Software Requirements Specification

For

(2D TRANSFORMATION OF VECTOR GRAPHIC SYSTEM)

Prepared by :

Imam najibullah HR

Norfateha binti mohammad

Muh syahabudin hlymi

Instructor:

Madam Sari Widya Sihwi

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Project background

1.2 Project goals

1.3 Stakeholders

2. Specific Requirements 2

2.1 Functional requirement

2.2 Non-Functional requirement

2.3 Methods used in requirement gathering

2.4 Activity diagram

2.5 Use case diagram

2.6 Use case description

3. Project Scheduled 3

4. References 4

**1.0 INTRODUCTION**

1. **Latar Belakang (Project Background)**

Geometri dan vektor adalah salah­ satu cabang matematika yang diajarkan di bangku sekolah, Meskipun telah diajarkan sejak jenjang sekolah dasar sampai sekolah menengah, namun pada kenyataannya geometri dan vektor masih dianggap sebagai materi yang sulit.

Salah satu penyebab dari kenyataan ini adalah kurangnya kemampuan visualisasi yang baik. Tidak cukup hanya kemamuan visualisasi, namun juga kemampuan visual thinking. Visual thinking adalah suatu pemikiran yang aktif dan proses analitis untuk memahami, menafsirkan dan memproduksi pesan visual, interaksi antara meliihat, membayangkan, dan menggambarkan sebagai tujuan dapat digunakan, dan canggih seperti berpikir verbal.

Visual thinking memegang peranan penting dalam keberhasilan pembelajaran geometri dan vektor. Sebab siswa yang belajar tanpa mengandalkan visual thinking, rawan mengalami miskonsepsi (kesalahan konsep). Sebagaimana kasus yang sering ditemui guru matematika dimana siswa terbias memahami rumus secara terpisah dengan objek geometrinya. Akibatnya siswa hanya menghafal rumus tanpa mengetahui kaitannya dengan objek geometrinya.

Transformasi geometri merupakan salah satu bahasan yang terdapat dalam materi geometri. Dalam komputer grafik pemahaman mengenai transformasi geometri dan vektor akan dibutuhkan. Pemahaman mengenai operasi-operasi vektor tidak hanya dibutuhkan di jenjang sekolah sebagai materi pelajaran saja, namun akan sangat berguna penerapannya dalam bidang-bidang lain yang lebih spesifik seperti fisika, ilmu komputer dan bidang – bidang lain.

1. **Tujuan (Project Goals)**

Aplikasi ini bertujuan untuk mewujudkan visualisasi dalam transformasi geometri dan vektor sebagai sarana pembelajaran yang interaktif dalam meningkatkan kemampuan visual dan visual thinking. Mempermudah guru dalam menjelaskan materi terkait transformasi 2d geometri dan vector.

1. **Stakeholders**
2. Guru
3. Dosen
4. Siswa
5. Umum

**2.0 SPECIFIC REQUIREMENTS**

1. **Functional Requirement**
2. The system will show the transformation of matrix on graphic system and connection that will related between mathematics and geometrical. The rules system of geometrical transformation must follow the list type of transformation which is consists of Translation, Rotation, Dilation, Reflection, and Shear.
3. The system should be able to rotate the position of vector s(x,y) which is able to makes an angle x-axis after the transformation to s’(x’,y’) to makes an angle Ɵ degrees. The degrees of rotation only can rotate 90°, 180°, 270° degrees.
4. The system can translate each object to a different position and this will be translated each point in 2D graphic transformation. To get a new coordinate, coordinate (Px, Py) will be added to the original coordinate (X,Y).
5. This system should able to do the shear transformation which is can slants the shape for each object on the two coordinates of (X, Y).
6. This system can reflect the line passing through the origin. It will show reflection between X and Y axis.
7. The system should able to sketch the Vector, Point, Line, Circle, and make shape when the user insert the coordinates of vector
8. **Non-Functional Requirement**
9. Performance

-Performance provide a limit operational speed of system such as the needed of respond, timing and needed of how the system will develop. This is what should system response times be and under what circumstances.

1. Usability

-Usability is related with interaction between users and system. The structure of the user manual and the consistent of interface can increase the needed of usability. Frequently the used of functions should be tested for usability.

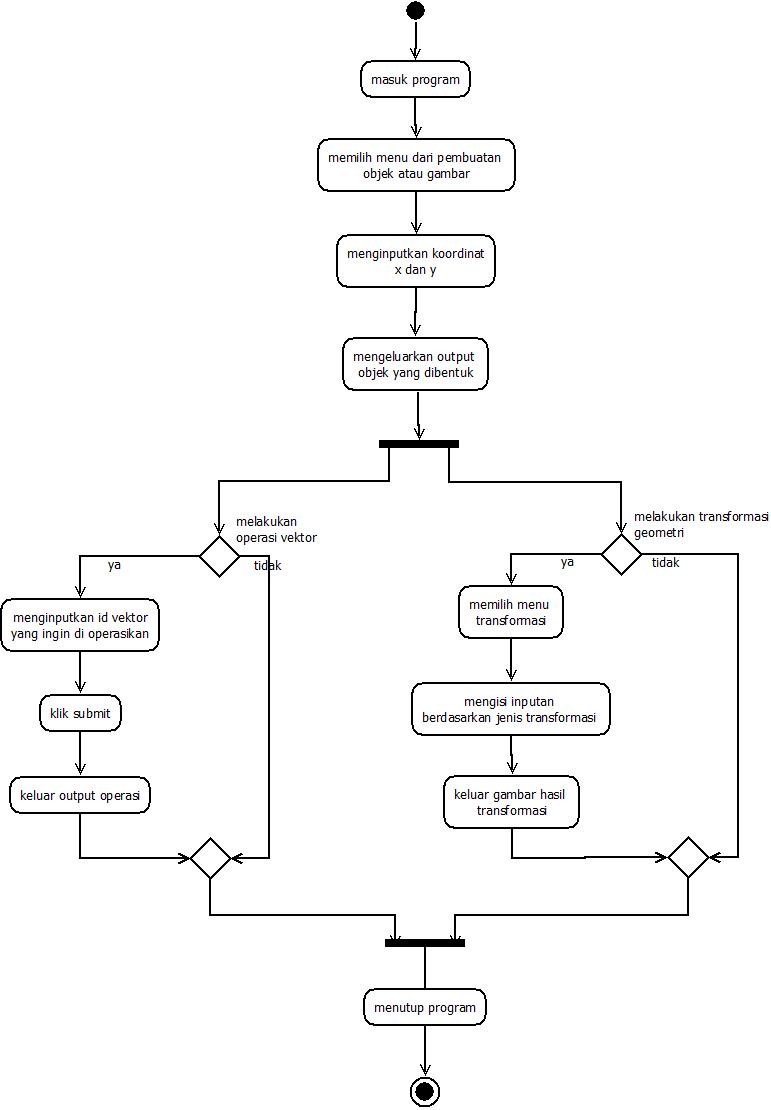
1. Testability

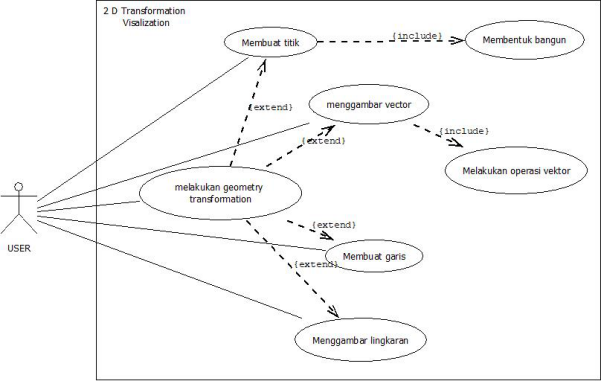
-Testability is determined by factors of controllability, observeability, understandability and Heterogeneity. The requirement of testability needs the criteria such as consistent, complete, quantitative and verification.

1. **Method used in requirement gathering**
2. Method that will be using in 2D transformation graphic system consists of big bang model. This method is ideal to the skills of multi-disciplined or full stack developers.
3. This method has no planning documentation or formalized requirements but it will allows to get straight into the code and focus on their development aspects and design of the system.
4. This method simply producing something tangible and functional in relatively short period time.
5. This method is ideal to apply because it will involve two or three developers working together to develop or create the projects.

ANALYSIS MODEL

1. **Activity Diagram**



1. **Use Case Diagram**
2. **Use Case Description**

|  |  |
| --- | --- |
| Nama Use-case | Membuat titik |
| Aktor | User |
| Dekskripsi | Membuat titik pada koordinat tertentu pada lembar kerja |
| Normal Course | 1. User : Menginputkan koordinat x dan y untuk menentukan dimana titik dibuat. 2. Relasi berupa incluude ke use case membentuk bangun. Karena dalam membentuk bangun kita harus mendapatkan titik-titik yang dapat dihubungkan menjadi sebuah bangun |
| Pre-Condition | Membuka program |
| Post-condition | Muncul output berupa titik dan letak koordinat nya |
| Asumption | - |

|  |  |
| --- | --- |
| Nama Use-case | Menggambar vektor |
| Aktor | User |
| Dekskripsi | Membuat vektor dengan arah koordinat tertentu pada lembar kerja |
| Normal Course | 1. User : Menginputkan koordinat x dan y untuk menentukan dimana tujuan arah dari vektor. 2. Relasi berupa include ke use case melakukan operasi vektor. Karena sebelum kita menggunakan fiture operasi vektor kita harus ,enggambar vektor terlebih dahlu |
| Pre-Condition | Membuka program |
| Post-condition | Muncul output berupa vektor dengan arah koordinat yang telah ditentukan |
| Asumption | - |

|  |  |
| --- | --- |
| Nama Use-case | Membuat garis |
| Aktor | User |
| Dekskripsi | Membuat garis pada koordinat tertentu pada lembar kerja |
| Normal Course | 1. User : Menginputkan koordinat x1 dan y2 untuk menentukan dimana titik awal pada garis dan menginputkan x2 dan y2 untuk mendapatkan titik akhir pada garis |
| Pre-Condition | Membuka program |
| Post-condition | Muncul output berupa garis dan letak koordinat nya |
| Asumption | - |

|  |  |
| --- | --- |
| Nama Use-case | Membuat lingkaran |
| Aktor | User |
| Dekskripsi | Membuat lingkaran dengan menentukan pusat dan besar diameter |
| Normal Course | 1. User : Menginputkan koordinat x dan y untuk menentukan pusat lingkaran dan input diameter lingkaran |
| Pre-Condition | Membuka program |
| Post-condition | Muncul output berupa lingkaran dan letak koordinat pusatnya serta besar diameter lingkaran |
| Asumption | - |

|  |  |
| --- | --- |
| Nama Use-case | Melakukan geometri transformation |
| Aktor | User |
| Dekskripsi | Mentransformasikan bidang atau objek yang dibuat sesuai dengan yang diinginkan (translasi, dilatasi, rotasi, refleksi, dan shear) |
| Normal Course | 1. User : user dapat menginputkan inputan berdasarkan pilihan transsformasi, yaitu bentuk apa yang ingin di trasformasikan dan inputan2 lainya 2. Relasi berupa extend ke membentuk titik, garis, lingkaran, menggambar vektor. Karena saat kita melakukan fungsi membuat titik, garis, lingkaran, dan menggambar vektor user dapat memilih ingin melakukan transformasi atau tidak |
| Pre-Condition | Objek output dari fungsi membentuk titik/garis/lingkaran/menggambar vektor |
| Post-condition | Muncul hasil transformasi dari objek yang telah dibuat |
| Asumption | - |

|  |  |
| --- | --- |
| Nama Use-case | Membuat bangun |
| Aktor | User |
| Dekskripsi | Membuat bidang tertentu dengan bantuan titik-titik yang dibuat |
| Normal Course | 1. User : Menginputkan beberapa koordinat x dan y menjadi titik, dan menghubungkan semua titik menjadi sebuah bidang atau bangun |
| Pre-Condition | Mendapatkan titik |
| Post-condition | Muncul output berupa bidang yang telah dibentuk dari gabungan titik |
| Asumption | - |

|  |  |
| --- | --- |
| Nama Use-case | Melakukan Operasi Vektor |
| Aktor | User |
| Dekskripsi | Mengoperasikan vektor yang telah dibuat dengan berbagai macam jenis operasi yaitu : penjumlahan vektor, perkalian vektor, cross product, dan dot product |
| Normal Course | 1. User : menginputkan vektor mana yang ingin diperasikan yaitu berupa id vektor. Selain itu juga menginputkan beberapa inputan sesuai dengan operasi yang diinginkan/dipilih |
| Pre-Condition | Mucul vektor dengan arah yang telah ditentukan |
| Post-condition | Muncul output vektor hasil penjumlahan dan perkalian, hasil skalar untuk dot product dan bidang untuk operasi cross product |
| Asumption | - |

**3.0 PROJECT SCHEDULLE AND EFFORT PLAN**

|  |  |
| --- | --- |
| **Time** | **Activity** |
| **27-28 march** | Planned and specify what the project need to develop |
| **10-13 April** | Mendiskusikan spesifikasi kebutuhan ,menyusun requirement, dan menentukan metode yang digunakan dalam pembuatan aplikasi. |
| **14-15 April** | Specify the functionality of user diagram, user description, and activity diagram |
| **24-30 April** | Mulai coding dan testing, serta membuat dokumentasi dari method dan fungsi yang terdapat dalam aplikasi |
| **28-31 May** | Final testing |

**4.0 References**

We'll start with two dimensions to refresh or introduce some basic mathematical principles. The system that we’ll develop is more relate to how we can use matrices to express transformations and how we can draw the vector in 2D transformation. We will also study how formulation of the matrix operations affects the connection between geometric reasoning and mathematical operations. The link below is to proof how we get the idea to create the transformation system by survey all the sources.

[**https://en.wikipedia.org/wiki/Transformation\_matrix**](https://en.wikipedia.org/wiki/Transformation_matrix)

[**https://www.tutorialspoint.com/computer\_graphics/2d\_transformation.htm**](https://www.tutorialspoint.com/computer_graphics/2d_transformation.htm)

[**http://nptel.ac.in/courses/Webcourse-contents/IIT-Delhi/Computer%20Aided%20Design%20&%20ManufacturingI/mod2/01.htm**](http://nptel.ac.in/courses/Webcourse-contents/IIT-Delhi/Computer%20Aided%20Design%20&%20ManufacturingI/mod2/01.htm)