**LAPORAN PRAKTIKUM METODE NUMERIK**

Laporan Ini Disusun Untuk Memenuhi Quiz Mata Kuliah Praktikum Metode Numerik

Yang Diampu Oleh Dosen Pipit Pratiwi Rahayu, S.Si., M.Sc.



Disusun Oleh:  
Yogi Anggara (19106010011)

Ibnu Fadilah (19106010020)

Imam Hafidz Nuur (19106010033)

Muhammad Ulil Fahmi (19106010045)

**PROGRAM STUDI MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA  
YOGYAKARTA  
2020/2021**

1. **Latar Belakang**

Seiring kompleksnya permasalahan di dunia nyata, semakin sulit kita dalam mencari sebuah solusi analitik secara matematik. Hal ini memotivasi saintis dalam mengembangkan metode baru yang bisa mengestimasi nilai analitik tadi dengan error yang kecil. Metode ini kemudian dikenal dengan sebutan metode numerik. Terdapat banyak sekali algoritma pada metode numerik dengan keunikannya masing-masing. Di samping itu, beberapa algoritma metode numerik juga memiliki beberapa kesamaan dengan algoritma yang lain. Contohnya seperti metode secant dan newton-rapson untuk mencari nilai akar persamaan, serta metode riemann dan trapesium untuk menghitung nilai integral. Meskipun memiliki kesamaan, tentunya metode-metode tersebut memiliki keandalan yang berbeda. Dari fakta tersbut kami tertarik untuk menelaah lebih dalam terkait keandalam metode secant dan newton-rapson dalam mencari nilai akar persamaan, serta kenadalan metode riemann dan trapesium dalam menghitung nilai integral.

1. **Studi Kasus**

Diberikan fungsi . Tentukanlah:

1. Nilai akar ketika  menggunakan:
2. Metode secant dengan titik awal 14 dan 15, serta error sebesar 0.0005
3. Metode newton-rapson dengan titik awal 15 dan error sebesar 0.0005

Bandingkan keandalam 2 metode tersebut

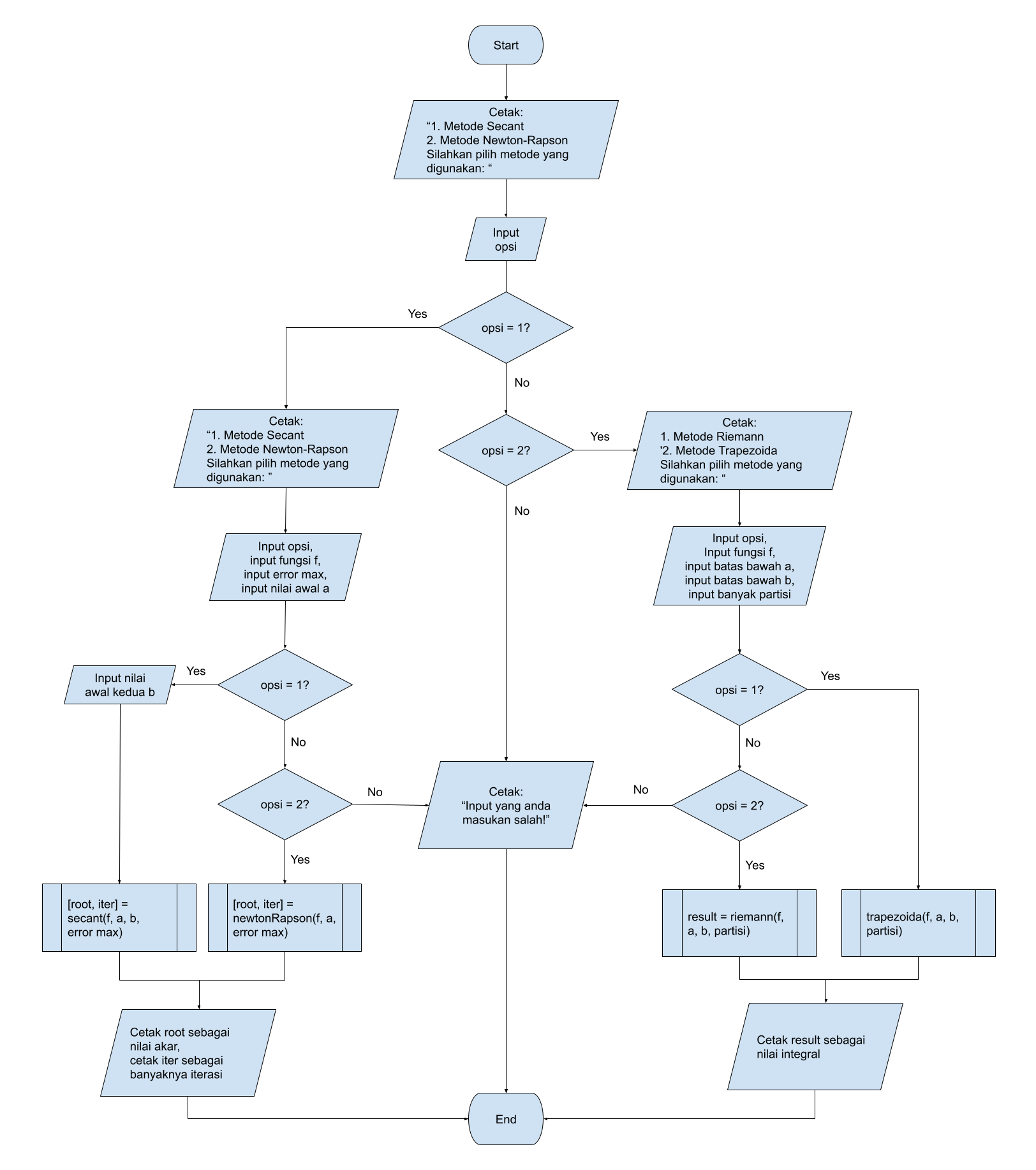
1. Nilai integral pada interval  menggunakan:
2. Metode riemann dengan jumlah partisi sebanyak 5
3. Metode trapesium dengan jumlah partisi sebanyak 5

Bandingkan keandalam 2 metode tersebut

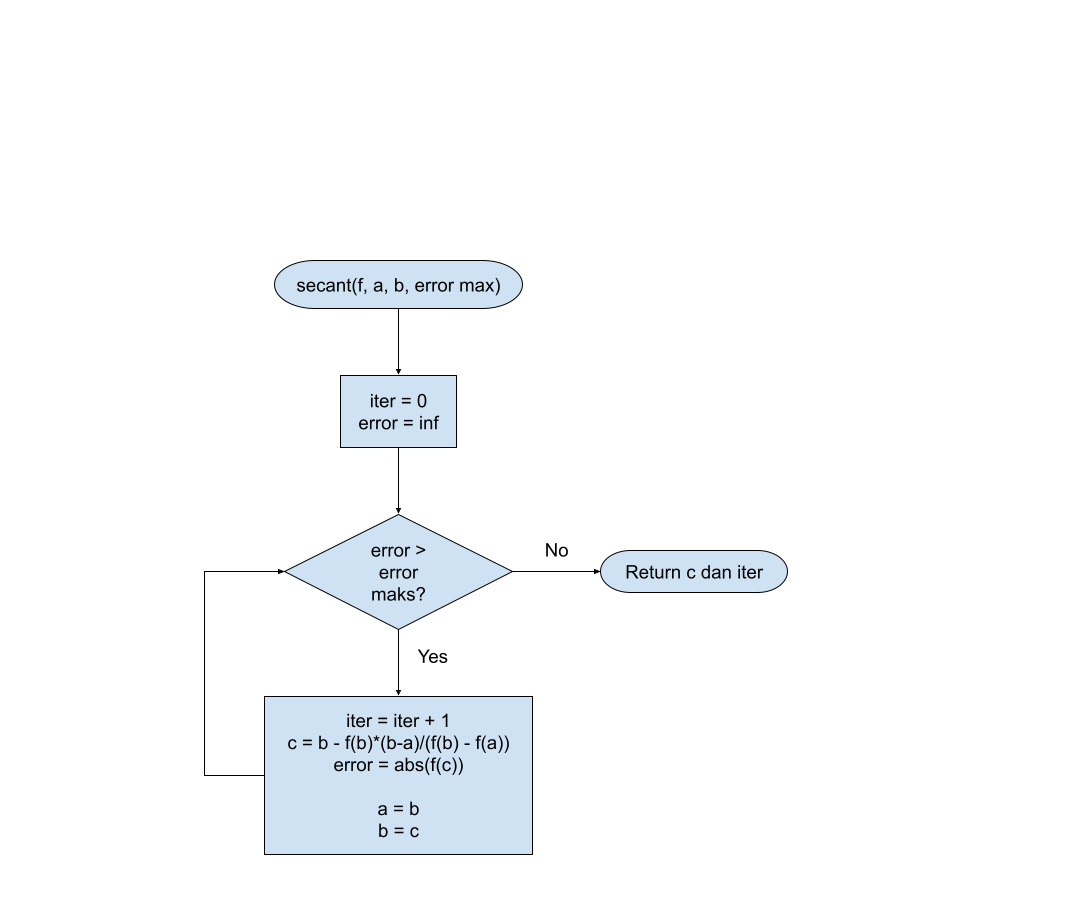
1. **Algoritma Metode yang Digunakan**

Algoritma yang kami gunakan di antaranya:

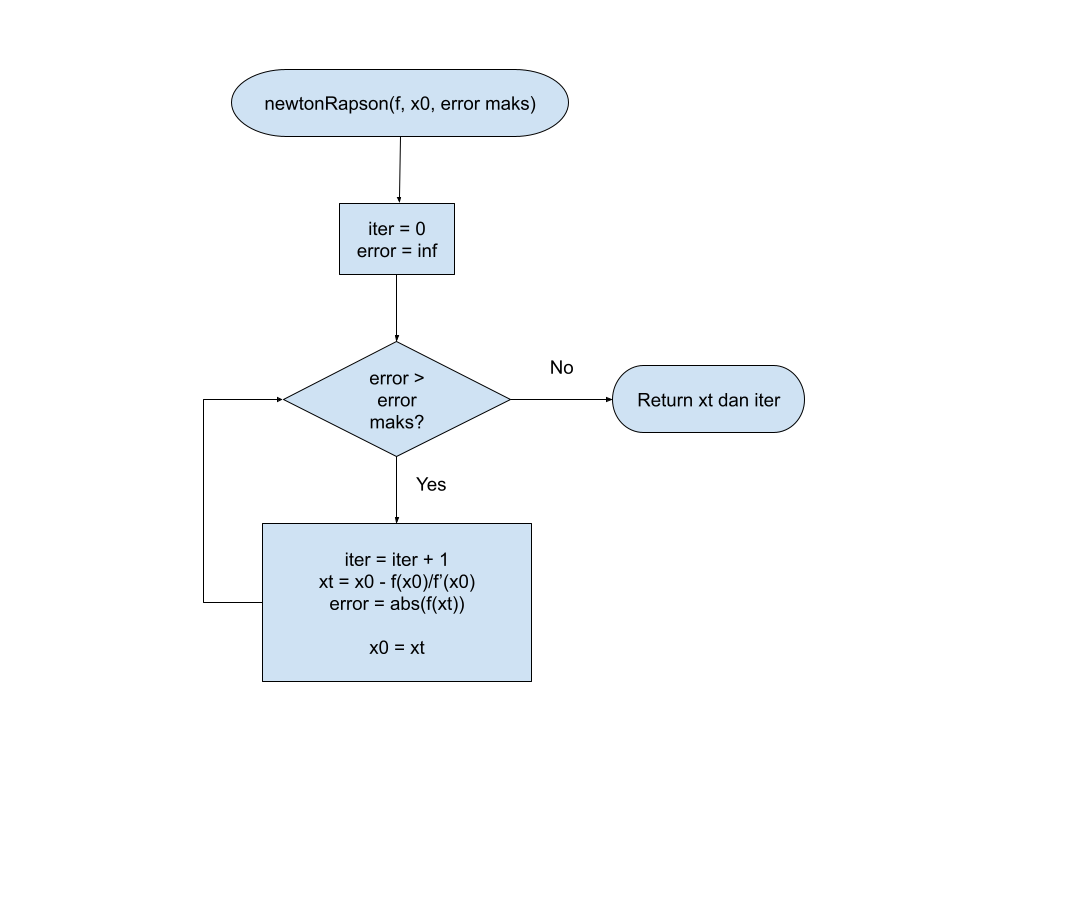
1. Metode secant dan newton-rapson untuk persamaan non linear
2. Metode riemann dan trapesium untuk integrasi numerik
3. **Flowchart**
4. Kode untuk tampilan (main.m)



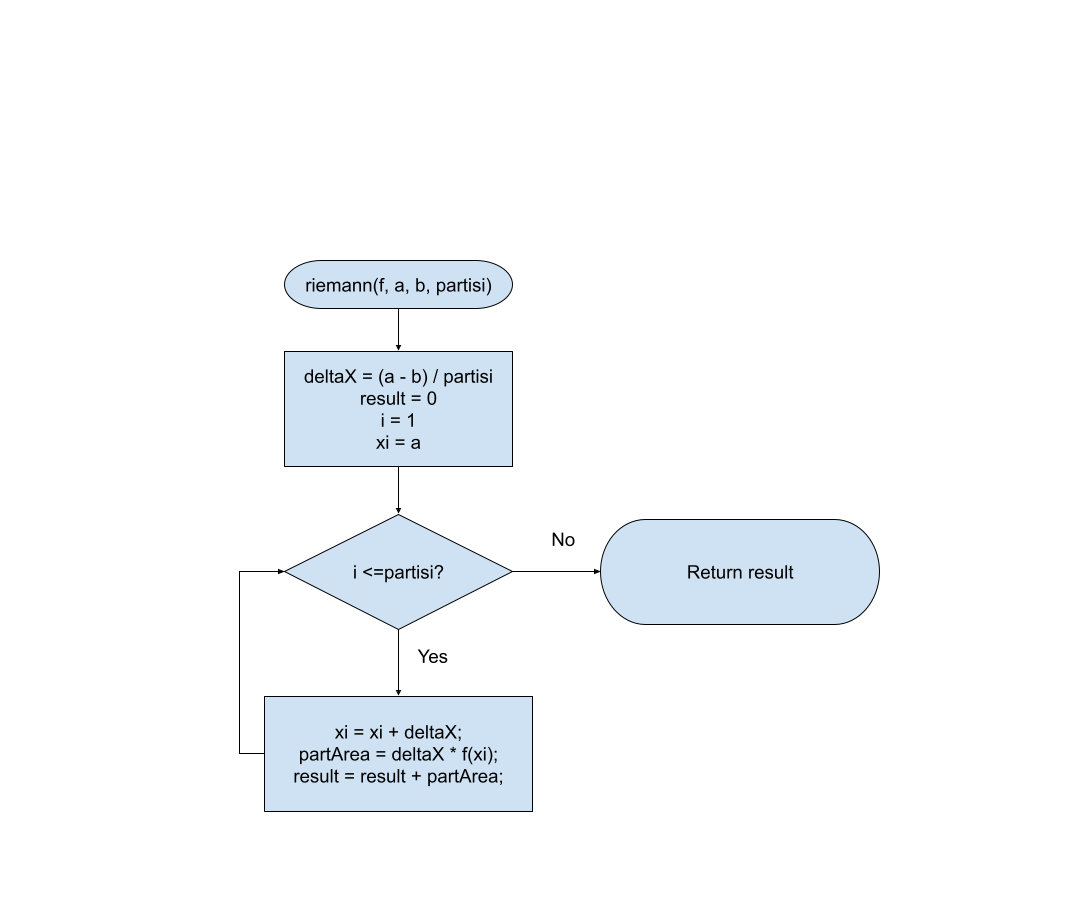
1. Metode secant (secant.m)



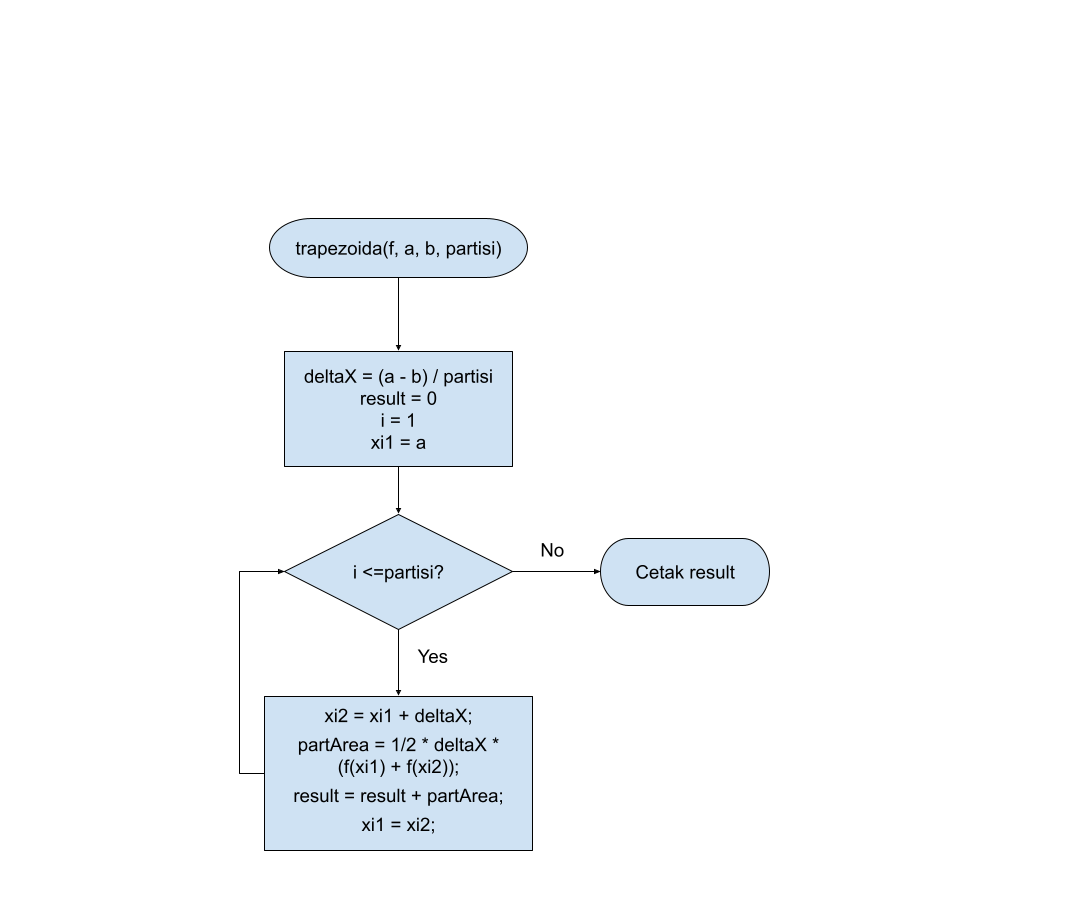
1. Metode newton-rapson (newtonRapson.m)



1. Metode riemann (riemann.m)



1. Metode trapesium (trapezoida.m)



1. **Input**

**main.m**

clc

disp('Kuis Metode Numerik');

disp(' ')

disp('Selamat datang!')

disp('1. Mencari nilai akar')

disp('2. Menghitung integral')

option = input('Silahkan pilih opsi: ');

disp(' ')

switch(option)

    case 1

        disp('1. Metode Secant')

        disp('2. Metode Newton-Rapson')

        option = input('Silahkan pilih metode yang digunakan: ');

        disp(' ')

        f = input('Masukan fungsinya (apit fungsi dg petik satu): ');

        errMax = input('Masukan error maksimum: ');

        a = input('Masukan nilai awal: ');

        switch(option)

            case 1

                b = input('Masukan nilai awal kedua: ');

                disp(' ')

                [root, iter] = secant(f, a, b, errMax);

                fprintf('Nilai akar: %f\n', root)

                fprintf('Banyak iterasi: %f\n', iter)

                break

            case 2

                disp(' ')

                [root, iter] = newtonRapson(f, a, errMax);

                fprintf('Nilai akar: %f\n', root)

                fprintf('Banyak iterasi: %f\n', iter)

                break

            otherwise

                disp('Input yang Anda masukan salah!')

                break

        end

    case 2

        disp('1. Metode Riemann')

        disp('2. Metode Trapezoida')

        option = input('Silahkan pilih metode yang digunakan: ');

        disp(' ')

        f = input('Masukan fungsinya: ');

        a = input('Masukan batas bawahnya: ');

        b = input('Masukan batas atasnya: ');

        part = input('Masukan banyak partisinya: ');

        disp(' ')

        switch(option)

            case 1

                result = riemann(f, a, b, part);

                fprintf('Nilai integralnya: %.f\n', result)

                break

            case 2

                result = trapezoida(f, a, b, part);

                fprintf('Nilai integralnya: %.f\n', result)

                break

            otherwise

                disp('Input yang Anda masukan salah!\n')

                break

        end

    otherwise

        disp('Input yang Anda masukan salah!')

        break

end

**secant.m**

function [c iter] = secant(f, a, b, errMax)

    f = inline(f);

    fa = f(a);

    fb = f(b);

    iter = 0;

    err = inf;

    fprintf('i \t x\_a \t x\_b \t x\_c \t f(x\_a) \t f(x\_b) \t f(x\_c)\n')

    while(err > errMax)

        iter = iter + 1;

        c = b - fb\*(b - a)/(fb - fa);

        fc = f(c);

        err = abs(fc);

        fprintf('%d \t %.3f \t %.3f \t %.3f \t %.3f \t %.3f \t %.3f \n', iter, a, b, c, fa, fb, fc)

        a = b;

        b = c;

        fa = fb;

        fb = fc;

    end

end

**newtonRapson.m**

function [xt iter] = newtonRapson(fun, x0, errMax)

    f = inline(fun);

    fx0 = f(x0);

    iter = 0;

    err = inf;

    try

        syms x;

        diffF = diff(fun, x);

        diffFx = inline(diffF);

        useSyms = true;

    catch e

        disp('======================================================')

        disp('Warning: Tidak bisa menggunakan Symbolic Math Toolbox!')

        fprintf(1,'The identifier was:\n%s\n',e.identifier);

        fprintf(1,'There was an error! The message was:\n%s\n',e.message);

        disp('Nilai turunan hanya akan diestimasi')

        disp('======================================================')

        useSyms = false;

    end

    fprintf('i \t x\_i \t x\_i+1 \t f(x\_i) \t f(x\_i+1) \n')

    while (err > errMax)

        iter = iter + 1;

        if(useSyms)

            m = diffFx(x0);

        else

            % metode selisih tengah

            h = 0.0001;

            m = (f(x0 + h) - f(x0 - h))/(2\*h);

        end

        xt = x0 - fx0/m;

        fxt = f(xt);

        err = abs(fxt);

        fprintf('%d \t %.3f \t %.3f \t %.3f \t %.3f \n', iter, x0, xt, fx0, fxt)

        x0 = xt;

        fx0 = fxt;

    end

end

**riemann.m**

function result = riemann(f, a, b, partition)

    f = inline(f);

    result = 0;

    deltaX = (b - a) / partition;

    xi = a;

    for i = 1:partition

        xi = xi + deltaX;

        partArea = deltaX \* f(xi);

        result = result + partArea;

    end

end

**trapezoida.m**

function result = trapezoida(f, a, b, partition)

    deltaX = (b - a) / partition;

    f = inline(f);

    result = 0;

    xi1 = a;

    for i = 1:partition

        xi2 = xi1 + deltaX;

        partArea = 1/2 \* deltaX \* (f(xi1) + f(xi2));

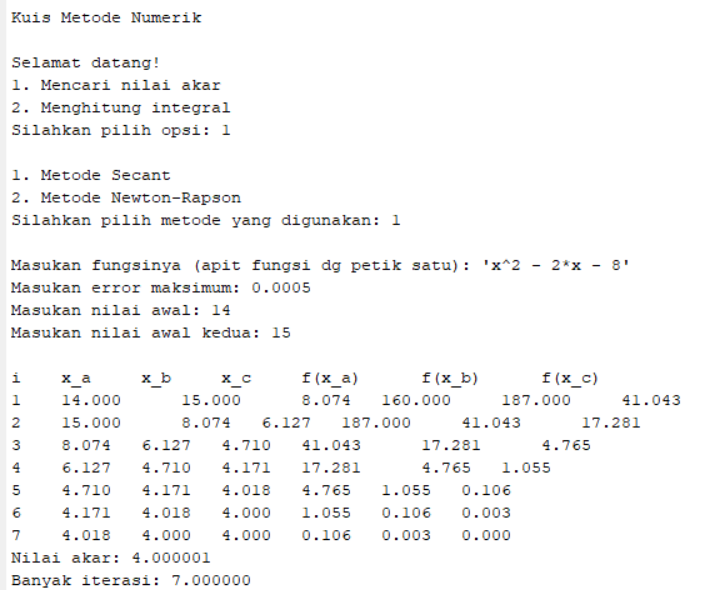
        result = result + partArea;

        xi1 = xi2;

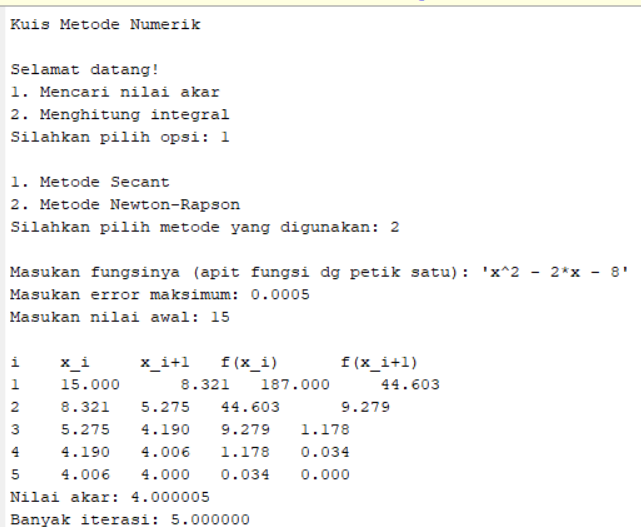
    end

end

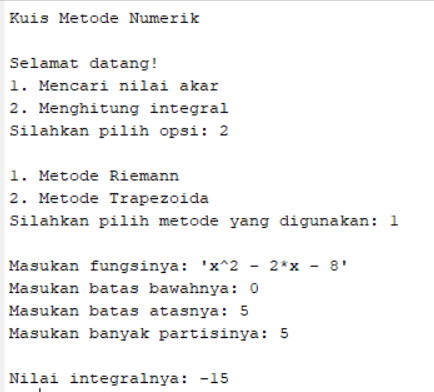
1. **Output**
2. Metode secant



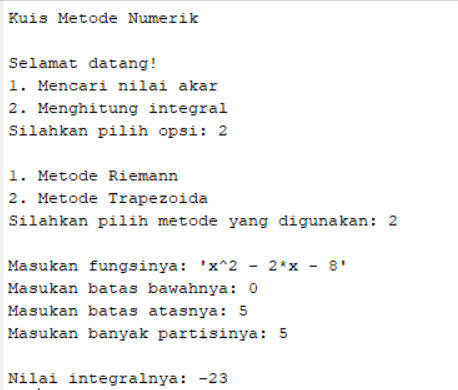
1. Metode newton-rapson



1. Metode riemann



1. Metode trapezoida



1. **Penutup**
2. Kesimpulan

Dari program yang kami buat untuk menyelesaikan studi kasus di aras, kami peroleh kesimpulan:

1. Dengan menggunakan metode secant, diperoleh akarnya sebesar 4.000001 dengan iterasi sebanyak 7 kali. Sedangkan dengan menggunakan metode newton-rapson, diperoleh akarnya sebesar 4.000005 dengan iterasi sebanyak 5 kali. Berdasarkan banyak iterasinya, dapat dikatakan bahwa pada kasus ini penurunan nilai error pada newton-rapson lebih cepat dibandingkan secant per iterasinya.
2. Dengan menggunakan metode riemann, diperoleh nilai integralnya sebesar -15. Sedangkan dengan menggunakan metode trapesium, diperoleh nilai integralnya sebesar 23. Ketika kami cek, nilai analitik dari integral tersebut sebesar , yang mana berarti pada kasus ini nilai error pada metode trapesium jauh lebih kecil dibandingkan error pada metode riemann.
3. Saran untuk tugas

Seharusnya pemberian tugas mempertimbangkan ketepatan waktunya juga. Memang *deadline* tugas terbilang lama, akan tetapi sebagian besar waktunya berada di waktu libur lebaran. Sehingga yang seharusnya mahasiswa bisa menikmati masa-masa liburannya, justru diharuskan untuk memikirkan dan mengerjakan tugas yang diberikan.

1. Komentar untuk tugas

Tugas yang diberikan terlalu banyak, sehingga waktu untuk lebih memahami metode dan implementasi ke kodenya menjadi kurang maksimal.