

## Implementasi *Linked List* pada Interaksi Antar *Marker Augmented Reality* untuk *Operand* dan *Operator Aritmetika*

Hermawan Wijaya<sup>1</sup>, Wibisono Sukmo Wardhono<sup>2</sup>, Issa Arwani<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>hermawanwijaya21@gmail.com, <sup>2</sup>wibiwardhono@ub.ac.id, <sup>3</sup>issa.arwani@ub.ac.id

### Abstrak

*Augmented Reality* (AR) merujuk kepada suatu persepsi langsung terhadap lingkungan sekitar yang sudah ditambahkan dengan komponen virtual komputer. Pada umumnya tidak terdapat mekanisme bawaan untuk menyimpan urutan pada *framework* AR, yang pada konteks ini adalah Vuforia SDK. *Linked List* merupakan salah satu metode dalam Algoritma dan Struktur Data yang digunakan untuk mengorganisasi struktur data dalam sebuah sistem. Urutan *operand* dan *operator* sangat diperhatikan dalam operasi aritmetika agar dapat melakukan kalkulasi aritmetika dengan benar, terutama untuk operasi yang tidak bersifat komutatif maupun asosiatif. Penelitian ini dilakukan untuk menerapkan penggunaan *Linked List* ke dalam AR sebagai sebuah metode untuk menyimpan urutan sementara *marker* yang berhasil dibaca perangkat untuk kemudian diolah lagi berdasarkan urutannya sebagai operasi aritmetika sederhana menggunakan bantuan *engine* Unity 3D. Dari penelitian ini juga dilakukan pengujian *whitebox* dengan nilai hasil *Cyclomatic Complexity* adalah 9, yang menyatakan sistem sudah memenuhi kebutuhan sesuai rancangan dengan kompleksitas yang rendah, serta pengujian *usability* dengan nilai hasil SUS rata-rata adalah 55 yang berarti sistem belum layak untuk digunakan pada pengguna awam.

**Kata kunci:** *Augmented Reality, Linked List, Struktur Data.*

### Abstract

*Augmented Reality* (AR) refers to a direct perception of the surrounding environment that has been added to the virtual component of the computer. There is generally no default mechanism for storing sequences in the AR framework, which in this context is Vuforia SDK. *Linked List* is one of the methods in Algorithm and Data Structure used to organize data structures in a system. The operand and operator sequences are highly considered in arithmetic operations in order to perform arithmetic calculations correctly, especially for non-commutative or non-associative operations. This research was conducted to apply the use of *Linked List* into AR as a method to store the temporary sequence of marker that successfully read by the device then processed again in sequence as simple arithmetic operation using Unity 3D engine aid. From this research also conducted *whitebox* testing with value of result of cyclomatic complexity is 9, which states the system meets the needs according to the design with low complexity, and usability testing with average SUS yield value is 55 which means the system is not yet feasible for use on the novice user.

**Keywords:** *Augmented Reality, Linked List, Data Structure.*

## 1. PENDAHULUAN

*Augmented Reality* (AR) merujuk kepada suatu persepsi langsung terhadap lingkungan sekitar yang sudah ditambahkan dengan komponen virtual komputer. Beberapa contoh penggunaan AR dalam kehidupan sehari-hari yang sudah tidak asing lagi didengar antara lain Google Glass dan HTC Vive. Perkembangan

pesat pada AR memungkinkan perkembangan di banyak sektor, antara lain investasi ekonomi, aplikasi, ekspektasi komersial, dan perkembangan sains (Raja, 2016).

Dalam implementasinya, tracking pada AR dibagi menjadi *Marker-based* dan *Markerless Tracking* (Petersen, 2015). *Marker-based AR tracking* membutuhkan media tambahan untuk digunakan sebagai markernya, seperti kartu atau foto. Penelitian ini akan menggunakan *Marker-*

based tracking untuk aplikasi perhitungan aritmetika sederhana.

Pada umumnya tidak terdapat mekanisme bawaan untuk menyimpan urutan pada *framework* AR. *Linked List* merupakan salah satu metode dalam Algoritma dan Struktur Data yang digunakan untuk mengorganisasi struktur data dalam sebuah sistem. *Linked List* memiliki keunggulan dibandingkan *Array* dalam hal menambahkan dan mengurangi elemennya (Ryan, 2008). Urutan *operand* dan *operator* sangat diperhatikan dalam operasi aritmetika agar dapat melakukan kalkulasi aritmetika dengan benar, terutama untuk operasi yang tidak bersifat komutatif maupun asosiatif (Tapson, 2006)

Penelitian ini bertujuan untuk menampilkan model 3D berdasarkan *markernya* serta menerapkan penggunaan *Linked List* ke dalam sistem sebagai sebuah metode untuk menyimpan urutan sementara *marker* yang berhasil dibaca perangkat untuk kemudian diolah lagi berdasarkan urutannya sebagai operasi aritmetika sederhana. Pengujian *usability* juga dilakukan untuk mengetahui seberapa mudah sistem digunakan oleh pengguna.

Beberapa pustaka yang digunakan antara lain pustaka mengenai *Augmented Reality*, *Linked List*, operasi aritmetika, kalkulator *postfix*, *Vuforia*, *White Box Testing*, serta *Usability Testing*.

## 2. METODOLOGI

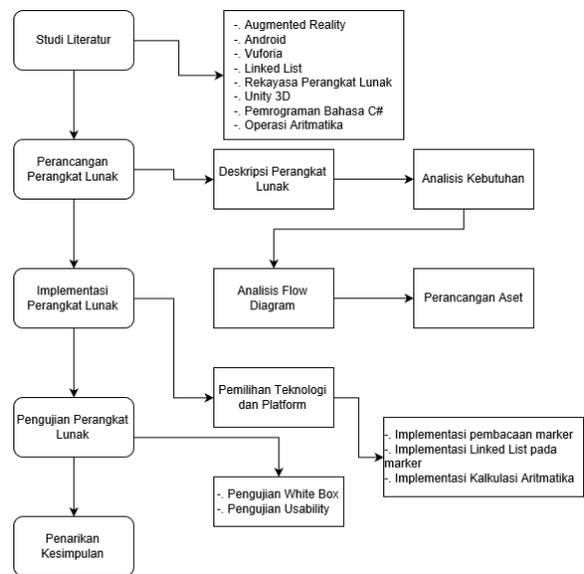
Pada bagian ini akan dijelaskan mengenai prosedur dan kegiatan-kegiatan yang akan dilakukan dalam pengerjaan skripsi. Dimulai dari studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian perangkat lunak, dan penarikan kesimpulan yang dapat dilihat di Gambar 1.

## 3. PERANCANGAN

### 3.1 Deskripsi Perangkat Lunak

Aplikasi ini memiliki tujuan untuk membaca urutan *marker* dan mendaftarkan urutan tersebut pada *Linked List* untuk diolah lagi. Cara penggunaan aplikasi ini adalah dengan mengarahkan kamera ke arah *marker* satu per satu. Pengguna akan mendapatkan informasi yang ditampilkan pada layar perangkat, antara lain model 3D yang ditampilkan saat *marker* terdeteksi, teks *marker* apa yang terdeteksi saat ini, konten dari *Linked List* saat ini, notifikasi

untuk error maupun hasil perhitungan, jumlah objek yang terdeteksi saat ini, serta hasil konversi dari notasi *infix* ke *postfix*.



Gambar 1 Diagram Blok Penelitian

### 3.2 Analisis Kebutuhan

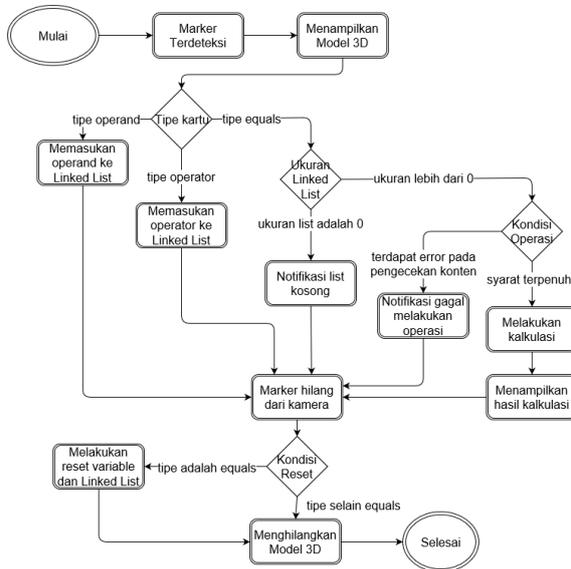
Berikut merupakan beberapa poin yang dibutuhkan dalam aplikasi yang dikembangkan.

1. Sistem dapat mengakses kamera perangkat yang digunakan.
2. Sistem dapat melakukan deteksi terhadap *marker* dan menampilkan model sesuai *markernya*.
3. Sistem dapat memasukan nilai dari *marker* operand dan operator ke dalam *Linked List*.
4. Sistem dapat menampilkan notifikasi ketika *marker equals* terbaca, yang berisi pemberitahuan list kosong, operasi gagal, ataupun hasil dari perhitungan aritmetika.
5. Sistem dapat melakukan kalkulasi dari masukan yang berupa *Linked List* dengan konten notasi *infix*.
6. Sistem dapat melakukan perhitungan aritmetika dengan melakukan konversi notasi *infix* ke notasi *postfix*.
7. Sistem dapat menghilangkan model 3D apabila *marker* tidak terdeteksi oleh kamera.
8. Sistem mampu melakukan reset terhadap konten *Linked List* dan variabel-variabel yang digunakan apabila *marker* yang hilang bertipe *equals*.

### 3.3 Analisis Flow Diagram

Gambar 2 merupakan analisis *flow* diagram dari sistem yang dikembangkan. Diagram

tersebut menggambarkan cara kerja aplikasi yang dikembangkan. Proses yang digambarkan dimulai saat *marker* terdeteksi hingga saat *marker* hilang dari pandangan kamera perangkat.



Gambar 2 Analisis Flow Diagram

### 3.4 Perancangan Aset

Aset dirancang dengan menggunakan software Inkscape untuk *Marker* 2D dan Blender untuk Model 3D. Aset dibuat se-sederhana mungkin agar dapat mudah diterima oleh pengguna namun tetap memiliki tingkat *augmentability* yang baik.

## 4. IMPLEMENTASI

Setelah tahap perancangan perangkat lunak telah dilakukan, penulis melakukan implementasi terhadap perangkat lunak yang dikembangkan. Pada tahap implementasi perangkat lunak akan dilakukan implementasi terhadap hasil rancangan aplikasi sesuai dengan yang telah dilakukan pada tahap perancangan perangkat lunak. Penelitian ini akan menggunakan teknologi *Augmented Reality* pada *game engine* Unity yang digunakan untuk mengembangkan aplikasi AR aritmetika sederhana ini dengan bantuan SDK Vuforia. Tahap ini terdiri dari pemilihan teknologi dan *platform*, implementasi pembacaan *marker*, implementasi *linked list* pada *marker*, dan implementasi kalkulasi aritmetika.

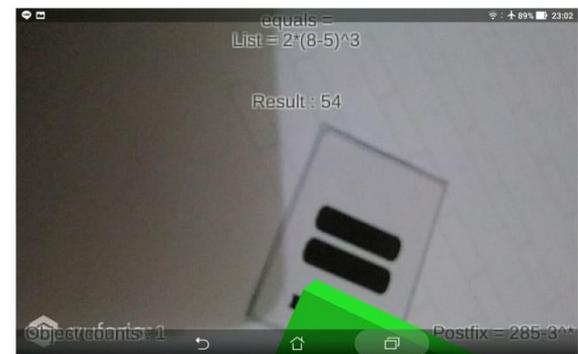
Bagian pemilihan teknologi dan platform menjelaskan kebutuhan perangkat yang digunakan selama proses implementasi. Kebutuhan yang digunakan dibagi menjadi

kebutuhan perangkat komputer untuk proses pengembangan dan kebutuhan perangkat bergerak untuk proses pengujian.

Implementasi pembacaan *marker* dan implementasi *linked list* pada *marker* dilakukan dengan menggunakan bahasa pemrograman C# dengan menggunakan Visual Studio sebagai IDE (*Integrated Development Environment*)nya. Editor Unity 2017.2 juga digunakan selama proses pengembangan aplikasi ini. SDK Vuforia yang digunakan merupakan SDK yang terintegrasi langsung sebagai *plugin* dari Unity 2017.2 itu sendiri. Gambar 3 merupakan tampilan saat *marker* operand 2 terbaca, dan gambar 4 merupakan hasil saat operasi  $2*(8-5)^3$  dilakukan dengan menampilkan *marker* secara urut.



Gambar 3 Marker Operand 2 Terbaca Kamera



Gambar 4 Hasil Operasi Aritmetika

## 5. PENGUJIAN

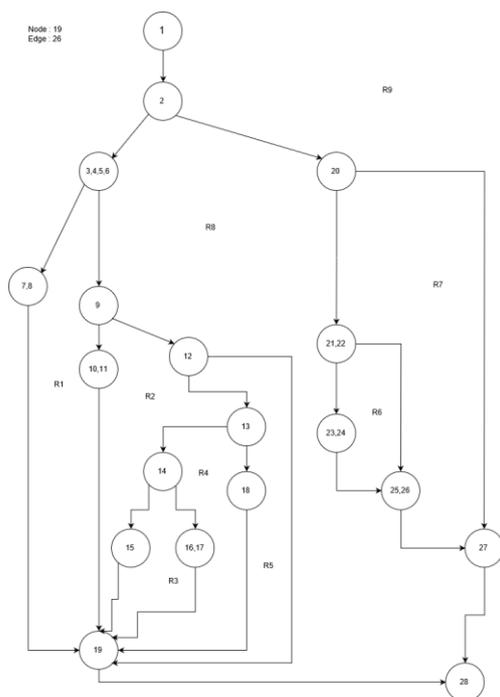
Setelah sistem dan aplikasi diimplementasikan, tahapan selanjutnya yang dilakukan adalah pengujian. Pada tahap pengujian akan dilakukan pengujian terhadap aplikasi AR aritmetika sederhana yang dikembangkan untuk menunjukkan bahwa aplikasi yang dikembangkan dapat bekerja sesuai dengan hipotesa dan harapan penelitian ini. Pengujian yang antara lain *white box testing* dan *usability testing*.

### 5.1 Basis Path Testing

Pengujian ini merupakan pengujian Whitebox dilakukan untuk mengetahui nilai *Cyclomatic Complexity* dari *method* perubahan state *OntrackableStateChanged* pada class *MathARTrackableEventHandler* yang digunakan sebagai dasar dari proses pembacaan dan pemrosesan *marker*. Terdapat beberapa istilah dalam pengujian ini, antara lain *edge* yang merupakan penghubung antar *node*, *region* yang merupakan daerah yang dibatasi oleh *node* dan *edge*, serta *predicate node* yang merupakan *node* yang mengeluarkan dua atau lebih *edge*. Berdasarkan flow graph pada Gambar 5, terdapat 9 *region*, 26 *edges*, 19 *node*, dan 8 *predicate node*. Nilai *Cyclomatic Complexity* dapat dihitung sebagai berikut :

1.  $V(G) = 9$ , karena terdapat 9 *region* (R1, R2, R3, R4, R5, R6, R7, R8, R9).
2.  $V(G) = 26 - 19 + 2 = 9$ , karena terdapat 26 *edges* dan 19 *node*.
3.  $V(G) = 8 + 1 = 9$ , karena terdapat 8 *predicate node*.

Dari perhitungan matematis *Cyclomatic Complexity* -  $V(G)$  berdasarkan *Flow Graph* di atas dapat disimpulkan bahwa nilai  $V(G)$  pada *Flow Graph* di atas adalah 9, yang berarti resiko kompleksitasnya tergolong rendah (Prabhu K, 2014) dan dapat dilakukan pengujian masing-masing jalur independen..



Gambar 5 Flow Graph Method OnTrackableStateChanged

### 5.2 Pengujian Usability

Pengujian *usability* dilakukan untuk mengetahui apakah sistem yang dikembangkan dapat digunakan dengan mudah atau tidak. Hal ini ditujukan untuk memenuhi rumusan masalah yaitu untuk menguji nilai *usability* dari sistem yang dikembangkan. Pengujian *usability* ini dilakukan dengan menggunakan kuesioner *System Usability Scale* atau SUS kepada 6 orang responden berusia 21 tahun yang dipilih secara acak dengan kriteria sebagai berikut :

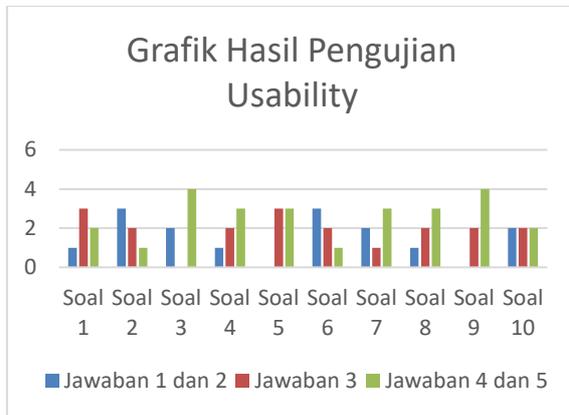
1. Tidak asing dengan menggunakan ponsel Android.
2. Tidak asing dengan operasi kalkulator konvensional.
3. Tidak ikut andil dalam proses pengembangan sistem yang dibuat.
4. Tidak mengerti secara pasti cara kerja sistem yang dibuat.

Responden dijelaskan secara singkat tentang aplikasi ini sebelum melakukan pengujian. Setelah pengisian kuesioner SUS, penguji menerima masukan dari responden untuk dipertimbangkan sebagai saran penelitian kedepannya. Terdapat 10 poin pernyataan yang diberikan kepada responden. Poin-poin tersebut antara lain :

1. Saya rasa saya akan sering menggunakan aplikasi ini.
2. Saya menemukan bahwa aplikasi ini rumit.
3. Saya merasa aplikasi ini mudah untuk digunakan.
4. Saya merasa perlu pendampingan dari orang yang paham untuk dapat menggunakan sistem ini.
5. Saya merasa berbagai fungsi dari sistem ini terintegrasi dengan baik.
6. Saya pikir sistem ini tidak memiliki konsistensi.
7. Saya membayangkan orang-orang akan sangat mudah menggunakan sistem ini.
8. Saya merasa beberapa orang akan kesusahan menggunakan aplikasi ini.
9. Saya cukup percaya diri ketika menggunakan aplikasi ini.
10. Saya perlu belajar banyak hal sebelum menggunakan aplikasi ini.

Hasil pengujian *usability* yang sudah dilakukan terdapat pada Gambar 6. Nilai yang didapatkan pada soal bernomor ganjil adalah

semakin besar semakin baik, sedangkan pada nomor soal genap semakin besar semakin buruk. Nilai SUS rata-rata dari hasil pengujian yang dilakukan adalah 55 yang berarti nilai *usability* dari sistem yang dikembangkan masih tergolong “Not Acceptable” atau masih belum layak untuk digunakan masyarakat awam (Bangor, 2009).



Gambar 6 Grafik Hasil Pengujian Usability

## 6. KESIMPULAN

Berdasarkan hasil pengujian *whitebox* dan pengujian *usability* yang dilakukan, dapat diambil beberapa kesimpulan antara lain :

1. Pembacaan *marker* dilakukan dengan menggunakan bantuan Vuforia SDK dan *game engine* Unity 3D. Sistem yang dikembangkan telah memenuhi kebutuhan sesuai dengan rancangan dengan tingkat kompleksitas yang rendah.
2. Algoritma untuk memodifikasi konten *Linked List* terdapat pada method *OnTrackableStateChanged* pada Vuforia SDK. *Linked List* pada sistem bekerja dengan baik sesuai dengan fungsinya untuk menyimpan urutan masukan operand dan operator aritmetika.
3. Berdasarkan pengujian *usability* yang dilakukan, nilai rata-rata SUS yang didapatkan adalah 55. Nilai ini berarti sistem masih tergolong belum layak digunakan kepada masyarakat awam.

## DAFTAR PUSTAKA

Bangor, A., Kortum, P. and Miller, J., 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), pp.114-123.

Petersen, N. dan Stricker, D., 2015. Cognitive augmented reality. *Computers & Graphics*, 53, pp.82-91.

Prabhu K, Gautham, 2014. *Software Metrics* [online] <<https://randcode.wordpress.com/2012/10/03/postfix-calculator-in-c/>> (Diakses 27 Desember 2017)

Raja, V. dan Calvo, P., 2017. Augmented reality: An ecological blend. *Cognitive Systems Research*, 42, pp.58-72.

Ryan, 2008. Array versus linked-list [online] <<https://stackoverflow.com/questions/166884/array-versus-linked-list>> (Diakses 10 Januari 2018)

Tapson, F., 2006. *The Oxford mathematics study dictionary*. Oxford University Press.