OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

Pengembangan Aplikasi Penghitung Luas Segitiga Dengan Java

Rizka Aprilia[1*], Robiyatul Adawiyyah[1], Fauzan Azhari[1], Muhammad Ilham Satiyar[1], Aditya Rizki Ramadhan[1]

[1]Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia
Email: [1*]rizkaaprilia152@gmail.com, [2]robiatulwiyah33@gmail.com, [3]fauzantkj178@gmail.com, [4]ilhamsatiar05@gmail.com, [5]adityarzk354@gmail.com
(* : coressponding author)

Abstrak−Penelitian ini adalah penelitian pengembangan aplikasi penghitung luas segitiga dengan java yang bertujuan untuk mengumpulkan tugas kelompok pemograman. Pengujian ini dilakukan dengan menggunakan aplikasi dibangun dengan bahasa pemograman java dan alat pengembangan NetBeans IDE, pemograman java merupakan keterampilan penting dalam bidang teknologi informasi. Maka dari itu membangun aplikasi sederhana untuk menghitung luas segitiga menggunakan bahasa pemogaraman java. Penelitian ini untuk memperkenalkan konsep dasar bagimana pengembangan aplikasi penghitung luas segitiga dengan java dalam bahasa pemograman java untuk mempermudah mahasiswa atau pemula yang bertujuan untuk menyelesaikan masalah ini kepada pemula, untuk mengajarkan langkah langkah dalam pembuat aplikasi java, serta memberikan pengalaman peraktis dalam implementasi konsep pemograman dalam proyek nyata. Data dikumpulkan dengan menggunakan teknik pembuatan flowchart kedalam kode java, pengujian, debungging, dan evaluasi hasil, aplikasi ini memenuhi kriteria kualitas perangkat lunak yang baik.

Kata Kunci: Pemograman Java, Pengembangan Aplikasi, Luas Segitiga, Flowchart

Abstract−*This research is research into the development of a triangle area calculator application using Java which aims to collect programming group assignments. This testing was carried out using an application built with the Java programming language and the NetBeans IDE development tool, Java programming is an important skill in the field of information technology. Therefore, build a simple application to calculate the area of a triangle using the Java programming language. This research is to introduce the basic concept of how to develop a triangle area calculator application using Java in the Java programming language to make it easier for students or beginners who aim to solve this problem for beginners, to teach the steps in creating Java applications, and to provide practical experience in implementing programming concepts in real project. Data was collected using techniques for creating flowcharts into Java code, testing, debugging, and evaluating results, this application meets the criteria for good software quality*

Keywords: *Java, Programming, Development, Application, Triangle Area, Flowchart*

1. PENDAHULUAN

Pemograman java merupakan ilmu yang sangat penting dalam dunia teknologi informasi, pada mata kuliah pemograman dasar, bahasa pemograman yang wajib untuk dikuasai adalah pemograman java. Salah satu solusi yang menarik adalah menggunakan bahasa pemograman java, java adalah sebuah bahsa pemograman yang bersifat open-source yang dikembangkan oleh sun microsystems (sekarang oracle). Keunggulan java terletak pada kemampuan untuk menghasilkan aplikasi yang dapat berjalan sebagai platform dengan menggunakan satu kode sumber.

Bab ini memperkenalkan proyek pengembangan aplikasi sederhana menggunakan java untuk menghitung luas segitiga, menghitung luas segitiga adalah salah satu rumus dasar dalam matematika yang penting,segitiga sendiri adalah sebuah bangunan datar yang terdiri dari tiga sudut, namun masih banyak siswa yang kesulitan untuk memahami konsep ini, dengan adanya kemajuan teknologi yang sangat cangih khususnya dalam bidang perangkat lunak, memungkinkan untuk mengembangkan aplikasi yang dapat membantu siswa dalam memahami dan menghitung luas segitiga dengan lebih mudah.

Meskipun java menawarkan banyak potensi dan manfaat, akan tetapi penelitian ini bertujuan untuk mengembangkan sebuah aplikasi penghitung luas segitiga berbasis Java yang dapat digunakan untuk sebagai alat bantu pendidikan, aplikasi ini diharapkan dapat memberikan kontribusi positif dalam proses pembelajaran materi menghitung luas segitiga

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

2. METODOLOGI PENELITIAN

2.1 Struktur Kode Java

Struktur kode java yang akan digunakan untuk implementasikan algoritma perhitungan java untuk mengimplementasikan algoritma perhitungan dalam java dapat menggunkan struktur dasar sebagai berikut:

2.1.1 Deklarasi Variable

Variable adalah sebuah tempat untuk menapung data dimemori dimana tempat tersebut dapat menapung nilai (data) yang dapat berubah-ubah selama proses program, variable juga disebut sebagai sebuah identifiter yang mempunyai nilai dinamis,dinamis adalah nilai variable yang dapat kita ubah sesuai kebutuhan dalam program.

Variable dibedakan menjadi dua yaitu;

a. Variable numeric adalah variable yang megukur atau menghitung suatu data
b. Variable text adalah sebuah nama yang mewakili sebuah nilai, variable bisa diisi dengan berbagai macam nilai seperti string (text), number (angka), objek array.

2.1.2 Proses Perhitungan

Proses perhitungan adalah sebuah proses yang disengaja untuk mengubah satu masukan atau lebih ke dalam hasil tertentu, dengan jumlah apapun, untuk melakukan operasi atau proses sesuai dengan algoritma yang ingin diimplementasikan contohnya adalah penjumlahan ‘a + b’.

2.1.3 Output Hasil

Output adalah hasil dari suatu proses yang diminta melalui proses input, baik berupa gambar, suara, atau hasil lainnya, tanpa output tidak mungkin mengetahui apakah perintah yang diminta dapat berjalan dengan baik atau tidak.

2.1.4 Jenis dan Contoh Output

a. Output Device

Adalah perangkat yang digunakan untuk menampilkan hasil dari proses yang dilakukan kepada komputer untuk menyampaikan informasi kepada pengguna atau sistem lain, dalam bentuk yang dapat diterima oleh pengguna atau perangkat.

a) Monitor berfungsi untuk menampilkan hasil yang dapat disaksikan baik untuk program aplikasi
b) Printer adalah bagian dari perangkat output yang dapat menampilkan hasil sesuai input user.
c) Speaker adalah output device yang mampu menghasilkan suara setelah sebelumnya diolah oleh program.

b. Output Program

Adalah perangkat yang digunakan untuk menampilkan hasil input atau output program komputer yang untuk mengeluarkan atau menampilkan hasil dari sebuah proses yang dilakukan.

a) web pencarian adalah salah satu produk output digital yang dapat menampilkan hasil input dari pencarian web yang dimasukan.
b) error website adalah contoh hasil output program berupa pesan ketika ada kesalahan program yang dijalankan, contoh error website yang paling sering dijumpai adalah ketika terdapat kesalahan syntax, yang bermunculan output pesan error.

Variable-variabel sesuai dengan algoritma yang ingin diimplementasikan, pastikan struktur dasar ini sudah termasuk dalam kelas utama (‘main’) dan metode ‘main’ yang diperlukan dalam java untuk menjalankan program.

2.1.5 Implementasi Algoritma dari *Flowchart*

a. Output Device

1. mulai
2. masukan pilihan untuk mencari luas segitiga yang kita inginkan

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

3. 3, setelah itu, masukan pilihan atau pilihan yang tidak terdaftar maka akan muncul eror
4. masukan value parameter pada masing-masing bangun datar, untuk garis persegi
5. masukan nilai sisi untuk persegi panjang, masukan nilai alas untuk nilai panjang dan lebar
6. hitung alas dan sisi untuk bangun datar yang dipilih
7. tampilkan luas dan tinggi datar yang dipilih
8. selesai

Pada pemograman ini mengasumasikan sudah mengetahui nilai dari sisi alas pada segitiga, kita hanya masukan saja dalam program dan mencari nilai luas tinggi segitiga pada variable tersebut. Penyelesaian masalah penghitung luas segitiga dengan sistem algoritma yang relavan.

A. INPUTAN:

```
< / >
Masukan alas : 36
Masukan tinggi : 40
```

B. KELUARAN

```
< / >
Hasil Luas : 720.0
```

b. *Flowchart*

Flowchart untuk menghitung luas segitiga Langkah-langkah dari input sisi dan alas hingga output luas dan tinggi pada segitiga akan ditujukan secara visual dalam flowchart tentang alur program.

Berikut ini adalah penjelasan flowchart pengembangan aplikasi penghitung luas segitiga dengan java:

1. Mulai
2. Identifikasi input (alas, sisi) lalu tentukan output (luas segitiga)
3. Buat diagram alur program dan rancang antarmuka pengguna
4. Pengembangan

1). Mulai Pengembangan

1. Buat class dengan menggunakan:
   Public class
2. Trianglearea
   a. Buat method untuk menginput data (alas dan tinggi)
   b. Buat method untuk menghitung luas
   c. Buat method untuk menampilkan hasil

2). Buat Antarmuka Pengguna

1. Gunakan java untuk ui
2. Tambahkan text untuk input alas dan tinggi
3. Tambahkan tombol hitung
4. Tambah label untuk menampilkan hasil
5. Pengujian

3). Unit Testing

Uji method dengan berbagai input

4). *Integration Testing*

Uji keseluruhan alur program dengan ui

2.1.6 Evaluasi dan Debugging

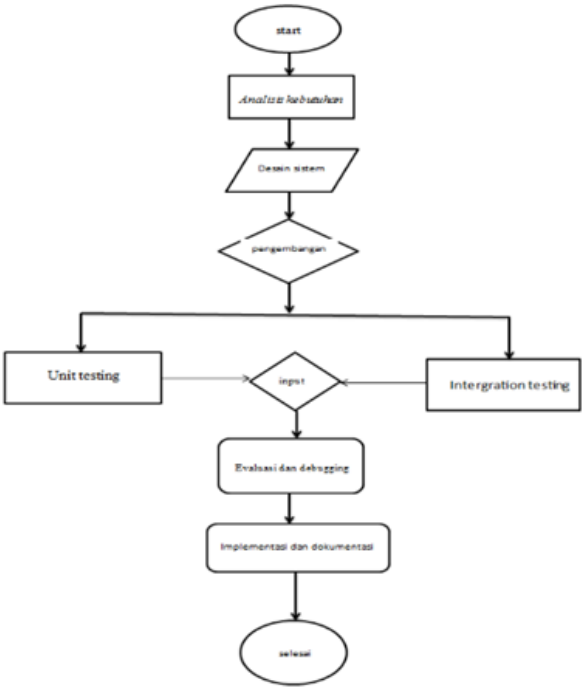
a. Analisis hasil pengujian
b. Perbaiki bug yang ditemukan
c. Optimalkan kinerja aplikasi

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

2.1.7 Implementasi dan Dokumentasi

a. Deploy aplikasi
b. Buat dokumentasi pengguna dan pengembang

2.1.8 Selesai



Gambar 1. *Flowchart*

3. ANALISA DAN PEMBAHASAN

3.1 *Java Codingan*

```
package calculatorsegitiga;
public class NewJFrame extends javax.swing.JFrame {
    public NewJFrame() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        a = new javax.swing.JTextField();
        b = new javax.swing.JTextField();
        c = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
```

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

```
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("MENGHITUNG LUAS SEGITIGA");

        jLabel2.setText("ALAS      :");
        jLabel3.setText("TINGGI   :");
        jButton1.setText("HITUNG");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel4.setText("LUAS      :");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 359,
Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(layout.createSequentialGroup()
                .addGap(38, 38, 38)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(c, javax.swing.GroupLayout.PREFERRED_SIZE, 237,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel3)
                            .addComponent(jLabel2))
                        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(a)
                            .addComponent(b, javax.swing.GroupLayout.PREFERRED_SIZE, 237,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
```

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(a, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(b, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addComponent(jButton1)
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
                .addComponent(c, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(19, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        double alas = Integer.parseInt(a.getText());
        double tinggi = Integer.parseInt(b.getText());
        double luas = 0.5 * alas * tinggi;
        c.setText(luas+"");
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
```

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

```
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTextField a;
    private javax.swing.JTextField b;
    private javax.swing.JTextField c;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    // End of variables declaration
}
```

4. IMPLEMENTASI

4.1 Pengujian Aplikasi

Merupakan bagian penting dalam silkus pengembangan perangkat lunak untuk memastikan bahwa aplikasi berfungsi dengan baik dan memenuhi syarat yang telah ditetapkan, pengujian aplikasi didefinisikan sebagai perangkat lunak yang dilakukan melalui skrip dengan motif menemukan kesalahan pada perangkat lunak.

Untuk pengujian aplikasi ini melibatkan berbagai fase yang mencakup analisis persyaratan, perencanaan pengujian, analisis pengujian, desain pengujian, pelaksanaan pengujian dan pelaporan bug, dll.

Siklus hidup pengujian aplikasi melibatkan empat tahap yaitu:

1. Tahap pertama yaitu rancang rencana pengujian berdasarkan persyaratan aplikasi
2. Tahap kedua yaitu kembangkan kasus pengujian manual dan skrip pengujian otomatis
3. Tahap ketiga jalankan tes fungsional untuk memvalidasi persyaratan aplikasi
4. Tahap ke empat jalankan tes beban dan sesuaikan kinerja aplikasi.

Memilih strategi yang tepat untuk Pengujian Aplikasi adalah cara yang terjamin untuk mendeteksi cacat pada aplikasi. Jadi, menjadi sangat penting bagi tim QA untuk mengikuti serangkaian proses standar untuk mendeteksi lebih banyak kesalahan dan dengan waktu yang lebih sedikit.

Untuk pengujian aplikasi, beberapa praktik terbaiknya meliputi

- Tentukan spesifikasi fungsional
- Review dan Inspeksi
- Kriteria Masuk dan Keluar Formal
- Variasi tes fungsional
- Pengujian multi-platform
- Eksekusi tes otomatis

4.2 *Debugging*

Adalah proses mengidentifikasi dan menghapus bug atau error di dalam kode, karena sistem pengkodean suatu program itu rumit dan kompleks, maka satu saja kesalahan kode dapat berpengaruh pada keseluruhan program, debugging adalah kegiatan rutin baik sebelum perilisan aplikasi maupun sesudahnya.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482

MENGAPA DEBUGGING PERLU DILAKUKAN?

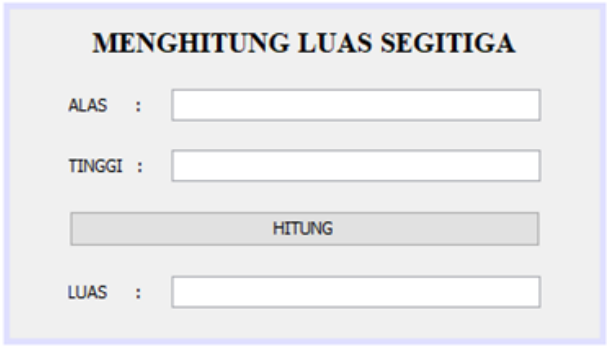
Karena untuk meningkatkan kualitas sistem secara mengidentifikasi dan meyelesaikan bug secara berkelanjutan untuk mendapatkan mengikatkan kualitas secara keseluruhan. Mengurangi sistem downtime untuk membuat sistem software bisa lebih stabil dan mengurangi kemungkinan downtime. Memahami sistem dengan lebih baik dapat membantu developer memahami lebih baik secara kerja sistem software dan bagimana berbagi komponan sistem berinteraksi satu sama lain. Memfasilitasi perubahan pada software karena bisa mengidentifikasi dan memperbaiki bug yang mungkin disebabkan oleh perubahan. Memfasilitasi testing menyelesaikan debugging memudahkan developer untuk menguji perangkat lunak untuk memastikannya memenuhi persyaratan.

Proses debugging biasanya melibatkan langkah-langkah seperti:

1. **Reproduksi Bug**, yaitu Mengulangi langkah-langkah atau situasi yang menyebabkan bug muncul untuk memastikan keberadaannya.
2. **Pencarian Bug**, yaitu Mengidentifikasi lokasi atau bagian kode yang menyebabkan bug. Ini bisa dilakukan dengan memeriksa log, menggunakan tools debugging, atau memeriksa kode secara langsung.
3. **Analisis Bug**, yaitu Memahami alasan munculnya bug dan dampaknya terhadap aplikasi.
4. **Perbaikan Bug**, yaitu Mengimplementasikan perubahan dalam kode untuk menghilangkan bug tersebut.
5. **Verifikasi**, yaitu Mengonfirmasi bahwa perubahan yang dilakukan berhasil memperbaiki bug tanpa menimbulkan dampak buruk lainnya.

4.3 Evaluasi Hasil

Berdasarkan tujuan dari pengembangan aplikasi penghitung luas segitiga dengan java



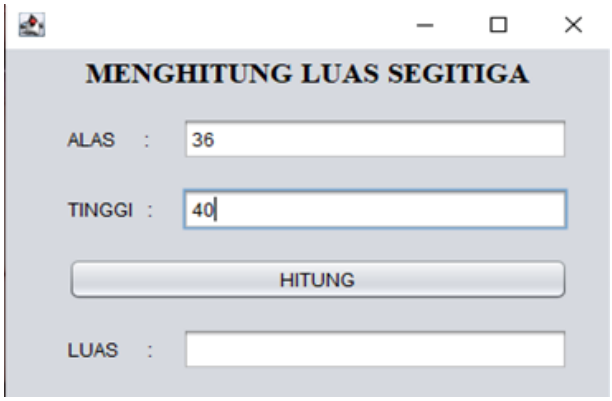
Gambar 2. Evaluasi Hasil

Disini kita dapat memilih hendak melakukan perhitungan pada bangun apa, karena disediakan pilihan seperti hasil yang diatas tersebut.

Gambar 3. Evaluasi Hasil

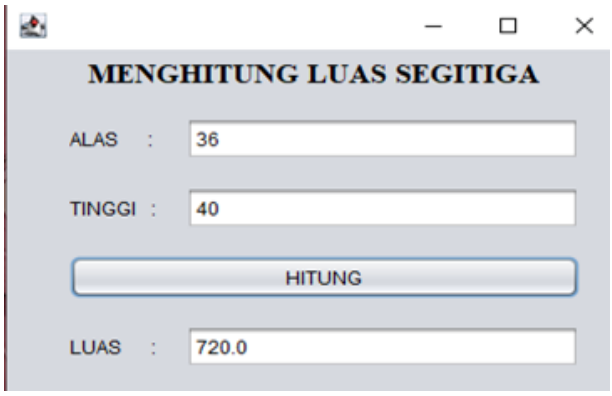
Disini kita melakukan pilihan untuk mengisi alas.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 3, No. 10, Oktober 2024
ISSN 2828-2442 (media online)
Hal 2474-2482



Gambar 4. Evaluasi Hasil

Lalu langkah selanjutnya kita mengisi tinggi.



Gambar 5. Evaluasi Hasil

Lalu langkah berikutnya kita klik hitung untuk mengetahui hasil nilai luas tersebut.

5. KESIMPULAN

Bab ini memberikan saran untuk pengembangan selanjutnya dari aplikasi, seperti penambahan fitur tambahan atau perbaikan yang mungkin diperlukan, pengembangan aplikasi penghitung luas segitiga dengan berbasis java ini telah berhasil dilakukan dengan menggunakan model pengembangan waterfall.

Bagai fitur matematika dasar hingga lanjutan berhasil diimplementasikan dengan baik, memberikan pengalaman yang optimal, pengguna aplikasi ini untuk mempermudah mahasiswa dalam menghitung luas segitiga. Pengujian kinerja menunjukan bahwa aplikasi berjalan dengan cepat dan stabil.

REFERENCES

Harahap, F., dkk. (2023). *Pemgembangan Aplikasi Kalkulator Multifungsi Menggunakan Flutter Framework. Jurnal Penelitian Teknologi Informasi Dan Sains, 1(3),* 76-84.

Vitaloca, D., & Badaruddin A. A.,(2021). *Pengembangan Aplikasi Kalkulator Penghitung Luminasi Berbasis Android. Jurnal Media Elektrik, 18(3).*

Yudertha, a., dkk (2021). *Pengembangan Modul Pelatihan Pengenalan Dasar Bahasa pemograman Java Dengan Model Arcs. Computer Based Information System Journal, 4(2).*