

Pengembangan Aplikasi Pembuat Gesture pada Samsung Gear VR untuk Game Engine Unity

Dwi Rama Malawat¹, Muhammad Aminul Akbar², Issa Arwani³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: ¹dwiramamalawat10@gmail.com, ² muhammad.aminul@ub.ac.id, ³ issa.arwani@ub.ac.id

Abstrak

Virtual Reality (VR) merupakan teknologi yang memungkinkan pengguna untuk dapat berinteraksi dengan dunia virtual yang disimulasikan komputer. Untuk memaksimalkan pengalaman dalam VR, perlu adanya penggunaan pergerakan alami sebagai input seperti gesture. Oleh karena itu, sistem pengenalan gesture penting dalam pengembangan VR. Pengembangan Mobile VR mendapatkan banyak perhatian karena meningkatnya penggunaan perangkat seluler. Salah satu perangkat mobile VR adalah Head-Mounted Display (HMD), yang merupakan perangkat dengan display monitor yang dipasang di kepala untuk menampilkan lingkungan virtual. Sistem pengenalan gesture dapat diimplementasikan pada HMD dengan perangkat haptic atau controller seperti Samsung Gear VR. Pengembangan aplikasi VR dengan kemampuan pengenalan gesture dapat memakan waktu yang lama, saat ini sudah ada beberapa library untuk membuat sistem pengenalan gesture tetapi library-library hanya mendukung perangkat VR Headset yang memiliki kemampuan 6DOF, sedangkan Samsung Gear VR hanya memiliki kemampuan 3DOF. Pada penelitian ini dikembangkan Aplikasi Pembuat Gesture pada Samsung Gear VR untuk Unity, aplikasi membantu pengembang untuk membuat dan mendefinisikan gesture baru yang kemudian dapat diimplementasikan pada aplikasi VR yang sedang dikembangkan. Dengan memberikan gesture melalui controller Gear VR, aplikasi ini akan merekam dan mendefinisikan gesture. Fungsionalitas dari aplikasi diuji dengan menggunakan metode blackbox testing dan hasilnya telah memenuhi kebutuhan sistem. Aplikasi ini menggunakan algoritme artificial neural network (ANN) untuk dapat melakukan prediksi dengan hasil akurasi 86,67%. Berdasarkan hasil pengujian usabilitas menggunakan metode System Usability Scale (SUS), aplikasi memiliki nilai usabilitas 65 yang berarti mendapatkan grade D atau Good dan termasuk dalam kategori Acceptable.

Kata kunci: *virtual reality, pengenalan gesture, samsung gear VR, artificial neural network, head-mounted display*

Abstract

Virtual Reality (VR) is a technology that provide a user to interact with virtual world simulated by computer. To maximize the user experience in VR, a VR system must be able to recognize natural movement such as gesture. That's why gesture recognition system is important. Mobile VR development gets a lot of attention because of the increase use of cellular devices. One of the mobile VR devices is Head-Mounted Display (HMD), which is a display monitor that attached to the head to display virtual environment. Gesture recognition system can be implemented on HMD with controller such as Samsung Gear VR. Developing VR applications that have gesture recognition capabilities can take a lot of time, currently there are libraries that provide the ability to create gesture recognition system but the libraries only support VR Headset devices with 6DOF, but Samsung Gear VR only have 3DOF. In this research we have develop Gesture Creator Application on Samsung Gear VR for Unity Game Engine, this tool help developer to create and define new gestures and implement it to their VR application. By giving gesture through the controller, this application can record and define the gestures. Functionality of this application has been tested with Blackbox Testing and the result are valid as expected of the system requirement. This application use artificial neural network algorithm to predict the result with 86.67% accuracy. The result of usability testing using System Usability Scale (SUS) method is 65, which means this application get grade D or Good and categorized as Acceptable.

Keywords: *virtual reality, gesture recognition, samsung gear VR, artificial neural network, head-mounted display*

1. PENDAHULUAN

Virtual Reality (VR) merupakan teknologi yang memungkinkan pengguna untuk dapat berinteraksi dengan dunia virtual yang disimulasikan komputer. Untuk memaksimalkan pengalaman dalam VR, perlu adanya penggunaan pergerakan alami sebagai *input* seperti *gesture*. Pada penelitian Xiaoming Chen menyimpulkan bahwa untuk mendapatkan pengalaman VR yang baik maka perlu adanya keterlibatan sensorik yang baik dan rasa *immersive* atau rasa realistik (Chen et al., 2019). Salah satu bentuk keterlibatan sensorik adalah *gesture*, yang dimana *gesture* tersebut dapat digunakan sebagai *input* dalam dunia *virtual*.

Untuk dapat menggunakan *gesture* sebagai *input*, sistem VR perlu memiliki sebuah pengenalan/pendeteksi *gesture*. *Gesture Recognition* merupakan teknologi inputan dinamis yang menggunakan sensor untuk mendeteksi gerakan atau pola yang dilakukan oleh pengguna. Penggunaan *gesture recognition* dalam VR sangat bermanfaat karena pengguna akan merasa lebih mudah dalam berkomunikasi dengan sistem melalui gerakan tubuh daripada menggunakan perangkat inputan konvensional seperti keyboard. Dengan memanfaatkan video kacamata sebagai umpan balik visual, earphone sebagai umpan balik pendengaran dan mekanisme getaran ponsel sebagai umpan balik haptic, sistem seluler yang luas dapat dibuat untuk menyediakan pengalaman bermain pribadi di mana-mana. *Gesture recognition* dianggap sebagai cara alami untuk berinteraksi dengan sistem seperti itu (Niezen and Hancke, 2008). Salah satu cara untuk memberikan kemampuan pendeteksi *gesture* adalah dengan menggunakan *machine learning*. Ji-Hae Kim melakukan penelitian pada tahun 2018 tentang pendeteksian *gesture* menggunakan sensor *gyroscope* dan *accelerometer* menunjukkan bahwa penggunaan *Deep Learning Neural Network* sangat cocok untuk mendeteksi *gesture*. Ji-Hae menyimpulkan bahwa penggunaan algoritme *neural network* dapat mencapai tingkat akurasi yang tinggi dalam melakukan prediksi dari beberapa *gesture* yang berbeda (Kim et al., 2018). Saat ini pendeteksi *gesture* yang ada adalah VR Infinite Gesture, 3D Motion Gesture. Namun pendeteksi *gesture* ini hanya dapat dikembangkan pada VR Headset untuk PC seperti Vive atau Oculus Rift

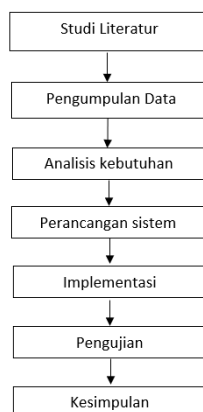
dan belum tersedia untuk perangkat mobile VR.

Mobile merupakan perangkat VR yang memanfaatkan perangkat seluler sebagai tampilan dunia *virtual*. Tujuan dari *mobile VR* adalah agar VR dapat terjangkau. Salah satu jenis perangkat *mobile VR* adalah VR *Head-Mounted Display* (HMD), yang merupakan perangkat dengan *display monitor* yang dipasang di kepala untuk menampilkan lingkungan *virtual* (Chihara dan Seo, 2018). Saat ini perangkat HMD yang ada adalah Google Cardboard, Google Daydream, Samsung Gear VR. Dari penelitian Coburn, menyatakan bahwa Samsung Gear VR merupakan perangkat yang lebih terjangkau seperti Google Cardboard tetapi memiliki kualitas seperti Oculus Rift (Coburn, Freeman and Salmon, 2017).

Perangkat VR yang dapat memberikan rasa *immersive* yang lebih adalah perangkat yang memiliki perangkat haptic atau *controller* yang berfungsi sebagai alat komunikasi antara pengguna dan sistem dengan menggunakan tombol dan juga untuk mendeteksi gerakan tangan pengguna (Chen et al., 2019). Penggunaan pendeteksi *gesture* dengan menggunakan gerakan tubuh pada VR memberikan tingkat *immersion* yang tinggi daripada penggunaan tombol dan pengguna mendapatkan pengalaman dalam dunia *virtual* yang lebih efektif (Lee, Choi dan Sohn, 2018). Google Cardboard tidak memiliki *controller* sedangkan Google Daydream dan Samsung Gear VR memiliki *controller* dimana kita dapat mengimplementasikan sistem pengenalan *gesture*.

Berdasarkan permasalahan yang telah dijabarkan, penulis melakukan penelitian berjudul pengembangan aplikasi pengenalan *gesture* untuk gear vr menggunakan artificial neural network. Penulis akan membuat aplikasi pendeteksi gerakan tangan menggunakan sensor *gyroscope* pada perangkat Samsung Gear VR yang akan direkam dan dilakukan pelatihan untuk penentuan tipe *gesture* menggunakan metode *neural network*. Aplikasi yang dibuat dapat menghasilkan pengeluaran *neural network* yang telah terlatih dan dapat digunakan pada pengembangan *game* atau simulasi VR.

2. METODOLOGI PENELITIAN



Gambar 1 Metode Pengembangan Sistem

Dalam pengembangan sebuah sistem diperlukan metode sebagai pedoman dalam merancang dan mengimplementasikannya pada sebuah sistem. Penelitian ini menggunakan metode seperti pada Gambar 1.

Pada tahap studi literature dilakukan pembelajaran mengenai literatur- literature yang mendukung pelaksanaan penelitian ini. Tahap ini bertujuan untuk memperdalam dan memahami mengenai teori maupun metode yang akan digunakan dalam memecahkan permasalahan yang ada.

Pada tahap pengumpulan data Teknik pengumpulan data kualitatif yang dilakukan pada penelitian ini adalah wawancara. Metode wawancara akan dilakukan terhadap calon pengguna aplikasi. Di dalam penelitian ini 5 narasumber yang dipilih adalah narasumber yang menjalani kegiatan atau berkaitan langsung dengan topik penelitian. Narasumber yang diwawancara adalah mahasiswa Fakultas Ilmu Komputer(FILKOM) Universitas Brawijaya(UB) Malang yang berpengalaman dalam pengembangan game.

Pada tahap analisis kebutuhan dilakukan setelah data yang diperlukan dalam penelitian didapatkan. Kemudian dari data yang didapatkan akan dibuat kebutuhan fungsional dan non-fungsional yang dibutuhkan untuk pengembangan sistem dalam penelitian ini. Kemudian dilakukan identifikasi kebutuhan dengan memberi penomoran pada tiap kebutuhan yang telah dibuat dengan *Use Case Diagram* dan *Use Case Scenario*.

Pada tahap perancangan dilakukan untuk mengubah model kebutuhan menjadi model

perancangan sistem yang berupa rancangan detail arsitektur perangkat lunak, rancangan struktur data, dan rancangan komponen yang diperlukan untuk mengembangkan sistem. Perancangan digambarkan dengan perancangan arsitektur sistem dan perancangan *UML* yang terdiri dari perancangan *class diagram* dan perancangan *sequence diagram*, serta perancangan aplikasi yang terdiri dari *screenflow*, *wireframe* dan *mockup UI*.

Pada tahap implementasi dilakukan realisasi dari rancangan perangkat lunak menjadi sistem yang dapat dioperasikan. Tahap ini akan mewujudkan seluruh model yang telah dihasilkan pada proses perancangan menjadi sebuah kode program. Pada akhir tahap ini akan didapatkan hasil implementasi kode program dan implementasi antarmuka yang dapat dioperasikan oleh pengguna.

Setelah tahap implementasi akan dilakukan tahap pengujian yang bertujuan untuk mengetahui apakah perangkat lunak sesuai dengan kebutuhan. Pada tahap ini akan dilakukan pengujian fungsionalitas dengan menggunakan metode *blackbox*, pengujian akurasi, dan pengujian usabilitas menggunakan *System Usability Scale (SUS)*.

3. ANALISIS KEBUTUHAN

3.1. Gambaran Umum Aplikasi

Dalam penelitian ini dibangun sistem pembuat pengenalan gesture untuk Samsung Gear VR dalam bentuk mobile Android bernama “Aplikasi Pembuat *Gesture* pada Samsung Gear VR untuk *Game Engine Unity*”. Tujuan dari dikembangkan sistem ini adalah untuk dapat mempermudah pengimplementasian sistem pengenalan gesture dalam *Virtual Reality* yang menggunakan Samsung Gear VR. Pengguna akan diberikan pilihan untuk merekam *gesture*, melatih *gesture*, menguji *gesture*, dan mengeksport *gesture*. Untuk membuat *gesture* baru maka pengguna akan diminta untuk merekam *gesture*, *gesture* dari pergerakan controller Samsung Gear VR yang akan direkam, jika telah terekam sebuah *gesture* pengguna dapat melakukan pelatihan *gesture* atau menguji *gesture*. Setelah dilatih dan diuji maka pengguna dapat mengeksport pengenalan *gesture* tersebut dan akan diimplementasikan melalui *library* pada Unity.

3.2. Identifikasi Aktor

Aktor adalah seseorang yang berinteraksi langsung dengan sistem. Adapun actor yang berhubungan dengan sistem pengenalan gesture dapat dilihat pada Table 1 dibawah ini

Tabel 1. Deskripsi Aktor Sistem

Aktor	Deskripsi
Pengguna	Merupakan orang yang langsung berinteraksi dengan aplikasi dan ingin mengimplementasikan pengenalan gesture pada pengembangan Virtual Reality.

3.3. Kebutuhan Fungsional

Kebutuhan fungsional sistem merupakan fungsi di dalam sistem atau komponen yang harus ada. Di dalam penulisannya harus diberi suatu identitas untuk mempermudah proses identifikasi kebutuhan juga menjaga konsistensi terkait dengan kebutuhan suatu sistem, mulai dari proses perancangan hingga pengujian selesai dilakukan. Kebutuhan fungsionalitas sistem dapat dilihat pada Tabel 2 dibawah ini

Tabel 2. Kebutuhan Fungsionalitas Sistem

No.	Kode Kebutuhan Sistem	Deskripsi Kebutuhan
1	GRSGVR-001	Sistem harus dapat menyediakan mekanisme untuk membuat proyek pengenalan gesture
2	GRSGVR-002	Sistem harus dapat menambahkan gesture kepada proyek
3	GRSGVR-003	Sistem harus dapat menghapus gesture dari proyek
4	GRSGVR-004	Sistem harus dapat menyediakan mekanisme untuk membuat gesture
5	GRSGVR-005	Sistem harus dapat merekam data percobaan untuk gesture
6	GRSGVR-006	Sistem harus dapat menyediakan mekanisme untuk melatih proyek pengenalan gesture
7	GRSGVR-007	Sistem harus dapat menyediakan mekanisme untuk menguji proyek pengenalan gesture

8	GRSGVR-008	Sistem harus dapat menyediakan mekanisme untuk mengeksport proyek pengenalan gesture
---	------------	--

4. PERANCANGAN

4.1. Class Diagram

Kelas yang dibuat terdiri dari kelas *View*, *Model*, *Controller*. Diantaranya:

- *View*
ProjectListPanel, *ProjectPanel*, *GestureListPanel*, *GesturePanel*,
- *Controller*
ProjectController, *Projects*, *GestureGallery*, *GestureController*
- *Model*
Project, *Gesture*, *Sample*

4.2. Perancangan Algoritme

Perancangan algoritme merupakan perancangan lebih detail komponen dari dekomposisi sub sistem yang dimodelkan, salah satu perancangan algoritme dibawah ini adalah algoritme melatih *neural network* proyek yang dituliskan dalam Tabel 3 dibawah ini

Tabel 3. Algoritme Melatih *Neural Network* Proyek

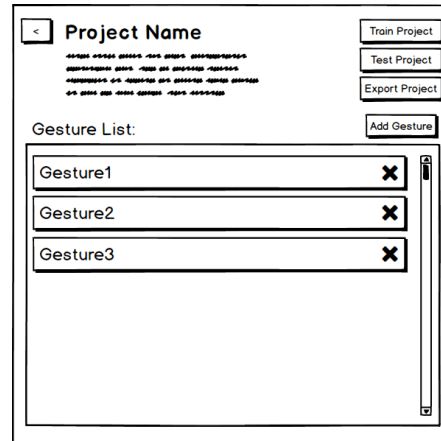
Melatih <i>neural network</i> proyek FUNCTION Melatih <i>neural network</i> proyek Pass In: tidak ada Membuat output yang dibutuhkan sesuai dengan gesture dalam proyek FOR count to 5000 FOR setiap gesture dalam daftar gesture terpilih proyek Membuat input yang dibutuhkan dengan menggunakan daftar delta quaternion dari percobaan FOR setiap percobaan dalam koleksi percobaan gesture Set count to 0 Memanggil fungsi feed forward neural network dan diberikan nilai input Memanggil fungsi backpropagation neural network dan diberikan nilai output yang sesuai dengan input ENDFOR ENDFOR ENDFOR

Pass Out: tidak ada

4.3. Perancangan Antarmuka

Perancangan antarmuka dibuat untuk mempermudah proses implementasi saat membuat tampilan untuk user.

Berdasarkan perancangan antarmuka juga inilah kita dapat melihat bagaimana alur perpindahan dari 1 halaman menuju halaman lain. Contoh pada Gambar 2 dibawah ini adalah antarmuka projek



Gambar 2 Perancangan Antarmuka Projek

5. IMPLEMENTASI

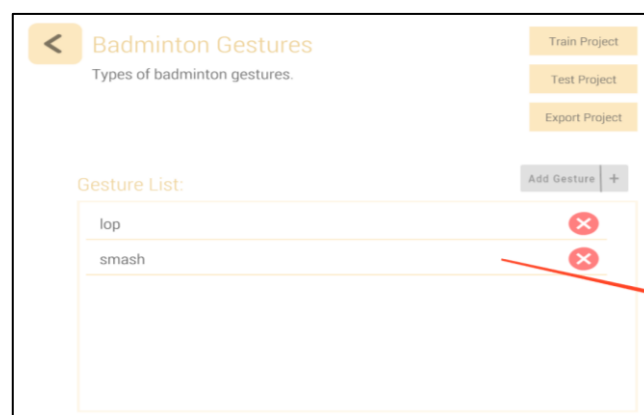
5.1. Implementasi Kode

Implementasi kode program terkait kebutuhan fungsional sistem dibagi menjadi 3 komponen besar, mengatur projek, mengatur gesture pada projek, dan mengatur sample pada gesture. Implementasi dilakukan dengan menggunakan *game engine* Unity sehingga akan menghasilkan kode dalam bahasa

pemrograman C#. Pengimplementasian untuk dapat digunakan pada Samsung Gear VR menggunakan *library Oculus Integretion*.

5.1. Implementasi Antarmuka

Pengimplementasian antarmuka mengacu kepada perancangan antarmuka yang sudah dibuat sebelumnya. Berikut adalah antarmuka projek yang ditunjukkan pada gambar 3



Gambar 3 Implementasi Antarmuka Projek

6. PENGUJIAN

6.1. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan menggunakan metode *blackbox testing*.

Pengujian dilakukan dengan menentukan kasus uji dan hasil yang diharapkan, kemudian aplikasi diuji sesuai dengan kasus yang ditentukan dapat menghasilkan hasil yang diharapkan. Hasil dari pengujian aplikasi dari 8 kasus uji bernilai valid.

6.2. Pengujian Akurasi

Pengujian akurasi dilakukan melatih neural network aplikasi menggunakan metode *k-fold cross validation* dengan membagi 60 data menjadi 5 kelompok atau *fold* untuk menentukan *learning rate* yang optimum, Nilai akurasi akan dihitung dengan menggunakan persamaan berikut:

$$\%AP = \frac{T_{\text{benar}}}{T_{\text{data}}} \times 100\%$$

Keterangan :

$\%AP$ = Akurasi prediksi pukulan tenis

T_{benar} = Jumlah prediksi yang benar pada pengujian

T_{data} = Jumlah data yang diujikan pada pengujian

Hasil pengujian akurasi bernilai sebagai berikut:

Tabel 4. Hasil Akurasi

Nilai <i>Learning Rate</i>	Total Akurasi	Rata-rata Akurasi
0.02	425	85
0.04	433,34	86,67
0.08	433,34	86,67
0.16	433,34	86,67

Berdasarkan hasil pengujian akurasi pada Tabel 4, dapat disimpulkan bahwa akurasi perangkat lunak dapat mencapai optimum dengan menggunakan *learning rate* diantara 0.04 dan 0.16.

6.3. Pengujian Usabilitas

Pengujian usabilitas yang dilakukan adalah dengan menggunakan metode *System Usability Scale* (SUS). Responden yang akan memberikan penilaian adalah pengembang Unity. Kuisisioner SUS diambil dari penelitian berjudul “*An Indonesian Adaptation of the System Usability Scale (SUS)*” penulis bernama Zahra Sharfina dan Harry Budi Santoso. Kuisisioner SUS terdiri dari 10 pertanyaan dan responden diminta untuk menilai menggunakan skala *Likert*. Hasil dan skor kuisisioner SUS kemudian diolah sesuai dengan kaidah-kaidah mengolah SUS, setiap pertanyaan bernomor ganjil akan dihitung dengan cara mengurangi 1 dari skor yang didapatkan. Setiap pertanyaan bernomor genap akan dihitung dengan cara nilai 5 dikurangi dengan skor yang didapat. Kemudian hasil skor dijumlahkan lalu dikalikan dengan nilai 2.5 sehingga akan didapatkan nilai *usability* dari

responden. Hasil dari pengujian usabilitas dengan metode SUS dapat dilihat pada tabel 4 dibawah:

Tabel 5. Hasil pengujian usabilitas

Jumlah Total Nilai <i>Usability</i>	390
Rata-Rata Nilai <i>Usability</i>	65

7. KESIMPULAN

Berdasarkan hasil penelitian dari tahap analisis kebutuhan, perancangan, implementasi dan pengujian, maka dapat diambil kesimpulan seperti dibawah ini:

1. Analisis kebutuhan aplikasi pembuat *gesture* pada Samsung Gear VR untuk *game engine* Unity diawali dengan pengumpulan data dengan cara wawancara terhadap responden. Pertanyaan yang diberikan difokuskan kepada pentingnya pengenalan *gesture* dan masalah yang didapatkan oleh responden dalam pengembangan sebuah aplikasi *virtual reality*. saat Kemudian akan menghasilkan deskripsi umum dari sistem dan kebutuhan sistem. Hasil dari tahap ini didapatkan 8 kebutuhan fungsional dan 2 kebutuhan non-fungsional.
2. Perancangan aplikasi dilakukan dengan membuat pembuatan *class diagram* berdasarkan kebutuhan yang telah ditemukan, dari hasil tersebut dapat dilakukan perancangan algoritme terhadap operasi yang berhubungan dengan kebutuhan fungsional. Dalam tahap perancangan juga dibuat rancangan antarmuka aplikasi.
3. Pada tahap implementasi, hasil perancangan dari kebutuhan yang telah dibuat diimplementasikan menjadi sebuah aplikasi. Implementasi aplikasi menggunakan *game engine* Unity 3D dengan bahasa pemrograman C# (C Sharp).
4. Dalam tahap pengujian dilakukan 3 macam pengujian, pengujian fungsional, pengujian usabilitas, dan pengujian akurasi. Pengujian fungsional dilakukan dengan menggunakan metode pengujian blackbox dimana dibuat uji kasus dan hasil yang diharapkan dari aplikasi apakah sudah sesuai dengan kebutuhan fungsional yang telah dibuat, hasil pengujian ini membuktikan aplikasi ini memiliki fungsionalitas yang sesuai dengan kebutuhan. Pengujian usabilitas dilakukan dengan menggunakan metode *System Usability Scale*

(SUS) dimana responden diberikan kuisioner dengan 10 pertanyaan dan harus menjawab dengan menggunakan skala likert dan hasilnya akan dihitung untuk menentukan usability dari sistem, pada penelitian ini sistem memiliki nilai usability 65 dan termasuk dalam kategori Acceptable. Pengujian akurasi dilakukan dengan menggunakan *k-fold cross validation*, 60 data dibagi menjadi 5 kelompok atau fold, hasil dari pengujian ini artificial neural network aplikasi dapat mencapai akurasi 86,67% dengan menggunakan learning rate diantara 0.04 dan 0.16.

8. DAFTAR PUSTAKA

- Chen, X. et al. (2019) 'ImmerTai: Immersive Motion Learning in VR Environments', *Journal of Visual Communication and Image Representation*, 58, pp. 416–427. doi: 10.1016/j.jvcir.2018.11.039.
- Chihara, T. and Seo, A. (2018) 'Evaluation of physical workload affected by mass and center of mass of head-mounted display', *Applied Ergonomics*. Elsevier, 68(December 2017), pp. 204–212. doi: 10.1016/j.apergo.2017.11.016.
- Kim, J. H. et al. (2018) 'deepGesture: Deep learning-based gesture recognition scheme using motion sensors', *Displays*. Elsevier, 55(December 2017), pp. 38–45. doi: 10.1016/j.displa.2018.08.001.
- Lee, Y., Choi, W. and Sohn, B. S. (2018) 'Immersive gesture interfaces for 3D map navigation in HMD-based virtual environments', *International Conference on Information Networking*, 2018-Janua, pp. 963–965. doi: 10.1109/ICOIN.2018.8343267.
- Niezen, G. and Hancke, G. P. (2008) 'Gesture recognition as ubiquitous input for mobile phones', *DAP (Devices that Alter Perception) Workshop at the 10th International Conference on Ubiquitous Computing (UBICOMP '08)*, (Dap).
- Sharfina, Zahra & Santoso, Harry. (2016). An Indonesian Adaptation of the System Usability Scale (SUS). 10.1109/ICACIS.2016.7872776.
- Gareth, J., Daniela, W., Trevor, H. & Robert, T., 2013. *An Introduction to Statistical Learning with Applications in R*. 1st ed. New York: Springer-Verlag New York.