

Perancangan Multiplatform Aplikasi *Mobile* untuk Penentuan Arah Mortir

Suharjanto Utomo ^{#1}, Iswanto ^{#3}

Progam studi Informatika, Universitas Nurtanio Bandung
Jalan Pajajaran no 219 Bandung

¹ suharjanto.utomo@gmail.com

Received on dd-mm-yyyy, revised on dd-mm-yyyy, accepted on dd-mm-yyyy

Abstract

Saat ini perkembangan perangkat *mobile* semakin pesat. Hal ini dibuktikan semakin banyak penggunaan Android, iOS, an Windows. Dan permintaan aplikasi *mobile* otomatis semakin tinggi. Tentunya diperlukan usaha yang efektif untuk dapat mengembangkan aplikasi yang dapat berjalan dalam beberapa sistem operasi seperti Android, iOS, dan Windows. Salah satu kebutuhan aplikasi *mobile* adalah aplikasi untuk penentuan arah mortir. Hal ini diperlukan untuk menjamin kecepatan dan ketelitian perhitungan dibandingkan dengan perhitungan dengan menggunakan kertas planset. Hasil perhitungan yang diperlukan adalah elevasi dan azimuth daripada mortir. Untuk pengembangan prototipe aplikasi *mobile* dilaksanakan dengan menggunakan Xamarin.

Keywords: multiplatform, aplikasi *mobile*, elevasi, azimuth

I. INTRODUCTION

Mortir merupakan salah satu jenis senjata yang berfungsi untuk memberikan bantuan tembakan pada pasukan infanteri yang berada di garis depan. Untuk melaksanakan tugas dalam pelaksanaan penembakan diperlukan pengetahuan tentang karakteristik mortir. Selain itu juga pengetahuan untuk menentukan arah mortir terhadap sasaran. Secara umum penentuan arah tersebut bisa dilaksanakan dengan bantuan papan kertas peta planset. Pada papan peta tersebut dapat ditarik garis lurus dari posisi mortir ke sasaran untuk mendapatkan arah tembakan dan jarak sasaran. Untuk menentukan arah elevasi nya dapat diketahui dari tabel tembak yang ada. Dengan cara manual ini akan memerlukan waktu yang cukup lama, selain itu ketelitian sangat bergantung pada kemampuan personil.

Piranti bergerak dengan *mobile application* memudahkan pengguna dalam menjalankan aktifitas yang berhubungan dengan layanan perangkat lunak. Pengguna hanya dengan memegang piranti bergerak seperti gawai, dan dengan jari sudah bisa menjalankan aplikasi yang diinginkan. Saat ini perkembangan perangkat bergerak semakin pesat hal ini dapat dilihat dari banyaknya penggunaan Android, iOS, dan Windows Phone.

Pengembangan perangkat lunak dengan kemampuan lintas platform atau multiplatform merupakan tantangan pada saat ini. Kebutuhan aplikasi yang bisa dijalankan pada sistem operasi yang berbeda sangat dibutuhkan pada masa kini. Dalam kesempatan ini penulis akan membuat sebuah perancangan prototipe aplikasi *mobile* dengan lingkungan pengembangan perangkat lunak yang menyediakan kemampuan membuat aplikasi lintas platform atau multiplatform. Prototipe ini yang akan dibangun untuk perhitungan penentuan arah mortir terhadap sasaran. Dalam pembangunan perangkat lunak menggunakan model proses Waterfall dengan pertimbangan kebutuhan sistem sudah jelas sehingga tahapan yang akan dilaksanakan sesuai dengan

kasus yang ada. Model Waterfall merupakan metode yang klasik yang menekankan tahapan yang berurutan dan sistematis.

II. LITERATURE REVIEW

Pada penelitian yang dilakukan oleh Mounim Latif, Younes Lakhri, El Habib Nfaoui, dan Najia Es-Sbai, pada tahun 2016 dengan judul “Cross platform approach for mobile application development : A survey”, dalam penelitian ini menjelaskan berbagai pendekatan dalam membangun aplikasi *mobile* dengan lingkungan pengembangan perangkat lunak yang punya kemampuan *cross platform* atau multiplatform. Salah satu bentuk dalam multiplatform adalah *Cross-Compiled Approach* yang mana pengembang menulis kode dengan menggunakan bahasa pemrograman yang umum. Salah satu platform yang digunakan adalah Xamarin dengan bahasa C#.

Pada penelitian yang dilakukan oleh Mrs. V.C. Kullooli, Ashih Pohare, Sujit Raskar, Tania Bhattacharyya, Shashikant Bhure pada tahun 2013 dengan judul “Cross Platform Mobile Application Development”, dalam penelitian ini menjelaskan bagaimana *single source code* dapat jalan pada semua sistem operasi perangkat bergerak atau *mobile*. Untuk implementasi aplikasi dilakukan dengan PhoneGap Framework.

Pada penelitian yang dilakukan oleh Hammoudeh S. lamri, Balsam A. Mustafa pada tahun 2014 dengan judul “Software Engineering Challenges in Multi Platform Mobile Application Development”, menjelaskan tantangan pengembangan aplikasi *mobile* di tengah permintaan yang semakin meningkat dari waktu ke waktu. Salah satu pendekatan yang diusulkan adalah multi platform yaitu dengan berbagai metode yaitu *Cross-compiler*, *Virtual machine*, Web based, dan Hybrid.

Pada penelitian ini penulis merancang sebuah aplikasi *mobile* dengan kemampuan multiplatform untuk penentuan arah mortar. Penentuan arah mortar berupa data sudut azimuth dan sudut elevasi. Untuk framework yang dipilih dengan menggunakan Xamarin dari Microsoft Visual Studio 2017 Community.

III. RESEARCH METHOD

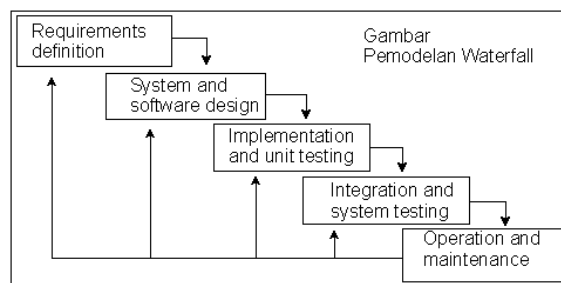
3.1. Multiplatform

Dalam pengembangan aplikasi, pengembang perangkat lunak menggunakan lingkungan pengembangan sesuai dengan target platform yang spesifik. Aplikasi hanya dihubungkan dengan lingkungan tunggal saja. Sebagai contoh adalah aplikasi Android yang mana diprogram dengan menggunakan bahasa Java. Begitu dengan iOS yang menggunakan Objective-C atau Swift untuk membuat sebuah aplikasi. Kesulitan dalam mengembangkan berbagai lingkungan pengembangan perangkat lunak maka muncul pengembangan dengan lintas platform atau multiplatform yang akan memudahkan dan efisien.

Pengembangan lintas platform merupakan solusi untuk permasalahan multi platform maka diperlukan teknologi terpadu yang kompatibel terhadap berbagai sistem operasi perangkat bergerak [4]. Teknologi untuk pengembangan multiplatform yang terpadu semakin berkembang seperti Xamarin dan Phone-Gab [2,3,4].

3.2. Model Pengembangan Sistem

Nama model ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut dengan “classic life cycle” atau model waterfall. Model ini adalah model yang muncul pertama kali yaitu sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam Software Engineering (SE). Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing / verification, dan maintenance. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap requirement. Secara umum tahapan pada model waterfall dapat dilihat pada gambar berikut :



Gambar 1 Model Waterfall

Gambar di atas adalah tahapan umum dari model proses ini. Akan tetapi Roger S. Pressman memecah model ini menjadi 6 tahapan meskipun secara garis besar sama dengan tahapan-tahapan model waterfall pada umumnya. Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model ini menurut Pressman:

- a) *System / Information Engineering and Modeling*
Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk software. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware, database, dsb. Tahap ini sering disebut dengan Project Definition.
- b) *Software Requirements Analysis*
Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dsb. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada pelanggan.
- c) *Design*
Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk "blueprint" software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.
- d) *Coding*
Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer.
- e) *Testing / Verification*
Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsi-fungsi software harus diujicobakan, agar software bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.
- f) *Maintenance*
Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada errors kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

3.3. Mortir

Mortir adalah senjata artileri yang diisi dari pasukan depan, dan menembakkan peluru dengan kecepatan yang rendah, jarak yang jangkauan dekat, dan dengan perjalanan peluru yang tinggi lengkungan parabola. Meski sudah digunakan sejak era Perang Dunia I, nyatanya perkembangan teknologi turut mempertahankan

keberadaan mortir sebagai unsur bantuan tembakan infanteri yang cukup diandalkan hingga kini. Bentuknya yang ringkas membuat mortir tak bisa dilepaskan dari setiap pergerakan infanteri. Dengan bentuknya yang ringkas, mortir mudah dibawa ke sana ke mari bersama pergerakan pasukan infanteri. Manfaat ini terasa betul bagi pasukan khusus yang bergerak di belakang garis musuh, yang seringkali tidak memiliki dukungan tembakan artileri.[6].

IV. RESULTS AND DISCUSSION

4.1. Rancangan Sistem

Perancangan antarmuka aplikasi ini dapat dilihat pada gambar 2, ada beberapa komponen yang digunakan dalam antarmuka yaitu label, teks, option, dan button. Secara umum dalam antarmuka bertujuan memberikan kemudahan dalam pengoperasian aplikasi. Bagian atas berupa input yang memberikan gambaran akan posisi mortir dan posisi sasaran. Dan bagian bawah merupakan hasil perhitungan sesuai dengan input yang diberikan. Untuk memperoleh hasil perhitungan harus dilakukan klik pada tombol Hitung.

Mesin Hitung Mortir 81 MM

Arah ke TD perluasan Jarak ke TD meter

Permintaan Tembakan

Sudut Kompas perluasan

Arah Samping Kanan meter
 Kiri

Arah Jarak Maju meter
 Mundur

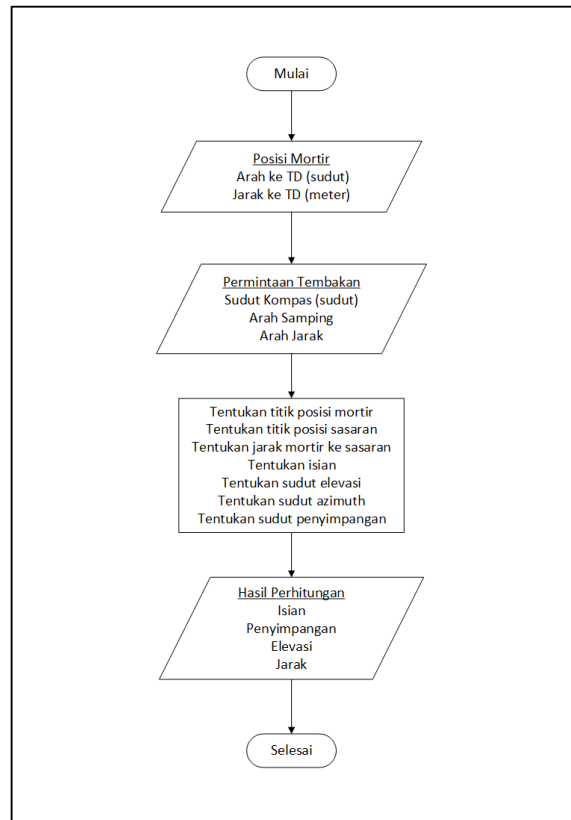
HITUNG

Isian :
Penyimpangan :
Elevasi :
Jarak :

Hasil Perhitungan

Gambar 2 Desain Antarmuka

Untuk memudahkan dalam menentukan hasil perhitungan maka dibuat urutan langkah dari mulai memasukkan nilai input posisi mortir, permintaan tembakan, kemudian adanya beberapa proses sebelum didapatkan hasil perhitungan. Proses secara berurutan adalah menentukan posisi mortir, menentukan posisi sasaran, menentukan jarak mortir ke sasaran, menentukan isian mortir, menentukan sudut elevasi mortir ke sasaran, menentukan sudut azimuth mortir ke sasaran, dan menentukan sudut penyimpangan. Kemudian hasil perhitungan adalah isian, sudut penyimpangan, elevasi, dan jarak.

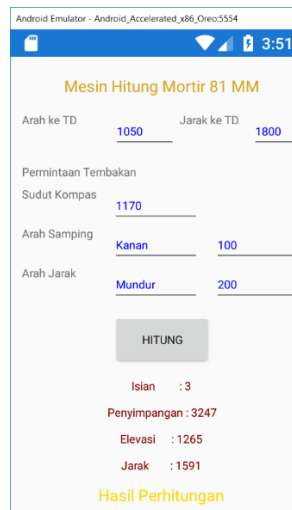


Gambar 3 Flowchart Perhitungan Arah Mortir

4.2. Implementasi

Prototipe aplikasi multiplatform ini memanfaatkan Xamarin Form Visual Studio 2017 Community. Dengan menggunakan Xamarin Forms dapat menggunakan satu User Interface yang dapat digunakan pada platform yang berbeda. Dengan Xamarin Forms cukup dikembangkan satu aplikasi dengan target platform yang berbeda. Xamarin Forms menggunakan format XAML (eXtensible Application Markup Language) untuk membuat tampilan User Interface.

Pada tampilan aplikasi perhitungan arah mortir terdiri dari bagian input dan bagian output. Bagian input ada 2 hal pokok yaitu input berhubungan dengan posisi mortir ditinjau dari titik dasar dan input berkaitan dengan permintaan tembakan pada titik sasaran. Untuk berhubungan dengan posisi mortir berupa input yang terdiri isian untuk arah ke Titik Dasar (TD) dan Jarak ke Titik Dasar. Sedangkan input berkaitan titik sasaran berupa sudut kompas, pilihan arah samping apakah kanan atau kiri, besarnya arah samping, pilihan arah jarak apakah maju atau mundur, besarnya jarak dalam meter. Untuk output berupa isian, penyimpangan, elevasi, dan jarak. Untuk lebih jelasnya dapat dilihat pada gambar 4 di bawah ini.



Gambar 4 Tampilan Antarmuka

V. Conclusion

Dengan adanya prototipe aplikasi penentuan arah mortir dapat memudahkan awak mortir dalam pelaksanaan tugas penembakan. Prototipe dapat melakukan perhitungan lebih cepat dan teliti dibandingkan dengan menggunakan perhitungan kertas planset.

ACKNOWLEDGMENT

Penelitian ini didanai oleh Kementerian Riset Teknologi dan Pendidikan Tinggi (Ristekdikti) melalui Program Penelitian Dosen Pemula. Terima kasih banyak disampaikan kepada Ristekdikti.

REFERENCES

- [1] Hammou Hammoudeh S. Alamri, Balsam A. Mustafa, "Software Engineering Challenges in Multi Platform Mobile Application Development", Journal of Computational and Theoretical Nanoscience May 2014.
- [2] Mrs. V.C. Kullooli, Ashish Pohare, Sujit Raskar, Tania Bhattacharyya, Shashikant Bhure, "Cross Platform Mobile Application Development", International Journal of Computer Trends and Technology (IJCTT) volume 4 Issue 5 May 2013.
- [3] Paulo Roberto Martins de Andrade, Fatima Aguiar da Silva, "An Empirical Study Mobile Multi Platform Development Using Phone Gap and HTML5", International Journal of Advanced Information Technology (IJAIT) Vol. 6, No. 1, February 2016.
- [4] K. Samuel David, "An Analysis of Cross Platform Mobile Application Development", International Journal of Advanced Engineering Research and Application (IA-ERA) Volume-2, Issue-8, Desember 2016.
- [5] Safaat, N. Android "Pemrograman aplikasi mobile smartphone dan tablet PC berbasis Android". Bandung: Informatika 2011
- [6] <http://www.indomiliter.com/mortir-81mm-mobilitas-tinggi-senjata-andalan-bantuaninfanteri/> diakses 12 April 2018.
- [7] Mounaim Latif, Younes Lakhriissi, El Habib Nfaoui, Najia Es-Sbai, "Cross platform approach for mobile application development: a survey", International Conference on Information Technology for Organizations Development (IT4OD) 2016