

Investigasi Pengaruh Skema *Stride* dan *Step Training* untuk Deteksi Jari Pada Region-based Fully Convolutional Network (R-FCN) dalam Teknologi *Augmented Reality*

Hashfi Fadhillah¹, Suryo Adhi Wibowo², Rita Purnamasari³

^{1,2,3}Fakultas Teknik Elektro, Universitas Telkom

Jl. Telekomunikasi, Terusan Buah Batu, Sukapura, Kec. Dayeuhkolot, Bandung 40257

E-mail : ¹hasviva@student.telkomuniversity.ac.id, ²suryoadhiwibowo@telkomuniversity.ac.id,

³ritapurnamasari@telkomuniversity.ac.id

Abstract

Combining the real world with the virtual world and then modeling it in 3D is an effort carried on Augmented Reality (AR) technology. Using fingers for computer operations on multi-devices makes the system more interactive. Marker-based AR is one type of AR that uses markers in its detection. This study designed the AR system by detecting fingertips as markers. This system is designed using the Region-based Deep Fully Convolutional Network (R-FCN) deep learning method. This method develops detection results obtained from the Fully Connected Network (FCN). Detection results will be integrated with a computer pointer for basic operations. This study uses a predetermined step scheme to get the best IoU parameters, precision and accuracy. The scheme in this study uses a step scheme, namely: 25K, 50K and 75K step. High precision creates centroid point changes that are not too far away. High accuracy can improve AR performance under conditions of rapid movement and improper finger conditions. The system design uses a dataset in the form of an index finger image with a configuration of 10,800 training data and 3,600 test data. The model will be tested on each scheme using video at different distances, locations and times. This study produced the best results on the 25K step scheme with IoU of 69%, precision of 5.56 and accuracy of 96%.

Keyword: Augmented Reality, Region-based Convolutional Network, Fully Convolutional Network, Pointer, Step training

Abstrak

Menggabungkan dunia nyata dengan dunia virtual lalu memodelkannya bentuk 3D merupakan upaya yang diusung pada teknologi *Augmented Reality* (AR). Menggunakan jari untuk operasi komputer pada *multi-device* membuat sistem yang lebih interaktif. *Marker-based AR* merupakan salah satu jenis AR yang menggunakan marker dalam deteksinya. Penelitian ini merancang sistem AR dengan mendeteksi ujung jari sebagai *marker*. Sistem ini dirancang menggunakan metode *deep learning* Region-based Fully Convolutional Network (R-FCN). Metode ini mengembangkan hasil deteksi yang didapat dari Fully Connected Network (FCN). Hasil deteksi akan diintegrasikan dengan *pointer* komputer untuk operasi dasar. Penelitian ini menggunakan skema *step training* yang telah ditentukan untuk mendapatkan parameter IoU, presisi dan akurasi yang terbaik. Skema pada penelitian ini menggunakan skema *step* yaitu: 25K, 50K dan 75K *step*. Presisi tinggi menciptakan perubahan titik *centroid* yang tidak terlalu jauh. Akurasi yang tinggi dapat meningkatkan kinerja AR dalam kondisi pergerakan yang cepat dan kondisi jari yang tidak tepat. Perancangan sistem menggunakan dataset berupa citra jari telunjuk dengan konfigurasi 10.800 data latih dan 3.600 data uji. Model akan diuji pada tiap skema dilakukan menggunakan video pada jarak, lokasi dan waktu yang berbeda. Penelitian ini menghasilkan hasil terbaik pada skema *step* 25K dengan IoU sebesar 69%, presisi sebesar 5,56 dan akurasi sebesar 96%.

Kata kunci: Augmented Reality, Region-based Convolutional Network, Fully Convolutional Network, Pointer, Step training

I. PENDAHULUAN

Menggabungkan dunia nyata dengan dunia virtual merupakan upaya yang diusung pada teknologi *Virtual Reality*. Perangkat *Virtual Reality* hanya dapat menampilkan tampilan pada sebuah media dan manusia dapat berinteraksi dengan lingkungan buatan namun tidak secara langsung dengan lingkungan nyata [1]. Teknologi tersebut terus dikembangkan hingga diciptakan teknologi terbaru yaitu *Augmented Reality* (AR). AR merupakan teknologi yang menggabungkan dunia nyata dengan dunia virtual secara *real-time* dan memodelkannya dalam bentuk 3D [2]. Penggunaan AR masih kerap dijumpai pada *device* dengan ukuran yang besar, untuk *device* dengan ukuran kecil seperti smartphone hanya sebatas memunculkan model 3D pada citra tangkapan kamera. Menciptakan teknologi AR yang lebih interaktif pada *multi-device* merupakan latar belakang penulis dalam melakukan penelitian.

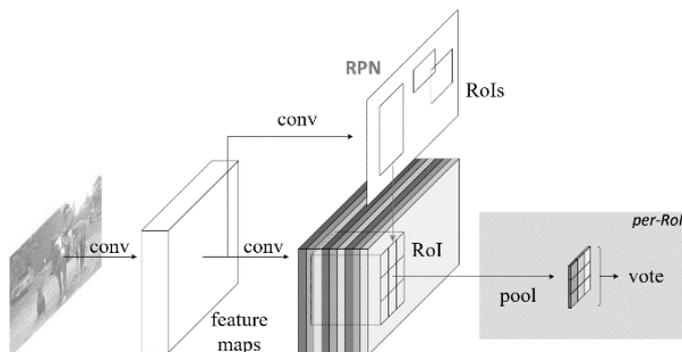
Teknologi ini terbagi menjadi 3 jenis yaitu: Spasial AR berbasis proyektor-kamera, AR berbasis gerakan tangan, dan *Mobile AR* [3]. Banyak penelitian yang mengembangkan teknologi AR. Penelitian *sixth-sense* menggunakan *particle filter* [4] dalam pendeteksiannya. Metode tersebut membutuhkan komputasi tinggi untuk pengolahannya. Penelitian *Augmented Reality* menggunakan ekstraksi tangan [5] [6] [7] juga telah dilakukan. Proses deteksi tersebut membutuhkan komputasi tinggi serta proses yang cukup lama karena harus melewati tahap ekstraksi terlebih dahulu sebelum terdeteksi sebagai tangan. Penulis akan merancang sistem AR menggunakan metode *deep learning* Region-based Fully Convolutional Network (R-FCN). Metode tersebut dapat mendeteksi objek kurang dari 1 detik dan membutuhkan komputasi yang cukup ringan [14]. Penelitian dilakukan menggunakan R-FCN dengan skema *training* tertentu untuk mendapatkan *tracking* yang lebih mulus.

Sistem AR yang penulis rancang dapat diintegrasikan dengan *pointer* komputer. Sebuah komputer dapat dioperasikan melalui gerakan jari. AR ini juga diharapkan dapat dijadikan *daily gear* atau perlengkapan sehari-hari manusia dalam menjalankan aktivitasnya. Dengan memanfaatkan teknologi ini manusia tidak perlu lagi membawa alat untuk menampilkan antarmuka pada sebuah layar. Dengan AR ini hampir semua permukaan padat dapat dijadikan sebagai layar untuk pengoperasiannya, sehingga interaksi AR dapat dilakukan dimana saja. Dari penelitian ini diharapkan dapat merancang sebuah alat yang dapat menggabungkan dunia nyata dengan dunia digital dan dapat diterapkan di berbagai sektor salah satunya sektor pendidikan

II. METODOLOGI PENELITIAN

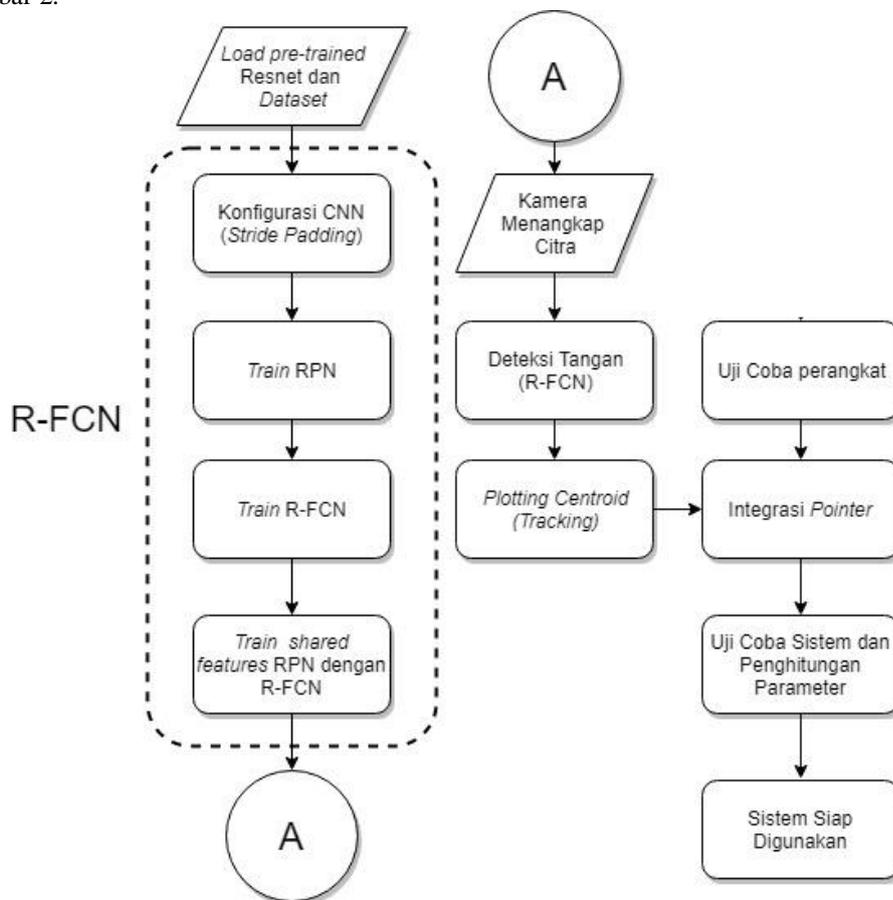
Rancangan ini menggunakan *dataset* berupa jari dengan sistematika yang telah ditentukan, metode Region-based Fully Convolutional Network (R-FCN) digunakan untuk mendeteksi citra jari dari video. *Tracking* dilakukan untuk membentuk sebuah gerakan jari atau gestur. Gerakan ini digunakan untuk mengoperasikan perangkat seperti komputer.

R-FCN merupakan metode deteksi yang berbasis pada region/wilayah. R-FCN merupakan pengembangan dari metode Fully Convolutional Network (FCN). Dengan mendapatkan keluaran dari FCN, R-FCN menambahkan lapisan Region of Interest (RoI) yang peka terhadap posisi. R-FCN juga mengadopsi metode objek deteksi seperti Region Proposal Network (RPN) dan Region Classification. RPN merupakan *Neural Network* yang berguna untuk mengajukan region dari penghitungan. RPN akan menghasilkan *bounding box*. Pada tiap *bounding box* yang dihasilkan memiliki 2 nilai probabilitas atau *vote* untuk memvalidasi pada region tersebut terdapat objek atau tidak. RPN akan mendeteksi citra dengan beberapa ukuran skala dan rasio. Pada RPN terdapat 9 tipe *anchor* secara keseluruhan. *Anchor* berfungsi untuk menghasilkan kemungkinan terdapat objek pada region tertentu. RPN akan mengabaikan *cross-boundary anchor* dan menerapkan Non-max Supression (NMS) untuk mendapatkan satu anchor memiliki nilai Intersect of Union tertinggi pada salah satu objek [15]. R-FCN juga menggunakan RoI yang dihasilkan oleh RPN. R-FCN dan RPN akan saling membagi fitur untuk mendapatkan region yang nantinya R-FCN akan mengklasifikasi antara *background* dengan objek. Keluaran dari R-FCN berupa *position-sensitive RoI pooling layer*. Arsitektur dari R-FCN dapat diilustrasikan pada gambar 1.



Gambar 1. Arsitektur R-FCN yang menerapkan Region Proposal Network atau RPN

Untuk merancang suatu sistem maka dibuat diagram blok perancangan sehingga perancangan dapat berjalan dengan sistematis. Diagram blok dirancang mulai dari *load* dataset hingga ujicoba sistem. Sistem diuji pada video yang telah ditentukan. Proses pada R-FCN terpisah dimulai dari konfigurasi *Convolutional Neural Network* (CNN), *Train RPN*, *Train R-FCN* hingga *Train shared features RPN* dengan R-FCN. Diagram blok perancangan seperti terlihat pada Gambar 2.

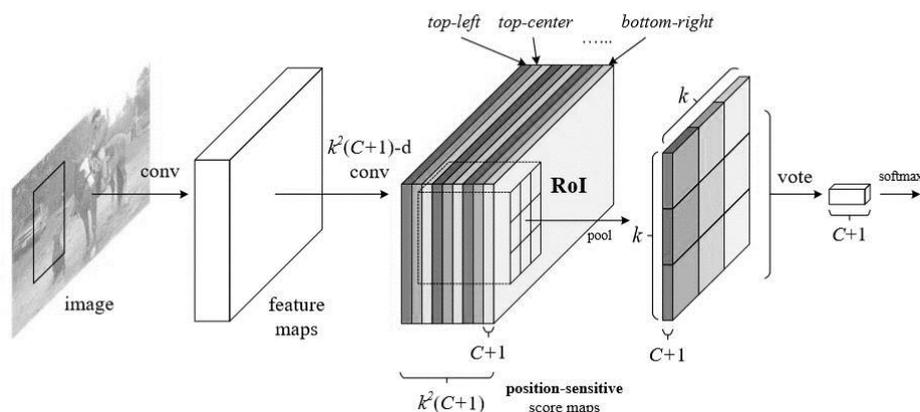


Gambar 2. Diagram blok perancangan sistem *Augmented Reality* berbasis gerakan jari

Perancangan ini menggunakan jaringan konvolusi Residual Network (ResNet). Resnet merupakan arsitektur Convolutional Neural Network (CNN) yang telah dilatih menggunakan dataset PASCAL VOC 2007. CNN memiliki dua metode yaitu klasifikasi menggunakan feedforward dan tahap pembelajarannya menggunakan backpropagation [8]. CNN juga dapat meningkatkan kinerja dari proses *tracking* pada objek [9]. Penelitian ini menggunakan Resnet-101 dengan 100 Convolutional Layer dan 80 kelas. Jumlah kelas yang dibutuhkan sebanyak 1 kelas. *Load* ini dilakukan sekali saat memulai sistem.

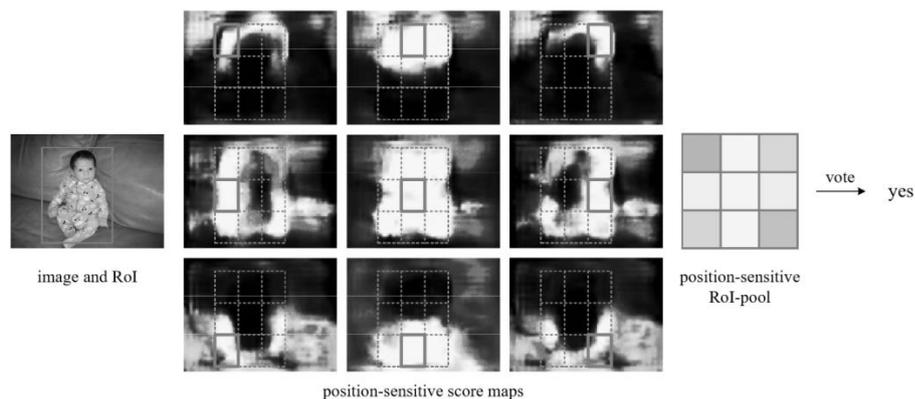
Stride menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka konvolusi filter akan bergeser sebanyak 1 *pixel* secara horizontal lalu vertikal. Semakin kecil *stride* maka akan semakin detail informasi yang didapatkan dari sebuah input, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar. *Stride* kecil tidak menjamin untuk mendapatkan kinerja yang baik. Penelitian ini menggunakan *stride* 32 *pixel* yang direduksi menjadi 16 *pixel*. Seluruh *convolutional filter* dimodifikasi menggunakan "hole algorithm" untuk mengimbangi dari pengurangan *stride*. *Convolutional layer* adalah layer yang terdiri dari neuron yang tersusun sehingga membentuk sebuah filter yang berfungsi untuk deteksi dan mengekstraksi ciri pada input yang diberikan [10]. *Convolutional layer* digunakan sebagai ekstraksi ciri / fitur pada citra masukan. Untuk klasifikasi menggunakan *fully-connected layer*.

Train RPN bertujuan untuk membentuk kandidat RoI untuk di-*propose*. Kandidat RoI yang telah di-*propose* lalu diterapkan pada penghitungan skor-map. Semua *layer* beban dilakukan secara konvolusional. Komputasi yang dilakukan pada tiap RoI diabaikan. *Train R-FCN* dilakukan menggunakan *backbone* Fully Convolutional Network (FCN). Untuk memasukkan variasi dari FCN, dibentuk satu *set position-sensitive score maps* dengan menggunakan kumpulan *specialized convolutional layers* sebagai keluaran FCN. Fitur dari CNN dapat meningkatkan visual tracking [11] [12]. Pooling layer dilakukan yang mereduksi dimensi dari feature map secara parsial menggunakan operasi down-sampling sehingga dapat mempercepat komputasi dan mengatasi overfitting. Fungsi pooling yang digunakan ada 3, yaitu max pooling, average pooling, dan norm pooling. Pooling yang umum dipakai adalah max pooling [13]. Proses membentuk *Position-sensitive score map* dapat dilihat pada Gambar 3.

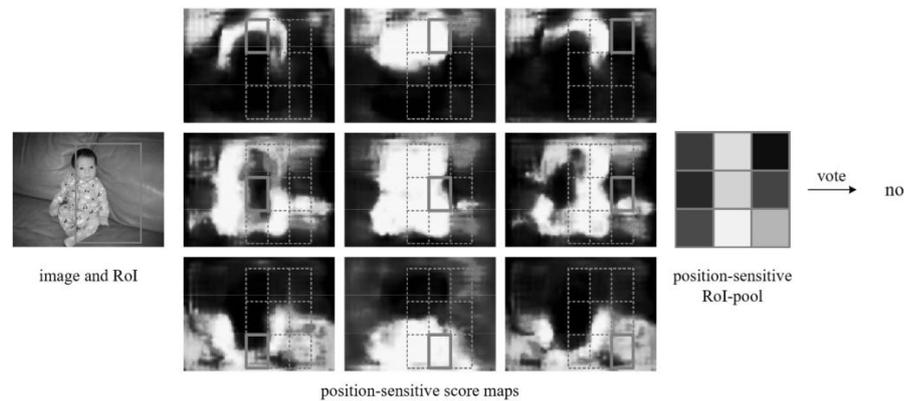


Gambar 3. *Position-sensitive score map* yang telah dibentuk oleh FCN.

Gambar 3 menunjukkan score-map yang dibentuk oleh FCN yang selanjutnya akan diproses oleh R-FCN untuk menentukan *position-sensitive RoI pooling layer*. Dari hasil *train RPN* dan R-FCN selanjutnya akan dilakukan *pooling* dari kumpulan RoI yang telah terbentuk. Mengklasifikasi RoI ke dalam kategori objek dan background. Pada R-FCN semua layer beban dilakukan secara kovolusional dan dihitung pada keseluruhan citra. Hasil akhir R-FCN berupa *position-sensitive RoI pooling layer*.



Gambar 4. Visualisasi dari R-FCN saat RoI tepat *overlap* dengan objek.



Gambar 5. Visualisasi saat RoI tidak tepat *overlap* dengan objek.

Pada Gambar 4 dan Gambar 5 menunjukkan RoI yang terdapat pada gambar tepat pada objek yang diinginkan dengan menghasilkan *position-sensitive RoI pool* dan mendapatkan skor *yes* pada vote. Sedangkan pada Gambar 5 RoI yang dihasilkan tidak tepat pada objek yang diinginkan sehingga menghasilkan *position-sensitive RoI pool* yang memiliki skor vote rendah yaitu *no* [14].

Kamera akan menangkap citra jari di depan pengguna. Kamera menangkap gerakan jari didepan pengguna. Gambar yang diambil berupa video. Akuisisi citra dilakukan dengan mengambil tiap *frame* pada video. Deteksi jari dilakukan pada komputer laptop. Deteksi jari menggunakan metode R-FCN yang berdasar pada region proposal. Hasil dari deteksi berupa *bounding box* pada daerah objek. Dari *bounding box* didapat titik *centroid* untuk dilakukan *plotting*. Melakukan plot pada setiap titik *centroid* yang dibuat. Hasil plot pada tiap *bounding box* akan dijadikan sebagai *track*. Hasil track sebagai keluaran akhir dari sistem ini. Hasil tracking akan diintegrasikan dengan *pointer* pada komputer. Pergerakan *pointer* akan mengikuti pergerakan jari pengguna pada video. Integrasi ini dilakukan untuk menambah fitur pada sistem ini. Pengoperasian komputer dapat dilakukan menggunakan pergerakan jari tanpa perangkat seperti *mouse*. Ujicoba dilakukan untuk memeriksa hasil rancangan. Pada pengujian juga dilakukan penghitungan parameter kinerja. parameter kinerja yang dihitung adalah presisi, *Intersect of Union* (IoU) dan akurasi.

Sistematika data pada penelitian ini adalah *dataset* yang berupa citra ujung jari telunjuk dengan posisi punggung jari menghadap ke pengguna. *Dataset* diambil dengan jarak 30cm, 40cm, dan maksimal peneliti (± 50 cm). *Dataset* yang digunakan dibuat menjadi satu kelas. *Dataset* terdiri dari: data latih, data validasi dan data tes. Dengan rincian rencana *dataset* pada Tabel 1.

Tabel 1. Rancangan Dataset.

Jarak	Data Latih	Data Validasi	Data Uji	Lokasi
30 cm	1800	900	600	Indoor
40 cm	1800	900	600	Indoor
50 cm	1800	900	600	Indoor
30 cm	1800	900	600	Outdoor
40 cm	1800	900	600	Outdoor
50 cm	1800	900	600	Outdoor

Skema yang digunakan adalah dengan memberi banyak *step* pada *training* model sesuai dengan skema yang telah dibuat. Skema *step* diperlukan untuk menentukan besar *loss* dan parameter yang diperlukan. Perancangan sistem ini menggunakan skema *step training* 25K, 50K, dan 75K. Pada perancangan ini juga akan diuji seberapa

akurat model yang dibuat dengan beberapa parameter. Parameter yang akan diuji dari hasil perancangan ini yaitu IoU, presisi, dan akurasi. Parameter penghitungan dilakukan bertujuan untuk menentukan model yang memiliki kinerja baik untuk sistem AR.

Parameter akurasi mengukur kebenaran detektor dalam mengenali bentuk jari (rekognisi) dengan persamaan

$$A = \frac{B}{T} \times 100\%, \quad (1)$$

dimana A adalah besar kebenaran dari rekognisi dalam persentase (akurasi), B adalah data yang terdeteksi yang benar dan T adalah total keseluruhan data. Besar nilai akurasi dalam satuan persentase.

Parameter *Intersect of Union* (IoU) digunakan untuk menghitung ketimpaan atau luas area yang berpotongan antara *bounding box* hasil deteksi dengan *Groundtruth*. IoU dapat dihitung menggunakan persamaan

$$IoU = \frac{B_{GT} \cap B_{ac}}{B_{GT} \cup B_{ac}}, \quad (2)$$

dimana B_{GT} merupakan luas area *Groundtruth*, B_{ac} merupakan luas area untuk *bounding box* aktual dari hasil deteksi. IoU yang mendekati dengan nilai 1 akan dikategorikan baik. IoU yang didapat dari hasil penghitungan akan di-*filter* dengan ambang diatas 0,5 untuk hasil yang terdeteksi.

Presisi merupakan penghitungan jarak antara *centroid Groundtruth* dan *centroid bounding box* actual dengan persamaan

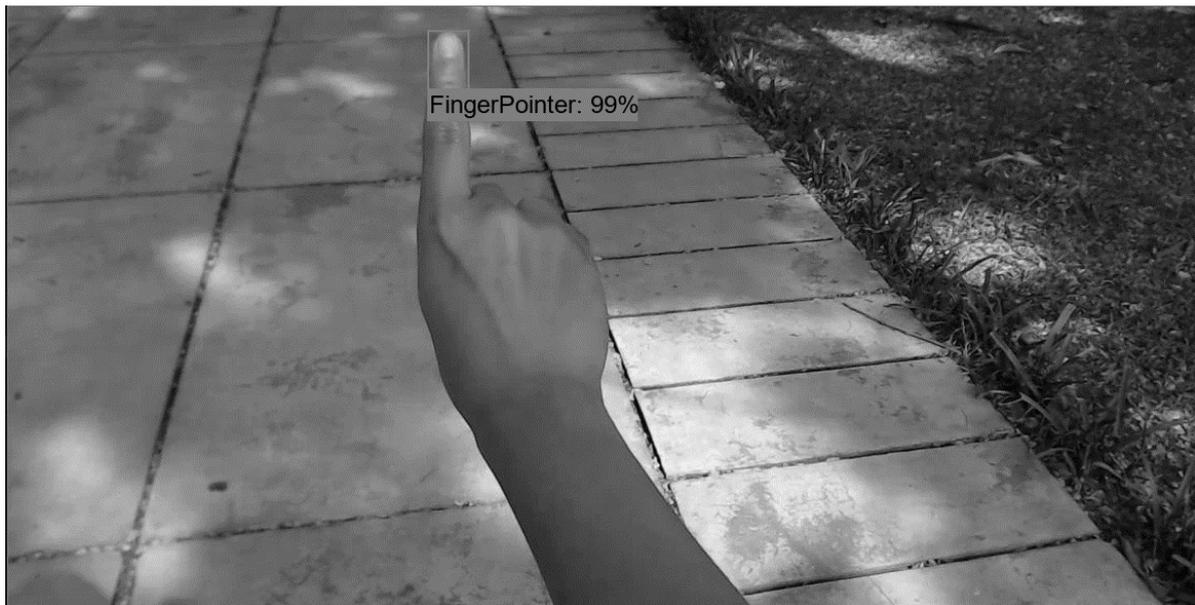
$$P = \sqrt{(C_{x_{GT}} - C_{x_{ac}})^2 + (C_{y_{GT}} - C_{y_{ac}})^2}, \quad (3)$$

dimana P merupakan parameter presisi, $C_{x_{GT}}$ adalah koordinat x untuk *centroid Groundtruth*, $C_{x_{ac}}$ adalah koordinat x untuk *centroid actual* hasil deteksi, $C_{y_{GT}}$ adalah koordinat y untuk *centroid Groundtruth* dan $C_{y_{ac}}$ adalah koordinat y untuk *centroid actual* hasil deteksi.

Parameter uji digunakan untuk menentukan besarnya kesalahan pada saat mendeteksi tangan. Parameter Uji juga digunakan untuk melakukan optimasi dan tuning hingga didapat hasil tracking yang mulus.

III. HASIL DAN PEMBAHASAN

Hasil dari perancangan sistem ini yaitu berupa video dengan *bounding box* hasil deteksi. Dari tiap *bounding box* yang telah di-*generate* lalu dihitung titik centroid. Titik centroid ini akan digunakan sebagai plotting *tracking*. Plotting juga digunakan untuk koordinat *pointer*. Tracking yang telah terbentuk selanjutnya dapat digunakan untuk membuat suatu gerakan atau gestur tangan. Dari gestur tersebut dapat dilakukan operasi dasar pada komputer seperti menggerakkan dan *swipe*. Untuk hasil dari deteksi pada salah satu *frame* dapat dilihat pada Gambar 6.



Gambar 6. Hasil deteksi pada jari.

Gambar 6 menunjukkan *bounding box* hasil dari deteksi menggunakan model yang sesuai dengan skema *step 75K*. *Bounding box* hasil deteksi menunjukkan kelas yang digunakan yaitu *FingerPointer* dan *Confident level* yang didapat sebesar 99%. Koordinat yang didapat dari Gambar 6 yaitu: 406, 446, 24, 82. Koordinat tersebut merupakan koordinat dari *xmin*, *xmax*, *ymin* dan *ymax* secara berurutan.

Hasil deteksi akan dihitung parameter presisi, akurasi dan *IoU* dengan data *groundtruth* yang telah dibuat. Data *groundtruth* dibuat dengan cara memberi *bounding box* secara manual tepat pada objek jari yang diinginkan. Dari *groundtruth* didapat koordinat dari *bounding box* yang telah dibuat. Koordinat dari seluruh video akan diekstrak dan disimpan. Koordinat pada *groundtruth* juga akan diekstrak dan disimpan. Seluruh koordinat yang telah tersimpan akan dibandingkan satu sama lain. Gambar *groundtruth* dapat dilihat pada Gambar 7.



Gambar 7. *Groundtruth* pada salah satu *frame*.

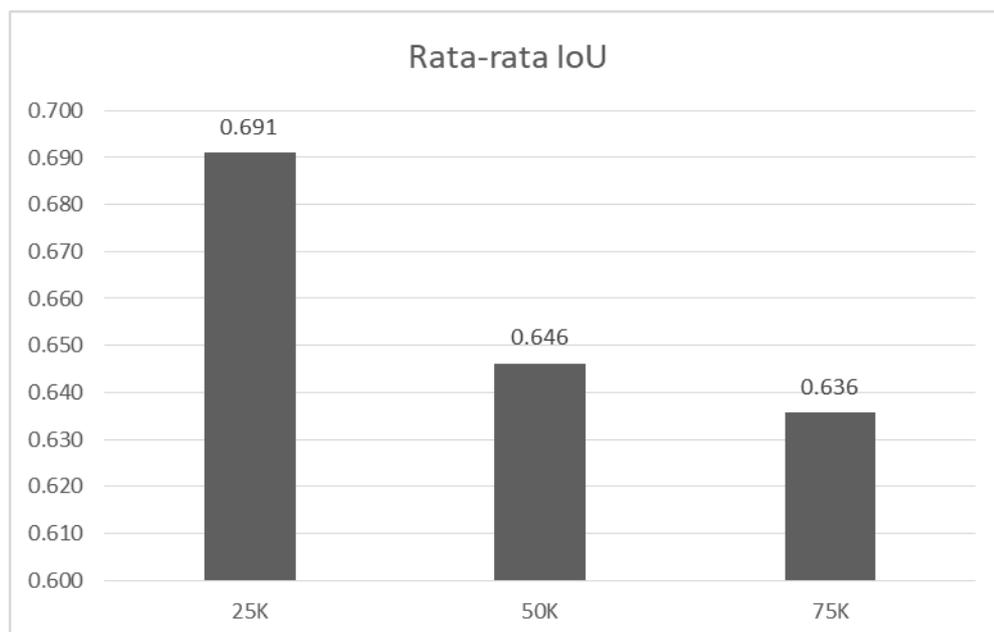
Gambar 7 menunjukkan *Groundtruth* dari citra jari yang diambil. *Groundtruth* menghasilkan koordinat untuk *xmin*, *xmax*, *ymin*, *xmax* secara berurutan yaitu: 408, 444, 27, 78.

Video yang digunakan untuk melakukan tes telah ditentukan dengan skema video tes. Video tes memiliki durasi 10 detik dengan FPS sebesar 30, sehingga didapat 300 *frame* pada tiap video. Skema video tes dapat dilihat pada tabel 2.

Tabel 2. Skema video tes

No.	Jarak	Lokasi	Waktu
1	30cm	Indoor	Pagi Hari
2	40cm	Indoor	Pagi Hari
3	50cm	Indoor	Pagi Hari
4	30cm	Outdoor	Pagi Hari
5	40cm	Outdoor	Pagi Hari
6	50cm	Outdoor	Pagi Hari
7	30cm	Indoor	Sore Hari
8	40cm	Indoor	Sore Hari
9	50cm	Indoor	Sore Hari
10	30cm	Outdoor	Sore Hari
11	40cm	Outdoor	Sore Hari
12	50cm	Outdoor	Sore Hari

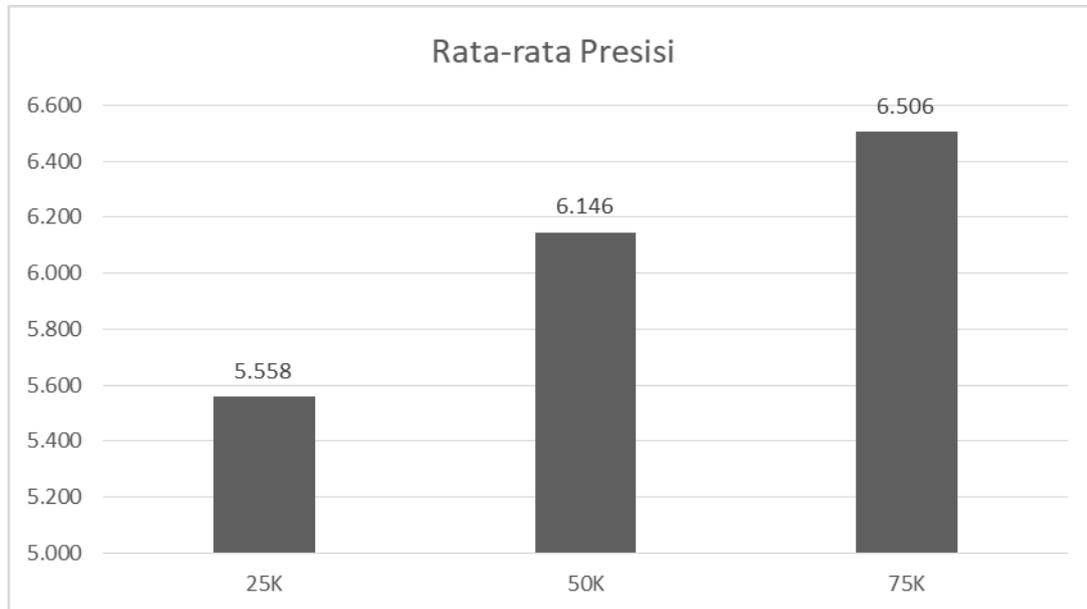
Pada setiap skema *step* akan menghasilkan tiap-tiap model yang telah dibuat. Tiap model akan dilakukan uji menggunakan video yang terdapat pada skema video tes. Hasil uji akan menghasilkan koordinat *bounding box*. Koordinat yang didapat dari hasil ujicoba sebanyak 12×300 tiap skema. Koordinat akan dihitung IoU, Presisi dan Akurasi menggunakan data koordinat *Groundtruth*. Dari hasil tes yang telah dilakukan, didapat IoU dari hasil perhitungan seperti pada Gambar 8.



Gambar 8. Hasil IoU pada tiap model.

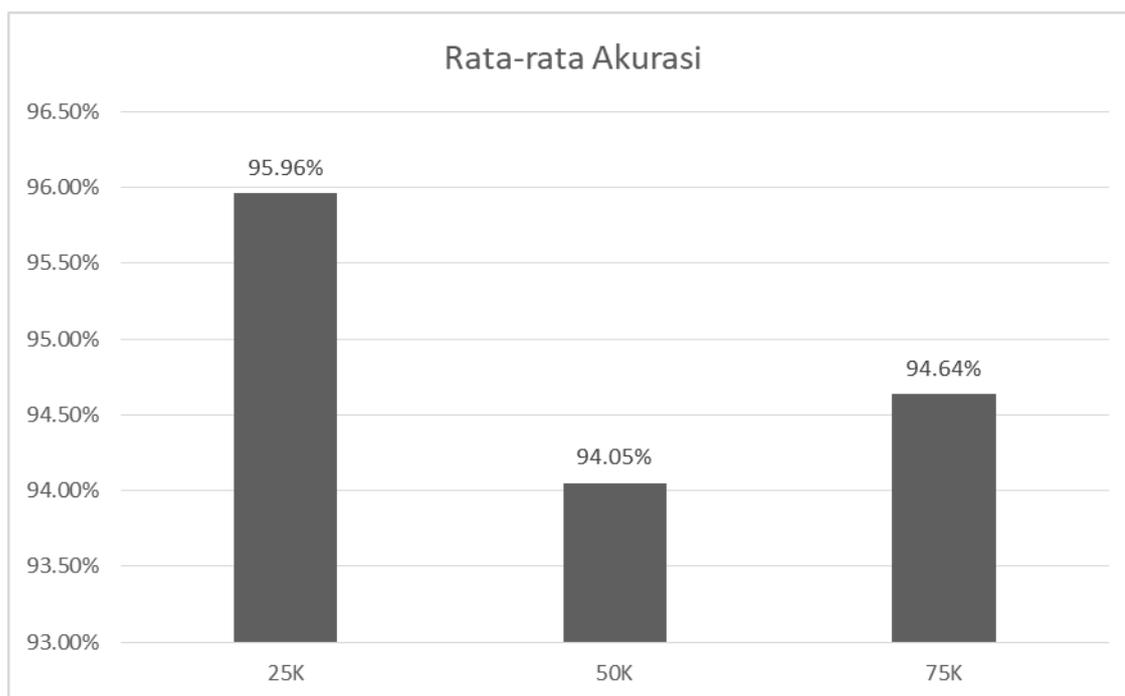
Gambar 8 menunjukkan rata-rata IoU dari hasil tes yang dilakukan pada tiap tiap model. Keterangan pada Gambar 8 menunjukkan besar *step*. Semakin tinggi nilai IoU maka kinerja yang dihasilkan akan semakin baik. Data yang didapat menunjukkan model dengan skema *step* 75K menghasilkan IoU terendah dengan nilai sebesar 0,636.

Sedangkan model dengan skema *step* 25K menghasilkan IoU tertinggi dengan nilai sebesar 0,691, model ini juga menjadi model dengan IoU terbaik. IoU semakin menurun seiring dengan meningkatnya jumlah *step* dikarenakan *loss* pada *training* yang semakin kecil, pada tiap *step training* akan divalidasi menggunakan dataset validasi untuk ditentukan besar *loss*. Semakin kecil *loss* maka besar kemungkinan terjadi *over-fitting*, hal ini menyebabkan objek yang terdeteksi semakin spesifik. Hasil deteksi yang semakin spesifik akan memperkecil luas *bounding box* yang dihasilkan, sehingga nilai IoU semakin kecil. Parameter presisi dihitung pada tiap model yang telah dibuat. Hasil presisi dari tiap model dapat dilihat pada Gambar 9.



Gambar 9. Hasil presisi pada tiap model.

Gambar 9 menunjukkan rata-rata presisi dari hasil tes yang dilakukan pada tiap tiap model. Keterangan pada Gambar 10 menunjukkan besar *step*. Semakin rendah nilai presisi maka kinerja yang dihasilkan akan semakin baik. Data yang didapat menunjukkan model dengan skema *step* 25K menghasilkan presisi terendah dengan nilai sebesar 5,558 , model ini juga menjadi model dengan presisi terbaik. Sedangkan model dengan skema 75K menghasilkan presisi tertinggi dengan nilai 6,506. Rata-rata presisi semakin meningkat seiring meningkatnya jumlah *step*, hal ini dikarenakan semakin meningkatnya jumlah *step* pada *step* 75K objek selain jari semakin banyak terklasifikasi pada kelas *FingerPointer*. Objek selain jari tangan dapat menambah jarak *centroid* yang dihasilkan, hal ini menyebabkan nilai presisi yang semakin tinggi. Parameter akurasi dihitung pada tiap model yang telah dibuat. Hasil presisi dari tiap model dapat dilihat pada Gambar 10.



Gambar 10. Hasil akurasi pada tiap model.

Gambar 10 menunjukkan rata-rata presisi dari hasil tes yang dilakukan pada tiap tiap model. Keterangan pada Gambar 10 menunjukkan besar *step*. Semakin tinggi nilai akurasi maka kinerja yang dihasilkan akan semakin baik. Data yang didapat menunjukkan model dengan *step* 50K menghasilkan akurasi terendah dengan nilai sebesar 94,05%. Sedangkan model dengan *step* 25K menghasilkan akurasi tertinggi dengan nilai 95,96% , model ini juga menjadi model dengan presisi terbaik. Akurasi cenderung stabil pada ambang 0,9 hingga 1, nilai akurasi yang stabil dikarenakan skema konfigurasi parameter pada *extractor* dan *classifier* tepat sesuai dengan fitur yang ada pada citra.

Untuk mendapatkan kinerja pergerakan AR yang mulus maka diperlukan parameter yang diperhatikan sesuai urutan prioritas mulai dari presisi, akurasi, dan IoU. IoU mempengaruhi posisi *centroid* yang juga akan berpengaruh pada presisi deteksi. IoU yang mendekati nilai 1 (satu) memberikan jarak *centroid* yang semakin kecil. Presisi diperlukan agar perpindahan *pointer* yang dibentuk tidak terlalu besar, perpindahan yang besar akan menimbulkan efek bergetar pada kursor dalam masa deteksi. Akurasi juga menjadi parameter penting untuk meningkatkan kinerja AR, akurasi yang tinggi menciptakan pergerakan AR yang mulus dalam kondisi pergerakan cepat dan posisi jari yang tidak selalu tepat (*miss-position*).

IV. KESIMPULAN

Dari hasil dan pembahasan yang dilakukan maka dapat disimpulkan model dengan skema *step* 25K (Model A) menjadi model yang mendapat kinerja terbaik dalam sistem AR. Model ini memiliki nilai IoU sebesar 0,691, presisi sebesar 5,556, dan akurasi sebesar 95,96%. Model ini dapat digunakan untuk menghasilkan sistem AR dengan pergerakan yang mulus. Model dengan skema *Step* 50K (Model B) memberikan kinerja yang cukup baik untuk AR dengan nilai IoU sebesar 0,646, presisi sebesar 6,146, dan akurasi 94,05%. Hasil pada tiap model menunjukkan nilai yang semakin memburuk dikarenakan adanya *over-fitting* dan banyaknya citra non-jari yang terklasifikasi ke dalam kelas. Model A memiliki getaran pada *pointer* yang lebih kecil dibanding dengan model B. Getaran yang terjadi pada *pointer* dapat dikurangi dengan menambahkan algoritma untuk memberikan setengah jarak antara satu titik *centroid* dan titik setelahnya. Pergerakan *pointer* yang mulus dapat menjadikan AR lebih interaktif. Manfaat dari penelitian ini untuk memberikan model yang cocok untuk sistem *projector-camera-based AR* serta dapat mengoperasikan *notebook/laptop* dalam sehari-hari tanpa harus berinteraksi secara langsung. Pada sistem ini pengguna dapat mengoperasikan *notebook/laptop* dimana saja menggunakan benda padat sebagai layar.

V. DAFTAR PUSTAKA

- [1] Negrotti M. 2010. *Virtual Reality*. The Reality of the Artificial Edisi pertama Springer-Verlag Berlin Heidelberg. Jerman.
- [2] Aniket P. 2015. *Augmented Reality*. Intenational Journal of Research in Advance Engineering (IJRAE).
- [3] Lim C., Choi J., Park J. I. 2015. *Interactive Augmented Reality System Using Projector-Camera System and Smartphone*. EEE International Symposium On Consumer Electronics (ISCE).
- [4] Madni M. S., Rathod R. N. 2016. *Color Segementation for Sixth Sense Device*. Bonfring International Journal of Research in Communication Engineering.
- [5] Gupta J., Bankar A., Warankar M., Shelke A. 2014. *Augmented Reality by using Hand Gesture*. International Journal of Engineering Reesearch & Technology (IJERT).
- [6] Choi J., Seo B.-K., Park J.-I. 2009. *Robust Hand Detection for Augmented Reality Interface*.
- [7] Lambrecht J., Walzel H., Kruger J. 2013. *Robust finger gesture recognition on handheld devices for spatial programming of industrial robots*. Proceedings IEEE International Workshop on Robot and Human Interactive Communication.
- [8] Khan S., Rahmani H., Shah S. A. A., Bennamoun M. 2018. *A Guide to Convolutional Neural Networks for Computer Vision*. Synthesis Lectures on Computer Vision.
- [9] Wibowo S. A., Lee H., Kim E. K., Kim S.. 2017. *Convolutional shallow features for performance improvement of histogram of oriented gradients in visual object tracking*. Mathematical Problems in Engineering.
- [10] Darmadi R. 2018. *Mengenal Convolutional Layer Dan Pooling Layer*. Diambil pada tanggal 6 November 2019 dari <https://medium.com/nodeflux/mengenal-convolutional-layer-dan-poolinglayer-3c6f5c393ab2>
- [11] Wibowo S. A., Lee H., Kim E. K., Kim S.. 2018. *Collaborative learning based on convolutional features and correlation filter for visual tracking*. International Journal of Control, Automation and Systems.
- [12] Wibowo S. A., Lee H., Kim E. K., Kim S.. 2017. *Visual tracking based on complementary learners with distractor handling*. Mathematical Problems in Engineering.
- [13] Darmadi R.. 2018. *Mengenal Convolutional Layer Dan Pooling Layer* diambil pada tanggal 7 November 2019 dari <https://medium.com/nodeflux/mengenal-convolutional-layer-dan-poolinglayer-3c6f5c393ab2>
- [14] Dai J., Li Y., He K., Sun J. 2016. *R-FCN : Object Detection via Region-based Fully Convolutional Network*. Advances In Neural Information Systems 29 (NIPS 2016).
- [15] Ren S., He K., Girshick R., Sun J. 2017. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. IEEE Trans. Pattern Anal. Mach.