



# JURNAL IPTEK

## MEDIA KOMUNIKASI TEKNOLOGI

homepage URL : [ejurnal.itats.ac.id/index.php/iptek](http://ejurnal.itats.ac.id/index.php/iptek)



### Komputasi Penentuan Kualitas Pada Serat optik Berdasarkan Rugi Daya Dengan Gaussian Naïve Bayes Menggunakan Teknologi CUDA

Danang Haryo Sulaksono<sup>1</sup> dan Aslam Chitami Priawan Siregar<sup>2</sup>

Jurusan Teknik Informatika, FTIF, Institut Teknologi Adhi Tama Surabaya<sup>1,2</sup>

#### INFORMASI ARTIKEL

Jurnal IPTEK – Volume 22  
Nomer 2, Desember 2018

Halaman:  
35 – 42  
Tanggal Terbit :  
20 Desember 2018

DOI:  
10.31284/j.iptek.2018  
.v22i2.322

#### EMAIL

danang\_h\_s@itats.ac.id  
aslam.chitami@itats.ac.id

#### PENERBIT

LPPM- Institut Teknologi  
Adhi Tama Surabaya  
Alamat:  
Jl. Arief Rachman Hakim  
No.100,Surabaya 60117,  
Telp/Fax: 031-5997244

*Jurnal IPTEK by LPPM-  
ITATS is licensed under a  
Creative Commons  
Attribution-ShareAlike 4.0  
International License.*

#### ABSTRACT

Power loss is one aspect that can determine the quality of an optical fiber. In this study, SMS-based power losses are classified based on changes from the value of power losses based on shifts and divided into three classes: good, average, and bad. The division of this class is based on the magnitude of changes in each value on the optical fiber. In this classification, ten shift variables are used between 0–450  $\mu\text{m}$  with a shift interval of 50  $\mu\text{m}$ . This study uses optical fiber with a diameter of 5.5 cm. This study uses the Naïve Bayes Gaussian method to classify the existing power loss training data. The computational process of data classification in this study uses CUDA technology, a computing parallelization platform between the CPU and GPU. Classification results still show values below 50%, with a value of 42.86% so that the boosting method is still needed for pre-processing to get data values that are not imbalanced, while for CUDA markers itself shows very fast computational values between 600–1,000 ms.

**Kata kunci:** Fiber optic; Power loss; Gaussian Naïve Bayes; CUDA; Shifting

#### ABSTRAK

Rugi daya adalah salah satu aspek yang dapat menentukan kualitas dari sebuah serat optik. Dalam penelitian ini, rugi pada serat optik diklasifikasikan berdasarkan perubahan nilai dari rugi daya berdasarkan pergeseran dan dibagi menjadi tiga kelas: bagus, sedang, dan jelek. Pembagian kelas ini berdasarkan besaran perubahan pada setiap nilai pada serat optik. Dalam klasifikasi ini juga digunakan sepuluh variabel pergeseran antara 0–450  $\mu\text{m}$  dengan interval pergeseran sebesar 50  $\mu\text{m}$ . Penelitian ini menggunakan serat optik dengan diameter 5,5 cm. Penelitian ini menggunakan metode Gaussian Naïve Bayes untuk melakukan klasifikasi terhadap data training rugi daya yang ada. Proses komputasi pada klasifikasi data dalam penelitian ini menggunakan teknologi CUDA, sebuah platform paralelisasi komputasi antara CPU dan GPU. Nilai hasil klasifikasi masih menunjukkan nilai di bawah 50%, yaitu dengan nilai 42,86%. Sehingga, masih dibutuhkan metode *boosting* untuk *pre-processing* guna mendapatkan nilai data yang tidak *imbalanced* sedangkan hasil *benchmark* CUDA sendiri menunjukkan nilai rentang komputasi yang sangat cepat, antara 600–1.000 ms.

**Kata kunci:** Serat optik; Rugi daya; Gaussian Naïve Bayes; Cuda; Pergeseran

#### PENDAHULUAN

Sebuah serat optik adalah sebuah fiber yang fleksibel, transparan, dan dibuat dengan membuat diameter material kaca (bahan silika) atau plastik menjadi lebih kecil dari rambut manusia. Ada beberapa jenis serat optik, salah satunya adalah yang strukturnya mempunyai dua fiber *single mode* (SM) yang secara aksial disambungkan pada ujung dari sebuah inti *parabolic multimode* (MM) serat optik. Serat optik dengan struktur ini disebut juga dengan SMS (*single*

*mode-multimode-single mode*) [1]. Kualitas kelayakan serat optik salah satunya rugi daya. Rugi daya dapat terjadi dari beberapa sebab, seperti adanya kebocoran atau kurangnya kecepatan ketika proses penghantaran cahaya. Perubahan rugi daya yang baik adalah kontinu. Dengan kata lain, nilai rugi daya naik-turun akan mengganggu serat optik untuk melakukan komunikasi dari pangkal ke ujung.

Kualitas serat optik terhadap rugi daya dapat dikelompokkan dengan salah satu cabang *data mining* yaitu klasifikasi. Ada beberapa metode yang dapat dipakai dalam cabang ini, yaitu Naïve Bayes, Random Forest, C45, dan lain sebagainya. Dalam penelitian ini, klasifikasi kualitas serat optik berdasarkan rugi daya menggunakan metode Gaussian Naïve Bayes. Distribusi Gaussian yang berada dalam Naïve Bayes digunakan karena data yang dipakai berupa data numerik kontinu.

Penelitian ini menggunakan rugi daya berdasarkan pergeseran pada serat optik berstruktur SMS sebagai fitur klasifikasi untuk penentuan kualitas serat optik diameter 5,5 cm dengan metode Gaussian Naïve Bayes. Hasil akuisisi data diukur dengan variabel yaitu 0 µm, 50 µm, 100 µm, 150 µm, 200 µm, 250 µm, 300 µm, 350 µm, 400 µm, dan 450 µm. Variabel ini didapatkan dari pergeseran serat optik. Kelas penentuan klasifikasi dibagi menjadi tiga bagian yaitu bagus, sedang, dan jelek. Kelas ini didapatkan dari nilai perubahan rugi daya dalam serat optik karena perubahan naik-turun atau kenaikan signifikan rugi daya akan berpengaruh pada kualitas penghantar pada serat optik. Hasil pengujian didapatkan dengan bantuan *confusion matrix* untuk mengukur akurasi dari klasifikasi.

Sistem pengambilan keputusan membutuhkan kalkulasi. Sehingga, ketika *hardware* dihadapkan pada data yang digunakan, efisiensi komputasi dapat menurun. Oleh karena itu, dibutuhkan sebuah sistem yang dapat meningkatkan kerja proses klasifikasi. Dalam penelitian ini, dilakukan paralelisasi untuk proses klasifikasi metode Naïve Bayes dengan memanfaatkan teknologi CUDA (*Compute Unified Device Architecture*). CUDA adalah sebuah platform yang memungkinkan sebuah GPU (*Graphic Proccessing Unit*) NVIDIA memproses instruksi layaknya CPU (*Central Proccessing Unit*) dengan menggunakan bahasa tingkat tinggi.

## TINJAUAN PUSTAKA

### Naïve Bayes Classifier

Naïve Bayes adalah sebuah klasifikasi berdasarkan klasifikasi yang mengalkulasi sebuah set probabilitas dengan menambahkan frekuensi dan kombinasi dari nilai-nilai sebuah *dataset*. Algoritme ini menggunakan teorema Bayes dan mengasumsikan bahwa semua tribut bebas dan tidak bebas akan diberikan oleh nilai-nilai dalam kelas variabel. Pada definisi yang lain, Naïve Bayes adalah sebuah klasifikasi dengan probabilitas dan metode statistik yang memprediksi kemungkinan di masa depan berdasarkan *dataset*.

Naïve Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut bebas bersyarat ketika diberikan nilai *output*. Dengan kata lain, ketika diberikan nilai *output*, mengalkulasi nilai produk dari probabilitas individu. Keuntungan menggunakan Naïve Bayes adalah bahwa metode ini hanya membutuhkan sejumlah kecil data *training* untuk menentukan perkiraan parameter yang diperlukan dalam proses klasifikasi [2]. Rumus Naïve Bayes dapat dilihat pada Persamaan (1).

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \dots (1)$$

$p(x|C_k)$  atau dapat juga disebut dengan probabilitas *posterior*, adalah sebuah probabilitas dari sebuah hipotesis  $C_k$  yang berdasarkan kondisi  $x$ .  $p(C_k)$  juga disebut dengan *prior*, adalah sebuah probabilitas dari sebuah hipotesis dalam kelas  $C_k$ . Dalam penelitian ini, *prior* menggunakan tiga kelas untuk menghitung nilai dari hasil rugi daya serat optik, yaitu bagus, sedang, dan jelek.  $p(x|C_k)$  dapat juga disebut dengan *likelihood*. Nilai *likelihood* merupakan hasil dari vektor antara semua variabel pada setiap kelas. Penelitian ini menggunakan sepuluh variabel, semuanya adalah nilai dari pergeseran serat optik, yaitu 0 µm, 50 µm, 100 µm, 150 µm, 200 µm, 250 µm, 300 µm, 350 µm, 400 µm, dan 450 µm. Sedangkan  $p(x)$  yaitu *marginal likelihood* atau *evidence* yang didapatkan dari perhitungan  $p(C_k)p(x|C_k)$  [3].

Rugi daya yang menjadi fitur penelitian ini menggunakan data numerik kontinu. Diasumsikan bahwa nilai kontinu yang diasosiasikan pada setiap kelas akan didistribusikan dengan menggunakan distribusi Gaussian untuk menentukan nilai *posterior* atau  $p(x|C_k)$  [4] berdasarkan Persamaan (2).

$$p(x = v | C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}} \dots (2)$$

$x$  adalah sebuah set data *training* yang mempunyai nilai kontinu, dalam kasus ini adalah nilai dari rugi daya.  $\mu_k$  adalah nilai data-data dari  $x$  yang terasosiasikan dalam setiap kelas dari  $C_k$ , berdasarkan Persamaan (3).

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \dots (3)$$

$\sigma_k^2$  adalah nilai varians yang menghitung seberapa jauh satu set angka tersebar dari nilai rata-rata. Hasil  $\sigma_k^2$  diperoleh dari nilai  $x$  yang diasosiasikan oleh kelas  $C_k$ . Rumus variansnya dapat dilihat pada Persamaan (4).

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \dots (4)$$

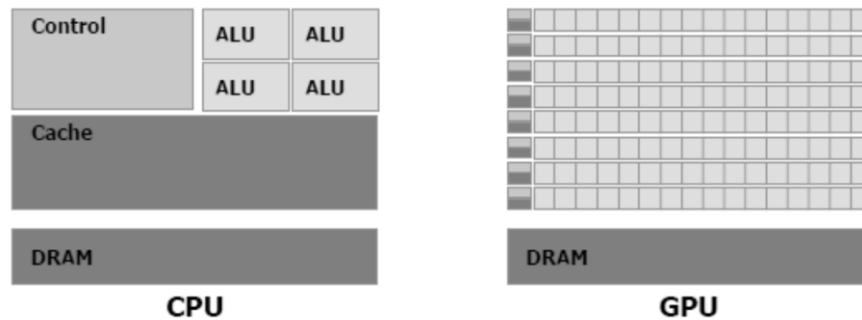
Dalam distribusi Gauss, nilai  $v$  adalah sebuah sampel data yang diujikan terhadap data *training*. Jadi, probabilitas distribusi dari nilai  $x$  dari kelas  $C_k$  adalah  $p(x = v | C_k)$  dan dapat dikalkulasikan dengan memasukkan nilai  $c$  ke dalam sebuah distribusi normal. Nilai  $\mu_k$  dan  $\sigma_k^2$  digunakan sebagai parameter. Dalam penelitian ini, pengujian hasil klasifikasi menggunakan rumus akurasi, yaitu perhitungan untuk menentukan data uji apakah presisi atau benar [5]. Nilai akurasi didapatkan dari hasil nilai yang didapatkan dari *confusion matrix*. Perhitungan akurasi menggunakan 10 data tes yang dihadapkan dengan data *training* berdasarkan Persamaan (5).

$$Accuracy = \frac{True\ Positive}{True\ Positive + False\ Positive + False\ Negative} * 100\% \dots (5)$$

## Teknologi CUDA

Cuda adalah sebuah sistem komputasi paralel CPU-GPU yang dapat menyediakan eksekusi cepat dari sebuah atau beberapa program ketika dibandingkan dengan komputasi CPU-centered. GPU dapat digunakan untuk melakukan komputasi paralel yang memanfaatkan komputasi dari *multiple cores* [6].

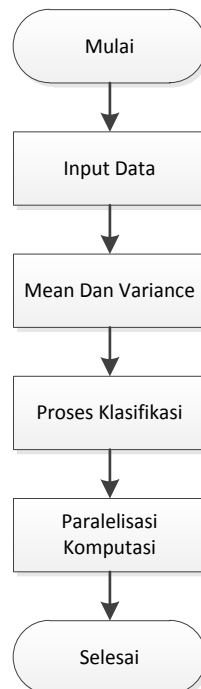
CUDA menggunakan arsitektur *general-purpose parallel computing*, dengan sebuah model pemrograman paralel dan dengan set arsitektur instruksi yang baru, yang memperluas *parallel computing engine* pada GPU NVIDIA. Tidak seperti generasi sebelumnya, komputasi pada GPU dibagi menjadi *vertex* dan *pixel shaders*. Pada arsitektur CUDA, terdapat sebuah *unified shader pipeline* yang memungkinkan tiap *Arithmetic Logic Unit (ALU)* pada *chip* dikendalikan oleh program untuk melakukan *general-purpose computation*, komputasi yang tidak terbatas hanya sekedar pengolahan visual dan grafik. Ilustrasi paralelisasi CPU dan GPU dapat dilihat pada Gambar 1. CUDA dilengkapi dengan sebuah lingkungan perangkat lunak yang memungkinkan pengembang untuk menggunakan bahasa C sebagai bahasa pemrograman tingkat tinggi [7].



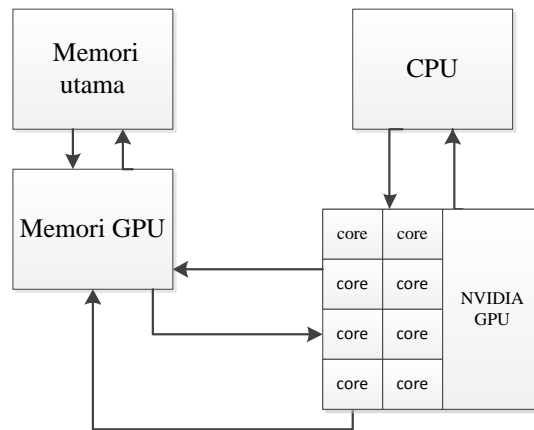
Gambar 1. Paralelisasi CPU dan GPU [8].

## METODE

Metode pada penelitian ini dapat dilihat pada Gambar 2. Nilai yang digunakan untuk menentukan kelayakan serat optik dalam aplikasi ini adalah nilai rugi daya dalam pergeseran data. Nilai ini diambil dari akuisisi data pada serat optik dengan diameter sebesar 5,5 cm. Dengan variabel 0  $\mu\text{m}$ , 50  $\mu\text{m}$ , 100  $\mu\text{m}$ , 150  $\mu\text{m}$ , 200  $\mu\text{m}$ , 250  $\mu\text{m}$ , 300  $\mu\text{m}$ , 350  $\mu\text{m}$ , 400  $\mu\text{m}$ , dan 450  $\mu\text{m}$ . Dari *input* data, akan dicari nilai *mean* dan *variance* untuk digunakan pada proses klasifikasi dalam perhitungan *likelihood*. Proses Naïve Bayes dijalankan dalam sistem paralelisasi dengan teknologi CUDA, yaitu ketika CPU mengirimkan instruksi kepada *core* dalam GPU yang nantinya dikirimkan kepada memori GPU dan lalu ke memori utama dalam VGA (*Video Graphic Array*). Setelah data diproses, maka semua dikembalikan ke CPU untuk dikeluarkan *output*-nya. Dalam GPU yang terimplementasi CUDA, arsitekturnya memungkinkan menjadi arsitektur terbuka seperti layaknya CPU (*Central Processing Unit*). Hanya, tidak seperti CPU, GPU memiliki arsitektur banyak-inti yang paralel. Ilustrasi alur proses paralelisasi antara CPU dan GPU dapat dilihat pada Gambar 3.



Gambar 2. Metode penelitian.



Gambar 3. Flow diagram paralelisasi CPU dan GPU.

## HASIL DAN PEMBAHASAN

### Hasil Akurasi Klasifikasi

Setelah dilakukan perhitungan, hasil dari *mean* dan *variance* dari setiap kelas  $C_k$  ditunjukkan dalam Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1.  $C_k$  = bagus.

	0 $\mu$ m	50 $\mu$ m	100 $\mu$ m	150 $\mu$ m	200 $\mu$ m	250 $\mu$ m	300 $\mu$ m	350 $\mu$ m	400 $\mu$ m	450 $\mu$ m
<i>mean</i>	3,5950	3,6518	3,7279	3,7770	3,8453	3,9178	4,0300	4,0830	4,1380	4,1910
<i>variance</i>	0,0079	0,0113	0,0178	0,0193	0,0354	0,0748	0,1040	0,1242	0,1205	0,1432

Tabel 2.  $C_k$  = sedang.

	0 $\mu$ m	50 $\mu$ m	100 $\mu$ m	150 $\mu$ m	200 $\mu$ m	250 $\mu$ m	300 $\mu$ m	350 $\mu$ m	400 $\mu$ m	450 $\mu$ m
<i>mean</i>	3.5931	3.6070	3.6210	3.6410	3.6665	3.6703	3.6860	3.7060	3.7290	3.7554
<i>variance</i>	0.0492	0.0496	0.0462	0.0490	0.0551	0.0548	0.0568	0.0661	0.0504	0.0379

Tabel 3.  $C_k$  = jelek.

	0 $\mu$ m	50 $\mu$ m	100 $\mu$ m	150 $\mu$ m	200 $\mu$ m	250 $\mu$ m	300 $\mu$ m	350 $\mu$ m	400 $\mu$ m	450 $\mu$ m
<i>mean</i>	3.5390	3.5467	3.5970	3.6876	3.6441	3.6630	3.6306	3.6510	3.6594	3.6670
<i>variance</i>	0.0365	0.0629	0.0737	0.1350	0.0812	0.0388	0.0457	0.0561	0.0517	0.0747

Setelah mendapatkan nilai *mean* dan *variance*, langkah selanjutnya adalah menghitung nilai dari *likelihood*  $p(x|C_k)$  dari setiap kelas menggunakan distribusi Gaussian. Sebuah set data *power loss* disubstitusikan ke dalam variabel  $v$  dalam persamaan distribusi Gauss. Nilai *likelihood* dalam setiap kelas yang diproduksi oleh distribusi Gauss digunakan untuk mengklasifikasikan semua sampel data untuk menghasilkan data tes yang terklasifikasi. Hasilnya dapat dilihat pada Tabel 4.

Tabel 4. Hasil klasifikasi.

0 $\mu$ m	50 $\mu$ m	100 $\mu$ m	150 $\mu$ m	200 $\mu$ m	250 $\mu$ m	300 $\mu$ m	350 $\mu$ m	400 $\mu$ m	450 $\mu$ m	$C_k$	Aktual
3.6471	3.6511	3.6576	3.7214	3.7354	3.7231	3.7395	3.7415	3.7613	3.7781	sedang	sedang
3.6016	3.6014	3.6010	3.7101	3.6705	3.6875	3.6701	3.6821	3.6875	3.7014	sedang	jelek
3.5360	3.5210	3.6209	3.7343	3.7621	3.8302	4.2408	4.3477	4.4015	4.4214	bagus	bagus
3.2053	3.0125	3.3077	3.6281	3.6547	3.5811	3.5454	3.6717	3.6712	3.7641	sedang	jelek
3.6016	3.6015	3.6100	3.7101	3.6705	3.6875	3.6701	3.6832	3.6875	3.7011	sedang	jelek
3.6040	3.6310	3.6450	3.6477	3.7360	3.7401	3.7452	3.7650	3.7634	3.7840	sedang	sedang
3.5704	3.7543	3.8642	3.8789	4.1258	4.3584	4.3698	4.4568	4.4770	4.6320	bagus	bagus
3.5623	3.5661	3.5740	3.6510	3.6320	3.6374	3.7001	3.8001	3.8122	3.8278	jelek	sedang
3.5172	3.6110	3.6210	3.6289	3.7031	3.7358	3.7481	3.7921	3.7970	3.8130	bagus	bagus
3.7127	3.7124	3.7010	3.8212	3.7816	3.7986	3.7812	3.7932	3.7986	3.8124	jelek	jelek

## Cofusion Matrix

Hasil perhitungan *confusion matrix* terdapat dalam Tabel 5.

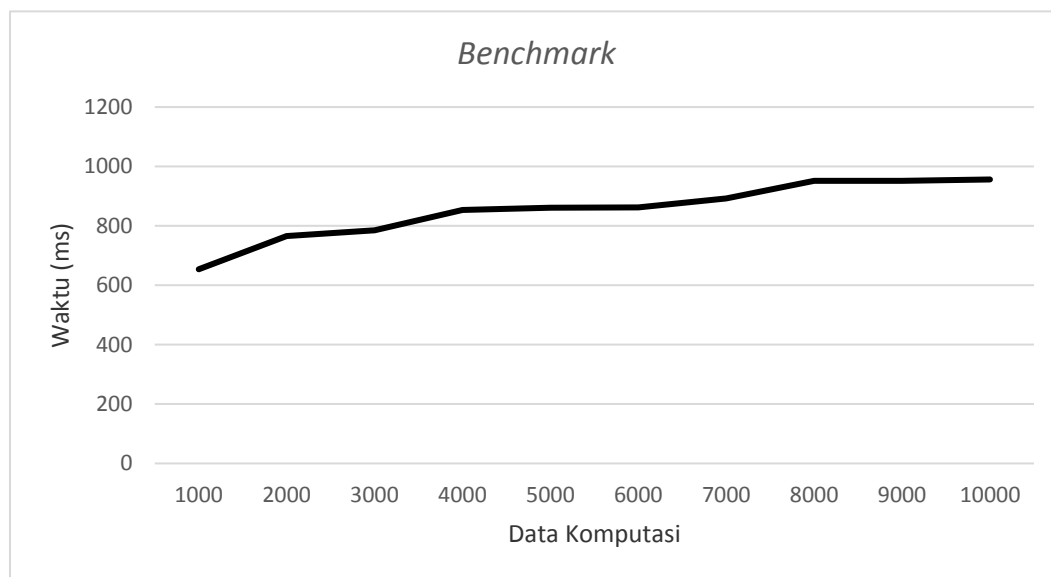
Tabel 5. Hasil *confusion matrix*.

		Prediksi			
		Bagus	Sedang	Jelek	FN
Aktual	Bagus	3	0	0	0
	Sedang	0	2	1	1
	Jelek	0	3	1	3
	FP	0	3	1	

Dari Tabel 5 didapatkan data *True Positive* + *True Negative* = 6, *False Positive* = 6, *False Negative* = 6, maka akurasi dapat dikalkulasikan dan menghasilkan nilai 42,86%.

## Benchmark Komputasi CUDA

Benchmark komputasi CUDA dalam penelitian ini meliputi lamanya waktu GPU dalam melakukan komputasi klasifikasi. Lamanya waktu ditunjukkan dalam ms (*millisecond*). Pengujian dilakukan dengan data secara bertahap untuk mengetahui lamanya GPU melakukan komputasi. Jumlah data berkisar antara 1.000–10.000. Dengan cara ini, beban komputasi dibagi antara *core* yang terdapat dalam GPU.



Gambar 4. Benchmark komputasi GPU.

Pengujian pada setiap kenaikan pemrosesan data yang ada pada Gambar 4 menunjukkan kenaikan komputasi ketika data ditambahkan pada interval kenaikan 1.000 data. Waktu rentang komputasi GPU antara 600–1000 ms. Kecepatan *core* GPU dalam mengalkulasi data lebih baik dengan adanya *virtual multithread*. Gambar 4 juga menunjukkan nilai komputasi yang tidak begitu jauh perbedaannya. Selain itu, dengan adanya *library* dari MPI, GPU lebih mudah membantu dan mendistribusikan data yang akan diolah.

## KESIMPULAN

Nilai hasil klasifikasi masih menunjukkan berada di bawah 50%, yaitu sebesar 42,86%. Sehingga, masih dibutuhkan metode *boosting* untuk *pre-processing* untuk mendapatkan nilai data yang tidak *imbalance*. Hasil *benchmark* CUDA menunjukkan nilai rentang komputasi yang sangat cepat antara 600–1.000 ms.

## DAFTAR PUSTAKA

- [1] A. C. P. Siregar dan D. H. Sulaksono, "Perancangan Sensor Suhu menggunakan Metode Interpolasi Lagrange Berbasis Serat Optik Berstruktur Sms (Singlemode-Multimode-Singlemode)," *JEEE-U J. Electr. Electron. Eng.-UMSIDA*, vol. 1, no. 2, hlm. 34–39, 2017.
- [2] L. Jiang, Z. Cai, H. Zhang, dan D. Wang, "Naive Bayes text classifiers: a locally weighted learning approach," *J. Exp. Theor. Artif. Intell.*, vol. 25, no. 2, hlm. 273–286, 2013.
- [3] B. Bustami, "Penerapan Algoritma Naïve Bayes Untuk Mengklasifikasi Data Nasabah Asuransi," *J. Inform.*, vol. 8, no. 1, 2014.
- [4] A. H. Jahromi dan M. Taheri, "A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features," dalam *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, 2017, hlm. 209–212.
- [5] U. W. Wijayanto dan R. Sarno, "An Experimental Study of Supervised Sentiment Analysis Using Gaussian Naïve Bayes," dalam *2018 International Seminar on Application for Technology of Information and Communication*, 2018, hlm. 476–481.
- [6] T. Mahajan dan S. Masih, "Enhancing Blowfish file encryption algorithm through parallel computing on GPU," dalam *2015 International Conference on Computer, Communication and Control (IC4)*, 2015, hlm. 1–4.
- [7] NVIDIA Corporation, *CUDA C Programming Guide*, Version 9.1. 2018.
- [8] A. Horga, "Genetic algorithms – CPU vs GPU implementation discussion Metaheuristics course report."

*Halaman ini sengaja dikosongkan*