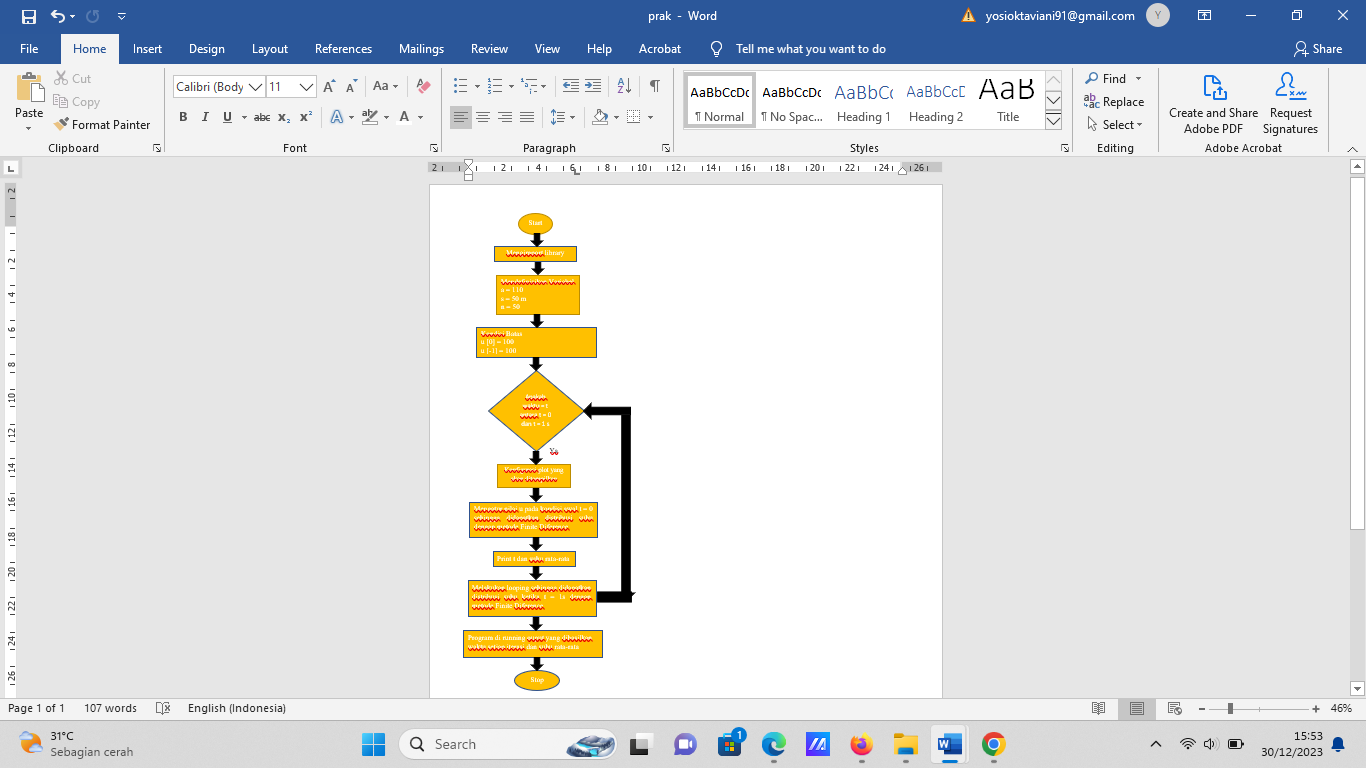
1. Buatlah Flow Chart yang menunjukkan algoritma pada Program Penyelesaian Persamaan Konduksi Panas pada 1 Dimensi dan 2 Dimensi
2. **Flow Chart 1 Dimensi**



Tahap pertama adalah import library numpy untuk melakukan operasi numerik dan matlotlib untuk membuat plot yang memvisualisasikan distribusi suhu seiring waktu atau posisi. Agar dapat diselesaikan secara komputasi, diperlukan beberapa atau mendefinisikan variable besaran yang diketahui diantaranya L = Panjang benda, t = waktu, α = Koefisien difusivitas termal, n = jumlah nude, dan kondisi awal.

Kemudian melakukan konfigurasi plot distribusi suhu.

* plt.subplots() digunakan untuk membuat sebuah gambar (figure) dan satu atau lebih sumbu (axes) di dalamnya. fig adalah objek gambar, dan ax adalah objek sumbu (atau array dari sumbu, tergantung pada jumlah subplot yang dibuat). Pemisahan ini memungkinkan pengaturan dan manipulasi terhadap gambar dan sumbu secara terpisah.
* ax.set\_xlabel("x (cm)") Mengatur label pada sumbu-x dengan teks "x (cm)". Ini memberikan keterangan pada sumbu-x untuk menunjukkan bahwa nilai-nilai pada sumbu tersebut diukur dalam satuan sentimeter.
* pcm = ax.pcolormesh([u], cmap=plt.cm.jet, vmin=0, vmax=100) pcolormesh digunakan untuk membuat visualisasi distribusi suhu sebagai peta warna. Argumen pertama, [u], adalah data yang akan divisualisasikan. cmap=plt.cm.jet menentukan peta warna yang digunakan (dalam hal ini, peta warna "jet"). vmin dan vmax mengatur nilai minimum dan maksimum untuk skala warna, dalam hal ini, dari 0 hingga 100.
* plt.colorbar(pcm, ax=ax) Menambahkan colorbar ke plot untuk memberikan referensi visual tentang nilai-nilai yang sesuai dengan warna-warna pada peta warna. pcm adalah objek mappable yang digunakan untuk mengaitkan colorbar dengan plot tersebut.
* ax.set\_ylim([-2, 3]) Mengatur batas skala pada sumbu-y dari -2 hingga 3. Ini berguna untuk mengatur batas area yang akan divisualisasikan pada sumbu-y.

Mengatur nilai u pada kondisi awal sehingga didapatkan distribusi suhu dengan Metode Finite Difference.

* w = u.copy() menyalin data suhu dari array u ke array w. Ini dilakukan untuk menyimpan salinan data suhu pada iterasi sebelumnya (waktu sebelumnya) agar dapat digunakan dalam perhitungan suhu baru pada iterasi berikutnya.
* for i in range(1, node-1) Memulai loop untuk menghitung suhu baru pada setiap titik grid kecuali batas. Loop ini iterates melalui titik-titik grid pada domain suhu, dan perhitungan suhu baru dilakukan berdasarkan persamaan difusi panas pada setiap iterasi.
* u[i] = (dt \* a \* (w[i-1] – 2\*w[i] + w[i+1]) / dx\*\*2) + w[i] perhitungan suhu baru (u[i]) pada titik grid ke-I, menggunakan metode beda hingga pada persamaan difusi panas. Nilai w[i-1], w[i] dan w[i+1] adalah suhu pada titik sekitarnya pada itersi sebelumnya. dt adalah langkah waktu, α adalah koefisien dufisivitas termal, dan dx adalah jarak antar titik grid,
* counter += dt menambahkan langkah waktu (dt) ke counter waktu simulasi, ini digunakan untuk melacak waktu simulasi.
* Print (“t: {:.3fs} s, Suhu rata-rata: {:.2f} Celcius”.format(counter,np.mean(u))) mencetak informasi waktu simulasi (t) dan suhu rata-rata saat ini (np.mean(u)) pada layer. Ini memberikan pemantauan terhadap evolusi suhu selama simulasi.
* Pcm.set\_array([u]) Memperbarui array data pada objek pcm yang digunakan untuk pcolormesh. Ini memastikan bahwa plot distribusi suhu diperbarui dengan nilai suhu baru setelah satu iterasi waktu.
* ax.set\_title(“Distribusi suhu pada t: {:.3f} s”.format(counter)) Mengatur judul (title) pada plot untuk mencantumkan waktu simulasi saat ini (t) Ini memastikan bahwa judul plot mencerminkan waktu pada setiap iterasi waktu.
* Plt.show agar plot dapat muncul
* Hasil output yang diprint adalah waktu dan suhu rata-rata serta distribusi suhu pada t=o.

Penggunaan metode Finite Difference Iloop tersebut dan perhitungan pada setiap titik grid mengimplementasikan solusi numerik dari persamaan difusi panas dalam satu dimensi menggunakan metode beda hingga.

Agar dapat mengetahui distribusi ketika waktu t = 1s. Cara yang dapat digunakan adalah dengan melakukan looping. Berikut adalah kode yang diubah sehingga diperoleh distribusi suhu ketika 1 sekon.

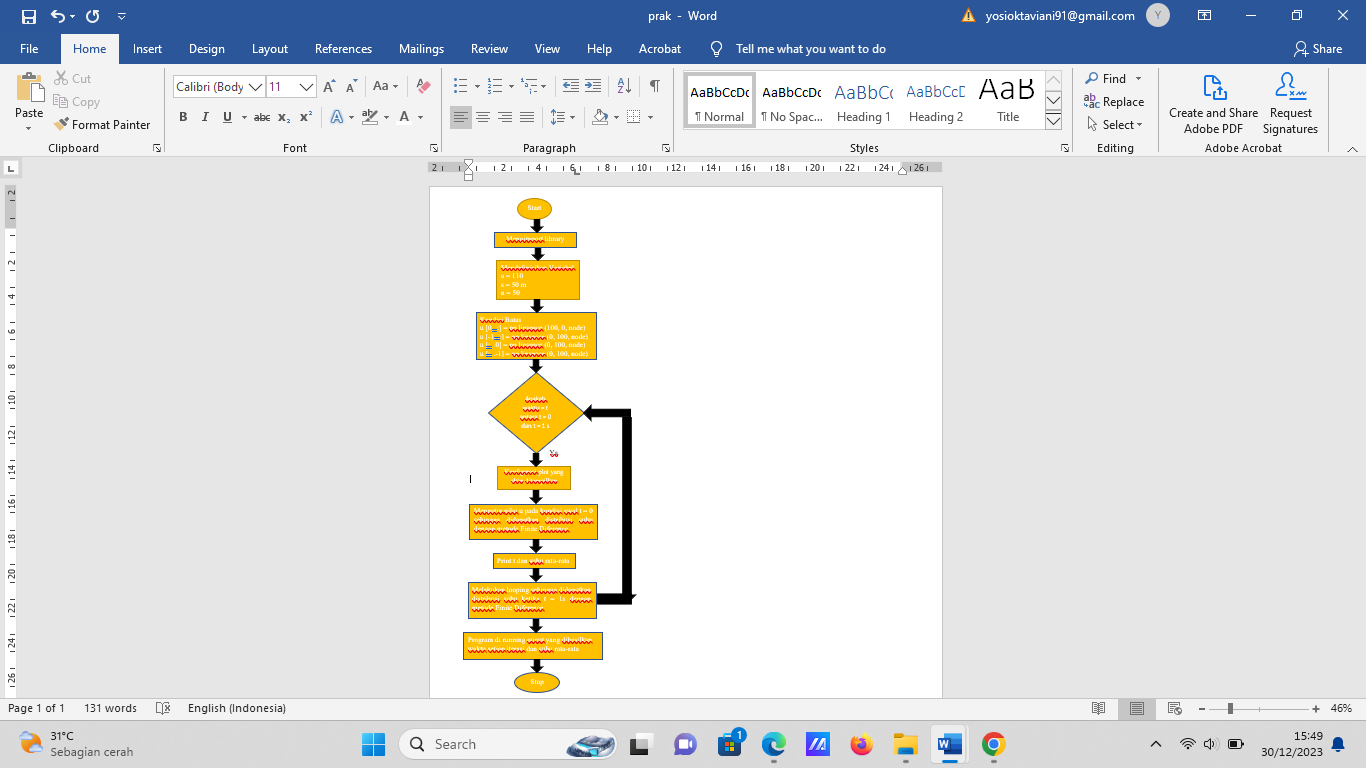
* Counter 0 inisialisasi variable counter untuk melacak waktu simulasi
* While counter < waktu memulai loop simulasi yang berjalan selama waktu simulasi yang ditentkan.
* W = u.copy() : menyalin data suhu dan iterasi sebelumnya untuk digunakan dalam perhitungan suhu baru.
* For i in range(1, node-1): looping melalui setiap titik grid kecuali batas.
* u[i] = (dt \* a \* (w[i-1] – 2\*w[i] + w[i+1]) / dx\*\*2) + w[i] Menggunakan metode beda hingga untuk menghitung suhu baru pada setiap titik grid berdasarkan persamaan difusi panas.

Maka plot setiap iterasi akan muncul dan suhu rata-rata serta waktu setiap iterasi akan muncul Setiap iterasi dipengaruhi oleh nilai dt yang berarti jarak antar iterasi dengan rumus berikut



pada program diimplementasikan pada dt = 0.5 \* dx \*\*2 / a

1. **Flow Chart 2 Dimensi**

****

1. Apa perbedaan antara Konduksi Panas 1 Dimensi dengan 2 Dimensi dalam penggunaan Metode Finite Difference?

**Jawab:**

* Pada model konduksi panas 1D, distribusi suhu hanya bervariasi dalam satu arah, misalnya, sepanjang sumbu x. Persamaan diferensial parsial (PDE) konduksi panas dalam 1D hanya bergantung pada koordinat x.
* Pada model konduksi panas 2D, distribusi suhu bervariasi dalam dua arah, misalnya, sepanjang sumbu x, dan y. Persamaan diferensial parsial (PDE) konduksi panas dalam 2D bergantung pada koordinat x dan y.
* Dalam 1D, formulasi metode beda hingga untuk mengaproksimasi turunan kedua suhu terhadap koordinat melibatkan perbedaan suhu antara titik-titik grid sekitarnya pada sumbu x.
* Dalam 2D, formulasi metode beda hingga untuk mengaproksimasi turunan kedua suhu terhadap koordinat ( ) melibatkan perbedaan suhu antara titik-titik grid sekitarnya pada kedua sumbu x dan y.
* Pada model 1D, matriks koefisien pada persamaan beda hingga memiliki bentuk yang lebih sederhana, karena hanya melibatkan satu dimensi koordinat.
* Pada model 2D, matriks koefisien pada persamaan beda hingga memiliki bentuk yang lebih kompleks karena melibatkan dua dimensi koordinat. Ini sering melibatkan matriks sparse untuk menggambarkan hubungan antara suhu pada titik-titik grid yang saling terkait.
* Pada model 1D, jumlah elemen dalam matriks dan sistem persamaan lebih sedikit karena hanya ada satu dimensi koordinat yang perlu dipertimbangkan.
* Pada model 2D, jumlah elemen dalam matriks dan sistem persamaan lebih besar karena melibatkan dua dimensi koordinat, sehingga solusi numerik dapat menjadi lebih komputasional mahal.