

# Rendering Karakter 3D Virtual secara Real-Time menggunakan Metode Light Estimation pada Augmented Reality Berbasis Lokasi

Kevin, Liliana, Kartika Gunadi.

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-mail: C14170003@john.petra.ac.id, lilian@petra.ac.id, kgunadi@petra.ac.id

## ABSTRAK

Aplikasi *augmented reality* sudah banyak terdapat pada perangkat *mobile*, tetapi kebanyakan aplikasi *augmented reality* masih mengasumsikan bahwa sumber cahaya yang berada pada dunia virtual selalu berasal dari atas objek dan memiliki arah yang selalu ke bawah sehingga bayangan yang dihasilkan selalu tepat di bawah objek, oleh karena itu dibutuhkan metode untuk mengestimasi cahaya agar arah bayangan yang dihasilkan lebih realistis, tetapi tetap dapat dijalankan pada perangkat *mobile*.

Untuk menjawab masalah di atas, digunakan metode *light estimation* pada *real-time rendering* aplikasi AR di perangkat *mobile* agar arah bayangan dari hasil *rendering* objek virtual sejajar dan searah dengan arah bayangan dari objek nyata pada lingkungannya, tetapi tetap menggunakan *resource* yang masih dapat digunakan pada perangkat *mobile*.

Hasil pengujian pada penelitian ini menunjukkan bahwa arah bayangan yang dihasilkan metode estimasi cahaya pada lingkungan *indoor* cukup akurat (sekitar 33°) dan cukup ringan untuk digunakan pada perangkat *mobile*, karena perbedaan FPS dan penggunaan RAM hampir sama dengan aplikasi tanpa penggunaan metode estimasi cahaya, walaupun ada peningkatan penggunaan CPU dan baterai, tetapi cukup kecil untuk tetap dapat digunakan pada perangkat *mobile*.

**Kata Kunci:** *real-time rendering*, estimasi cahaya, *augmented reality*, *markerless augmented reality*, aplikasi berbasis lokasi.

## ABSTRACT

*Augmented reality applications are already widely available on mobile devices, but most augmented reality applications assume that light source always comes from above the object and its direction is always downwards so that the shadow is always right under the object, therefore a method is needed to estimate light so that the direction of shadow produced is more realistic, but can still be run on mobile devices.*

*To answer the problem, light estimation method is used in real-time rendering of AR applications on mobile devices so that the shadow direction from virtual objects rendering is parallel and in the same direction as the shadow direction of real objects in their environment, but still uses resources that can be used on mobile devices.*

*Results in this study indicate that the direction of shadow produced by light estimation method in indoor environment is quite accurate (about 33°) and light enough to be used on mobile devices, because the difference in FPS and RAM usage is almost the same as the usage of application without the use of light estimation method,*

*although there is an increase in CPU and battery usage, it's small enough to still work on a mobile device.*

**Keywords:** *real-time rendering, light estimation, augmented reality, markerless augmented reality, location-based application.*

## 1. PENDAHULUAN

*Augmented Reality* (AR) adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Penggunaan AR untuk mempresentasikan sesuatu pada perangkat *mobile* lebih menarik daripada menggunakan halaman web karena penggunaan AR lebih imersif dan lebih dinikmati oleh pengguna [8]. Penggunaan aplikasi AR juga dapat mengurangi beban kognitif, menghasilkan efek yang positif terhadap kesadaran konteks, keamanan [2], persepsi, hubungan, dan perilaku pengguna [14]. Selain itu, pemahaman pengguna juga meningkat terhadap informasi yang dipresentasikan dan aplikasi menjadi lebih menarik [9], karena melihat objek yang dipresentasikan ditampilkan secara langsung kepada pengguna dapat meningkatkan motivasi pengguna [18].

Ada masalah yang terdapat pada *Rendering* aplikasi AR di perangkat *mobile*, yaitu metode *rendering* yang digunakan pada PC tidak dapat digunakan karena membutuhkan *resource* yang sangat besar, selain itu, karena harga perangkat *mobile* semakin terjangkau dan terdapat peningkatan performa dari CPU dan GPU untuk perangkat *mobile*, maka semakin banyak proses dan algoritma kompleks yang sebelumnya tidak dapat dijalankan pada perangkat *mobile* menjadi dapat dilakukan [3][17]. Oleh karena itu diperlukan metode *rendering* khusus pada perangkat *mobile* yang lebih ringan agar dapat dijalankan secara *real-time*, namun hasilnya tidak berbeda jauh dari *rendering* pada PC.

Penelitian ini akan menggunakan metode *light estimation* untuk *real-time rendering* pada aplikasi AR di perangkat *mobile* agar memiliki hasil yang serupa dengan *rendering* pada PC, tetapi tidak menggunakan *resource* yang terlalu besar.

Ada beberapa penelitian sebelum ini yang serupa, salah satunya adalah “sistem *markerless AR* dengan estimasi sumber cahaya untuk pencahayaan langsung” [4]. Penelitian ini sudah dapat menghasilkan bayangan yang realistis pada *markerless AR*, tetapi metode ini sangat tidak sesuai untuk digunakan pada perangkat *mobile*, karena membutuhkan lebih dari satu kamera, membutuhkan kamera *fish-eye*, dan membutuhkan waktu setelah kamera dipindahkan. Penelitian lainnya ada yang menggunakan *Deep Learning* untuk mengestimasi cahaya [10][7] Pada penelitian ini, metode yang digunakan tidak memerlukan peralatan khusus atau pengetahuan sebelumnya tentang lingkungannya, dan dapat

digunakan pada lingkungan *indoor* maupun *outdoor*, tetapi metode ini memerlukan data set dan pelatihan terlebih dahulu sebelum dapat digunakan.

Penelitian serupa pada topik *rendering* karakter 3D virtual adalah “pembuatan interaksi animasi karakter 3D untuk AR” [15]. Penelitian ini menghasilkan *rendering* pada perangkat dan menggunakan penanda pada AR untuk berinteraksi dengan animasi karakter 3D, tetapi animasi yang dapat digunakan terbatas pada penanda, tidak ada parameter yang dapat diatur pada animasi, harus menggunakan penanda khusus, dan tidak ada metode untuk menyamakan pencahayaan dengan dunia nyata. Metode *rendering* pada penelitian lainnya adalah metode *cloud-to-end* [20]. Metode ini dapat menjalankan *rendering* pada server di *cloud*, sehingga hasil *rendering* yang didapatkan oleh pengguna berkualitas lebih tinggi daripada hasil *rendering* yang dapat diproses oleh perangkat pengguna. Hasil *rendering* juga dapat dibagikan ke berbagai pengguna, sehingga berguna untuk digunakan pada aplikasi yang akan digunakan oleh banyak pengguna. Tetapi metode ini membutuhkan koneksi internet yang cukup cepat dan stabil agar dapat dijalankan secara *real-time*. Kelebihan dari metode *cloud-to-end* juga tidak terlalu dibutuhkan dalam penelitian ini.

Dari segi AR berbasis lokasi, penelitian yang serupa adalah “penggunaan *cloud* pada AR untuk pencarian lokasi” [11]. Penelitian ini menggabungkan teknologi *cloud* pada AR untuk pembuatan aplikasi pencarian lokasi berbasis web pada perangkat *mobile*, tetapi aplikasi tidak dapat menambah lokasi objek yang ditampilkan, dan dibutuhkan server yang selalu menyala untuk menyediakan informasi kepada aplikasi.

Oleh karena itu pada penelitian ini akan dilakukan penelitian untuk memodifikasi *shader* dan menggunakan *library* ARCore untuk menyesuaikan pencahayaan *rendering* karakter 3D virtual yang memiliki animasi pada perangkat *mobile* android agar arah bayangan yang dihasilkan oleh *rendering* karakter 3D virtual sejajar dan searah dengan arah bayangan dari objek nyata pada lingkungannya dengan menggunakan metode yang cukup ringan dan dapat digunakan oleh kebanyakan smartphone, yaitu dengan metode *light estimation*. Setelah itu menggabungkannya dengan aplikasi AR berbasis lokasi, sehingga menghilangkan kebutuhan untuk menggunakan penanda khusus sekaligus menambahkan kemampuan untuk menambah animasi yang dapat digunakan. Objek dan lokasi yang digunakan untuk *rendering* juga dapat ditambahkan langsung dan disimpan pada perangkat, sehingga tidak membutuhkan server khusus untuk menyimpan data objek dan lokasi.

## 2. LANDASAN TEORI

### 2.1 Markerless Augmented Reality

*Markerless Augmented Reality* adalah jenis *Augmented Reality* yang mengintegrasikan objek virtual 3D dengan objek 3D nyata pada suatu lingkungan secara *real-time*. Perbedaan mendasar dari AR berbasis penanda adalah metode yang digunakan untuk menempatkan objek virtual di dunia nyata. Pendekatan ini tidak didasarkan pada penggunaan penanda tradisional, yang perlu ditempatkan di dunia nyata untuk dilacak oleh sistem dengan tujuan mengetahui posisi dan orientasi [19].

*Augmented Reality* (AR) adalah salah satu teknologi *Extended Reality* yang dapat menampilkan pandangan *real-time* secara langsung atau tidak langsung dari sebuah lingkungan fisik dunia nyata yang secara mulus telah digabungkan dengan menambah objek atau informasi digital (suara, video, grafik, teks, atau lokasi

geografis) yang bersifat virtual dan mempresentasikan hasil penggabungannya kepada pengguna [5][6].

### 2.2 Rendering

*Rendering* adalah proses dari membangun gambar dari sebuah model (2D atau 3D), ataupun dari sebuah berkas adegan, melalui program komputer. Sebuah berkas adegan pada *rendering* terdiri dari objek-objek dalam sebuah struktur data, yang dapat berupa objek geometri, sudut pandang, tekstur, pencahayaan, dan informasi bayangan sebagai sebuah deskripsi dari adegan virtual. Data yang terisi dalam berkas adegan kemudian melewati program *rendering* untuk diproses dan menjadi hasil keluaran untuk sebuah gambar digital [1].

### 2.3 Light Estimation

Aplikasi AR merupakan adegan yang dinamis, sehingga membutuhkan metode *real-time rendering*, tetapi karena metode ini dijalankan dalam waktu nyata, maka adegan yang dihasilkan biasanya kurang realistis, sedangkan di dalam aplikasi AR biasanya diharapkan / diinginkan untuk menghasilkan *rendering* yang serealistis mungkin, semakin realistis hasil *rendering* yang didapatkan maka semakin baik. Oleh karena itu ada beberapa cara yang dapat dilakukan dengan tujuan untuk membuat hasil dari *real-time rendering* menjadi lebih realistis, salah satu caranya adalah dengan mengestimasi sumber cahaya pada suatu adegan.

*Light Estimation* adalah metode untuk mengestimasi sumber cahaya pada suatu adegan yang dapat mewakili berbagai sumber cahaya, dengan intensitas dan arah yang berbeda [10]. Salah satu cara untuk menggunakan metode *light estimation* adalah dengan menggunakan *lighting estimation* API pada dalam *library* ARCore. API (*Application Programming Interface*) adalah jenis antarmuka perangkat lunak, yang menawarkan layanan ke perangkat lunak lain dan memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi [16].

### 2.4 Metode Pelacakan

Pelacakan lokasi yang digunakan untuk aplikasi AR berbasis lokasi adalah Geo AR atau *Gravimetric AR*, yaitu metode yang menggunakan posisi perangkat untuk melacak lokasinya. Teknik ini tidak menggunakan *image processing*, melainkan menggunakan GPS, Wifi, Bluetooth, atau berbagai macam sensor lokasi lainnya untuk menentukan posisi perangkat [12].

Pelacakan permukaan yang digunakan pada aplikasi ini adalah dengan menggunakan metode SLAM, yaitu proses yang secara *real-time* dapat membangun peta dari sebuah lingkungan yang tidak diketahui dan pada saat yang bersamaan menggunakan peta tersebut untuk menentukan lokasinya [13].

## 3. ANALISA DAN DESAIN SISTEM

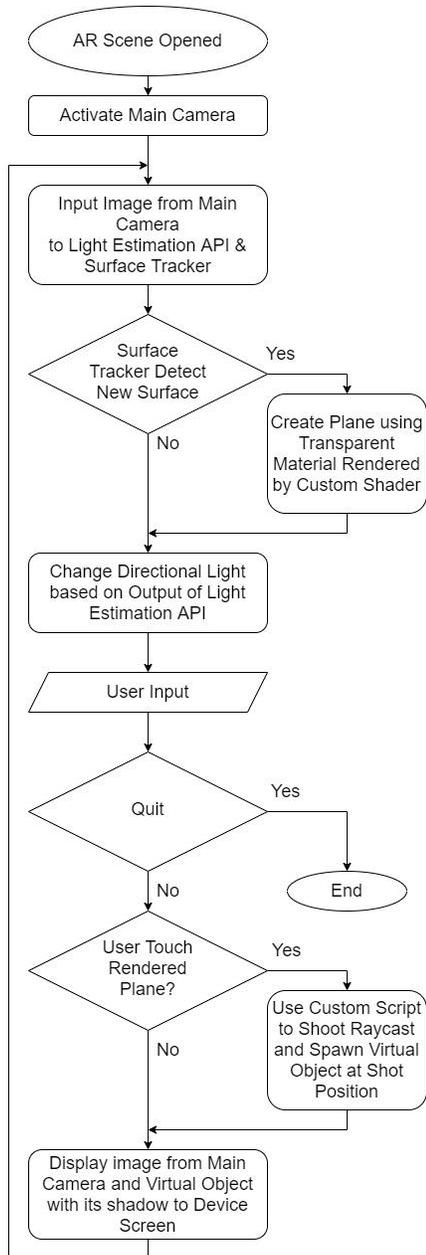
### 3.1 Analisis Permasalahan

Dapat dilihat bahwa masalah-masalah yang masih ada pada aplikasi penelitian yang serupa adalah membutuhkan perangkat tambahan untuk menjalankan aplikasinya, seperti beberapa kamera khusus, penanda untuk mengeluarkan objek virtual, dan server untuk menyediakan data. Oleh karena itu, fitur-fitur yang dibutuhkan untuk menyelesaikan masalah pada aplikasi penelitian ini adalah fitur untuk mengganti animasi, transformasi, dan model tanpa menggunakan penanda, lalu fitur untuk menyimpan dan mengganti-ganti data pada internet langsung dari aplikasi tanpa

perlu menyalakan server sendiri, dan fitur untuk mengestimasi cahaya pada perangkat *mobile* umum.

Fitur untuk mengganti animasi, transformasi, dan model dapat dilakukan dengan cara menampilkan tombol untuk mengganti animasi pada *user interface* aplikasi, lalu fitur untuk data pada internet dapat dilakukan dengan menggunakan penyedia layanan *database cloud*.

*Light estimation* seperti yang sudah dijelaskan pada subbab 2.3 merupakan metode yang digunakan untuk mengestimasi pencahayaan pada dunia nyata agar informasinya dapat digunakan dalam pencahayaan dunia virtual. Pada penelitian ini, metode *light estimation* digunakan sebagai solusi untuk menyelesaikan masalah pencahayaan AR di perangkat *mobile*. Berikut ini adalah penggunaan metode *light estimation* pada aplikasi penelitian ini.

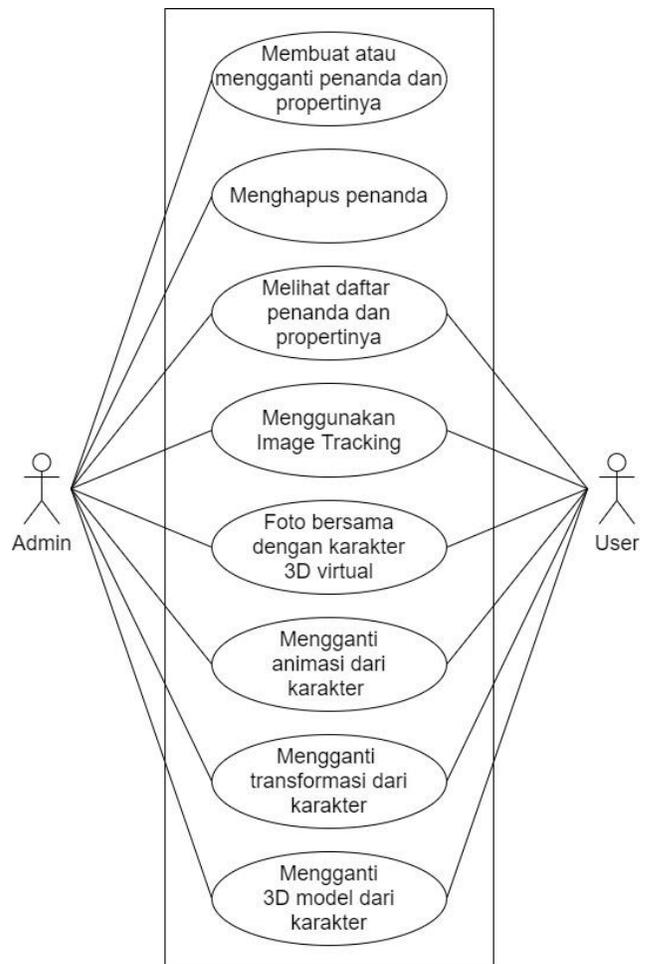


**Gambar 1.** Solusi masalah pencahayaan AR di perangkat *mobile*

Gambar 1 merupakan solusi yang menunjukkan bahwa pada aplikasi penelitian ini, metode *light estimation* digunakan pada mode AR. Ketika adegan AR dibuka, kamera utama perangkat diaktifkan untuk membuat gambar yang digunakan sebagai *input* metode *light estimation* API dan *surface tracker* pada *library ARCore*, dan sebagai dasar *output* yang ditampilkan pada layar perangkat. *Surface tracker* kemudian membuat bidang permukaan pada dunia virtual sesuai dengan permukaan pada dunia nyata, yang menggunakan material transparan yang dirender oleh *shader* khusus agar dapat menerima bayangan. Objek 3D virtual kemudian dapat dimunculkan dengan cara menggunakan *script* untuk menembak *raycast* pada bidang permukaan yang dihasilkan dan mengeluarkannya pada posisi hasil tembakan. Hasil dari *light estimation* API berupa informasi cahaya digunakan untuk mengatur pencahayaan pada dunia virtual agar bayangan yang dihasilkan oleh objek 3D virtual sesuai dengan bayangan yang ada pada dunia nyata. Objek dan bidang transparan dengan bayangan yang dihasilkan ini kemudian ditambahkan pada gambar yang dihasilkan kamera utama untuk ditampilkan sebagai *output* pada layar perangkat.

### 3.2 Desain Sistem

Dalam pembuatan aplikasi penelitian ini, desain sistem dapat dijelaskan dan dibuat dalam bentuk diagram berdasarkan analisis permasalahan yang dibahas pada subbab sebelum ini.



**Gambar 2.** Use Case Diagram

Aplikasi dibagi menjadi dua versi, yaitu untuk admin, dan untuk pengguna biasa. Gambar 2 menunjukkan bahwa aplikasi dimulai dengan memuat adegan awal. Pada adegan ini pengguna dapat memilih untuk keluar dari aplikasi atau melanjutkan ke adegan peta. Setelah pengguna memilih untuk melanjutkan dan adegan peta selesai dimuat, pengguna dapat memilih untuk memilih untuk membuka daftar penanda, memencet atau berjalan ke penanda, menuju adegan *image tracking*, atau kembali ke adegan awal. Jika pengguna adalah admin, pengguna juga dapat menambah penanda, mengedit penanda, dan menghapus penanda.

Pada adegan AR, pengguna dapat memilih untuk memencet permukaan, mengganti animasi, mengganti transformasi, mengganti model, memencet tombol ambil foto, melakukan reset sesi AR, membuka panel *settings*, dan kembali ke adegan peta.

#### 4. PENGUJIAN SISTEM

Pengujian sistem dilakukan pada dua perangkat dengan spesifikasi yang berbeda. Perangkat pertama memiliki spesifikasi :

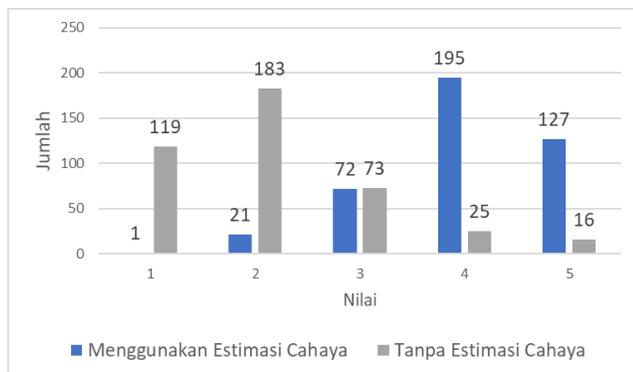
- Nama : Samsung Galaxy A8 (2018)
- Waktu rilis : 2018, Januari
- Sistem operasi : Android 9 Pie
- Prosesor : Exynos 7885 Octa-core (2x2.2 GHz & 6x1.6 GHz)
- GPU : Mali-G71
- RAM : 4GB

Perangkat kedua memiliki spesifikasi :

- Nama : Xiaomi Redmi Note 8 Pro
- Waktu rilis : 2019, 24 September
- Sistem operasi : Android 10
- Prosesor : Mediatek Helio G90T Octa-core (2x2.05 GHz & 6x2.0 GHz)
- GPU : Mali-G76 MC4
- RAM : 6GB

#### 4.1 Pengujian Perbandingan Arah Bayangan

Untuk mengetahui dengan lebih jelas apakah arah bayangan yang dihasilkan oleh karakter 3D virtual realistis atau tidak menurut pendapat berbagai orang, maka akan digunakan survei untuk pengukuran realistiknya intensitas cahaya dan hasil bayangan dari satu (sangat tidak realistis) sampai lima (sangat realistis).



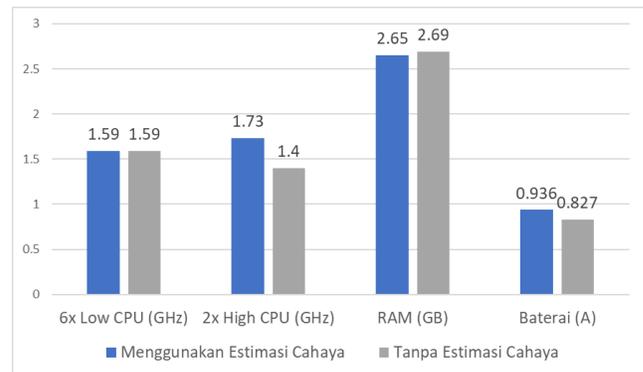
Gambar 3. Hasil survei

Gambar 3 merupakan hasil survei yang menunjukkan bahwa menurut 52 responden yang sudah mengikuti survei, foto yang

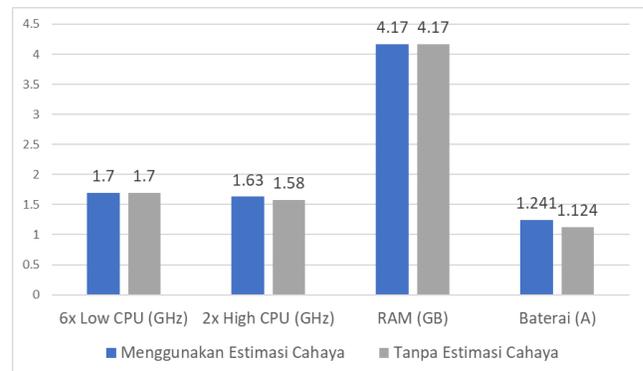
menggunakan estimasi cahaya memiliki nilai realistis 4.024 dari 5 poin dan foto yang tidak menggunakan estimasi cahaya memiliki nilai realistis 2.125 dari 5 poin. Oleh karena itu, dapat disimpulkan bahwa foto yang menggunakan estimasi cahaya lebih realistis jika dibandingkan dengan foto yang tidak menggunakan estimasi cahaya.

#### 4.2 Pengujian Penggunaan Resource

Pengujian perbedaan penggunaan *resource* dilakukan dengan tujuan untuk melihat penggunaan *resource* pada *rendering* objek 3D, yang dilakukan dengan cara membandingkan penggunaan CPU, penggunaan RAM, dan penggunaan baterai selama 30 detik, dan rata-rata waktu untuk merender (*Average Frames per Second*) yang stabil, di antara *rendering* yang menggunakan dan tidak menggunakan *light estimation*.



Gambar 4. Perbandingan nilai rata-rata resource dan FPS pada perangkat pertama

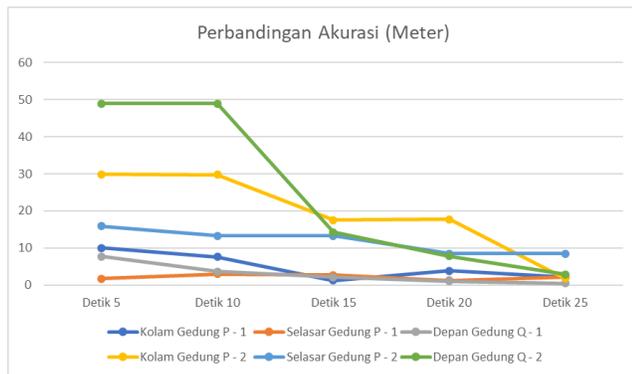


Gambar 5. Perbandingan nilai rata-rata resource dan FPS pada perangkat kedua

Gambar 4 merupakan perbandingan nilai rata-rata *resource* dan FPS pada perangkat pertama. Gambar 5 merupakan perbandingan nilai rata-rata *resource* dan FPS pada perangkat kedua. Dari kedua hasil pengujian penggunaan *resource* yang terdapat pada kedua gambar tersebut, dapat dilihat bahwa penggunaan RAM dan FPS kurang lebih sama, dan perbedaan penggunaan *resource* lainnya antara metode estimasi cahaya dengan penggunaan *resource* tanpa metode estimasi cahaya cukup kecil untuk tetap dapat digunakan pada perangkat *mobile*. Tetapi durasi waktu penggunaan yang berkepanjangan dalam suatu sesi akan meningkatkan temperatur baterai dan membuat sistem operasi untuk mempengaruhi penggunaan *resource* dalam pencegahan *Overheating*.

### 4.3 Pengujian Akurasi Pelacakan Lokasi

Gambar 6 merupakan pengujian yang dilakukan pada 3 lokasi yang berbeda, dan jarak durasi di antara masing-masing pelacakan sekitar 5 detik. Dapat dilihat dari kedua pengujian akurasi perangkat bahwa rata-rata akurasi pada detik ke 25 cukup kecil untuk perangkat pertama (1.59 Meter) dan perangkat kedua (4.4 Meter), tetapi perangkat kedua membutuhkan waktu lebih lama untuk mendapatkan hasil yang lebih akurat. Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa penggunaan GPS sebagai sumber informasi lokasi pada aplikasi berbasis lokasi bergantung pada perangkat yang digunakan, oleh karena itu, digunakan pelacakan *image tracking* sebagai cadangan pelacakan untuk menentukan lokasi pengguna. Selain itu, dari pengujian ini dapat dilihat bahwa semakin lama suatu perangkat melacak lokasi pada suatu tempat tertentu, maka rata-rata pelacakan lokasinya menjadi semakin akurat.



Gambar 6. Perbandingan akurasi pada perangkat pertama dan kedua

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

- Penggunaan fitur pada aplikasi sudah berjalan dengan baik karena semua fitur sudah dapat dijalankan secara *real-time* dari awal sampai pengambilan foto.
- Kondisi lingkungan lokasi tempat foto yang disarankan agar aplikasi dapat digunakan dengan baik adalah lingkungan yang menghasilkan satu bayangan *hard shadow* (lingkungan dengan satu sumber cahaya yang cukup terang), karena aplikasi kurang cocok digunakan pada lingkungan *indoor* yang menghasilkan lebih dari satu bayangan seperti pada siang hari.
- Arah bayangan yang dihasilkan pada lingkungan *indoor* cukup akurat (sekitar 33°) sehingga lebih baik daripada tanpa penggunaan metode estimasi cahaya.
- Metode estimasi cahaya memiliki kelemahan bayangan yang dihasilkan tidak dapat dihasilkan pada bidang vertikal, dan akurasi intensitas cahaya tergantung cahaya rata-rata dari gambar *input*.
- Penggunaan metode estimasi cahaya cukup ringan untuk digunakan pada perangkat *mobile* karena FPS yang dihasilkan masih sama (sekitar 20 FPS pada perangkat yang digunakan), dan penggunaan RAM hampir sama, walaupun ada peningkatan penggunaan CPU dan baterai, tetapi cukup kecil untuk tetap dapat digunakan pada perangkat *mobile*.
- Akurasi pelacakan lokasi tergantung dari perangkat yang digunakan, akurasi rata-rata pada detik ke 25 cukup kecil pada perangkat pertama (1.59 meter), dan pada perangkat kedua (4.4

meter), tetapi perangkat kedua membutuhkan waktu yang lebih lama untuk mendapatkan hasil yang akurat, sehingga jika hasil yang didapatkan kurang akurat dapat menggunakan pelacakan cadangan berupa *image tracking*, Selain itu *image tracking* juga dapat digunakan untuk membedakan lokasi antar lantai atau ketinggian.

### 5.2 Saran

- Membuat mode khusus untuk membedakan fotografi *indoor* maupun *outdoor*
- Menggunakan penanda khusus untuk setiap karakter yang disediakan pada aplikasi
- Dapat melakukan navigasi dari lokasi pengguna sampai ke lokasi penanda yang diinginkan
- Menggunakan objek 3D virtual berupa karakter maskot yang dibuat dan didesain sendiri oleh Universitas Kristen Petra.
- Membuat aplikasi yang dapat dijalankan pada sistem operasi iOS.

## 6. DAFTAR REFERENSI

- [1] Akenine-Möller, T., Haines, E., & Hoffman, N. 2018. *Real-Time Rendering*. A K Peters.
- [2] Aromaa, S., Väättänen, A., Aaltonen, I., Goriachev, V., Helin, K., & Karjalainen, J. 2020. Awareness of the real-world environment when using augmented reality head-mounted display. *Applied Ergonomics*, 88, 103145. doi:10.1016/j.apergo.2020.103145
- [3] Díaz-García, J., Brunet, P., Navazo, I., & Vázquez, P.-P. 2018. Progressive ray casting for volumetric models on mobile devices. *Computers & Graphics*, 73, 1-16. doi:10.1016/j.cag.2018.02.007
- [4] Frahm, J.-m., Koeser, K., Grest, D., & Koch, R. 2005. Markerless augmented reality with light source estimation for direct illumination. In *Conference on Visual Media Production CVMP*, 211-220.
- [5] Gomez-Jauregui, V., Manchado, C., Del-Castillo-Igareda, J., & Otero, C. 2019. Quantitative evaluation of overlaying discrepancies in mobile augmented reality applications for AEC/FM. *Advances in Engineering Software*, 127, 124-140. doi:10.1016/j.advengsoft.2018.11.002
- [6] Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., & Hujun, B. 2019. Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality. *Virtual Reality & Intelligent Hardware*, 1(4), 386-410. doi:10.1016/j.vrih.2019.07.002
- [7] Kán, P., & Kafumann, H. 2019. DeepLight: light source estimation for augmented reality using deep learning. *The Visual Computer*, 35(6-8), 873-883. doi:10.1007/s00371-019-01666-x
- [8] Kowalczyk, P., Siepmann, C., & Adler, J. 2021. Cognitive, affective, and behavioral consumer responses to augmented reality in e-commerce: A comparative study. *Journal of Business Research*, 124, 357-373. doi:10.1016/j.jbusres.2020.10.050
- [9] Liono, R. A., Amanda, N., Pratiwi, A., & Gunawan, A. A. 2021. A Systematic Literature Review: Learning with Visual by The Help of Augmented Reality Helps Students Learn

- Better. *Procedia Computer Science*, 179, 144-152. doi:10.1016/j.procs.2020.12.019
- [10] Marques, B. A., Clua, E. W., & Vasconcelos, C. N. 2018. Deep spherical harmonics light probe estimator for mixed reality games. *Computers & Graphics*, 76, 96-106. doi:10.1016/j.cag.2018.09.003
- [11] Meenakshi, S. V., Vasudevan, S. K., Ritesh, A., & Santhosh, C. 2015. An Innovative App with for Location Finding with Augmented Reality Using CLOUD. *Procedia Computer Science*, 50, 585-589. doi:10.1016/j.procs.2015.04.088
- [12] Michel, T., Genevès, P., Fourati, H., & Layaida, N. 2018. Attitude estimation for indoor navigation and augmented reality with smartphones. *Pervasive and Mobile Computing*, 46, 96-121. doi:10.1016/j.pmcj.2018.03.004
- [13] Moezzi, R., Krcmarik, D., Hlava, J., & Cýrus, J. 2020. Hybrid SLAM modelling of autonomous robot with augmented reality device. *Materials Today: Proceedings*, 32(2), 103-107. doi:10.1016/j.matpr.2020.03.036
- [14] Nikhashemi, S., Knight, H. H., Nusair, K., & Liat, C. B. 2021. Augmented reality in smart retailing: A (n) (A) Symmetric Approach to continuous intention to use retail brands' mobile AR apps. *Journal of Retailing and Consumer Services*, 60, 102464. doi:10.1016/j.jretconser.2021.102464
- [15] Pantuwong, N. 2016. A tangible interface for 3D character animation using augmented reality technology. *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1-6. doi:10.1109/ICITEED.2016.7863263
- [16] Reddy, M. 2011. *API Design for C++*. Elsevier Science.
- [17] Roberto, R., Lima, J. P., Uchiyama, H., Teichrieb, V., & Taniguchi, R.-i. 2019. Geometrical and statistical incremental semantic modeling on mobile devices. *Computers & Graphics*, 84, 199-211. doi:10.1016/j.cag.2019.09.003
- [18] Smink, A. R., Reijmersdal, E. A., Noort, G. v., & Neijens, P. C. 2020. Shopping in augmented reality: The effects of spatial presence, personalization and intrusiveness on app and brand responses. *Journal of Business Research*, 118, 474-485. doi:10.1016/j.jbusres.2020.07.018
- [19] Teichrieb, V., Lima, M., Lourenc, E., Bueno, M. A., Kelner, J., & Santos, I. H. 2007. A Survey of Online Monocular Markerless Augmented Reality. *International Journal of Modeling and Simulation for the Petroleum Industry*, 1-7.
- [20] Zhang, H., Zhang, J., Yin, X., Zhou, K., & Pan, Z. 2020. Cloud-to-end Rendering and Storage Management for Virtual Reality in Experimental Education. *Virtual Reality & Intelligent Hardware*, 2(4), 368-380. doi:10.1016/j.vrih.2020.07.001