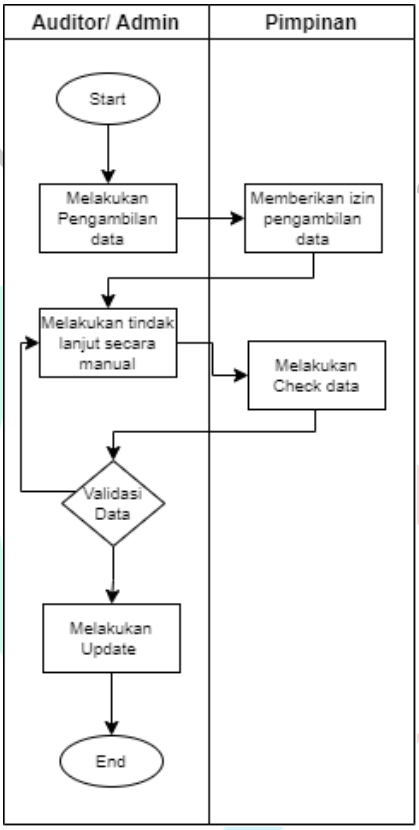
BAB IV
PERANCANGAN

4.1. Analisa Sistem Terdahulu



Gambar 4.1 Analisis Terdahulu

Pada gambar 4.1 merupakan sistem yang berjalan pada terdahulu. Auditor/admin melakukan proses tindak lanjut resiko secara manual dan birokrasi yang masih bersifat manual satu persatu.

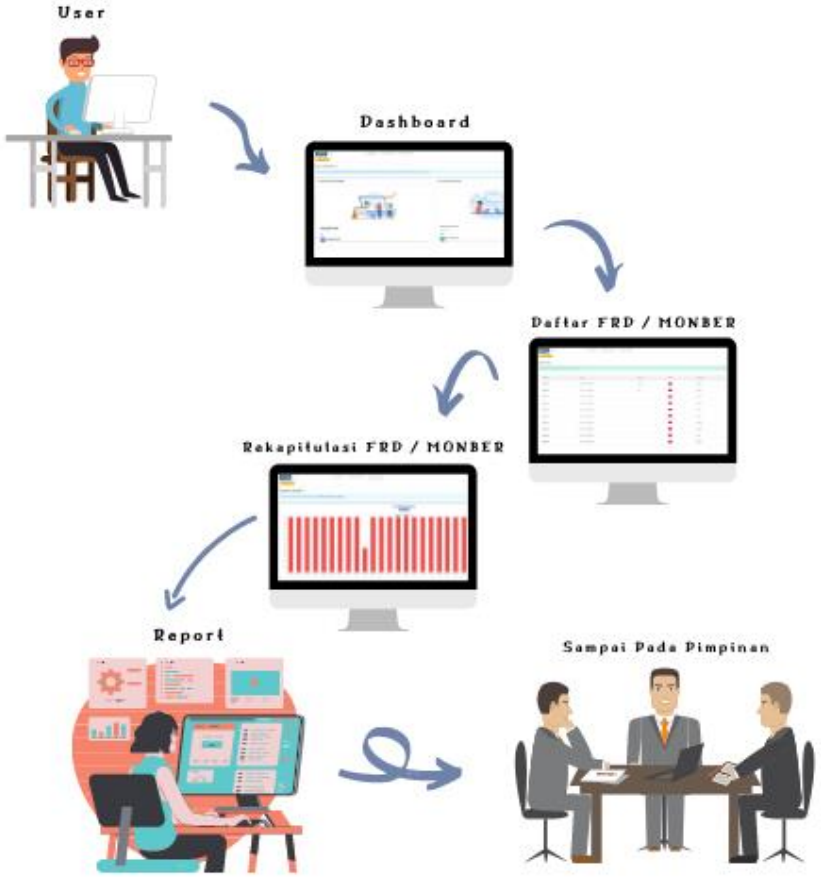
4.2. Spesifikasi yang akan dikembangkan

Rekayasa yang dibutuhkan pada tahap ini tentu akan mengenal lebih rinci rancangan aplikasi yang dibuat oleh peneliti, dan tujuan peneliti sendiri tentunya aplikasi dapat berjalan secara maksimal sebagaimana mestinya. Berdasarkan hasil penelitian yang dilakukan pada sistem terdahulu yang diperoleh yang dibutuhkan dalam pengembangan aplikasi antara lain :

1) Auditor / Admin dapat melihat hasil keseluruhan data yang ada di Audit Internal BRI.

2) Auditor /Admin dengan mudah melakukan tindak lanjut resiko dikarenakan terdapat algoritma forward chaining yang sudah di aplikasikan ke website ini.

4.2.1. Diagram Konseptual



Gambar 4.2.1 Diagram Konseptual

Gambar 4.2.1 merupakan diagram konseptual yang menjelaskan hubungan antara faktor-faktor yang memberikan informasi konsep tentang website ini. Pada aplikasi ini terdapat 3 menu utama yaitu ada dashboard, fraud risk detection dan monitoring berkala, disetiap menu utama terdapat sub bab menu. Sub bab menu dari dashboard dapat menampilkan presentase data FRD dan presentase data monber. Sub bab menu fraud risk detection terdapat sub bab menu daftar FRD dan rekapitulasi FRD. Sub bab menu monitoring berkala terdapat sub bab menu daftar monber dan rekapitulasi Monber. Setelah data direkapitulasi akan direport secara keseluruhan oleh user untuk diserakan kepada pimpinan.

4.2.2. Spesifikasi Kebutuhan Input ke Sistem

Pada tahap spesifikasi kebutuhan input ke sistem ini memiliki kebutuhan sebagai berikut:

1) Data user.

2) Data yang meliputi pengguna dan role pengguna.
3) Data FRD.
4) Data Monber.

4.2.3. Spesifikasi Kebutuhan Output ke Sistem

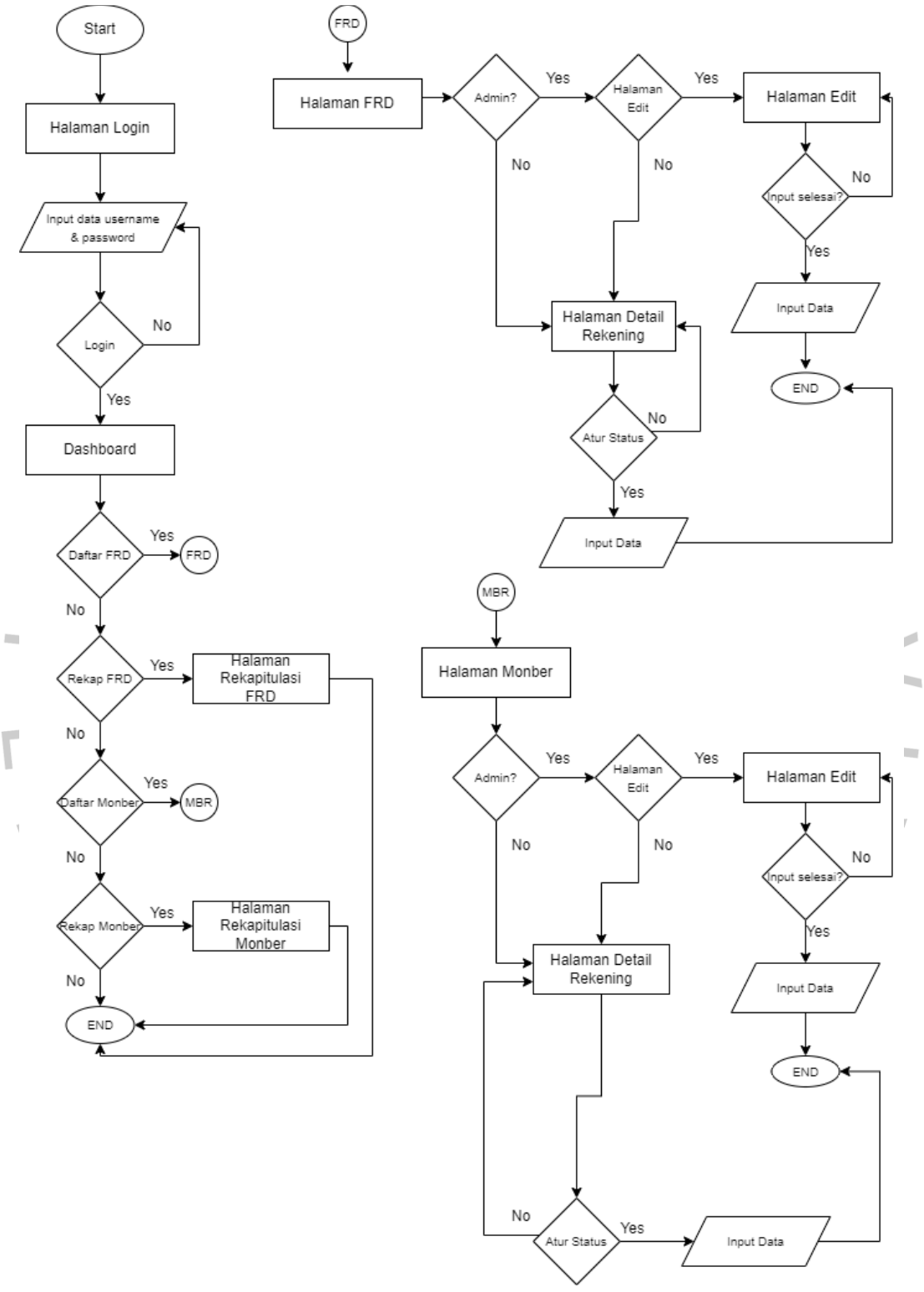
Pada tahap spesifikasi kebutuhan output ke sistem ini memiliki kebutuhan sebagai berikut :

1) Hasil dari pengguna dan role pengguna.
2) Hasil data yang ditindak lanjut.
3) Informasi jumlah data yang akan ditindak lanjut.
4) Informasi rekapitulasi data yang ditindak lanjut.

4.3. Perancangan Sistem

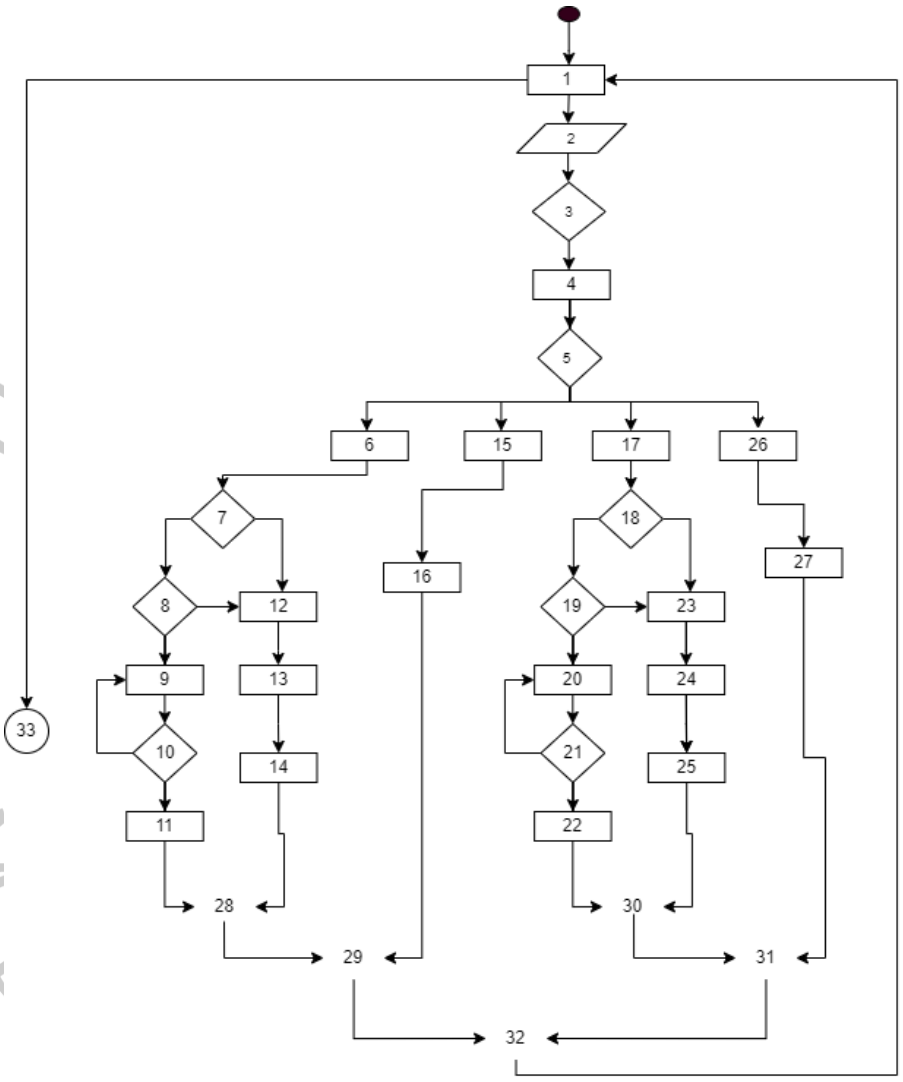
Pada perancangan sistem ini tentunya terdapat langkah-langkah peneliti dalam membuat suatu sistem aplikasi. Pada tahap ini pula peneliti merancanga dari awal mulai dari desain sampai prosedur apliaksi yang tentunay bertujuan menggambarkan jelas bagaimana aplikasi ini dapat memenuhi kebutuhan pengguna.

4.3.1. Flowchart

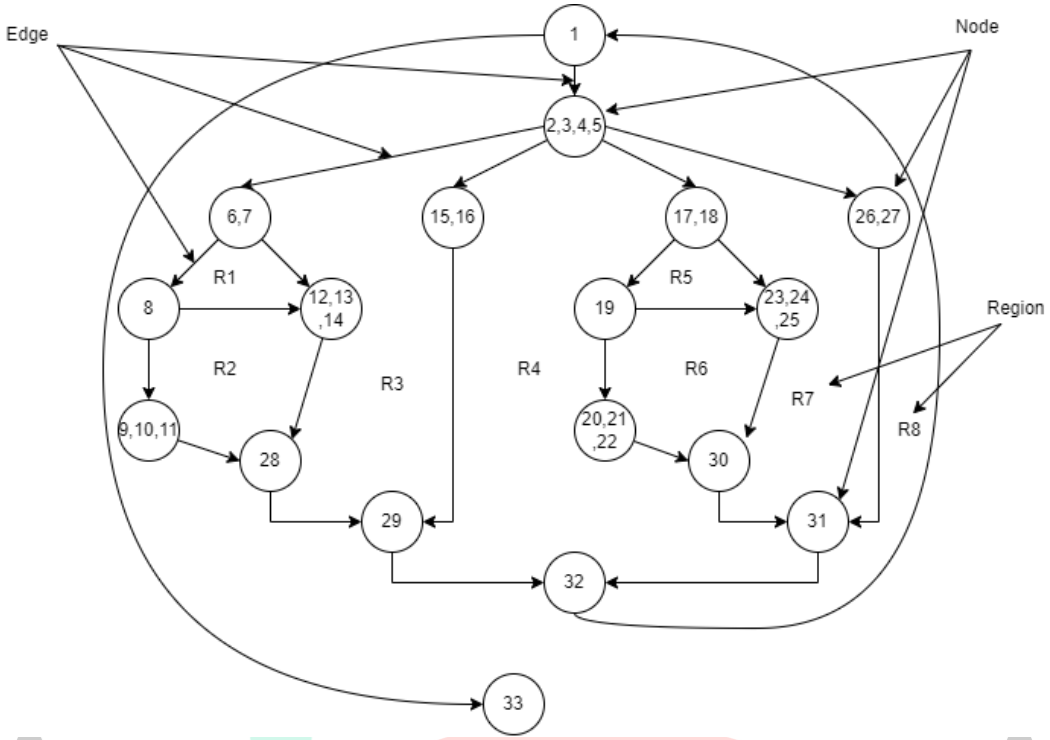


Gambar 4.3.1.1 Flowchart Monitoring Data Audit

Gambar 4.3.1.1. d iatas merupakan alur website Monitoring Data Audit Internal dimana harus memasukkan username dan password dengan benar agar dapat masuk ke dalam website tersebut. Admin tidak perlu melakukan registrasi karena akunnya telah dibuat langsung oleh admin. Untuk auditor sudah dipastikan memiliki akun secara langsung yang dibuat oleh admin.



Gambar 4.3.1.2 Penomoran Flowchart



Gambar 4.3.1.3 Flowgraph

Pada contoh Gambar 4.3.1.2 & Gambar 4.3.1.3 merupakan gambar flowchart dikonversi menjadi sebuah flowgraph. Semua notasi pada flowchart akan diubah menjadi notasi lingkaran/edge pada flowgraph. Berikut panduannya :

- Notasi yang berurutan baik berupa proses (persegi Panjang),input/output maupun seleksi (diamond) dapat digabungkan menjadi satu node.
    - Proses 2,4 dan decision 3,5 pada flowchart digabungkan menjadi satu node.
    - Proses 6 dan 7 digabungkan menjadi node 6,7.
- Notasi awal, digabungkan dengan notasi pertama menjadi satu node. Contoh: start node dan decision 1 pada flowchart menjadi node 1 pada flowgraph.
- Sebuah edge harus selalu dihubungkan dengan sebuah node, sehingga bentuk penggabungan dari beberapa edge dapat ditambahkan sebuah node sebelumnya menuju statement berikutnya meskipun node tersebut tidak memilik sebuah statement khusus. Contoh simpul pada nomor 28, 29, 30, 31, 32 dan 33 pada flowgraph.
- Final node pada flowchart diubah menjadi sebuah node pada flowgraph seperti contoh node 33.

Dari flowgraph yang sudah tersedia, dapat dibuatkan independent path dengan rumus dibawah ini :

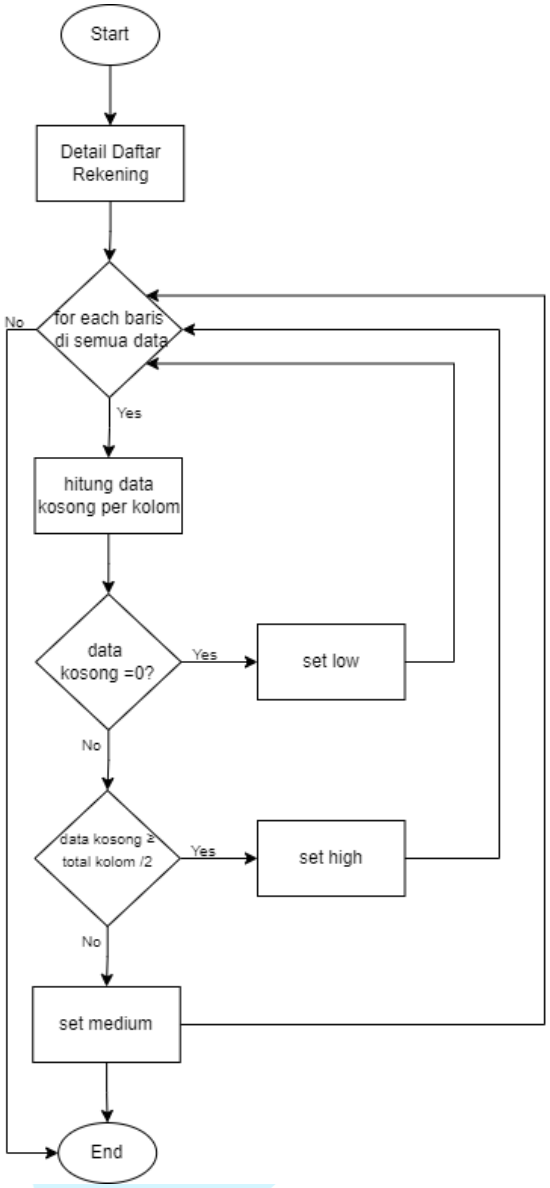
V(G) : Path Independent

E : Total jumlah Edge
N : Total jumlah Node
Rumus : V (G) = E – N + 2
Perhitungan sebagai berikut :
V (G) = 25 Edge – 18 nodes + 2 = 9

Hasil independent path pada contoh diatas dapat dijabarkan sebagai berikut :

Tabel 4.3.1 Idependent Path

Path 1	1-33
Path 2	1-2-3-4-5-6-7-8-9-10-11-28-29-32-1-33
Path 3	1-2-3-4-5-6-7-8-12-13-14-28-29-32-1-33
Path 4	1-2-3-4-5-6-7-12-13-14-28-29-32-1-33
Path 5	1-2-3-4-5-15-16-29-32-1-33
Path 6	1-2-3-4-5-17-18-19-20-21-22-30-31-32-1-33
Path 7	1-2-3-4-5-17-18-19-23-24-25-30-31-32-1-33
Path 8	1-2-3-4-5-17-18-23-24-25-30-31-32-1-33
Path 9	1-2-3-4-5-26-27-31-32-1-33

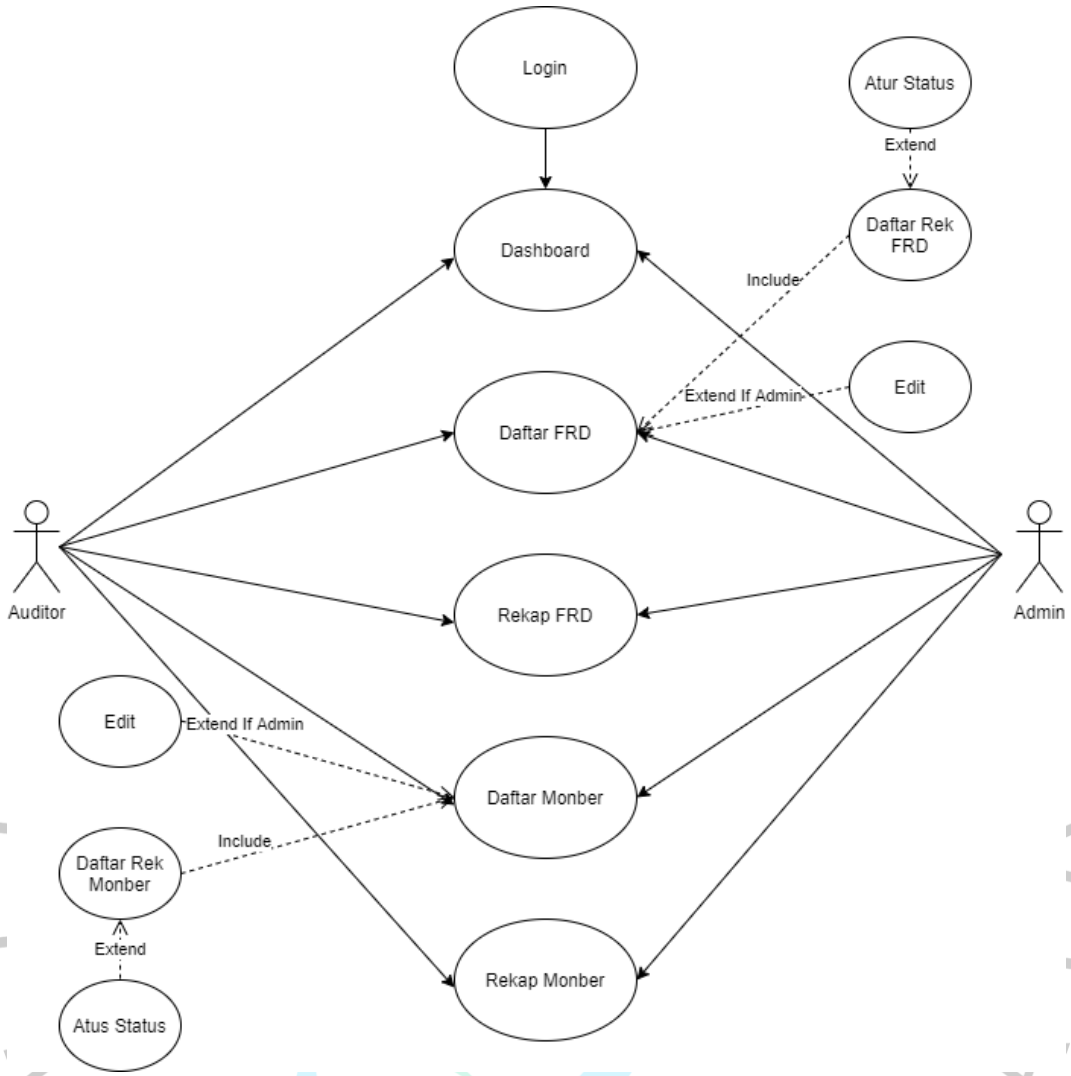


Gambar 4.3.1.4 Flowchart Algortima Mondi

Gambar 4.3.1.4 merupakan alur penghitungan hasil akhir yang menjadi rekomendasi dari website ini. User yang telah menginput semua parameter, kemudian melihat hasil dari rekomendasi yang datanya diolah menggunakan metode *forward chaining*. Hasil akhir proses ini ialah penentuan status prioritas.

4.3.2. Use Case Diagram

Pada perancangan use case website rekomendasi terdiri dari auditor dan admin. Penjelasan interaksi antara aktor pada diagram use case dipaparkan pada gambar berikut.

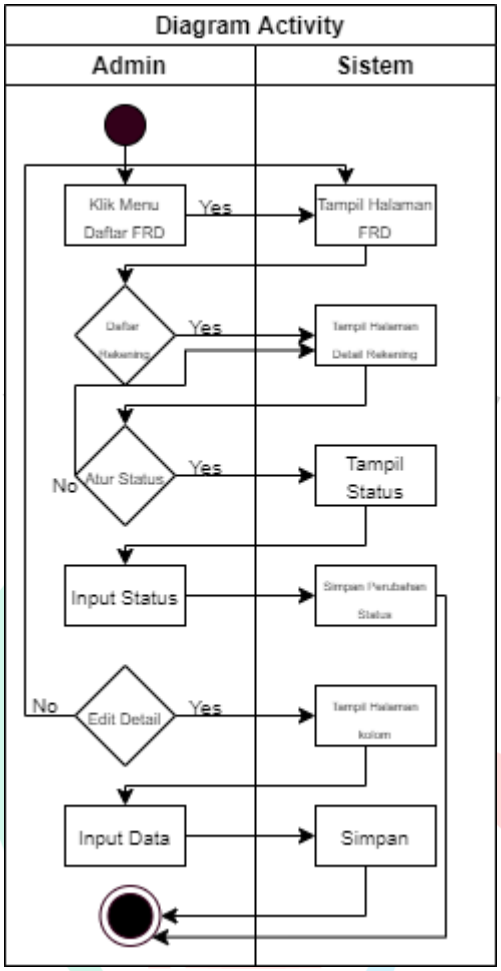


Gambar 4.3.2 Usecase Website Monitoring Data Audit

Auditor dalam aplikasi ini dapat melihat informasi dan mengatur daftar FRD, rekap FRD, daftar Monber dan rekap Monber. Kemudian dapat mengatur status yang ada di masing-masing tabel FRD dan Monber. Sedangkan Admin dalam aplikasi ini dapat mengatur dan melihat FRD, rekap FRD, daftar Monber dan rekap Monber serta dapat mengatur edit kolom yang ingin ditampilkan.

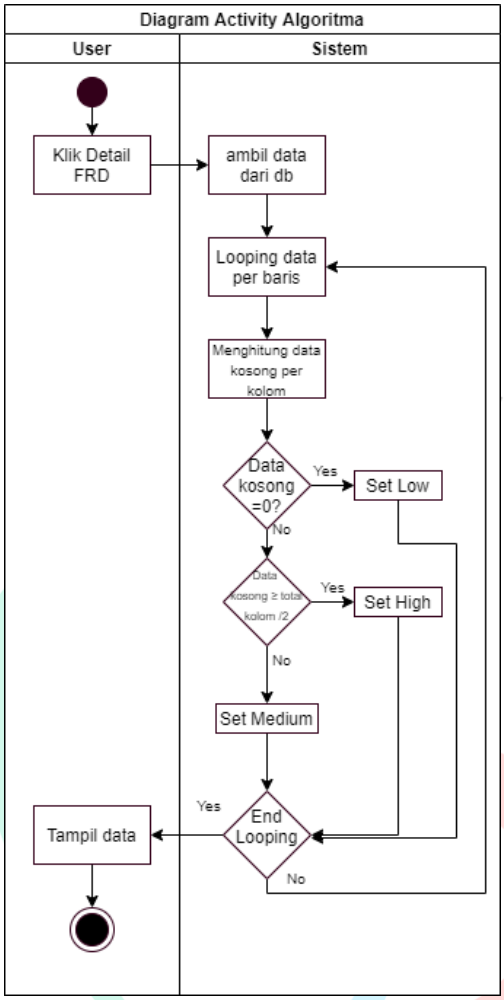
4.3.3. Activity Diagram

Pada activity diagram ini merupakan sebua gambaran proses sistem bekerja. Sistem berjalan dengan baik dan terarah jika memberikan alur proses interaksi dalam scenario proses sistem terhadap pengguna 40 nya. Representasi grafis menggunakan actor dalam proses aktivitasnya, action yang terjadi dipicu dari aktivitas actor dan memicu state lainnya. Berikut gambaran activity diagram pada perancangan website pemilihan bidang peminatan.



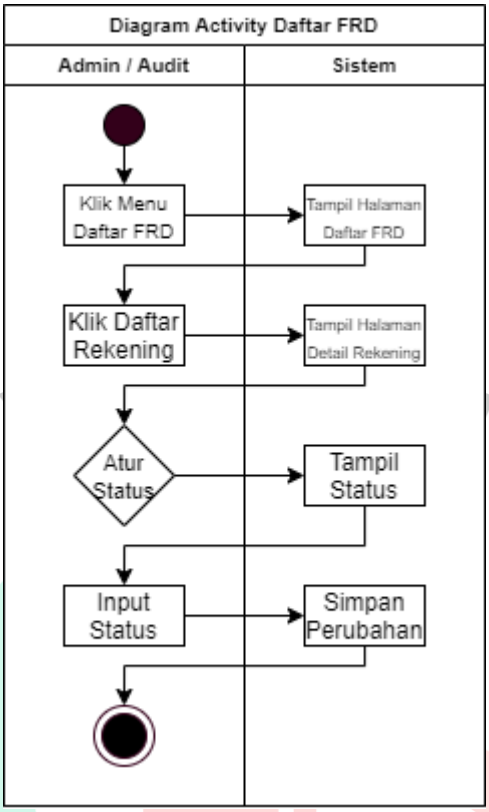
Gambar 4.3.3.1 Diagram Activity

Gambar 4.3.3.1 adalah rancangan jalan kerja keseluruhan sistem saat penggunakan menggunakannya. Terlihat pada gambar pengguna akan dapat melihat tampilan halaman FRD, detail rekening, dan tampil status. Pengguna juga dapat input status dan mengubah detail edit yang kemudian akan disimpan.



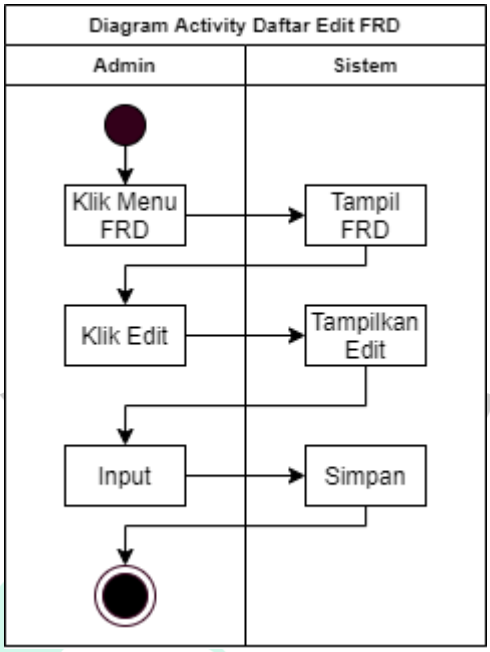
Gambar 4.3.3.2 Diagram Activity

Gambar 4.3.3.2 merupakan alur perhitungan menentukan hasil akhir yang menjadi rekomendasi dari website ini. User yang telah menginput semua parameter, kemudia melihat hasil rekomendasi yang datanya diolah menggunakan metode forward chaining. Hasil akhir proses ini ialah menentukan status yang merupakan prioritas.



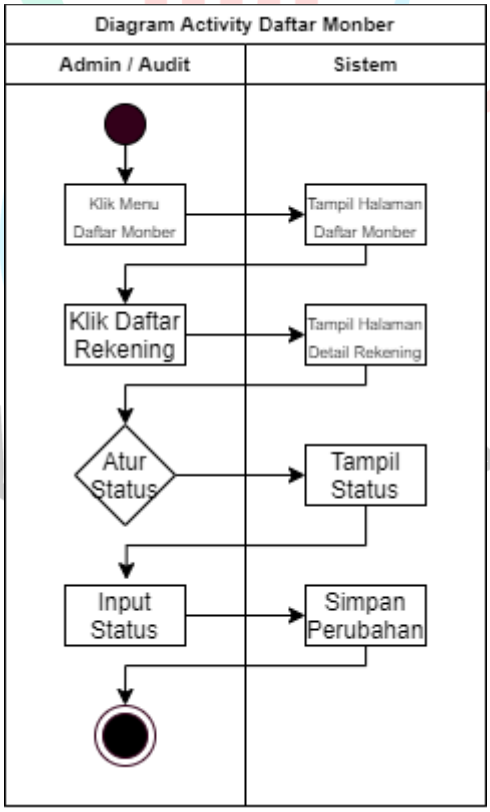
Gambar 4.3.3.3 Diagram Activity

Gambar 4.3.3.3 adalah rancangan jalan kerja saat admin ataupun auditor akan memulai klik menu Daftar FRD. Tampilan daftar FRD ini admin ataupun auditor akan dapat melihat tampilan halaman depan daftar FRD itu sendiri, detail rekening, dan halaman status. Aksi admin atau editor yang dapat dilakukan juga bisa melihat detail setiap data dan dapat mengatur status dari data-data tersebut.



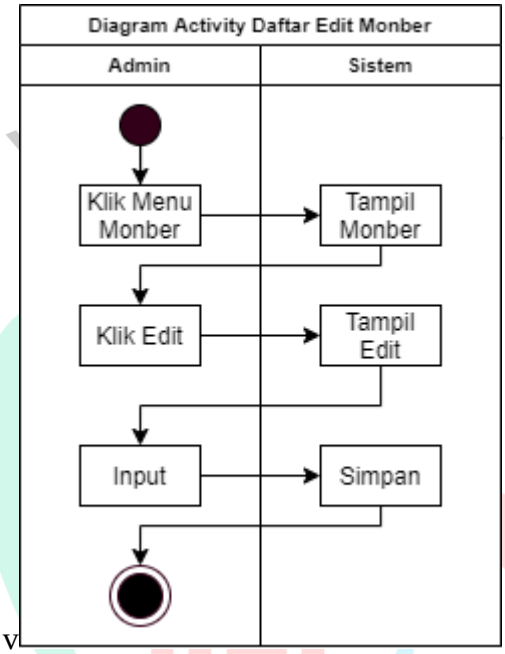
Gambar 4.3.3.4 Diagram Activity

Gambar 4.3.3.4 adalah rancangan jalan kerja saat admin ataupun auditor akan memulai klik menu Edit FRD. Tampilan edit FRD ini admin ataupun auditor akan dapat melihat tampilan FRD itu sendiri, dam tampilan untuk edit. Aksi admin atau auditor yaitu bisa mengedit data FRD tersebut.



Gambar 4.3.3.5 Diagram Activity

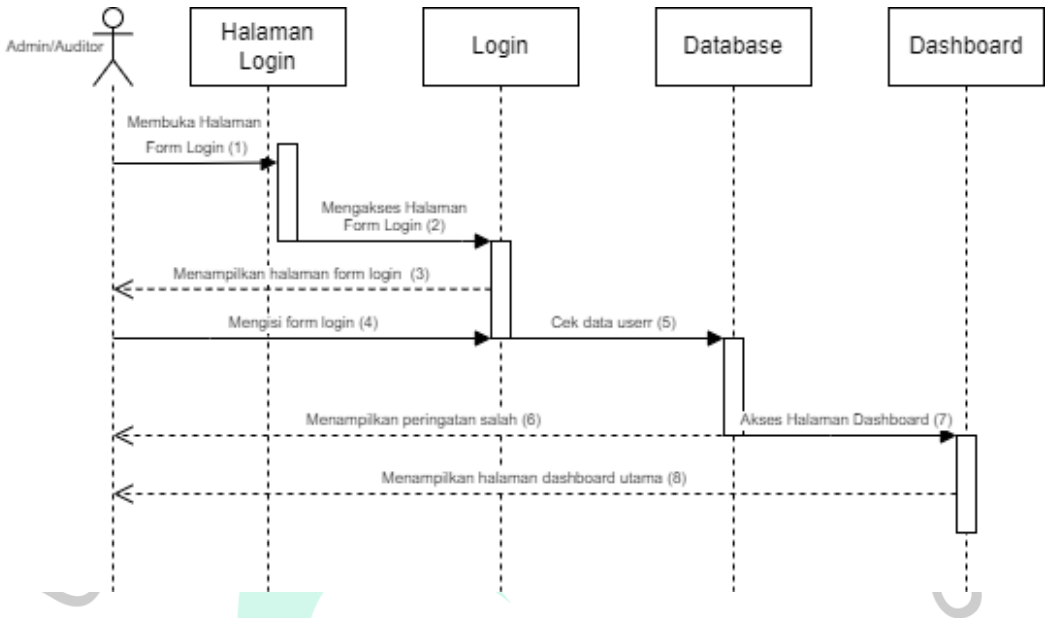
Gambar 4.3.3.5 adalah rancangan jalan kerja saat admin ataupun auditor akan memulai klik menu Daftar MONBER. Tampilan daftar MONBER ini admin ataupun auditor akan dapat melihat tampilan halaman depan daftar MONBER itu sendiri, detail rekening, dan halaman status. Aksi admin atau editor yang dapat dilakukan juga bisa melihat detail setiap data dan dapat mengatur status dari data-data tersebut.



Gambar 4.3.3.6 Diagram Activity

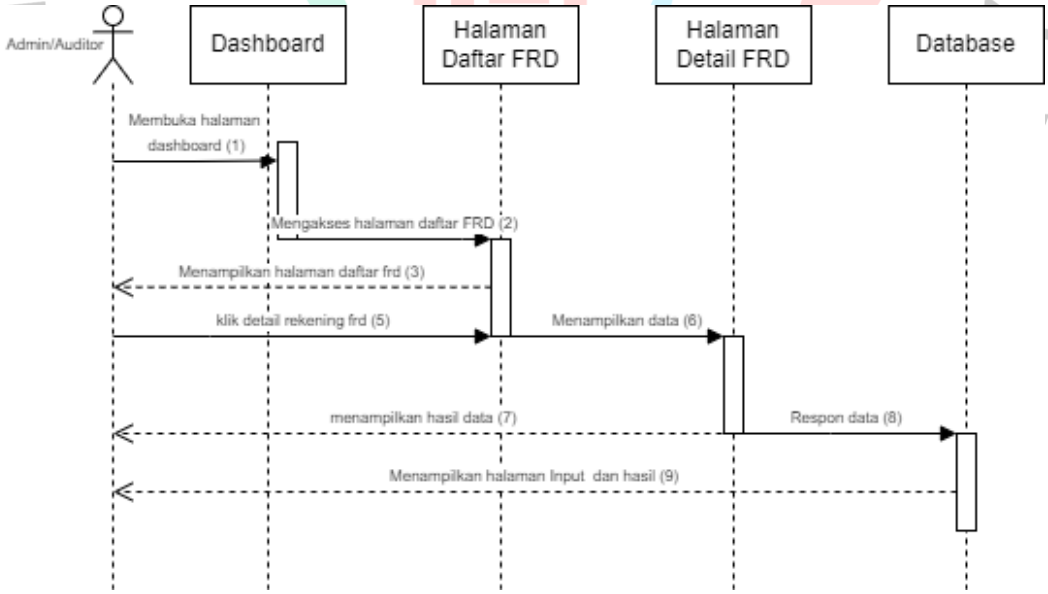
Gambar 4.3.3.6 adalah rancangan jalan kerja saat admin ataupun auditor akan memulai klik menu Edit MONBER. Tampilan edit MONBER ini admin ataupun auditor akan dapat melihat tampilan MONBER itu sendiri, dam tampilan untuk edit data. Aksi admin atau auditor yaitu bisa mengedit data MONBER tersebut.

4.3.4. Sequence Diagram



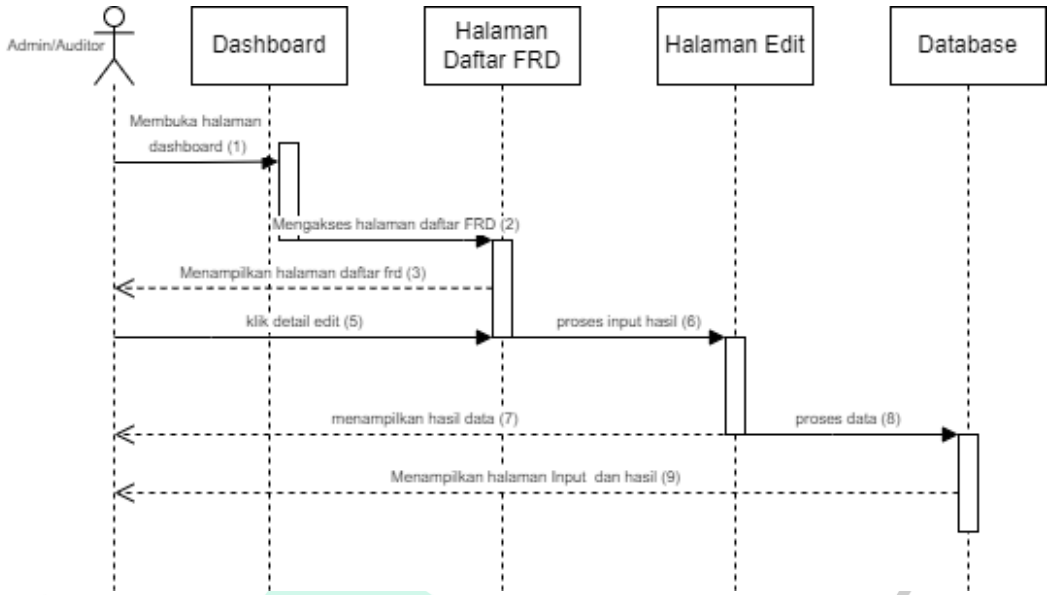
Gambar 4.3.4.1. Sequence Diagram Login User

Gambar 4.3.4.1 adalah rancangan tampilan login untuk Admin dan Auditor. User dan auditor sudah dibuatkan langsung oleh admin sehingga dapat mengakses database pada pada dashboard.



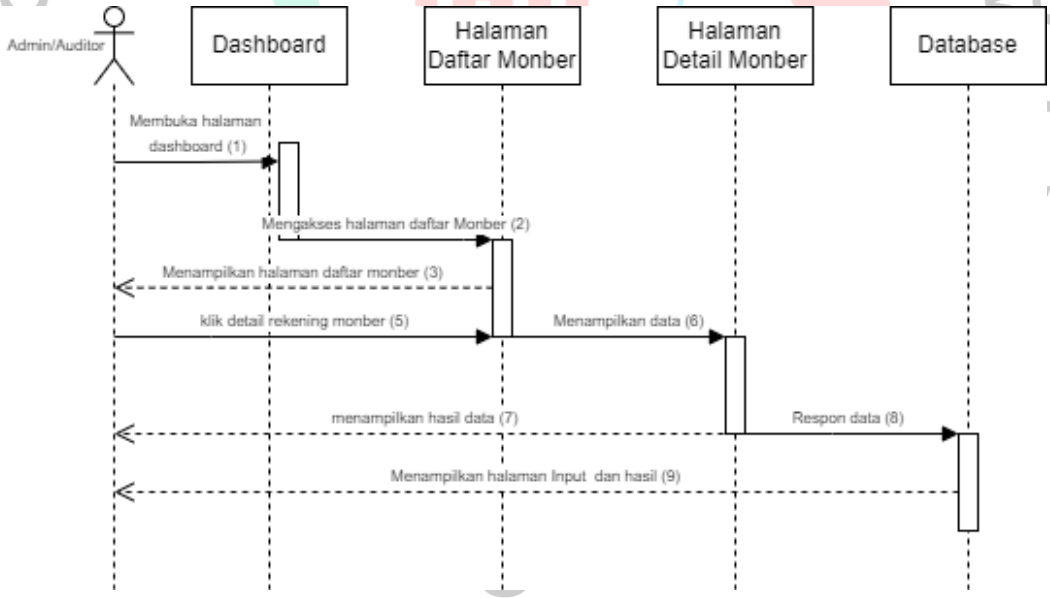
Gambar 4.3.4.2. Sequence Diagram Daftar FRD

Gambar 4.3.4.2 adalah rancangan tampilan Halaman Daftar FRD untuk Admin dan Auditor. User dan auditor sudah dibuatkan langsung oleh admin sehingga dapat mengakses detail data FRD pada database yang sudah tersedia.



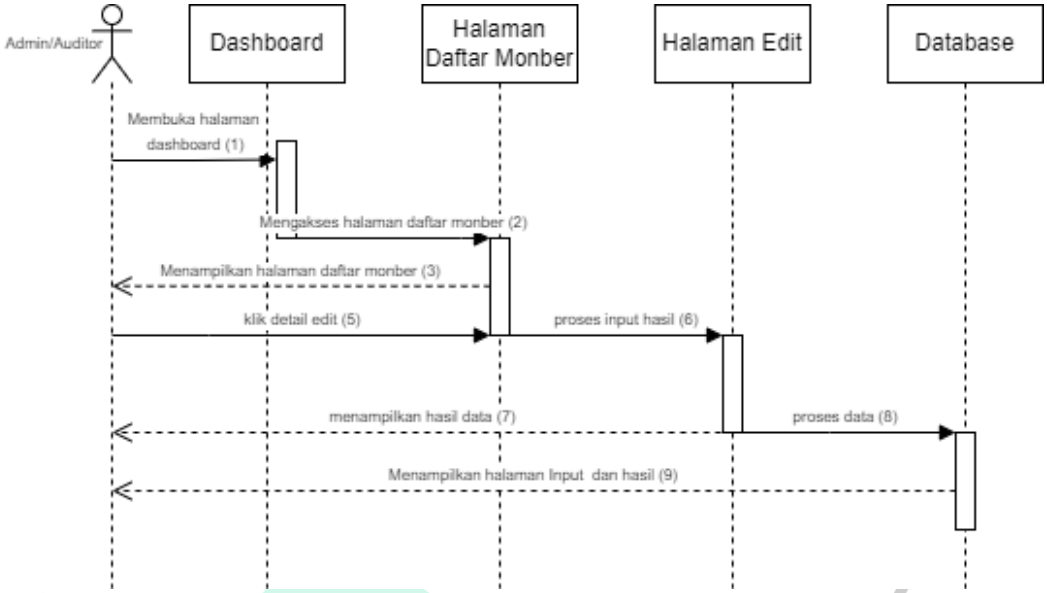
Gambar 4.3.4.3. Sequence Diagram Edit FRD

Gambar 4.3.4.3 adalah rancangan tampilan Halaman Daftar FRD untuk Admin dan Auditor untuk edit status. User dan auditor dapat menginput dan mengedit pada halaman ini.



Gambar 4.3.4.4. Sequence Diagram Daftar Monber

Gambar 4.3.4.4 adalah rancangan tampilan Halaman Daftar MONBER untuk Admin dan Auditor. User dan auditor sudah dibuatkan langsung oleh admin sehingga dapat mengakses detail data MONBER pada database yang sudah tersedia.

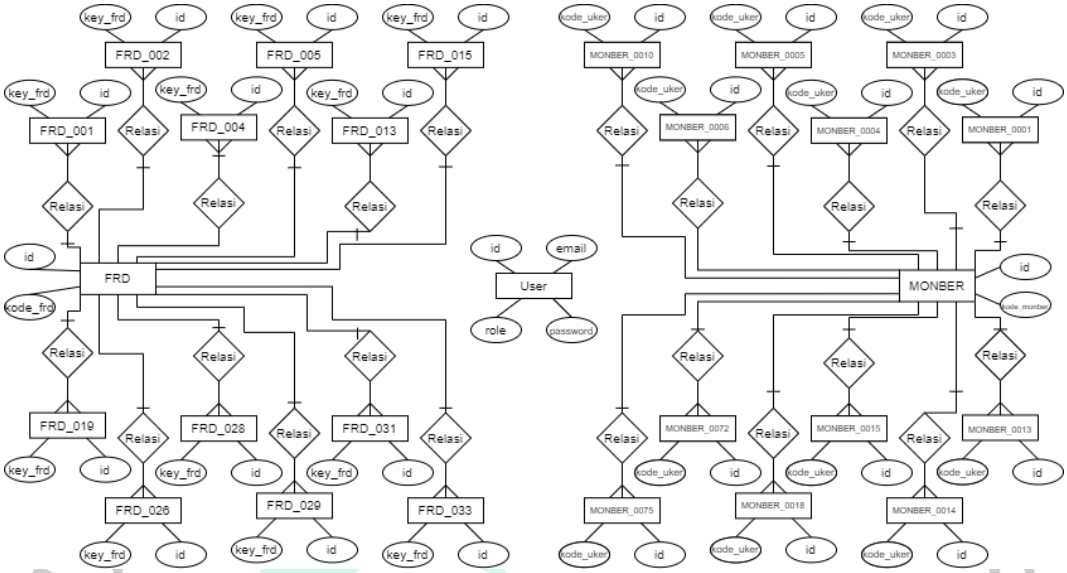


Gambar 4.3.4.5. Sequence Diagram Edit Monber

Gambar 4.3.4.5

adalah rancangan tampilan Halaman Daftar MONBER untuk Admin dan Auditor untuk edit status. User dan auditor dapat menginput dan mengedit pada halaman ini.

4.3.5. Entity Relationship Diagram (ERD)



Gambar 4.3.5. Entity Relationsip Diagram (ERD)

Gambar 4.3.5. terdapat tabel data yang saling berelasi satu sama lain dengan miliki sebuah hubungan atau kardinalitas yang akan dijabarkan sebagai berikut.

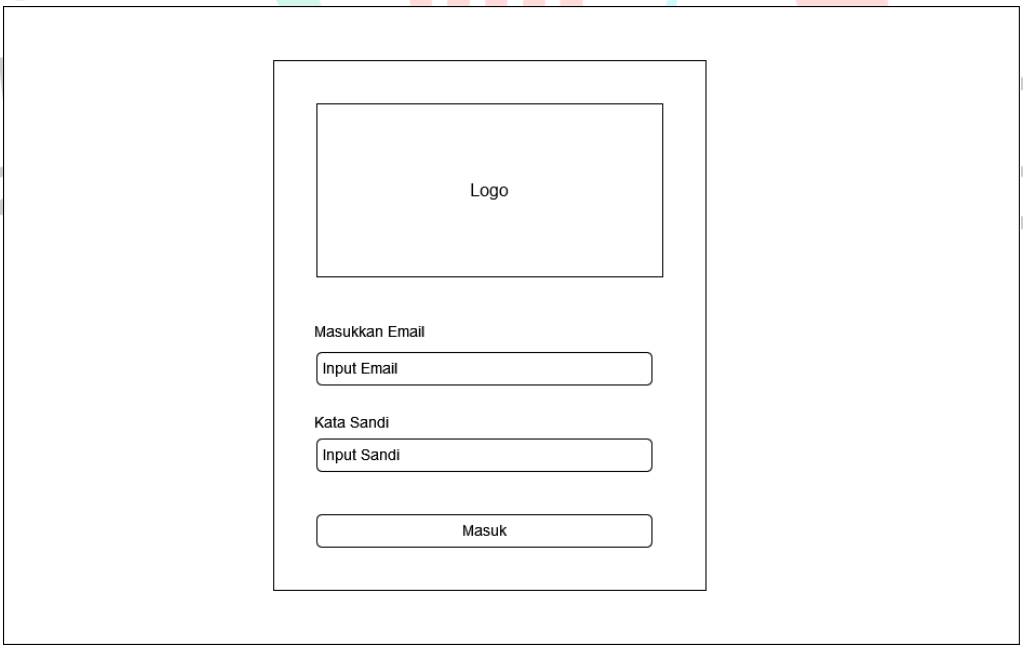
Tabel 4.3.5. Kardinalitas

No	*Entitas Awal*	*Entitas Akhir*	*Relasi*
1	FRD_001	FRD	Many to One
2	FRD_002	FRD	Many to One
3	FRD_004	FRD	Many to One
4	FRD_005	FRD	Many to One
5	FRD_013	FRD	Many to One
6	FRD_015	FRD	Many to One
7	FRD_019	FRD	Many to One
8	FRD_026	FRD	Many to One
9	FRD_028	FRD	Many to One
10	FRD_029	FRD	Many to One
11	FRD_031	FRD	Many to One
12	FRD_033	FRD	Many to One
13	MONBER_0001	MONBER	Many to One
14	MONBER_0003	MONBER	Many to One
15	MONBER_0004	MONBER	Many to One
16	MONBER_0005	MONBER	Many to One
17	MONBER_0006	MONBER	Many to One
18	MONBER_0010	MONBER	Many to One
19	MONBER_0013	MONBER	Many to One
20	MONBER_0014	MONBER	Many to One
21	MONBER_0015	MONBER	Many to One
22	MONBER_0018	MONBER	Many to One
23	MONBER_0072	MONBER	Many to One
24	MONBER_0075	MONBER	Many to One

Tabel 4.3.5. menjelaskan tentang kardinalitas dari masing entitas yang ada pada basis data website Dashboard Monitoring Data Audit Internal, dimana entitas FRD_001 dengan FRD memiliki relasi many to one, entitas FRD_002 dengan FRD

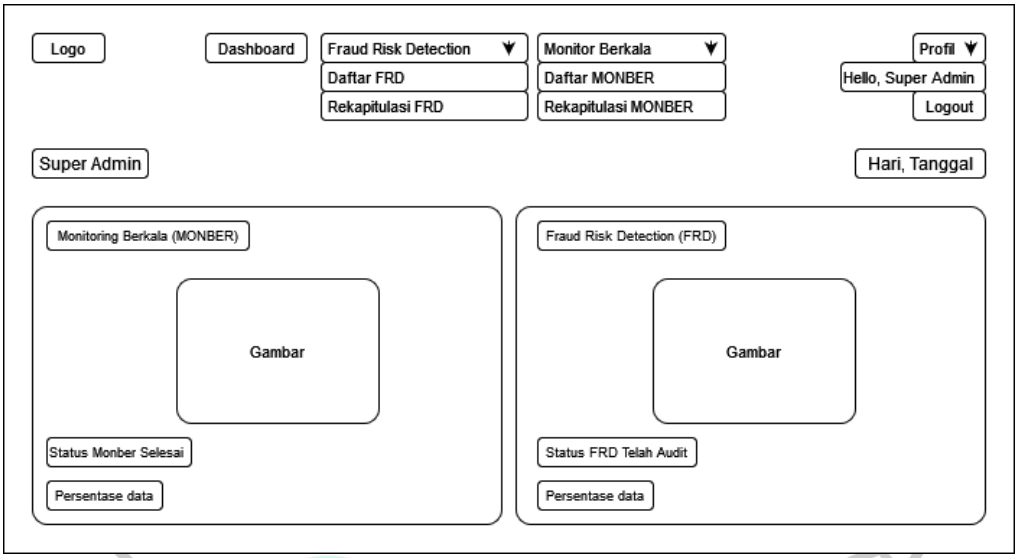
memiliki relasi many to one, entitas FRD_004 dengan FRD memiliki relasi many to one, entitas FRD_005 dengan FRD memiliki relasi many to one, entitas FRD_013 dengan FRD memiliki relasi many to one, entitas FRD_015 dengan FRD memiliki relasi many to one, entitas FRD_019 dengan FRD memiliki relasi many to one, entitas FRD_026 dengan FRD memiliki relasi many to one, entitas FRD_028 dengan FRD memiliki relasi many to one, entitas FRD_029 dengan FRD memiliki relasi many to one, entitas FRD_031 dengan FRD memiliki relasi may to one, entitas FRD_033 dengan FRD memiliki relasi many to one, entitas MONBER_0001 dengan MONBER memiliki relasi many to one, entitas MONBER_0003 dengan MONBER memiliki relasi many to one, entitas MONBER_0004 dengan MONBER memiliki relasi many to one, entitas MONBER_0005 dengan MONBER memiliki relasi many to one, entitas MONBER_0006 dengan MONBER memiliki relasi many to one, entitas MONBER_0010 dengan MONBER memiliki relasi many to one, entitas MONBER_0013 dengan MONBER memiliki relasi many to one, entitas MONBER_0014 dengan MONBER memiliki relasi many to one, entitas MONBER_0015 dengan MONBER memiliki relasi many to one, entitas MONBER_0018 dengan MONBER memiliki relasi many to one, entitas MONBER_0072 dengan MONBER memiliki relasi many to one, entitas MONBER_0075 dengan MONBER memiliki relasi many to one.

4.3.6. Perancangan Antar Muka



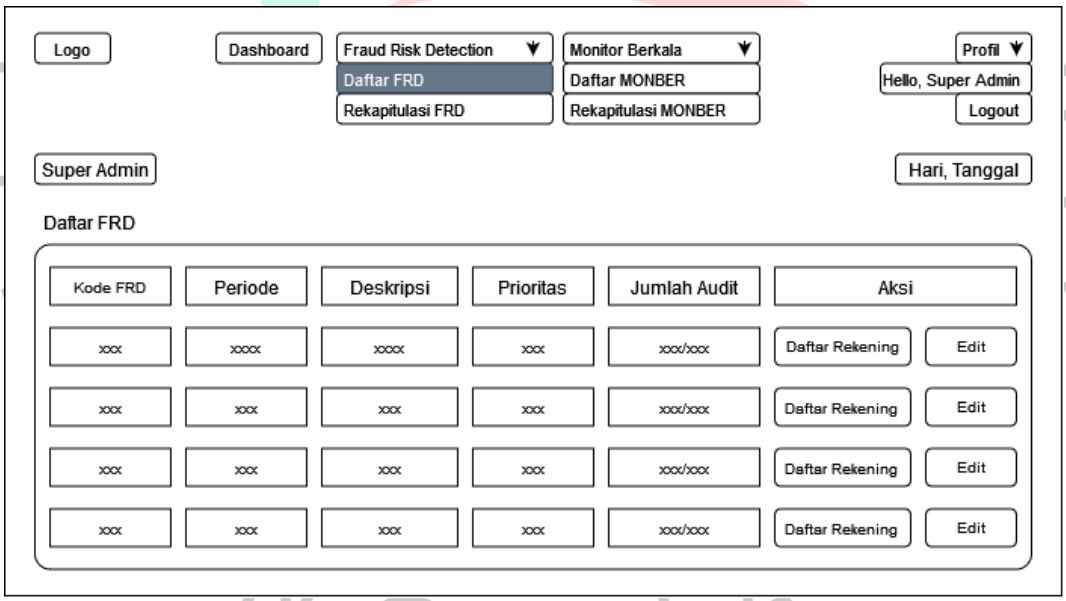
Gambar 4.3.6.1. Tampilan Login

Gambar 4.3.6.1. adalah rancangan tampilan login untuk Admin dan Auditor. User untuk auditor sudah dibuatkan langsung oleh admin.



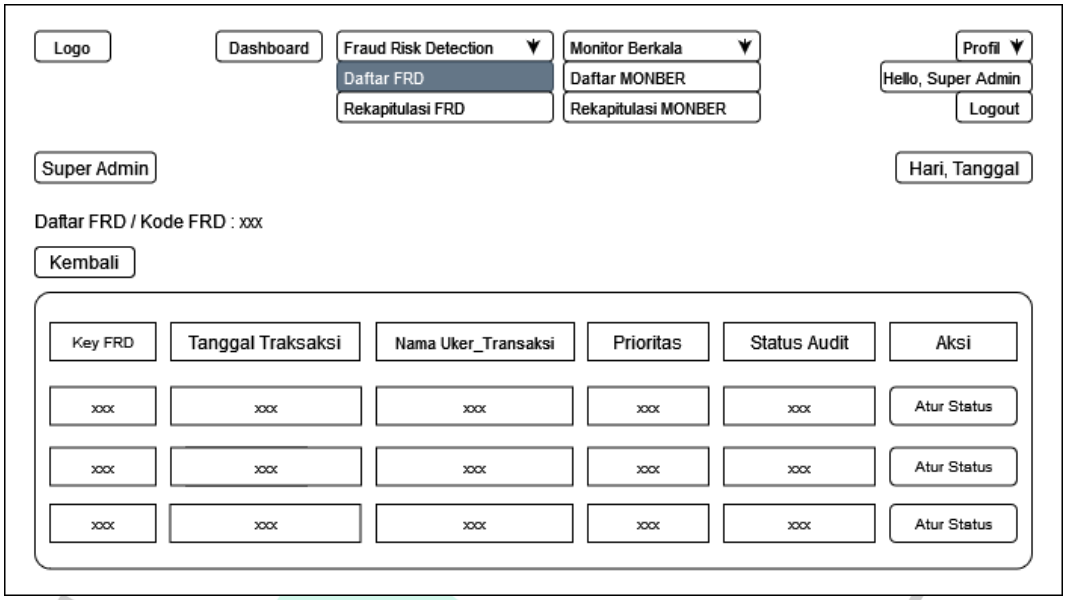
Gambar 4.3.6.2. Rancangan Tampilan Utama

Gambar 4.3.6.2. adalah rancangan tampilan halaman utama sebagai user admin pada website Monitoring Data Audit Internal, terdapat informasi status hasil yang sudah diaudit dan presentase dari data tersebut.



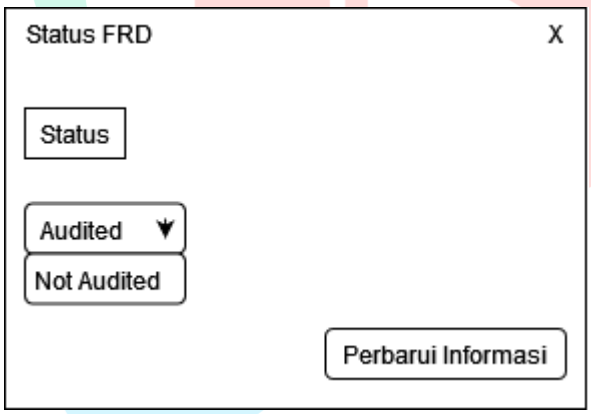
Gambar 4.3.6.3. Rancangan Tampilan Daftar FRD

Gambar 4.3.6.3. adalah rancangan tampilan daftar FRD sebagai user admin pada website Monitoring Data Audit Internal. Terdapat informasi mengenai data yang akan ditindak lanjut.



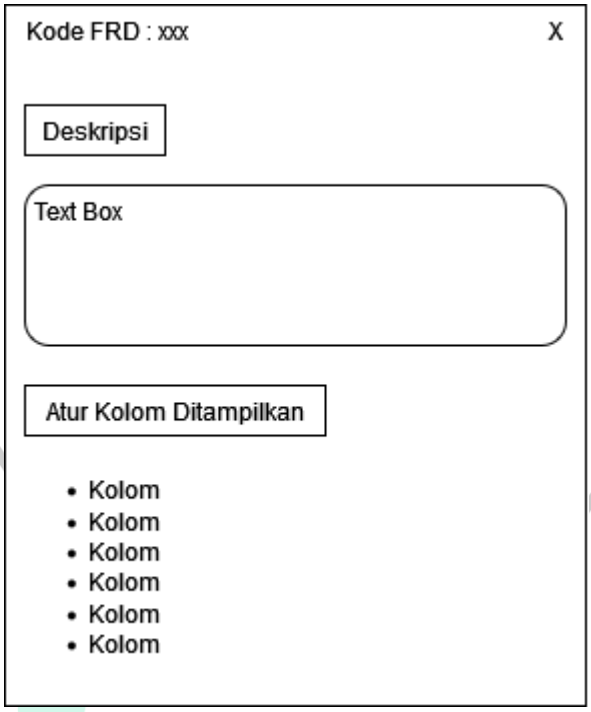
Gambar 4.3.6.4. Rancangan Tampilan Daftar Rekening pada Daftar FRD

Gambar 4.3.6.4. d iatas adalah rancangan tampilan daftar rekening yang terdapat pada daftar FRD. Terdapat informasi mengenai data yang lebih detail dan tombol aksi untuk mengatur status yang akan ditindak lanjut.



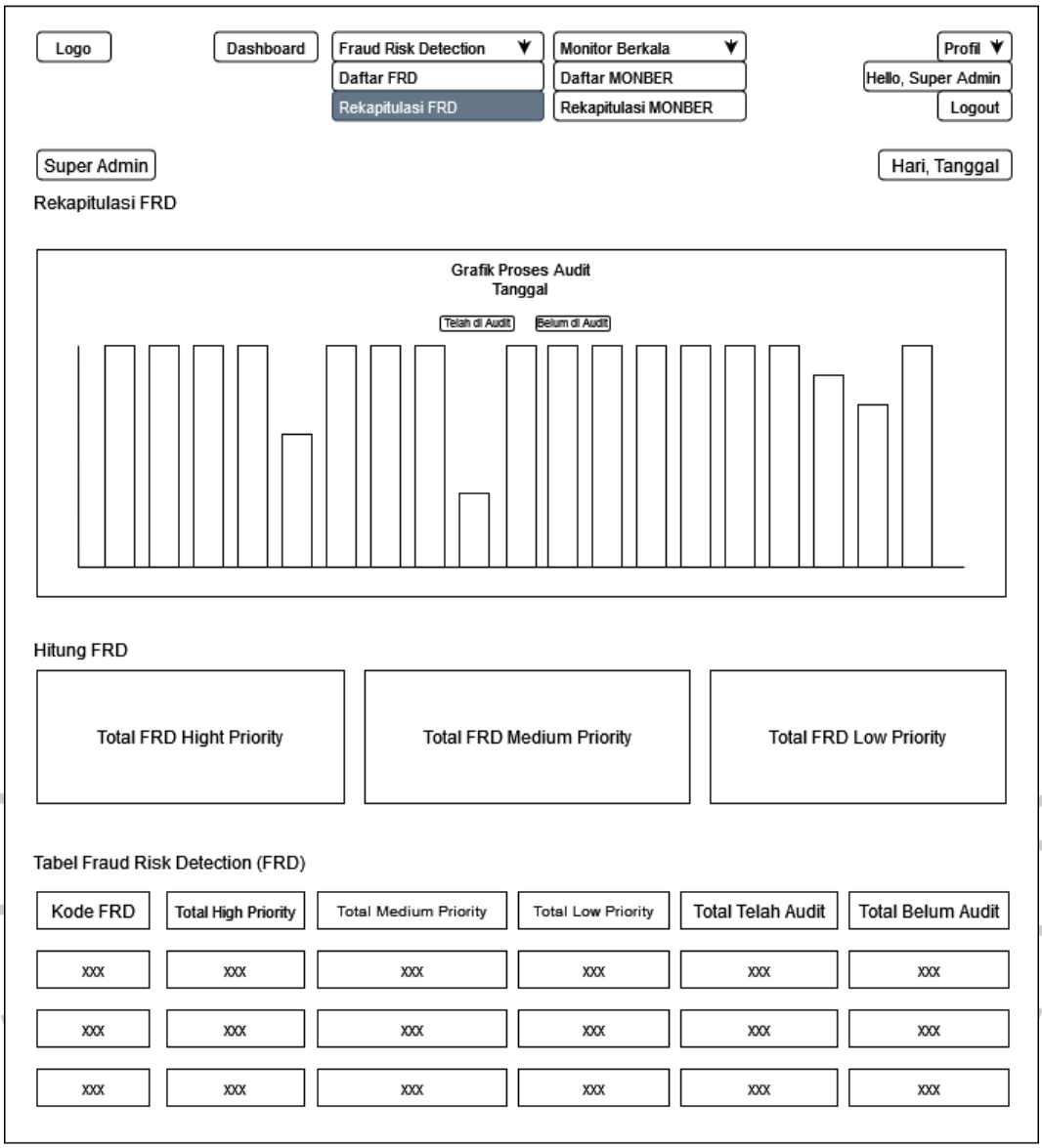
Gambar 4.3.6.5. Rancangan Tampilan Atur Status pada Daftar Rekening

Gambar 4.3.6.5. adalah rancangan tampilan atur status yang terdapat pada daftar rekening. Terdapat informasi untuk mengatur status yang akan ditindak lanjut.



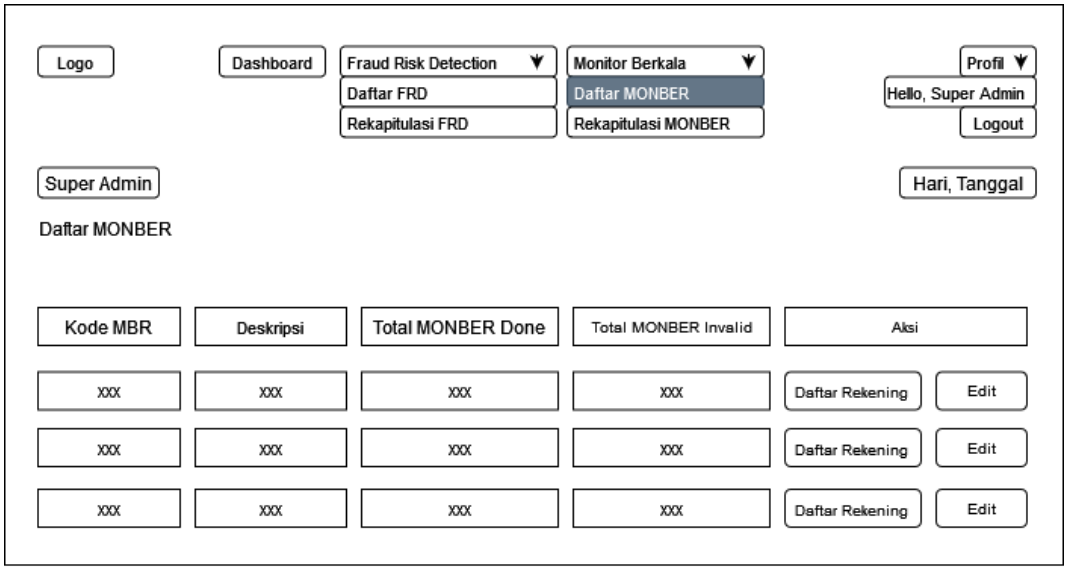
Gambar 4.3.6.6. Rancangan Tampilan menu Edit pada Daftar FRD

Gambar 4.3.6.6. adalah rancangan tampilan menu edit untuk mengatur tampilan kolom yang terdapat pada daftar rekening. Terdapat informasi untuk mengatur text box dan atur kolom.



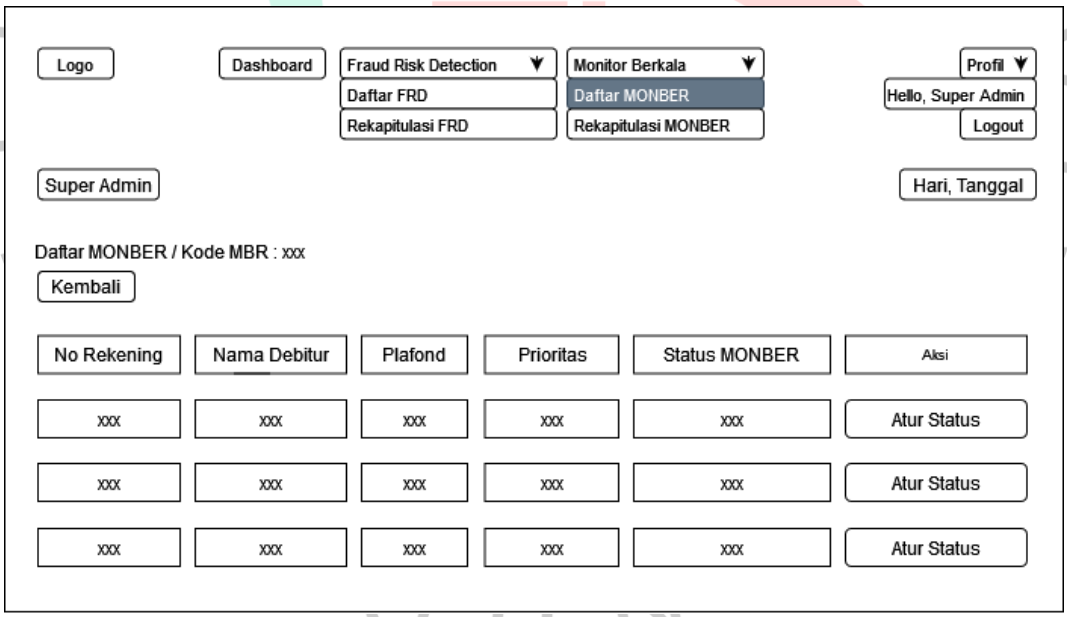
Gambar 4.3.6.7. Rancangan Tampilan Rekapitulasi FRD

Gambar 4.3.6.7. adalah rancangan tampilan menu rekapitulasi Monber untuk adminr. Terdapat informasi mengenai hasil grafik, total hitung dan tabel detail hasil Monber. Pengguna bisa melihat dengan garfik yang dapat difilteralisasi berdasarkan kelompok data dan berbentuk box.



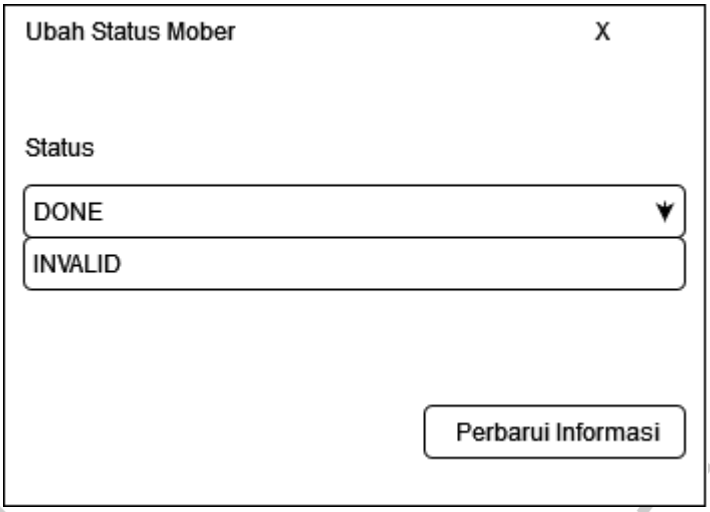
Gambar 4.3.6.8. Rancangan Tampilan Daftar Monber

Gambar 4.3.6.8. adalah rancangan tampilan daftar Monber sebagai user admin pada website Monitoring Data Audit Internal. Terdapat informasi mengenai data yang akan ditindak lanjut.



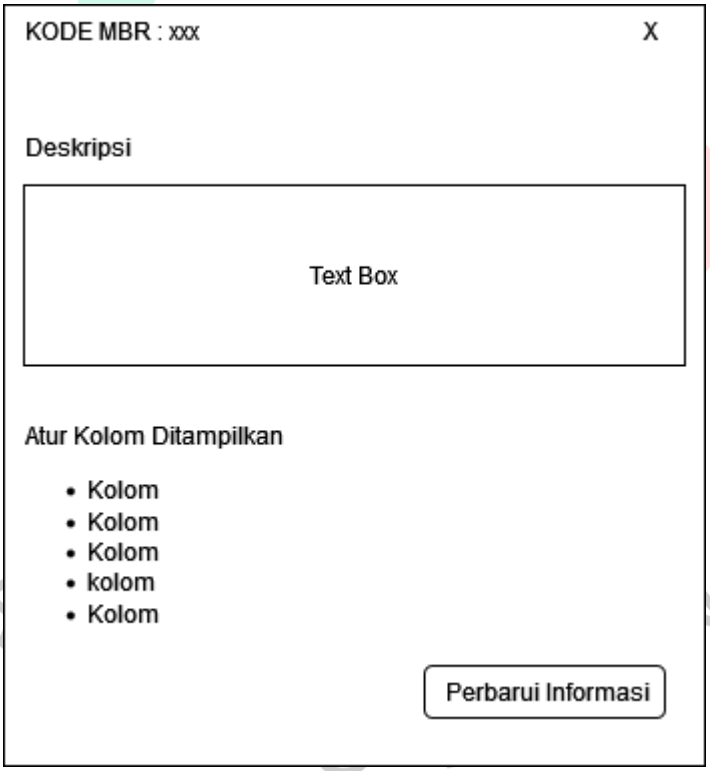
Gambar 4.3.6.9. Rancangan Tampilan Daftar Rekening pada Daftar Monber

Gambar 4.3.6.9. adalah rancangan tampilan daftar rekening yang terdapat pada daftar Monber. Terdapat informasi mengenai data yang lebih detail dan tombol aksi untuk mengatur status yang akan ditindak lanjut.



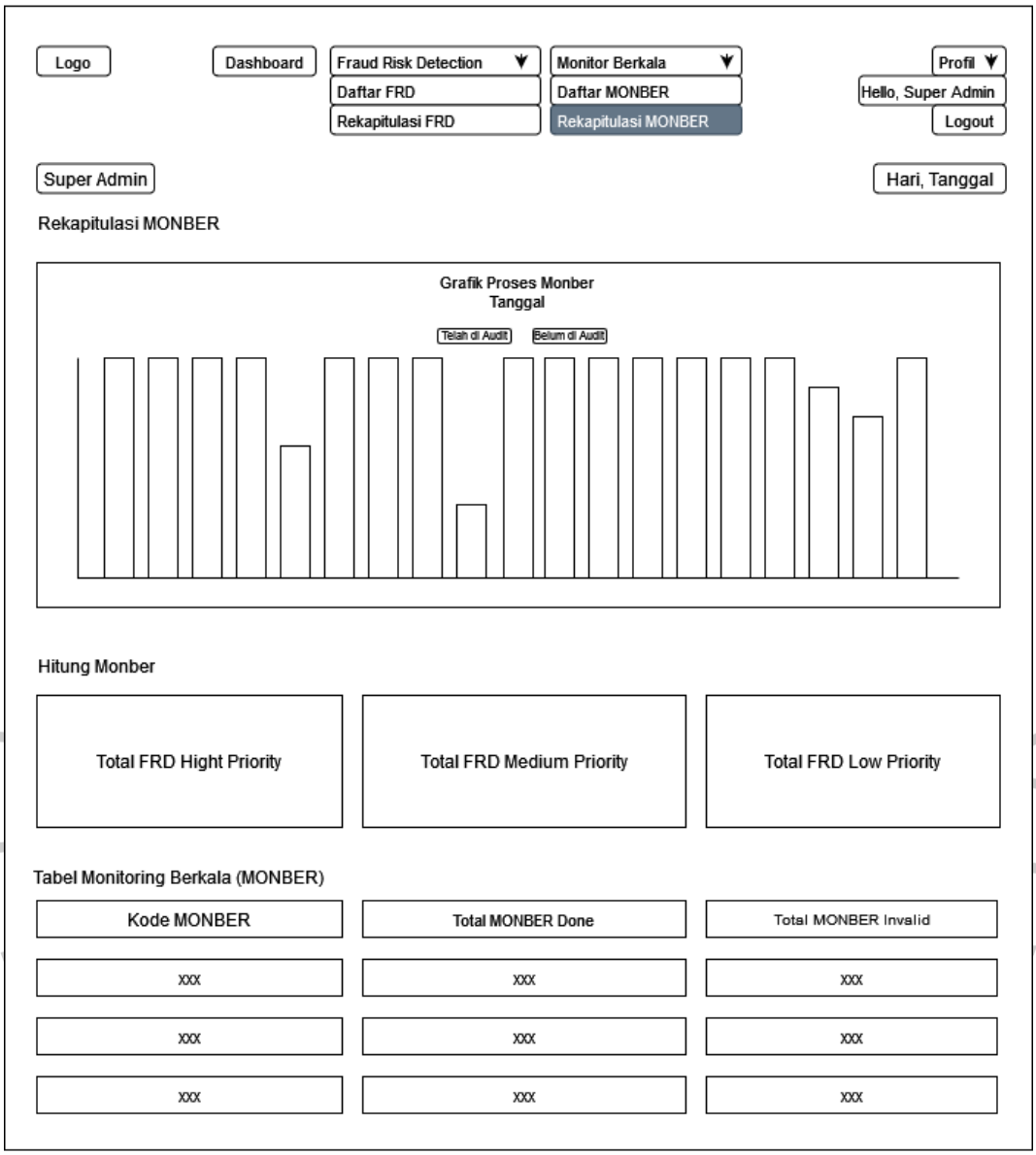
Gambar 4.3.6.10. Rancangan Tampilan Atur Status pada Daftar Monber

Gambar 4.3.6.10. adalah rancangan tampilan atur status yang terdapat pada daftar rekening. Terdapat informasi untuk mengatur status yang akan ditindak lanjut.



Gambar 4.3.6.11. Rancangan Tampilan menu Edit pada Daftar Monber

Gambar 4.3.6.11. adalah rancangan tampilan atur status yang terdapat pada daftar rekening. Terdapat informasi untuk mengatur status yang akan ditindak lanjut.



Gambar 4.3.6.12. Rancangan Tampilan Rekapitulasi Monber

Gambar 4.3.6.12. adalah rancangan tampilan menu rekapitulasi Monber untuk auditor. Terdapat informasi mengenai hasil grafik, total hitung dan tabel detail hasil Monber. Pengguna bisa melihat dengan garfik yang dapat difilteralisasi berdasarkan kelompok data dan berbentuk box.

4.3.7. Perancangan Pengujian

Pada tahap perancangan pengujian ini peneliti bertujuan untuk mencatat semua pengujian dengan menggunakan metode *black box* dan *white box testing*. Pengujian ini juga suda dibuatkan rencana dengan sistematis seingga pengujian akan dilakukan sesuai dengan tabel rencana yang telah dibuat.

4.3.7.1. Rancangan Pengujian Black Box

Rancangan pada tahap pengujian sistem menggunakan pengujian black box untuk mengetahui sisi fungsionalitas perangkat lunak. Table pengujian menggunakan black box dijabarkan sebagai berikut.

Tabel 4.3.7.1. Rancangan Pengujian Black Box

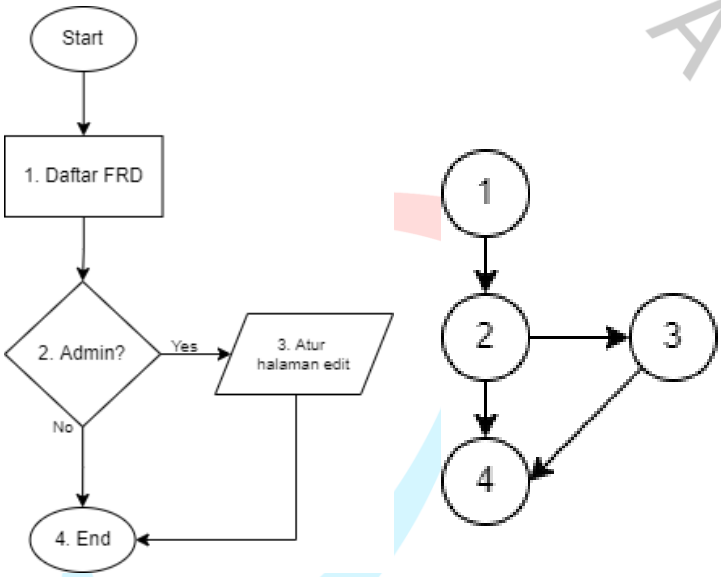
No	Pengujian	Hasil yang diharapkan	Hasil Pengamatan	Kesimpulan
1	Pengguna membuka website Monitoring Data Indikator	Menampilkan Halaman Utama		
2	Admin berhasil menginput username dan password	Menampilkan halaman Dashboard		
3	Admin melakukan Klik Daftar FRD	Menampilkan halaman Daftar FRD		
4	Admin melakukan klik detail rekening	Menampilkan halaman detail rekening		
5	Admin melakukan klik atur status	Menampilkan popup ubah status		
6	Admin melakukan klik edit pada halaman daftar frd	Menampilkan popup edit		
7	Admin melakukan klik Rekapitulasi FRD	Menampilkan halaman Rekapitulasi FRD		
8	Admin melakukan Klik Daftar Monitoring Berkala	Menampilkan halaman Daftar Monber		
9	Admin melakukan klik detail daftar rekening	Menampilkan halaman detail rekening		
10	Admin melakukan Klik ubah status pad a monber	Menampilkan pop up pada ubah status		
11	Admin melakukan klik detail daftar rekening	Menampilkan pop up edit pada daftar monber		

12	Admin melakukan klik Rekapitulasi Monber	Menampilkan halaman Rekapitulasi Monber		

4.3.7.2. Rancangan Pengujian White Box

Rancangan pengujian kali ini dengan menggunakan metode white box, yang nantinya hasil akan dicatat akan sesuai dengan tabel perencanaan yang telah dibuat oleh peneliti. Setiap pengujian akan dicatat secara rinci untuk mengetahui kelayakan aplikasi.

4.3.7.2.1. Rancangan pengujian white box admin



Gambar 4.3.7.2.1.1 Flowgraph Admin Daftar FRD

Gambar 4.3.7.2.1.1 adalah flowgraph pada saat admin mengatur halaman edit kolom yang akan ditampilkan pada daftar detail rekening FRD. Flowgraph tersebut memilik 2 jalur independent yaitu : 1) Saat user masuk ke daftar FRD, 2) Saat user bukan admin.

V(G) : Path Independent N : Total jumlah Node

E : Total jumlah Edge Rumus : V (G) = E – N + 2

Perhitungan sebagai berikut :

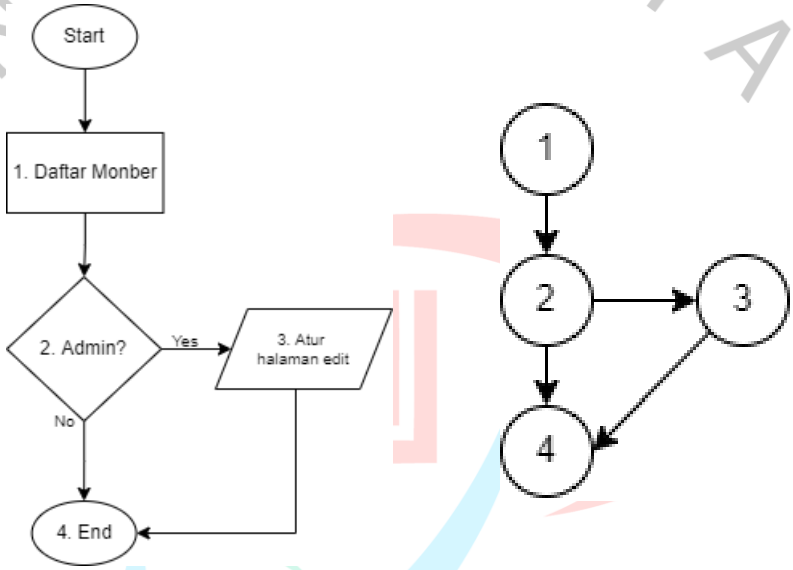
V (G) = 4 Edge – 4 nodes + 2 = 2

Hasil Independent path diatas adalah :

Path 1 :	1-2-3-4
Path 2 :	1-2-4

Jalur independent tersebut dijabarkan sebagai berikut :

1. User klik daftar FRD, setelah itu menuju nomor 2.
2. User apakah admin, jika admin maka menuju ke nomor 3, jika tidak maka menuju nomor 4
3. User mengatur halaman edit lalu menuju nomor 4
4. Flow selesai.



Gambar 4.3.7.2.1.2 Flowgraph Admin Monber

Gambar 4.3.7.2.1.2 adalah flowgraph pada saat admin mengatur halaman edit kolom yang akan ditampilkan pada daftar detail rekening Monber. Flowgraph tersebut memilik 2 jalur independent yaitu : 1) Saat user masuk ke daftar Monber, 2) Saat user bukan admin.

V(G) : Path Independent N : Total jumlah Node
E : Total jumlah Edge Rumus : V (G) = E – N + 2
Perhitungan sebagai berikut :

V (G) = 4 Edge – 4 nodes + 2 = 2

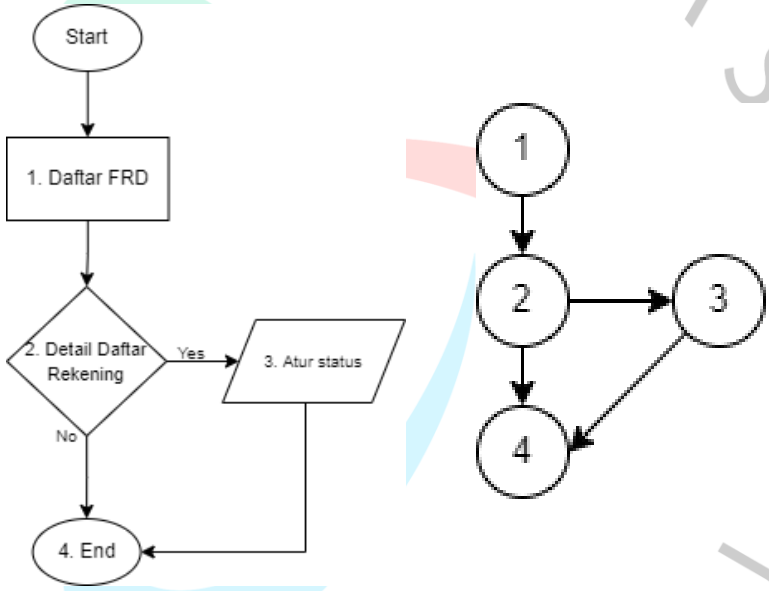
Hasil Independent path diatas adalah :

Path 1 :	1-2-3-4
Path 2 :	1-2-4

Jalur independent tersebut dijabarkan sebagai berikut :

1. User klik daftar Monber, setelah itu menuju nomor 2.
2. User apakah admin, jika admin maka menuju ke nomor 3, jika tidak maka menuju nomor 4
3. User mengatur halaman edit lalu menuju nomor 4
4. Flow selesai.

4.3.7.2.2. Rancangan pengujian white box auditor



Gambar 4.3.7.2.2.1 Flowgraph Auditor FRD

Gambar 4.3.7.2.2.1 adalah flowgraph pada saat auditor mengatur status untuk tindak lanjut resiko pada halaman FRD. Flowgraph tersebut memilik 2 jalur independent yaitu : 1) Saat user masuk ke daftar FRD, 2) Saat user tidak mengatur status.

V(G) : Path Independent
E : Total jumlah Edge
N : Total jumlah Node
Rumus : V (G) = E – N + 2

Perhitungan sebagai berikut :

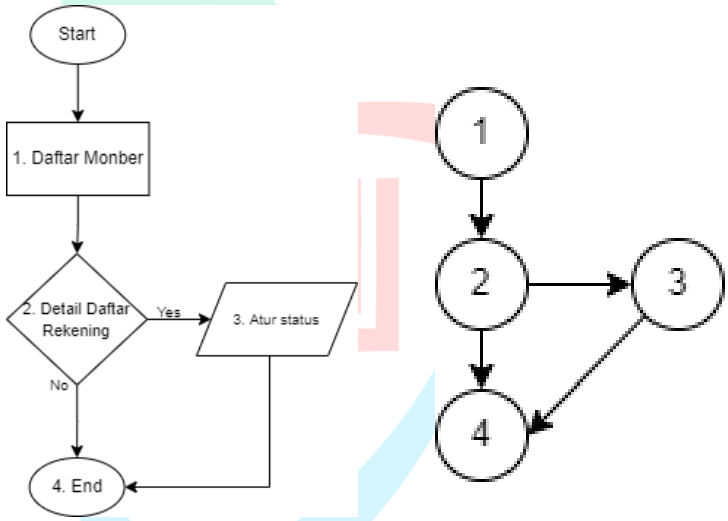
V (G) = 4 Edge – 4 nodes + 2 = 2

Hasil Independent path diatas adalah :

Path 1 : 1-2-3-4
Path 2 : 1-2-4

Jalur independent tersebut dijabarkan sebagai berikut :

1. User klik daftar FRD, setelah itu menuju nomor 2.
2. User klik Detail Daftar Rekening, jika user melakukan klik detail daftar rekening maka menuju ke nomor 3, jika tidak maka menuju nomor 4
3. User mengatur atur status lalu menuju nomor 4
4. Flow selesai.



Gambar 4.3.7.2.2.2 Flowgraph Auditor Monber

Gambar 4.3.7.2.2.2 adalah flowgraph pada saat auditor mengatur status untuk tindak lanjut resiko pada halaman Monber. Flowgraph tersebut memilik 2 jalur independent yaitu : 1) Saat user masuk ke daftar Monber, 2) Saat user tidak mengatur status.

V(G) : Path Independent
E : Total jumlah Edge
N : Total jumlah Node
Rumus : V (G) = E – N + 2

Perhitungan sebagai berikut :

V (G) = 4 Edge – 4 nodes + 2 = 2

Hasil Independent path diatas adalah :

Path 1 :	1-2-3-4
Path 2 :	1-2-4

Jalur independent tersebut dijabarkan sebagai berikut :

1. User klik daftar Monber, setelah itu menuju nomor 2.
2. User klik Detail Daftar Rekening, jika user melakukan klik detail daftar rekening maka menuju ke nomor 3, jika tidak maka menuju nomor 4
3. User mengatur atur status lalu menuju nomor 4
4. Flow selesai.

Tabel 4.3.7.2.2. Rancangan Pengujian White Box

No	Program
1	``` import React, { useState, useEffect } from 'react' import NumberFormat from 'react-number-format' import ImgPresentaseFRD from './presentasefrd.jpg' import style from './style.module.scss' const PresentaseFrd = () => { const [frds, setFrds] = useState([]); useEffect(() => { getDataFrd() }, []); const getDataFrd = () => { fetch(`${process.env.REACT_APP_API  }frd/daftarfrd`) .then(response => response.json()) .then(data => { setFrds(data); }); } // eslint-disable-next-line no-return-assign const totalAudit = frds.reduce((total, currentValue) => total += currentValue.total_audit,0); // eslint-disable-next-line no-return-assign const totalNotAudit = frds.reduce((total, currentValue) => total += currentValue.total_not_audit,0); const totalFrd = totalAudit + totalNotAudit; const totaPresentaseFrd = (totalAudit * 100) / totalFrd;; const thousandSeparator = true; return (<div> <p className="text-center mt-3"> </p> <div className="card-body"> <div className="text-dark font-size-18 font-weight-bold mb-1">Status FRD Telah Audit</div> <div className="text-gray-6 mb-2">Nilai Presentase</div> <div className="font-weight-bold font-size-36 text-success mb-2"> <NumberFormat value={totalAudit} displayType="text" thousandSeparator={thousandSeparator} /> </div> <div className="d-flex align-items-center"> <div className={`${style.progressIcon} bg-success text-white mr-3`}> <i className="fe fe-check-circle font-size-18" /> </div> <div className="flex-grow-1"> <div className="text-dark font-size-18 font-weight-bold text-nowrap mb-2"> <NumberFormat value={totaPresentaseFrd.toFixed(1)} displayType="text" thousandSeparator={thousandSeparator} />% dari <NumberFormat value={totalFrd} displayType="text" thousandSeparator={thousandSeparator} /> </div> <div className="progress"> <div className="progress-bar bg-success" style={{ width: `${totaPresentaseFrd.toFixed(1)}%` }} /> </div> </div> </div> </div> </div>) } export default PresentaseFrd ``` Hasil yang di harapkan : Menampilkan halaman dashboard dan memunculkan total keseluruhan data yang akan ditindak lanjuti.

2

```
// Get Data From Database
refreshList() {
    fetch(`${process.env.REACT_APP_API  }frd/daftarfrd`)
    .then(response => response.json())
    .then(data => {
        this.setState({frds:data});
    });
}
```

```
render() {

 const { roleUser } = this.props;

 const columns = [
   {
     title: 'Kode FRD',
     dataIndex: 'kode_frd',
     key: 'kode_frd',
     ...this.getColumnSearchProps('kode_frd'),
   },
   {
     title: 'Periode',
     dataIndex: 'periode',
     key: 'periode',
     render: (text) => (<span className="font-size-16">{`${text}`}</span>),
   },
   {
     title: 'Deskripsi',
     dataIndex: 'deskripsi',
     key: 'deskripsi',
     render: (text) => (<span className="font-size-16">{`${text}`}</span>),
   },
   {
     title: 'Prioritas',
     key: 'prioritas',
     render: (props) => (
       <span className="font-size-16">
         {props.total_high_priority > props.total_medium_priority && props.total_high_priority > props.total_low_priority &&
           <Tag color="#DA0037">High</Tag>
         }
         {props.total_medium_priority > props.total_high_priority && props.total_medium_priority > props.total_low_priority &&
           <Tag color="#FD8C04">Medium</Tag>
         }
         {props.total_low_priority > props.total_medium_priority && props.total_low_priority > props.total_high_priority &&
           <Tag color="#7C83FD">Low</Tag>
         }
       </span>
     ),
   },
   {
     title: 'Jumlah Audit',
     key: 'audit',
     render: (props) => (
       <span className="font-size-16 font-weight-bold">
         <span className="text-success">{`${props.total_audit}`}</span> /
         <span className="text-danger">{` ${props.total_audit + props.total_not_audit}`}</span>
       </span>
     ),
   },
   {
     title: 'Aksi',
     key: 'action',
     width: 100,
     render: (props) => (
       <span>
         <Link to={`frd/${props.id}/${props.kode_frd}/list`} className="btn btn-primary mr-2">
           <i className="fe fe-eye mr-2" />
           Detail Daftar Rekening
         </Link>
         {roleUser === "super admin" &&
           <button type="button" className="btn btn-warning text-white" onClick={() => this.showModal(true, props.kode_frd.replace('_',''), props.deskripsi, props.id)}>
             <i className="fe fe-edit mr-2" />
             Edit
           </button>
         }
       </span>
     ),
   },
 ]
```

Hasil yang di harapkan :
Menampilkan data pada halaman daftar FRD dan memunculkan total keseluruhan data yang akan ditindak lanjuti.

3

```
// Tampil Data Daftar Rekening
useEffect(() => {
  fetch(`${process.env.REACT_APP_API  }frd/daftarfrd/${idfrd}`)
  .then(response => response.json())
  .then(table => {
    if(table.tampil_kolom) {
      setData(table.tampil_kolom.split(","));
    }
  });
}, [idfrd]);
```

Hasil yang di harapkan :
Menampilkan data pada halaman daftar detail FRD dan memunculkan total keseluruhan data yang akan ditindak lanjuti.

4

```
// Update Atur Status Daftar Rekening
handleUpdate = () => {

  const body = this.state;

  fetch(`${process.env.REACT_APP_API  }frd/ubahrekening`, {
    method: 'POST',
    headers: {
      'Content-Type':'application/json'
    },
    body: JSON.stringify({
      idfrd: body.idfrd,
      id: body.idData,
      status: body.status,
      prioritas: body.prioritas
    })
  })
  .then(response => response.json())
  .then(() => {
    notification.success({
      message: 'Perubahan Berhasil',
      description: `Data Berhasil Diperbaharui!`,
    });
    this.refreshList();
  });

  this.handleReset();

}

// Tampil pop up atur status
showModal = (modal, idData, status, prioritas, idfrd) => {
  if(idData > 0) {
    fetch(`${process.env.REACT_APP_API  }frd/ambilrekening/${idData}/${idfrd}`)
    .then(response => response.json())
    .then(() => {
      this.setState({
        visible: modal,
        idData,
        status: status !== "" && status !== null ? status : 'Not Audited',
        prioritas: prioritas !== "" && prioritas !== null ? prioritas : 'High',
        idfrd,
      })
    });
  } else {
    this.setState({
      visible: modal
    })
  }

}
```

Hasil yang di harapkan :
Menampilkan halaman atur edit pada daftar FRD

5	``` { title: 'STATUS AUDIT', dataIndex: 'Status_Audit', key: 'Status_Audit', fixed: 'right', width: 150, render: (text) => (<> {text && text !== '' && text !== 'Not Audited' ? <Tag color="green">AUDITED</Tag> : <Tag color="red">NOT AUDITED</Tag> } </>) }, ``` Hasil yang di harapkan : Menampilkan pop up atur status pada halaman detail daftar rekening FRD

6

```
const GrafikAudit = () => {

  const [chartData, setChartData] = useState({});

  const chart = () => {
    const frdL = [];
    const frdA = [];
    const frdNa = [];

    fetch(`${process.env.REACT_APP_API  }frd/daftarfrd`)
      .then(response => response.json())
      .then(data => {
          for (const dataObj of data)
          {
            frdL.push(dataObj.kode_frd.replace('_','-'));
            frdA.push(dataObj.total_audit);
            frdNa.push(dataObj.total_not_audit);
          }

          setChartData({
            labels: frdL,
            datasets: [
              {
                label: 'Telah di Audit',
                backgroundColor: 'rgba(75, 192, 192, 0.7)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 1,
                data: frdA,
                barPercentage: 0.7,
                order: 1
              },
              {
                label: 'Belum di Audit',
                backgroundColor: 'rgba(255, 99, 132, 0.7)',
                borderColor: 'rgba(255,99,132,1)',
                borderWidth: 1,
                data: frdNa,
                barPercentage: 0.7,
                order: 2
              }
            ],
          });
      });
  }
```

```
  const barOptions = {
    tooltips: {
      mode: 'label',
      callbacks: {
          label(t, d) {
            const dstLabel = d.datasets[t.datasetIndex].label;
            const {yLabel} = t;
            return `${  yLabel  } FRD ${ dstLabel }`;
          }
      }
    },
    scales: {
      xAxes: [
        {
          stacked: true,
        },
      ],
      yAxes: [
        {
          stacked: true,
        },
      ],
    },
  }

  useEffect(() => {
    chart();
  }, []);

  return (
    <div>
      <Bar data={chartData} options={barOptions} width={400} height={100}
    </div>
  )
}

export default GrafikAudit
```

```
// Algoritma Utama
const defaultColumn = [
  {
    title: 'PRIORITAS',
    // dataIndex: 'Prioritas',
    key: 'Prioritas',
    fixed: 'right',
    width: 150,
    render: props => {
      let totalColumnEmpty = 0;
      for (let index = 0; index < columns.length; index++) {
        if(!props[columns[index]]) {
          totalColumnEmpty += 1;
        }
      }


      let colorPriority = '';
      let textPriority = '';
      if (totalColumnEmpty === 0) {
        colorPriority = '#7C83FD';
        textPriority = 'LOW';
      } else if (totalColumnEmpty >= Math.floor(columns.length / 2)) {
        colorPriority = '#DA0037';
        textPriority = 'HIGH';
      }
      else {
        colorPriority = '#FD8C04';
        textPriority = 'MEDIUM';
      }

      return (
        <>
          <span className="font-size-22 font-weight-bold">
            <Tag color={colorPriority}>{textPriority}</Tag>
          </span>
        </>
      )

    }
  },
```

Hasil yang di harapkan :
Menampilkan seluruh data untuk rekapitulasi FRD

7

```
// Get Data From Database
refreshList() {
  fetch(`${process.env.REACT_APP_API  }monber/daftarmonber`)
  .then(response => response.json())
  .then(data => {
      this.setState({mbrs:data});
  });
}
```

```
render() {

  const { roleUser } = this.props;

  const columns = [
    {
      title: 'Kode MBR'.toUpperCase(),
      dataIndex: 'kode_monber',
      key: 'kode_monber',
      ...this.getColumnSearchProps('kode_monber'),
    },
    {
      title: 'Deskripsi'.toUpperCase(),
      dataIndex: 'deskripsi',
      key: 'deskripsi',
      render: (text) => (<span className="font-size-16">{`${text}`}</span>),
    },
    {
      title: 'Total MONBER Done'.toUpperCase(),
      dataIndex: 'total_done',
      key: 'total_done',
      render: (text) => (
        <span className="font-size-22 font-weight-bold">
          <Tag color="green">{`${text}`}</Tag>
        </span>
      ),
    },
    {
      title: 'Total MONBER Invalid'.toUpperCase(),
      dataIndex: 'total_invalid',
      key: 'total_invalid',
      render: (text) => (
        <span className="font-size-22 font-weight-bold text-white">
          <Tag color="red">{`${text}`}</Tag>
        </span>
      ),
    },
{
  title: 'Aksi'.toUpperCase(),
  key: 'action',
  width: 100,
  render: (props) => (
    <span>
      <Link to={`monber/${props.id}/${props.kode_monber}/list`} className="btn btn-primary mr-2">
        <i className="fe fe-eye mr-2" />
        Detail Daftar Rekening
      </Link>
      {roleUser === "super admin" &&
        <button type="button" className="btn btn-warning text-white" onClick={() => this.showModal(true, props.kode_monber.replace('MONBER','MBR').replace('_','
          <i className="fe fe-edit mr-2" />
          Edit
        </button>
      }
    </span>
  ),
},
]
```

Hasil yang di harapkan :
Menampilkan data halaman daftar Monber dan memunculkan total keseluruhan data yang akan ditindak lanjuti.

8

```
// Get Data From Database
refreshList() {

  fetch(`${process.env.REACT_APP_API  }monber/daftardebitur/${this.id}`)
  .then(response => response.json())
  .then(data => {
      this.setState({mbrs:data});
  });

}
```

Hasil yang di harapkan :
Menampilkan data pada halaman daftar detail Monber dan memunculkan total keseluruhan data yang akan ditindak lanjuti.

9

```
// Atur Status
handleUpdate = () => {

  const body = this.state;

  fetch(`${process.env.REACT_APP_API  }monber/ubahdebitur`, {
    method: 'POST',
    headers: {
      'Content-Type':'application/json'
    },
    body: JSON.stringify({
      idmonber: body.idmonber,
      id: body.idData,
      status: body.status
    })
  })
  .then(response => response.json())
  .then(() => {
    notification.success({
      message: 'Perubahan Berhasil',
      description: `Data Berhasil Diperbaharui!`,
    })
    this.refreshList();
  });

  this.handleReset();
}
```

Hasil yang di harapkan :
Menampilkan halaman atur edit pada daftar Monber

10

```
// Update Data Monber
handleUpdate = () => {

  const body = this.state;

  fetch(`${process.env.REACT_APP_API  }monber/ubahmonber`, {
    method: 'POST',
    headers: {
      'Content-Type':'application/json'
    },
    body: JSON.stringify({
      id: body.idmonber,
      deskripsi: body.deskripsi,
      tampil_kolom: body.columnChecked.toString()
    })
  })
  .then(response => response.json())
  .then(() => {
    notification.success({
      message: 'Perubahan Berhasil',
      description: `Deskripsi ${body.kodeMbr} Berhasil Diperbaharui!`,
    })
    this.refreshList();
  });

  this.handleReset();
}
```

```
// Algoritma Utama
const defaultColumn = [
  {
    title: 'PRIORITAS',
    // dataIndex: 'Status_Monber',
    key: 'Priotas',
    fixed: 'right',
    width: 150,
    render: props => {
      let totalColumnEmpty = 0;
      for (let index = 0; index < columns.length; index++) {
        if(!props[columns[index]]) {
          totalColumnEmpty += 1;
        }
      }

      let colorPriority = 'orange';
      let textPriority = 'INCOMPLETED';
      if (totalColumnEmpty === 0) {
        colorPriority = 'blue';
        textPriority = 'COMPLETED';
      }

      return (
        <>
          <span className="font-size-22 font-weight-bold">
            <Tag color={colorPriority}>{textPriority}</Tag>
          </span>
        </>
      )

    }
  },
```

Hasil yang di harapkan :
Menampilkan pop up atur status pada halaman detail daftar rekening Monber

11

```
const GrafikAudit = () => {

  const [chartData, setChartData] = useState({});

  const chart = () => {
    const mbrLabel = [];
    const mbrDone = [];
    const mbrInvalid = [];

    // get data grafik Monber
    fetch(`${process.env.REACT_APP_API  }monber/daftarmonber`)
      .then(response => response.json())
      .then(data => {
          for (const dataObj of data)
          {
            mbrLabel.push(dataObj.kode_monber.replace('MONBER','MBR').replace('_',''));
            mbrDone.push(dataObj.total_done);
            mbrInvalid.push(dataObj.total_invalid);
          }

          setChartData({
            labels: mbrLabel,
            datasets: [
              {
                label: 'Done',
                backgroundColor: 'rgba(41, 223, 153, 1)',
                borderColor: 'rgba(41, 223, 153, 1)',
                borderWidth: 1,
                data: mbrDone,
                barPercentage: 0.7,
                order: 1
              },
              {
                label: 'Invalid',
                backgroundColor: 'rgba(239, 80, 80, 1)',
                borderColor: 'rgba(239, 80, 80, 1)',
                borderWidth: 1,
                data: mbrInvalid,
                barPercentage: 0.7,
                order: 2
              }
            ],
          });
      });
  }
```

Hasil yang di harapkan :
Menampilkan seluruh data untuk rekapitulasi Monber