

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1 Analisis Kebutuhan Software

Sistem informasi manajemen polis asuransi jiwa berbasis web memungkinkan pengelolaan data polis secara digital, terpusat, dan real-time. Sistem ini mempermudah input, pemantauan, pencetakan dokumen polis kepada tertanggung secara efisien. Berikut adalah spesifikasi kebutuhan (*system requirement*):

1. Kebutuhan Fungsional Admin Operasional

- a. Admin Operasional bisa melakukan login dengan akun yang sudah dibuat;
- b. Admin Operasional bisa melakukan download template upload data calon tertanggung;
- c. Admin Operasional bisa melakukan upload dan melihat data calon tertanggung;
- d. Admin Operasional bisa melakukan upload dokumen pendukung data calon tertanggung;
- e. Admin Operasional bisa mencetak dan mendownload dokumen polis seperti *cover note* serta surat tolak;
- f. Admin Operasional bisa mendownload laporan rekap data pertanggung;

2. Kebutuhan SPV Operasional

- a. SPV Operasional bisa melakukan login dengan akun yang sudah dibuat;
- b. SPV Operasional bisa melakukan keputusan data calon tertanggung disetujui atau ditolak;

3. Kebutuhan Fungsional Administrator

- a. Administrator bisa melakukan *login* dengan akun yang sudah dibuat;

- b. Administrator bisa melakukan pengelolaan *user* seperti menambah, mengedit dan menonaktifkan akun pengguna;
- c. Administrator bisa melakukan penentuan *role* akses setiap *user*;

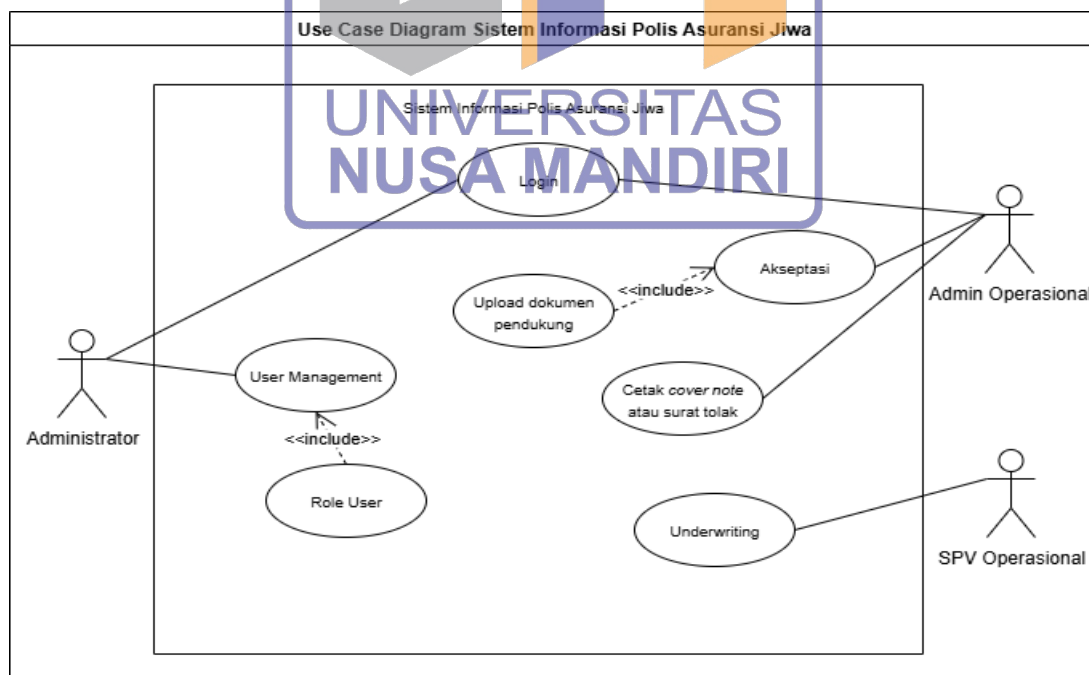
4.2 Desain

4.2.1 Desain Pemodelan Sistem

Desain pemodelan sistem adalah proses menggambarkan bagaimana sistem akan bekerja, baik dari sisi struktur data, alur proses, hingga interaksi antar komponen dan pengguna, dalam bentuk visual atau diagram. Tujuan utamanya adalah untuk memberikan gambaran yang jelas dan terstruktur sebelum sistem dibangun.

4.2.1.1 Use Case Diagram

Pemodelan *Use Case Diagram* adalah representasi visual dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara sistem (*software* atau aplikasi) dan aktor (pengguna) dalam mencapai tujuan tertentu.



Gambar IV. 1 Use Case Diagram Sistem Informasi Polis Asuransi

Gambar IV.1 menggambarkan *Use Case Diagram* dari *Sistem Informasi Polis Asuransi Jiwa*, yang menunjukkan bagaimana interaksi tiga aktor utama yaitu Administrator, Admin Operasional dan SPV Operasional terhadap sistem sesuai dengan fungsi dan wewenangnya masing-masing.

Tabel IV. 1 Deskripsi Use Case Diagram Login

Use Case Name	:	Login
Use Case Description	:	Administrator, Admin Operasional dan SPV Operasional dapat masuk ke dalam sistem untuk mengakses fungsionalitas sistem Login
Actors	:	Administrator, Admin Operasional dan SPV Operasional
Pre-Condition	:	1. Pengguna harus terhubung ke jaringan intranet sistem 2. Mengakses form login 3. Masukan User Name dan Password
Post-Condition	:	Jika validasi sesuai dan login berhasil, <i>actor</i> bisa mengakses masuk ke dalam system
Fault Condition	:	Jika validasi tidak sesuai, sistem akan menolak dan proses selesai
Main Scenarios	Serial No.	Step
Administrator, Admin Operasional dan SPV Operasional	1	Masukan User Name
	2	Masukan Password
	3	Pengecekan apakah valid User Name, Password yang dimasukan
		Mengizinkan untuk mengakses sistem sesuai dengan hak akses masing-masing actor
Extentions*	2a	User Name tidak valid sistem menampilkan pesan <i>error</i>

Tabel IV. 2 Deskripsi Use Case Diagram User Management

Use Case Name	:	User Management
Use Case Description	:	Administrator mengelola data pengguna sistem, termasuk menambahkan, mengedit, menonaktifkan, atau menghapus akun user
Actors	:	Administrator
Pre-Condition	:	Administrator sudah berhasil login ke dalam system
Post-Condition	:	Data user berhasil ditambahkan, diperbarui, dinonaktifkan, atau dihapus sesuai tindakan yang dilakukan.
Fault Condition	:	Jika User Name yang dimasukan sudah digunakan
Main Scenarios	Serial No.	Step
Administrator	1	Pilih Menu User Management

	2	Jika Input data baru isi data yang telah disediakan
	3	Jika Edit atau delete data pilih user terlebih dahulu
	4	Pengecekan apakah data yang dimasukan sudah sesuai atau tidak
	5	Tekan tombol simpan untuk selesai
Extensions*	2a	User Name sudah digunakan sistem menampilkan pesan <i>error</i>

Tabel IV. 3 Deskripsi Use Case Diagram Role User

Use Case Name	:	Role User
Use Case Description	:	Administrator mengatur peran (<i>role</i>) masing-masing user sesuai hak akses yang dibutuhkan
Actors	:	Administrator
Pre-Condition	:	Administrator sudah membuat akun <i>user</i>
Post-Condition	:	Data Role user sudah sesuai pengaturan.
Fault Condition	:	Jika gagal menyimpan <i>role user</i>
Main Scenarios	Serial No.	Step
Administrator	1	Pilih Menu User Management
	2	Memilih akun <i>user</i> yang akan diatur perannya (<i>role</i>)
	3	Memilih hak akses <i>user</i> sesuai yang dibutuhkan
	4	Tekan tombol <i>simpan</i> untuk selesai
Extensions*	4a	Jika gagal <i>simpan</i> Sistem menampilkan pesan <i>error</i>

Tabel IV. 4 Deskripsi Use Case Diagram Akseptasi

Use Case Name	:	Akseptasi
Use Case Description	:	Admin Operasional melakukan proses akseptasi data calon bertanggung. Proses meliputi pengunduhan template data, pengunggahan dan submit data calon bertanggung,
Actors	:	Admin Operasional
Pre-Condition	:	Admin Operasional telah login ke sistem dan memiliki hak akses akseptasi
Post-Condition	:	Data calon bertanggung berhasil diunggah dan tersimpan di sistem.
Fault Condition	:	1. Jika data calon bertanggung sudah Pernah diupload 2. Jika Ref Number dan No Loan tidak diisi 3. Jika No KTP tidak sesuai
Main Scenarios	Serial No.	Step
Admin Operasional	1	Pilih Menu Covering
	2	Melakukan download template upload

Extensions*	3	Mengisi data calon bertanggung di template kemudian memilih file yang akan diupload
	4	Upload Data Calon Bertanggung
	5	Tekan tombol submit untuk proses data
	4a	Data Calon Bertanggung sudah pernah diupload sistem menampilkan pesan <i>error</i>
	4b	Ref Number dan No Loan tidak sesuai ketentuan Sistem menampilkan pesan <i>error</i>
	4c	No KTP harus 16 Karakter Sistem menampilkan pesan <i>error</i>
	5a	Jika gagal submit data Sistem menampilkan pesan <i>error</i>

Tabel IV. 5 Deskripsi Use Case Diagram Upload Dokumen Pendukung

Use Case Name	:	Upload Dokumen Pendukung
Use Case Description	:	Admin Operasional melakukan pengunggahan dokumen pendukung yang relevan.
Actors	:	Admin Operasional
Pre-Condition	:	Admin Operasional telah melakukan upload data calon bertanggung
Post-Condition	:	Dokumen calon bertanggung berhasil diunggah dan tersimpan di sistem.
Fault Condition	:	Jika ukuran file dokumen terlalu besar
Main Scenarios	Serial No.	Step
Admin Operasional	1	Memilih data calon bertanggung
	2	Pilih Dokumen pendukung yang ingin diupload
	3	Melakukan Upload dokumen pendukung yang dibutuhkan
	4	Tekan tombol simpan untuk upload dokumen
Extensions*	2a	Ukuran File melebihi 4MB sistem menampilkan pesan <i>error</i>
	4a	Jika gagal simpan Sistem menampilkan pesan <i>error</i>

Tabel IV. 6 Deskripsi Use Case Diagram Underwriting

Use Case Name	:	Underwriting
Use Case Description	:	SPV Operasional melakukan proses pengambilan keputusan terhadap data calon bertanggung yang telah melewati proses akseptasi. SPV Operasional dapat menentukan apakah permohonan polis disetujui atau ditolak.
Actors	:	SPV Operasional

Pre-Condition	:	Data calon bertanggung dan dokumen pendukung telah berhasil diunggah
Post-Condition	:	Keputusan underwriting tercatat di sistem
Fault Condition	:	Jika gagal menyimpan Keputusan underwriting
Main Scenarios	Serial No.	Step
SPV Operasional	1	Pilih Menu Underwriting
	2	Melakukan Pencarian data calon bertanggung
	3	Melakukan peninjauan data dan validasi kelengkapan dokumen
	4	Jika disetujui tekan tombol <i>Approve</i>
	5	Jika ditolak tekan tombol <i>Reject</i>
Extensions*	4a	Jika gagal proses <i>Approve</i> sistem menampilkan pesan <i>error</i>
	5a	Jika gagal proses <i>Reject</i> Sistem menampilkan pesan <i>error</i>

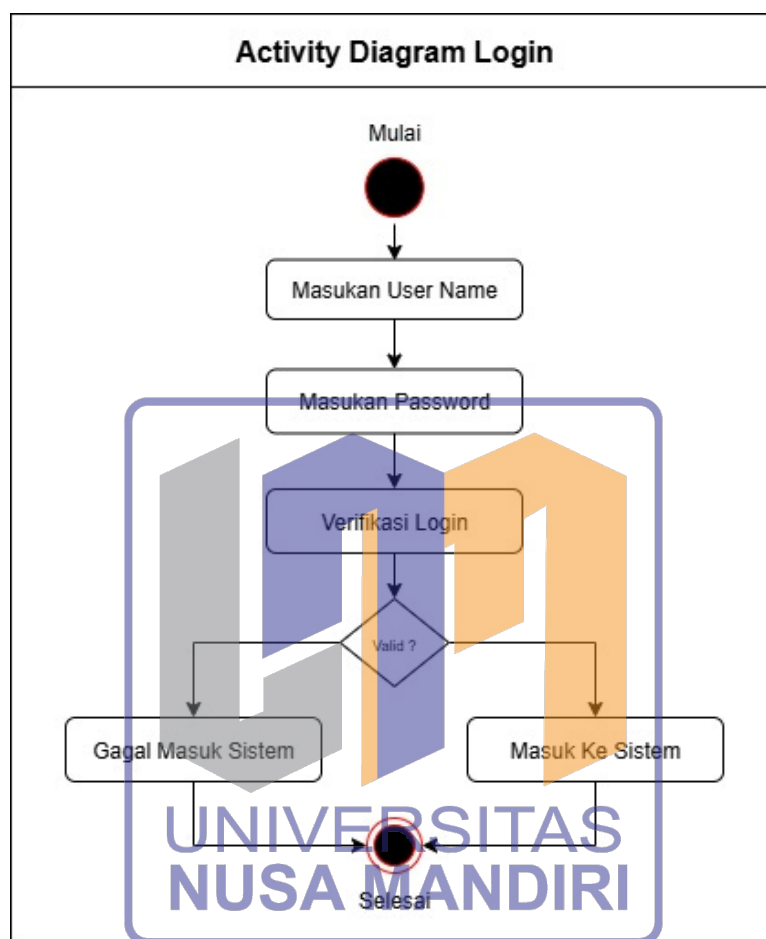
Tabel IV. 7 Deskripsi Use Case Diagram Cetak Cover Note atau Surat Tolak

Use Case Name	:	Cetak <i>Cover Note</i> atau Surat Tolak
Use Case Description	:	Admin Operasional <i>mencetak</i> dokumen terkait seperti <i>cover note</i> atau surat tolak.
Actors	:	Admin Operasional
Pre-Condition	:	Data calon bertanggung sudah dilakukan keputusan oleh SPV Operasional
Post-Condition	:	Dokumen hasil keputusan dapat dicetak atau diunduh.
Fault Condition	:	Jika gagal cetak atau unduh <i>cover note</i> atau surat cetak
Main Scenarios	Serial No.	Step
Admin Operasional	1	Pilih Menu Underwriting
	2	Melakukan pencarian data bertanggung
	3	Jika status sudah disetujui cetak <i>cover note</i>
	4	Jika status sudah ditolak cetak surat tolak
Extensions*	3a	Jika gagal cetak <i>cover note</i> sistem menampilkan pesan <i>error</i>
	4a	Jika gagal cetak surat tolak Sistem menampilkan pesan <i>error</i>

4.2.1.2 Activity Diagram

1. Activity Diagram Login

Berikut tampilan Activity Diagram Login, guna menerapkan alur utama dari *use case login* sistem.

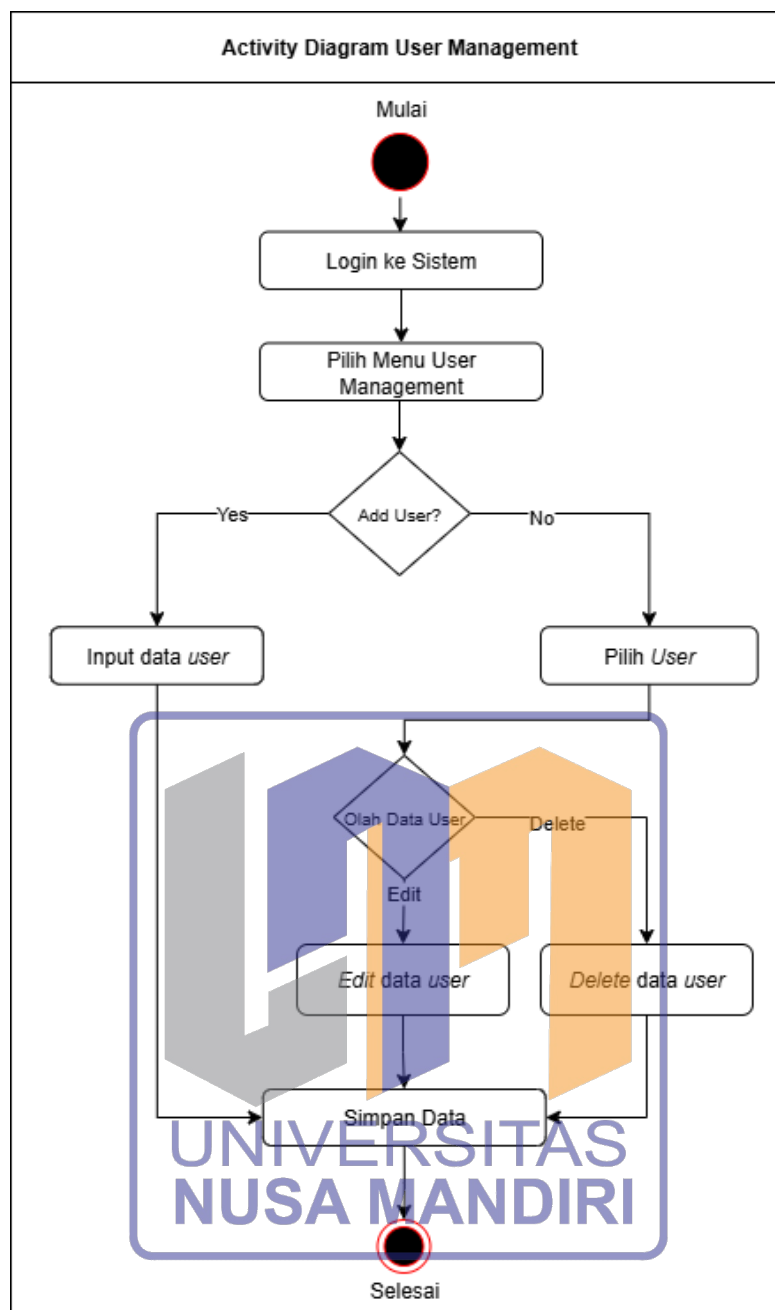


Gambar IV. 2 Activity Diagram Login

Gambar IV.2 Pada Activity Diagram Login menggambarkan aktivitas ketika aktor melakukan *login* ke dalam *system* untuk mengakses ke Menu-menu yang akan diakses atau digunakan.

2. Activity Diagram User Management

Berikut tampilan *Activity User Management*, guna menerapkan alur utama dari *use case User Management*.

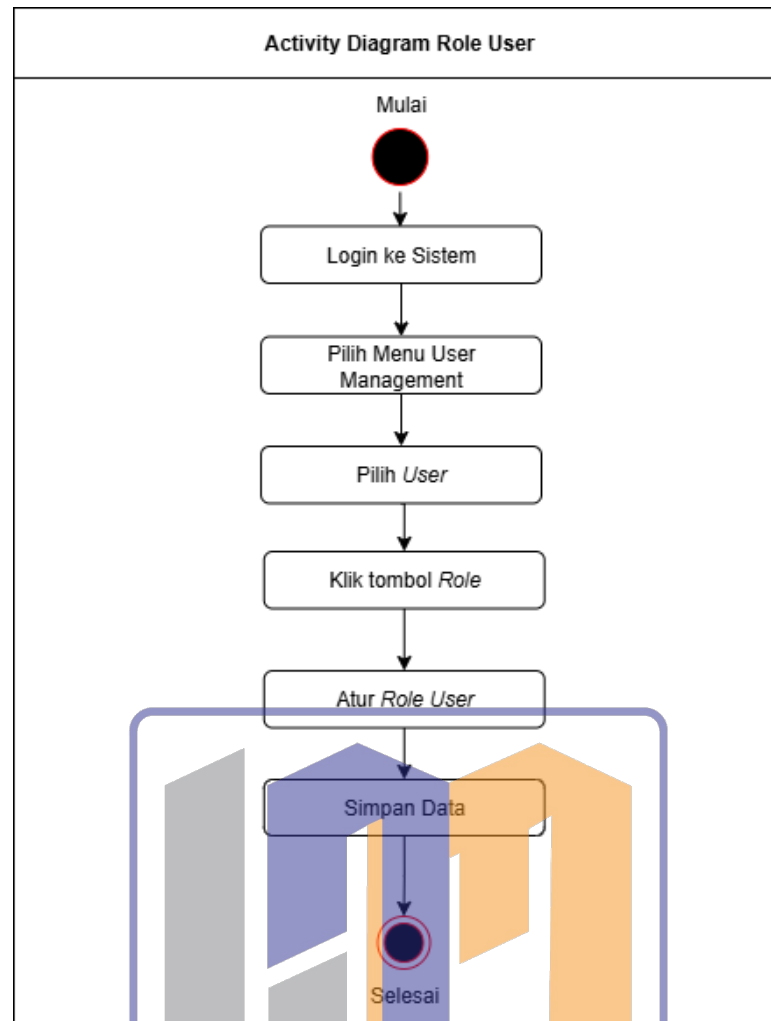


Gambar IV. 3 Activity Diagram User Management

Gambar IV.3 Bahwa dalam Activity Diagram tersebut aktivitas dalam mengelola data pengguna sistem, termasuk menambahkan, mengedit, atau menghapus akun user.

3. Activity Diagram *Role User*

Berikut tampilan *Activity Role User*, guna menerapkan alur utama dari *use case Role User*.

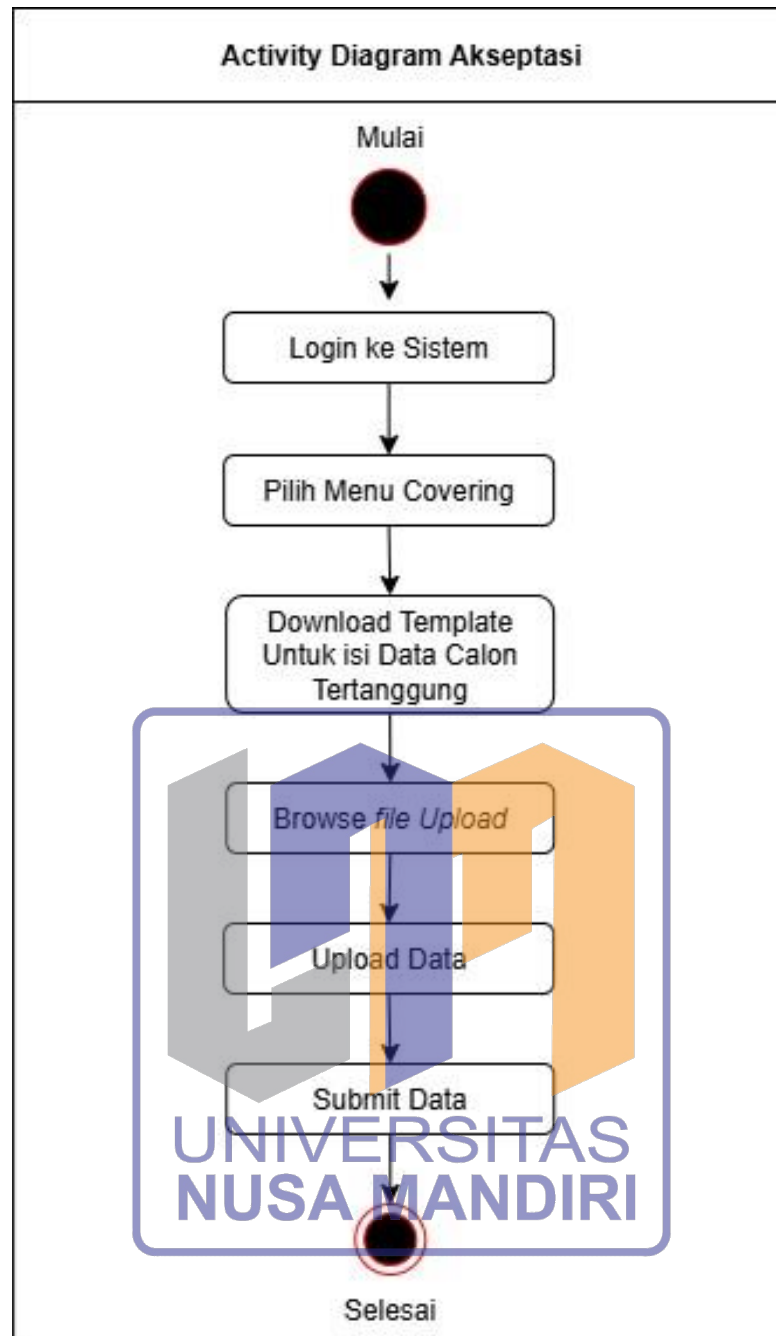


Gambar IV. 4 Activity Diagram Role User

Gambar IV.4 Bahwa dalam Activity Diagram tersebut aktivitas dalam mengatur peran (*role*) masing-masing user sesuai hak akses yang dibutuhkan.

4. Activity Diagram Akseptasi

Berikut tampilan *Activity* Akseptasi, guna menerapkan alur utama dari *use case* Akseptasi.

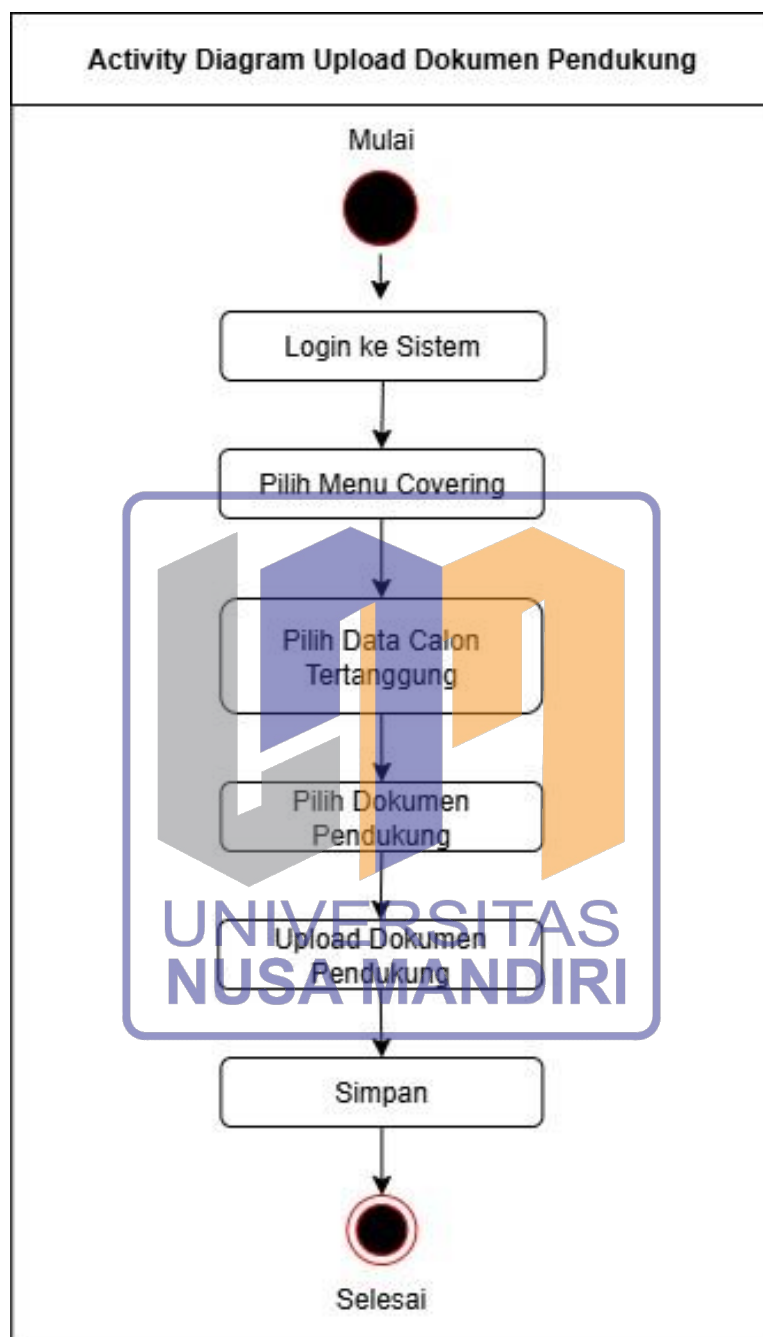


Gambar IV. 5 Activity Diagram Akseptasi

Gambar IV.5 Menggambarkan bahwa Admin Operasional melakukan proses akseptasi data calon tertanggung. Proses meliputi pengunduhan *template* data, memilih *file* yang akan di-*upload*, kemudian upload data calon tertanggung, serta submit data.

5. Activity Diagram Upload Dokumen Pendukung

Berikut tampilan *Activity* Upload Dokumen Pendukung, guna menerapkan alur dari *use case* Upload Dokumen Pendukung.



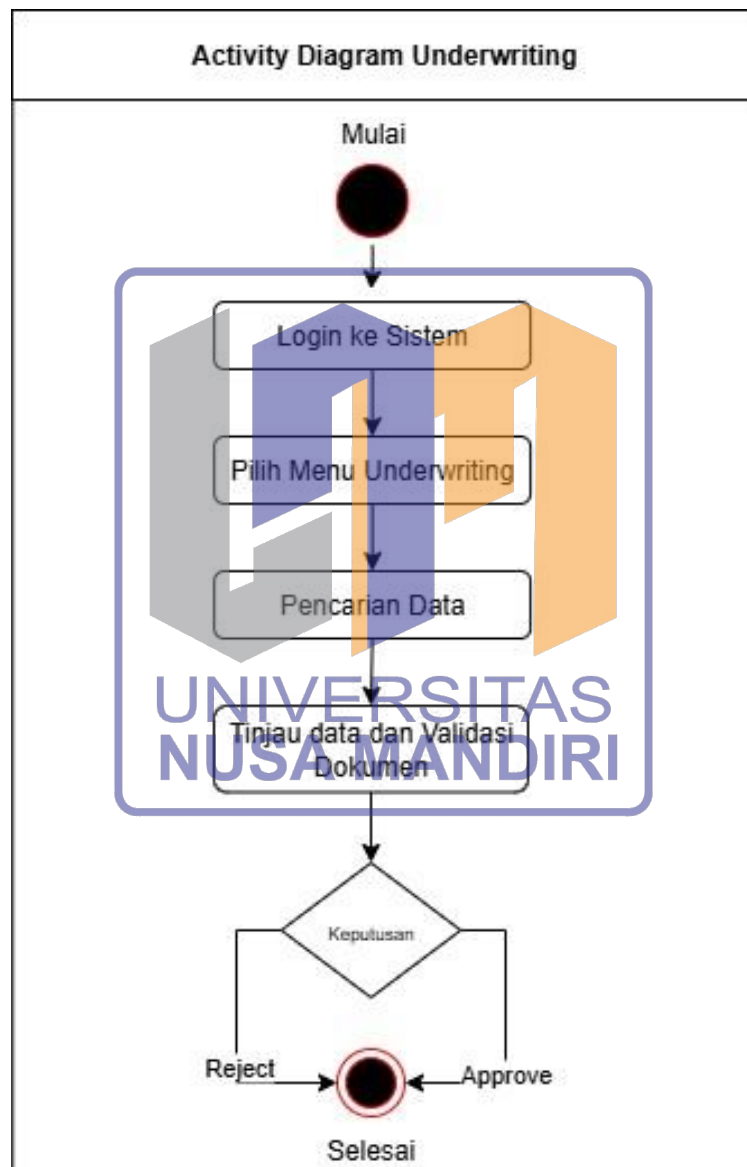
Gambar IV. 6 Activity Diagram Upload Dokumen Pendukung

Gambar IV.6 Menggambarkan bahwa Admin Operasional melakukan *upload* dokumen pendukung data calon tertanggung. Proses meliputi memilih data calon

tertanggung, kemudian memilih *file* dokumen pendukung yang akan di-*upload*, kemudian melakukan upload dokumen pendukung data calon tertanggung, kemudian simpan data dokumen pendukung.

6. Activity Diagram Underwriting

Berikut tampilan *Activity* Underwriting, guna menerapkan alur utama dari *use case* Underwriting.

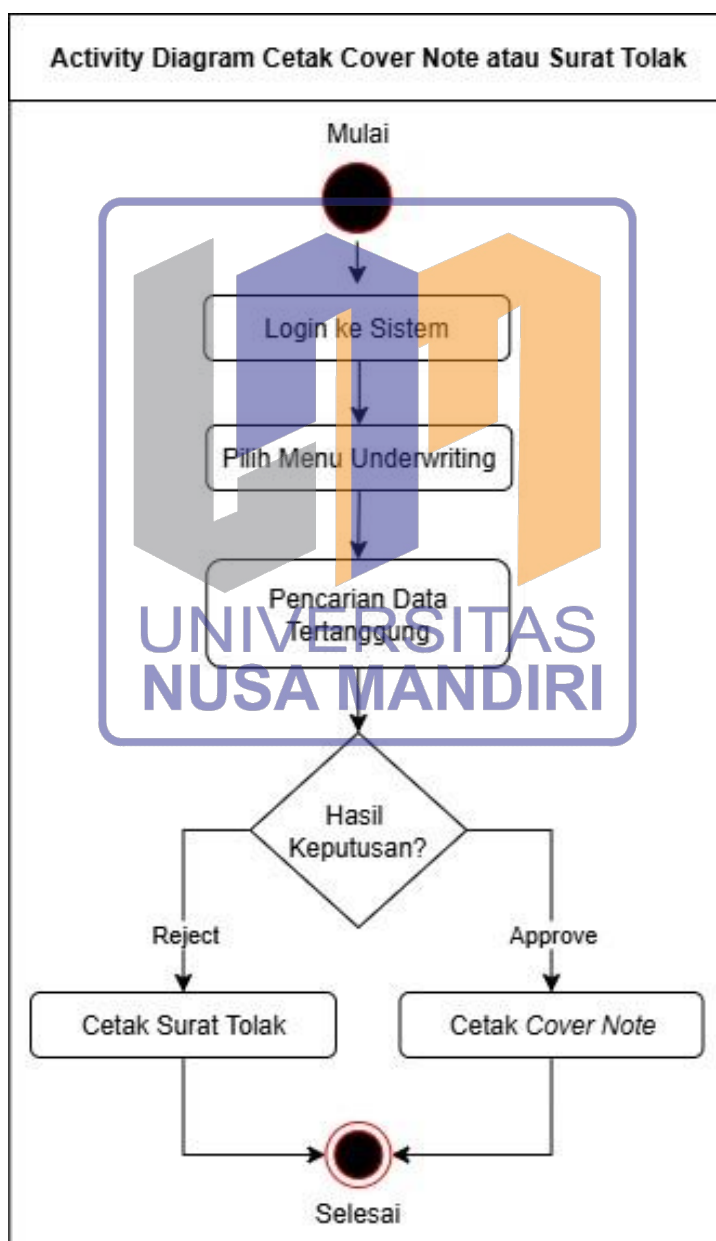


Gambar IV. 7 Activity Diagram Underwriting

Gambar IV.7 Menggambarkan bahwa SPV Operasional melakukan proses peninjauan data dan validasi dokumen pengajuan data calon tertanggung yang telah melewati proses akseptasi. SPV Operasional dapat menentukan apakah permohonan pengajuan disetujui atau ditolak.

7. Activity Diagram Cetak Cover Note atau Surat Tolak

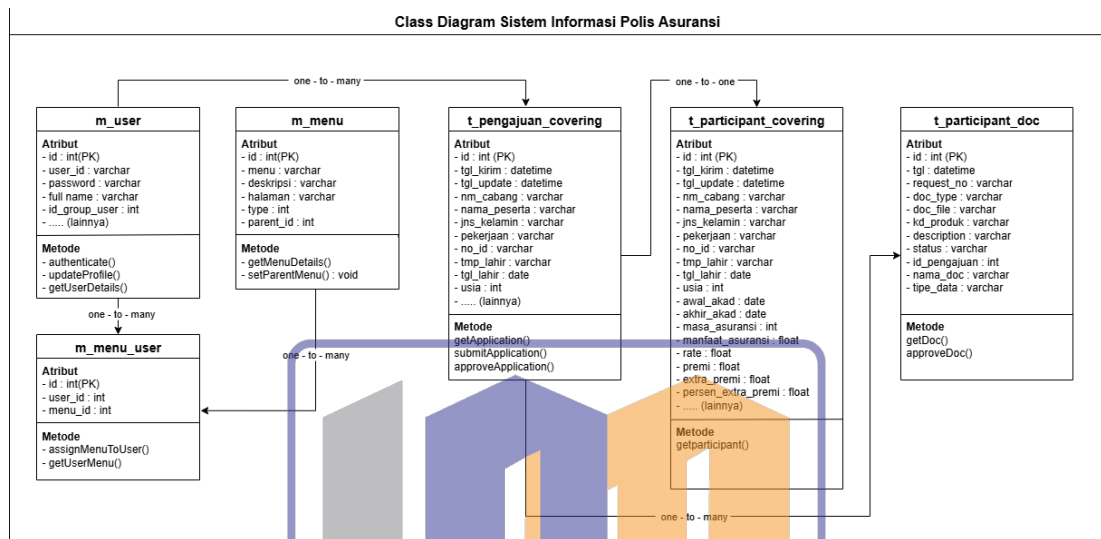
Berikut tampilan *Activity Cetak Cover Note atau Surat Tolak*, guna menerapkan alur utama dari *use case Cetak Cover Note atau Surat Tolak*.



Gambar IV. 8 Activity Diagram Cetak Cover Note atau Surat Tolak

Gambar IV.8 menggambarkan bahwa Admin Operasional melakukan proses cetak *Cover Note* untuk pengajuan yang sudah disetujui atau cetak Surat Tolak untuk pengajuan yang sudah dilakukan proses penolakan.

4.2.1.3 Class Diagram



Gambar IV. 9 Class Diagram Sistem Informasi Polis Asuransi

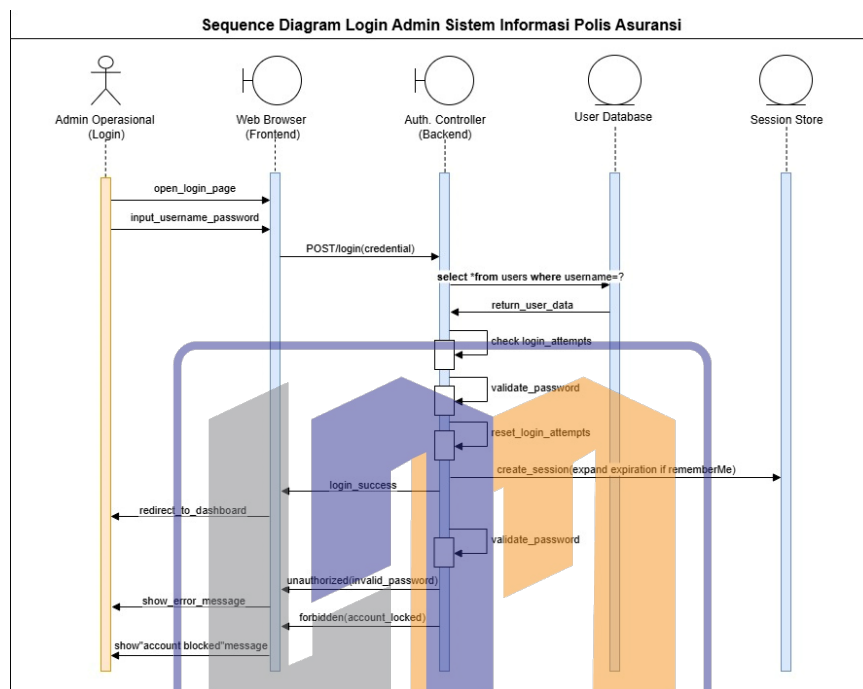
Gambar IV.9 Class Diagram ini menggambarkan struktur entitas utama dalam sistem serta hubungan antar kelasnya. Diagram ini mencakup 6 (Enam) kelas utama: *m_user*, *m_menu*, *m_menu_user*, *t_pengajuan_covering*, *t_participant_covering*, dan *t_participant_doc*.

1. Kelas *m_user* menyimpan data pengguna sistem seperti ID, *username*, dan informasi login, serta memiliki metode untuk autentikasi dan manajemen profil.
2. Kelas *m_menu* merepresentasikan daftar menu yang ada dalam aplikasi,
3. Kelas *m_menu_user* memiliki daftar menu yang dapat diakses user dalam aplikasi,
4. Kelas *t_pengajuan_covering* menangani proses pengajuan pertanggungan, memuat data cabang, peserta, dan informasi pengajuan lainnya.
5. Kelas *t_participant_doc* menangani data pengajuan dokumen pendukung.

6. Kelas *t_participant_covering* berisi informasi detail pertanggung jawaban yang sudah dicakup oleh polis, termasuk data premi dan manfaat asuransi.

4.2.1.4 Sequence Diagram

1. Sequence Diagram Login Admin Operasional

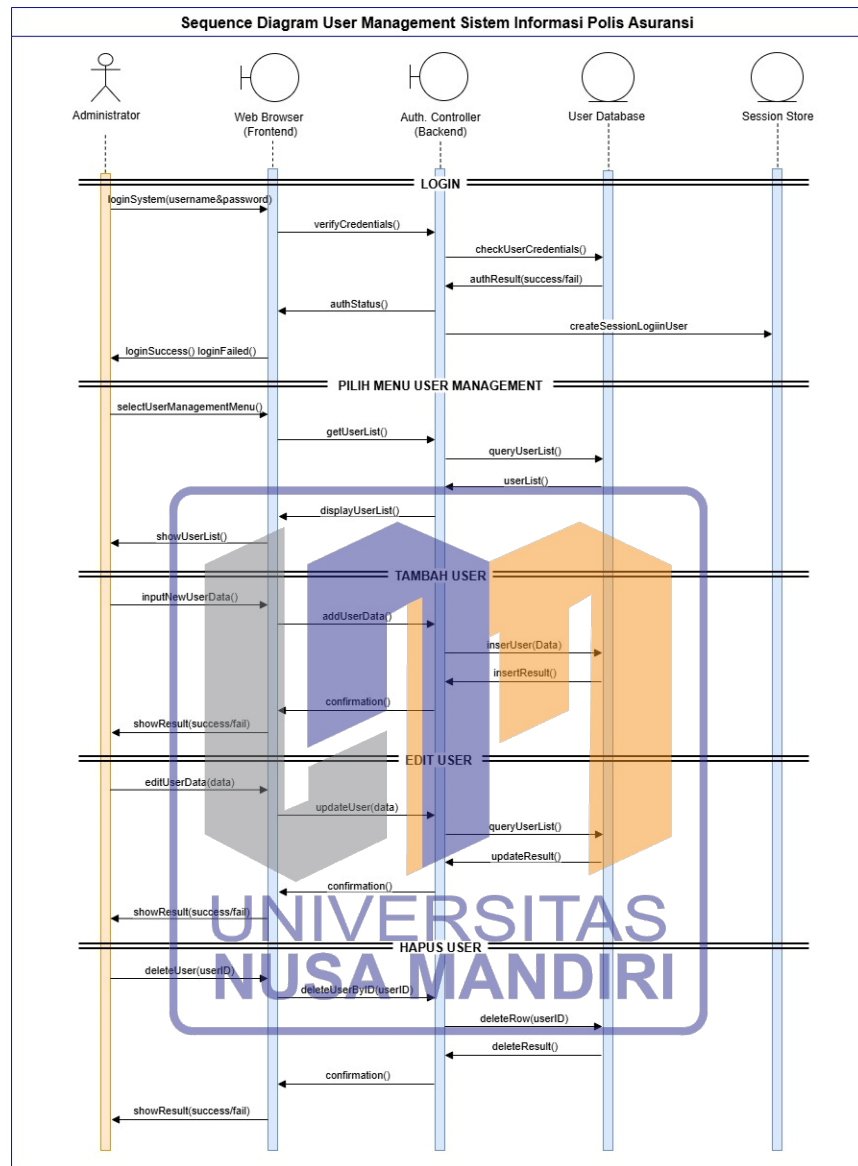


Gambar IV. 10 Sequence Diagram Login Operasional

Gambar IV.10 menggambarkan alur proses dengan dimulai saat Admin membuka halaman login melalui *web browser*, lalu memasukkan *username* dan *password*. Permintaan login dikirim ke *Authentication Controller* di *backend* melalui metode *POST(login)*, yang kemudian mengambil data pengguna dari *database* berdasarkan *username*. Setelah data dikembalikan, sistem akan memeriksa jumlah percobaan login (*check login_attempts*), memvalidasi *password*, dan jika berhasil, mereset jumlah percobaan dan membuat sesi login di *Session Store*. Jika login berhasil, pengguna akan diarahkan ke *dashboard* (*redirect_to_dashboard*). Namun, jika *password* salah, sistem akan menampilkan pesan kesalahan (*show_error_message*),

dan jika akun diblokir karena terlalu banyak percobaan gagal, akan muncul pesan *account blocked*.

2. Sequence Diagram User Management (Administrator)

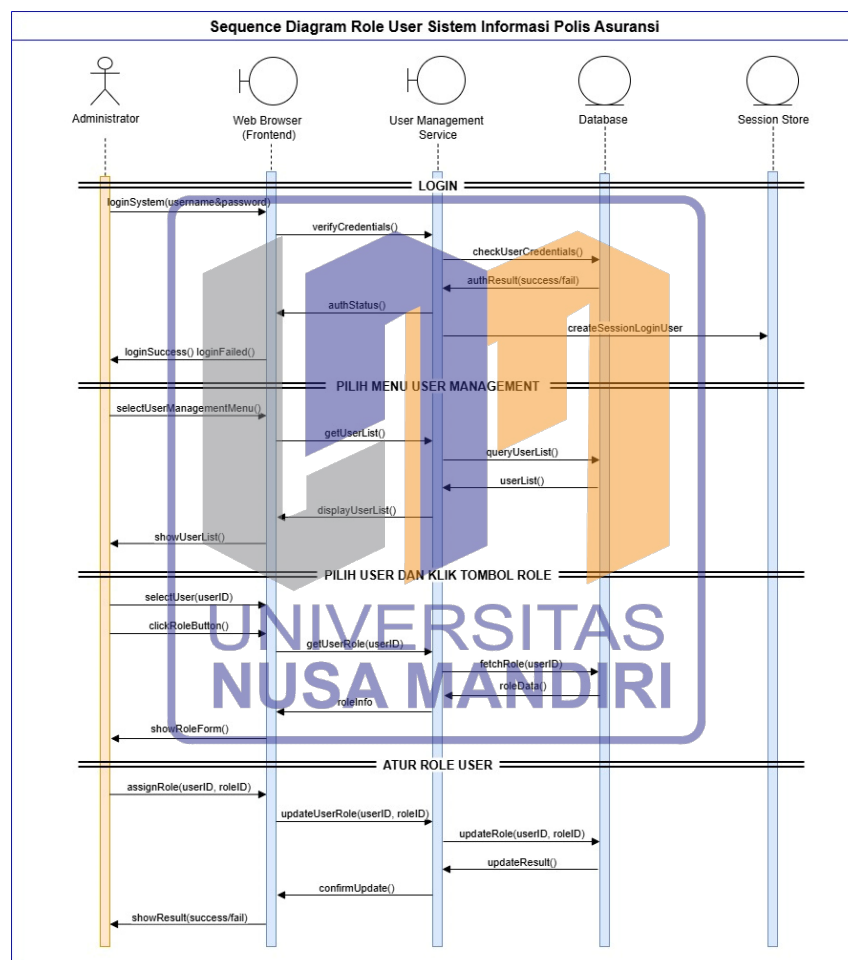


Gambar IV. 11 Sequence Diagram User Management (Administrator)

Gambar IV.11 menggambarkan dengan diawali dengan *login* menggunakan *username* dan *password*, yang kemudian diverifikasi oleh *Authentication Controller* dengan mencocokkannya ke basis data dan membuat sesi *login* jika berhasil. Setelah berhasil *login*, *Administrator* memilih menu *User Management* dan sistem menampilkan daftar user berdasarkan data dari *database*. Untuk

menambah user, administrator menginput data baru yang akan dikirim ke *backend* untuk disimpan ke *database*, lalu mendapat konfirmasi hasil. Proses *edit user* dilakukan dengan memilih *data user* yang akan diubah, mengirimkan data baru ke *backend*, lalu diperbarui di *database* dan dikonfirmasi ke Administrator. Sedangkan untuk menghapus *user*, sistem akan menghapus baris data berdasarkan ID *user*, dan setelah proses berhasil atau gagal, hasilnya ditampilkan kembali ke Administrator.

3. Sequence Diagram Role User.

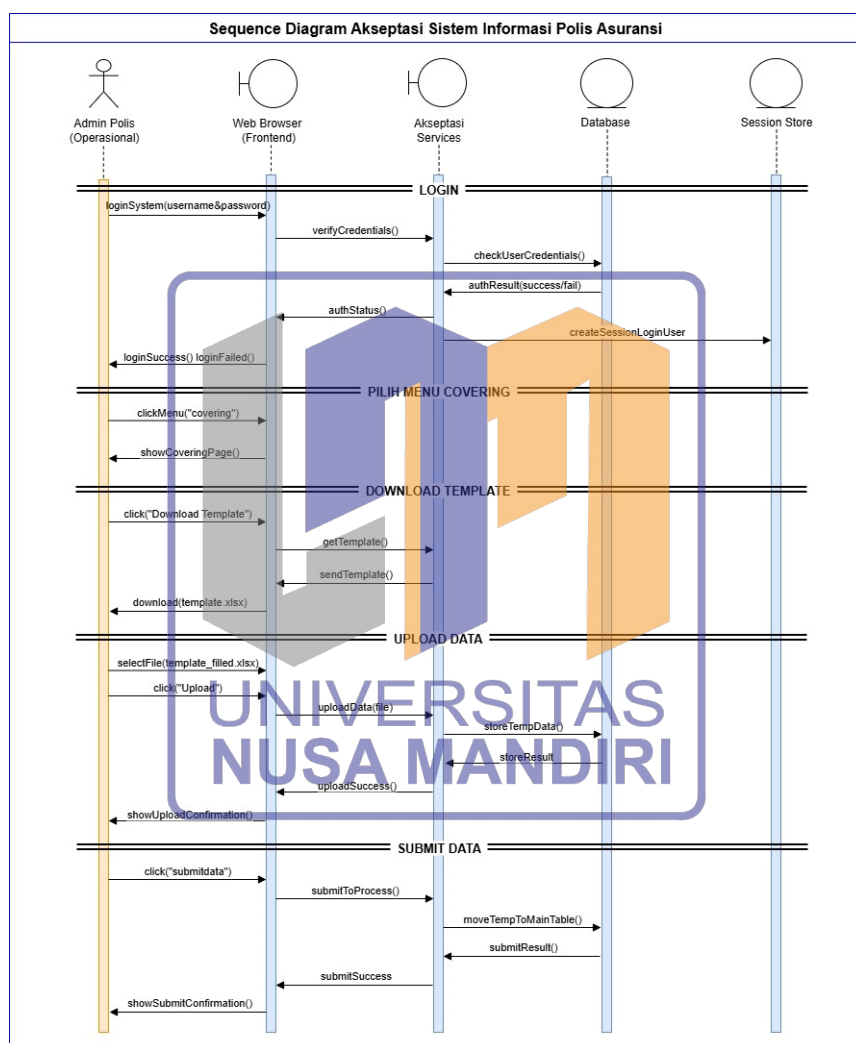


Gambar IV. 12 Sequence Diagram Role User

Gambar IV.12 menggambarkan proses yang dimulai ketika administrator melakukan *login*. Selanjutnya administrator memilih menu *User Management*, yang memicu permintaan daftar pengguna ke *backend*, lalu diambil dari database

dan ditampilkan di antarmuka. Setelah memilih *user* dan mengklik tombol *Role*, sistem akan meminta informasi *role* pengguna tersebut dan menampilkannya dalam *form* pengaturan *role*. Administrator kemudian menentukan peran yang sesuai dan mengirimkan permintaan untuk memperbarui data *role ke backend*, yang diteruskan ke database untuk di-*update*.

4. Sequence Diagram Akseptasi

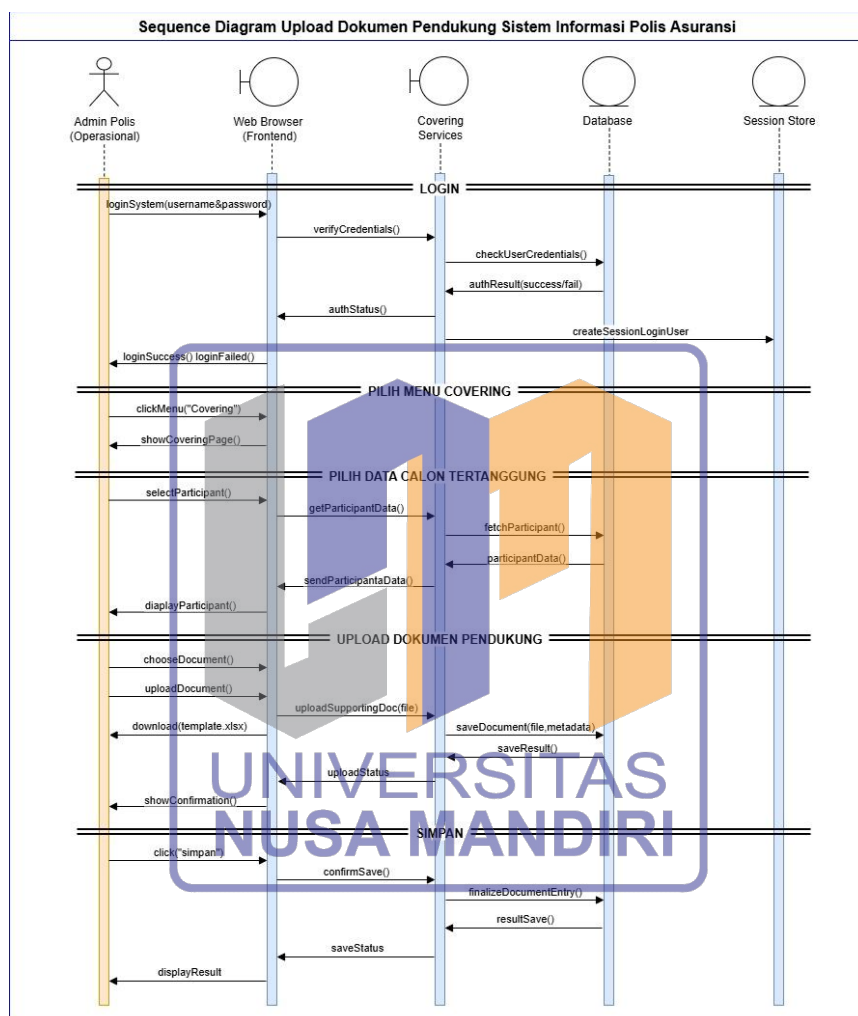


Gambar IV. 13 Sequence Diagram Akseptasi

Gambar IV.13 menggambarkan bahwa proses dimulai ketika Admin Polis *login* ke *system*. Selanjutnya, admin dapat mengunduh *template* pengisian data peserta dengan mengklik tombol *Download Template*, di mana sistem mengambil *file*

template dari layanan Akseptasi dan mengirimkannya ke *User*. Setelah itu, admin mengunggah *file* yang telah diisi melalui proses *upload* yang mengirimkan data ke sistem dan disimpan di *database*. Langkah terakhir adalah *submit* data, di mana *file* yang telah diunggah diproses dan dipindahkan dari tabel sementara ke tabel utama.

5. Sequence Diagram Upload Dokumen Pendukung

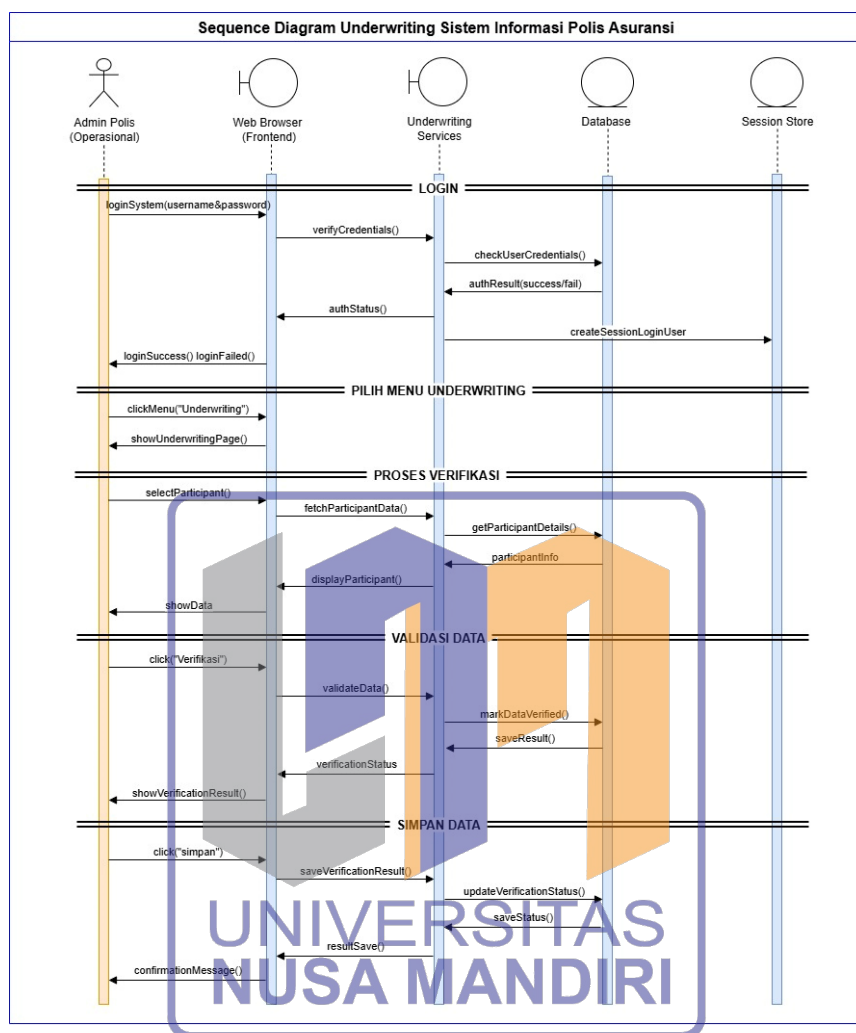


Gambar IV. 14 Sequence Diagram Upload Dokumen Pendukung

Gambar IV.14 menggambarkan alur proses *upload* dokumen pendukung pada Sistem Informasi Polis Asuransi, dimulai dari admin polis melakukan login melalui *web browser* yang kemudian diverifikasi oleh layanan *covering* dan *database*, dilanjutkan dengan pemilihan menu *covering* dan data peserta yang akan diasuransikan, lalu admin *upload* dokumen pendukung melalui *browser* ke layanan

covering yang menyimpannya ke database, hingga akhirnya proses diakhiri dengan penyimpanan data secara permanen dan konfirmasi keberhasilan kepada pengguna.

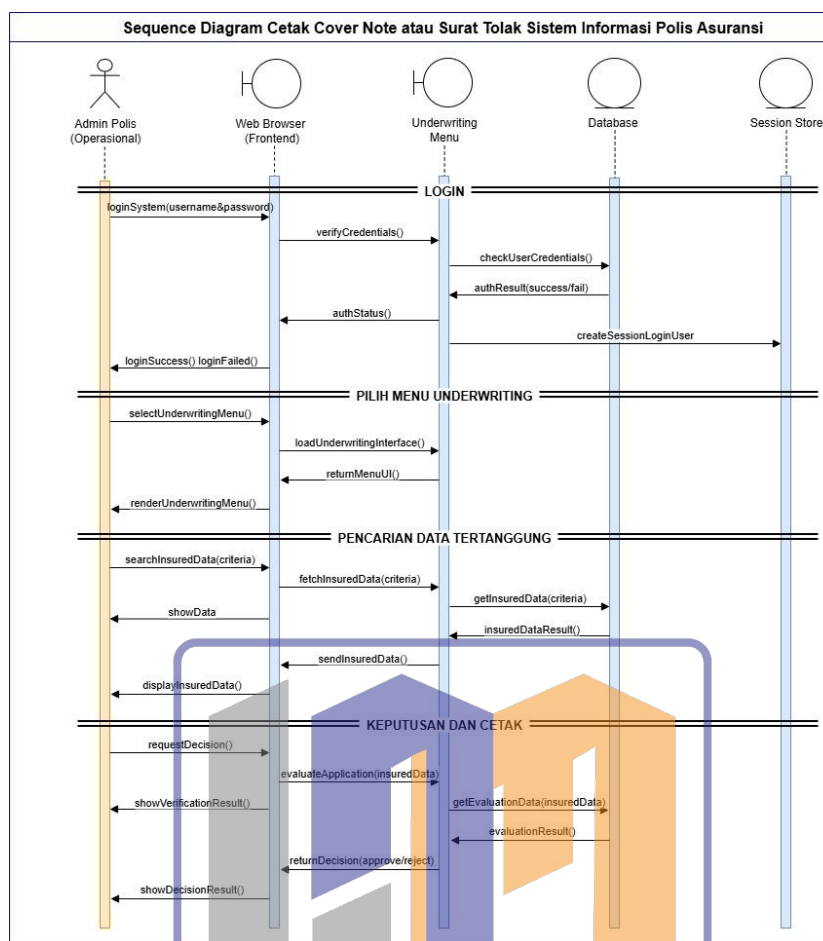
6. Sequence Diagram Underwriting



Gambar IV. 15 Sequence Diagram Underwriting

Gambar IV.15 menjelaskan bahwa Admin Polis mengirimkan permohonan *submit application* kemudian melakukan verifikasi data kemudian mengembalikan hasil verifikasi. Jika data valid, modul *Underwriting* akan menyimpan status validasi ke database. Jika data ditolak (*reject*), maka sistem akan mengirimkan *request correction* untuk memperbaiki data. Jika data valid, maka sistem akan melanjutkan dengan proses *submit* untuk mencetak dan menyimpan dokumen permohonan yang sudah disetujui.

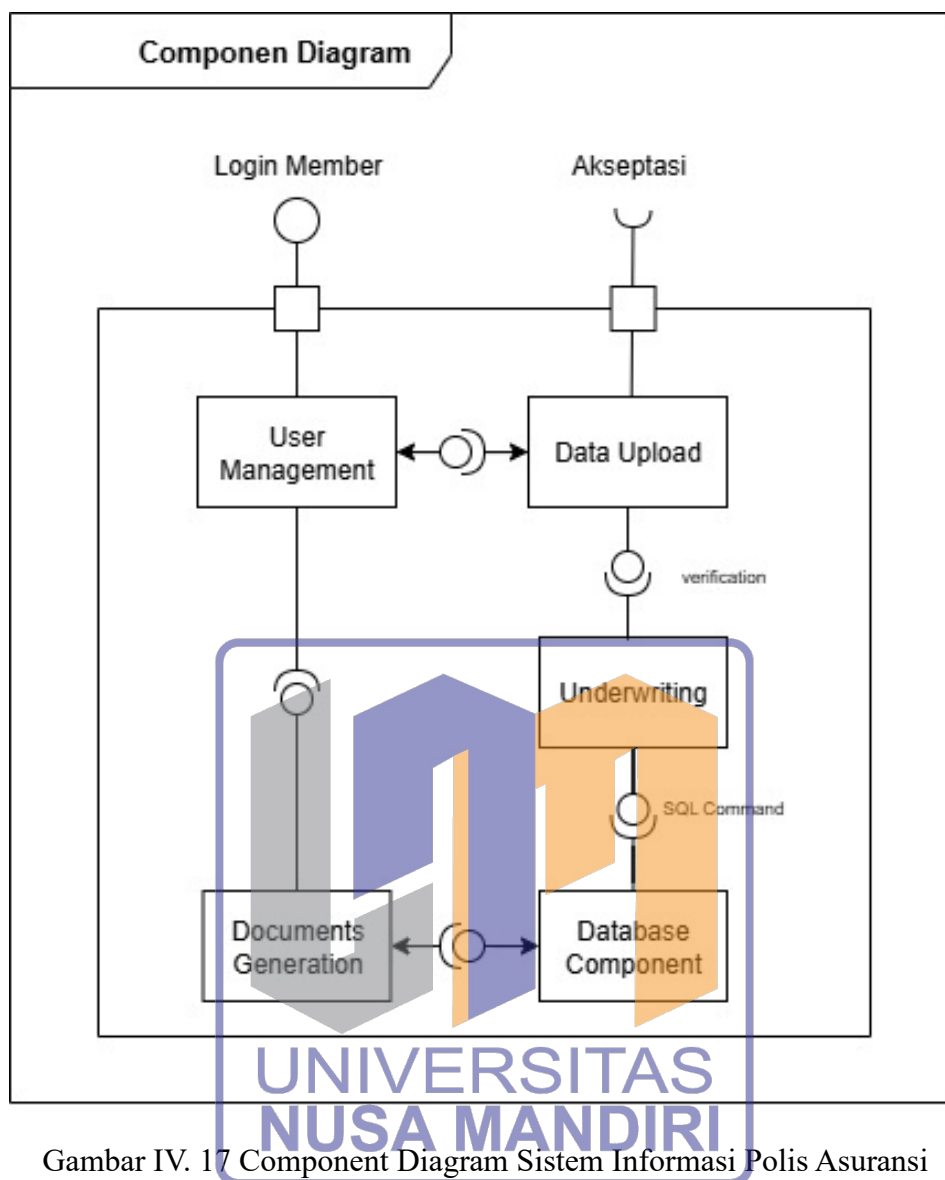
7. Sequence Diagram Cetak *Cover Note* atau Surat Tolak



Gambar IV. 16 Sequence Diagram Cetak Cover Note atau Surat Tolak

Gambar IV.16 menggambarkan setelah admin polis melakukan *login* melalui web browser yang diverifikasi oleh sistem *underwriting* dan *database*, kemudian admin memilih menu *underwriting* dan melakukan pencarian data tertanggung berdasarkan kriteria tertentu, setelah data ditemukan dan ditampilkan, sistem melakukan evaluasi untuk mengambil keputusan apakah data disetujui atau ditolak, dan berdasarkan keputusan tersebut, sistem menampilkan hasil verifikasi dan mencetak *Cover Note* jika disetujui atau Surat Tolak jika ditolak.

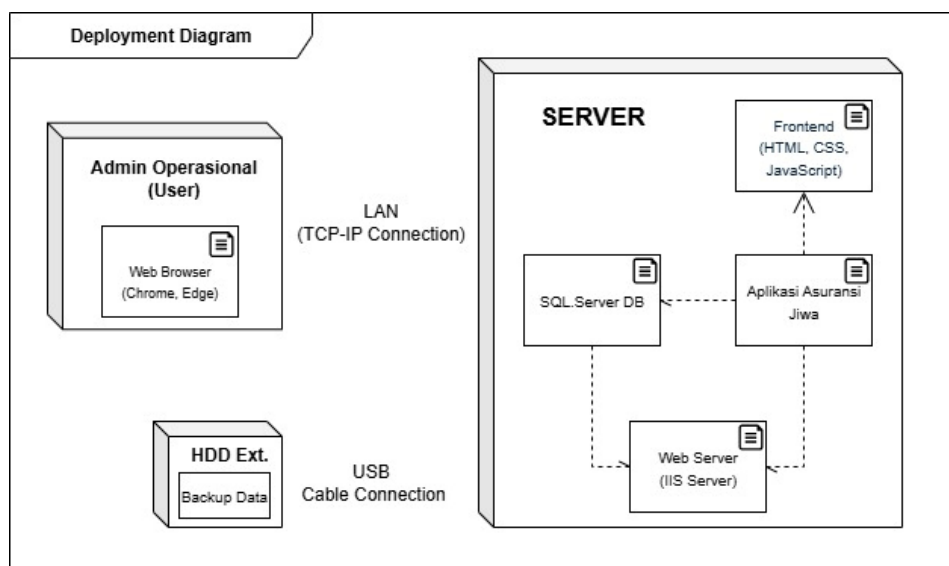
4.2.1.5 Component Diagram



Gambar IV. 17 Component Diagram Sistem Informasi Polis Asuransi

Gambar IV.17 pada *Component Diagram* ini menggambarkan arsitektur aplikasi web yang melibatkan *Login Member* dan *Akseptasi* sebagai entitas eksternal. Bagian dari *User Management* akan mengelola data pengguna dan mengirimkannya ke *Data Upload*, yang kemudian diverifikasi sebelum diteruskan ke *Underwriting*. *Underwriting* memproses data, berinteraksi dengan *Database Component* melalui perintah SQL, dan mendukung *Documents Generation* untuk membuat dokumen berdasarkan data yang disimpan.

4.2.1.6 Deployment Diagram



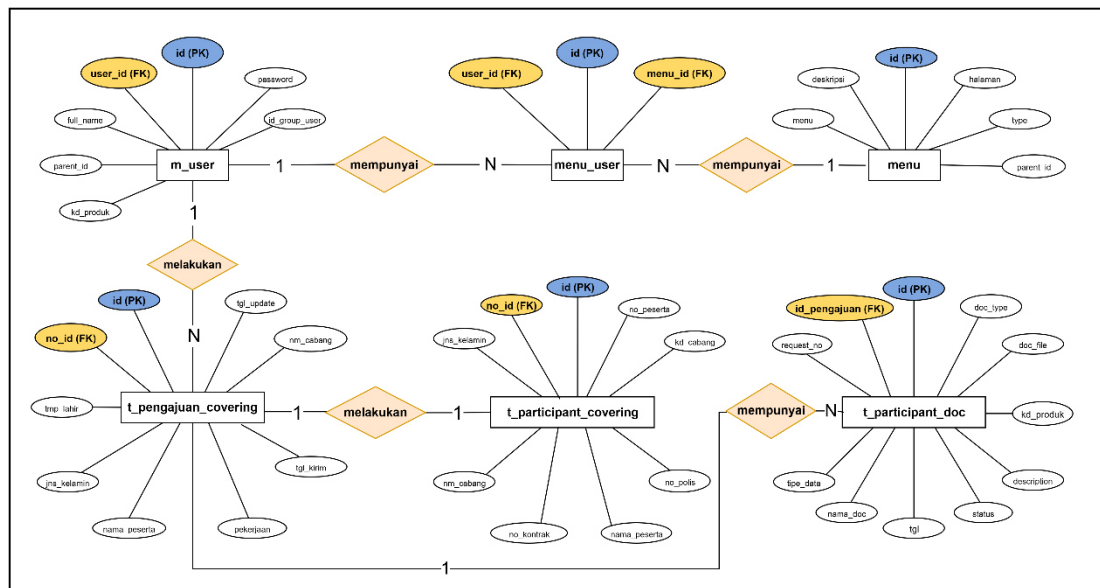
Gambar IV. 18 Deployment Diagram Sistem Informasi Polis Asuransi

Gamnbar IV.18 pada Deployment diagram tersebut menggambarkan arsitektur sistem aplikasi asuransi jiwa yang diakses oleh admin operasional melalui *web browser* (Chrome atau Edge) menggunakan koneksi LAN (TCP-IP). Di sisi server, terdapat komponen aplikasi asuransi jiwa yang berjalan di atas *Web Server (IIS Server)* dan terhubung ke basis data *SQL Server*. Antarmuka pengguna disajikan melalui *frontend* berbasis HTML, CSS, dan *JavaScript*. Selain itu, terdapat media penyimpanan eksternal (HDD eksternal) yang digunakan untuk *backup* data melalui koneksi kabel USB ke server.

4.2.2 Desain Pemodelan Data

Desain pemodelan data adalah proses perencanaan dan pengembangan struktur data yang digunakan dalam sebuah sistem informasi atau aplikasi untuk menyimpan, mengelola, dan mengakses data secara efisien.

4.2.2.1 Pemodelan Data Entity Relationship Diagram (ERD)

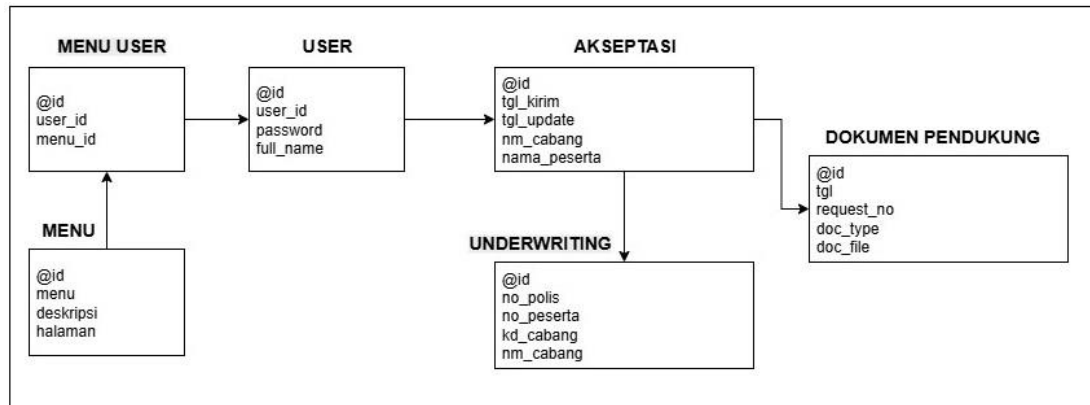


Gambar IV. 19 Entity Relationship Diagram Sistem Informasi Polis Asuransi

Gambar IV.19 menggambarkan hubungan relasi antar entitas, dimana hubungan *user* bisa memiliki banyak akses menu (satu menu juga bisa diakses oleh banyak *user*) dan satu *user* bisa melakukan proses banyak pengajuan, serta satu pengajuan bisa memiliki banyak dokumen.

4.2.2.2 Logical Record Structure

Logical Record Structure (LRS) adalah representasi logis dari struktur data dalam suatu sistem informasi, khususnya menggambarkan organisasi data dalam file atau database. LRS menunjukkan bagaimana data disusun, diakses, dan disimpan secara logis, walaupun tidak menggambarkan rincian teknis fisik seperti penyimpanan di disk.



Gambar IV. 20 Logical Record Structure Sistem Informasi Polis Asuransi

Pada gambar IV.20 adalah transformasi atau perubahan dari bentuk abstrak kedalam bentuk struktur logika *record*.

4.2.2.3 Spesifikasi File Database

Tujuan Spesifikasi *File Database* adalah menjadi panduan teknis untuk *developer* saat membuat tabel database, memastikan struktur data sesuai dengan kebutuhan sistem, menjamin integritas data (data akurat dan konsisten) dan memudahkan proses pemeliharaan dan pengembangan sistem di masa depan.

Berikut spesifikasi tabel-tabel pada struktur *Databasenya*:

1. Spesifikasi Tabel: *m_user*

- Nama Database : NSS (*Nasional Smart Solution*)
- Nama File : *dbo.m_user*

Tabel IV. 8 Spesifikasi File Tabel User

No	Nama Field	Type Data	Ukuran Field	Jenis Key
1	Id	int	-	Primary Key
2	user_id	varchar	20	
3	Password	varchar	20	
4	full name	varchar	30	
5	id_group user	int	-	
6	flag_active	char	1	
7	last_update	datetime	-	
8	last_opr	varchar	20	

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
9	Authkey	varchar	MAX	
10	nama field lainnya (18 field)	-	-	-

2. Spesifikasi Tabel: *m_menu*

a. Nama Database : NSS (*Nasional Smart Solution*)

b. Nama File : *dbo.m_menu*

Tabel IV. 9 Spesifikasi File Tabel Menu

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
1	Id	int	-	Primary Key
2	Menu	varchar	50	
3	Deskripsi	varchar	100	
4	Halaman	varchar	50	
5	Type	int	-	
6	parent id	int	-	

3. Spesifikasi Tabel: *m_menu_user*

a. Nama Database : NSS (*Nasional Smart Solution*)

b. Nama File : *dbo.m_menu_user*

Tabel IV. 10 Spesifikasi File Tabel Menu User

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
1	id	int	-	Primary Key
2	user_id	int	-	
3	menu_id	int	-	

4. Spesifikasi Tabel: *t_pengajuan_covering*

a. Nama Database : NSS (*Nasional Smart Solution*)

b. Nama File : *dbo.t_pengajuan_covering*

Tabel IV. 11 Spesifikasi File Tabel Pengajuan Covering

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
1	id	int	-	Primary Key
2	tgl_kirim	datetime	-	
3	tgl_update	datetime	-	

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
4	nm_cabang	varchar	30	
5	nama_peserta	varchar	30	
6	jns_kelamin	varchar	10	
7	pekerjaan	varchar	20	
8	no_id	varchar	20	
9	tmp_lahir	varchar	30	
10	nama field lainnya (44 field)	-	-	-

5. Spesifikasi Tabel: *t_participant_covering*

a. Nama Database : NSS (*Nasional Smart Solution*)

b. Nama File : *dbo.t_participant_covering*

Tabel IV. 12 Spesifikasi File Tabel Participant Covering

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
1	id	Int	-	Primary Key
2	no_polis	Varchar	20	
3	no_peserta	Numeric	(18, 0)	
4	kd_cabang	Nvarchar	(10)	
5	nm_cabang	Varchar	30	
6	no_id	Varchar	20	
7	no_kontrak	Varchar	20	
8	nama_peserta	Varchar	30	
9	jns_kelamin	Varchar	10	
10	nama field lainnya (74 field)	-	-	-

6. Spesifikasi Tabel : *t_participant_doc*

c. Nama Database : NSS (*Nasional Smart Solution*)

d. Nama File : *dbo.t_participant_doc*

Tabel IV. 13 Spesifikasi File Tabel Participant Document

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
1	Id	Int	-	Primary Key
2	Tgl	Datetime	-	
3	Request_no	Varchar	20	
4	Doc_type	Varchar	10	
5	Kd_produk	Varchar	15	
6	Description	Varchar	100	
7	Status	Varchar	2	

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key
8	Id_pengajuan	int	-	
9	Nama_doc	Varchar	30	
10	Tipe_data	Varchar	10	

4.2.3 Desain User Interface

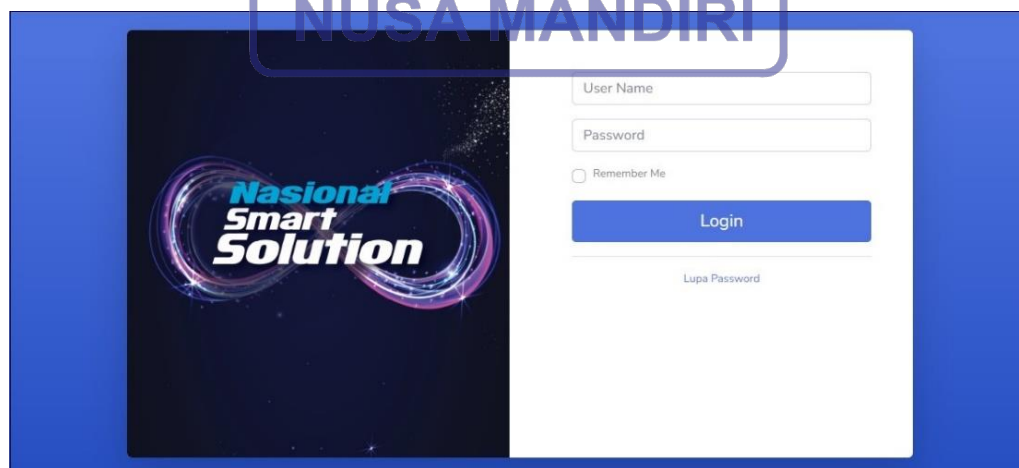
Desain User Interface (UI) adalah proses merancang tampilan visual dari aplikasi atau sistem yang akan digunakan oleh pengguna (*user*). Desain UI mencakup semua elemen yang terlihat dan interaktif, seperti tata letak (*layout*) halaman, tombol (ikon, menu, form input), warna, tipografi, gambar, navigasi antar halaman dan *layout* atau tombol lainnya.

Tujuan *Desain User Interface*:

1. Membuat tampilan aplikasi menarik, mudah digunakan, dan intuitif.
2. Memastikan pengguna mudah memahami fungsi-fungsi sistem.
3. Meningkatkan efisiensi dan kenyamanan interaksi pengguna dengan aplikasi.

4.2.3.1 Login

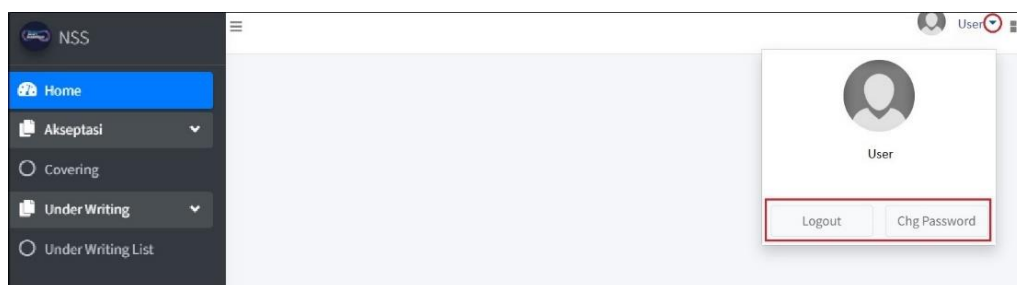
1. Desain Halaman Login



Gambar IV. 21 Desain Halaman Login Sistem Informasi Asuransi Jiwa

4.2.3.2 Menu Akseptasi

1. Dashboard Akseptasi

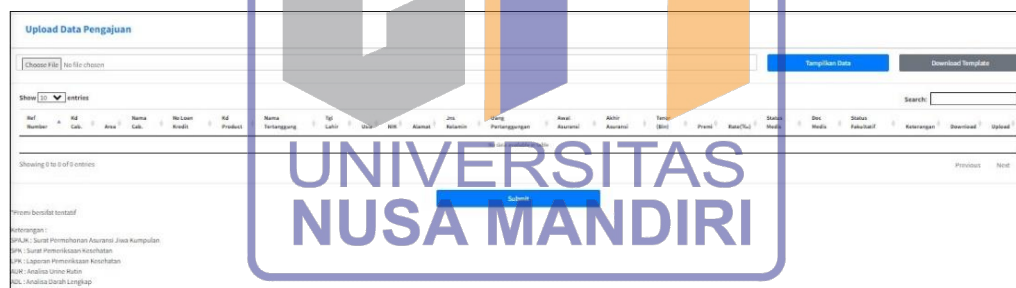


Gambar IV. 22 Dashboard Akseptasi Sistem Informasi Asuransi Jiwa

Penjelasan:

- Akseptasi, untuk proses pengajuan peserta asuransi jiwa.
- Chg Password*, pada tampilan *dashboard* digunakan untuk mengganti password user.
- Logout*, digunakan untuk keluar dari Aplikasi.

2. Akseptasi Covering



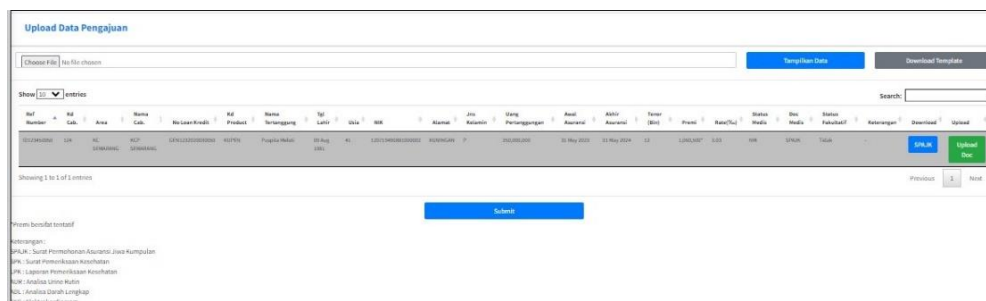
Gambar IV. 23 Tampilan Sub Menu Covering

Penjelasan:

- Submenu *covering* digunakan untuk pengajuan data peserta yang akan di-*cover* oleh pihak asuransi.
- Tombol *Choose File* digunakan untuk *upload file* data peserta.
- Tombol *Tampilkan Data* untuk menampilkan data peserta.

- d. Tombol *Download Template* untuk men-download template ter-update dengan hasil file download dengan format Excel.

3. Tampilan Data Peserta



Gambar IV. 24 Tampilan Data Peserta yang Sudah di-upload

Penjelasan:

- Tombol SPAJK untuk mendownload dokumen SPAJK dan dokumen lain yang disyaratkan.
- Tombol *Upload Doc* untuk mengupload dokumen yang telah Anda siapkan sebelumnya
- Tombol *Submit* untuk simpan data pengajuan peserta, kemudian akan muncul pesan *Submit Success* yang menandakan bahwa proses submit data peserta berhasil.

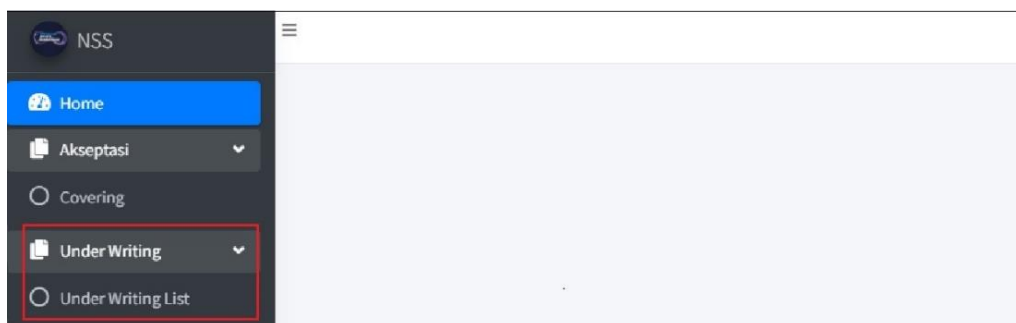
4. Tampilan Upload Document

Gambar IV. 25 Tampilan Upload Document Peserta

Klik *OK* untuk melakukan proses *upload* atau klik *Batal* untuk batal *upload*.

4.2.3.3 Menu Underwriting

1. Dashboard Underwriting

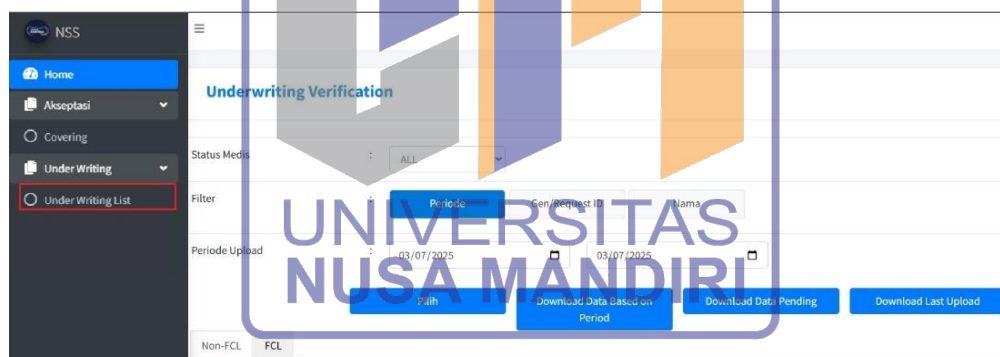


Gambar IV. 26 Dashboard Underwriting Sistem Informasi Asuransi Jiwa

Penjelasan:

- Menu *Underwriting*, yaitu proses mengidentifikasi dan menyeleksi risiko.
- Underwriting List*, Submenu *underwriting list* digunakan untuk menyeleksi risiko peserta yang mengajukan perlindungan asuransi.

2. Dashboard Underwriting Verification



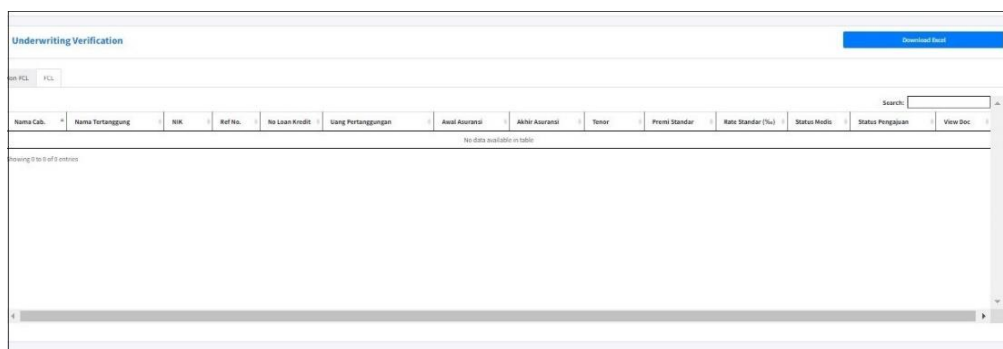
Gambar IV. 27 Dashboard Underwriting Verification

3. Underwriting List Peserta

Underwriting Verification										
Non-FCL		FCL								
Nama Cab.	Nama Tertanggung	NIK	Uang Pertanggungan	Awal Asuransi	Akhir Asuransi	Tenor	Premi Standar	Rate Standar	Status Medis	Status Pengajuan
KCP SEMARANG	TES 2	1.2345658569e+011	150,000,000	25 May 2023	25 May 2026	36	915,000	6.10	FCL	APPROVE
test	SOEMARNI	123456545648	50,000,000	10 Mar 2020	10 Mar 2030	120	1,399,500	27.99	FCL	APPROVE
test	ZAIFUDDIN	123456545645	55,774,800	10 Feb 2020	10 May 2022	27	417,753	7.49	FCL	APPROVE
test	AU SOBIRIN	123456545646	79,786,454	10 Mar 2022	10 Mar 2023	12	152,392	1.91	FCL	APPROVE
test	SOETOYO	123456545649	20,000,000	10 Mar 2018	10 Mar 2028	120	472,800	23.64	FCL	APPROVE
test	ALI HASANI	123456545647	79,786,454	10 Mar 2019	10 Mar 2021	24	305,582	3.83	FCL	CANCEL

Gambar IV. 28 Underwriting List Peserta

4. Download File Export



Gambar IV. 29 Download File Export Seluruh Peserta

Penjelasan:

- Dapat melihat peserta dengan status pengajuan *Approve* jika permohonan asuransi telah disetujui dan *Reject* jika permohonan asuransi terhadap peserta tersebut ditolak.
- Tombol *download excel* untuk mendownload *file export* seluruh peserta.

5. Underwriting Verification Document



Gambar IV. 30 Underwriting Verification Document

Penjelasan:

- Tombol *view* untuk melihat dokumen pendukung yang sudah diupload pada saat proses akseptasi (contoh SPAJK).

- b. Jika memiliki *feedback* terkait dokumen tersebut maka dapat menyampaikannya pada kolom *keterangan* dan mengklik tombol *proccess* untuk memproses komentar tersebut.
- c. Setelah mengklik tombol *proccess* maka akan muncul pesan *Process Success* yang menandakan bahwa proses pengiriman pesan berhasil. Jika muncul pesan *error*, maka bisa menghubungi *Administrator*.

4.3 Code Generation

Code generation pada skripsi adalah bagian yang menjelaskan bagaimana desain diimplementasikan ke dalam *source code* lengkap dengan *environment*, struktur *folder*, teknik otomatisasi, dan contoh potongan kode sehingga pembaca yakin bahwa sistem benar-benar direalisasikan sesuai rancangan.

4.3.1 Code Generation Login (*WebSignON*)

1. *Function* yang diperlukan `btnsubmit_Click`:

```
<asp:Button ID="btnsubmit" CssClass="btn btn-primary btn-block
btn-lg " runat="server" Text="Login" onclick="btnsubmit_Click" />

protected void btnsubmit_Click(object sender, EventArgs e)
{
    string userId = txt_user_id.Value.Trim();
    string password = txt_password.Value.Trim();

    // Contoh validasi hardcoded (sebaiknya gunakan database untuk
    validasi nyata)
    if (userId == "admin" && password == "admin123")
    {
        // Login sukses → redirect ke halaman utama
        Response.Redirect("Dashboard.aspx");
    }
    else
    {
        // Login gagal → tampilkan pesan error (bisa gunakan
        label atau alert)
        Response.Write("<script>alert('User ID atau Password
        salah!');</script>");
    }
}
```

2. Catatan : `txt_user_id.Value` dan `txt_password.Value` diambil dari inputan *form login* (karena `runat="server"`).

4.3.2 Code Generation Dashboard (*PageMaster*)

1. *Function* untuk menampilkan nama dan foto *user*

```
<span runat="server" id="lblusr">User</span>

```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        // Ambil data user dari session atau database
        if (Session["username"] != null)
        {
            lblusr.InnerText = Session["username"].ToString();
            lblusr2.InnerText = Session["username"].ToString();
            imgusr.Src = "Image/foto_user.png"; // atau
            berdasarkan nama file di DB
            imgusr2.Src = "Image/foto_user.png";
        }
    }
}
```

2. *Function* untuk *generate menu*

```
<% for (int i = 0; i < dtmenu.Rows.Count; i++) { ... } %>
```

```
protected DataTable dtmenu;
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        BindMenu();
    }
}
private void BindMenu()
{
    // Ambil data menu dari database sesuai hak akses user
    string userId = Session["userid"].ToString();
    dtmenu = GetMenuFromDatabase(userId); // method custom
}
```

3. *Function* *logout* dan *ganti password* (`logout.aspx.cs`)

```
<a href="logout.aspx" class="btn btn-default btn-lock">Logout</a>
<a href="ChPassword.aspx" class="btn btn-default btn-block">Chg
Password</a>
```

```
protected void Page_Load(object sender, EventArgs e)
{
    Session.Clear();
    Session.Abandon();
    Response.Redirect("WebSignOn.aspx");
}
```

Tabel IV. 13 Kesimpulan Function Yang Dibutuhkan Di Page.Master

No	Fungsi	Deskripsi
1	Page_Load	Menampilkan nama user, inialisasi menu
2	BindMenu()	Mengambil data menu sesuai user dari database
3	GetMenuFromDatabase(userId)	(opsional) method untuk query m_menu_user.

4.3.3 Code Generation Akseptasi (Covering)

- Berikut potongan function C# (code-behind) yang dibutuhkan untuk file CoveringNB.aspx.

```
#region ► FIELD PENYIMPAN DATA (SESSION / VIEWSTATE)
private const string KEY_DATATABLE = "tblCoveringUpload"; //
simpan di Session
#endregion

protected void Upload(object sender, EventArgs e)
{
    if (!fileUpload.HasFile) return;
    DataTable dt = ParseExcel(fileUpload.PostedFile);
    txtTotPremi.Text = HitungTotalPremi(dt).ToString("#,##0");
    Session[KEY_DATATABLE] = dt;
    tblview2 = dt;
    DataBind();
}

protected void DownloadTemplate(object sender, EventArgs e)
{
    string path =
Server.MapPath("~/Template/Template_Upload_Pengajuan.xlsx");
    FileInfo fi = new FileInfo(path);
    Response.Clear();
    Response.ContentType =
"application/vnd.openxmlformats-officedocument.spreadsheetml.sheet";
    Response.AppendHeader("Content-Disposition", $"attachment;
filename={fi.Name}");
    Response.WriteFile(fi.FullName);
    Response.End();
}
```

```

protected void Simpan(object sender, EventArgs e)
{
    if (Session[KEY_DATATABLE] == null) return;
    DataTable dt = (DataTable)Session[KEY_DATATABLE];
    using (var conn = new SqlConnection("<connection string>"))
    {
        conn.Open();
        foreach (DataRow r in dt.Rows)
        {
            if (r["Status Fakultatif"].ToString() != "FCL")
                continue;
            var cmd = new SqlCommand(@"INSERT INTO t_pengajuan_covering
(ref_number,kd_cabang,area,nm_cabang,no_loan,kd_produk,nama_peserta,
tgl_lahir,usia,no_id,alamat,jns_kelamin,uang_pertanggungan,awal_asuransi,akhir_asuransi,tenor,premi,rate,medical_status,doc_medis,
status_fakultatif,keterangan)
VALUES
(@ref,@kd,@area,@ncab,@loan,@prod,@nama,@tgl,@usia,@nik,@alm,
@jk,@up,@awal,@akhir,@tenor,@premi,@rate,@med,@doc,@fcl,@ket)",
conn);

            cmd.Parameters.AddWithValue("@ref", r[0]);
            cmd.Parameters.AddWithValue("@kd", r[1]);
            cmd.Parameters.AddWithValue("@area", r[2]);
            cmd.Parameters.AddWithValue("@ncab", r[3]);
            cmd.Parameters.AddWithValue("@loan", r[5]);
            cmd.Parameters.AddWithValue("@prod", r[23]);
            cmd.Parameters.AddWithValue("@nama", r[7]);
            cmd.Parameters.AddWithValue("@tgl",
DateTime.Parse(r[8].ToString()));
            cmd.Parameters.AddWithValue("@usia", r[9]);
            cmd.Parameters.AddWithValue("@nik", r[10]);
            cmd.Parameters.AddWithValue("@alm", r[11]);
            cmd.Parameters.AddWithValue("@jk", r[12]);
            cmd.Parameters.AddWithValue("@up",
decimal.Parse(r[13].ToString()));
            cmd.Parameters.AddWithValue("@awal",
DateTime.Parse(r[14].ToString()));
            cmd.Parameters.AddWithValue("@akhir",
DateTime.Parse(r[15].ToString()));
            cmd.Parameters.AddWithValue("@tenor", r[16]);
            cmd.Parameters.AddWithValue("@premi",
decimal.Parse(r[17].ToString()));
            cmd.Parameters.AddWithValue("@rate", r[18]);
            cmd.Parameters.AddWithValue("@med", r[19]);
            cmd.Parameters.AddWithValue("@doc", r[20]);
            cmd.Parameters.AddWithValue("@fcl", r[19]); //
            status fakultatif (FCL)
            cmd.Parameters.AddWithValue("@ket", r[22]);
            cmd.ExecuteNonQuery();
        }
    }
    Session.Remove(KEY_DATATABLE);
    ScriptManager.RegisterStartupScript(this, GetType(), "alert",
        "alert('Data berhasil disimpan');",
        location.href='CoveringNB.aspx';", true);
}
private DataTable ParseExcel(HttpPostedFile hpf)

```

```

{
    var dt = new DataTable();
    using (var pck = new ExcelPackage(hpf.InputStream))
    {
        ExcelWorksheet ws = pck.Workbook.Worksheets[0];
        bool firstRow = true;
        foreach (var cell in ws.Cells[1, 1, 1,
ws.Dimension.End.Column])
            dt.Columns.Add(cell.Text);
        for (int row = 2; row <= ws.Dimension.End.Row; row++)
        {
            var dr = dt.NewRow();
            for (int col = 1; col <= ws.Dimension.End.Column;
col++)
                dr[col - 1] = ws.Cells[row, col].Text;
            dt.Rows.Add(dr);
        }
    }
    return dt;
}
private decimal HitungTotalPremi(DataTable dt)
{
    decimal total = 0;
    foreach (DataRow r in dt.Rows)
        if (decimal.TryParse(r[17].ToString(), out decimal p))
            total += p;
    return total;
}

```

2. Tabel Ringkasan Fungsi *Covering* (Akseptasi)

Tabel IV. 14 Ringkasan Function *Covering*

No	Function	Tugas Utama
1	<i>Upload</i>	Membaca <i>file</i> Excel yang di- <i>upload</i> , mengubah ke <i>DataTable</i> , menampilkan <i>grid</i> , menghitung total premi.
2	<i>DownloadTemplate</i>	Mengirim <i>file template</i> Excel kosong ke <i>browser</i> .
3	Simpan	Menyimpan baris valid (FCL) dari <i>DataTable</i> ke tabel <i>t_pengajuan_covering</i> , lalu memberi notifikasi.
4	<i>ParseExcel</i>	Utilitas: konversi <i>HttpPostedFile</i> → <i>DataTable</i> (<i>library EPPlus</i>).
5	HitungTotalPremi	Menjumlahkan nilai premi dari kolom ke 17 <i>DataTable</i> .

4.3.4 Code Generation Underwriting (Underwriting List)

1. Berikut adalah fungsi *JavaScript* atau *jQuery* Underwriting List:

```
$(function () {

    $("#[id*=gridPesertaNFCL]").DataTable({
        bLengthChange: true,
        lengthMenu: [[5, 10, -1], [5, 10, "All"]],
        bFilter: true,
        bSort: true,
        bPaginate: false
    });

});

$(function () {
    $("#[id*=gridPesertaFcl]").DataTable({
        bLengthChange: true,
        lengthMenu: [[5, 10, -1], [5, 10, "All"]],
        bFilter: true,
        bSort: true,
        bPaginate: false
    });

});

$(document).ready(function () {
    $('#tblPeserta').dataTable();
});

$(document).ready(function () {
    $('#btnfcl').click(function () {
        $('#bttnfcl').removeClass("active");
        $('#btnfcl').addClass("active");
    });
    $('#bttnfcl').click(function () {
        $('#btnfcl').removeClass("active");
        $('#bttnfcl').addClass("active");
    });
    var gridmed = document.getElementById('<%= gridMed.ClientID
%>');
    if (gridmed != null) {
        if (gridmed.rows.length > 0) {
            for (i = 1; i < gridmed.rows.length; i++) {
                var idDoc =
gridmed.rows[i].cells[7].childNodes[1];
                var rowMedStat =
gridmed.rows[i].cells[2].childNodes[1];
                if (rowMedStat.innerHTML != "PENDING") {
                    gridmed.rows[i].cells[0].style.visibility =
"hidden";
                    gridmed.rows[i].cells[3].childNodes[1].disabled = "true";
                    gridmed.rows[i].querySelector("[id*=btnEditDocModal]").setAttribu
te("hidden", "hidden");
                }
                var rowmed =
gridmed.rows[i].cells[5].childNodes[1];
                gridmed.rows[i].cells[4].childNodes[1].href =
rowmed.innerHTML;
            }
        }
    }
});
```

```

    }
    var gridmed2 = document.getElementById('<%= gridMed2.ClientID
%>');
    if (gridmed2 != null) {
        if (gridmed2.rows.length > 0) {
            for (i = 1; i < gridmed2.rows.length; i++) {
                var rowmed =
gridmed2.rows[i].cells[2].childNodes[1];
                gridmed2.rows[i].cells[1].childNodes[1].href =
rowmed.innerHTML;
            }
        }
    }
    var gridpesertaNFCL =
document.querySelector("[id*=gridPesertaNFCL][class*=table]");
    var totPremiAll = 0;
    if (gridpesertaNFCL != null) {
        if (gridpesertaNFCL.rows[1].cells.length != 1) {
            for (i = 1; i < gridpesertaNFCL.rows.length - 1; i++)
{
                var rowpesertadoc =
gridpesertaNFCL.rows[i].querySelector("[id*=lbldoc_]");
                var rowpesertastatusmed =
gridpesertaNFCL.rows[i].querySelector("[id*=lblstdoc_]");
                var rowpesertastatus =
gridpesertaNFCL.rows[i].querySelector("[id*=lblstatus_]");
                var rowpesertamedis =
gridpesertaNFCL.rows[i].querySelector("[id*=lblstatusmedis_]");
                var id =
gridpesertaNFCL.rows[i].querySelector("[id*=lblidparticipant_]");
                if (rowpesertastatus.innerHTML == "14" ||
rowpesertastatus.innerHTML == "15" || rowpesertastatus.innerHTML
== "4") {
gridpesertaNFCL.rows[i].querySelector("[id*=txtrate_]").disabled
= "true";
                }
                if (rowpesertastatus.innerHTML == "14" ||
rowpesertastatus.innerHTML == "15") {
gridpesertaNFCL.rows[i].querySelector("[id*=cbPeserta_]").style.v
isibility = "hidden";
gridpesertaNFCL.rows[i].querySelector("[id*=txtrateext_]").disabl
ed = "true";
gridpesertaNFCL.rows[i].querySelector("[id*=txtket_]").disabled =
"true";
                }
                if (rowpesertastatus.innerHTML == "4") {
gridpesertaNFCL.rows[i].querySelector("[id*=txtrateext_]").disabl
ed = "true";
                }
                if (rowpesertastatusmed.innerHTML != "" ||
rowpesertadoc.innerHTML != "") {
gridpesertaNFCL.rows[i].querySelector("[id*=btnact_]").classList.
add("btn-danger");
                }
                if (rowpesertastatus.innerHTML == "13" ||
rowpesertastatus.innerHTML == "3"

```

2. Berikut daftar *function* (*code-behind*) yang dipanggil oleh halaman VerifUWNB.aspx

Tabel IV. 15 Function Pada Halaman Underwriting List

No	Function	Triggered By
1	ChangeData()	btn Pilih - OnClick="ChangeData"
2	DownloadBasedPeriodeFilter	btn Download Data Periode
3	DownloadDataPending()	btn Download Data Pending
4	DownloadLastUpload	btn Download Last Upload
5	GridViewNonFcl_RowCommand	GridView NFCL- OnRowCommand
6	SimpanPeserta	btn "Approve Peserta"
7	DeclinePesertaReject	btn "Tolak Peserta"
8	DeclinePesertaCancel	btn "Batal Peserta (NFCL)"
9	CancelPesertaFCL	btn "Batal Peserta (FCL)"
10	CancelPesertaNFCL	btn Cancellation NFCL (dalam markup yg di-comment)
11	SimpanMed	btn "Approve Doc"
12	DeclineMed	btn "Reject Doc"
13	UploadMedicalDoc	btn Upload dokumen medis (Repeater)
14	TambahDoc	btn "Tambah" (modal Tambah Dokumen Medis)
15	EditMedicalDoc	btn Edit Doc (modal Edit Dokumen)
16	PengajuanPembatalan	btn "Ajukan Pembatalan"
17	Simpan	btn Submit (file CoveringNB)

4.4 Testing

Tujuan dari pengujian (*testing*) pada aplikasi ini adalah untuk memastikan bahwa aplikasi bekerja sesuai kebutuhan pengguna, bebas dari kesalahan fungsional (*error* atau *bug*), memberikan hasil yang akurat dan konsisten dan mudah digunakan. Pada pengujian aplikasi pengelolaan polis asuransi ini menggunakan metode *Black Box Testing* dengan memprioritaskan pada fungsi, seperti aplikasi berjalan dengan baik di berbagai browser, fiturnya sesuai fungsi (*login*, *input*, *upload*, *download* ataupun

logout), dan kemudahan penggunaan aplikasi web dan uji kecepatan serta kestabilan saat banyak user mengakses.

4.4.1 Tahap Pengujian Aplikasi

1. Pengujian *Performance*

Pada aplikasi ini tidak dilakukan pengujian *performance* dengan menggunakan aplikasi atau *tools* pihak ketiga (*third party*), sehingga tidak mengetahui pengguna (*user*) yang terlibat atau akses, tidak bisa melihat waktu respons dari menit awal hingga menit terakhir pengujian serta tidak bisa membaca *file-file* yang *diupload* ataupun *download* (PDF, XLSX)

2. Pengujian Keamanan Website

Aplikasi yang dibuat ini tidak dipublikasikan di Internet, karena digunakan khusus untuk kebutuhan internal perusahaan dan hanya diakses oleh *user* internal perusahaan juga, maka aplikasi web tersebut tidak dipasang SSL dan tidak perlu juga diuji dengan menggunakan *tools* atau aplikasi keamanan website.

4.4.2 Tahap Pengujian Kenerimaan Sistem

Tahap Pengujian Penerimaan Sistem (*System Acceptance Testing*) adalah tahap akhir dalam proses pengujian aplikasi untuk memastikan bahwa sistem telah berfungsi sesuai kebutuhan pengguna dan siap digunakan secara operasional. Berikut tabel-tabel Pengujian Penerimaan Sistem atau UAT.

Tabel IV. 16 User Acceptance Testing (UAT)

Dokumen <i>User Acceptance Testing</i>			
Nama Proyek	:	Rancang Bangun Sistem Informasi Manajemen Polis Asuransi Jiwa Berbasis Web di PT Asuransi Jiwa Nasional	 NASIONAL <i>Life Insurance</i>
Studi Kasus / Mitra	:	PT Asuransi Jiwa Nasional	
Manajer Proyek	:	Lukman Hakim	

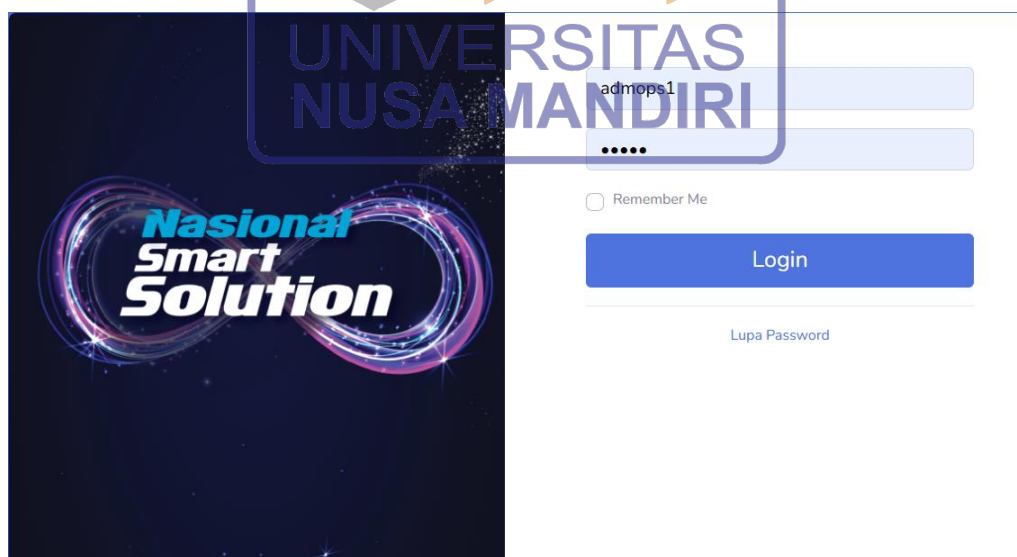
Proses Pengujian					
No	Use Case	Hasil Uji	Nama Penguji	Tanggal Pengujian	Catatan
1	<i>Use case Uji: Login</i> Deskripsi: Verifikasi terhadap pengguna Sistem Kasus Pengujian <i>Username</i> : admops1 <i>Password</i> : 12345 Saat pertama kali <i>login</i>, sistem akan menyarankan mengganti <i>Password</i> baru. Hasil yang diharapkan : - Jika <i>login</i> berhasil, maka akan masuk kedalam halaman <i>Dashboard Menu</i> - Jika <i>login</i> salah, maka sistem aplikasi akan muncul notifikasi (<i>kamu memiliki 2 X kesempatan</i>)	Berhasil	Abdurrahman	18 Juli 2025	
2	<i>Use case Uji: Akseptasi</i> Kasus Pengujian Melakukan input Data Pengajuan dengan cara Upload File dan menekan tombol submit. Format File <i>Upload</i> dalam bentuk <i>file</i> Excel (*.xlsx) Nama Tertanggung: Ronaldo Tgl Lahir: 4 May 1989 Uang Pertanggungan: 350.000.000 Hasil yang diharapkan: - Jika berhasil akan tampil pesan “submit success”	Berhasil	Abdurrahman	18 Juli 2025	

Proses Pengujian					
No	Use Case	Hasil Uji	Nama Penguji	Tanggal Pengujian	Catatan
	- Jika ada data kepesertaan yang sama, maka kolom keterangan akan keluar pesan <i>"Data sudah pernah di upload"</i> .				
3	<i>Use case Uji: Upload Dokumen Pendukung</i> Kasus Pengujian Melakukan <i>upload</i> dokumen pendukung SPAJK. Format file <i>upload</i> dalam bentuk File PDF (*.pdf) Nama File: spajk Ronaldo.pdf Hasil yang diharapkan: - Jika berhasil tampil pesan "submit success" - Jika gagal, keluar pesan "Save Failed"	Berhasil	Abdurrahman	18 Juli 2025	
4	<i>Use Case Uji: Underwriting</i> Kasus Pengujian Melakukan proses Keputusan pengajuan <i>Approve</i> Nama Tertanggung: Ronaldo Tgl Lahir: 4 May 1989 Uang Pertanggungan: 350.000.000 Hasil Keputusan: <i>Approve</i> Hasil yang diharapkan: - Jika berhasil tampil pesan "approve success" - Jika gagal tampil pesan "Save failed"	Berhasil	Lukman Hakim	18 Juli 2025	

Proses Pengujian					
No	Use Case	Hasil Uji	Nama Penguji	Tanggal Pengujian	Catatan
5	<i>Use case Uji: Cetak cover note</i> Kasus Pengujian Melakukan cetak <i>cover note</i> Nama Tertanggung: Ronaldo Tgl Lahir: 4 May 1989 Uang Pertanggungan: 350.000.000 Hasil Keputusan: <i>Approve</i> Hasil yang diharapkan: - Jika berhasil tampil <i>cover note</i> pada new tab browser - Jika gagal, <i>cover note</i> tidak tampil pada new tab browser	Berhasil	Abdurrahman	18 Juli 2025	

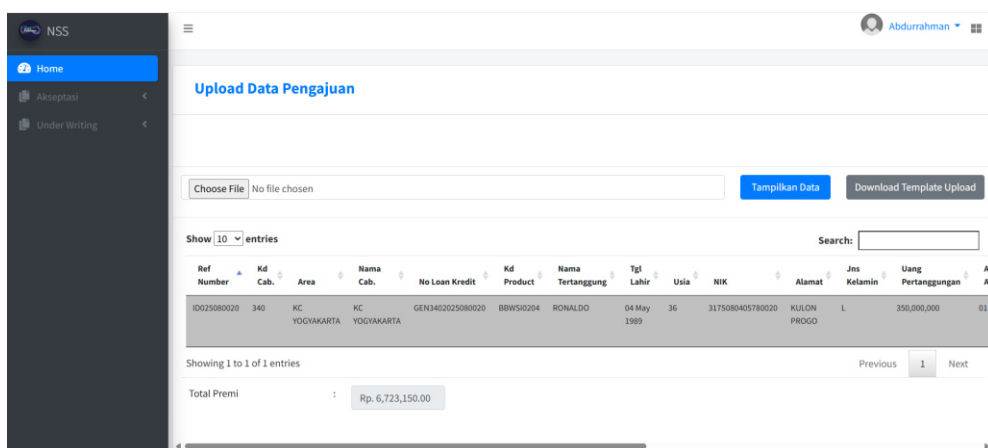
Berikut uji keterimaan dari tampilan sistem:

1. Use case Uji : *Login*



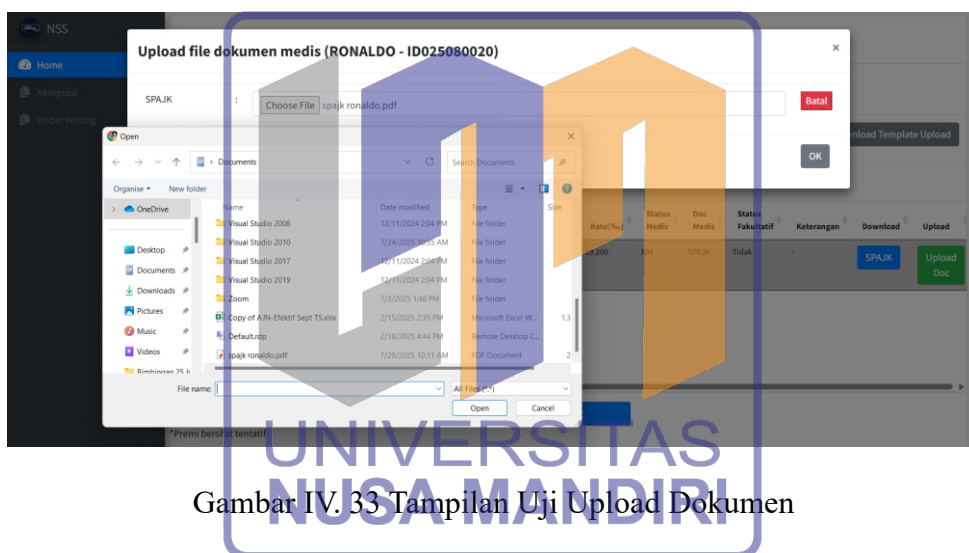
Gambar IV. 31 Tampilan Uji Login

2. Use case Uji: Akseptasi



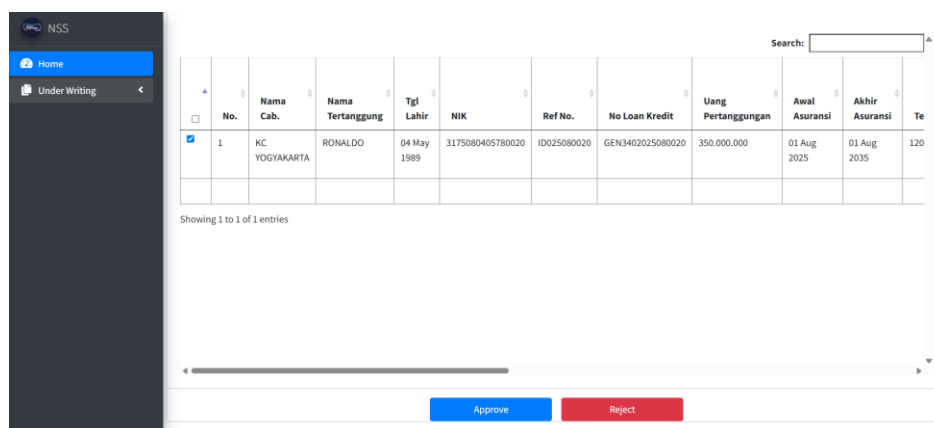
Gambar IV. 32 Tampilan Uji Akseptasi

3. Use case Uji: Upload Dokumen



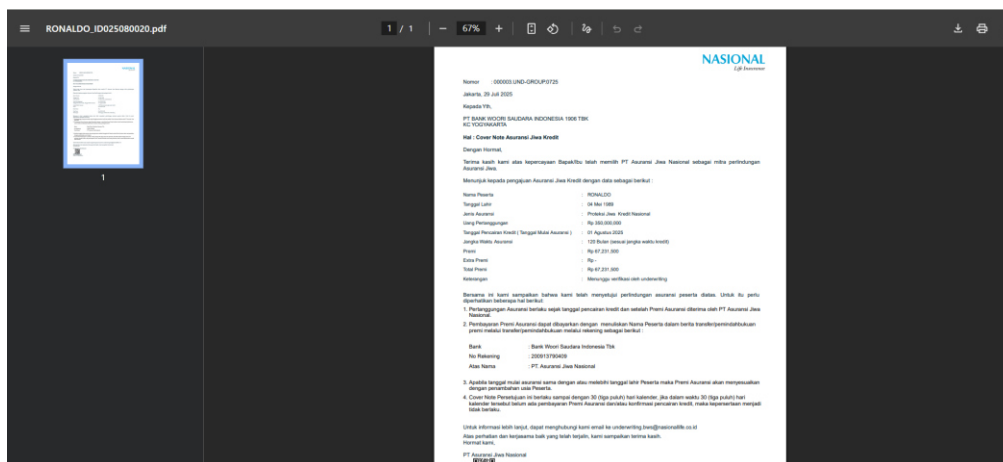
Gambar IV. 33 Tampilan Uji Upload Dokumen

4. Use case Uji: Underwriting



Gambar IV. 34 Tampilan Uji Underwriting

5. Use case Uji: Cetak *cover note*



Gambar IV. 35 Tampilan Uji Cetak *cover note*

4.5 Support

Berikut penjelasan lengkap tentang *support* dalam hal publikasi web, serta spesifikasi *hardware* dan *software* yang dibutuhkan, khususnya untuk aplikasi web internal (*intranet*)

4.5.1 Publikasi Web

Aplikasi sistem manajemen polis asuransi jiwa berbasis web ini tidak dipublikasikan (*publish*) di Internet, karena program atau aplikasi tersebut hanya digunakan untuk pengguna internal perusahaan (*intranet*) dan tidak dipublikasikan secara *online*:

1. Aplikasi hanya diperuntukkan bagi staf operasional atau admin tertentu dalam perusahaan.
2. Tidak ada kebutuhan bagi pihak luar (nasabah, publik) untuk mengakses sistem tersebut.
3. Tidak perlu membeli *SSL Certificate*, firewall tambahan, atau konfigurasi keamanan tingkat lanjut jika tidak dipublikasikan ke internet.

4. Tidak perlu menyewa server *cloud* atau *hosting online* karena bisa menggunakan server internal perusahaan.
5. Aplikasi *web* berbasis intranet atau lokal memberikan fleksibilitas dan efisiensi penggunaan seperti halnya *web online*, namun tetap menjaga keamanan, kontrol akses, dan kerahasiaan data internal perusahaan.

4.5.2 Spesifikasi Hardware dan Software

Berikut penjelasan lengkap mengenai Spesifikasi *Hardware* dan *Software* yang dibutuhkan untuk mendukung implementasi Rancang Bangun Sistem Informasi Manajemen Polis Asuransi Jiwa Berbasis Web, baik untuk server (*backend*) maupun pengguna (*client*).

4.5.2.1 Spesifikasi Perangkat Keras (Hardware)

Berikut adalah spesifikasi perangkat keras (*hardware*) yang digunakan:

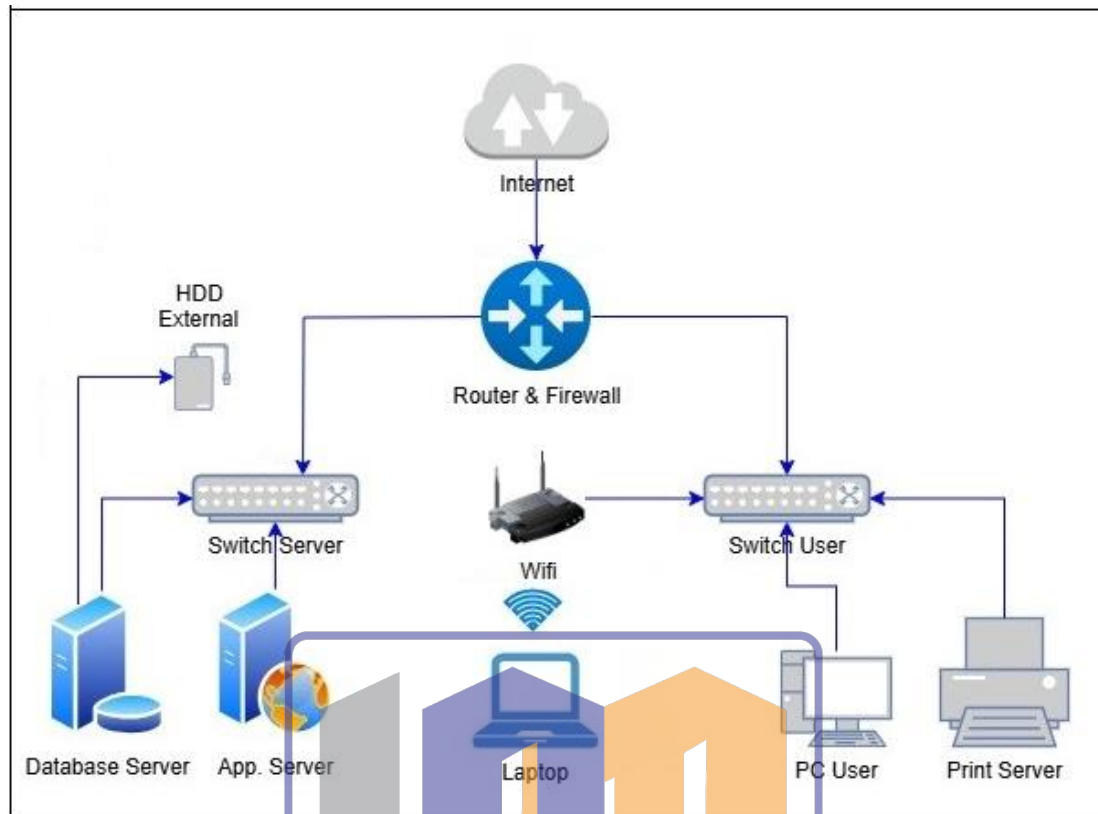
Tabel IV. 17 Spesifikasi Perangkat Keras (*Hardware*)

No	Perangkat Keras	Spesifikasi
1	Server	- Intel Xeon Processor (Icelake) (8 CPU's) – 2.1 GHz - RAM 16 GB - HDD 1 TB
2	PC User atau Laptop User	- Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz - RAM 16 GB - HDD 512 GB
3	Switch Local Area Network (LAN)	Switch Aruba 48 PORT – MANAGABLE
4	Wireless Router (<i>Wifi Access Point</i>)	Linksys EA 6900
5	Printer	Printer Fujixerox V C2263 (Print, Scan, Copy)
6	HDD External	External HDD 4 TB Seagate Barracuda
7	Router dan Firewall	Sangfor NGAF M5100

Berikut adalah fungsi dari perangkat keras (*hardware*) yang digunakan:

1. Server: Perangkat komputer atau sistem perangkat lunak yang menyediakan layanan (*Web Application*) dan data-data (*Database* atau *Data User*) kepada perangkat lain (PC atau Laptop User) yang terhubung ke server tersebut.
2. PC *User* atau Laptop: Penggunaan komputer Desktop atau Laptop Perusahaan untuk akses ke Aplikasi *web* yang ada di Server
3. Switch *Local Area Network* (LAN): Perangkat jaringan yang digunakan untuk menghubungkan beberapa perangkat, seperti komputer, printer, dan server, dalam satu jaringan lokal (LAN)
4. Wireless Router (*Wifi Access Point*): Perangkat yang berfungsi untuk mengirim atau menerima paket data antar perangkat (Server, PC dan ataupun Printer) secara nirkabel atau menggunakan media WiFi (tanpa kabel).
5. Printer: Perangkat untuk mencetak (*print*), menyalin (*copy*) atau memindai (*scan*) dokumen.
6. HDD External: HDD eksternal untuk penyimpanan atau *backup* data (*Database* atau *Data User*).
7. Router dan Firewall: Mengarahkan dan meneruskan data antar jaringan, serta menjaga keamanan dengan menyaring data yang masuk dan keluar.

Berikut Sistem Jaringan yang digunakan *Local Area Network (LAN)*:



Gambar IV. 32 Topologi Jaringan Lokal Aplikasi Web

Sistem Jaringan (*networking*) yang sedang berjalan menggunakan topologi yang sederhana, yang mana koneksi antar perangkat seperti PC User, Laptop, Server dan Print Server saling terhubung menggunakan perangkat *Switch User*, *Switch Server* atau Wifi, sehingga dengan jaringan yang terhubung tersebut, maka PC User ataupun Laptop dapat mengakses aplikasi web dan Database yang berada di kedua Server (*App. Server* dan *Database Server*).

4.5.2.2 Spesifikasi Perangkat Lunak (Software)

Ada beberapa perangkat lunak (*software*) yang akan digunakan untuk mendukung berbagai fungsi pengembangan (*development*) dan operasional.

Berikut adalah komponen utama perangkat lunak yang sering digunakan:

Tabel IV. 18 Sistem Perangkat Lunak (*Software*)

No	Perangkat Lunak	Spesifikasi
1	Microsoft Windows	Windows 10 Professional – 64 Bit
2	Microsoft Office	Office Home and Business 2019
3	Microsoft Windows Server	Windows Server 2016 Standar 64 Bit
4	Microsoft SQL Server	SQL Server 2008 R2 (10.0.1600)
5	Microsoft Visual Studio	Visual Studio 2010 Ver.10
6	Microsoft Visual Studio Code	Visual Studio Code 1.96 (User Setup)
7	Web Browser Google Chrome	Version 138 (Official Build) (64-bit)

Berikut adalah fungsi dari perangkat lunak (*software*) yang digunakan:

1. Microsoft Windows: Sistem Operasi untuk Admin atau *User*.
2. Microsoft Office: Aplikasi untuk membuka, mencatat, atau membuat data *Upload* dan laporan.
3. Microsoft Windows Server: Sistem Operasi *Server* untuk Aplikasi Web dan *Database*.
4. Microsoft SQL Server: Aplikasi Desktop untuk manajemen Database Server.
5. Microsoft Visual Studio: Aplikasi Desktop untuk *Develop* Aplikasi Web (basis C#)
6. Microsoft Visual Studio Code: Aplikasi *Code Editor* untuk membantu proses pengembangan sebuah aplikasi.
7. Web Browser Google Chrome: Aplikasi Client untuk Mengakses Aplikasi Web.

4.6 Spesifikasi Dokumen Sistem Usulan

Spesifikasi dokumen yang digunakan untuk pengajuan data peserta yang akan *discover* oleh pihak Asuransi adalah sebagai berikut :

1. Nama Dokumen : SPAJK
 - Fungsi : Dokumen pendukung pengajuan pertanggungan
 - Sumber : Klien
 - Tujuan : Admin Operasional

- Media : *Softcopy* PDF
- Frekuensi : Setiap terjadi pengajuan pertanggungan
- Format : Lampiran B.1
2. Nama Dokumen : Data Calon Tertanggung
- Fungsi : Untuk input data pertanggungan
- Sumber : Klien
- Tujuan : Admin Operasional
- Media : *File Excel*
- Frekuensi : Setiap terjadi pengajuan pertanggungan
- Format : Lampiran B.2
3. Nama Dokumen : *Cover Note*
- Fungsi : Sebagai luaran bukti pertanggungan
- Sumber : Admin Operasional
- Tujuan : Klien
- Media : *Softcopy* PDF (download melalui sistem)
- Frekuensi : Setiap terjadi persetujuan pertanggungan
- Format : Lampiran B.3
4. Nama Dokumen : Surat Tolak
- Fungsi : Sebagai luaran keputusan penolakan data tertanggung
- Sumber : Admin Operasional
- Tujuan : Klien
- Media : *Softcopy* PDF (download melalui sistem)
- Frekuensi : Setiap terjadi keputusan tolak pertanggungan asuransi
- Format : Lampiran B.4



- 5 Nama Dokumen : Laporan Data Pertanggungungan
- Fungsi : Sebagai luaran laporan rekap data pengajuan
- Sumber : Admin Operasional
- Tujuan : Klien
- Media : *file Excel* (download melalui sistem)
- Frekuensi : Setiap selesai proses pengajuan pertanggungungan
- Format : Lampitan B.5

