

Penerjemahan Bahasa Isyarat Indonesia Menggunakan Kamera pada Telepon Genggam *Android*

Muhammad Yunus A, Dr. Ir. Djoko Purwanto, M.Eng., Ronny Mardiyanto, S.T., M.T., Ph.D.
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
e-mail: djoko@ee.its.ac.id, ronny@ee.its.ac.id, muhammad.yunus12@mhs.ee.its.ac.id

Abstrak—Penginderaan visual atau *machine vision* merupakan suatu proses manipulasi data citra. Data tersebut dapat digunakan untuk melakukan interpretasi banyak hal, salah satunya yaitu pengenalan *gesture*. Pengenalan *gesture* adalah antarmuka yang dapat mengenali gerak-isyarat seorang manusia dan mentranslasikan gerakan tersebut sebagai instruksi yang dapat dipahami oleh komputer. Pengenalan *gesture* dapat digunakan untuk menerjemahkan bahasa isyarat pada orang tunawicara. Hal ini karena banyaknya orang yang tidak mengerti bahasa tangan orang tunawicara. Sehingga, orang tunawicara kesulitan dalam berinteraksi di masyarakat.

Pada tugas akhir ini pengenalan *gesture* untuk penerjemahan bahasa isyarat lebih mengarah pada *hand recognition*, yaitu pendeteksian perubahan gerak tangan, dengan menggunakan *android mobile phone* sebagai divisinya. *Android mobile phone* memiliki kamera untuk menangkap citra orang tuna wicara saat berkomunikasi menggunakan bahasa isyarat berupa gerakan tangan. Selanjutnya, citra diproses oleh *processing unit android* untuk melakukan proses *hand recognition*. Setelah proses tersebut selesai, maka layar *display* akan memunculkan huruf atau kata dari perubahan posisi gerak tangan yang dilakukan orang tunawicara yang berada di depan kamera.

Kata kunci—pengenalan *gesture*, kamera, *machine vision*, *android*, *hand recognition*.

I. PENDAHULUAN

Berkomunikasi merupakan kebutuhan manusia dalam berinteraksi antara satu dengan yang lain. Banyak cara yang dilakukan untuk berkomunikasi, diantaranya adalah dengan berbicara melalui lisan atau dengan tangan melalui bahasa isyarat, serta tulisan. Dalam masyarakat terdapat kaum tunawicara yang karena keterbasannya tidak dapat menggunakan bahasa lisan. Mereka hanya dapat mengandalkan komunikasi melalui bahasa isyarat atau tulisan. Bahasa isyarat yang digunakan oleh kaum tunawicara ini sulit dipahami oleh masyarakat pada umumnya, sehingga kaum tunawicara merasa terasingkan bagi lingkungan disekitarnya.

Seiring dengan kemajuan teknologi, telah dilakukan penelitian dalam rangka untuk menghasilkan piranti bantu untuk menerjemahkan bahasa isyarat ke dalam tulisan bahkan suara. Kategori penelitian yang dilakukan dapat dibedakan menjadi dua yaitu pendekatan berbasis visi *computer (computer vision)* dan pendekatan berbasis sensor. Pada pendekatan berbasis visi *computer* digunakan *file* video atau langsung melalui kamera yang menangkap gerakan bahasa isyarat. Pada pendekatan ini sebelum dilakukan proses

pengenalan harus dilakukan pra proses berupa pengolahan citra dulu, seperti segmentasi dan *tracking* tangan. Sedangkan pendekatan dengan sensor terintegrasi dengan sarung tangan (*glove*) yang menghasilkan besaran listrik yang terukur, untuk mengetahui derajat tekukan jari jari tangan dan gerakan tangan.

Dalam bahasa isyarat, setiap gerakan sudah menugaskan makna, dan aturan yang kuat dari konteks dan tata bahasa dapat diterapkan untuk membuat pengakuan penurut. Bahasa Isyarat Indonesia (BISINDO) adalah bahasa pilihan untuk kebanyakan tuli dan bisu di Indonesia. Sistem Isyarat Bahasa Indonesia (SIBI) menggunakan sekitar 6.000 gerak kata-kata umum dan jari ejaan kata-kata untuk berkomunikasi mengaburkan atau kata benda [1]. Namun, mayoritas penandatanganan adalah dengan kata-kata penuh, memungkinkan percakapan ditandatangani untuk melanjutkan pada tentang laju percakapan lisan. Tata bahasa BISINDO memungkinkan lebih banyak fleksibilitas dalam urutan kata dari bahasa Indonesia dan kadang-kadang menggunakan redundansi untuk penekanan kata atau ejaan.

Untuk memberikan kemudahan terhadap penyandang tunawicara dan tunarungu, maka perlu adanya modul yang bisa membantu penyandang tunawicara untuk berinteraksi dengan orang lain. Dengan menggunakan telepon genggam yang telah dilengkapi dengan kamera dan sistem *android* untuk menerjemahkan bahasa isyarat. Penyandang tunawicara bisa lebih mudah berinteraksi dengan masyarakat umum. Serta masyarakat umum bisa belajar untuk mengerti arti bahasa yang diisyaratkan penyandang tunarungu dan tunawicara.

II. TEORI PENUNJANG

A. *Contour Convexity and Convexity Defects*

Cara lain yang berguna untuk memahami bentuk objek atau kontur adalah untuk menghitung *hull convex* untuk objek dan kemudian menghitung sifat busung yang cacat. Bentuk banyak kompleks objek juga ditandai dengan cacat tersebut. Gambar dibawah menggambarkan konsep *convexity defects* menggunakan gambar tangan manusia. *Convexity defects* digambarkan sebagai garis gelap di sekitar tangan, dan daerah-daerah yang berlabel A sampai H adalah setiap "*defect*" relatif terhadap *hull* itu. Seperti yang Anda lihat, *convexity defects* ini menawarkan sarana mencirikan tidak hanya tangan sendiri, tetapi juga pusat tangan.



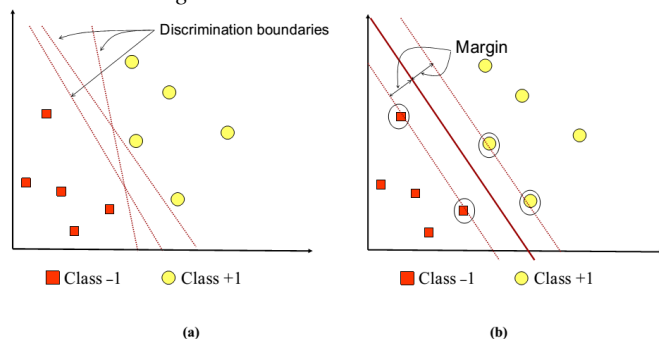
Gambar 1. Convex hull enclosing the hand region [2]

Ada tiga metode penting *OpenCV* yang berhubungan dengan *complex hulls* dan *convexity defects*. Pertama hanya menghitung *hulls of a contour* yang kami telah mengidentifikasi, dan kedua memungkinkan kita untuk memeriksa apakah kontur diidentifikasi sudah *convex*. Ketiga menghitung sifat *convexity defects* dalam kontur yang *convex hull* dikenal [2], [3].

B. Support Vector Machine (SVM) Classification

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition*. Sebagai salah satu metode *pattern recognition*, usia SVM terbilang masih relatif muda. Walaupun demikian, evaluasi kemampuannya dalam berbagai aplikasinya menempatkannya sebagai *state of the art* dalam *pattern recognition*, dan dewasa ini merupakan salah satu tema yang berkembang dengan pesat. SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada input *space*. Tulisan ini membahas teori dasar SVM dan aplikasinya dalam bioinformatika, khususnya pada analisa ekspresi gen yang diperoleh dari analisa *microarray*.

• **Pattern Recognition**



Gambar 2. SVM berusaha menemukan *hyperplane* terbaik yang memisahkan kedua *class* -1 dan +1

Pattern Recognition merupakan salah satu bidang dalam komputer sains, yang memetakan suatu data ke dalam konsep tertentu yang telah didefinisikan sebelumnya. Konsep tertentu ini disebut *class* atau *category*. Aplikasi *pattern recognition* sangat luas, di antaranya mengenali suara dalam sistem sekuriti, membaca huruf dalam OCR, mengklasifikasikan

penyakit secara otomatis berdasarkan hasil diagnosa kondisi medis pasien dan sebagainya. Berbagai metode dikenal dalam *pattern recognition*, seperti linear, *discrimination analysis*, *hidden markov model* hingga metode kecerdasan buatan seperti *artificial neural network*. Salah satu metode yang akhir-akhir ini banyak mendapat perhatian sebagai *state of the art* dalam *pattern recognition* adalah *Support Vector Machine* (SVM). *Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti margin *hyperplane* (Duda & Hart tahun 1973, Cover tahun 1965, Vapnik 1964, dsb.), kernel diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain. Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut.

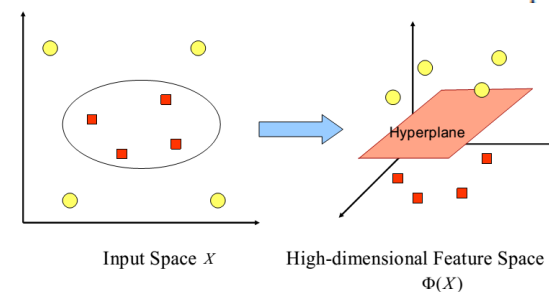
• **Kernel Trick dan Nonlinear Classification SVM**

Pada umumnya masalah dalam domain dunia nyata (*real world problem*) jarang yang bersifat *linear separable*. Kebanyakan bersifat nonlinear. Untuk menyelesaikan *problem* nonlinear, SVM dimodifikasi dengan memasukkan fungsi Kernel.

Dalam nonlinear SVM, pertama-tama data \vec{x} dipetakan oleh fungsi $\Phi(\vec{x})$ ke ruang vektor yang berdimensi lebih tinggi. Pada ruang vektor yang baru ini, *hyperplane* yang memisahkan kedua *class* tersebut dapat dikonstruksikan. Hal ini sejalan dengan teori Cover yang menyatakan. Jika suatu transformasi bersifat nonlinear dan dimensi dari *feature space* cukup tinggi, maka data pada input *space* dapat dipetakan ke *feature space* yang baru, dimana *pattern-pattern* tersebut pada probabilitas tinggi dapat dipisahkan secara linear.

Ilustrasi dari konsep ini dapat dilihat pada gambar dibawah. Pada gambar a diperlihatkan data pada *class* kuning dan data pada *class* merah yang berada pada input *space* berdimensi dua tidak dapat dipisahkan secara linear. Selanjutnya gambar b menunjukkan bahwa fungsi Φ memetakan tiap data pada input *space* tersebut ke ruang vektor baru yang berdimensi lebih tinggi (dimensi 3), dimana kedua *class* dapat dipisahkan secara linear oleh sebuah *hyperplane*. Notasi matematika dari *mapping* ini adalah sebagai berikut.

$$\Phi: \mathbb{R}^d \rightarrow \mathbb{R}^q \quad d < q \quad (2.15)$$



Gambar 3. Fungsi Φ memetakan data ke ruang vektor yang berdimensi lebih tinggi, sehingga kedua *class* dapat dipisahkan secara linear oleh sebuah *hyperplane*.

Tabel 1.
Jenis Kernel pada SVM

Jenis Kernel	Definisi
Polynomial	$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^p$
Gaussian	$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\ \vec{x}_i - \vec{x}_j\ ^2}{2\sigma^2}\right)$
Sigmoid	$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i \cdot \vec{x}_j + \beta)$

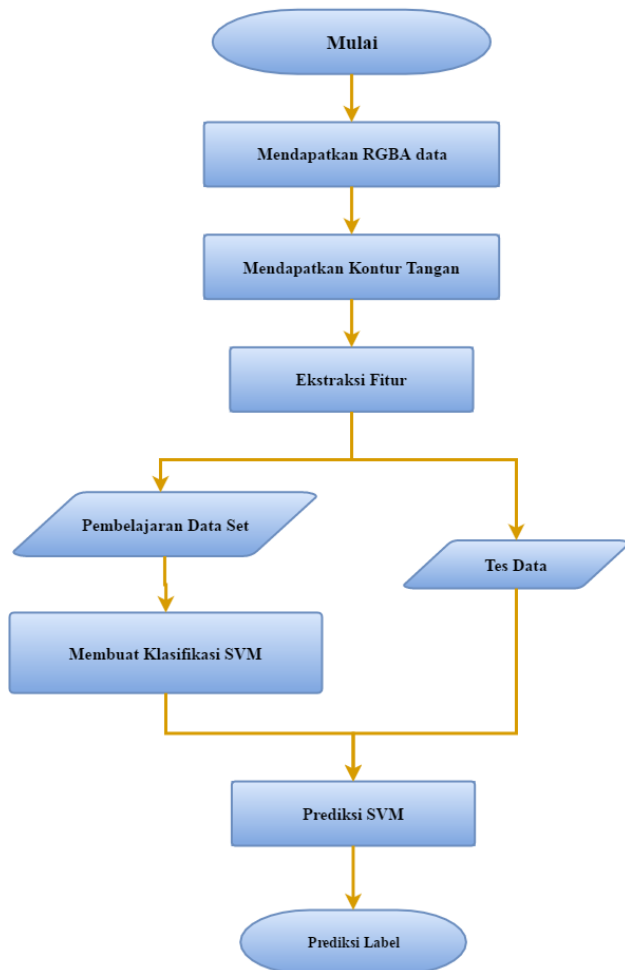
Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada input *space* akan berjarak dekat juga pada *feature space*, sebaliknya dua data yang berjarak jauh pada input *space* akan juga berjarak jauh pada *feature space*.

III. PERANCANGAN SISTEM

Desain sistem dijelaskan dalam diagram alur, seperti dalam gambar 3. Secara umum, studi ini dibagi menjadi dua tahap, antara lain:

Tahap pertama adalah deteksi tangan gambar yang diambil. Selain itu, gambar diambil untuk menemukan fitur dari tangan

Tahap kedua adalah fitur klasifikasi tahap sebelumnya. Klasifikasi ini digunakan sebagai sebuah set data untuk memprediksi label.

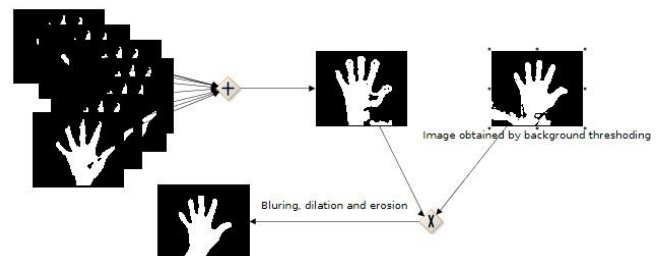


Gambar 4. Diagram alur sistem

IV. IMPREMENTASI DAN PENGUJIAN

A. Presampling dan Segementasi Tangan

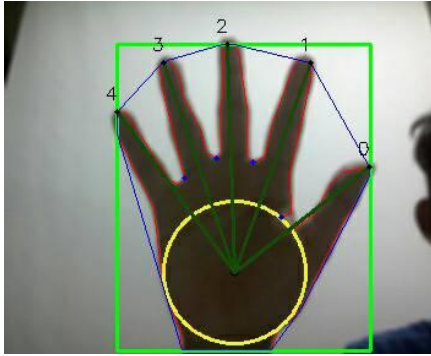
Aplikasi ini dimulai dengan dua langkah *pra-sampling*, yang mengumpulkan warna latar belakang dan tangan pengguna menggunakan 7 kotak kecil yang ditampilkan pada layar. Data warna ini digunakan untuk menghitung batas untuk mendapatkan gambar biner dari input RGBA data. Untuk kesederhanaan, adaptif metode tidak digunakan untuk menemukan ambang terbaik. Sebaliknya, pengguna diminta untuk memasukkan tangan dekat dengan layar untuk menutupi kotak 7 sehingga program dapat memperoleh data warna tangan. Dicatat bahwa jumlah kotak hanya nilai empiris dan dapat nilai lainnya. Setelah data 7 warna untuk tangan yang diperoleh, batas-batas atas dan bawah 7 untuk daerah tangan dihitung, yang dapat digambarkan sebagai vektor dimensi 6. Vektor melompat-lompat ditentukan oleh pengembang dan tidak bisa diubah. Oleh karena itu, kinerja segmentasi benar-benar tergantung pada pilihan dari vektor yang terus bergerak. Untuk lebih memahami pengaruh dari setiap elemen dari vektor pada kinerja segmentasi, Ruang warna RGB asli dikonversi menjadi banyak ruang warna lain seperti HSV, YCrCb, CIE L * * b *. Karena untuk tangan yang sama, warna biasanya bervariasi pada dimensi pencahayaan dan lebih kecil di hue dan dimensi warna lain, Ruang warna yang diinginkan harus mampu memisahkan mereka. Setelah beberapa percobaan ruang tiga warna yang tercantum adalah semua ditemukan untuk melakukan cukup baik, tapi lebih mudah untuk menemukan vektor terbaik melompat-lompat dengan menggunakan ruang LAB. Oleh karena itu ruang warna LAB digunakan [2].



Gambar 5. Segmentasi Filter [3]

Setelah batas-batas untuk masing-masing warna 7 data yang diperoleh, 7 biner gambar tangan bisa dihitung, yang kemudian diringkas menggunakan operasi Logis "atau". Hal yang sama yang dilakukan pada warna latar belakang, menghasilkan gambar biner yang lain. Operasi Logis "dan" kemudian dilakukan pada dua gambar biner dan hasilnya adalah kabur, melebar terkikis untuk menghilangkan suara dan gambar biner akhir diperoleh, seperti yang ditunjukkan pada gambar di atas [4].

B. Ekstraksi Fitur



Gambar 6. *Gesture Extraction*

Fitur digunakan untuk mewakili tangan adalah fitur tingkat rendah, yang pada dasarnya vektor jari. Untuk menghitung mereka, hal ini diperlukan untuk mengetahui lokasi pusat *palm* dan ujung jari. Beberapa fungsi *OpenCV* langsung dapat mengembalikan kontur dan cembung *hull* point tangan dari gambar biner yang mengandung sisi tersegmentasi. Menggunakan koordinat di bawah ini dan kotak melompat-lompat, Pusat dan jari-jari lingkaran tertulis dari kontur tebing bisa dihitung. Bagian ini cukup produk mahal dan dilaksanakan dalam kode C++ asli.

Lokasi dari ujung jari dihitung menggunakan titik-titik konveks. Untuk melakukannya, titik-titik konveks dieliminasi dengan memeriksa kendala pada kedalaman dan sudut poin yang cacat dan sebagainya, seperti yang ditunjukkan pada gambar di atas [3]. Selanjutnya Cacat poin kiri adalah mengatur kembali dan ujung jari yang Diperoleh dari koordinat kembali *OpenCV* fungsi. Akhirnya vektor jari dihitung dan dibagi dengan jari-jari lingkaran *inscribe* untuk mendapatkan vektor fitur akhir:

$$X=[x0/r,y0/r,x1/r,y1/r\dots x4/r,y4/r]$$

C. Training set

Saat ini 5 gerakan telah dikumpulkan merupakan kumpulan data awal, seperti yang ditunjukkan di atas gambar. Catatan bahwa meskipun diekstrak fitur vektor invarian untuk ukuran yang berbeda dari tangan, ini adalah varian untuk orientasi berbeda karena gerak gerak macam orientasi diperlakukan sebagai gerakan yang berbeda. Juga dapat dilihat dari angka yang "gerakan" diberi label sebagai 0 yang sebenarnya tidak sikap. Label 0 dicadangkan untuk mewakili apa fitur yang tidak terdeteksi. Bisa dihitung sebagai mewakili contoh negatif. Ini juga menunjukkan salah satu batasan bahwa sikap "kepala" tidak dapat diakui.

Pengguna dapat menambahkan apa pun gerakan mereka ingin set data awal atau membangun satu lagi menggunakan aplikasi. Proses pelatihan dapat dilakukan dalam kondisi apapun selama tangan dapat dideteksi baik. Karena kendala waktu, sangat sedikit jumlah bingkai yang dikumpulkan untuk setiap gerakan (hanya 10 *frame*). SVM multi kelas digunakan untuk membangun model pelatihan dan membuat prediksi.

V. HASIL DAN ANALISIS

Untuk menunjukkan hasil, beberapa tes dilakukan dalam kondisi pencahayaan yang berbeda dan latar belakang yang berbeda. Untuk setiap gerakan, 3 *frame* ditangkap dengan label diperkirakan ditampilkan sebagai nomor merah dalam foto. Tiga *frame* jauh lebih sedikit daripada cukup untuk membuktikan akurasi. Namun menurut pengamatan, hasil akhir pengujian sangat tergantung pada hasil deteksi. Dengan kata lain, jika tangan baik terdeteksi, yang berarti vektor fitur diekstraksi dengan benar mewakili gerakan tangan, maka tangan dapat diklasifikasikan ke dalam kategori yang tepat dengan probabilitas tinggi. Oleh karena itu apa yang sebenarnya penting adalah benar-benar kinerja tangan deteksi dan bukannya *gesture*. Deteksi tangan dapat dilihat dari gambar yang diambil, yang menampilkan unsur-unsur seperti vektor kontur dan jari. Tiga *frame* yang cukup untuk melihat apakah unsur-unsur ini benar dan jika mereka tidak stabil atau tidak.

Aplikasi ini diuji dalam tiga skenario dan hasil yang ditampilkan dalam angka. Hal ini dapat dilihat bahwa aplikasi bekerja cukup baik dalam skenario 1 dan skenario 3: Semua label diperkirakan sudah benar dan sebagian besar gerak tubuh benar terdeteksi dan cukup stabil, kecuali bahwa ada satu gerakan tidak terdeteksi (*Random* kesalahan). Dan alasan utama untuk keberhasilan adalah bahwa ada kontras yang tajam dalam warna antara latar belakang dan tangan. Apakah latar belakang berantakan atau tidak masalah, hanya warna penting.

Dalam skenario 2 beberapa dari gerak tubuh tidak baik terdeteksi. Penyebab utama untuk ini adalah bahwa warna kuning di latar belakang semacam mengganggu ekstraksi kontur tangan. Juga penyesuaian *autofocus* kamera menjadi lebih kuat dalam skenario ini yang mungkin karena cahaya.

Singkatnya, dengan bantuan dari warna-warna *presample*, sistem dapat mendeteksi dan mengenali tangan cukup baik dan stabil dalam sebagian besar kondisi pencahayaan dan latar belakang, selama sebagai kontras latar belakang dan tangan ini tidak terlalu rendah. Namun kendala yang adalah bahwa mereka (latar belakang) harus tetap sama setelah menyelesaikan proses *presample*. Dan itu berarti kedua tangan dan telepon tidak harus bergerak terlalu banyak, untuk menghindari perubahan warna.

DAFTAR PUSTAKA

- [1] Hurlbut, Hope M, "The Signed Languages of Indonesia: An Enigma," Indonesia, Survey Report.
- [2] R. Shrivastava, "A hidden Markov model based dynamic hand *gesture* recognition system using *OpenCV*," 2013, pp. 947-950.
- [3] R. M. Gurav and P. K. Kadbe, "Real time finger *tracking* and contour detection for *gesture* recognition using *OpenCV*," 2015, pp. 974-977.
- [4] A. S. Konwar, B. S. Borah, and C. T. Tuithung, "An American Sign Language detection system using HSV color model and edge detection," 2014, pp. 743-747.