

Augmented Reality Berbasis Cloud Recognition pada Majalah Film

YOULLIA INDRAWATY N , JASMAN PARDEDE, AFIS SISWANTINI

Jurusan Teknik Informatika Institut Teknologi Nasional, Bandung

Email: youllia@itenas.ac.id

ABSTRAK

Film telah menjadi salah satu hiburan bagi manusia termasuk pencinta film itu sendiri. Untuk menyampaikan informasi mengenai film di media lain, maka diterapkan teknologi Augmented Reality pada majalah melalui smartphone. Penerapan Augmented Reality berbasis cloud computing dapat memudahkan pihak redaksi mengunggah dan mengubah konten agar dapat ditampilkan serta diakses oleh pembaca siapa saja dan kapan saja dengan terkoneksi internet. Layanan yang dapat menggabungkan teknologi Augmented Reality dan cloud computing adalah Vuforia Cloud Recognition. Maka dari itu, diuji bagaimana kualitas marker mempengaruhi kinerja pendeteksian marker dengan tiga kondisi marker yaitu pudar, miscolour, dan blur dengan berbagai intensitas. Dari hasil penelitian, didapat data bahwa marker dengan intensitas keputaran 25%, semua marker tidak dapat dikenali karena struktur gambar berubah. Pada marker dengan keadaan miscolour, semua marker dapat dikenali karena struktur gambar tidak berubah. Sedangkan pada marker dengan keadaan blur dengan intensitas 75%, semua marker tidak dapat dikenali karena struktur gambar berubah. Ini membuktikan bahwa struktur gambar mempengaruhi kinerja pendeteksian marker.

Kata Kunci : *Film, Augmented Reality, Cloud Computing*

ABSTRACT

Film has become one of entertainment for humans as well as the movie lovers itself. To deliver information about movie on the other media, it will be applied Augmented Reality magazine via smartphones so consumers are more interested. The application will facilitate the editors when it's cloud-based computing, where content is created and uploaded by the editors can be displayed and accessed by the reader anybody and at any time with internet connected. The service which can integrate augmented reality and cloud computing is Vuforia Cloud Recognition. From the result, marker with 25% intensity of fade, all markers cannot be detected because the image structure is changing. Markers with miscolour condition still can be detected because the image structure is not changing. Otherwise, the markers with blur condition with 75% intensity cannot be detected because the image structure is changing. It means the image structure can affect marker scanning.

Keywords : *Movie, Augmented Reality, Cloud Computing*

1. PENDAHULUAN

Film merupakan salah satu hiburan bagi manusia termasuk pencinta film itu sendiri. Beberapa cara penyampaian informasi mengenai film yang dapat kita jumpai yaitu melalui media cetak seperti majalah, tabloid, dan koran. Inovasi yang dilakukan diantaranya tidak hanya menerbitkan dalam bentuk media cetak namun juga dalam bentuk *softcopy* (digital). Saat ini pun banyak tersedia website-website maupun aplikasi mobile mengenai informasi film. Untuk menyampaikan informasi mengenai film di media lain, maka diterapkan teknologi *Augmented Reality* pada *majalah* melalui *smartphone*.

Penerapan aplikasi AR dapat memanfaatkan internet atau tanpa menggunakan internet. Kelebihan aplikasi yang menggunakan internet adalah kemungkinan untuk terus-menerus memperbaharui data yang disajikan. Data disimpan dalam sebuah server, dimana redaksi dapat menyimpan dan memanggil data, penggunaan teknologi semacam ini sering disebut sebagai *Cloud Computing*. Layanan yang dapat menggabungkan teknologi AR dan *cloud computing* adalah Vuforia Cloud Recognition. Masalah yang didapat tentunya tidak hanya bagaimana menerapkan teknologi *Cloud Computing* pada aplikasi AR, tetapi juga pada *marker*. AR bekerja berdasarkan deteksi citra, dan citra yang digunakan adalah sebagai *marker*. Sedangkan *marker* terdapat di majalah versi cetak. Kesalahan pada saat pencetakan majalah dapat mempengaruhi proses *scanning*.

Maka dilakukan penelitian mengenai bagaimana kualitas marker dapat mempengaruhi kinerja Vuforia dalam pendeteksian *marker* dengan menggunakan algoritma FAST sebagai pendeteksi *feature point* yang terdapat di *marker* dan algoritma Brute-Force untuk pencocokkan *feature point* yang terdapat di *marker* acuan yang ada di majalah.

2. METODOLOGI PENELITIAN

Metodologi penelitian ini meliputi metoda pengumpulan data dan metode pengembangan sistem.

2.1 Metode Pengumpulan Data

Berikut metode pengumpulan data yang digunakan dalam melakukan penelitian ini diantaranya:

1. Rancangan Penelitian

Jenis penelitian yang digunakan adalah *Excisting Variable* dan pengembangan. *Excisting Variable* yaitu penelitian yang dilakukan dengan menggunakan data yang diperoleh dari buku, jurnal dan artikel sejenis yang telah dilakukan dari penelitian sebelumnya, selanjutnya penelitian pengembangan yaitu mengembangkan dari penelitian yang telah dilakukan sebelumnya (Sugiyono, 2002).

2. Subjek Penelitian

Subjek penelitian pada penelitian ini adalah marker dengan berbagai kondisi yang dicetak di atas media kertas, yang dideteksi menggunakan kamera *smartphone* yang digunakan untuk menganalisis efektifitas pendeteksian dan parameter yang mempengaruhi hasil pendeteksian. Teknik pengambilan informasi yang digunakan adalah *purpose sampling* yaitu dengan mengambil subjek penelitian yang memenuhi kriteria. Dimana kriteria tersebut dibuat oleh peneliti sendiri (Nordwood, 2000).

3. Pengumpulan Data

Metode pengumpulan data yang dilakukan adalah dengan melakukan studi literatur.

4. Studi Literatur

Studi ini dilakukan dengan cara mencari sekaligus mempelajari beberapa literatur dan artikel mengenai *Augmented Reality* sebagai acuan dalam perencanaan dan pembuatan sistem atau aplikasi.

2.2 Metode Pengembangan Sistem

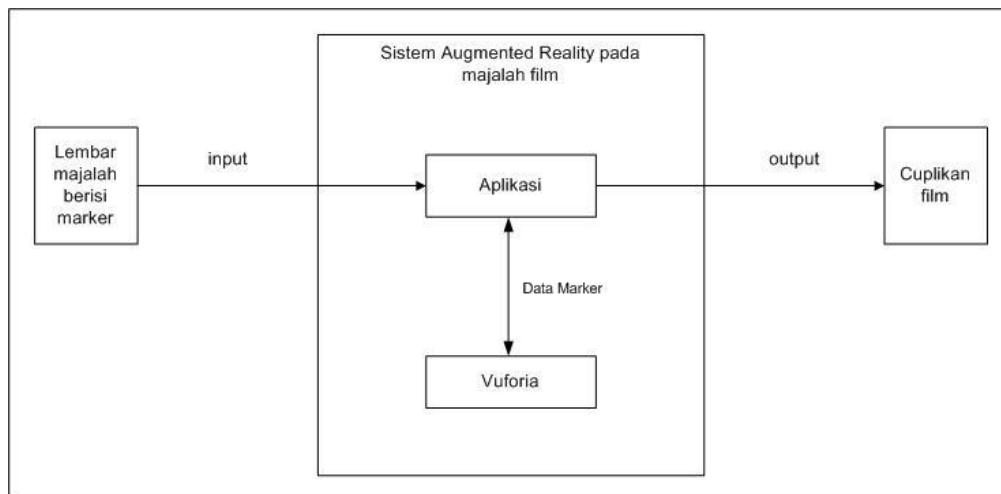
Metode pengembangan sistem yang digunakan dalam penelitian ini yaitu Model *Prototype*. Aktivitas dalam membangun *prototype* adalah sebagai berikut:

1. Mempelajari Augmented Reality dan Vuforia Cloud Recognition.
2. Menentukan fitur sistem yang dibutuhkan untuk aplikasi penerapan Augmented Reality berbasis Cloud Recognition pada majalah film.
3. Membangun Prototype dengan menggunakan bahasa pemrograman C#
4. Evaluasi hasil prototype dengan menggunakan aplikasi tambahan menggunakan bahasa pemrograman Python dengan menggunakan algoritma FAST dan Brute-Force.
5. Menyempurnakan prototype sesuai dengan kebutuhan dan desain awal sistem.

3. ANALISIS DAN PEMBAHASAN

3.1 Proses Kerja Aplikasi

Dimulai dari lembar majalah yang berisi *marker* berupa poster film dipindai dengan kamera pada aplikasi sebagai input. Setelah itu aplikasi mengirimkan data *marker* untuk dicocokkan dengan *marker* acuan yang telah diunggah di *cloud database* Vuforia. Jika *marker* cocok, Vuforia mengirimkan metadata dari *marker* ke aplikasi berupa URL video untuk menampilkan cuplikan film sebagai output.

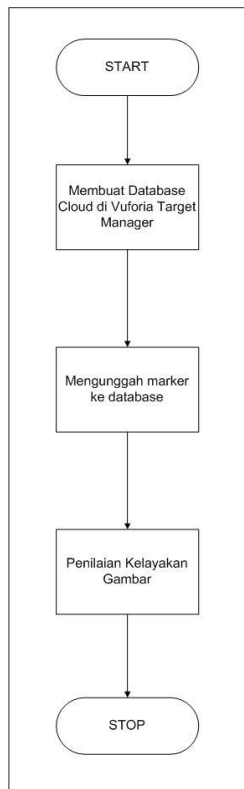


Gambar 1. Proses Kerja Aplikasi

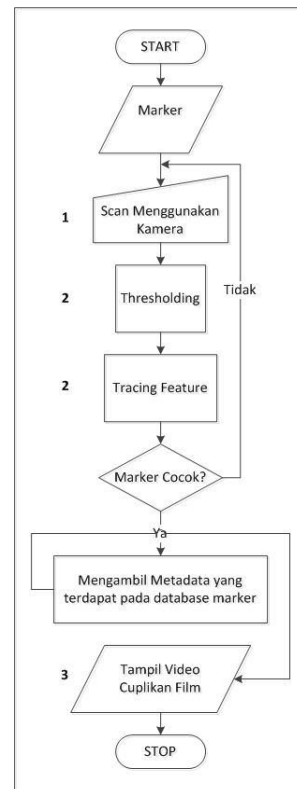
3.2 Flowchart Sistem

Flowchart terdiri dari *flowchart* proses input *marker* ke *database cloud*. Dimulai dari membuat *database cloud* di Vuforia Target Manager. Lalu mengunggah *marker* ke *database*, setelah itu dinilai kelayakan gambar dengan hasil berupa rating 1 sampai 5. *Flowchart* sistem secara umum hanya terdiri dari pemindaian *marker* dengan kamera. Setelah aplikasi terbuka, kamera telah siap untuk memindai *marker*. *Marker* akan diproses oleh Vuforia untuk dilacak *feature*-nya untuk mencocokkan *feature marker* yang dipindai dengan *feature marker* yang ada di

database dengan proses *thresholding* lalu *tracing feature*. Jika *marker* cocok dengan *marker* yang berada di database, maka Vuforia akan mengambil metadata berupa URL video yang terdapat pada database *marker*, lalu muncul video cuplikan film. Tetapi jika proses pencocokan *marker* tidak berhasil, maka proses *scanning* akan terus berjalan untuk mengulang kembali pemindaian *marker*.



Gambar 2. Flowchart input marker ke database cloud



Gambar 3. Flowchart Sistem

Keterangan *Flowchart* Sistem:

1. Scan Menggunakan Kamera

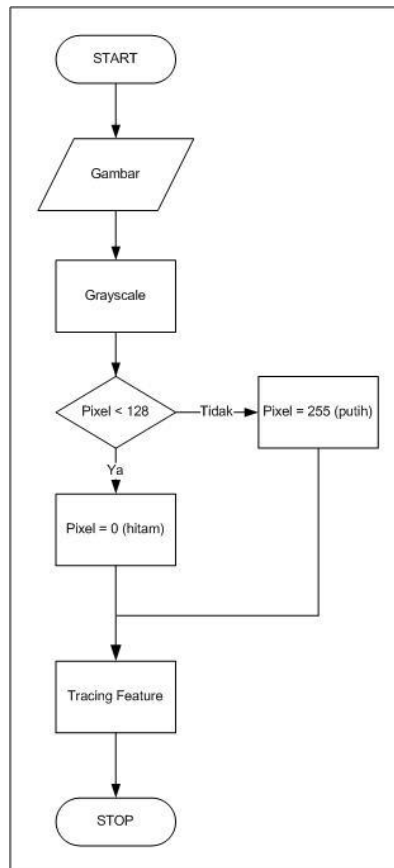
Pembaca mengaktifkan aplikasi, kemudian mengarahkan smartphone ke poster film yang berada di majalah.



Gambar 4. Scan Menggunakan Kamera

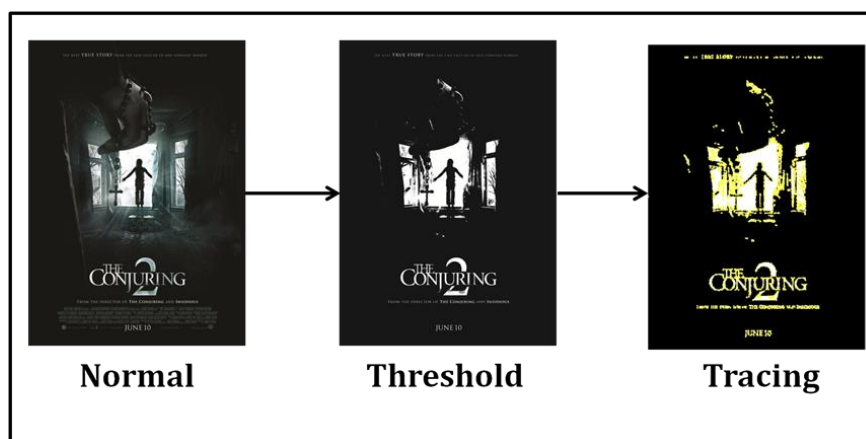
2. Thresholding

Proses *thresholding* dimulai dari gambar *marker* sebagai input. Pada proses *thresholding*, gambar *marker* diubah citranya menjadi abu-abu (*Grayscale*) terlebih dahulu untuk mengetahui segmentasi pixel. Pixel yang memiliki nilai intensitas di bawah 128 diubah menjadi warna hitam (nilai intensitas = 0), sedangkan pixel yang memiliki nilai intensitas diatas 128 diubah menjadi warna putih (nilai intensitas = 255).



Gambar 5. Flowchart Thresholding dan Tracing Feature

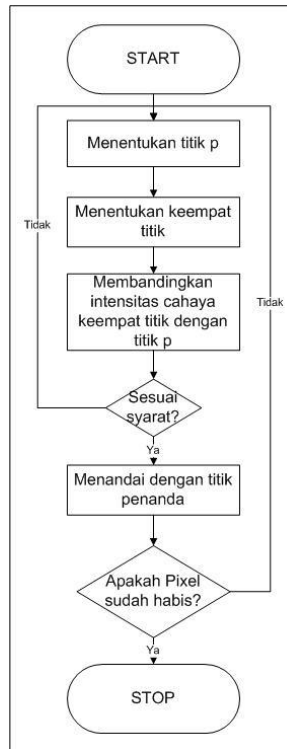
Berikut contoh hasil proses *thresholding* dan *tracing feature*:



Gambar 6. Contoh Hasil Proses Thresholding dan Tracing Feature

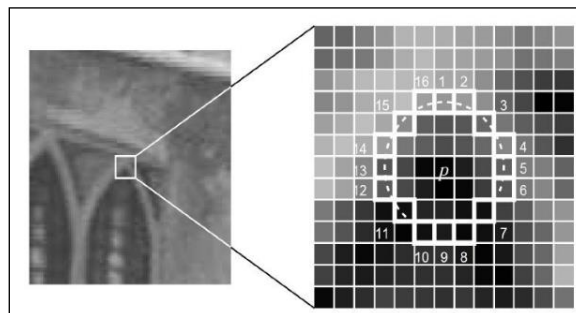
3. *Tracing feature* dan pencocokkan *feature point*

Gambar 7 merupakan proses *tracing feature* dengan menggunakan algoritma FAST dan diimplementasikan dengan bahasa Python dan OpenCV sebagai library dalam pemrosesan *tracing feature*.



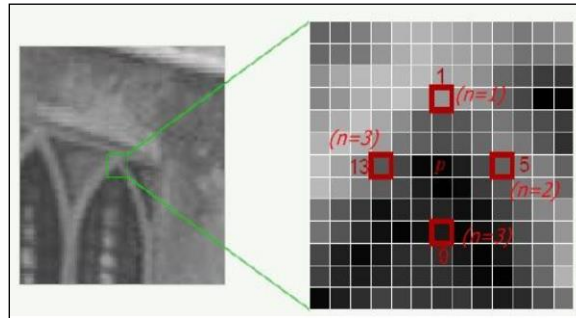
Gambar 7. Proses tracing feature menggunakan algoritma FAST

- Tentukan sebuah titik p pada citra dengan posisi awal (x_p, y_p) .



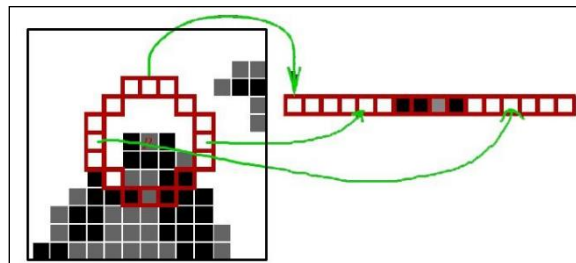
Gambar 8. Titik awal yang diuji

- Tentukan keempat titik. Titik pertama $(n=1)$ terletak pada koordinat (x_p, y_p+3) , titik kedua $(n=2)$ terletak pada koordinat (x_p+3, y_p) , titik ketiga terletak pada koordinat $(n=3)$ terletak pada koordinat (x_p, y_p-3) , titik keempat $(n=4)$ terletak pada koordinat (x_p-3, y_p) .



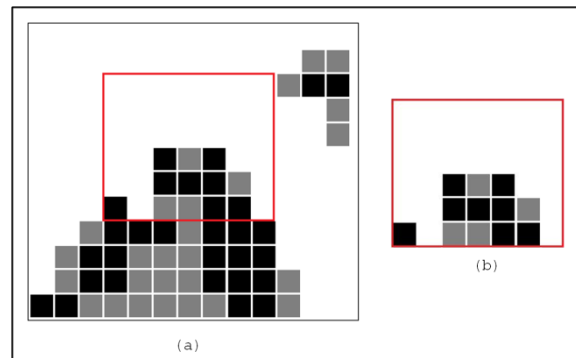
Gambar 9. Keempat titik koordinat

- Bandingkan intensitas titik pusat p dengan keempat titik disekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat p adalah titik sudut.
 - Untuk menentukan titik suatu sudut, seluruh piksel akan dibagi dengan tiga subset yaitu ; Pixel dark, Pixel similiar, dan Pixel brighter.
- Ulangi proses sampai seluruh titik pada citra sudah dibandingkan intensitasnya.



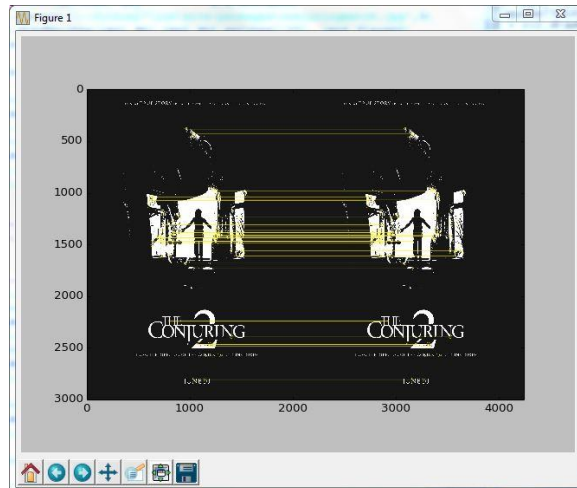
Gambar 10. Tiga titik yang memenuhi syarat FAST Corner Detection

Untuk pencocokan feature point, digunakan algoritma Brute-Force sebagai berikut.



Gambar 11. Pixel gambar utama (a) dan pixel gambar potongan yang akan dicocokkan (b)

Setelah mendapatkan potongan pixel-pixel yang akan dicocokkan, permasalahan yang semula pencarian lokasi potongan gambar menjadi pencarian lokasi potongan pixel. Pertanyaannya menjadi "Dimanakah koordinat pixel potongan di dalam pixel gambar utama?". Pada tahap selanjutnya, kita akan memakai konsep *pattern matching* dalam pencarian koordinat. Idanya adalah mencari tingkat kemiripan dari setiap koordinat. Kita akan menelusuri setiap elemen di Pixel-pixel gambar utama untuk mendapatkan tuple(pasangan)koordinat-tingkat kemiripan. Setelah tracing feature diimplementasikan oleh bahasa Python, marker dicocokkan feature pointnya dengan algoritma Brute-Force seperti yang telah dijelaskan di Gambar 11. Feature yang cocok ditandai dengan garis berwarna kuning.



Gambar 12. Contoh hasil proses matching feature

4. Tampil video cuplikan film

Jika *marker* cocok, Vuforia mengirimkan metadata dari *marker* ke aplikasi berupa URL video untuk menampilkan cuplikan film sebagai output.



Gambar 6 : Cuplikan Film yang Ditampilkan

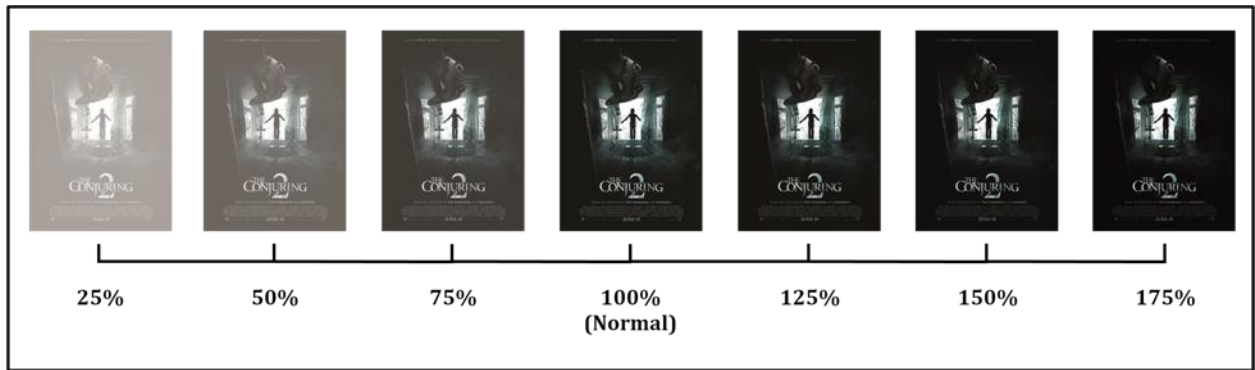
3.3 Pengujian

Untuk menguji bagaimana kualitas dan keadaan gambar *marker* dapat mempengaruhi pemindaian, maka disediakan gambar *marker* dengan berbagai kondisi seperti normal, pudar, *miscolour*, dan *blur*. Dimana:

1. **Normal dan Pudar.**

Normal: Keadaan gambar *marker* asli dari pembuat poster tanpa adanya perubahan struktur dan warna pada gambar. *Marker* yang normal ini lah yang menjadi acuan untuk mencocokkan *marker* yang dipindai.

Pudar: Keadaan gambar *marker* lebih pudar dari gambar *marker* yang normal. Dimana keadaan gambar normal memiliki nilai *opacity* 100%. Jika nilai kepudaran kurang dari 100% maka gambar semakin pudar dan jika nilai kepudaran lebih dari 100% maka gambar semakin jelas.



Gambar 7 : Marker Pudar dengan Tingkatan Intensitas yang Berbeda

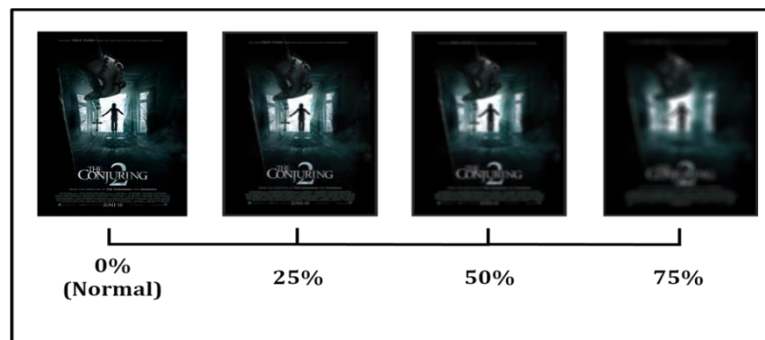
2. **Miscolour dan Blur.**

Miscolour: Keadaan warna gambar *marker* tidak sesuai dengan gambar *marker* yang normal.



Gambar 8 : Marker dengan Keadaan Miscolour

Blur. Keadaan gambar kabur atau tidak jelas. Dimana keadaan gambar normal memiliki nilai *blur* 0%. Jika nilai *blur* lebih dari 0% maka gambar semakin kabur.



Gambar 9 : Marker dengan Keadaan Blur

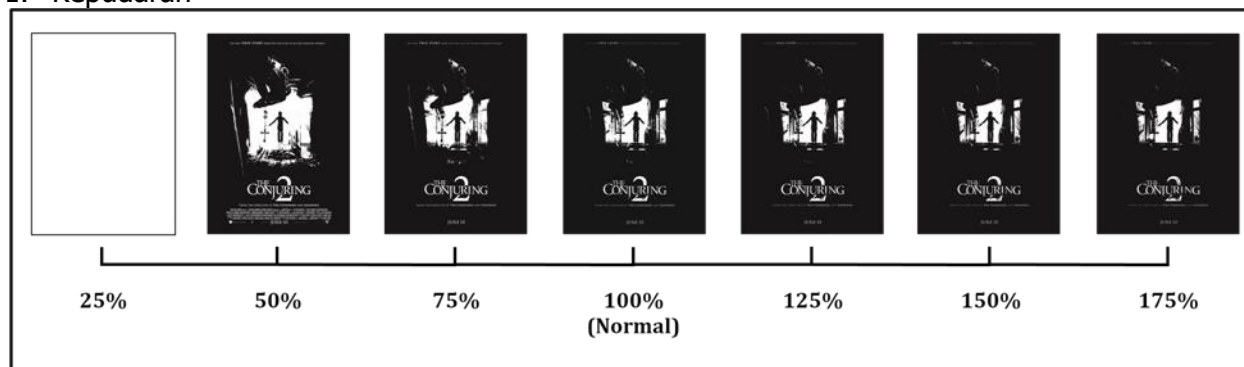
Tabel 1. Hasil pengujian marker opacity, miscolour, dan blur.

	Percobaan ke -	Opacity						MiscoLOUR	Blur		
		0.25	0.5	0.75	1 (Normal)	1.25	1.5		0.25	0.5	0.75
Marker 1 (Conjuring 2)	1	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	2	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	3	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	4	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	5	*	✓	✓	✓	✓	✓	✓	✓	✓	*
Persentase terdeteksi		0%	100%	100%	100%	100%	100%	100%	100%	100%	0%
Marker 2 (Ghostbuster)	1	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	2	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	3	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	4	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	5	*	✓	✓	✓	✓	✓	✓	✓	✓	*
Persentase terdeteksi		0%	100%	100%	100%	100%	100%	100%	100%	100%	0%
Marker 3 (The Secret Life of Pets)	1	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	2	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	3	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	4	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	5	*	✓	✓	✓	✓	✓	✓	✓	✓	*
Persentase terdeteksi		0%	100%	100%	100%	100%	100%	100%	100%	100%	0%
Marker 4 (Star Trek)	1	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	2	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	3	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	4	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	5	*	✓	✓	✓	✓	✓	✓	✓	✓	*
Persentase terdeteksi		0%	100%	100%	100%	100%	100%	100%	100%	100%	0%
Marker 5 (The Legend of Tarzan)	1	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	2	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	3	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	4	*	✓	✓	✓	✓	✓	✓	✓	✓	*
	5	*	✓	✓	✓	✓	✓	✓	✓	✓	*
Persentase terdeteksi		0%	100%	100%	100%	100%	100%	100%	100%	100%	0%

3.4 Analisis Pengujian

Hasil pengujian menunjukkan bahwa:

1. Kepudaran



Gambar 10 : Marker Dengan Keadaan Pudar yang telah di-Threshold

Marker dengan intensitas pudar 50%, 75%, 125%, 150%, dan 175% masih terbaca dengan baik karena garis bentuk gambar yang telah di-threshold terlihat jelas. Sedangkan marker dengan intensitas pudar 25% tidak terbaca karena garis bentuk gambar tidak ada sama sekali.

2. MiscoLOUR

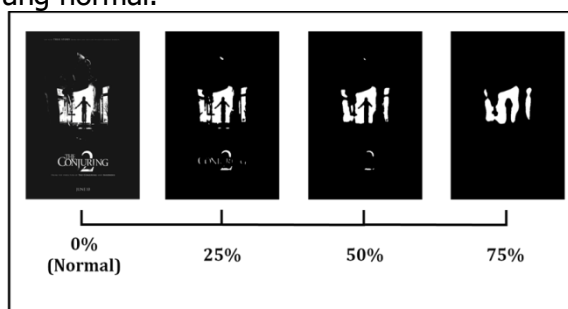


Gambar 11 : Marker dengan Keadaan Miscalour yang telah di-Threshold

Pada pengujian, terlihat bahwa keadaan *miscalour* tidak mempengaruhi proses pencocokan *marker*. Gambar *marker threshold* yang telah di-*recolour* tidak berbeda dari gambar *threshold* yang normal. Ini menunjukkan bahwa struktur gambar tidak berubah saat proses *recolour*.

3. Blur

Tingkat kejelasan gambar *marker* mempengaruhi terbacanya *marker* karena struktur gambar berubah. *Marker* dengan intensitas *blur* 25% dan 50% masih terbaca dengan baik karena garis bentuk gambar masih terlihat jelas dan *feature point*-nya masih cocok dengan *feature point marker* yang normal.



Gambar 12 : Marker dengan Keadaan Miscalour yang telah di-Threshold

KESIMPULAN

Kesimpulan yang didapat dari hasil penelitian penerapan *Augmented Reality* berbasis *Cloud Recognition* pada majalah film adalah sebagai berikut.

4. Algoritma FAST dan Brute-Force mampu mendeteksi dan mencocokkan *feature point* yang ada pada *marker*.
5. Tingkat kepedaran *marker* yang telah dicetak di majalah mempengaruhi proses pemindaian *marker*. *Marker* dengan intensitas pudar dibawah 25% tidak terbaca.
6. Perubahan warna *marker* tidak mempengaruhi proses pemindaian *marker*.
7. Perubahan struktur gambar mempengaruhi proses pemindaian *marker*.

DAFTAR RUJUKAN

- [1] Rujianto Eko Saputro, Dhanar Intan Surya Saputra (2014). Pengembangan Media Pembelajaran Mengenal Organ Pencernaan Manusia Menggunakan Teknologi *Augmented Reality*. STMIK AMIKOM Purwokerto.
- [2] Muhammad Rifa'i, Tri Listyorini, Anastasya Latubessy (2014). Penerapan Teknologi *Augmented Reality* Pada Aplikasi Katalog Rumah Berbasis Android. Universitas Muria Kudus.

- [3] Youllia Indrawaty, Rio Korio Utoro, Mochamad Dika Lathifuddin (2014). Pengembangan Aplikasi Digital Pet Dengan Implementasi Markerless Augmented Reality. Institut Teknologi Nasional.
- [4] _____, 'Apa itu Augmented Reality?', Augmented Reality Indonesia, www.augmentedrealityindonesia.com
- [5] Yoze Rizki, Mochamad Hariadi (2013). *Markerless Augmented Reality* Pada Perangkat Android. ITS Surabaya.