

# SISTEM PENERJEMAH SANDI SEMAPHORE DENGAN SENSOR KINECT SEBAGAI MEDIA PEMBELAJARAN

Ino Suryana<sup>1)</sup>, Erick Paulus<sup>2)</sup>, Betty Subartini<sup>3)</sup>, Abdi Hutama<sup>4)</sup>

Departemen Matematika FMIPA Universitas Padjadjaran

Jl. Raya Bandung-Sumedang KM 21 Jatinangor Sumedang 45363

Email : inosur@gmail.com<sup>1)</sup>, erick\_paulus@yahoo.com<sup>2)</sup>, betty.subartini@gmail.com<sup>3)</sup>, abdi.hutama@gmail.com<sup>4)</sup>

## ABSTRACT

*Semaphore is a way to send and receive the news by using banner media, paddles, rods, bare hands or with gloves. The information obtained is translated through the position of the flag, the flag called semaphores. Semaphore's code has a lot of variety of motion, but at this time there are only 36 kind of motions, they are the letters from A until Z and the numbers from 0 until 9.*

*The translation of the Semaphore's code is implemented by using the theory of the Boolean's system, that is, if the hands touch the different point of Kinect detection and it's contained in the rules that have been made, then the application will respond by displaying the letters or numbers on the screen of the translation. There are eight points with a circular pattern made with the implementation of the Cartesian's coordinate system method. This application is built by using Microsoft Visual Studio C# programming 2012 Professional Edition software with the Library of Microsoft Kinect version 1.8.0.0.*

*This application will be integrated with the camera Kinect sensor. Actually, Kinect is a series of tools that have the ability to detect a motion. With Kinect, this application can be operated only with the motion command. This application is also have a game feature that make it more interactive and fun. Users need to pay attention to the light intensity and the distance between users and the Kinect sensor to obtain a more optimal result.*

### Keywords:

*human computer interaction, semaphore flag, kinect, motion sensor, scout, c# programming*

## 1. Pendahuluan

Dalam berkomunikasi sehari-hari, manusia menggunakan bahasa verbal untuk menyampaikan maksud kepada lawan bicara. Bagaimana jika sedang

dalam keadaan darurat dan komunikasi verbal tidak memungkinkan? Salah satu solusi yang bisa digunakan adalah menggunakan bahasa sandi. Bahasa sandi mempunyai banyak ragam, namun yang akan di bahas oleh peneliti di sini adalah sandi bendera *Semaphore*.

Sandi *Semaphore* juga sering diajarkan dalam pramuka. Dengan cara konvensional, para siswa akan mencari rekan atau orang yang telah menghafal semua gerakan sandi *Semaphore* untuk mendampinginya. Dari sini muncul ide awal untuk mencoba memadukan sandi *Semaphore* dengan teknologi *Kinect*. Dengan menggunakan bantuan *Kinect*, akan dibuat sebuah sistem aplikasi permainan edukasi yang diberi nama Penerjemah Sandi *Semaphore* Virtual.

Sebelumnya, Trianto Juliatmojo dan Eko Ariwibowo [1] pernah melakukan penelitian serupa. Namun penelitian yang dilakukan hanya sebatas memperlihatkan contoh simulasi gerakan melalui tampilan animasi yang dibuat dengan menggunakan *software* Macromedia Flash 8. Sedangkan pada penelitian kali ini pengguna bisa memperagakan ragam gerakan sandi *Semaphore* secara langsung, bukan hanya melihatnya pada layar. Jadi pengguna bisa langsung mengetahui dan mengoreksi gerakan sandi *Semaphore* yang mereka peragakan.

Penerjemah sandi *Semaphore* virtual ini nantinya diharapkan dapat digunakan sebagai alat bantu pada pelatihan pramuka di sekolah, angkatan laut, atau instansi lain yang membutuhkan. Oleh karena itu, penulis tertarik untuk melakukan penelitian tentang penerjemah sandi *Semaphore* virtual ini.

## 2. Tinjauan Pustaka

### 2.1 Interaksi Manusia Komputer

Secara singkat, interaksi manusia dan komputer (IMK) bisa dikatakan bidang ilmu yang mempelajari tentang interaksi antara manusia dan komputer. Interaksi manusia komputer merupakan disiplin ilmu yang mempelajari

hubungan antara manusia dan komputer yang meliputi perancangan, evaluasi, dan implementasi antarmuka pengguna komputer agar mudah digunakan oleh manusia[2]. Pengertian Interaksi adalah komunikasi dua arah antara manusia (*user*) dan sistem komputer. Interaksi menjadi maksimal apabila kedua belah pihak mampu memberikan stimulan dan respon (aksi & reaksi) yang saling mendukung, jika salah satu tidak bisa, maka interaksi akan mengalami hambatan atau bahkan menuju pembiasaan tujuan.

## 2.2 Microsoft Kinect

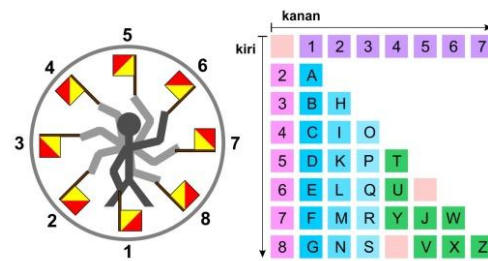
Kinect dikembangkan secara internal oleh Rare, anak perusahaan dari Microsoft Game Studios milik Microsoft, dan teknologi kamera oleh pengembang Israel, PrimeSense. PrimeSense mengembangkan sistem yang dapat menginterpretasikan *gestur* secara spesifik, sehingga kontrol secara *hands-free* dapat dilakukan pada perangkat elektronik menggunakan proyektor inframerah dan kamera, serta *microchip* khusus untuk melacak pergerakan objek dan individu pada bidang tiga dimensi. Sistem 3D *scanner* tersebut dinamakan *Light Coding* yang menggunakan variasi dari rekonstruksi gambar 3D. Kinect memiliki IR depth-sensor dan *RGB-camera*, yang masing-masing menggunakan resolusi  $640 \times 480$  piksel pada *frame rate* sekitar 30Hz[3]. Salah satu penerapan kinect yang mampu mengukur tinggi badan telah meningkatkan kinerja pengenalan karakter[4].

## 2.3 Operasi Logika

Operasi logika yang digunakan adalah *NOT*, *AND*, *OR*, *NOR*, *NAND*, *XOR* dan *XNOR*

## 2.4 Sandi Semaphore

Sistem sandi bendera Semaphore adalah suatu cara yang digunakan untuk menyampaikan huruf alfabet yang berdasar pada gerakan sepasang bendera pada tangan dengan pola tertentu [5]. Pada dasarnya, gerakan sandi *Semaphore* ini menggunakan gerakan melingkar dengan dua tangan sejauh  $360^\circ$  dan dibagi menjadi pola delapan titik utama. Delapan titik utama didapat dari hasil bagi sudut  $360^\circ$  dengan sudut  $45^\circ$  (karena jarak antara titik satu dengan lainnya sebesar  $45^\circ$ ) maka di dapatlah pola delapan titik utama. Delapan pola titik ini yang nantinya akan ditampilkan pada layar dan dijadikan acuan gerakan oleh pengguna.



Gambar 1. Ragam Gerakan Sandi *Semaphore*

## 3. Hasil Percobaan

### 3.1 Analisa Sistem

Tahap analisa sistem meliputi identifikasi *input* dan identifikasi *output*. Permasalahan yang akan dibangun dalam sistem ini terbatas pada tahapan penerjemahan gerakan sandi *Semaphore* dari huruf abjad A-Z serta angka 0-9.

#### 1. Identifikasi *Input*

Identifikasi *input* diperlukan untuk melakukan pengumpulan data-data dan informasi yang mendukung dalam pembuatan aplikasi ini yang diperlukan untuk memecahkan masalah dan selanjutnya akan diproses oleh sistem yang telah dibangun dalam aplikasi ini. Pada layar, sistem akan menunjukkan pola sepuluh titik dalam lingkaran  $360^\circ$  yang telah ditentukan kemudian dengan mengarah ke kamera sensor Kinect, pengguna akan melakukan gerakan-gerakan sandi *Semaphore* yang mana sepuluh titik yang telah dibuat merupakan acuan. Kemudian pengguna harus menunggu selama dua detik sampai sistem merespon gerakan pengguna.

#### 2. Identifikasi *Output*

Identifikasi *output* ini merupakan hasil keluaran dari sistem aplikasi ini. Sistem akan menerima masukan (*input*) dari pengguna melalui gerakan-gerakan pada sensor Kinect, kemudian sistem akan merespon dan memprosesnya sesuai aturan yang telah dibuat pada sistem dan selanjutnya hasil penerjemahan gerakan akan ditampilkan pada layar.

### 3.2 Perancangan Sistem

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan pembuatan alur atau langkah-langkah dalam penyelesaian masalah yang disusun secara sistematis, atau disebut algoritma. Algoritma ini digunakan untuk membantu dalam mengkonversikan permasalahan yang ada ke dalam bahasa pemrograman. Dalam pembangunan aplikasi penerjemah sandi *Semaphore* ini, algoritma yang disusun akan menggambarkan tahapan atau

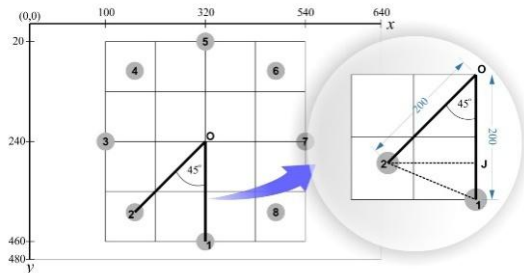
proses dari dimulainya *input* atau perintah berupa gerakan dari pengguna hingga munculnya keluaran atau hasil terjemahan terhadap gerakan sandi *Semaphore* yang diperagakan oleh pengguna.



Gambar 2. Alur kerja (*flowchart*) aplikasi

Perancangan sistem dimulai dengan mengidentifikasi permasalahan hingga pengujian. Permasalahan spesifik yang akan diangkat adalah mengenai tahapan pendeteksian gerakan sandi *Semaphore*, berapa banyak ragam gerakannya, berapa persen tingkat akurasi pendeteksian aplikasi terhadap gerakan yang dilakukan pengguna, hingga solusi untuk memperbaiki jika terjadi kesalahan dalam pembacaan gerakannya.

### 3.2.1 Perhitungan Jarak dan Titik Koordinat dengan konsep Trigonometri



Gambar 3. Pola titik pada sandi *Semaphore*

Perhitungan titik-titik koordinat *x* dan *y* pada pola titik sandi *Semaphore*:

- Titik 1  
Titik koordinat *x* pada titik 1 didapat dari panjang keseluruhan garis koordinat *x* dibagi dua  $\frac{640 \text{ piksel}}{2} = 320$ .  
Sedangkan titik koordinat *y* didapat dari hasil penjumlahan antara titik tengah koordinat  $y \left( \frac{480 \text{ piksel}}{2} = 240 \text{ piksel} \right)$  dengan 200 piksel (jarak antara titik tengah dengan titik acuan yang telah

ditentukan). Maka titik koordinat  $y = 240 \text{ piksel} + 200 \text{ piksel} = 440$ .

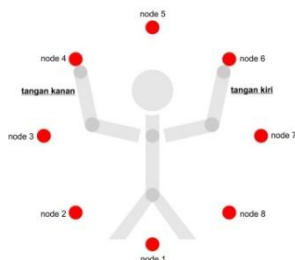
Jadi titik koordinat titik 1 = (320, 440).

- Titik 2  
Perhatikan segitiga sama kaki 2O1 pada gambar 3.3 di atas. Untuk mendapatkan koordinat *x* pada titik 2, terlebih dahulu akan dicari panjang garis 2J. Untuk mencari panjang garis 2J digunakan persamaan trigonometri sebagai berikut:  
 $\sin 45^\circ = \frac{2J}{200} = \frac{2J}{200}$  maka  $2J = 200 \times \sin 45^\circ = 200 \times \frac{1}{2}\sqrt{2} = 141,42135$  dibulatkan menjadi 141 piksel. Maka koordinat *x* pada titik 2 =  $320 \text{ piksel} (\text{titik tengah koordinat } x) - 141 \text{ piksel} = 179$ .  
Selanjutnya untuk mendapatkan titik koordinat *y* pada titik 2, terlebih dahulu akan dicari panjang garis OJ. Perhatikan segitiga 2OJ, karena segitiga 2OJ yang merupakan segitiga sama kaki, maka panjang garis OJ = 2J, yakni 141 piksel. Maka titik koordinat *y* pada titik 2 =  $240 \text{ piksel} (\text{titik tengah koordinat } y) + 141 \text{ piksel} = 381$ .  
Jadi titik koordinat titik 2 = (179, 381).
- Titik 3  
Titik koordinat *x* pada titik 3 =  $320 \text{ piksel} - 200 \text{ piksel} = 120$ .  
Titik koordinat *y* pada titik 3 sama dengan titik tengah koordinat *y*, yakni 240.  
Jadi titik koordinat titik 3 = (120, 240).
- Titik 4  
Titik koordinat *x* pada titik 4 sejajar dengan titik koordinat *x* pada titik 2, yakni 179.  
Titik koordinat *y* =  $240 \text{ piksel} - 141 \text{ piksel} = 99$ .  
Jadi titik koordinat titik 4 = (179, 99).
- Titik 5  
Titik koordinat *x* pada titik 5 sama dengan titik tengah koordinat *x*, yakni 320.  
Titik koordinat *y* =  $240 \text{ piksel} - 200 \text{ piksel} = 40$ .  
Jadi titik koordinat titik 5 = (320, 40).
- Titik 6  
Titik koordinat *x* =  $320 \text{ piksel} + 141 \text{ piksel} = 461$ .  
Titik koordinat *y* =  $240 \text{ piksel} - 141 \text{ piksel} = 99$ .  
Jadi titik koordinat titik 6 = (461, 99).
- Titik 7  
Titik koordinat *x* =  $320 \text{ piksel} + 200 \text{ piksel} = 520$ .  
Titik koordinat *y* pada titik 7 sama dengan titik tengah koordinat *y*, yakni 240.  
Jadi titik koordinat titik 7 = (520, 240).
- Titik 8  
Titik koordinat *x* =  $320 \text{ piksel} + 141 \text{ piksel} = 461$ .  
Titik koordinat *y* =  $240 \text{ piksel} + 141 \text{ piksel} = 381$ .  
Jadi titik koordinat titik 8 = (461, 381).

### 3.2.2 Cara Kerja Aplikasi

Cara kerja aplikasi ini dibagi menjadi beberapa tahap, yaitu:

1. Deteksi pola titik acuan (*node*) yang akan dihubungkan dengan sensor Kinect (ilustrasi pada gambar 5).
2. Identifikasi skeleton serta posisi tangan kiri dan kanan (ilustrasi pada gambar 5).



Gambar 4. Ilustrasi pola *node* dan deteksi *skeleton*

3. Menghitung jarak antara *skeleton* yang ada di tangan (kiri dan kanan) dengan *node* sandi *Semaphore* yang telah dibuat.
4. Aplikasi akan melakukan proses selanjutnya jika jarak antara titik pada tangan lebih kecil dari radius titik *Semaphore* yang telah dibuat.
5. Selanjutnya akan dilakukan proses boolean jika tangan kiri dan kanan masing-masing berada dalam radius titik pada pola sandi *Semaphore*.
6. Hasil terjemahan gerakan akan ditampilkan pada layar jika gerakan atau titik yang disentuh sesuai dengan aturan yang ada pada tabel 1:

Tabel 1. Pemetaan Gerakan dan hasilnya

No	Input Gerakan		Output (Hasil Terjemahan)
	Kanan	Kiri	
1.	Titik 1	Titik 2	Huruf A atau angka 1
2.	Titik 1	Titik 3	Huruf B atau angka 2
3.	Titik 1	Titik 4	Huruf C atau angka 3
4.	Titik 1	Titik 5	Huruf D atau angka 4
5.	Titik 6	Titik 1	Huruf E atau angka 5
6.	Titik 7	Titik 1	Huruf F atau angka 6
7.	Titik 8	Titik 1	Huruf G atau angka 7
8.	Titik 2	Titik 3	Huruf H atau angka 8
9.	Titik 2	Titik 4	Huruf I atau angka 9
10.	Titik 7	Titik 5	Huruf J
11.	Titik 5	Titik 2	Huruf K atau angka 0
12.	Titik 6	Titik 2	Huruf L
13.	Titik 7	Titik 2	Huruf M
14.	Titik 8	Titik 2	Huruf N
15.	Titik 4	Titik 3	Huruf O
16.	Titik 5	Titik 3	Huruf P
17.	Titik 6	Titik 3	Huruf Q
18.	Titik 7	Titik 3	Huruf R
19.	Titik 8	Titik 3	Huruf S
20.	Titik 5	Titik 4	Huruf T

No	Input Gerakan		Output (Hasil Terjemahan)
	Kanan	Kiri	
21.	Titik 6	Titik 4	Huruf U
22.	Titik 8	Titik 1	Huruf V
23.	Titik 3	Titik 2	Huruf W
24.	Titik 2	Titik 4	Huruf X
25.	Titik 3	Titik 8	Huruf Y
26.	Titik 3	Titik 4	Huruf Z

### 3.3 Implementasi dan Hasil Analisis

Dalam implementasi dan pembahasan program ini akan dibahas mengenai tampilan dan fungsi pada halaman menu utama, *practice mode*, *game mode*, *information of the app*, serta *pop up message*.

#### 3.3.1 Menu Utama



Gambar 5. Tampilan menu utama saat aplikasi pertama kali dijalankan

#### 3.3.2 Practice Mode

Halaman *practice mode* berisi pola delapan titik berwarna hijau yang dirangkai melingkar sejauh 360° dan digunakan sebagai acuan gerakan sandi Semaphore. Kemudian terdapat pula tombol “reset” yang digunakan untuk menghapus semua hasil deteksi dan terjemahan yang telah dilakukan serta tombol “exit” untuk kembali ke menu utama.

Pada bagian ini juga terdapat tampilan bendera kuning-merah yang secara otomatis tersambung pada ujung tangan pengguna. Selanjutnya saat pengguna telah melakukan gerakan-gerakan atau perintah, maka Kinect akan merespon dan hasilnya akan ditampilkan pada layar seperti pada gambar 6 di bawah ini.



Gambar 6. Aplikasi merespon perintah gerakan yang dilakukan

#### 3.3.3 Game Mode

Pada halaman game mode akan muncul huruf A sampai Z yang ditampilkan secara acak. Selanjutnya tugas pengguna adalah menebak gerakan sandi Semaphore dari huruf-huruf acak tersebut. Jika gerakan pengguna benar, maka poin akan bertambah pada bagian “score” yang terdapat di sebelah kanan layar.



Gambar 7. Poin bertambah saat melakukan gerakan yang benar

### 3.3.4 Pop up Message



Gambar 8. Pop up message

Halaman *pop up message* akan muncul ketika waktu permainan yang telah ditentukan habis, maka permainan telah selesai. Selanjutnya halaman ini akan menampilkan jumlah poin total yang telah dikumpulkan oleh pengguna selama permainan berlangsung. *Pop up message* telah diprogram tertutup secara otomatis setelah muncul selama lima detik, jadi pengguna tidak perlu menekan tombol untuk menutupnya. Setelah halaman ini ditutup, maka pengguna akan diarahkan kembali ke halaman *game mode* untuk memulai permainan baru.

### 3.3.5 Information of the App



Gambar 9. Video tutorial penggunaan aplikasi

Halaman *information of the app* akan diisi dengan video tutorial penggunaan aplikasi sistem penerjemah sandi *Semaphore* ini. Setelah mempelajari video tutorial, pengguna dapat menekan tombol "close" kemudian halaman akan berpindah kembali ke menu utama.

## 4. Kesimpulan

1. Aplikasi sistem penerjemah gerakan sandi *Semaphore* virtual ini dirancang dengan menggunakan sistem *boolean* untuk menentukan hasil terjemahan dari gerakan yang dilakukan, serta penerapan konsep sistem koordinat kartesius dalam menentukan pola titik acuan pada layar. *Output* atau hasil keluaran dari aplikasi ini adalah dapat menerjemahkan 36 ragam gerakan sandi *Semaphore*, yakni huruf A-Z serta angka 0-9.
2. Dalam aplikasi ini juga terdapat menu sistem pengujian yang bisa digunakan oleh pengguna untuk mengukur

sejauh mana kemampuan mengingat ragam gerakan sandi *Semaphore* yang dipelajarinya. Fitur ini dibuat dalam bentuk permainan interaktif sehingga dapat memberikan kesan belajar yang lebih menyenangkan kepada penggunanya.

3. Aplikasi sistem penerjemah sandi *Semaphore* ini dapat digunakan pada ruangan dengan intensitas cahaya yang normal untuk memperoleh tingkat keakuratan tinggi serta untuk mendapatkan kenyamanan saat menggunakan aplikasi. Intensitas cahaya yang terlalu tinggi dapat mengganggu sensor pada kamera Kinect sehingga dapat mengganggu kinerjanya. Jarak antara sensor Kinect dengan pengguna juga perlu diperhatikan. Jarak yang disarankan adalah antara 1-2m di depan kamera.
4. Aplikasi ini dirancang dengan memperhatikan konsep desain *interface* agar lebih *user-friendly* sehingga mudah digunakan dan dipelajari serta menambah kesan menyenangkan saat mengoperasikannya.

## REFERENSI

- [1] Trianto Juliatmojo, E. A., 2013, Pembelajaran Sandi Morse dan Sandi Semaphore dalam bentuk simulasi berbasis multimedia. *Jurnal Sarjana Teknik Informatika Volume 1 Nomor 1, Juni 2013* Yogyakarta
- [2] Booth, P. A., 1989., *An Introduction to Human-computer Interaction*. Lawrence Erlbaum Associates Ltd.
- [3] Openkinect community. <http://openkinect.org>, June 2012.
- [4] Lee, H.W. etc., 2015. Kinect Who's Coming - Applying Kinect to Human Body Height Measurement to Improve Character Recognition Performance *Smart Science Vol. 3, No. 2*,
- [5] Scouting Resources UK. <http://scoutingresources.org.uk>, 2009

**Abdi Hutama**, merupakan Mahasiswa Sarjana Matematika dari Universitas Padjadjaran, Indonesia tahun 2009.

**Ino Suryana**, memperoleh gelar Drs. Dari Matematika Universitas. Padjadjaran, tahun 1986. Tahun 1998 memperoleh gelar M.Kom dari Ilmu Komputer Universitas Indonesia. Saat ini sebagai Staf Pengajar Program Studi Teknik Informatika Universitas Padjadjaran.

**Erick Paulus**, memperoleh gelar .S.Si dari Matematika Universitas Padjadjaran, tahun 2004. Kemudian tahun 2006 memperoleh gelar M.Kom dari Ilmu Komputer Universitas Gadjah Mada,. Saat ini sebagai Staf Pengajar Program studi Teknik Informatika Universitas Padjadjaran.

**Betty Subartini**, memperoleh gelar Dra. dari Matematika Universitas Padjadjaran, tahun 1987. Tahun 2008 memperoleh gelar M.Si. dari Matematika Institut Teknologi Bandung. Saat ini sebagai Staf Pengajar Program Studi Matematika Universitas Padjadjaran