Aplikasi Findgo-ITATS Berbasis Android Dengan Algoritma SURF Untuk Menampilkan Informasi Lokasi Di ITATS

Rinci Kembang Hapsari[1], Nur Sulaiman[2], Luky Agus Hermanto[3]
[1,2,3]Teknik Informatika - ITATS, Jl. Arief Rahman Hakim 100 Surabaya
Email: [1]rincikembang@itats.ac.id, [2]iman.siiip@gmail.com, [3]luky@gmail.com

Abstract. *Institut Teknologi Adhi Tama Surabaya (ITATS) is an institute that has relatively wide territory and complicated building arrangement for outsiders especially related to identification of buildings that they want to visit. To overcome this problem, an android based application that can be used to gain information related to those buildings, locations of places or important places in real-time is required. Augmented Reality (AR) is the appropriate technology to display environment and locations information at ITATS in real-time. The implementation of Augmented Reality technology on android based smartphones using Speeded Up Robust Features (SURF) can identify pictures continuously and has proper identification speed. Speeded Up Robust Features (SURF) is an algorithm that has been commonly applied in correspondence matching because it is faster than Scale Invariant Feature Transform (SIFT) and has appropriate and accurate performance maintenance. In designing this application, there are three main stages that should be considered, namely: initialization, tracking marker, and object rendering. Initialization is the stage where images that becomes the database is preliminary processed with Speeded Up Robust Features (SURF) algorithm and the preparation of the displayed information on the users' smartphones. The second is tracking marker, smartphone camera takes pictures continuously while processing every inputted image applying Speeded Up Robust Features (SURF) and conducting matching process of images in the database. The final stage, after a match is found, this application displays the text information which corresponds with the matching result. The reliability of this system in recognizing locations at ITATS is 81.66% and average time required is 2.333 seconds.*

Keywords: *Augmented Reality, Speeded Up Robust Features, SURF, real-time object recognition, android, ITATS.*

Abstrak. Institut Teknologi Adhi Tama Surabaya (ITATS) adalah perguruan tinggi yang memiliki luas area yang cukup besar dan gedung-gedung yang cukup membingungkan bagi orang awam, khususnya dalam hal pengenalan lokasi-lokasi atau tempat-tempat yang ingin mereka kunjungi. Untuk mengatasi masalah tersebut, dibutuhkan aplikasi berbasis *android* yang dapat digunakan untuk mendapatkan informasi terkait gedung-gedung, lokasi-lokasi atau tempat-tempat penting yang sering dikunjungi secara *real-time*. Teknologi *Augmented Reality* (AR), merupakan teknologi yang tepat untuk menampilkan lingkungan nyata dan informasi lokasi di ITATS secara *real-time*. Implementasi teknologi *Augmented Reality* (AR) pada *smartphone* bersistem operasi *android* dengan menggunakan algoritma *SURF (Speeded Up Robust Features)* dapat mengenali gambar secara terus menerus dengan kecepatan pengenalan yang baik. *SURF (Speeded Up Robust Features)* merupakan algoritma yang telah banyak digunakan untuk memecahkan masalah pencocokan korespondensi karena lebih cepat daripada *SIFT (Scale Invariant Feature Transform)* dan dengan cermat mempertahankan kinerja yang cocok. Dalam pembuatan, ada 3 bagian utama yang harus diperhatikan, yaitu inisialisasi, *tracking marker* dan *rendering* objek. Pada tahap inisialisasi, citra yang dijadikan basis data pengenalan akan diolah terlebih dahulu melalui algoritma *SURF (Speeded Up Robust Features)* dan menyiapkan informasi yang akan ditampilkan ke *smartphone* pengguna. Di tahap *tracking marker*, kamera *smartphone* akan mengambil gambar secara terus menerus

sambil mengola setiap masukan gambar menggunakan algoritma *SURF (Speeded Up Robust Features)* dan melakukan proses pencocokan dengan citra yang ada di dalam basis data. Tahap terakhir adalah setelah ditemukan kecocokan, aplikasi akan menampilkan teks informasi yang sesuai dengan hasil pencocokan. Keberhasilan sistem ini dalam mengenali lokasi yang berada di ITATS sebesar 81.66% dan rata-rata waktu yang dibutuhkan sebesar 2.333 detik.

Kata kunci:*Augmented Reality*, *SURF*, *Speeded Up Robust Features*, pengenalan objek *real-time*, *android*, ITATS.

1. Pendahuluan

Perkembangan sistem dan teknologi informasi sekarang ini sangatlah pesat khususnya dalam hal pengendalian tampilan yang dapat dimanfaatkan untuk informasi dan pengetahuan dalam bentuk visual. Penggambaran objek visual dalam bentuk Augmented Reality (AR) adalah penggabungan antara objek *virtual* dengan objek nyata, yaitu lingkungan yang memasukkan objek *virtual* ke dalam lingkungan nyata.

Smartphone bersistem operasi android dipilih karena penggunaannya yang mudah untuk dibawa kemana saja. Selain itu, pengguna smartphone ini sudah banyak digunakan di kalangan masyarakat, sehingga harapannya masyarakat tidak terlalu sulit saat menggunakan aplikasi FINDGO-ITATS yang menjadi tugas akhir penulis. Tema ini diambil karena mengingat pentingnya informasi lokasi saat ada pengunjung yang baru pertama kali berada di ITATS.

Metode yang digunakan untuk menampilkan sebuah informasi adalah Speeded Up Robust Features (SURF), yaitu algoritma yang memungkinkan untuk mengidentifikasi objek yang selanjutnya diolah agar mendapatkan informasi yang dibutuhkan oleh sistem. Algoritma ini digunakan untuk pengenalan gambar terus menerus dan pelacakan dalam video. Pendeskripsi fitur SURF beroperasi dengan mengurangi ruang pencarian dari kemungkinan titik penting. SURF menambahkan banyak fitur untuk meningkatkan kecepatan dalam setiap langkah. Kualitas pengenalan obyek ini sangat penting untuk kebutuhan pelacakan real-time, dan algoritma pelacakan tidak boleh mengganggu kinerja aplikasi.

2. Tinjauan Pustaka

2.1 *Augmented Reality (AR)*

Teknologi *Augmented Reality* (AR), merupakan teknologi yang menggabungkan benda maya dua dimensi ataupun tiga dimensi ke dalam sebuah lingkungan nyata hasil proyeksi benda-benda maya secara *real-time* atau singkatnya merupakan teknologi yang menggabungkan dunia nyata dengan dunia *virtual* (Slamet Budiyatno, 2012). Tujuan utama dari A*ugmented Reality* adalah untuk menciptakan sensasi objek virtual yang hadir di dunia nyata. Untuk mencapai efek ini maka dibutuhkan penggabungan antara *virtual reality* (VR) dengan dunia nyata.

Augmented reality akan sangat efektif jika elemen virtual ditambahkan secara realtime. Oleh karena hal tersebut, *augmented reality* umumnya menambahkan objek 2D atau 3D pada sebuah video digital secara *real-time*. Objek virtual yang ditambahkan kedalam adegan diketahui sebagai visual AR. Menurut definisi, element AR tidak terlihat dengan mata telanjang sehingga visual AR bergantung pada semacam layar seperti *canvas* pada *HTML5* (Cawood & Mark, 2008).

Ada tiga karakteristik yang menyatakan suatu teknologi menerapkan konsep *Augmented Reality* (Affan dan Eddo, 2012), yaitu: 1) Mampu mengkombinasikan dunia nyata dan dunia maya. 2) Mampu memberikan informasi secara interaktif dan *real-time*. 3)Mampu menampilkan dalam bentuk 3D (tiga dimensi).

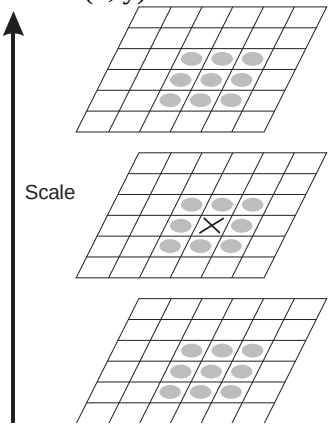
2.2 Algoritma *SURF (Speeded Up Robust Features)*

Sebuah pencocokan korespondensi adalah salah satu tugas penting dalam visi komputer, dan itu tidak mudah untuk menemukan titik-titik yang sesuai pada variabel lingkungan di mana

skala, rotasi, sudut pandang dan pencahayaan yang berubah. Algoritma *SURF (Speeded Up Robust Features)* telah banyak digunakan untuk memecahkan masalah pencocokan korespondensi karena lebih cepat daripada *SIFT (Scale Invariant Feature Transform)* dengan cermat mempertahankan kinerja yang cocok (Hongbo Li, Ming Qi And Yu Wu, 2012).

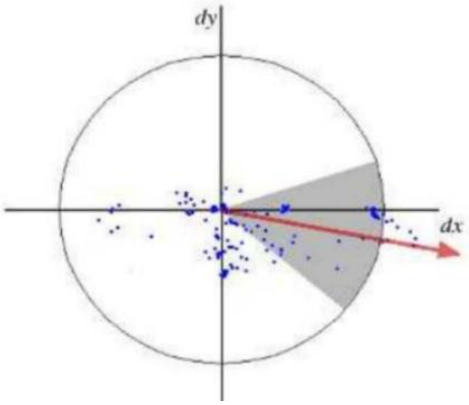
Algoritma *SURF (Speeded Up Robust Features)* digunakan untuk pengenalan gambar terus menerus dan pelacakan dalam video. Pendeskripsi fitur - *SURF* beroperasi dengan mengurangi ruang pencarian dari kemungkinan titik penting (*keypoints*). Gambar 1 memperlihatkan proses *nonmaximum suppression* untuk mendapatkan *keypoints*.

SURF menambahkan banyak fitur untuk meningkatkan kecepatan dalam setiap langkah. Kualitas pengenalan obyek penting untuk kebutuhan pelacakan *real-time*, dan algoritma pelacakan tidak boleh mengganggu kinerja.*SURF* bagus untuk menangani gambar dengan kualitas kabur dan rotasi pada gambar. Algoritma ini sangat cepat karena penggunaan gambar yang tidak terpisahkan di mana nilai pixel (x, y) adalah jumlah dari semua nilai dalam persegi panjang yang didefinisikan oleh asal dan (x, y).



Gambar 1. Proses nonmaximum suppression (Ken Ditha, 2010)

SURF didasarkan pada jumlah respon *Haar wavelet 2D* dan membuat efisiensi penggunaan gambar yang tidak terpisahkan. Fitur *SURF* melacak objek dengan pencocokan titik penting.*SURF* terus menerus mengekstraksi fitur untuk pengenalan. Untuk ekstraksi fitur *SURF* menggunakan jumlah respon *Haar wavelet* sekitar titik penting.Keuntungan utama dari transformasi *Haar wavelet* adalah kecepatan perhitungan.Hasil penyebaran titik – titik penting terlihat pada Gambar 2.



Gambar 2. Persebaran titik – titik penting (Ken Ditha, 2010)

Selain itu, *SURF* menggunakan integral gambar untuk meningkatkan kecepatan proses. Hal tersebut bertujuan untuk mengurangi waktu komputasi pada ekstraksi fitur dan pada proses *matching* fitur, yang juga telah terbukti meningkatkan *robust*.

Proses awal yang akan dilakukan adalah mencocokan orientasi yang dihasilkan berdasarkan informasi dari daerah yang berbentuk lingkaran disekitar piksel yang menjadi *keypoint*. Kemudian membuat daerah berbentuk kotak pada orintasi yang terpilih dan mengekstrak deskriptor *SURF* dari daerah tersebut. Pada proses selanjutnya, proses *matching* fitur antara dua gambar dilakukan.

Asosiasi ini sangat sulit ketika objek bergerak cepat yang relatif terhadap *frame rate*. Ketika objek dilacak mengubah orientasi dari waktu ke waktu yang meningkatkan kompleksitas.Untuk situasi ini sistem pelacakan video yang biasanya menggunakan model gerak.Model gerak menggambarkan bagaimana gambaran target yang mungkin berubah dan kemungkinan gerakan yang berbeda dari objek.

SURF menggunakan pendekatan bilangan bulat. Untuk mendeteksi fitur matriks *Hessian (H)* telah terpasang, di mana Lxx adalah konvolusi dari turunan kedua dari *Gaussian* dengan gambar pada titik. Matriks *Hessian* direpresentasikan pada rumus(1).

$$H = \begin{bmatrix} Lxx & Lxy \\ Lxy & Lyy \end{bmatrix} \quad (1)$$

SURF menggunakan skala yang berbeda dari *Gaussian*, sedangkan skala gambar selalu berubah. *SURF* adalah rotasi invarian, rotasi ditangani dengan mencari arah fitur dan rotasi area sampling untuk menyelaraskan dengan sudut tersebut. Bangun *region* persegi berpusat di sekitar titik fitur. Ukuran area adalah 20s x 20s yang diambil di sekitar titik penting yang terdeteksi, di mana s adalah ukuran. Setelah lingkungan rotasi diperoleh, dibagi menjadi 16 sub persegi. Setiap sub persegi dibagi lagi menjadi 4 persegi (J. Jasmine Anitha and S.M. Deepa, 2014).

3. Metode Penelitian

3.1 Gambaran Umum Sistem

Aplikasi yang dibangun untuk memenuhi kebutuhan akan informasi adalah aplikasi *android* yang bernama *"FindGo-ITATS"*. Aplikasi ini diharapkan bisa membantu pengguna untuk mempermudah dalam mencari tahu informasi lokasi yang ada di ITATS. Dengan sistem pengenalan objek 3D tersebut, pengguna dapat mengenali lokasi yang ada di depannya secara *real-time*menggunakan kamera yang terdapat pada *smartphone* hanya dengan mengarahkan ke area lokasi di depannya melalui aplikasi *FindGo-ITATS.*

Perancangan sistem pengenalan objek 3D meliputi dua aspek yang berkaitan satu dengan lainnya, yaitu perancangan sistem aplikasi *android* dan pemanfaatan algoritma *SURF*. Pada perancangan sistem pengenalan objek 3D, sistem akan melakukan proses pengenalan objek yang melibatkan algoritma *SURF* yang tertanam pada *smartphone*. Hasil pengenalan objek tersebut berupa teks informasi terkait objek yang didapatkan kamera. Bentuk penyajian tersebut ditampilkan dalam bentuk *Augmented Reality* sehingga tampilan lebih terlihat interaktif dan *real-time*.

3.2 Algoritma Program

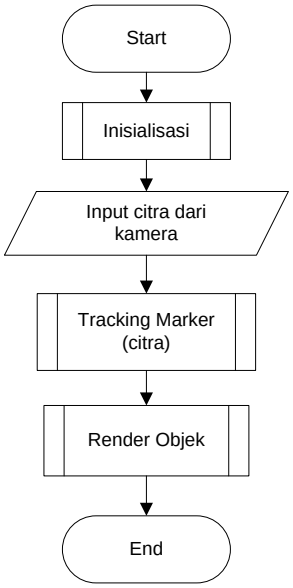
Algoritma dalam kasus ini merupakan penjelasan bagaimana proses *augmented reality* dari awal inisialisasi, *tracking marker*, sampai dengan proses *rendering* informasi dengan algoritma *SURF* untuk penerapan teknologi *markeless augmented reality* yang dapat berjalan di *smartphone* bersistem operasi *android*.

Perancangan aplikasi ini, menggabungkan objek *virtual* dengan objek nyata, dalam studi kasus ini yang menjadi objek *virtual* adalah teks informasi mengenai bangunan dan objek nyatanya berupa bangunan yang diterima kamera dengan pola masing-masing bangunan (*markerless*).Sistem *Augmented Reality* ini berjalan di *smartphone android.*

Secara umum, proses sistem *augmented reality* dapat dilihat pada diagram alir pada Gambar 3. Di dalam diagram alir tersebut terdapat 3 proses penting yang mewakili keseluruhan sistem yang akan dibangun untuk *smartphone* bersistem operasi *android*. Aplikasi akan melakukan proses inisialisasi terlebih dahulu, kemudian melakukan *tracking marker* secara

terus menerus. *Marker*akan dideteksi dari setiap masukan gambar kamera *android*, jika *marker* cocok atau dikenali oleh aplikasi maka teks berupa informasi mengenai *marker* tersebut akan di-*render* di atas *marker* yang terdeteksi. Perancangan aplikasi *FindGo ITATS* ini harus melibatkan 3 tahap penting berikut ini: Inisialisasi

1. *Tracking marker*
2. *Rendering* Objek



Gambar 3. Diagram alir sistem *Augmented Reality*

3.3 Inisialisasi

Pada tahap ini, ditentukan *marker* yang digunakan (basis data gambar yang akan dikenali) dan teks informasi yang mewakili informasi dari *marker*. Aplikasi melakukan persiapan awal untuk mengatur tampilan di *smartphone*, seperti pengaturan *layout* yang digunakan. Aplikasi juga akan memulai proses pembuatan deskriptor terhadap data citra yang telah disiapkan untuk data pengenalan. Data citra yang ada akan diolah di dalam proses inisialisasi ini sebanyak satu kali saja, yaitu disaat aplikasi pertama kali diinstall di dalam *smartphone* (saat basis data deskriptor belum terbuat).

Di dalam proses pembuatan deskriptor ini, aplikasi akan mengambil citra dari *resource* yang ada di dalam sistem *android*. Citra yang didapatkan akan diproses melalui algoritma *Canny* terlebih dahulu. Setelah mendapatkan garis tepi dari proses *Canny*, sistem akan mencari fitur dari citra tersebut. Setelah didapatkan fiturnya, proses selanjutnya adalah mencari deskriptor dari citra tersebut. Deskriptor inilah yang dibutuhkan untuk proses pencocokan citra basis data dengan citra yang ditangkap oleh kamera di dalam aplikasi *FindGo-ITATS*. Semua proses ini akan dilakukan sampai semua citra yang ada di dalam *resource android* telah diolah semua.

3.4 *Tracking marker*

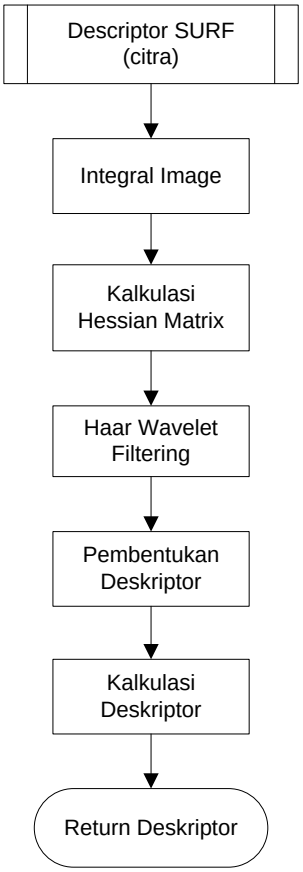
Pendeteksian gambar dan perhitungan posisi gambar dilakukan dengan menggunakan algoritma *SURF*, dimana aplikasi memanfaatkan kamera dari *smartphone* untuk pengambilan gambar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek atau model atau gambar ke dalam posisi gambar atau *markerless*. Ada 4 tahapan, dalam proses kerja *Tracking marker*.

1. Mengambil gambar dari kamera *android*
2. Ekstraksi fitur menggunakan deskriptor *SURF*

3. *Corner Detection*
4. Pencocokan dan Pengenalan

Penghalusan citra dengan *filter Gaussian* diperlukan untuk menghilangkan *noise* pada citra dan meningkatkan kualitas detail citra. Setelah didapatkan citra dengan kualitas detail terbaik, dihitunglah *Gradient Magnitude* untuk mendeteksi keberadaan *edge* pada suatu titik dalam citra. Proses selanjutnya adalah menghitung arah *edge* dari hasil deteksi keberadaan *edge* diproses sebelumnya. Perhitungan arah *edge* ini, nantinya digunakan untuk proses *nonmaximum suppression*.

Pada proses *nonmaximum suppression*, pembuangan potensi gradien di suatu piksel dilakukan dari kandidat *edge*, jika piksel tersebut bukan merupakan maksimal lokal pada arah *edge* di posisi piksel tersebut. Hasilnya adalah citra yang berisi kandidat *edge* serta intensitas dari kekuatan *edge* di posisi piksel tersebut.



Gambar 4. Ekstraksi Deskriptor *SURF*

Langkah terakhir adalah melakukan klasifikasi dengan dua buah nilai *High-threshold* dan *Low-Threshold*. Suatu piksel disahkan sebagai piksel *edge* jika nilainya lebih besar atau sama dengan *High-Threshold* (*thresholding* umum) atau (di sini kaidah tambahannya) jika piksel tersebut memiliki intensitas kekuatan *edge* yang lebih besar dari *Low-Threshold* dan terhubung dengan piksel yang nilainya lebih besar dari *High-Threshold*.

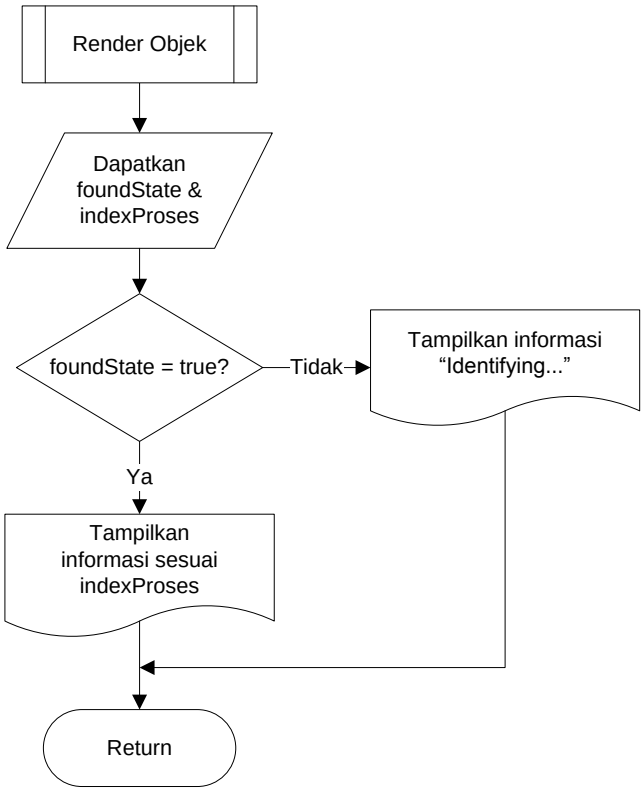
Setelah mendapatkan hasil akhir di atas, langkah berikutnya adalah menemukan area yang berdampingan dalam gambar yang di *threshold*. Area yang berdampingan diberi tanda sebagai persegi (*marker outline*) menggunakan algoritma *SURF (Speeded Up Robust Feature)*. Proses awal yaitu, menentukan *keypoint*. Ketika *threshold* dinaikan, jumlah *keypoint* yang terdeteksi lebih kecil begitu juga dengan sebaliknya.Titik-titik dari sebuah gambar yang nilainya tetap ketika mengalami perubahan skala, rotasi, *blurring*, pencahayaan, dan juga perubahan

bentuk. Perubahan bentuk ini bisa terjadi karena bentuk gambar *query* yang tidak utuh atau tidak sempurna seperti gambar yang ada di dalam basis data gambar tersebut.

Proses selanjutnya adalah *nonmaximum suppression*. Proses ini dilakukan untuk mencari sekumpulan calon *keypoint* dengan membandingkan tiap-tiap piksel gambar pada *scale space* dengan 26 tetangga. 26 tetangga piksel tersebut terdiri dari 8 titik di skala asli dan 9 titik di tiap-tiap skala di atas dan di bawahnya. Proses inilah yang menghasilkan *keypoint* dari suatu gambar.

Proses mencari lokasi *keypoint* dilakukan setelah didapatkan hasil dari *nonmaximum suppression*. Proses ini mencari lokasi *keypoint* menggunakan interpolasi data yang dekat dengan *keypoint* hasil proses sebelumnya. Ini dilakukan dengan mencocokkan *quadratic 3D* yang diajukan oleh *Brown*H( ) adalah determinan *Hessian*. Lokasi ekstrim yang diinterpolasi, x( ), ditemukan dengan mencari turunan dari fungsi di atas dan diberi nilai nol.

Kemudian, respon *wavelet* dx dan dy dijumlahkan untuk setiap *sub-region*. Hal ini akan memberikan informasi tentang polar dari perubahan intensitas, dan juga akan menghasilkan jumlah nilai absolut dari respon |dx| dan |dy|. Masing-masing *sub-region* mempunyai 4 dimensi deskriptor vektor v, yaitu dx, dy, |dx|, dan |dy|. Untuk 4 x 4 *sub-region*, maka panjang vektor deskriptornya berjumlah 64.



Gambar 5. Flowchart menampilkan informasi

Proses pembentukan deskriptor dilakukan dengan membuat daerah kotak persegi di sekitar *keypoint*, dimana *keypoint* sebagai pusat dari daerah kotak persegi tersebut dan orientasinya di sekitar orientasi yang ditentukan. Sehingga yang akan dihasilkan hanyalah area yang memiliki bentuk segi empat dan pola-pola gambar yang sudah ditandai. Pola – pola tersebut dibentuk untuk menghasilkan deskriptor yang siap diolah untuk keperluan pencocokan citra.Deskriptor tersebut mengandung informasi terkait koordinat *keypoint* dan jumlah *keypoint* yang dihasilkan dari hasil komputasi.

Tahap berikutnya adalah pencocokan pola, yaitu setelah area persegi dan pola-pola gambar ditandai, gambar yang berada di dalam persegi dianalisa dan dibandingkan polanya dengan sekumpulan pola yang telah ditentukan sebelumnya.

3.5 *Rendering* Objek

Setelah didapatkan hasil dari proses pencarian menggunakan algoritma *SURF* (cocok atau tidak cocok), maka proses selanjutnya adalah menampilkan hasilnya. Citra yang ditangkap oleh kamera dikatakan cocok ketika variabel *foundState* yang diproses saat proses *tracking marker* memiliki status *TRUE*. Setelah diketahui bahwa status ditemukannya sama dengan *TRUE*, maka informasi mengenai objek tersebut akan ditampilkan berdasarkan variabel *indexProses*. Informasi yang ditampilkan berupa teks dan akan diperbaharui secara *realtime* setiap *frame* kamera yang sudah dicocokkan dengan basis data gambar. Gambar proses *render*objek ditunjukkan pada Gambar 5.

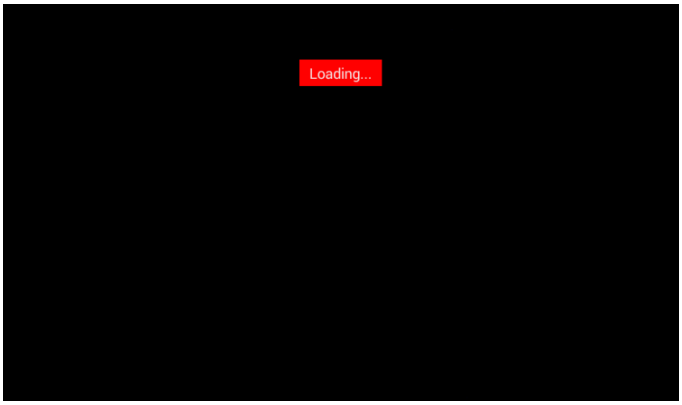
4. Hasil dan Pembahasan

4.1 Implementasi

Implementasi dilakukan dengan menggunakan *smartphone android* yang telah diinstall *opencv manager* versi 2.20 untuk pustaka dari program yang dibuat dan tersedianya kamera belakang.Konfigurasinya hanya menginstall *apk* dari aplikasi *android* yang telah dibuat.

Program akan melakukan proses pengambilan data di dalam basis data untuk mendapatkan semua deskriptor dari citra yang dijadikan acuan pengenalan objek. Proses pengambilan data ini dilakukan setiap aplikasi memulai menjalankan atau menyalakan kamera di *smartphone*. Gambar 6 menunjukkan proses pengambilan data (deskriptor citra) untuk proses pengenalan objek.

Setelah semua deskriptor didapatkan, program akan memulai proses *identifying* (mengenali). Di dalam proses ini, aplikasi akan menjalankan *thread* untuk mengola data citra yang didapatkan dari kamera *smartphone*. Dengan adanya *thread*, dapat membuat program lebih efisien dalam hal kecepatan maupun penggunaan sumber daya, karena program dapat membagi proses dalam aplikasi pada waktu yang (hampir) sama. *Thread* umumnya digunakan untuk pemrograman *multitasking*, *networking*, yang melibatkan pengaksesan ke sumber daya secara kongruen.



Gambar 6. Proses Pengambilan Deskriptor Citra

Di setiap *thread* yang dijalankan, program akan melakukan proses *Canny*, pencarian *keypoint*, mengekstrak deskriptor menggunakan algoritma *SURF*, pencocokan citra masukan dengan basis data sampai *rendering* informasi sesuai dengan hasil yang didapatkan saat melakukan pencocokan citra. Semua proses di dalam *thread*akan dimulai di setiap masukan *frame* yang didapatkan oleh kamera.Saat aplikasi menemukan citra masukan yang cocok dengan basis data yang ada di dalam aplikasi, informasi akan ditampilkan secara *real-time*. Lihat contoh tampilan yang ditunjukkan oleh aplikasi di Gambar 7.

Gambar 7. Aplikasi Mendapatkan Citra Masukan Yang Cocok

4.2 Analisis Program

Analisis program dilakukan untuk menunjukkan presentase keberhasilan mengenali objek dan rata-rata waktu yang dibutuhkan untuk mengenali objek yang ada. Dalam menguji program, terdapat kriteria yang akan diujikan untuk mendapatkan hasil pengujian yang optimal. Adapun kriteria pengujian yang dilakukan beserta hasil presentase akurasi pengenalan dalam pengujian di setiap kriteria sebagai berikut:

1. Pengujian dengan 15 lokasi yang berbeda bentuk objeknya
2. Pengujian dengan 15 bentuk objek yang hampir sama di lokasi berbeda
3. Pengujian dengan 15 ruangan yang identic
4. Pengujian dengan 15 objek pencerminan

Dari ke empat kriteria pengujian di atas didapatkan 4 nilai presentase akurasi pengujian, yaitu 1.73.33%, 2.80%, 3.100% dan 4.73.33%. Perbedaan dari hasil pengujian dipengaruhi oleh kondisi objek yang dikenali, meliputi fitur yang berhasil didapatkan aplikasi dan juga adanya kemungkinan titik poin dan koordinat pengenalan yang sama.

Dari presentase akurasi pengenalan objek berdasarkan kriteria di atas, didapatkan pula rata-rata presentase tingkat keakurasian pengenalan objek menggunakan aplikasi *FindGo-ITATS*.Dari pengujian menggunakan 60 objek dan 4 kriteria yang disebutkan di atas, didapatkan rata-rata akurasi pengenalan yang baik, yaitu 81.66%.Hal ini dapat membuktikan bahwa aplikasi *FindGo-ITATS* dapat digunakan untuk mengenali sebagian besar dari objek yang sudah didaftarkan di dalam aplikasi tersebut.

Dari hasil uji coba di atas, didapatkan rata-rata lama pengenalan objek yang dilakukan terhadap bangunan dan ruangan yang ada di kampus ITATS sebesar 2.333 detik dari 49 objek terkenali. Dan semakin besar fitur dalam proses pengenalan objek, maka semakin besar pula waktu yang dibutuhkan untuk mengenali objek yang ditangkap oleh kamera.

5. Kesimpulan

Setelah dilakukan pengujian data pada saat mengarahkan aplikasi *FindGo-ITATS* ke lokasi yang ingin diketahui informasinya dan berada di kampus ITATS, dapat disimpulkan bahwa:

1. Hasil uji coba yang dilakukan terhadap bangunan dan ruangan yang ada di kampus ITATS dengan 4 kriteria yang telah ditentukan, didapatkan presentase akurasi pengenalan di setiap kriteria sebagai berikut:
   a. Pengujian dengan 15 lokasi yang berbeda bentuk objeknya sebesar 73.33%
   b. Pengujian dengan 15 bentuk objek yang hampir sama di lokasi berbeda sebesar 80%
   c. Pengujian dengan 15 ruangan yang identik sebesar 100%
   d. Pengujian dengan 15 objek pencerminan 73.33%

Dan dari keempat kriteria tersebut didapatkan rata-rata tingkat keberhasilan dalam mengenali objek sebesar 81.66%

2. Dari hasil uji coba didapatkan rata-rata lama pengenalan objek yang dilakukan terhadap bangunan dan ruangan yang ada di kampus ITATS sebesar 2.333 detik.
3. Dan semakin besar fitur dalam proses pengenalan objek, maka semakin besar pula waktu yang dibutuhkan untuk mengenali objek yang ditangkap oleh kamera.

6. Daftar Pustaka

Anitha, J. Jasmine and S.M. Deepa. 2014. "*Tracking and Recognition of Objects using SURF Descriptor and Harris Corner Detection*". Nehru Institute of Engineering and Technology (Anna University), Coimbatore. India

Affan dan Eddo, Mahtarami, Ertino. 2012. "*Game Marble Maze Berbasis Augmented Reality*". Universitas Islam Indonesia.Yogyakarta.

Budiyatno, Slamet. 2012. "*Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial: Penerapan Augmented Reality Sebagai Penampil Informasi Hasil Pengenalan Wajah Pada Perangkat Android*". Fakultas Teknik - Program Studi Teknik Komputer. Depok.

Cawood, Stephen, Fiala, Mark. 2008. "*Augmented Reality: A Practical Guide"*, The Pragmatic Bookshell, Raleigh. Texas

Hongbo Li, Ming Qi And Yu Wu. 2012. "*A Real-Time Registration Method Of Augmented Reality Based On SURF And Optical Flow*". Institute of Web Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, 400065. China

Tania, Ken Ditha. 2010. "*Pengenalan Gambar Menggunakan Sebagian Data Gambar*". Universitas Sriwijaya. Palembang.