

SISTEM PENDETEKSI KANTUK PADA MATA PENGENDARA RODA EMPAT DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK

Michael Yan Petra¹, Lina²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia

E-mail: ¹michael.535189102@stu.untar.ac.id, ²lina@fti.untar.ac.id

Abstrak

Mengantuk merupakan sifat alamiah manusia dikarenakan mengantuk menandakan manusia membutuhkan waktu istirahat yang cukup, namun mengantuk pada waktu dan tempat yang tidak tepat sangat dapat menjadi sumber kecelakaan, salah satunya mengantuk pada saat berkendara. Karena itu dirancang sebuah sistem untuk mendeteksi kantuk dengan cara mendeteksi mata manusia dengan model YOLO yang akan diklasifikasi dengan model CNN apakah kedua mata pengguna sedang tertutup atau terbuka, jika kedua mata tertutup maka akan mengeluarkan output berupa alarm yang diharapkan membangunkan pengguna yang terdeteksi mengantuk. Hasil kalkulasi AP terhadap model pendeteksian dengan akurasi sebesar 100% dan hasil pengujian confusion matrix terhadap model klasifikasi memiliki akurasi sebesar 96% pada nilai presisi, 97% pada nilai recall, dan 97% pada F1-Score

Kata kunci— Convolutional Neural Network, Deteksi Kantuk, Nano Jetson, Single Board Computer, You Only Look Once

Abstract

Drowsiness is human nature because sleepiness indicates that humans need adequate rest time, but sleepiness at the wrong time and place can be a source of accidents, one of which is drowsiness while driving. Because it is designed a system to detect sleepiness by detecting the human eye with the YOLO model which will be classified by the CNN model whether both the user's eyes are closed or open, both eyes are closed, it will issue an output if it is expected to wake the detected user. The AP calculation results on the detection model with an accuracy of 100% and the results of the confusion matrix testing on the classification model have an accuracy of 96% on the precision value, 97% on the recall value, and 97% on the F1-Score

Keywords— Convolutional Neural Network, Sleep Detection, Nano Jetson, Single Board Computer, You Only Look Once

1. PENDAHULUAN

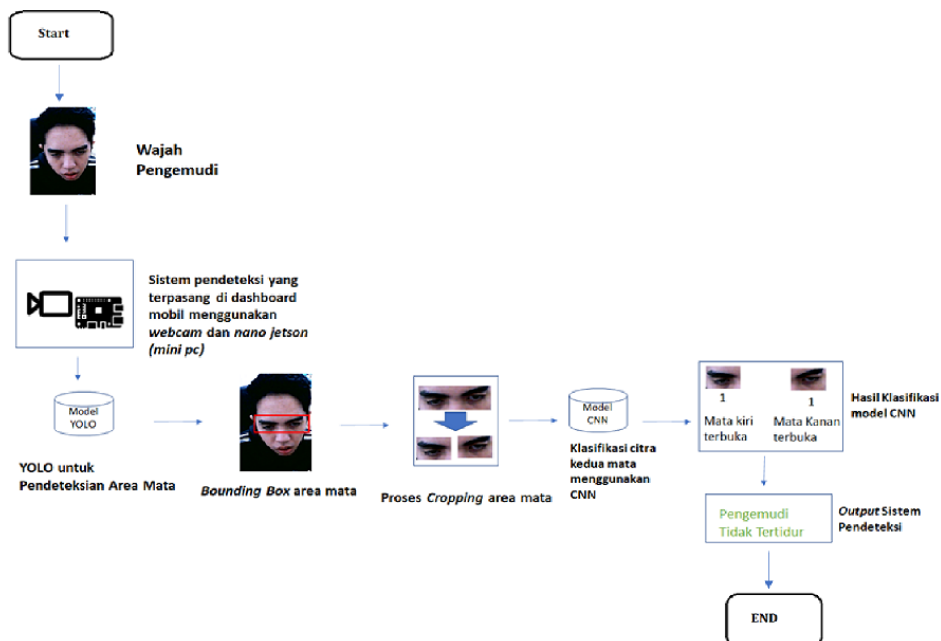
Kantuk merupakan hal serius yang harus ditangani dalam berkendara. Satu dari lima kecelakaan maut melibatkan pengemudi yang mengantuk. pada saat berkendara, mengantuk dapat menyebabkan kejadian baik dari kerugian materi, korban luka ringan hingga berat sampai yang paling fatal yaitu meninggal dunia.

Kantuk pada berkendara dapat disebabkan oleh kurangnya kadar oksigen pada tubuh, dengan tetap bergerak aktif, oksigen yang dibutuhkan otak, pembuluh darah, serta otot dapat kembali tercukupi. Rasa kantuk di malam hari juga merupakan tanda alami dari tubuh setiap makhluk hidup, tanda tersebut mengartikan bahwa otak sudah mulai lelah dan perlu beristirahat. Jika ini terjadi, sebaiknya pengemudi diharapkan telah berhenti sebelum melaksanakan perjalanannya. Menurut hasil penelitian dari AAA Foundation for Traffic Safety menemukan bahwa pengemudi dengan waktu tidur kurang dari lima jam mirip dengan pengemudi yang sedang mabuk, pengemudi yang mengantuk mengakui tidak sadar saat tertidur [1].

Dibangun sebuah sistem yang ditujukan untuk mendeteksi kantuk melalui mata pengemudi, sehingga pada saat terdeteksi bahwa si pengemudi mengantuk/tidur, alarm berbunyi sesuai dengan lama pengemudi tersebut mengantuk. Sistem ini yang kemudian akan ditempatkan kedalam sebuah Single Board Computer yaitu Nvidia Nano Jetson 4GB. Diharapkan sistem ini juga dapat secara bebas dipergunakan untuk hal kegiatan lainnya seperti menghilangkan rasa kantuk pada saat belajar, saat kegiatan belajar daring online, dan lain sebagainya.

2. METODE PENELITIAN

Sistem yang dibangun akan menerima input dari citra yang diambil secara *realtime* melalui camera bertipe CSI (Camera Serial Interface), lalu dari citra akan dilakukan pendeteksian sepasang mata manusia menggunakan algoritma You Only Look Once (YOLO), kemudian output dari YOLO yang merupakan koordinat bounding box akan digunakan untuk dilakukan pemotongan citra di dalam bounding box, citra yang dipotong akan diklasifikasikan kedalam model Convolutional Neural Network (CNN) yang menerima input berupa citra mata kiri dan mata kanan, jika kedua mata terdeteksi tertutup lebih dari 1 detik, maka sebuah alarm akan berbunyi. Adapun diagram alur yang dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Sistem Pendeteksi Kantuk

2.1. Dataset

Pada perancangan ini, dikumpulkan beberapa dataset dari berbagai sumber yang dimana untuk model YOLO menggunakan dataset berbeda dengan model CNN. Adapun dataset yang tertera dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1. Dataset untuk mendeteksi mata dengan model YOLO

<i>Training / Testing</i>			
Dataset	Total	Data Latih 80%	Data Uji 20%
<i>Kaggle face dataset</i>	1038	831	207
<i>Validation</i>			
Dataset	Total		
Dataset Evaluasi 1	65		
Dataset Evaluasi 2	61		

Tabel 2. Dataset untuk klasifikasi mata dengan model CNN

<i>Training / Testing</i>					
Dataset	Data mata terbuka	Data mata tertutup	Total	Data Latih 80%	Data Uji 20%
<i>MRL Eye Dataset</i>	12.947	12.947	25.894	20716	5178
<i>Personal Dataset</i>	18.666	23.537	42.203	33763	8440
<i>Dataset Gabungan</i>	36.484	31.613	68.097	54479	13618
<i>Validation</i>					
Dataset	Data mata terbuka		Data mata tertutup		
<i>Dataset Evaluasi 3</i>	3040		858		

Adapun penjelasan terhadap jenis dataset yakni :

1. MRL Eye Dataset : diambil dari website resmi Media Research Lab, terdiri atas citra mata terbuka dan tertutup dalam tipe gambar grayscale dengan cahaya yang cukup.
2. Personal dataset : diambil secara pribadi melalui webcam komputer, terdiri atas mata tanpa kacamata dan mata yang berkacamata, diambil dalam kondisi cukup cahaya dan sedikit cahaya. Diambil dengan cara menggunakan model YOLO yang telah dibuat, setiap mata yang terdeteksi kemudian dipotong untuk dijadikan dataset.
3. Kaggle Face Dataset : diambil dari website kaggle, terdiri atas 1038 citra berupa wajah manusia yang berbeda-beda.
4. Dataset evaluasi 1: diambil secara pribadi menggunakan CSI Camera yang terpasang dalam nano jetson yang berlokasi di dalam mobil dengan subjek citra wajah seorang pria dengan cahaya yang cukup berupa video berdurasi 1 menit 4 detik, kemudian dipecah menjadi 1 frame per detiknya, sehingga menghasilkan 65 frame atau 65 citra
5. Dataset evaluasi 2: diambil secara pribadi menggunakan CSI Camera yang terpasang dalam nano jetson yang berlokasi di dalam ruangan dengan subjek citra wajah seorang

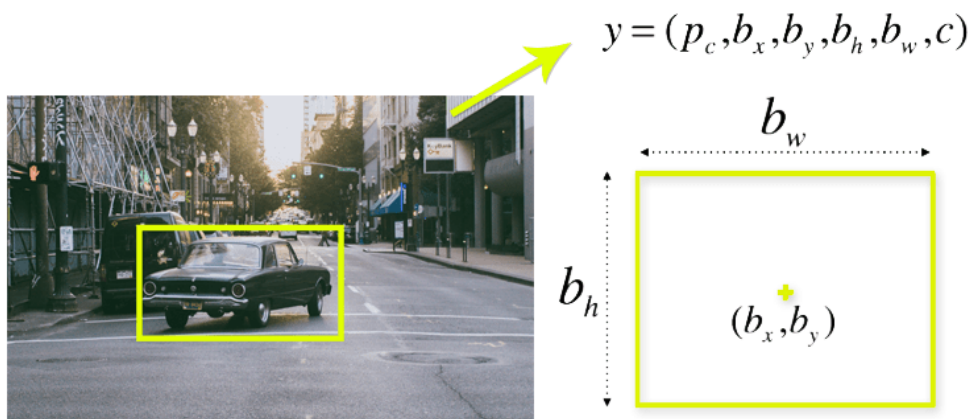
wanita dengan cahaya yang cukup berupa video berdurasi 1 menit, kemudian dipecah menjadi 1 frame per detiknya, sehingga menghasilkan 61 frame atau 61 citra

6. Dataset evaluasi 3: diambil secara pribadi menggunakan CSI Camera yang terpasang dalam nano jetson yang berlokasi di dalam mobil dan dilakukan pemotongan atas bounding box yang dihasilkan oleh model yolo sehingga didapatkan dataset berupa citra mata kiri dan mata kanan.

2.1. You Only Look Once

YOLO (You Only Look Once) pertama kali diciptakan oleh Joseph Redmon pada tahun 2015, merupakan algoritma untuk mendeteksi objek pada gambar dengan melihat keseluruhan gambar satu kali. Algoritma YOLO bekerja dengan adanya teknik Residual Blocks, Bounding box regression dan Intersection Over Union (IOU).

Pada bagian Residual Blocks, YOLO membagi sebuah citra / gambar dan video menjadi bentuk kisi (grids) sebesar $S \times S$. Dimana setiap sel (cell) pada grids bertanggung jawab atas pendeteksian 5 atribut bounding box yakni Box Width, Box Height, Class, dan Bx dan By, Sebagai contoh dapat dilihat pada Gambar 2



Gambar 2 . Atribut bounding Box pada algoritma You Only Look Once (YOLO) (Sumber: Jędrzej Świeżewski, YOLO Algorithm and YOLO Object Detection: An Introduction, <https://appsilon.com/object-detection-yolo-algorithm/>, 28 Agustus 2021)

Arsitektur yang dipakai dalam sistem ini merupakan arsitektur Tiny-YOLOv3 yang dimana versi Tiny mengusung kecepatan dalam memproses sebuah gambar [2], model tersebut kemudian di konversi kedalam format TensorRT, TensorRT merupakan sebuah software development kit yang dibangun oleh Nvidia dengan tujuan utama yaitu mengoptimalkan sebuah model agar mempercepat inference time sebuah model [3], penulis menggunakan git berikut untuk mengkonversi tiny-yolo darknet kedalam format tensorrt : https://github.com/jkjung-avt/tensorrt_demos

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis jaringan syaraf tiruan yang biasa digunakan pada suatu data berupa gambar (citra). Secara garis besar, CNN tidak jauh berbeda dengan jaringan syaraf tiruan lainnya. CNN terdiri atas neuron, weight, bias, dan activation function, dan yang paling utama yaitu memiliki sebuah layer yaitu Convolutional Layer. CNN dipergunakan untuk mengenali pattern yang ada pada input yang biasanya digunakan untuk pengenalan / klasifikasi. Arsitektur yang dipakai pada sistem ini dapat dilihat pada Tabel 3.

Tabel 3. Arsitektur LeNet-5 modifikasi

Layer		Feature Map	Size	Kernel Size	Stride	Activation	Regularizer
Input	Image	1	32x32	-	-	-	-
1	Convolution	6	28x28	5x5	1	Relu	-
2	Max Pooling	6	14x14	2x2	2	-	-
3	Convolution	16	10x10	5x5	1	Relu	-
4	Max Pooling	16	5x5	2x2	2	-	-
5	FC	120	1x1	5x5	1	Relu	L1(0.001), L2(0.001)
6	Dropout (0.5)	-	-	-	-	-	-
7	FC	-	84	-	-	Relu	L1(0.001), L2(0.001)
8	Dropout (0.5)	-	-	-	-	-	-
9	FC	-	1	-	-	Sigmoid	-

Setelah dilatih dengan hyperparameter yaitu epoch sebesar 50, 100, dan 200, dan dengan dataset pribadi, dataset MRL, dan dataset gabungan.

2.2 Metode Pengujian

Ada beberapa metode pengujian yang dilakukan untuk mendapatkan model yang terbaik dengan beberapa metrics terkenal yaitu accuracy, precision, recall, dan f1-score, Intersection Over Union, Average Precision.

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \quad (1)$$

Accuracy baik digunakan jika data uji antar kelas memiliki jumlah yang tidak terlalu berbeda jauh. *Accuracy* bukan tolak ukur yang baik jika jumlah data uji antar kelas terlalu berbeda jauh.

Precision dan *Recall* merupakan sepasang tolak ukur untuk memberi support terhadap hasil *accuracy*. *Precision* merupakan tolak ukur terhadap seberapa baik model dalam memprediksi data benar positif (TP) diantara keseluruhan data yang terprediksi positif. *Recall* merupakan tolak ukur terhadap seberapa akurat model dalam memprediksi data benar positif diantara keseluruhan data yang benar positif. *Precision* dan *Recall* dihitung menggunakan rumus berikut :

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-Score merupakan tolak ukur yang lebih baik ketimbang *Accuracy* jika ingin melihat keseimbangan antara *precision* dan *recall* pada distribusi kelas yang tidak merata. *F1-Score* sendiri merupakan perbandingan yang dibobotkan antara rata-rata *precision* dan *recall*. Adapun rumus *F1-Score* sebagai berikut :

$$F1 - score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \quad (4)$$

Digunakan pula teknik Intersection Over Union yang digunakan untuk mengukur overlapping antara 2 bounding box antara hasil prediksi model dan ground truth. Adapun rumus IoU yakni :

$$IoU = \frac{\text{Area}(BB \text{ prediksi } \cap BB \text{ Ground truth})}{\text{Area}(BB \text{ Prediksi } \cup BB \text{ Ground truth})} \quad (5)$$

Jika nilai IoU diatas 0,5, maka hasil akan menjadi TP, dan sebaliknya, jika dibawah maka hasil akan menjadi TN.

Average Precision adalah ukuran yang menggabungkan recall dan precision. Untuk satu Adapun rumus daripada Average Precision adalah sebagai berikut :

$$\text{Average Precision} = \int_0^1 p(r)dr \quad (6)$$

dalam evaluasi menggunakan metode AP, penulis menggunakan git berikut : <https://github.com/Cartucho/mAP>

3. HASIL DAN PEMBAHASAN

Dilakukan pengujian terhadap dua model yaitu YOLO dan CNN, untuk model YOLO dilakukan teknik perhitungan Average Precision (AP) dengan IoU sebesar 50 terhadap data evaluasi 1 dan data evaluasi 2 dengan hasilnya didapatkan sebesar 1 yang merupakan nilai tertinggi dari perhitungan AP, contoh sample pengujian dapat dilihat pada Table 4

Tabel 4. Sample pengujian dataset evaluasi

Image	IoU	Result
	81.89%	Match
	67.55%	Match

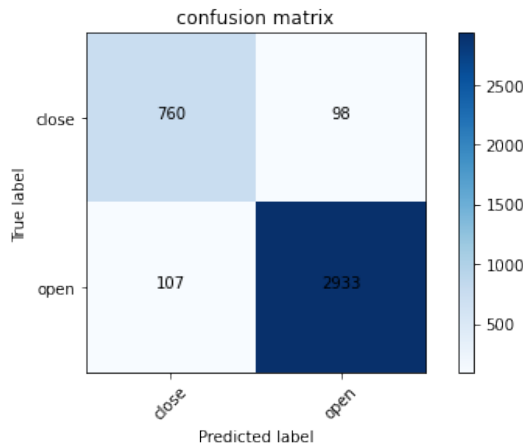
Michael Yan Petra: Sistem Pendeteksi Kantuk pada Mata Pengendara Roda Empat dengan Algoritma Convolutional Neural Network

Didapatkan hasil yang paling maksimu yaitu pelatihan terhadap dataset pribadi dengan epoch 200 dan batch size sebesar 64. Yang dapat dilihat pada Tabel 5 dan didapatkan model 14 sebagai model yang baik.

Tabel 5. Sample pengujian

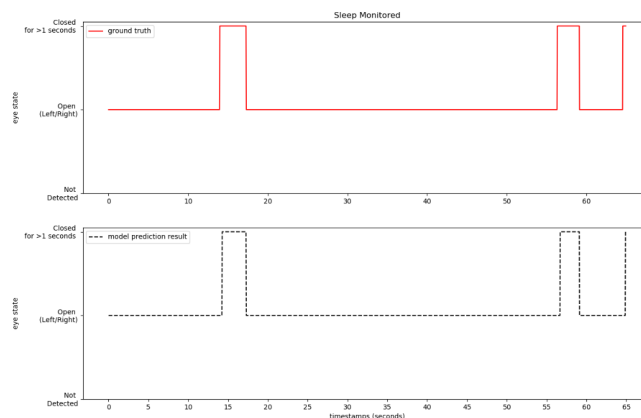
No	Dataset	Epoch	Precision		Recall		F1-Score		Accuracy
			Open	Close	Open	Close	Open	Close	
11	Personal Dataset	50	0.96	0.98	0.99	0.89	0.98	0.92	96%
13		100	0.96	0.97	0.99	0.91	0.98	0.93	97%
14		200	0.99	0.94	0.98	0.97	0.99	0.95	98%
12	Dataset Gabungan	50	0.97	0.93	0.98	0.90	0.98	0.82	98%
15		100	0.97	0.88	0.96	0.89	0.97	0.88	96%
16		200	0.97	0.83	0.95	0.91	0.96	0.87	94%

Adapun confusion matrix yang dapat dilihat untuk model 14 dapat dilihat pada Gambar 3

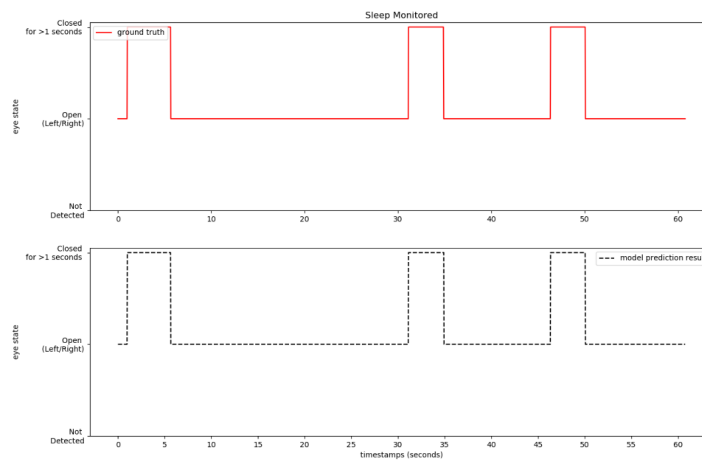


Gambar 3 . Confusion Matrix model akhir pengklasifikasian.

Kemudian kedua model digabungkan dan di evaluasi terhadap dataset evaluasi 1 dan 2 dan didapatkan bahwa akurasi sebesar 100% yang dapat dilihat pada Gambar 4 dan Gambar 5



Gambar 4. Grafik Evaluasi model pendeteksian kantuk terhadap dataset evaluasi 1



Gambar 5. Grafik Evaluasi model pendeteksian kantuk terhadap dataset evaluasi 2

4. KESIMPULAN

Ditarik beberapa poin untuk dijadikan kesimpulan yaitu sebagai berikut :

1. Model pendeteksian mata menggunakan algoritma YOLO dengan variasi tiny-YOLO dapat berjalan dengan baik pada sebuah single board computer Nvidia Nano Jetson 4GB dan akurasi yang dihasilkan sangat baik untuk mendeteksi mata manusia dari sebuah Camera Serial Interface (CSI) yang sebelumnya telah dilakukan pelatihan sebanyak 2000 epoch
2. Model klasifikasi mata menggunakan algoritma CNN dengan arsitektur LeNet-5 dan beberapa konfigurasi telah dilakukan yakni teknik dropout dan weight regularization dan didapatkan akurasi yang cukup baik untuk pengklasifikasian.
3. Sistem pendeteksian kantuk yang dibangun ini dapat menerima input secara realtime yang diambil dari sebuah camera bertipe CSI kemudian dikeluarkan output berupa sebuah alarm yang mendakan jika pengguna terdeteksi telah memejamkan matanya selama lebih dari 1 detik.

Terdapat beberapa saran yang diberikan berdasarkan hasil perancangan dan pengujian sistem pendeteksi kantuk agar lebih baik yakni :

1. Untuk pengembangan kedepannya, karena kurangnya dataset sepasang mata tertutup dari internet, maka perlunya dilakukan pengambilan secara manual dan beragam agar model YOLO dapat mendeteksi mata manusia dengan tingkat confidence yang tinggi.
2. Dikarenakan harga perangkat tergolong mahal, maka kedepannya dapat dilakukan pengujian terhadap perangkat SBC yang lebih murah seperti Raspberry Pi.
3. Pada perancangan ini, model dibatasi mendeteksi dalam kondisi cahaya yang cukup, untuk kedepannya disarankan penggunaan alat yang dapat mendukung dalam kondisi cahaya yang kurang seperti pemasangan modul infrared pada camera.
4. Dapat menggunakan sumber daya yang lebih dari 5 volt , disarankan 10 volt dengan tidak menggunakan micro usb namun menggunakan port barrel jack untuk mendapatkan daya yang maksimal.

5. Untuk meningkatkan performa dan akurasi, kedepannya dapat dilakukan penambahan variasi dataset terhadap target kelas yang dituju, seperti citra mata yang tertutup harus seimbang dengan citra mata terbuka, menggunakan variasi arsitektur lain.

DAFTAR PUSTAKA

- [1] Owens, J. M., Dingus, T. A., Guo, F., Fang, Y., Perez, M., McClafferty, J., Tefft, B. Prevalence of Drowsy Driving Crashes: Estimates from a Large-Scale Naturalistic Driving Study. (Research Brief.) Washington, D.C.: AAA Foundation for Traffic Safety. February, 2018.
- [2] Pranav, A. Pratibha, R., Manoj, K., 2000 YOLO v3-Tiny: Object Detection and Recognition using one stage improved model, *2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS)*, hal. 691
- [3] Ildar, R. Increasing FPS for single board computers and embedded computers in 2021 (Jetson nano and YOVOv4-tiny). Practice and review, *Journal of National Institutes of Health*, Hal 4.