



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 PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Penelitian akan dilakukan dalam beberapa tahap, yaitu telaah literatur, analisis kebutuhan, pemrograman sistem, *testing* dan *debug*, implementasi sistem, serta konsultasi dan penulisan laporan. Penjabaran setiap tahap penelitian adalah sebagai berikut.

1. Telaah literatur dilakukan dengan cara mengumpulkan dan mempelajari jurnal tentang metode-metode analisis sentimen, Multinomial Naïve Bayes Classification, dan Natural Language Processing.
2. Analisis kebutuhan dilakukan bersamaan dengan telaah literatur. Berdasarkan informasi yang didapat selama telaah literatur, kebutuhan sistem dirancang dan dianalisa. Perancangan sistem dimulai dengan membuat *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Kemudian diikuti dengan rancangan struktur tabel dan antarmuka untuk aplikasi *web*.
3. Dalam implementasi sistem kegiatan yang dilakukan diantaranya membuat *database* MySQL, mengimplementasikan algoritma perhitungan model ke dalam *coding*, *training dataset*, dan modifikasi model. *Dataset* yang digunakan dalam penelitian ini diambil dari kumpulan *user feedback* suatu aplikasi. *Dataset* tersebut disusun menjadi sebuah data tabular dan masing-masing diberi label untuk menandakan *feedback* positif atau *feedback* negatif. *Dataset* yang digunakan untuk *training* diberi label terlebih dahulu, “positif”

untuk *feedback* positif dan “negatif” untuk *feedback* negatif. Teknik pembelajaran mesin ini disebut *supervised learning* karena sudah terdapat *variable* yang ditargetkan, dengan kata lain mengelompokan suatu data ke data yang sudah ada.

4. *Testing* dan *debug* dilakukan untuk menguji apakah sistem berjalan dengan baik, serta memperbaiki *coding* apabila terdapat *bug*. Pada tahap ini juga dilakukan pengujian pada *dataset* yang telah disiapkan untuk *testing* dilakukan menggunakan model yang telah didapat dari hasil *training*. Dilakukan juga evaluasi performa sistem dengan menghitung *accuracy*, *precision*, dan *recall* dari hasil klasifikasi teks. Kemudian nilai *precision* dan *recall* digunakan untuk menghitung nilai *F1 score*. Model juga dimodifikasi untuk meningkatkan performa sistem.
5. Dibuat laporan sebagai dokumentasi dari penelitian dan pembuatan aplikasi. Laporan dibuat secara bertahap mulai dari pendahuluan hingga kesimpulan dan saran.

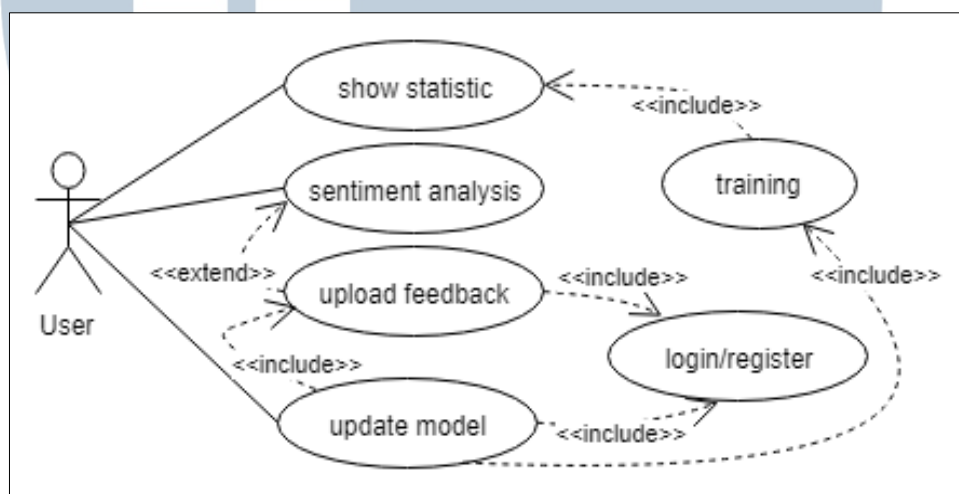
3.2 Perancangan Sistem

Perancangan sistem terdiri dari *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, perancangan struktur tabel, dan antarmuka aplikasi *web*.

3.2.1 Use Case Diagram

Gambar 3.1 merupakan *use case diagram* dari sistem yang dibuat. Terdapat satu aktor pada sistem ini yaitu User. Selain itu terdapat enam *use case* yaitu *show*

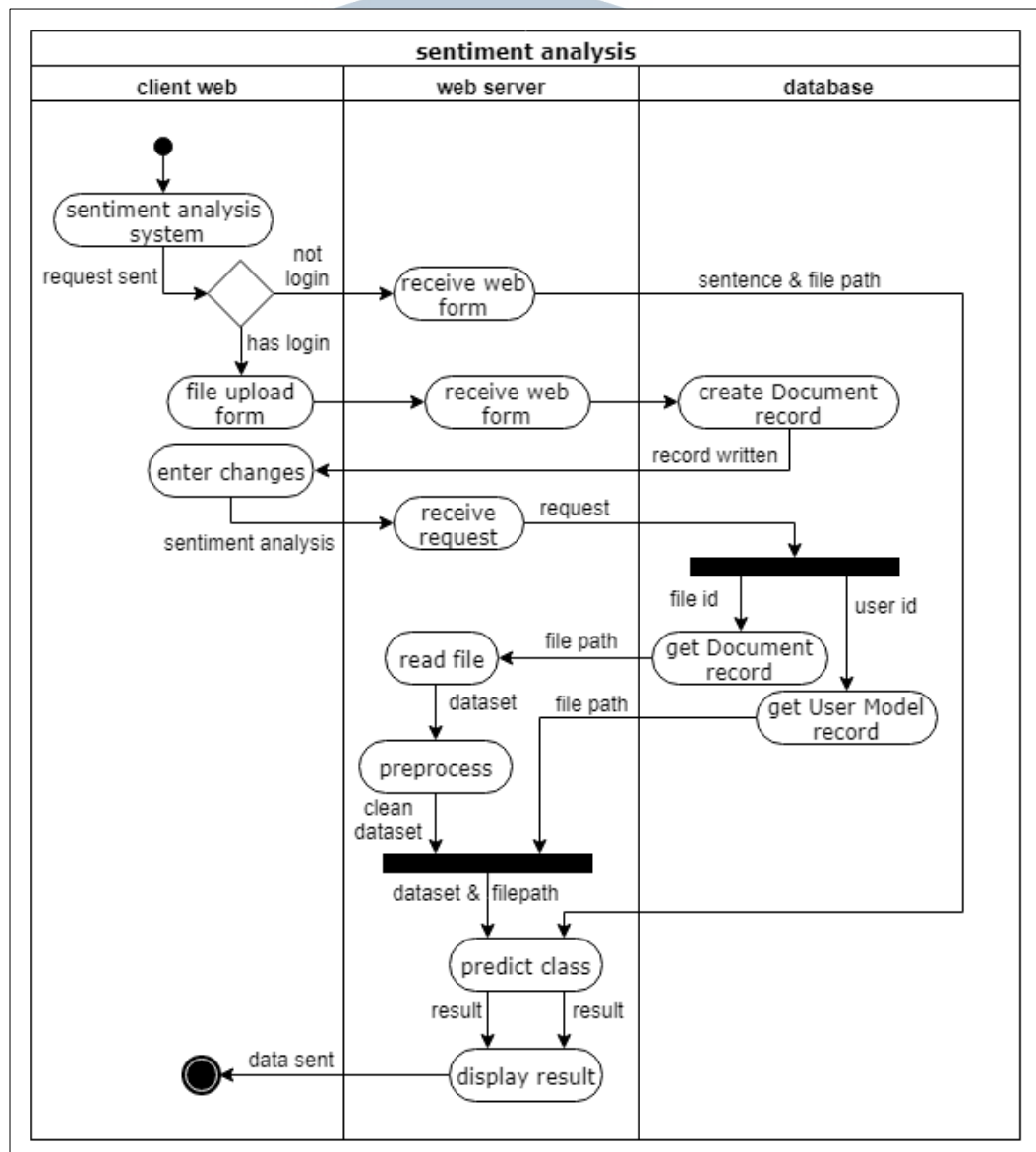
statistic, sentiment analysis, upload feedback, update model, training, dan login/register. User dapat melakukan aktivitas *show statistic* dan *sentiment analysis* tanpa harus melakukan aktivitas *login/register*. Sedangkan untuk melakukan aktivitas *update model*, User harus melakukan *login/register* terlebih dahulu. Untuk melakukan aktivitas *update model*, User harus mengunggah sebuah *file feedback* terlebih dahulu. *File* tersebut akan digunakan dalam aktivitas *training*.



Gambar 3.1 Use Case Diagram Sistem

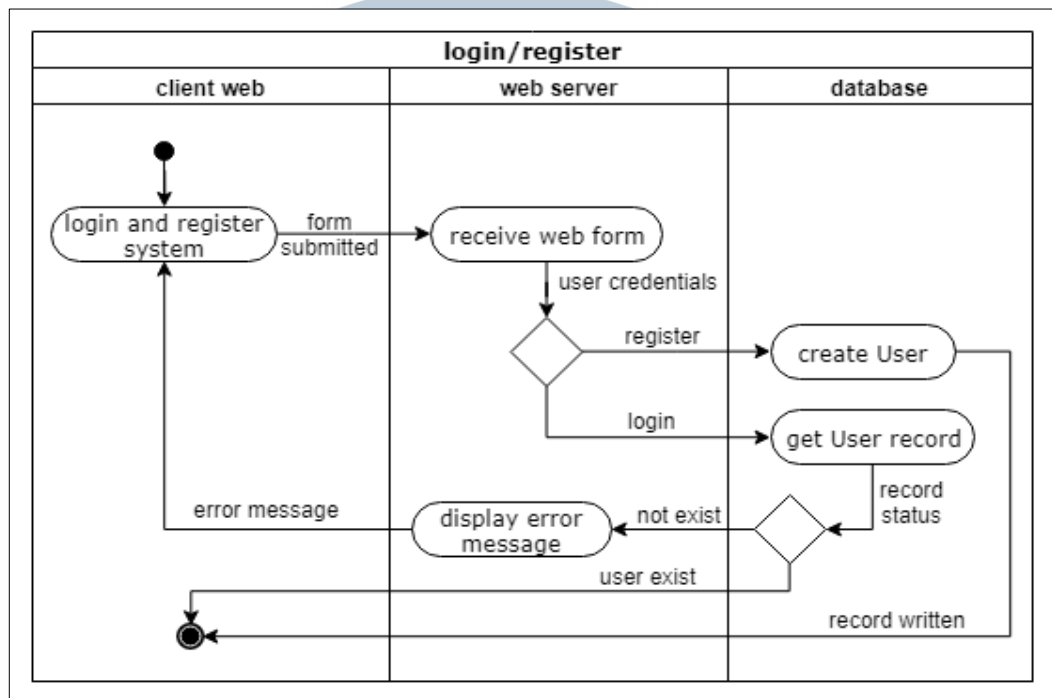
3.2.2 Activity Diagram

Gambar 3.2 merupakan *activity diagram sentiment analysis*. Diagram dimulai dengan *request* yang diterima dari *client web page*. Apabila *request* dibuat sebelum pengguna melakukan *login*, maka pengguna hanya dapat melakukan analisis sentimen dengan model yang sudah disediakan dari aplikasi. Sedangkan untuk pengguna yang sudah melakukan *login*, maka pengguna dapat mengunggah *file* bertipe *csv* yang berisikan *feedback*, sehingga pengguna dapat melakukan analisis sentimen beberapa *feedback* sekaligus.



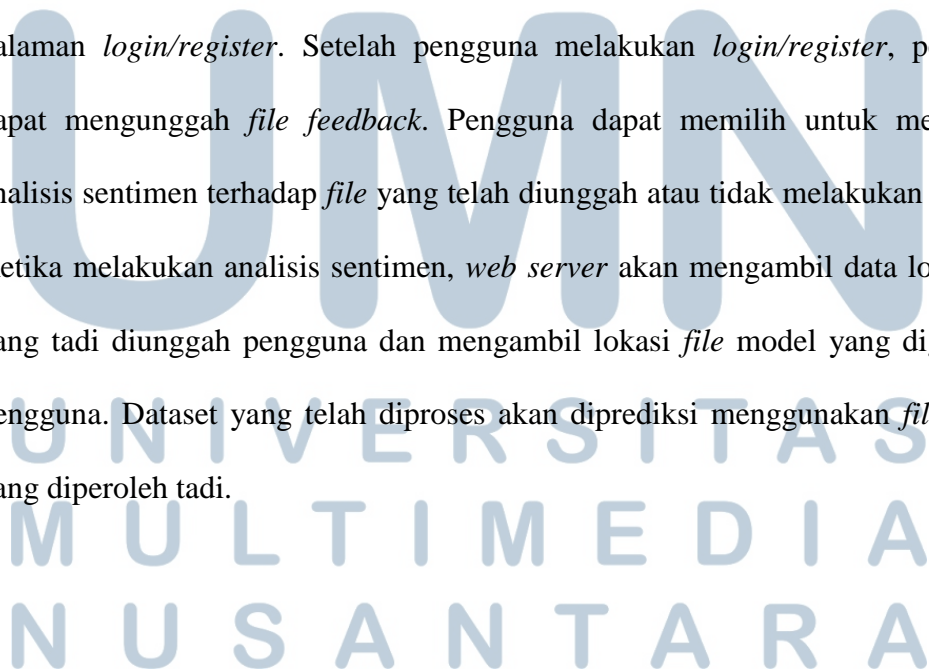
Gambar 3.2 Activity Diagram Sentiment Analysis

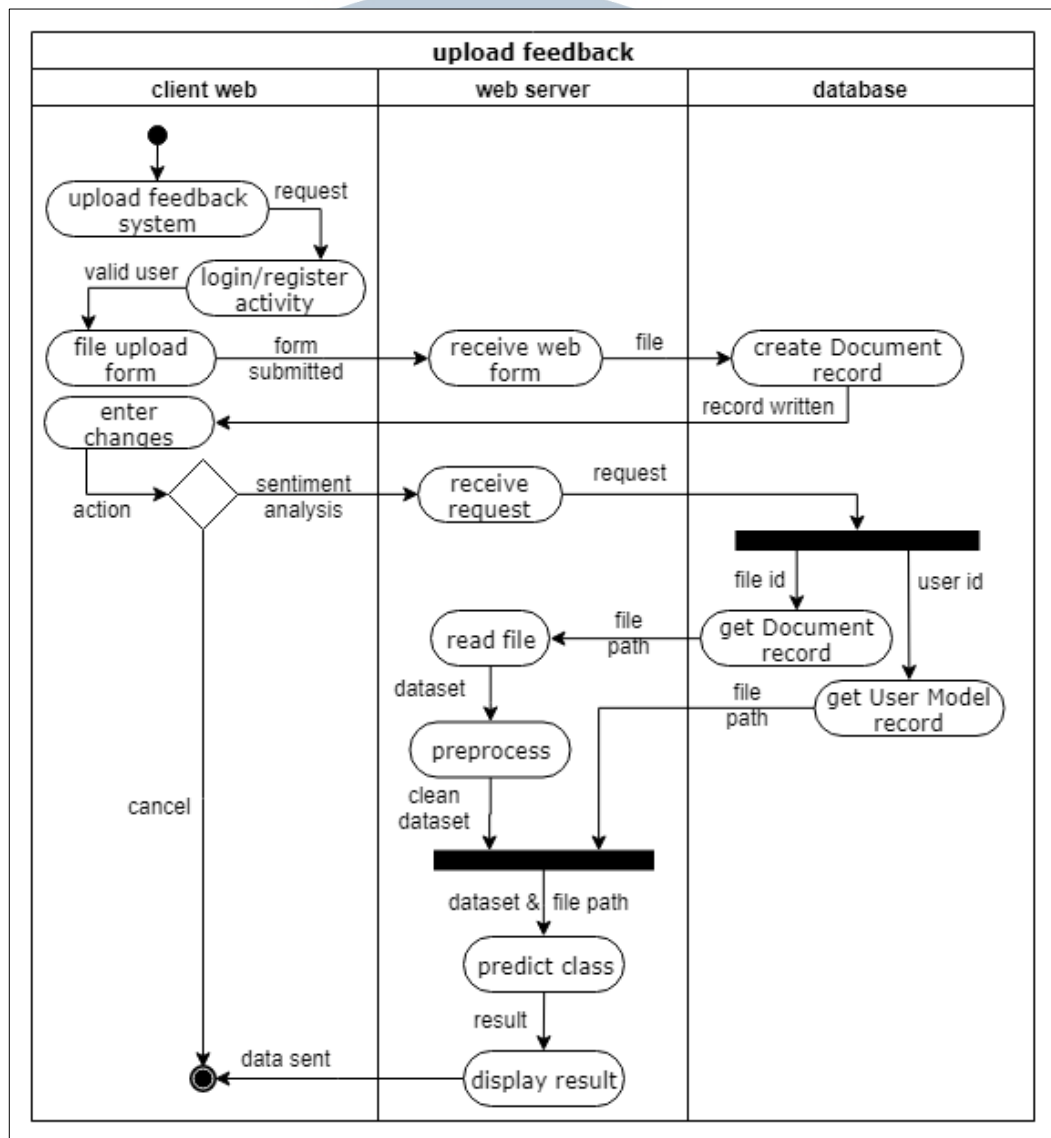
Gambar 3.3 merupakan *activity diagram login/register*. Diagram ini dimulai dengan form yang di *submit* pengguna. Apabila pengguna melakukan registrasi maka data pengguna akan dimasukkan ke *database*. Sedangkan untuk pengguna yang melakukan *login* maka *username* dan *password* yang dimasukkan pengguna akan diautentikasi. Pengguna dengan data yang valid akan diarahkan ke halaman menu.



Gambar 3.3 Activity Diagram Login/Register

Gambar 3.4 merupakan *activity diagram upload feedback*. Diagram dimulai dengan mengecek apakah pengguna sudah melakukan *login/register*. Apabila pengguna belum melakukan *login/register*, maka pengguna akan diarahkan ke halaman *login/register*. Setelah pengguna melakukan *login/register*, pengguna dapat mengunggah *file feedback*. Pengguna dapat memilih untuk melakukan analisis sentimen terhadap *file* yang telah diunggah atau tidak melakukan analisis. Ketika melakukan analisis sentimen, *web server* akan mengambil data lokasi *file* yang tadi diunggah pengguna dan mengambil lokasi *file* model yang digunakan pengguna. Dataset yang telah diproses akan diprediksi menggunakan *file* model yang diperoleh tadi.

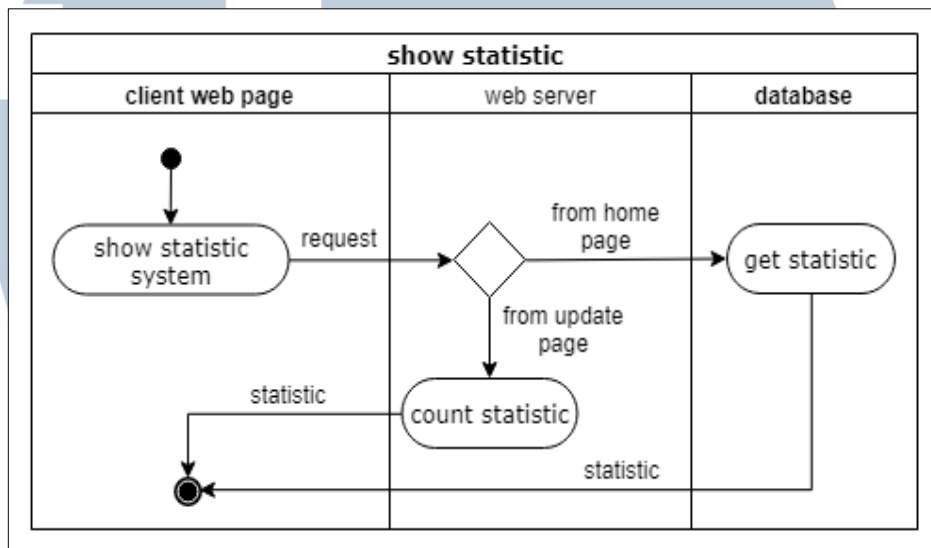




Gambar 3.4 Activity Diagram Upload Feedback

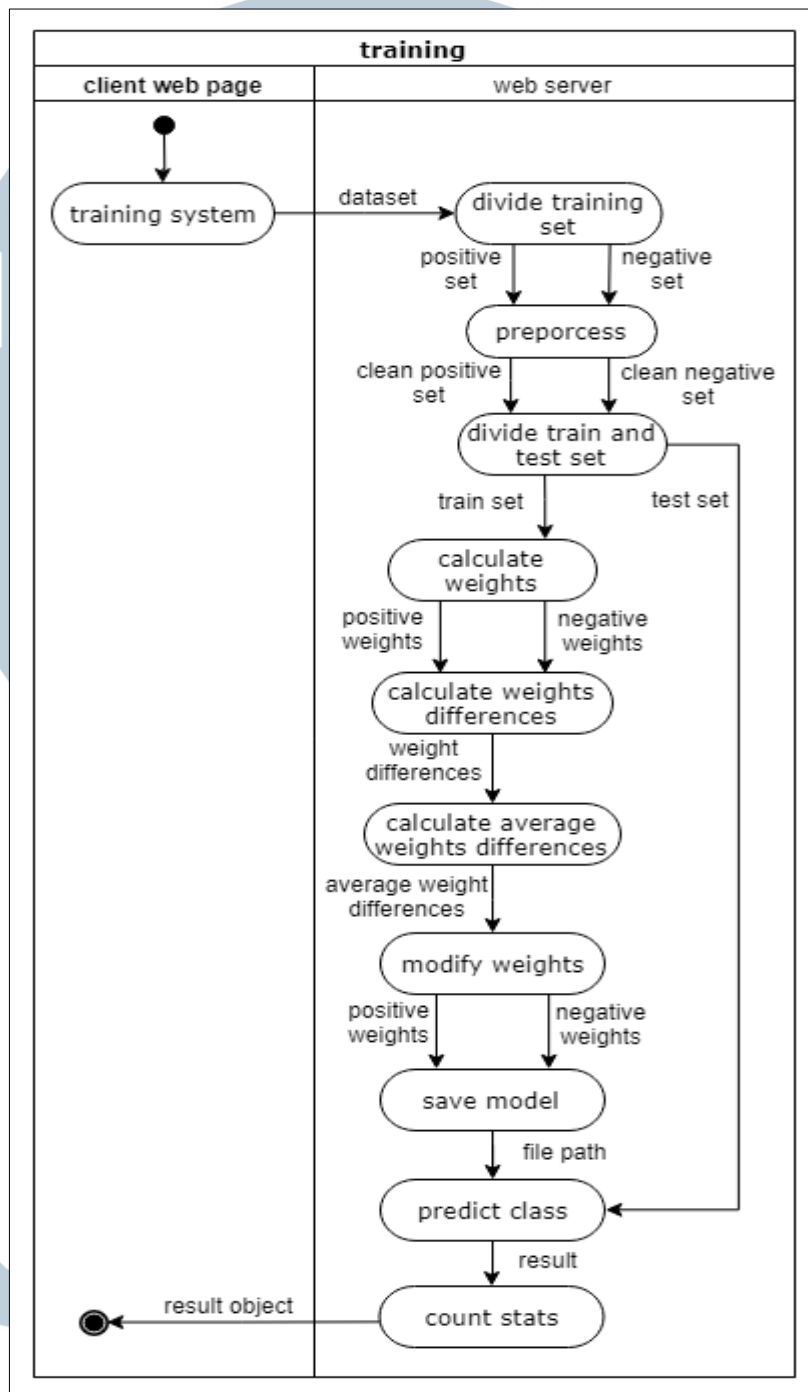
Gambar 3.5 merupakan *activity diagram* show statistic. Diagram ini dimulai dengan *request* yang diterima dari *client web page*. Apabila *request* dikirimkan dari halaman Home, maka *web server* akan mengambil data statistik dari *database*. Data statistik tersebut adalah nilai akurasi model basis yang telah disediakan oleh aplikasi. Pengguna yang tidak melakukan *login* hanya dapat menggunakan model basis yang telah disediakan. Pengguna yang telah melakukan *login* dapat membuat

model sendiri dengan melakukan *update model*. Sedangkan untuk *request* yang dikirimkan dari halaman Update, maka akan dilakukan perhitungan nilai statistik baru.



Gambar 3.5 Activity Diagram Show Statistic

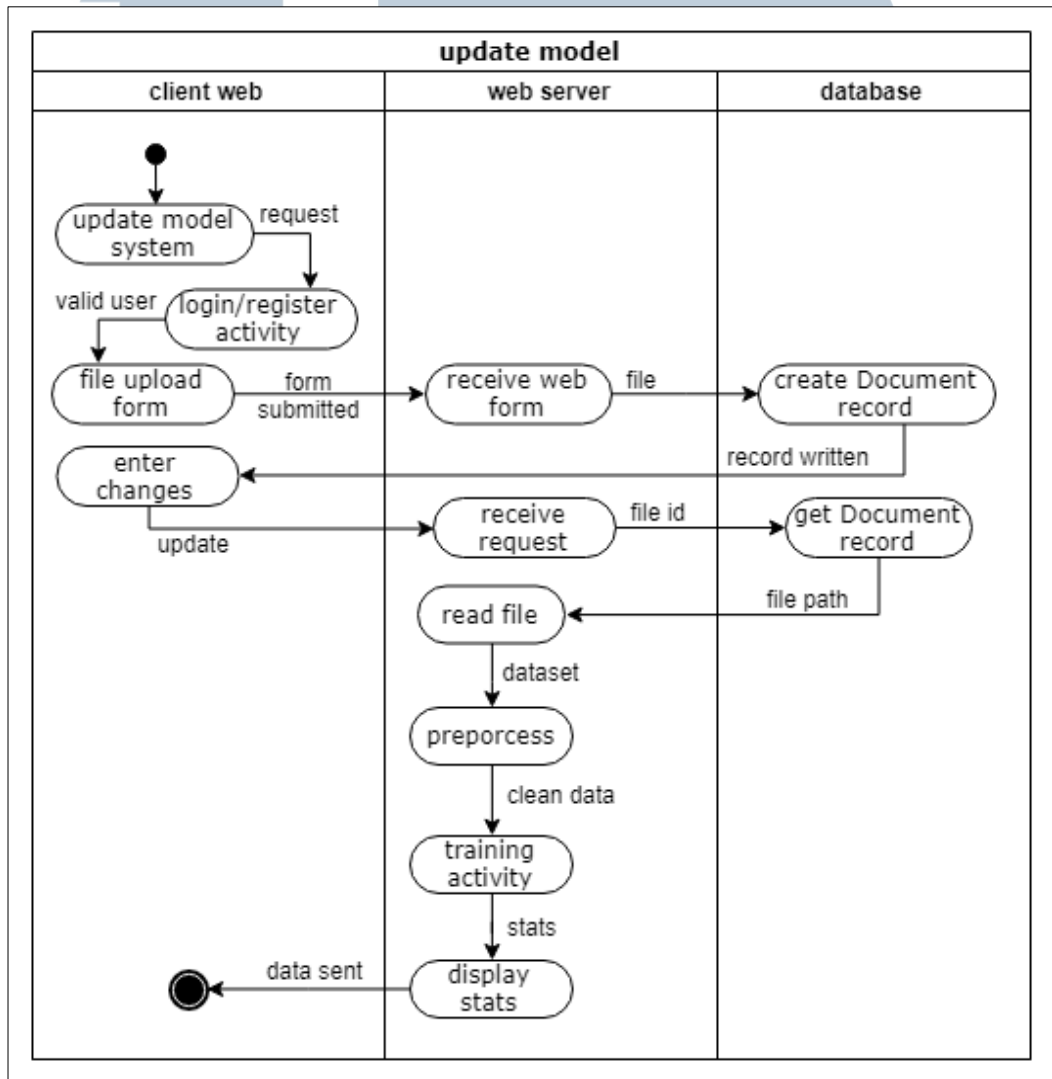
Gambar 3.6 merupakan *activity diagram* training. Diagram dimulai dengan membagi dataset menjadi dataset positif dan negatf. Kedua dataset tersebut di-*preprocessing* terlebih dahulu sebelum dibagi lagi menjadi dataset untuk *training* dan *testing*. Bobot positif dan negatif tiap kata dalam dataset *traning* dihitung. Kemudian selisih antara bobot positif dan negatif tiap kata dihitung lalu dirata-rata. Bobot positif dan negatif tiap kata dimodifikasi berdasarkan nilai rata-rata selisih bobot. Bobot akhir yang diperoleh setelah modifikasi disimpan sebagai model dalam sebuah *file* teks. Model tersebut akan digunakan kembali untuk melakukan prediksi dataset *testing*. Perhitungan nilai statistik model yang baru dibuat berdasarkan hasil prediksi.



Gambar 3.6 Activity Diagram Training

Gambar 3.7 merupakan *activity diagram* update model. Diagram dimulai dengan mengecek apakah pengguna telah melakukan *login/register*. Pengguna yang telah melakukan *login* dapat mengunggah *file* untuk di-*training*. *File* yang akan di-

training diambil lokasinya terlebih dahulu dari *database*. Setelah *file* dibaca dan di-*preprocessing*. Kemudian dataset di-*training* untuk mendapatkan model baru. Hasil dari *training* beserta nilai akurasi modelnya akan ditampilkan pada halaman *web*.

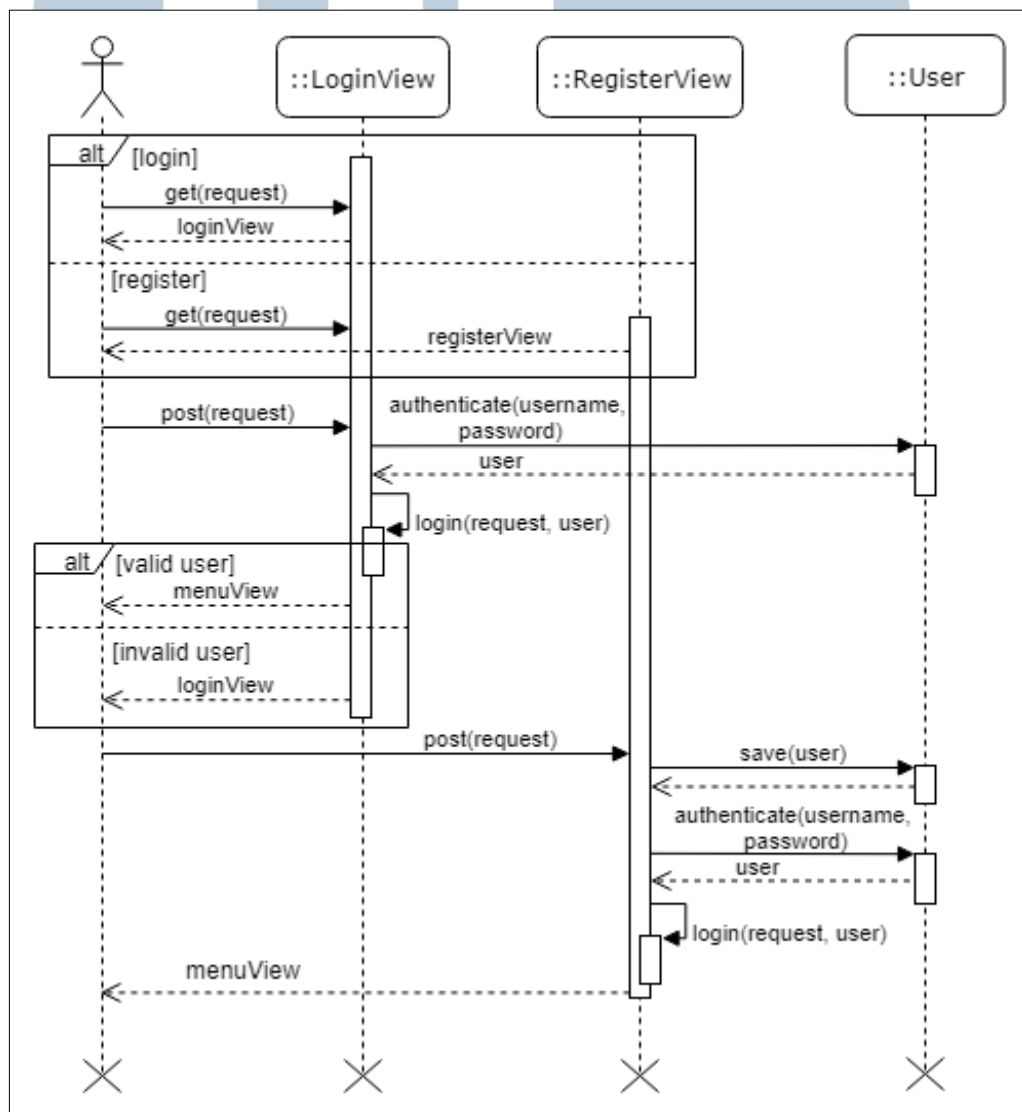


Gambar 3.7 Activity Diagram Update Model

3.2.3 Sequence Diagram

Gambar 3.8 merupakan *sequence diagram* login/register. Diagram dimulai dengan mengirimkan *request* GET ke view. Apabila *request* ditujukan ke halaman *login*, maka view akan mengembalikan *loginView*. Sedangkan *request* yang

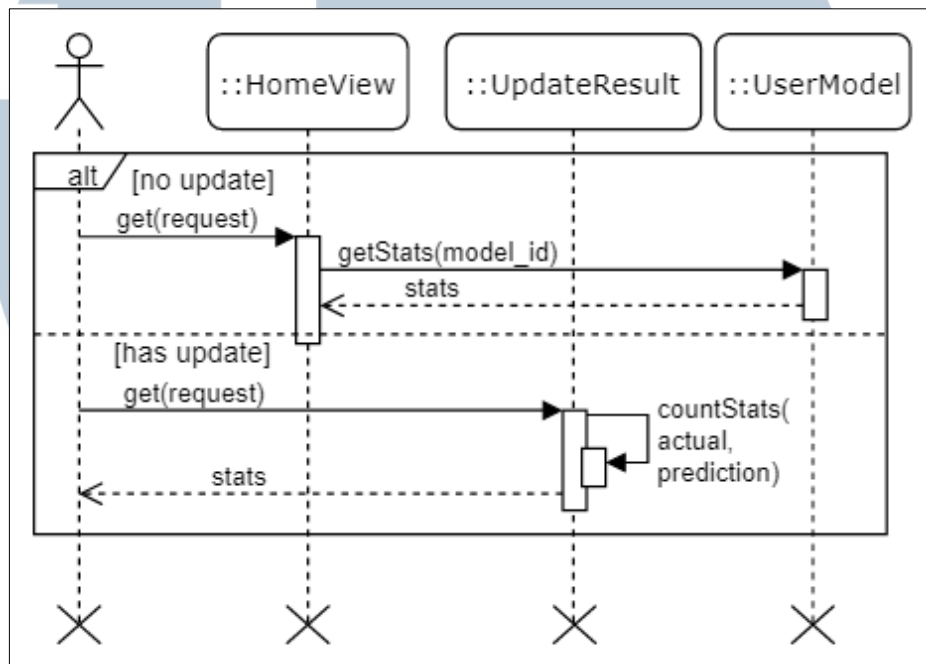
ditujukan ke halaman *register*, maka view akan mengembalikan *registerView*. Ketika pengguna melakukan *login* atau *register*, *request* POST akan dikirimkan ke view. Data pengguna yang melakukan *register* akan disimpan dalam *database*. Saat *login* maupun *register*, *method* *authenticate* akan dijalankan dan hanya pengguna dengan data valid yang dapat masuk ke *view* menu.



Gambar 3.8 Sequence Diagram Login/Register

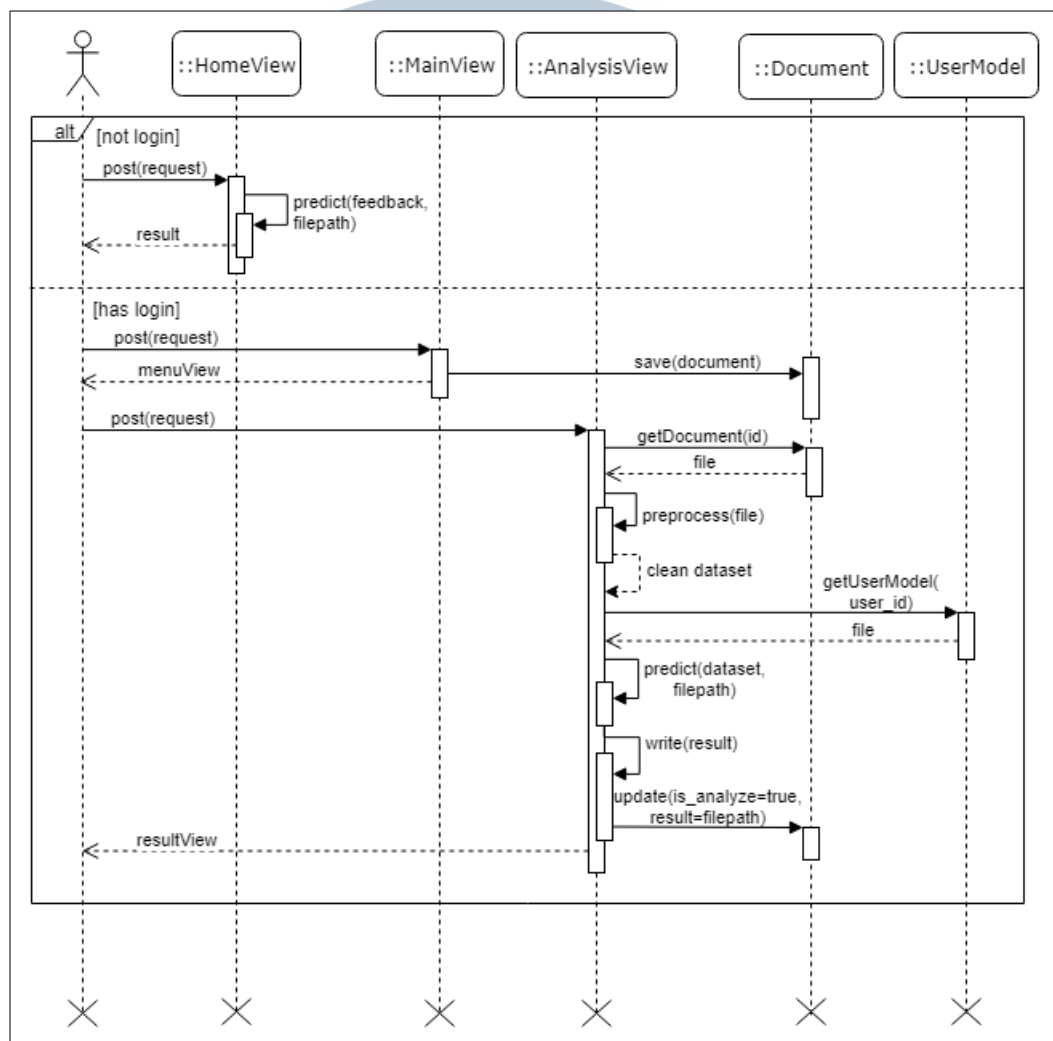
Gambar 3.9 merupakan *sequence diagram* show statistic. Diagram dimulai dengan mengirimkan *request* GET ke view. Jika *request* dikirimkan setelah

melakukan *update* model, maka *method* *countStats* akan dipanggil. Apabila tidak dilakukan *update*, *method* *getStats* akan dijalankan untuk mengambil data statistik dari *database*. Data statistik akan dikembalikan bersama dengan *view*.



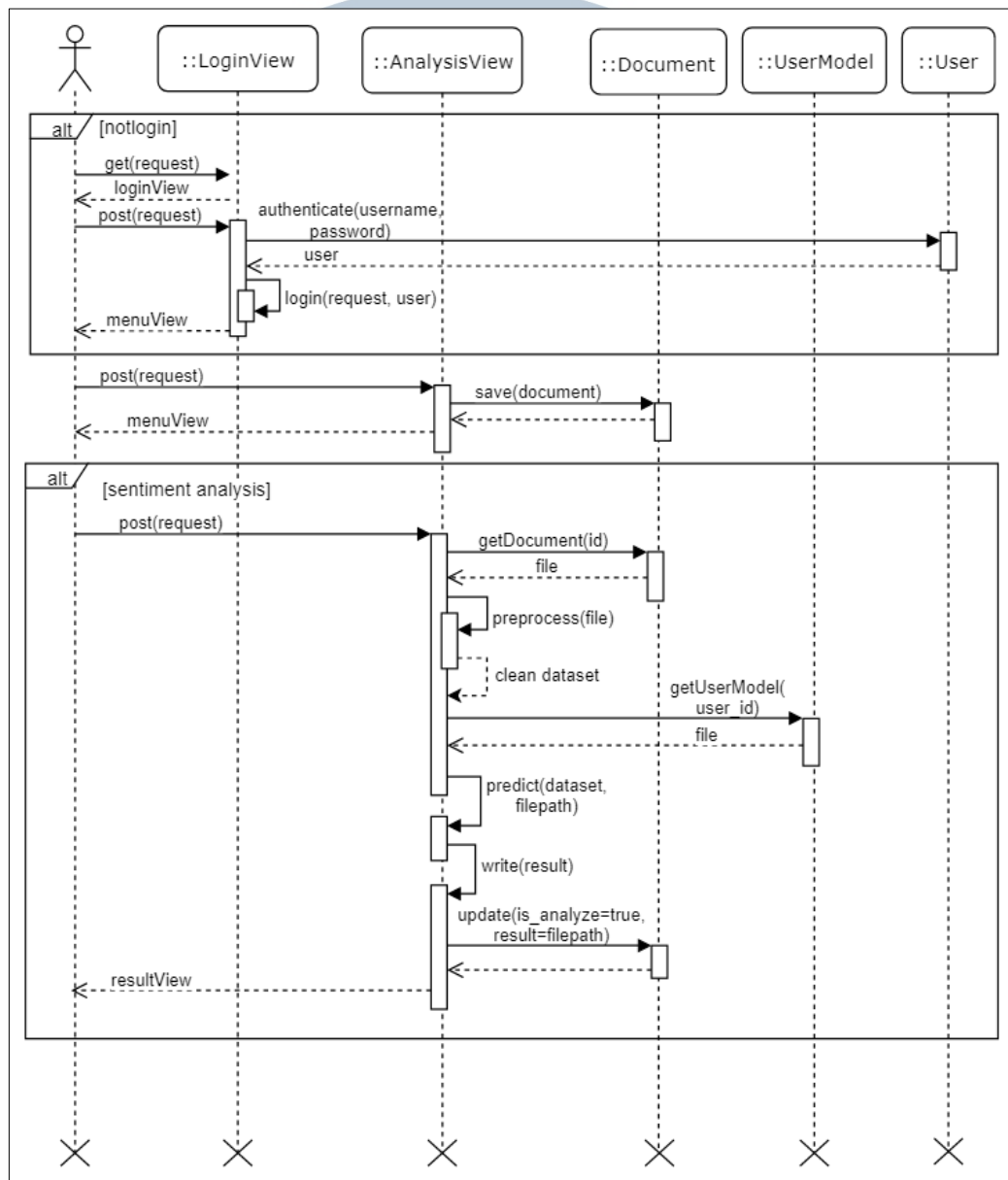
Gambar 3.9 Sequence Diagram Show Statistic

Gambar 3.10 merupakan *sequence diagram* sentiment analysis. Pengguna yang belum melakukan *login* hanya dapat melakukan analisis sentimen menggunakan model yang telah disediakan oleh aplikasi. Sedangkan pengguna yang sudah melakukan *login*, dapat mengunggah *file* ke *database* dan melakukan analisis sentimen menggunakan model yang pengguna tentukan. Hasil dari analisis sentiment akan disimpan ke dalam *file*. Setelah melakukan analisis sentimen, data *file* yang sudah dianalisis akan diperbarui. *Field* *is_analyze* akan diubah menjadi *true* dan nama *file* hasil analisis akan ditambahkan ke dalam tabel.



Gambar 3.10 Sequence Diagram Sentiment Analysis

Gambar 3.11 merupakan *sequence diagram* upload feedback. Diagram dimulai dengan mengirimkan *request* GET ke view. Pengguna yang telah melakukan *login*, dapat mengunggah *file* yang akan digunakan untuk analisis sentimen maupun *training*. *File* tersebut akan dimasukkan ke *database* Document. Pengguna dapat memilih untuk melakukan analisis sentimen atau tidak melakukan analisis.

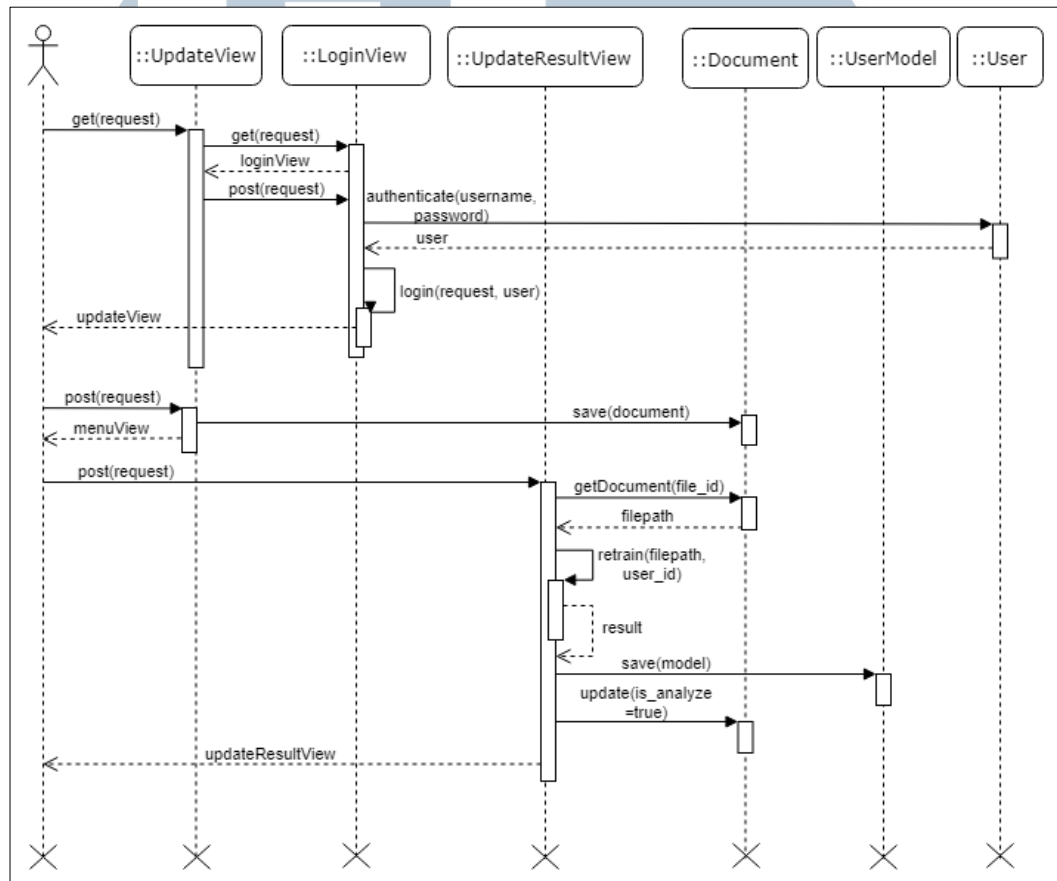


Gambar 3.11 Sequence Diagram Upload Feedback

Gambar 3.12 merupakan *sequence diagram* update model. Diagram dimulai dengan mengecek apakah pengguna telah melakukan *login* atau belum. Pengguna yang telah melakukan *login* dapat mengunggah *file feedback*. *File* yang telah diunggah kemudian di-*training*. Model hasil *training* akan disimpan ke dalam

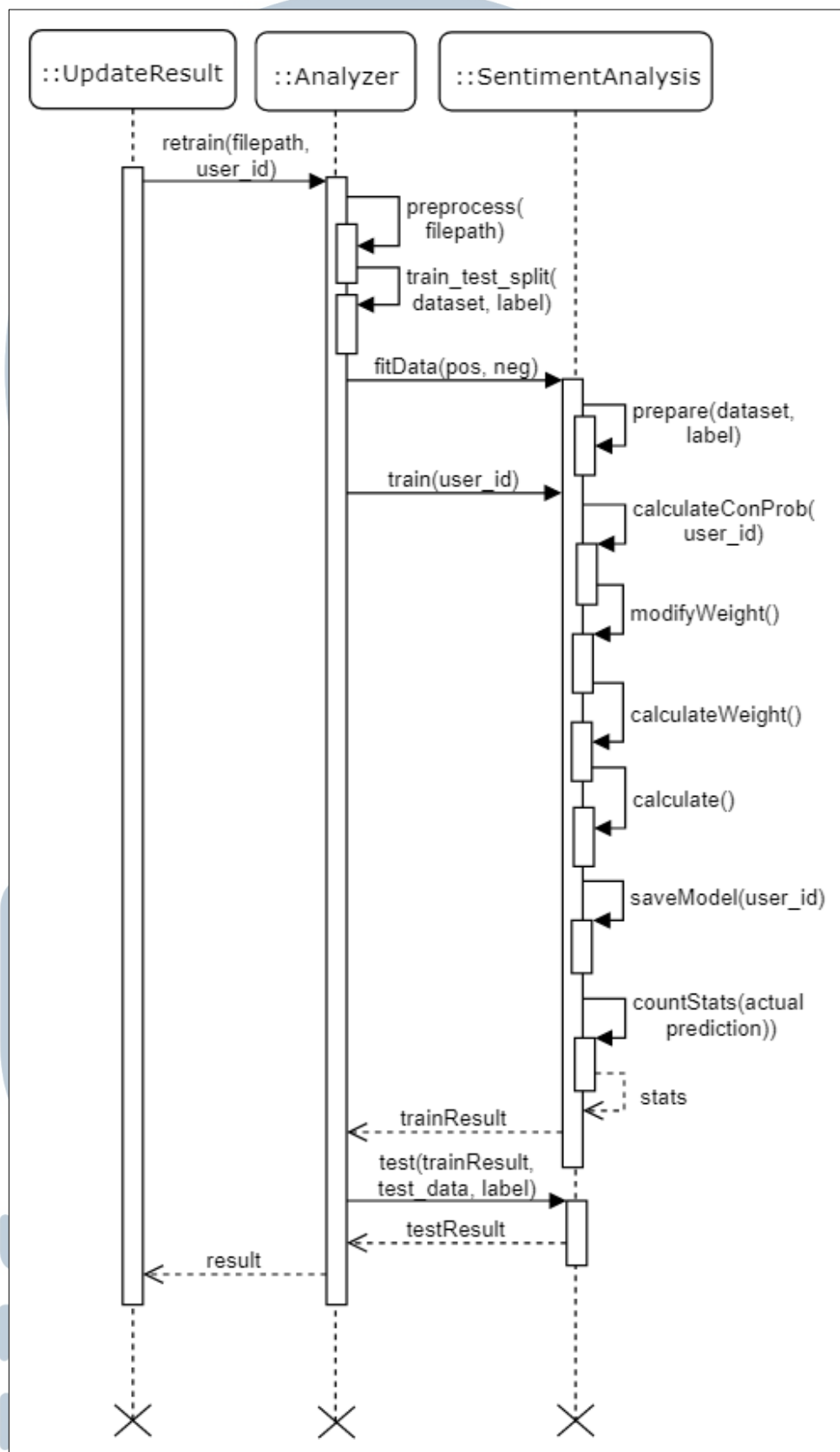
UNIVERSITAS
MULTIMEDIA
NUSANTARA

database pada tabel UserModel. Field *is_analyzed* milik data *file* yang telah di-*training* akan diperbarui menjadi *true* pada tabel Document.



Gambar 3.12 Sequence Diagram Update Model

Gambar 3.13 merupakan *sequence diagram* training. Diagram dimulai dengan panggilan *method* *retrain* ke *class* *Analyzer*. *Method* *preprocess* akan melakukan *preproceesing* data yang akan digunakan untuk *training*. Data kemudian dibagi menjadi *train* dan *test* data. Ketika *method* *train* telah selesai dijalankan *method* *saveModel* dipanggil untuk menyimpan model hasil *training* ke sebuah *file* teks. Kemudian tes dilakukan menggunakan model yang tadi dihasilkan. Hasil dari test dikembalikan ke view.



Gambar 3.13 Sequence Diagram Training

3.2.4 Class Diagram

Class diagram terdiri dari 16 *class* utama dalam sistem, diantaranya terdapat *class controller* dan *model*. *Class controller* pada perancangan sistem ini pada umumnya memiliki *method get* dan *post* yang berfungsi untuk menerima dan memproses *request* yang diterima, serta mengatur apa yang akan ditampilkan pada *view*. *Class controller* tersebut yaitu *class* HomeView, MainView, AnalysisView, UpdateView, ManageView, UpdateResult, LoginView, dan RegisterView.

Terdapat tiga *class model* dalam sistem ini, yaitu *class* Document, UserModel, dan User. *Class model* tidak memiliki *method* karena Django secara otomatis memberikan *API database-abstraction* untuk melakukan CRUD ke *database*. *Class-class* yang terdapat dalam sistem ini memiliki relasi *association* yang menunjukkan keterkaitan antar *class* maupun *object*.



3.2.5 Rancangan Struktur Tabel

Tabel 3.1 merupakan struktur tabel User pada *database*. Tabel ini memiliki 11 kolom. Kolom *is_superuser* bernilai 1 untuk *user* admin dan 0 untuk *user* non admin.

Tabel 3.1 Struktur Tabel User

Nama Kolom	Tipe Data	Atribut
id	int(11)	PRIMARY KEY
password	varchar(128)	NOT NULL
last_login	datetime(6)	DEFAULT NULL
is_superuser	tinyint(1)	NOT NULL
username	varchar(150)	NOT NULL
first_name	varchar(30)	NOT NULL
last_name	varchar(150)	NOT NULL
email	varchar(254)	NOT NULL
is_staff	tinyint(1)	NOT NULL
is_active	tinyint(1)	NOT NULL
date_joined	datetime(6)	NOT NULL

Tabel 3.2 merupakan struktur tabel Document pada *database*. Tabel ini memiliki 6 kolom. Kolom *is_analyze* bernilai 1 untuk dokumen yang sudah dianalisis atau di-*training* dan 0 untuk dokumen yang belum dianalisis atau di-*training*. Kolom *document_type* berisi 'train' untuk dokumen yang digunakan untuk *training* dan 'test' untuk dokumen yang digunakan untuk analisis sentimen.

Kolom *user_id* merupakan *foreign key* yang menunjuk ke kolom *id* pada tabel User.

Tabel 3.2 Struktur Tabel Document

Nama Kolom	Tipe Data	Atribut
id	int(11)	PRIMARY KEY
document	varchar(100)	NOT NULL

Tabel 3.2 Struktur Tabel Document (Lanjutan)

Nama Kolom	Tipe Data	Atribut
is_analyze	tinyint(1)	NOT NULL
uploaded_time	datetime(6)	NOT NULL
document_type	varchar(6)	NOT NULL
user_id	int(11)	FOREIGN KEY
result	varchar(255)	DEFAULT NULL

Tabel 3.3 merupakan struktur tabel UserModel pada *database*. Tabel ini memiliki 8 kolom. Kolom *is_default* bernilai 1 untuk model yang ditetapkan sebagai model standar oleh pengguna dan 0 untuk model yang bukan standar model. Seorang pengguna hanya dapat memiliki satu model standar. Kolom *user_id* merupakan *foreign key* yang menunjuk ke kolom *id* pada tabel User. Kolom *document_id* merupakan *foreign key* yang menunjuk ke kolom *id* pada tabel Document.

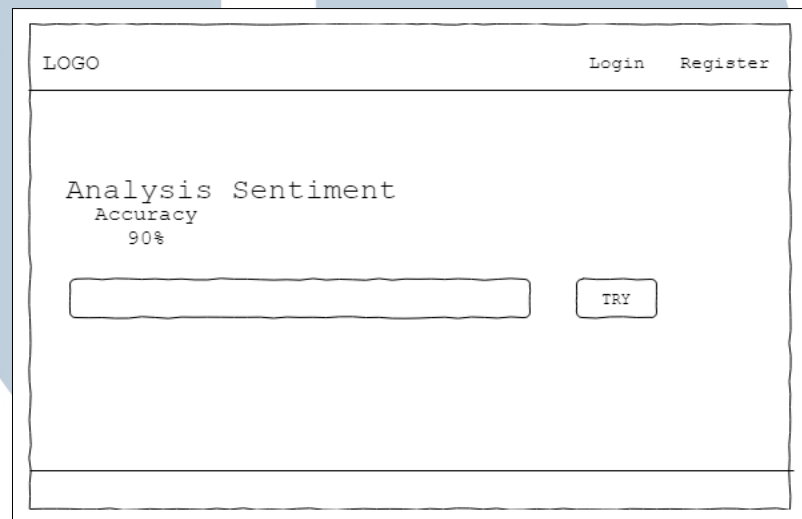
Tabel 3.3 Struktur Tabel Usermodel

Nama Kolom	Tipe Data	Atribut
id	int(11)	PRIMARY KEY
model_file	varchar(255)	NOT NULL
uploaded_time	datetime(6)	NOT NULL
train_acc	double	NOT NULL
test_acc	double	NOT NULL
is_default	tinyint(1)	NOT NULL
document_id	int(11)	FOREIGN KEY
user_id	int(11)	FOREIGN KEY

3.2.6 Rancangan Tampilan Antarmuka

Tampilan antarmuka aplikasi *web* terbagi menjadi tujuh halaman, yaitu halaman Home, Login, Register, Menu, Manage, Preview, dan Result.

Gambar 3.15 merupakan rancangan antarmuka halaman Home. Halaman ini berisi form analisis sentiment, dimana pengguna dapat menganalisis sentimen suatu *feedback*. Informasi akurasi yang ditampilkan merupakan akurasi model yang digunakan aplikasi.

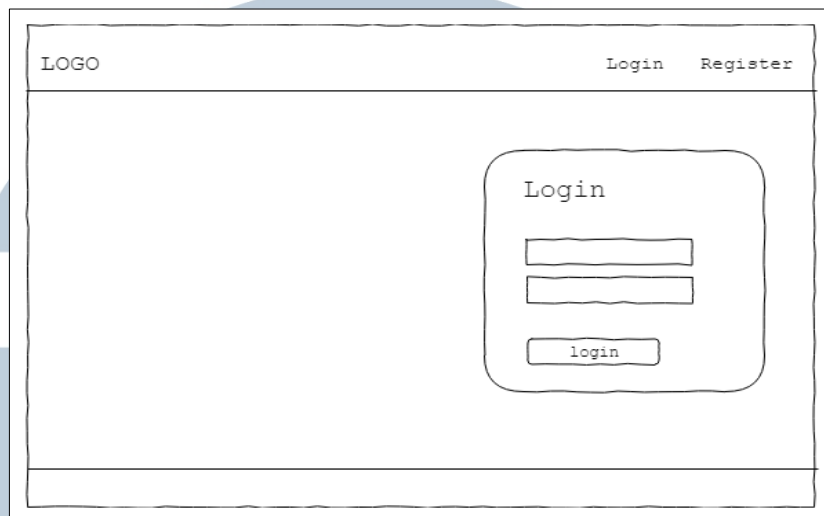


The image shows a wireframe of a web application's home page. At the top left is a placeholder for a 'LOGO'. At the top right are two links: 'Login' and 'Register'. The main content area is titled 'Analysis Sentiment' and displays 'Accuracy 90%'. Below this, there is a large rectangular input field for text and a 'TRY' button to the right of it. The entire interface is enclosed in a simple rectangular border.

Gambar 3.15 Halaman Home

Gambar 3.16 dan 3.17 merupakan rancangan antarmuka halaman Login dan Register. Pengguna dapat melakukan *login/register* pada aplikasi ini untuk dapat menggunakan fitur-fitur tambahan. Form Login terdiri dari dua *input*, yaitu *username* dan *password*. Form Register terdiri dari empat *input*, yaitu *email*, *username*, *password*, dan *repeat password*.

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



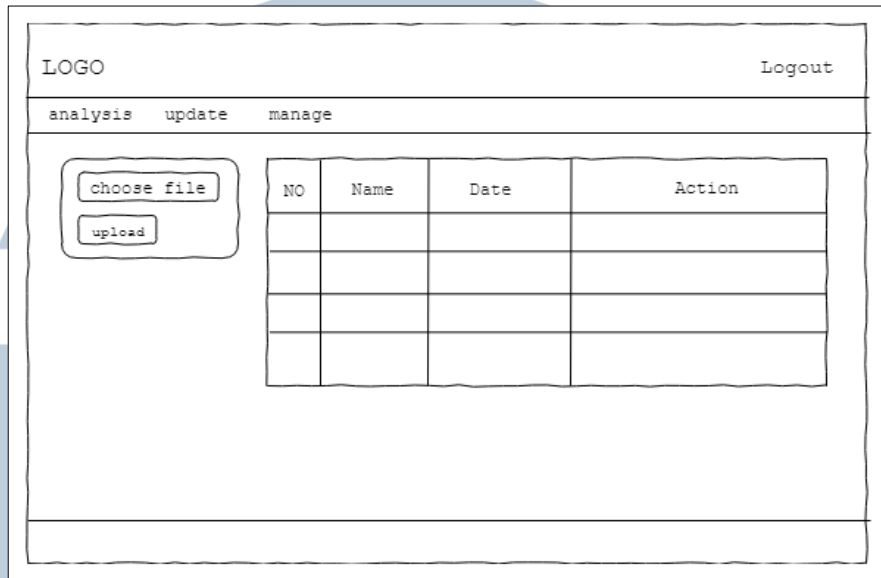
Gambar 3.16 Halaman Login



Gambar 3.17 Halaman Register

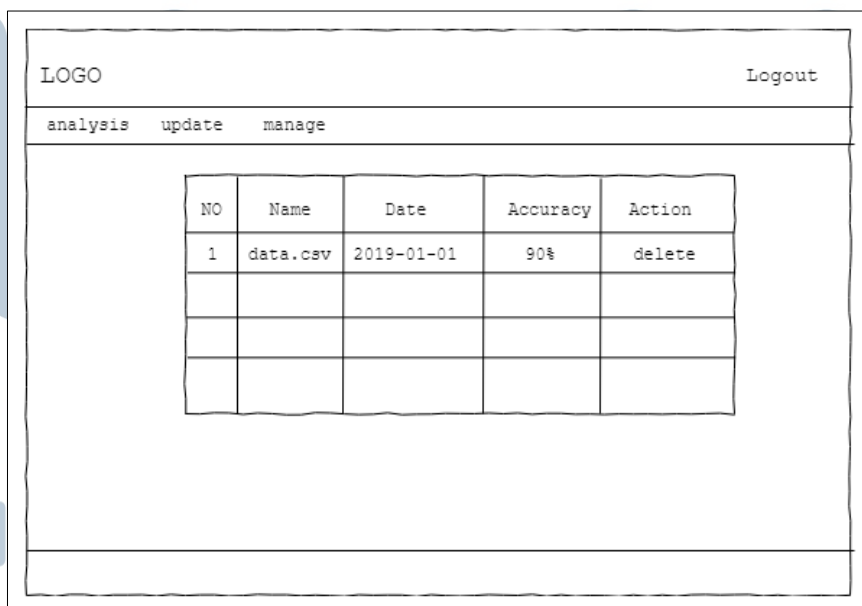
Gambar 3.18 merupakan rancangan antarmuka halaman Menu. Halaman ini berisi tombol menu *analysis*, tombol menu *update*, tombol menu *manage*, tabel daftar *file* dan form *upload feedback*. Selain itu halaman Menu memiliki form *file upload* dan tabel yang menampilkan *file-file* yang sudah diunggah pengguna.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



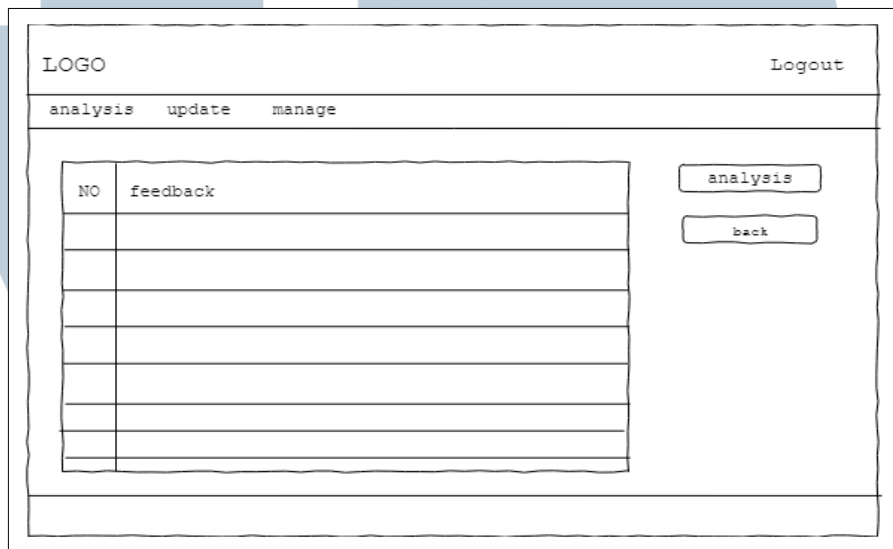
Gambar 3.18 Halaman Menu

Gambar 3.19 merupakan rancangan antarmuka halaman Manage. Halaman ini berisi tombol menu *analysis*, tombol menu *update*, tombol menu *manage*, dan tabel daftar model. Halaman Manage berfungsi untuk mengatur model-model hasil dari *training* yang dilakukan pengguna.

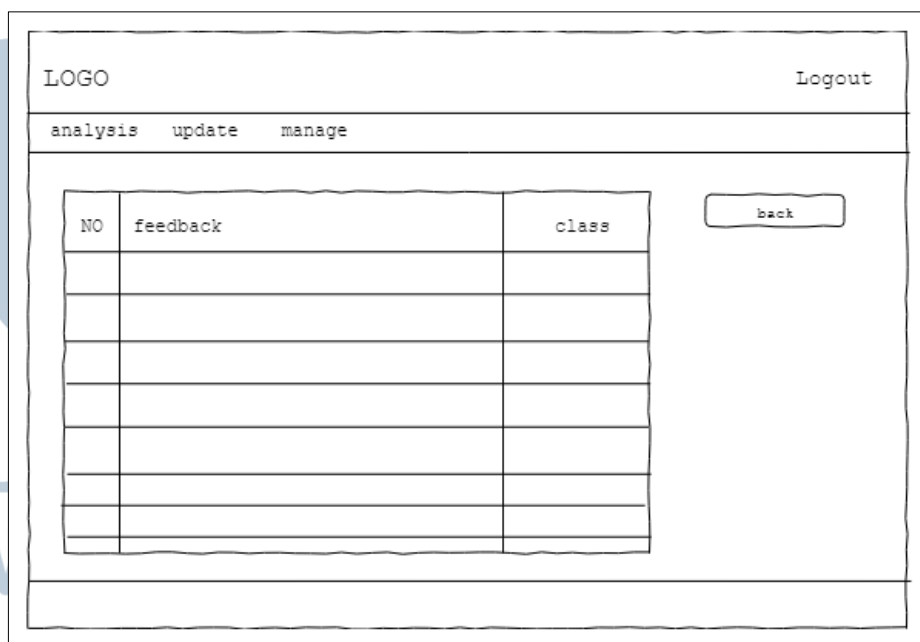


Gambar 3.19 Halaman Manage

Gambar 3.20 merupakan rancangan antarmuka halaman Preview. Halaman ini berisi tombol *analysis/training*, tombol *back* dan tabel daftar *feedback*. Halaman Result pada gambar 3.21, menampilkan hasil dari analisis maupun *training* yang dilakukan pengguna.



Gambar 3.20 Halaman Preview



Gambar 3.21 Halaman Result