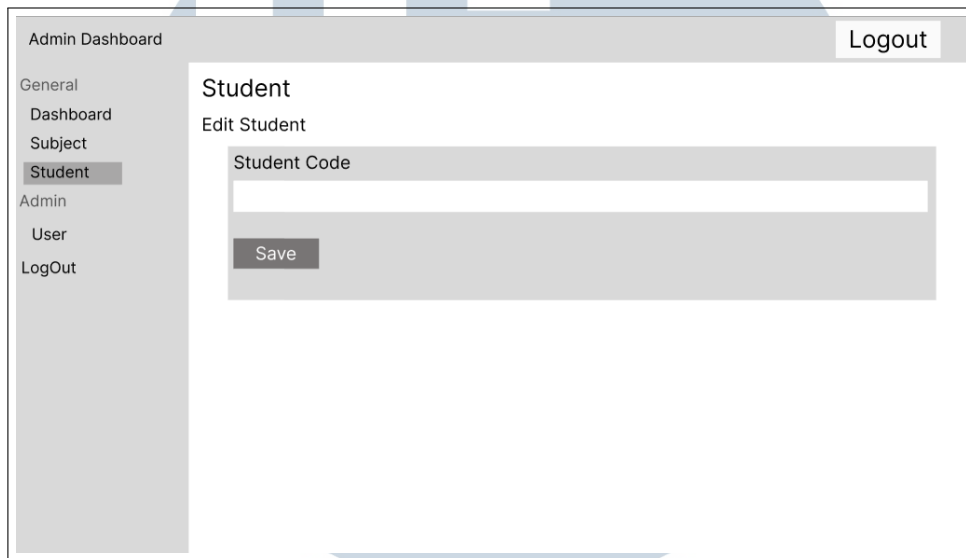
Pada Gambar 3.26 merupakan tampilan *wireframe* untuk halaman *Subject Page*. Halaman ini digunakan untuk menampilkan *list* data *student* yang diambil dari *database*. Pada halaman ini bisa menekan tombol *Add* maka berpindah ke halaman *Add Student Page*, jika menekan tombol *Edit* maka akan berpindah ke halaman *Edit Student Page*, dan ketika menekan tombol *Delete* maka akan melakukan proses *delete* data. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.



Gambar 3.27. Wireframe Add Student Page

Pada Gambar 3.27 merupakan tampilan *wireframe* untuk halaman *Add*

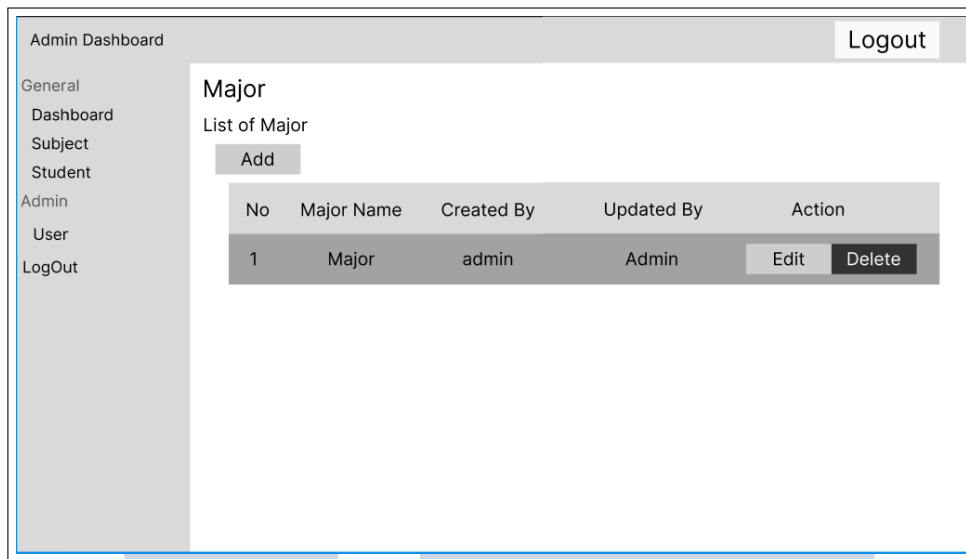
Student. Pada halaman ini terdapat sebuah form untuk menyimpan data kedalam *database*. *Input* yang disediakan berupa *input text*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.



Gambar 3.28. Wireframe Edit Student Page

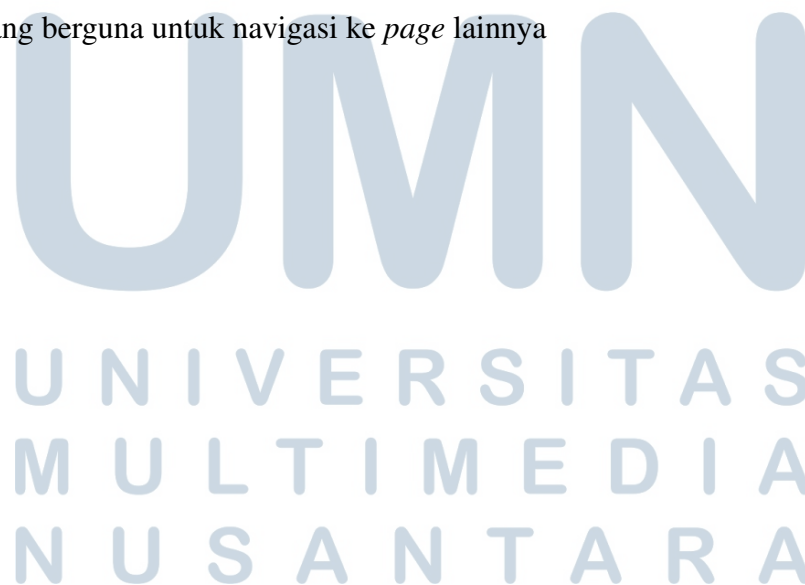
Pada Gambar 3.25 merupakan tampilan *wireframe* untuk halaman *Edit Subject*. Pada halaman ini terdapat sebuah form untuk mengedit data yang sudah ada dan menyimpan data yang telah diedit kedalam *database*. *Input* yang disediakan berupa *input text*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.

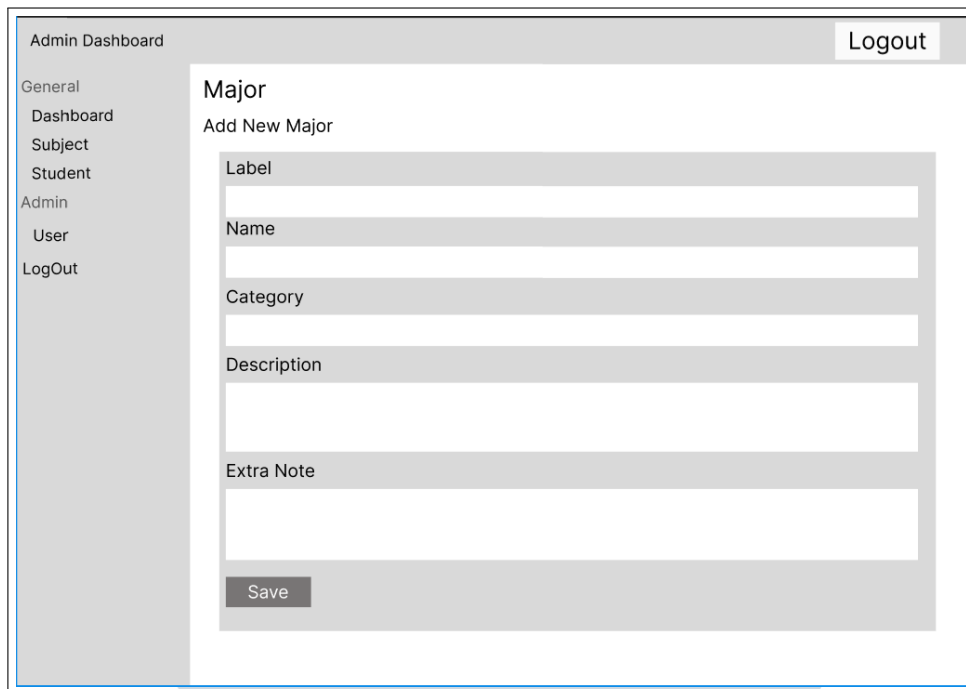
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.29. Wireframe Major Page

Pada Gambar 3.29 merupakan tampilan *wireframe* untuk halaman *Major Page*. Halaman ini digunakan untuk menampilkan *list* data *Major* yang diambil dari *database*. Pada halaman ini bisa menekan tombol *Add* maka berpindah ke halaman *Add Major Page*, jika menekan tombol *Edit* maka akan berpindah ke halaman *Edit Major Page*, dan ketika menekan tombol *Delete* maka akan melakukan proses *delete* data. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya

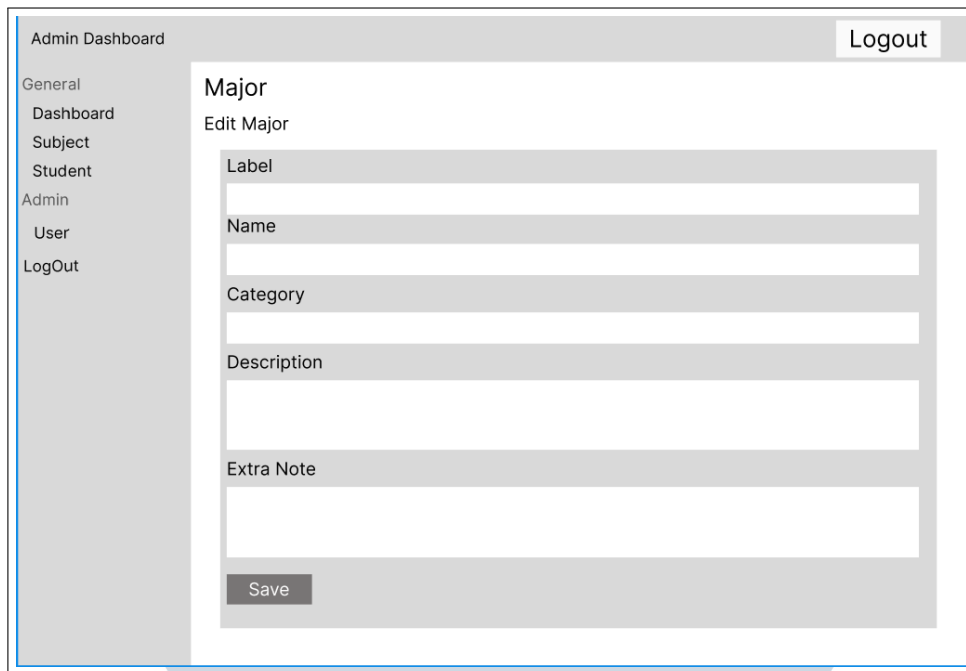




Gambar 3.30. Wireframe Add Major Page

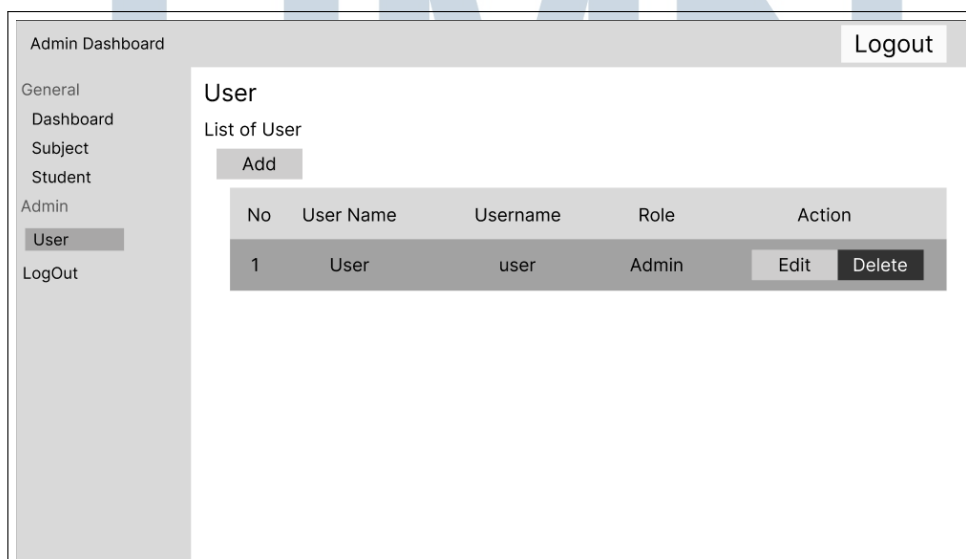
Pada Gambar 3.30 merupakan tampilan *wireframe* untuk halaman *Add Major*. Pada halaman ini terdapat sebuah form untuk menyimpan data kedalam *database*. *Input* yang disediakan berupa *input text* dan *select*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.31. Wireframe Edit Major Page

Pada Gambar 3.31 merupakan tampilan *wireframe* untuk halaman *Edit Major*. Pada halaman ini terdapat sebuah form untuk mengedit data yang sudah ada dan menyimpan data yang telah diedit kedalam *database*. *Input* yang disediakan berupa *input text* dan *select*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.



Gambar 3.32. Wireframe User Page

Pada Gambar 3.32 merupakan tampilan *wireframe* untuk halaman *User*

Page. Halaman ini digunakan untuk menampilkan *list* data *User* yang diambil dari *database*. Pada halaman ini bisa menekan tombol *Add* maka berpindah ke halaman *Add User Page*, jika menekan tombol *Edit* maka akan berpindah ke halaman *Edit User Page*, dan ketika menekan tombol *Delete* maka akan melakukan proses *delete* data. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya



Gambar 3.33. Wireframe Add User Page

Pada Gambar 3.33 merupakan tampilan *wireframe* untuk halaman *Add User*. Pada halaman ini terdapat sebuah form untuk menyimpan data kedalam *database*. *Input* yang disediakan berupa *input text* dan *select*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.34. *Wireframe Edit User Page*

Pada Gambar 3.34 merupakan tampilan *wireframe* untuk halaman *Edit User*. Pada halaman ini terdapat sebuah form untuk mengedit data yang sudah ada dan menyimpan data yang telah diedit kedalam *database*. *Input* yang disediakan berupa *input text* dan *select*. Pada bagian header terdapat logo *website* dan judul *website* serta *button Logout* untuk keluar dari *website Admin Dashboard*. Pada bagian *body* terdapat *sidebar* yang berguna untuk navigasi ke *page* lainnya.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA