

PENERAPAN METODE TERTUTUP PADA PERANCANGAN APLIKASI PERHITUNGAN PERSAMAAN NON LINEAR BERBASIS ANDROID

Tony Darmanto¹, Kartono², Friska³

¹²³Teknik Informatika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak
e-mail: ¹tony.darmanto@yahoo.com, ²kartono1102@gmail.com, ³friskaalfavania@gmail.com

Abstract

The use of smartphone technology has penetrated almost all fields of human life, one of which is the field of education. Determination of the roots of non-linear equations requires complex calculations because it involves decimal calculations. Therefore the find (X) application is designed to facilitate the process of determining the roots of non linear equations. The author uses data collection methods with literature study. The analysis technique used is the Unified Modeling Language (UML) in modeling the system. In designing the application the author uses the Android Studio as an Integrated Development Environment (IDE). This research produces the application find (X). This application has features to determine the roots of non-linear equations with a closed method namely Biseksi (Bisection) and Regula Falsi (Regula Falsi Method). Applications can only calculate non linear equations which have powers of two to seven. From the entire research process it can be concluded that with the application of find (X), it is expected to overcome and minimize the problems that occur when the process of calculating non-linear equations with the Bisection method and Rule Falsi.

Keywords: Calculation of Non Linear Equations, Android, Application

Abstrak

Pemanfaatan teknologi *smartphone* sudah merambah pada hampir semua bidang kehidupan manusia, salah satunya adalah bidang pendidikan. Penentuan akar persamaan non linear memerlukan perhitungan yang kompleks karena melibatkan perhitungan desimal. Oleh karena itu dirancang aplikasi *find(X)* untuk memudahkan proses penentuan akar persamaan non linear. Penulis menggunakan metode pengumpulan data dengan studi pustaka. Teknik analisis yang digunakan adalah *Unified Modeling Language (UML)* dalam memodelkan sistem. Dalam perancangan aplikasi penulis menggunakan Android Studio sebagai *Integrated Development Environment (IDE)*. Penelitian ini menghasilkan aplikasi *find(X)*. Aplikasi ini memiliki fitur untuk menentukan akar persamaan non linear dengan metode tertutup yaitu Biseksi (*Bisection*) dan Regula Falsi (*Regula Falsi Method*). Aplikasi hanya dapat melakukan perhitungan persamaan non linear yang memiliki pangkat dua sampai dengan tujuh. Dari seluruh proses penelitian dapat disimpulkan bahwa dengan adanya aplikasi *find(X)* ini, diharapkan dapat mengatasi dan meminimalisir permasalahan yang terjadi saat proses perhitungan persamaan non linear dengan metode Biseksi dan Regula Falsi.

Kata Kunci: Perhitungan Persamaan Non Linear, Android, Aplikasi

1. PENDAHULUAN

Di era modern ini, teknologi berkembang dengan sangat pesat. Atas perkembangan-perkembangan tersebut, hampir semua aspek pekerjaan manusia memerlukan bantuan teknologi, baik berupa teknologi komputer maupun inovasi teknologi lainnya. Salah satu teknologi yang mengalami perkembangan sangat pesat adalah perkembangan teknologi mobile. Perangkat mobile yang dimaksud adalah *smartphone* yang dulunya disebut sebagai *handphone*. Disebut sebagai *smartphone* atau telepon pintar karena perangkat mobile tersebut kini tidak hanya dapat digunakan sebagai perangkat untuk keperluan telepon saja, namun telah berkembang menjadi perangkat serba guna, yang dapat digunakan untuk mendengarkan musik, bermain game dan masih banyak lagi.

Pemanfaatan teknologi *smartphone* sudah merambah pada hampir semua bidang kehidupan manusia, salah satunya adalah bidang pendidikan. Dalam bidang pendidikan, pemanfaatan teknologi dapat digunakan sebagai alat bantu perhitungan matematika. Tidak sebatas perhitungan aritmatika, namun dengan teknologi yang makin perhitungannya, penentuan akar pada persamaan non linear sangat membutuhkan bantuan kalkulator karena melibatkan perhitungan desimal yang tentunya akan sulit dilakukan jika masih menggunakan perhitungan manual.

Dalam penyelesaian persamaan non linear, terdapat beberapa metode yang dapat digunakan, diantaranya metode Biseksi (*Bisection*) dan Regula Falsi (*Regula Falsi Method*). Perhitungan dengan menggunakan kedua metode tersebut akan menghasilkan akar yang berbeda-beda, namun saling mendekati. Metode Biseksi (*Bisection*) dan Regula Falsi (*Regula Falsi Method*) merupakan metode tertutup, yaitu metode yang memerlukan dua variabel angka dalam penentuan akar dan perhitungan akar yang dihasilkan terletak di antara dua variabel tersebut.

Seperti yang telah disebutkan sebelumnya, penentuan akar persamaan non linear memerlukan perhitungan yang kompleks karena melibatkan perhitungan desimal, maka penulis ingin merancang suatu aplikasi perhitungan persamaan non linear yang dapat melakukan penentuan akar dengan menggunakan metode Biseksi (Bisection) dan Regula Falsi (Regula Falsi Method). Aplikasi dirancang pada perangkat mobile yang menggunakan sistem operasi berupa Android. Penggunaan sistem operasi Android dikarenakan sistem operasi ini bersifat open source sehingga memudahkan para pengembang untuk menciptakan dan memodifikasi aplikasi atau fitur-fitur yang belum ada di sistem operasi Android tersebut.

2. METODE PENELITIAN

2.1 Rancangan Penelitian, Teknik Pengumpulan Data, Teknik Analisis Sistem, dan Perancangan Sistem

2.1.1 Rancangan Penelitian

Rancangan penelitian yang dilakukan oleh penulis yaitu berupa desain penelitian deskriptif, yang mana penelitian yang dilakukan dengan cara menggambarkan komponen apa saja yang diperlukan untuk merancang.

2.1.2 Metode Pengumpulan Data

Adapun metode pengumpulan data yang digunakan adalah :

2.1.2.1 Studi Literatur

Metode literatur yang dilakukan penulis adalah dengan membaca sejumlah buku-buku yang berhubungan dengan masalah yang diteliti atau dengan cara mengumpulkan data melalui referensi dan literatur yang berkaitan dengan penelitian ini.

2.1.3 Teknik Analisis Sistem

Teknik analisis sistem yang digunakan penulis adalah *Unified Modeling Language (UML)* untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan alur sistem yang akan dibuat.

2.1.4 Teknik Perancangan Sistem

Teknik perancangan sistem yang digunakan oleh penulis dalam penelitian ini menggunakan bahasa pemrograman java dan android studio sebagai *Integrated Development Environment (IDE)*.

2.2 Landasan Teori

2.2.1 Perancangan Sistem

Perancangan sistem adalah proses menyusun atau mengembangkan sistem informasi yang baru^[1]. Perancangan sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah ke dalam suatu kesatuan yang utuh dan berfungsi^[2].

2.2.2 Persamaan Non Linear

Akar persamaan atau diistilahkan dengan sistem persamaan nonlinear merupakan bentuk persamaan aljabar yang nilainya sama dengan nol. Jika hanya, satu variabel bebas x dapat ditulis sebagai $f(x) = 0$. Bentuk ini banyak digunakan dalam model persamaan bidang teknik maupun sains^[3]. Pada dasarnya salah satu dari persoalan dasar dari metode numerik atau metodekomputasi adalah persoalan memperoleh akar (riil maupun kompleks) dari suatu fungsi $f(x)$. Proses ini melibatkan diperolehnya akar atau penyelesaian dari suatu persamaan nonlinear satu variabel bentuk $f(x) = 0$ secara proses berulang (iterasi)^[4].

2.2.3 Android

Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama^[5]. *Android is a Linux-based operating system, and it's a Google open source platform for mobile phones. Android is the world's most widely used smartphone operating system*^[6].

2.2.4 Matematika

Matematika berarti ilmu pengetahuan yang didapatkan dengan berpikir. Matematika adalah suatu cara berpikir, suatu cara pembuktian. Beberapa matematika melibatkan suatu eksperimen atau suatu observasi, tapi hampir semua bagian matematika berhubungan dengan pembuktian secara deduktif^[7].

2.2.5 Metode Tertutup

Metode Tertutup adalah pencarian akar dalam selang $[a,b]$ dan dipastikan pada selang ini setidaknya terdapat satu akar oleh sebab itu metode ini selalu berhasil menemukan akar. Metode ini disebut juga metode konvergen karena pada proses pencarian akar disetiap iterasi selalu diperoleh nilai konvergen menuju akar^[8]. Metode tertutup yaitu nilai akar yang dicari terletak di antara nilai tebakan awal $[x_b, x_a]$ ^[4].

2.2.6 Smartphone

Smartphone adalah telepon seluler yang kemampuannya tidak terbatas untuk komunikasi melalui telepon dan *text messaging*, tetapi juga untuk melakukan fungsi sebagaimana halnya komputer^[9]. *Smartphone* adalah perangkat telepon genggam yang dilengkapi fitur-fitur menarik dan bermanfaat yang dapat membantu Anda berkomunikasi secara lebih mudah serta nyaman dengan keluarga, teman, juga klien bisnis^[10].

2.2.7 Java

Java adalah bahasapemrograman yang dapat dijalankan diberbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam^[11]. Java merupakan perangkat lunak untuk pemrograman beberapa tujuan (multipurpose), multiplatform (dapat berjalan di beberapa sistem operasi), mudah dipelajari dan powerful^[12].

2.2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standart untuk merancang model sebuah system^[13]. *Unified Modeling Language* (UML) adalah bahasa untuk mengspesifikasi, memvisualisasikan, serta mengonstruksi bagaimana dasar sistem perangkat lunak, termasuk melibatkan pemodelan aturan-aturan bisnis^[14].

2.2.9 Android Studio

Android Studio merupakan IDE (Integrated Development Environment) official berbasis IntelliJ IDEA untuk membuat aplikasi Android^[15]. Android Studio merupakan IDE (Integrated Development Environment) resmi dari Android. Android Studio dibangun dengan tujuan mempercepat proses pembangunan aplikasi yang berkualitas tinggi untuk device Android^[16].

3. HASIL DAN PEMBAHASAN

3.1 Hasil Analisis

Kehidupan manusia sekarang ini tidak dapat lepas dari teknologi yang dapat membantu memudahkan segala aktivitas manusia. Salah satu bentuk dari teknologi tersebut adalah *smartphone*. *Smartphone* merupakan salah satu teknologi yang cukup populer karena fitur-fitur yang dimiliki. Sekarang ini *smartphone* tidak hanya digunakan sebagai alat komunikasi dan alat penunjang dalam bidang pendidikan.

Dalam penyelesaian persamaan non linear, terdapat beberapa metode yang dapat digunakan, diantaranya metode Biseksi (*Bisection*) dan Regula Falsi (*Regula Falsi Method*). Perhitungan dengan menggunakan kedua metode tersebut akan menghasilkan akar yang berbeda-beda, namun saling mendekati. Penentuan akar persamaan non linear memerlukan perhitungan yang kompleks karena melibatkan perhitungan desimal, oleh karena itu penulis merancang aplikasi find(X) untuk mempermudah perhitungan persamaan akar non linear.

3.2 Prosedur Pengoperasian Aplikasi

3.2.1 Prosedur Menghitung Akar Persamaan

Bahwa untuk melakukan perhitungan akar *user* diharuskan untuk mengakses *menu* Hitung dan kemudian sistem akan menampilkan *layout Input* Persamaan. Saat *layout Input* Persamaan sudah ditampilkan, *user* diharuskan untuk menginputkan koefisien dari persamaan yang akan dihitung. Setelah koefisien sudah diinputkan, maka *user* dapat mengakses *menu* Lanjut untuk menampilkan *layout Input* Lanjutan. Pada *layout Input* Lanjutan tersebut, *user* diharuskan untuk memilih metode perhitungan serta menginputkan nilai a, b dan jumlah iterasi.

3.2.2 Prosedur Mengakses Contact

Untuk mengakses *contact*, *user* diharuskan untuk mengakses *menu contact* dan kemudian sistem akan menampilkan *layout Contact*. *User* dapat melihat isi *contact person* pengembang aplikasi pada *layout* tersebut.

3.2.3 Prosedur Mengakses About

Untuk melihat deskripsi dari aplikasi perhitungan persamaan ini, *user* diharuskan untuk mengakses *menu About* yang terdapat pada *layout Menu*.

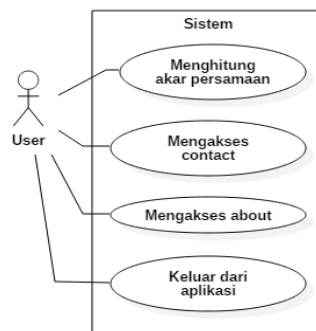
3.2.4 Prosedur Keluar dari Aplikasi

Untuk keluar dari aplikasi. Jika *user* memilih *menu* konfirmasi Ya, maka sistem akan menutup aplikasi. Sedangkan jika *user* memilih *menu* konfirmasi Tidak, maka sistem akan mengembalikan ke *layout Menu*.

3.3 Gambaran Umum Rancangan Aplikasi Melalui Use Case Diagram

Diagram *use case* untuk perancangan aplikasi dapat dilihat pada gambar 1. diagram *Use Case* di atas, terdapat sebuah *actor*, yaitu *user* atau pengguna aplikasi, dimana *user* tersebut dapat berinteraksi secara langsung pada aplikasi. *User* dapat melakukan perhitungan akar persamaan, mengakses *menu contact*, mengakses *menu about*, dan keluar dari aplikasi.

3.3.1 Diagram Use Case



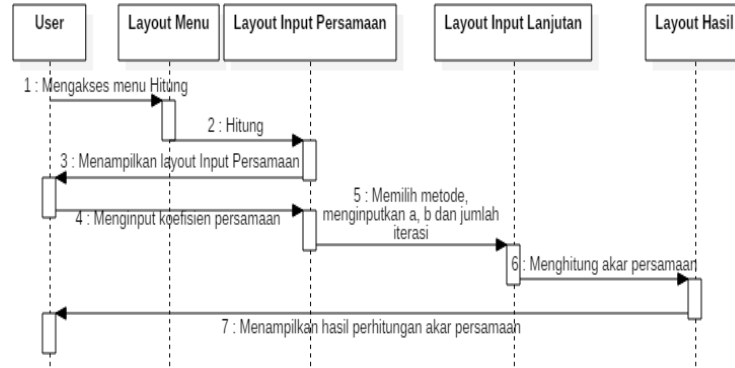
Gambar 1. Use Case Diagram

3.2 Diagram Sekuensial Perancangan Aplikasi

3.2.1 Diagram Sekuensial Urutan Menghitung Akar Persamaan

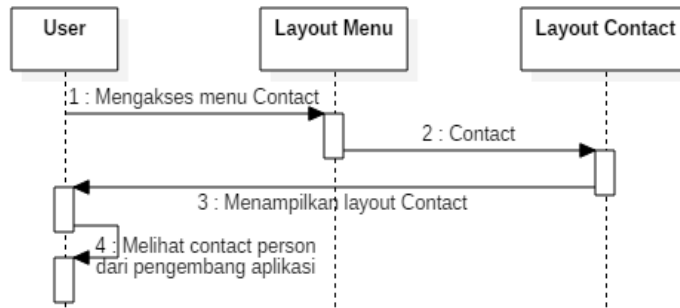
Gambar 2 merupakan diagram diagram sekuensial Urutan Menghitung Akar Persamaan. Diagram Urutan Menghitung Akar Persamaan dapat dijelaskan sebagai berikut:

- a. *User* mengakses *menu* Hitung agar sistem dapat menampilkan *layoutInput* Persamaan.
- b. Saat *layoutInput* Persamaan telah ditampilkan, *user* diharuskan untuk meng-*input*-kan koefisien persamaan.
- c. Setelah koefisien telah diisi, maka *user* dapat mengakses *menu* Lanjut untuk menampilkan *layoutInput* Lanjutan. Pada *layout* tersebut, *user* diharuskan untuk memilih metode perhitungan dan meng-*input*-kan nilai a,b dan jumlah iterasi. Setelah itu *user* dapat mengakses *menu* Hitung dan sistem akan melakukan perhitungan yang ditampilkan pada *layout* Hasil.



Gambar 2. Diagram Sekuensial Urutan Menghitung Akar Persamaan

3.2.2 Diagram Sekuensial Urutan Mengakses Contact

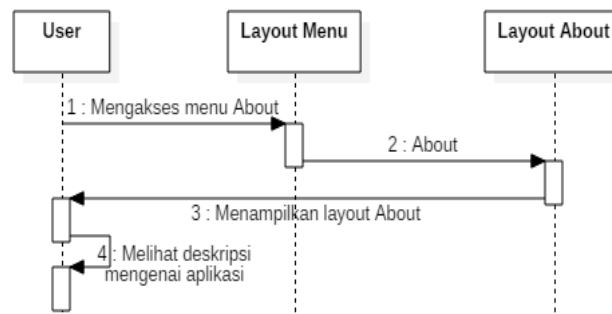


Gambar 3. Diagram Sekuensial Mengakses Contact

Gambar 3 merupakan diagram sekuensial Mengakses Contact. Diagram urutan mengakses *Contact* dapat dijelaskan sebagai berikut:

- a. *User* diharuskan untuk mengakses *menuContact* yang terdapat pada *layoutMenu*.
- b. Setelah itu, sistem akan menampilkan *layoutContact* dan *user* dapat melihat *contactperson* dari pengembang aplikasi.

3.2.3 Diagram Sekuensial Mengakses About



Gambar 4. Diagram Sekuensial Mengakses About

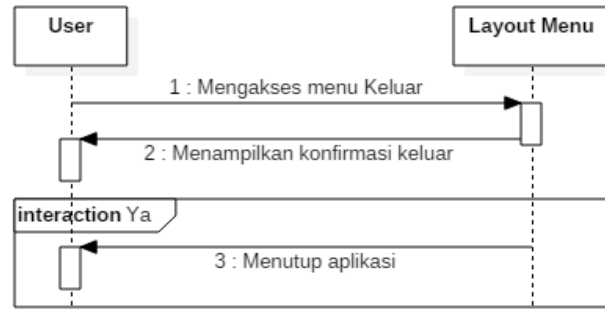
Gambar 4 merupakan diagram sekuensial Mengakses About. diagram urutan mengakses *about* di atas, maka dapat dijelaskan bahwa:

- User* diharuskan untuk mengakses *menuAbout* yang terdapat pada *layoutMenu*.
- Setelah itu sistem akan menampilkan *layoutAbout* dan *user* dapat melihat deskripsi mengenai aplikasi perhitungan persamaan non linear ini.

3.2.4 Diagram Sekuensial Keluar dari Aplikasi

Gambar 5 merupakan diagram sekuensial Keluar dari Aplikasi. diagram urutan keluar dari aplikasi di atas, maka dapat dijelaskan bahwa:

- Untuk keluar dari aplikasi, *user* dapat mengakses *menu* Keluar yang terdapat pada *layoutMenu*.
- Kemudian sistem akan menampilkan konfirmasi untuk keluar. Jika *user* memilih *menuYa*, maka aplikasi akan ditutup. Sedangkan jika *user* memilih *menu* Tidak, maka sistem akan mengembalikan ke *layoutMenu*.



Gambar 5. Diagram Sekuensial Keluar dari Aplikasi

3.3 Tampilan Interface Aplikasi

3.3.1 Tampilan Interface Menu Utama



Gambar 6. Tampilan Interface Menu Utama

Gambar 6 merupakan tampilan dari menu utama. *Layout Menu* merupakan *layout* yang pertama kali ditampilkan saat aplikasi find x ini dijalankan. Pada *layout* ini terdapat empat buah *menu*, yaitu *menu* Hitung, *menu* Contact, *menu* About dan *menu* Keluar. Setiap *menu* tersebut dirancang dengan menggunakan *cardview* yang dijadikan *button*. *Menu* Hitung berfungsi untuk menampilkan *layout* Input Persamaan sebagai tahapan *input*-an awal untuk proses perhitungan. Sedangkan *menu* Contact berfungsi untuk menampilkan *layout* Contact, *menu* About berfungsi untuk menampilkan *layout* About dan *menu* Keluar untuk menutup aplikasi find x ini.

3.3.2 Tampilan Tabel Proses Pencarian Akar dengan Metode Biseksi

Berdasarkan proses pencarian akar persamaan $f(x) = x^3 + 4x^2 - 10$ dengan menggunakan nilai tebakan pada *interval* [1,2] yang dilakukan iterasi sebanyak 12 pengulangan, diperoleh akar senilai 1,364991 (nilai c pada pengulangan terakhir). Jika dilihat dari nilai $f(c)$, setiap pengulangan akan memperkecil nilai $f(c)$ tersebut. Dengan demikian, dapat disimpulkan, jika semakin banyak pengulangan yang terjadi, maka nilai akar yang diperoleh dari persamaan akan semakin mendekati 0 atau dengan kata lain akar yang diperoleh akan lebih akurat.

3.3.3 Tampilan Tabel Proses Pencarian Akar dengan Metode Regula Falsi

Berdasarkan proses pencarian akar persamaan $f(x) = x^3 + 4x^2 - 10$ dengan menggunakan nilai tebakan pada interval $[1,2]$ yang dilakukan iterasi sebanyak 12 pengulangan, diperoleh akar senilai 1,365230 (nilai c pada pengulangan terakhir). Jika dilihat dari nilai $f(c)$, setiap pengulangan akan memperkecil nilai $f(c)$ tersebut. Dengan demikian, dapat disimpulkan, jika semakin banyak pengulangan yang terjadi, maka nilai akar yang diperoleh dari persamaan akan semakin mendekati 0 atau dengan kata lain akar yang diperoleh akan lebih akurat.

Gambar 7. Tampilan Interface Input Persamaan

Gambar 7 merupakan tampilan dari Interface Input Persamaan. *Layout Input Persamaan* ini merupakan *layout* yang akan ditampilkan saat *user* mengakses *menu* Hitung yang terdapat pada *layout Menu*. Pada *layout* ini terdapat delapan buah *checkbox* dan delapan buah *edittext*. Setiap *checkbox* dicentang, maka nilai koefisien dari variabel x tersebut akan menjadi bilangan negatif. Misalnya pada x^6 *checkbox* tersebut dicentang dan *edittext* diberi nilai 8, maka nilai yang dimaksud tersebut adalah $-8x^6$. Setiap *input*-an koefisien dan nilai *checkbox* akan ditampilkan pada *textview* di atas untuk membentuk susunan persamaan secara utuh. Selain itu, terdapat sebuah *button* Lanjut yang berfungsi untuk melanjutkan *input*-an pada *layout Input Lanjutan*.

3.3.4 Tampilan Interface Input Lanjutan

Gambar 8. Tampilan Interface Input Lanjutan

Gambar 8 merupakan tampilan *Interface layout Input* Lanjutan akan ditampilkan saat *user* menekan *button* Lanjut yang terdapat pada *layout Input* Persamaan. Pada *layout* ini, *user* diharuskan untuk memilih metode pencarian akar yang akan digunakan. Aplikasi find x ini menyediakan dua metode pencarian dengan metode tertutup, yaitu Biseksi dan Regula Falsi. Selain itu, *user* juga diminta untuk meng-*input*-kan nilai a, b dan jumlah pengulangan yang diperlukan. Jika semua *input*-an telah terisi, *user* dapat melanjutkan perhitungan dengan mengakses *button* Hitung yang telah tersedia.

3.4 Tampilan Interface Hasil

3.4.1 Tampilan Interface Hasil Biseksi



Gambar 9. Tampilan Interface Hasil Biseksi

Gambar 9 merupakan tampilan interface hasil Biseksi. Pada tampilan *layout* Hasil di atas, dapat dilihat bahwa dari persamaan $f(x) = x^3 + 4x^2 - 10$ dengan menggunakan nilai $a = 1$ dan $b = 2$ serta pengulangan sebanyak 12 kali dengan menggunakan metode Biseksi, diperoleh nilai akar dari persamaan tersebut dengan nilai 1,364991.

3.4.2 Tampilan Interface Hasil Regula Falsi



Gambar 10. Tampilan Interface Hasil Regula Falsi

Gambar 10 merupakan interface hasil Regula Falsi. Pada tampilan *layout* Hasil di bawah ini, dapat dilihat bahwa dari persamaan $f(x) = x^3 + 4x^2 - 10$ dengan menggunakan nilai $a = 1$ dan $b = 2$ serta pengulangan sebanyak 12 kali dengan menggunakan metode Regula Falsi, diperoleh nilai akar dari persamaan tersebut dengan nilai 1,365230.

Berikut merupakan contoh proses pencarian akar dengan menggunakan metode tertutup:

- Berdasarkan proses pencarian akar persamaan $f(x) = x^3 + 4x^2 - 10$ dengan menggunakan nilai tebakan pada *interval* [1,2] yang dilakukan iterasi sebanyak 12 pengulangan, diperoleh akar senilai 1,364991 (nilai c pada pengulangan terakhir). Jika dilihat dari nilai $f(c)$, setiap pengulangan akan memperkecil nilai $f(c)$ tersebut. Dengan demikian, dapat disimpulkan, jika semakin banyak pengulangan yang terjadi, maka nilai akar yang diperoleh dari persamaan akan semakin mendekati 0 atau dengan kata lain akar yang diperoleh akan lebih akurat.

Tabel 1
Proses Pencarian Akar Dengan Metode Biseksi

| Iterasi | A | b | C | f(a) | f(b) | f(c) | f(a).f(c) |
|---------|----------|----------|----------|-----------|----------|-----------|-----------|
| 1 | 1 | 2 | 1,5 | -5 | 14 | 2,375 | - |
| 2 | 1 | 1,5 | 1,25 | -5 | 2,375 | -1,79688 | + |
| 3 | 1,25 | 1,5 | 1,375 | -1,79688 | 2,375 | 0,162109 | - |
| 4 | 1,25 | 1,375 | 1,3125 | -1,79688 | 0,162109 | -0,84839 | + |
| 5 | 1,3125 | 1,375 | 1,34375 | -0,84839 | 0,162109 | -0,35098 | + |
| 6 | 1,34375 | 1,375 | 1,359375 | -0,35098 | 0,162109 | -0,09641 | + |
| 7 | 1,359375 | 1,375 | 1,367188 | -0,09641 | 0,162109 | 0,032364 | - |
| 8 | 1,359375 | 1,367188 | 1,363282 | -0,09641 | 0,032364 | -0,03214 | + |
| 9 | 1,363282 | 1,367188 | 1,365235 | -0,03214 | 0,032364 | 0,000082 | - |
| 10 | 1,363282 | 1,365235 | 1,364259 | -0,03214 | 0,000082 | -0,016027 | + |
| 11 | 1,364259 | 1,365235 | 1,364747 | -0,016027 | 0,000082 | -0,007974 | + |
| 12 | 1,364747 | 1,365235 | 1,364991 | -0,007974 | 0,000082 | -0,003946 | + |

- b. Berdasarkan proses pencarian akar persamaan $f(x) = x^3 + 4x^2 - 10$ dengan menggunakan nilai tebakan pada interval $[1,2]$ yang dilakukan iterasi sebanyak 12 pengulangan, diperoleh akar senilai 1,365230 (nilai c pada pengulangan terakhir). Jika dilihat dari nilai $f(c)$, setiap pengulangan akan memperkecil nilai $f(c)$ tersebut. Dengan demikian, dapat disimpulkan, jika semakin banyak pengulangan yang terjadi, maka nilai akar yang diperoleh dari persamaan akan semakin mendekati 0 atau dengan kata lain akar yang diperoleh akan lebih akurat.

Tabel 2
Proses Pencarian Akar Dengan Metode Regula Falsi

| Iterasi | A | b | C | f(a) | f(b) | f(c) | f(a).f(c) |
|---------|----------|---|----------|-------------|------|-------------|-----------|
| 1 | 1 | 2 | 1,263158 | -5 | 14 | -1,60227 | + |
| 2 | 1,263158 | 2 | 1,338828 | -1,60227 | 14 | -0,43036 | + |
| 3 | 1,338828 | 2 | 1,358555 | -0,43036 | 14 | -0,10987 | + |
| 4 | 1,358555 | 2 | 1,363550 | -0,10987 | 14 | -0,02772 | + |
| 5 | 1,363550 | 2 | 1,364808 | -0,02772 | 14 | -0,00697 | + |
| 6 | 1,364808 | 2 | 1,365124 | -0,00697 | 14 | -0,00175 | + |
| 7 | 1,365124 | 2 | 1,365203 | -0,00175 | 14 | -0,00045 | + |
| 8 | 1,365203 | 2 | 1,365223 | -0,00045 | 14 | -0,00012 | + |
| 9 | 1,365223 | 2 | 1,365228 | -0,00012 | 14 | -0,000033 | + |
| 10 | 1,365228 | 2 | 1,365229 | -0,000033 | 14 | -0,000017 | + |
| 11 | 1,365229 | 2 | 1,365230 | -0,000017 | 14 | -0,00000022 | + |
| 12 | 1,365230 | 2 | 1,365230 | -0,00000022 | 14 | -0,00000005 | + |

- c. Berdasarkan hasil di bawah, dapat dilihat jika hasil akar x yang diperoleh dari kedua metode tersebut berbeda, namun saling mendekati, dengan selisih 0,00024. Dari kedua metode tersebut, nilai $f(x)$ yang paling mendekati 0 adalah metode Regula Falsi.

Tabel 3
Perbandingan Hasil Akar Kedua Metode

| Metode | X | f(x) |
|--------------|----------|-------------|
| Biseksi | 1,364991 | -0,003946 |
| Regula Falsi | 1,365230 | -0,00000005 |

4. KESIMPULAN

Berdasarkan analisis, perancangan, pengoperasian dan pengujian aplikasi perhitungan persamaan non linear menggunakan metode tertutup berbasis Android maka penulis memperoleh kesimpulan yang dapat diambil sebagai berikut:

- Dengan aplikasi find x, pengguna dapat melakukan perhitungan akar dengan menggunakan metode tertutup. Pada aplikasi find x, terdapat dua buah metode tertutup yang dapat digunakan, yaitu metode biseksi dan metode Regula Falsi. Untuk melakukan perhitungan dengan aplikasi find x ini, pengguna dapat meng-*input*-kan persamaan hingga ke variabel x berpangkat 7 dan juga pengguna harus meng-*input*-kan nilai a dan b sebagai tebakan awal dimana akar tersebut dapat berada pada interval a dan b tersebut.
- Perhitungan akar dengan suatu persamaan pada metode Biseksi dan Regula Falsi dapat menghasikan nilai yang berbeda, namun nilainya saling mendekati.
- Untuk kedua metode, semakin banyak pengulangannya maka nilai akar yang diperoleh untuk memenuhi persamaan $f(x) = 0$ akan semakin akurat.

5. SARAN

Setelah melakukan analisis pada hasil perancangan aplikasi perhitungan persamaan non linear menggunakan metode tertutup berbasis Android, penulis menyadari bahwa aplikasi masih belum sempurna dan memiliki kekurangan. Adapun beberapa saran dari penulis untuk pengembangan aplikasi lebih jauh antara lain:

- Diharapkan pengembangan aplikasi ini dapat dilakukan dengan menambahkan metode-metode perhitungan lain, seperti metode Tabulasi dan sebagainya.
- Untuk memudahkan dalam peng-*input*-an dan pemahaman persamaan yang di-*input*-kan, setiap derajat variabel dapat dituliskan dengan *superscript*.
- Aplikasi dapat dikembangkan agar dapat memuat derajat atau pangkat variabel yang lebih banyak.
- Interface* atau tampilan dapat dirancang menjadi lebih menarik.

UCAPAN TERIMA KASIH

Dalam penyusunan penelitian, penulis telah banyak mendapat bantuan bimbingan, data, saran, dan dukungan moril dari berbagai pihak, pada kesempatan ini penulis ingin mengucapkan terimakasih kepada civitas akademika Sekolah Tinggi Manajemen Informatika dan Komputer Widya Dharma Pontianak dan kepada pihak-pihak lain yang sudah sangat membantu penulis secara teknis dan moril dalam menyelesaikan penelitian ini.

DAFTAR PUSTAKA

- [1] Muharto, Arisandy Ambarita. (2016). *Metode Penelitian Sistem Informasi: Mengatasi Kesulitan Mahasiswadalam Menyusun Proposal Penelitian*. Deepublish. Yogyakarta.
- [2] Yakub, (2012), *Pengantar Sistem Informasi*, Ed.1, Graha Ilmu, Yogyakarta.
- [3] Sasongko. (2010). *Metode Numerik Dengan Scilab*. Andi. Yogyakarta.
- [4] Utomo. (2018). *Pengantar Metode Komputasi*. Gadjah Mada University Press. Yogyakarta.
- [5] Juhara, Zamrony P. (2016). *Panduan Lengkap Pemrograman Android*, Andi. Yogyakarta.
- [6] Tamma, Rohit, Satish Bommisetty dan Heather Mahalik. (2018). *Practical Mobile Forensics*. Edisi Ketiga. Packt Publishing. UK.
- [7] Yusup, Muhamad. (2010). *MATEMATIKA: Kelompok Sosial, Administrasi Perkantoran, dan Akuntansi untuk Sekolah Menengah Kejuruan Kelas X*. PT Grafindo Media Pratama. Jakarta.
- [8] Sriyanto. (2017). *Mengobarkan Api Matematika*. Jejak. Jawa Barat.
- [9] Rahmad, Cahya, Deasy Sandhya Elya Ikawati dan Yan Watequlis Syaifudin. (2018). *Metode Numerik*. Polinema Press. Malang.
- [10] Heriyanto, Andri Puspo. (2016). *Mobile Phone Forensics Theory: Mobile Phone Forensics dan Security Series*. CV. Andi Offset. Yogyakarta.
- [11] Thoivids dan Hojanto. (2017). *Presentation Mastery*. Gramedia Pustaka Utama. Jakarta.
- [12] Wahana Komputer. (2015). *Webmaster Series: Menguasai HTML*. Penerbit Andi. Yogyakarta.
- [13] Supardi. (2010). *Semua Bisa Menjadi Programmer Java Case Study*. PT Elex Media Komputindo. Jakarta.
- [14] Yasin, Verdi. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek: Permodelan, Arsitektur dan Perancangan (Modeling, Architecture and Design)*. Mitra Wacana Media. Jakarta.
- [15] Nugroho, Adi. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. C.V Andi Offset. Yogyakarta.
- [16] Jubilee Enterprise. (2015). *Mengenal Dasar-dasar Pemrograman Android*. PT. Elex Media Komputindo. Jakarta.