

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sumber energi listrik sangat penting untuk perkembangan teknologi dan industri kontemporer. Dengan ketersediaan energi listrik yang stabil, berbagai industri dapat beroperasi secara efisien dan berkelanjutan. Selain itu, energi listrik memiliki sifat fleksibel dan efisien karena dapat diubah menjadi berbagai bentuk energi lain dengan tingkat kerugian yang relatif kecil. Oleh karena itu, jumlah energi listrik yang diperlukan terus meningkat setiap tahun seiring dengan pertumbuhan populasi dan aktivitas ekonomi (Saadat, 2010).

Meningkatnya kebutuhan akan mengharuskan sistem tenaga listrik dirancang dan dioperasikan dengan keandalan yang tinggi. Untuk mencapai hal ini, analisis yang tepat diperlukan agar operator dapat memahami secara menyeluruh kondisi sistem. Studi aliran daya adalah salah satu analisis yang paling penting karena memberikan gambaran tentang bagaimana jaringan beroperasi dalam kondisi normal. Alat analisis ini memungkinkan operator dan perencana untuk memprediksi efek gangguan atau perubahan beban terhadap kinerja sistem (Grainger & Stevenson, 1994).

Studi aliran daya menghitung tegangan pada setiap bus, aliran daya aktif dan reaktif, dan rugi-rugi daya jaringan. Informasi ini sangat penting untuk menjamin sistem berada pada batas operasi yang aman dan efektif. Studi aliran daya juga sering digunakan sebagai dasar untuk memperluas sistem tenaga. Ini termasuk menambah saluran transmisi, transformator, dan pembangkit. Analisis ini memperkecil kemungkinan terjadinya gangguan dan ketidakstabilan sistem (Glover, Sarma, & Overbye, 2012).

Kebutuhan komputasi yang semakin kompleks menuntut kemajuan dalam teknik penyelesaian aliran daya. Tiga metode yang paling umum digunakan adalah *Gauss-Seidel*, *Newton-Raphson*, dan *Fast Decoupled*. Masing-masing metode memiliki karakteristik yang berbeda, sehingga pilihan yang tepat sangat bergantung pada ukuran sistem dan tingkat akurasi yang dibutuhkannya. Berbagai metode ini memungkinkan analisis dengan memilih pendekatan yang paling cocok untuk kondisi jaringan (Hasibuan et al., 2020).

Metode *Newton Raphson* dianggap memiliki tingkat konvergensi tercepat dibandingkan dengan metode lainnya. Metode ini digunakan untuk menyelesaikan sistem persamaan nonlinear dengan menggunakan metode numerik berbasis turunan parsial. Meskipun membutuhkan komputasi matriks yang lebih kompleks, keuntungan terletak pada ketepatan hasil dalam sistem berskala besar. Karakteristik ini membuat metode ini menjadi pilihan utama untuk banyak program analisis sistem tenaga (Harun dan Yusuf, 2012).

Sebaliknya, proses iterasi yang lebih cepat dapat dicapai dengan metode *Fast Decoupled Load flow* (FDLF). Metode ini menggunakan sifat decoupling daya aktif dan reaktif terhadap sudut tegangan. Proses perhitungan dapat dilakukan lebih mudah tanpa mengurangi akurasi dengan menyederhanakan model matriksnya. Hal ini membuat metode FDLF ideal untuk digunakan dalam analisis jaringan besar yang membutuhkan efisiensi tinggi (Isa, 2007).

Meningkatnya kompleksitas sistem tenaga listrik membuat penggunaan perangkat lunak dalam penyelesaian metode aliran daya sangat penting. MATLAB adalah salah satu perangkat lunak yang paling populer di industri dan dunia akademik. Ini karena kemampuan untuk mengolah data numerik, memvisualisasikan hasil, dan menyediakan library fungsi yang sangat lengkap. MATLAB juga memungkinkan simulasi sistem tenaga yang fleksibel, cepat, dan akurat untuk memenuhi kebutuhan penelitian (Shertukde, 2019).

MATLAB memiliki GUI (Graphical User Interface) yang membuat program lebih mudah digunakan. Dengan GUI, pengguna tidak perlu memahami sintaks pemrograman untuk menjalankan algoritma dengan menekan tombol atau memasukkan data melalui form. Hal ini sangat membantu pengguna yang baru mengenal perintah berbasis teks. Antarmuka visual memungkinkan simulasi dan pembelajaran dilakukan dengan lebih cepat dan lebih mudah dipahami (Widiarsono, 2005).

App Designer tersedia dalam MATLAB untuk pengembangan GUI, yang memungkinkan pengguna membuat antarmuka dengan fitur *drag-and-drop*. Desain ini memudahkan mengintegrasikan elemen visual dengan logika pemrograman yang mengontrol proses komputasi. *App Designer* juga memungkinkan visualisasi data secara real-time, sehingga hasil analisis aliran daya dapat ditampilkan langsung pada grafik atau tabel. Dengan demikian, pengguna dapat lebih mudah memahami hasil simulasi (Isa, 2007).

Pengembangan GUI MATLAB untuk analisis aliran daya adalah langkah yang sangat relevan mengingat kebutuhan akan sistem analisis yang mudah digunakan, interaktif, dan tetap akurat. Jika metode Newton-Raphson, Gauss-Seidel, dan *Fast Decoupled* digabungkan dalam satu aplikasi, ini akan memberi pengguna fleksibilitas untuk memilih metode perhitungan yang sesuai. Diharapkan aplikasi ini akan membantu pengguna memahami metode aliran daya secara menyeluruh. Oleh karena itu, sangat bermanfaat untuk mendukung riset, industri, pendidikan dan pengembangan sistem tenaga listrik kontemporer.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, penelitian ini berupaya menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana menggunakan *App Designer* untuk menerapkan studi aliran daya pada aplikasi MATLAB?
2. Bagaimana hasil implementasi studi aliran daya menggunakan aplikasi GUI MATLAB?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut:

1. Membuat perhitungan studi aliran daya pada aplikasi MATLAB menggunakan *App Designer*.
2. Membuat aplikasi menggunakan fitur GUI untuk perhitungan aliran daya metode Newton-Raphson, *Gauss Seidel* dan Fast-Decoupled.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini, yaitu:

1. Mahasiswa dapat memperoleh pemahaman yang lebih baik tentang penggunaan studi aliran daya dan menggunakan temuan penelitian ini sebagai referensi.
2. Pemerintah dan masyarakat dapat menggunakan temuan tugas akhir ini untuk membantu merawat dan memperluas bidang kelistrikan.
3. Bidang akademik dapat memperoleh pemahaman yang lebih baik tentang GUI (*Graphical User Interface*).

1.5 Batasan Masalah

1. Tugas akhir ini menggunakan aplikasi MATLAB sebagai pengaplikasian studi aliran daya dengan fitur *App Designer*.
2. Menggunakan studi aliran daya *Newton Raphson*, *Gauss Seidel* dan *Fast Decoupled*.

1.6 Kajian Pustaka

1.6.1 Studi Aliran Daya

Analisis aliran daya mengukur distribusi daya aktif dan reaktif dalam sistem tenaga listrik pada kondisi steady-state. Tujuan utama dari analisis ini adalah untuk menghitung tegangan pada setiap bus sistem dan aliran daya pada saluran transmisi. Dalam sistem tenaga listrik, aliran daya diperlukan untuk mengevaluasi bagaimana daya disalurkan dari pembangkit ke beban melalui saluran transmisi. Analisis aliran daya sangat penting untuk merencanakan operasi dan pengembangan sistem tenaga, baik untuk kondisi saat ini maupun proyeksi masa depan. Operator sistem dapat memonitor dan mengontrol distribusi daya untuk memastikan efisiensi dan kestabilan sistem (Glover, Sarma, & Overbye, 2012).

Dalam studi aliran daya, kita menghitung tegangan dan sudut fasa setiap bus, serta daya aktif dan reaktif yang mengalir melalui saluran. Ini dilakukan untuk menghindari gangguan atau kerusakan pada peralatan. Hasil kalkulasi menunjukkan bagaimana sistem akan bertindak dalam kondisi operasional tertentu dan apakah sistem dapat menyalurkan daya dengan aman dan efisien (Saadat, 2010).

Persamaan non-linear yang muncul dalam aliran daya dimodelkan oleh metode *Newton Raphson*, yang banyak digunakan karena tingkat konvergensi yang cepat. Persamaan non-linear ini mewakili hubungan antara tegangan, arus, dan daya dalam sistem tenaga. Selain itu, tergantung pada karakteristik sistem dan ukurannya, metode *Gauss Seidel* dan *Fast Decoupled* juga digunakan. Setiap metode memiliki kelebihan dan kelemahan dalam hal efisiensi komputasi dan akurasi. Ini bervariasi tergantung pada kompleksitas sistem yang dianalisis (Kothari & Nagrath, 2008).

Sistem per-unit digunakan untuk menyederhanakan analisis sistem tenaga yang memiliki berbagai tingkat tegangan. Dalam sistem ini, besaran listrik seperti tegangan, arus, daya, dan impedansi diwakili sebagai pecahan dari nilai dasar yang telah ditetapkan. Penggunaan sistem per-unit menghilangkan kebutuhan untuk mengubah impedansi dari satu sisi transformator ke sisi lainnya, yang sangat menyederhanakan perhitungan dalam jaringan yang kompleks (Saadat, 2010).

Analisis aliran daya penting untuk evaluasi sistem saat ini, tetapi juga untuk pengelolaan sistem tenaga yang lebih baik dalam jangka panjang. Data bus (tipe bus, beban P dan Q, pembangkitan P, dan spesifikasi tegangan), data saluran transmisi (resistansi, reaktansi, dan suseptansi), dan data transformator (rating dan impedansi) adalah data utama yang diperlukan untuk studi aliran daya. Aliran daya aktif (MW) dan reaktif (MVAR) pada setiap saluran, magnitudo dan sudut fasa tegangan pada setiap bus, dan total rugi-rugi daya pada sistem secara keseluruhan adalah hasil atau output dari penelitian ini (Glover, Sarma, & Overbye, 2012).

Dalam studi aliran teardapat beberapa bus, yaitu:

1. Bus referensi (*slack bus* atau *swing bus*)

Bus referensi atau *slack bus* digunakan untuk menyeimbangkan daya yang tidak dapat dihitung dari bus lainnya. Besaran tegangan (V) dan sudut fase tegangan (δ) ditentukan dan tetap pada bus ini. *Slack bus* biasanya digunakan untuk menyediakan daya aktif dan reaktif yang diperlukan untuk menjaga keseimbangan antara daya total yang dikonsumsi oleh beban dan daya yang dihasilkan oleh generator dalam sistem. Ini karena bus yang terhubung dengan pembangkit listrik terbesar dalam sistem mampu menyuplai daya lebih banyak dengan variasi daya yang minimal.

2. Generator bus (PV bus)

Bus yang digunakan untuk menghubungkan pembangkit listrik disebut bus generator atau PV bus. Bus ini memiliki dua parameter yang diketahui: daya aktif (P) dan tegangan magnitude (V). Namun, dua parameter tambahan, daya reaktif (Q) dan sudut fase tegangan (δ), perlu dihitung. Bus generator melakukan dua fungsi utama: menghasilkan daya aktif dan menjaga tegangan

pada tingkat yang stabil. Pembangkit listrik bus ini berusaha mengontrol tegangan pada sistem sesuai dengan kapasitas pembangkit.

3. Bus pembebanan (*Load bus*)

Sebelum melakukan analisis aliran daya, komponen sistem yang terdiri dari pembangkit (generator), transformator, saluran transmisi, dan beban harus digambarkan atau dimodelkan dalam diagram satu garis. Dalam hal ini, sistem tiga fasa harus dalam keadaan seimbang. Manfaat dari diagram ini adalah untuk memberikan gambaran singkat tentang sistem tenaga listrik secara keseluruhan (Supriyadi, 2016).

Bus beban atau bus PQ adalah bus yang menunjukkan lokasi beban dalam jaringan di mana daya aktif (P) dan daya reaktif (Q) diketahui. Hanya ada dua parameter yang diketahui pada bus ini, yaitu daya aktif dan daya reaktif, tegangan dan sudut fase tegangan, yang harus dihitung selama analisis aliran daya. Bus beban memainkan peran penting dalam menunjukkan bagaimana daya dikonsumsi dalam sistem, yang berdampak pada aliran daya dan daya yang digunakan sistem tenaga bernilai negatif, sedangkan daya yang ditambahkan bernilai positif. Bus ini hanya terhubung dengan beban, tanpa generator (Supriyadi, 2016).

Menurut Stevenson (1996), data-data yang dibutuhkan untuk analisis aliran daya adalah sebagai berikut :

- a. Data pembangkit, yaitu kapasitas daya aktif (P) dan reaktif (Q) dalam satuan Megawatt (MW), tegangan terminal (V) dalam satuan Kilovolt (KV), dan reaktansi sinkron (X) dalam satuan Ohm (Ω).
- b. Data beban, yaitu daya aktif (P) dalam Megawatt (MW) dan daya reaktif (Q) dalam satuan Megavolt Ampere (MVA).
- c. Data transformator daya, termasuk kapasitas tiap trafo dalam satuan Megavolt Ampere (MVA), yaitu tegangan (V) dalam satuan Kilovolt (KV), dan reaktansi bocor (X) dalam satuan Ohm.
- d. Data saluran transmisi, yaitu resistansi (R) dalam ohm (Ω) dan reaktansi (X) dalam ohm (Ω).

Analisis aliran daya bertujuan untuk memperoleh kondisi mula perencanaan sistem yang baru dan memperoleh kondisi awal untuk studi-studi selanjutnya seperti studi selanjutnya seperti studi hubung singkat, stabilitas dan pembebanan ekonomis. Mengetahui tegangan-tegangan pada setiap bus yang ada dalam sistem, baik magnitude maupun sudut fasa. Daya aktif dan daya reaktif yang mengalir dalam setiap saluran yang ada dalam sistem dapat diketahui dengan melakukan analisis aliran daya, bahkan memungkinkan untuk mengetahui kondisi semua peralatan, apakah memenuhi batas batas yang ditentukan untuk menyalurkan daya listrik yang diinginkan (Supriyadi, 2016).

1.6.2 Aliran daya metode *Newton Raphson*

Metode *Newton Raphson* sering digunakan dan dijelaskan dalam buku teks besar. Metode ini sering digunakan untuk menganalisis sistem aljabar non-linier

secara bersamaan. Metode *Newton Raphson* adalah pendekatan berurutan yang didasarkan pada perkiraan awal satu dimensi yang diberikan oleh deret Taylor. Metode *Newton Raphson*, yang menggunakan matriks admitansi dalam ekspansi deret Taylor pertama atau kedua, telah dievaluasi sebagai solusi terbaik untuk keandalan dan konvergensi yang cepat (Isa, 2007).

$$f(x) = 0 \quad (1)$$

Untuk menyelesaikan persamaan ini, pilih inisial dari nilai x^0 . Perbedaannya adalah nilai inisial dan hasil akhir akan menjadi Δx^0 . Kemudian $x = x^0 + \Delta x^0$ adalah solusi dari persamaan non linear dari persamaan (1), adalah

$$f(x(0) + \Delta x(0)) = 0 \quad (2)$$

Memperluas persamaan (2) dengan deret taylor, maka akan mendapatkan

$$f(x^{(0)}) + \Delta x^0 = f(x^0) + f'(x^0)\Delta x^0 + f''(x^0)\frac{(\Delta x^0)^2}{2!} + \dots + f^{(n)}(x^0)\frac{(\Delta x^0)^n}{n!} + \dots = 0 \quad (3)$$

Dimana $f'(x^0), \dots, f^{(n)}(x^0)$ adalah turunan dari fungsi $f(x)$

Jika perbedaan $\Delta x^{(0)}$ sangat kecil (artinya adalah nilai inisial x^0 mendekati solusi dari fungsi), dengan syarat yang kedua dan lebih tinggi persamaannya dapat diabaikan. Dengan demikian persamaan (3) menjadi persamaan yang linear seperti pada persamaan (4)

$$f(x^0 + \Delta x^0) = f(x^0) + f'(x^0)\Delta x^0 = 0 \quad (4)$$

Kemudian akan mendapatkan

$$\Delta x^0 = -\frac{f(x^0)}{f'(x^0)} \quad (5)$$

Maka solusi baru yang didapatkan adalah

$$x^1 = x^0 + \Delta x^0 = x^0 - \frac{f(x^0)}{f'(x^0)} \quad (6)$$

Sejak persamaan (4) adalah persamaan yang dapat diperkirakan, nilai dari Δx^0 juga dapat diperkirakan. Dengan demikian solusi dari nilai x itu bukan nilai asli. Maka diperlukan iterasi lebih lanjut. Persamaan iterasi adalah sebagai berikut

$$x^{k+1} = x^k + \Delta x^k = x^k - \frac{f(x^k)}{f'(x^k)} \quad (7)$$

Iterasi dapat dihentikan ketika mendapatkan kondisi seperti pada persamaan

(8) dan (9)

$$|\Delta x^k| < \epsilon_1 \quad (8)$$

atau

$$|f(x^k)| < \epsilon_2 \quad (9)$$

Dimana ϵ_1 , ϵ_2 , merupakan konvergensi yang diizinkan, adalah bilangan positif yang kecil.

1.6.3 Aliran daya metode Gauss Seidel

Dalam analisis aliran daya pada sistem tenaga listrik, metode *Gauss Seidel* adalah salah satu metode iteratif yang banyak digunakan untuk menyelesaikan persamaan non-linear. Karena metode ini dikenal sebagai metode perpindahan berturut-turut, nilai variabel yang dihitung langsung digunakan untuk iterasi berikutnya. Metode ini bergantung pada persamaan yang telah disusun untuk memperbarui variabel satu per satu secara berurutan. Anggap persamaan non-linear berikut yang akan diselesaikan untuk menunjukkan bagaimana metode ini dapat digunakan:

$$F(x) = 0 \quad (10)$$

Selanjutnya, persamaan ini disusun ulang menjadi bentuk yang lebih sederhana sebagai:

$$x = g(x) \quad (11)$$

Dalam kasus ini, fungsi $g(x)$ berfungsi untuk menggantikan persamaan $F(x)$ dan memberikan pembaruan nilai (x) pada iterasi berikutnya. Jika $x^{(k)}$ adalah perkiraan awal variabel (x) , maka urutan iterasi dapat dibentuk dengan persamaan berikut:

$$X^{(k+1)} = g(x^{(k)}) \quad (12)$$

Di mana nilai $x^{(k)}$ adalah nilai dari iterasi ke- k dan nilai $X^{(k+1)}$ adalah nilai dari iterasi berikutnya. Proses iteratif ini akan berlanjut, dengan nilai x yang lebih baik diperoleh pada setiap langkahnya.

Ketika perbedaan antara nilai mutlak iterasi berturut-turut menjadi sangat kecil, proses konvergensi telah tercapai, dan solusi atau hasil yang tepat dihasilkan. Sebagai contoh, kondisi konvergensi dapat dituliskan sebagai berikut:

$$|x^{(k+1)} - x^{(k)}| \leq \epsilon \quad (13)$$

di mana ϵ adalah batas akurasi atau toleransi yang diinginkan. Nilai yang lebih kecil dari ϵ menunjukkan tingkat akurasi yang lebih tinggi, tetapi nilai yang lebih besar juga menunjukkan jumlah iterasi yang diperlukan untuk mencapai konvergensi (Isa, 2007).

Meskipun metode *Gauss Seidel* mudah digunakan, kelemahannya terletak pada kecepatan konvergensinya, terutama pada sistem yang besar atau ketika kondisi awal sangat jauh dari solusi yang sebenarnya. Dibandingkan dengan metode lain, seperti *Newton Raphson*, metode *Gauss Seidel* cenderung membutuhkan lebih banyak iterasi untuk mencapai solusi yang diinginkan dan tidak selalu menjamin konvergensi dalam beberapa situasi, terutama pada sistem dengan rasio impedansi yang besar. Namun, metode ini tetap relevan dan digunakan secara luas dalam berbagai aplikasi sistem tenaga listrik, terutama untuk analisis dasar dan studi kecil hingga menengah. Ini tetap menjadi pilihan yang baik untuk analisis aliran daya pada sistem tenaga dengan jumlah bus yang relatif kecil atau ketika aplikasi membutuhkan kecepatan implementasi dan penggunaan memori yang efisien.

1.6.4 Aliran daya metode *Fast Decoupled*

Untuk analisis aliran daya dalam sistem tenaga listrik, terutama sistem besar, metode *Fast Decoupled* (FD) sangat efektif. Metode ini diciptakan untuk mengatasi masalah komputasi yang kompleks yang biasanya muncul dalam analisis aliran daya dengan metode lain, seperti *Newton Raphson* dan *Gauss Seidel*. Metode *Newton Raphson* disederhanakan menjadi metode *Fast Decoupled*. Metode ini mengabaikan beberapa elemen matriks Jacobian, memisahkan daya aktif dan reaktif. Ini mempercepat proses konvergensi dengan menghasilkan dua matriks yang lebih sederhana (Isa, 2007).

Untuk sistem daya berskala besar, saluran transmisi biasanya memiliki rasio X/R yang sangat tinggi. Demikian pula, perubahan daya nyata ΔP kurang sensitif terhadap perubahan besar tegangan dan paling sensitif terhadap perubahan sudut fase. Daya reaktif juga kurang sensitif terhadap perubahan besar tegangan dan sudut fase (Isa, 2007).

Tingkat perubahan kompleksitas komputasi matriks lebih besar daripada tingkat perubahan ukuran matriks. Ini menunjukkan bahwa mengurangi kompleksitas komputasi secara signifikan dapat dilakukan dengan mengurangi ukuran matriks yang digunakan dalam perhitungan. Metode *Fast Decoupled* adalah salah satu metode yang paling terkenal yang memanfaatkan karakteristik ini, menggunakan hubungan yang kuat antara daya aktif dan sudut fasa, serta antara daya reaktif dan tegangan dalam sistem tenaga. Perhitungan ΔP dan ΔQ dalam metode N-R dilakukan seperti yang ditunjukkan pada

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \theta} & \frac{\partial Q}{\partial V} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \quad (14)$$

Namun dengan menetapkan

$$\frac{\partial P}{\partial V} = 0, \frac{\partial Q}{\partial \theta} = 0 \quad (15)$$

Persamaan (15) menjadi

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \theta} & 0 \\ 0 & \frac{\partial Q}{\partial V} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \quad (16)$$

Modifikasi ini memungkinkan pengurangan besar dalam kompleksitas komputasi dan dengan demikian meningkatkan efisiensi komputasi (Ochi et al, 2013).

1.6.5 MATLAB

MATLAB, juga dikenal sebagai MATrix LABoratory, adalah lingkungan komputasi numerik dan bahasa pemrograman berkualitas tinggi yang dikembangkan oleh MathWorks. MATLAB, yang awalnya dibuat untuk mempermudah manipulasi aljabar linear dan matriks, kemudian berkembang menjadi alat yang sangat fleksibel yang digunakan di banyak bidang, termasuk teknik, fisika, keuangan, ilmu data, dan kecerdasan buatan. Keunggulan utamanya adalah kemudahan penggunaan untuk analisis data, visualisasi, dan pengembangan algoritma (Houcque, 2005).

MATLAB memiliki ekosistem toolbox yang luas, yang merupakan kumpulan fungsi khusus yang dikhususkan untuk bidang aplikasi tertentu. Misalnya, dalam bidang teknik elektro, ada Simscape Electrical (sebelumnya dikenal sebagai *Sim Power Systems*), yang memungkinkan simulasi sistem tenaga listrik yang kompleks. Selain itu, karena diintegrasikan dengan MATLAB yang berbasis matriks, implementasi algoritma numerik, seperti metode *Newton Raphson* untuk studi aliran daya, sangat mudah (Kaur & Bala, 2023).

Lebih lanjut, MATLAB menyediakan kapabilitas visualisasi data yang sangat kuat. Pengguna dapat dengan mudah membuat berbagai jenis plot dari grafik 2D, 3D hingga animasi dan antarmuka grafis interaktif. Kemampuan ini yang memungkinkan representasi visual dari data numerik kompleks misalnya profil tegangan bus atau aliran daya pada saluran yang sangat penting untuk interpretasi hasil dan pengambilan keputusan dalam analisis sistem tenaga. Fitur-inilah yang dimanfaatkan secara ekstensif dalam pembuatan GUI dalam penelitian ini (Mathworks, 1996).

1.6.6 GUI (Graphical User Interface)

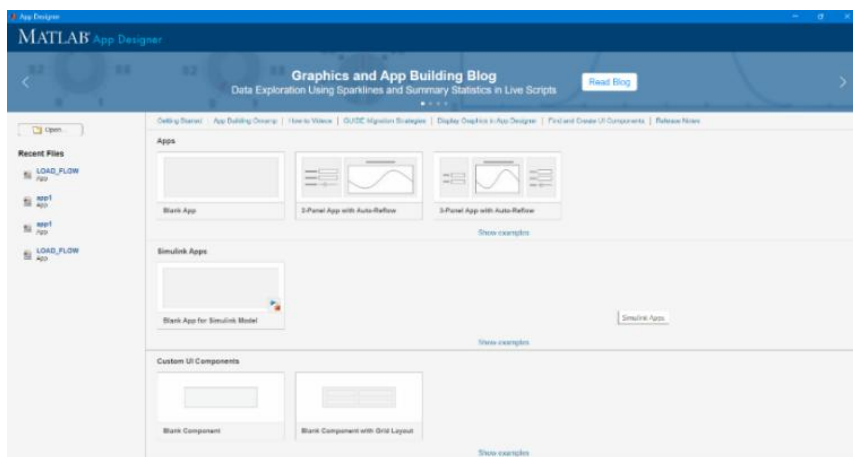
Grafik antarmuka pengguna (GUI) adalah antarmuka pengguna yang memudahkan interaksi antara manusia dan komputer dengan menggunakan elemen visual seperti ikon, menu, tombol, dan jendela. Berbeda dengan antarmuka berbasis perintah tertulis (Command-Line Interface/CLI), GUI menggunakan representasi grafis yang

mudah dipahami sehingga pengguna dapat menjalankan program, mengelola *file*, atau mengakses fitur sistem dengan klik, geser, atau sentuh (Sinaga, Pratama, & Siburian, 2021).

Kesuksesan GUI sangat bergantung pada penerapan desain antarmuka yang berpusat pada pengguna (*user-centered design*), di mana konsistensi menjadi prinsip utama: elemen dan tindakan yang sama menghasilkan hasil yang serupa di seluruh bagian aplikasi, sehingga mengurangi beban kognitif pengguna. Selain itu, umpan balik, juga dikenal sebagai *feedback*, adalah ketika sistem memberikan respons kepada tindakan pengguna, seperti pesan status, bilah kemajuan, atau perubahan kursor, sehingga pengguna dapat memahami apa yang sedang terjadi. Desain yang baik juga harus mengutamakan *discoverability*, yang berarti bahwa pengguna baru dapat menemukan fitur dan fungsinya dengan mudah tanpa belajar banyak (Ejaz dkk., 2019).

Paradigma pemrograman berbasis kejadian juga dikenal sebagai pemrograman berbasis kejadian memungkinkan alur eksekusi program dipicu oleh berbagai kejadian yang tidak dapat diprediksi, seperti klik tombol, *input* dari *keyboard*, atau gerakan mouse. Ini adalah karakteristik antarmuka pengguna grafis (GUI). Aplikasi GUI akan tetap berada dalam keadaan menunggu hingga sebuah peristiwa (*event*) terdeteksi dalam model ini. Aplikasi akan memanggil *event handler* atau *callback function* yang telah diprogram untuk menangani kejadian ketika terjadi. Metode ini membuat eksekusi aplikasi lebih fleksibel dan interaktif, dengan setiap tindakan dikendalikan oleh peristiwa yang terjadi pada antarmuka pengguna, daripada mengikuti urutan linier tetap seperti dalam pemrograman *procedural* (Rachmat, 2016).

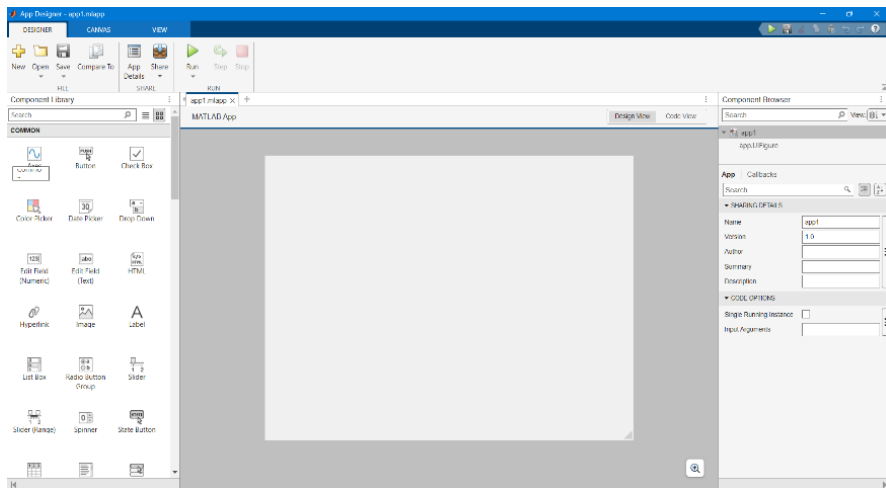
Tampilan awal *App Designer* saat pertama kali digunakan.



Gambar 1. Tampilan *App Designer*

Gambar 1 menunjukkan beberapa pilihan yang dapat dipilih oleh pengguna untuk memulai dari halaman yang telah terisi atau kosong. Untuk tujuan memudahkan, ada juga opsi untuk membuka *file* yang telah dibuat sebelumnya.

Setelah memilih lembar kosong, pengguna akan melihat tampilan yang mirip dengan Gambar 2.



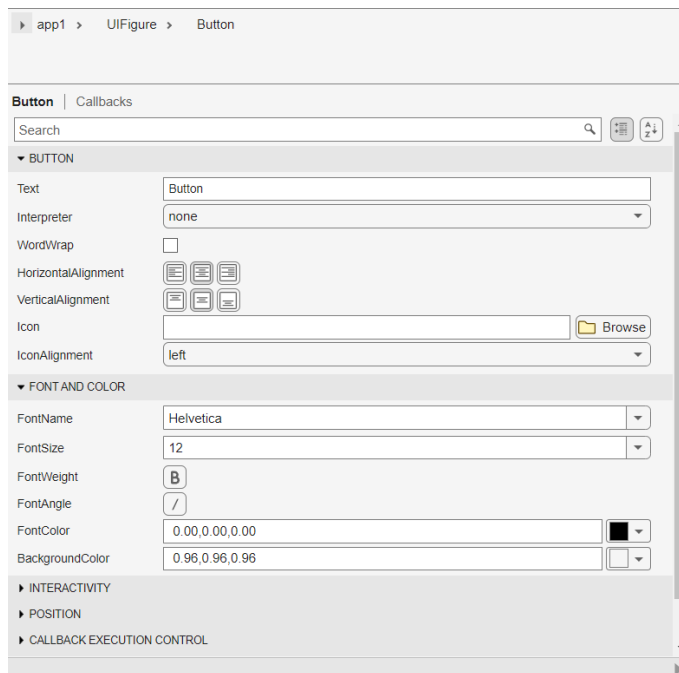
Gambar 2. Tampilan *Homepage*

Gambar 2. menunjukkan tampilan *homepage App Designer* yang memiliki berbagai fungsi seperti yang di perlihatkan pada Tabel 1:

Tabel 1. Fitur GUI MATLAB

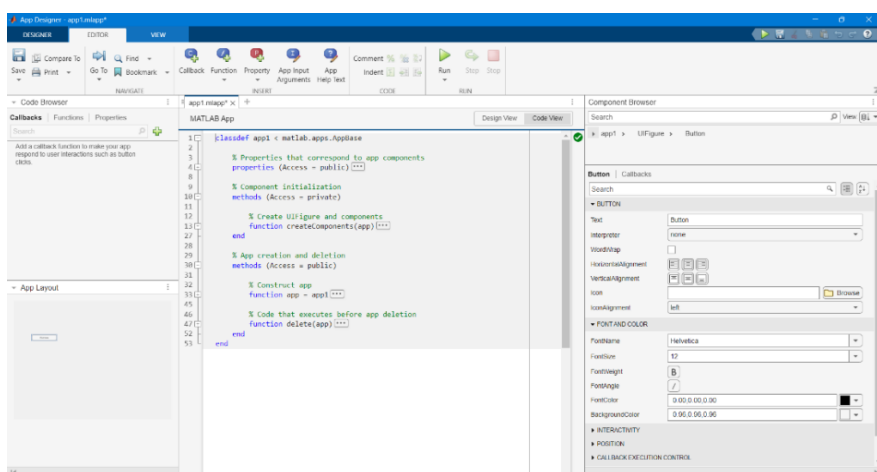
No.	Fitur	Gambar	Fungsi
1.	<i>Button</i>		Fungsi tombol dapat ditambahkan. Jika pengguna menekannya, hal-hal akan terjadi sesuai dengan perintah atau <i>event</i> yang telah diprogramkan.
2.	<i>Axes</i>		Axes digunakan untuk menampilkan berbagai jenis data dalam bentuk grafik.
3.	<i>Label</i>		Label memungkinkan penambahan teks di area yang dipilih.
4.	Tabel		Tabel dapat mengolah atau menampilkan data dengan rapih.
5.	<i>Image</i>		Dengan menggunakan <i>Image</i> , memudahkan memasukkan gambar ke dalam desain yang akan dibuat.
6.	<i>Run</i>		Pemrograman dan desain yang telah dibuat dapat dijalankan dengan menggunakan <i>Run</i> .

Adapun fungsi yang dapat membuat penanda pada suatu properti dan juga dapat membuat variasi ukuran tulisan, warna, dan lain lain seperti pada Gambar 3.



Gambar 3. Tampilan Fungsi *Button*

Setelah itu, Anda dapat menambahkan kode ke properti melalui tampilan, yang memungkinkan properti GUI bekerja seperti yang ditunjukkan pada Gambar 4.



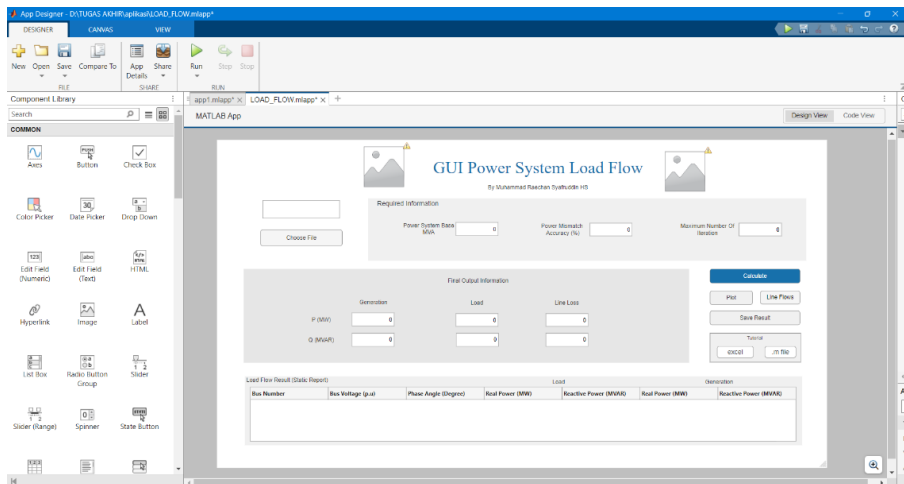
Gambar 4. Tampilan *property* pada GUI

Seperti yang ditunjukkan pada Gambar 4, *App Designer* menggunakan banyak bahasa pemrograman, seperti:

- *Properties* = menambahkan properti pada halaman
- *Function* = memberikan fungsi pada properti
- *End* = mengakhiri perintah

Gambar 5 menunjukkan contoh hasil desain dan pemrograman yang telah dibuat.

- Desain



Gambar 5. GUI Power System Load Flow

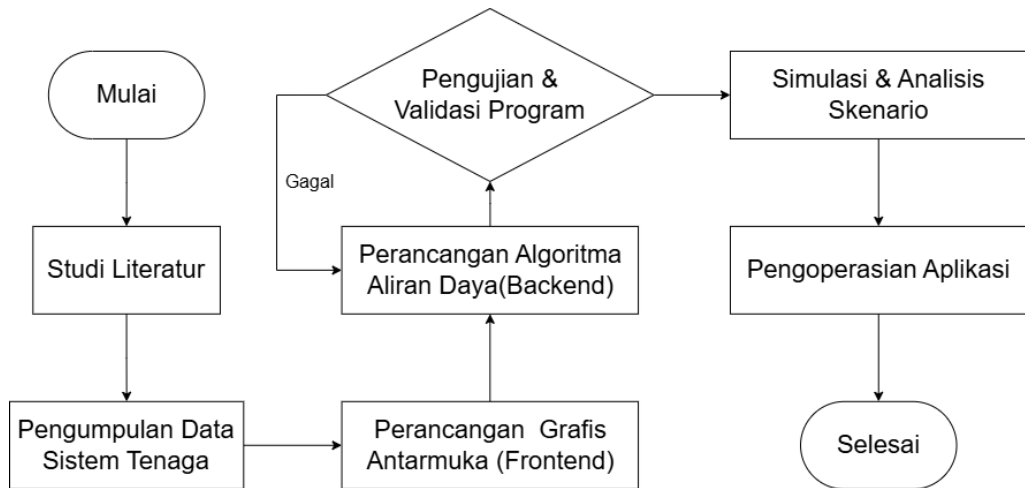
- Pemrograman

App Designer pada MATLAB menggunakan bahasa pemrograman untuk membuat antarmuka aplikasi berbasis GUI dan menghubungkan elemen GUI ke logika aplikasi yang berjalan di belakangnya. Kode MATLAB yang dibuat dalam aplikasi ini bertanggung jawab untuk menjalankan logika perhitungan seperti menghitung aliran daya dalam sistem tenaga listrik menggunakan metode *Newton Raphson*, *Gauss Seidel* atau *Fast Decoupled*. *App Designer* memungkinkan pemrograman untuk membuat fungsi *callback* yang menghubungkan tindakan pengguna, seperti mengklik tombol atau memilih *file*, dengan logika aplikasi. Misalnya, aplikasi akan memproses data dan menampilkan hasil perhitungan dalam tabel atau grafik ketika pengguna menekan tombol "*Calculate*". Selain itu, pemrograman juga dapat digunakan untuk berinteraksi dengan *file* eksternal (seperti *file* data.m atau.xlsx), membaca data, dan mengubahnya untuk digunakan untuk analisis.

Contoh fungsi dan pemrograman yang digunakan dalam *App Designer* untuk aplikasi MATLAB adalah komponen aplikasi (*UI Components*) dimana elemen-elemen visual yang dapat dilihat dan diinteraksikan oleh pengguna di antarmuka aplikasi disebut komponen antarmuka pengguna (UI). Komponen ini didefinisikan dalam kode aplikasi MATLAB menggunakan *App Designer* di bawah properti seperti (*Button*, *UI Figure Table*, dll).

BAB II METODE PENELITIAN

2.1 Diagram Alir Penelitian



Gambar 6. Diagram Alir Penelitian

Gambar 12 menyajikan kerangka alur penelitian, mulai dari tahap mulai studi literatur hingga proses pengoperasian aplikasi selesai. Penjelasan rinci dari setiap bagian pada gambar ialah:

1. Mulai, adalah langkah awal dari seluruh penelitian. Di sini, peneliti menetapkan topik penelitian, tujuan penelitian, dan gambaran umum tentang proses yang akan dilakukan.
2. Studi literatur, peneliti mempelajari teori, gagasan, dan penelitian terdahulu tentang aliran daya sistem tenaga, metode perhitungan, dan pengembangan aplikasi. Tujuan dari tahap ini adalah untuk membuat landasan teoritis yang kuat dan menentukan metode penelitian yang paling sesuai.
3. Pengumpulan data system tenaga, data yang dikumpulkan termasuk parameter sistem, data bus, dan saluran serta data pendukung lainnya yang diperlukan untuk proses perancangan, pengujian, dan simulasi aplikasi.
4. Perancangan Grafis Antarmuka (*Frontend*), Tahap ini berfokus pada perancangan tampilan aplikasi yang mudah digunakan. Informatif, sederhana, dan mampu menampilkan data input dan hasil analisis dengan jelas. Tahap ini sangat penting untuk memastikan bahwa aplikasi berjalan dengan baik, mudah, dan nyaman dioperasikan.
5. Perancangan algoritma aliran daya (*backend*), merancang logika utama sistem, yaitu algoritma untuk menghitung aliran daya. Proses backend mencakup pengolahan data masukan, menjalankan metode perhitungan, dan

menghasilkan keluaran berupa parameter sistem tenaga yang dianalisis. Fungsi aplikasi yang dikembangkan bergantung pada tahap *backend*.

6. Pengujian dan validasi program, program diuji untuk memastikan sistem dapat berfungsi sesuai rancangan. Untuk validasi, hasil program dibandingkan dengan perhitungan atau referensi yang digunakan. Tujuan dari tahap ini adalah untuk memastikan kestabilan, ketepatan, dan keandalan aplikasi.
7. Gagal, jika terdapat kesalahan, ketidaksesuaian hasil, atau program tidak berjalan sebagaimana mestinya selama tahap pengujian dan validasi, proses kembali ke tahap perancangan. Ini ditunjukkan pada diagram ini oleh alur kembali yang menuju ke tahap perancangan untuk melakukan perbaikan dan penyempurnaan hingga program memenuhi kriteria yang diharapkan.
8. Simulasi dan analisis skenario, aplikasi digunakan untuk menjalankan skenario sistem tenaga. Hasil simulasi kemudian dianalisis untuk mengetahui kinerja sistem, perubahan parameter, dan respon aplikasi bertindak terhadap berbagai kondisi.
9. Pengoperasian aplikasin, merupakan penerapan aplikasi yang telah dirancang, diuji, dan dianalisis. Aplikasi digunakan untuk kebutuhan penelitian, baik sebagai alat bantu analisis maupun sebagai cara yang lebih efisien untuk menguji sistem tenaga.
10. Selesai, menandai selesai seluruh proses penelitian.

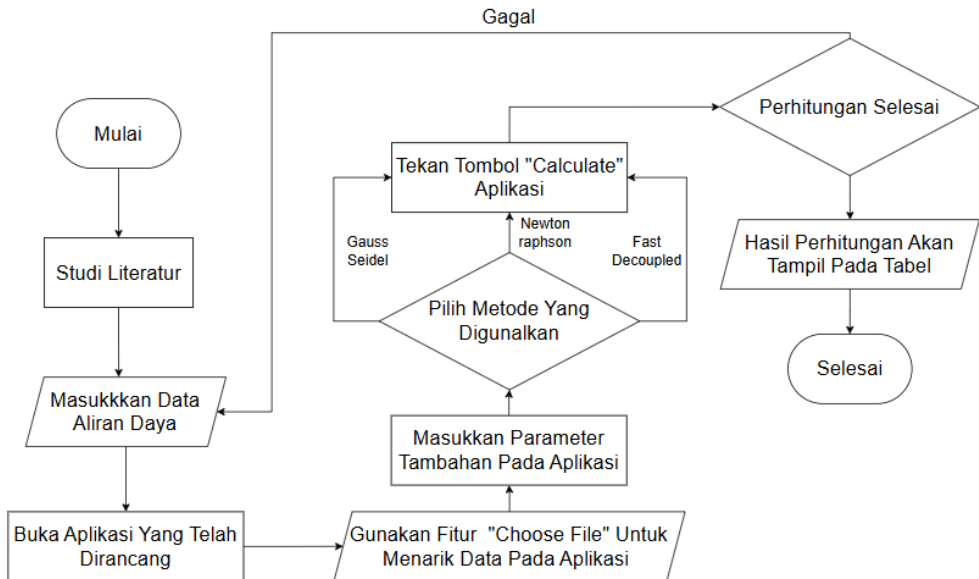
2.2 Waktu dan Lokasi Penelitian

Pelaksanaan penelitian ini dilakukan di Laboratorium Relay dan Pengukuran, Departemen Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin. Penelitian ini dilakukan pada bulan Oktober 2025 hingga April tahun 2026.

2.3 Peralatan dan Bahan Penelitian

Penelitian ini menggunakan software MATLAB yang memiliki fitur desain aplikasi, untuk membuat aplikasi berbasis GUI dengan menggunakan dua buku referensi data: *Power system Analysis* edisi 1994 oleh *Power system Analysis* edisi 1999 oleh Hadi Saadat.

2.4 Diagram Alir Penggunaan Aplikasi



Gambar 7. Diagram Alir Penggunaan Aplikasi

Gambar 13 menyajikan kerangka alur penelitian, mulai dari tahap mulai studi literatur hingga proses pengoperasian aplikasi selesai. Penjelasan rinci dari setiap bagian pada gambar ialah:

1. Mulai
Pada tahap ini, proses penelitian dan implementasi aplikasi dimulai. Peneliti memulai semua kegiatan penelitian sesuai dengan tujuan yang telah ditetapkan.
2. Studi Literatur
Proses ini merupakan proses untuk menciptakan landasan teoritis yang kuat melalui penelitian teori dan penelusuran referensi yang berkaitan dengan analisis aliran daya, sistem tenaga listrik, metode perhitungan seperti *Newton Raphson*, *Gauss Seidel*, dan *Fast Decoupled*, serta konsep perancangan aplikasi berbasis MATLAB.
3. Masukkan Data Aliran Daya
Setelah tahap studi literatur, peneliti menyiapkan dan memasukkan data sistem untuk digunakan dalam simulasi. Data yang dimasukkan termasuk data bus dan saluran, nilai impedansi, beban, pembangkitan, dan parameter analisis aliran daya lainnya.
4. Buka Aplikasi yang Telah Dirancang
Pada tahap ini aplikasi yang telah dikembangkan dibuka untuk memulai proses simulasi dan pengujian data sistem tenaga yang telah disiapkan.
5. Gunakan Fitur “Choose File” untuk Menarik Data pada Aplikasi

Tahap ini pengguna memilih fitur *Choose File* untuk mengambil *file* data yang telah disiapkan sebelumnya, baik dalam format *.m* maupun *.xlsx*, agar data dapat diproses oleh aplikasi.

6. Masukkan Parameter Tambahan pada Aplikasi

Pada tahap ini, pengguna mengisi parameter tambahan yang diperlukan untuk proses perhitungan, seperti base MVA, akurasi mismatch daya, dan maksimum jumlah iterasi. Parameter-parameter ini digunakan untuk mengontrol proses konvergensi metode numerik.

7. Pilih Metode yang Digunakan

Selanjutnya, pengguna memilih cara untuk melakukan analisis aliran daya. Ada tiga metode yang tersedia untuk aplikasi ini: *Gauss Seidel*, *Newton Raphson*, dan *Fast Decoupled*. Metode yang dipilih disesuaikan dengan kebutuhan analisis dan karakteristik sistem tenaga yang diuji.

8. Tekan Tombol “*Calculate*” pada Aplikasi

Setelah memilih metode, pengguna menekan tombol *Calculate* untuk memulai perhitungan aliran daya. Pada tahap ini, aplikasi mulai memproses data yang dimasukkan sesuai dengan metode yang telah dipilih sebelumnya.

9. Perhitungan Selesai

Pada tahap ini, aplikasi telah menyelesaikan proses komputasi. Sistem akan memeriksa apakah proses perhitungan telah berhasil mencapai kondisi konvergen.

10. Gagal

Apabila hasil perhitungan belum konvergen atau terjadi kesalahan input maupun proses komputasi, maka alur kembali ke tahap sebelumnya, yaitu proses *input* data atau pengaturan parameter, untuk dilakukan perbaikan.

11. Hasil Perhitungan Akan Tampil pada Tabel

Jika proses perhitungan berhasil, maka hasil analisis akan ditampilkan pada tabel *output* aplikasi. Hasil tersebut meliputi nilai tegangan tiap bus, sudut fasa, daya aktif, daya reaktif, serta parameter hasil lainnya.

12. Selesai

Tahap akhir menandakan bahwa seluruh proses simulasi dan analisis menggunakan aplikasi telah selesai dilakukan.