Analisis Performansi Exponential Mechanism Pada Dataset Student's Alcohol Consumptions Dalam Memenuhi ε-Differential Privacy

Shafira Salsabilla Pradina Putri[1], Vera Suryani[2], Erwid M. Jadied[3]

[1,2,3] Universitas Telkom, Bandung

shafirasalsabilla@students.telkomuniversity.ac.id[1], verasuryani@telkomuniversity.ac.id[2], jadied@telkomuniversity.ac.id[3]

Abstrak

***Differential Privacy* diusulkan untuk memberikan solusi terhadap masalah pengamanan data yang dapat menjamin bahwa utilitas data dan privasi data dapat dijaga dan dikontrol dengan baik. *Differential Privacy* semakin berkembang pesat dan dianggap sebagai metode yang menjanjikan dan telah digunakan oleh berbagai pihak seperti Apple dan Google pada sistem keamanannya . Dalam penelitian ini, dilakukan uji eksperimen pada kueri agregat '*coun*t' untuk mempelajari performansi dari salah satu mekanisme *differential privacy* yaitu *Exponential Mechanism* pada dataset *Student's Alcohol Consumptions*. Dataset *Student's Alcohol Consumptions* relevan dengan penelitian sebelumnya. Dimana dataset ini berisikan informasi-informasi sensitif terkait konsumsi alkohol siswa sekolah menengah sehingga harus dijaga privasinya. Selanjutnya, ditentukan nilai ε(epsilon) yang optimal melalui pengujian nilai *relative error* terhadap penggunaan nilai ε(epsilon) serta menganalisis pengaruhnya terhadap *privacy tradeoff*. Studi eksperimen yang dilakukan menunjukkan bahwa nilai ε = 10 pada dataset *Student's Alcohol Consumptions* menjaga *privacy tradeoff* dengan baik (dengan rata-rata error pada tiap kategori = 22%) sehingga dataset memenuhi ε-differential privacy.**

Kata kunci : Differential Privacy, Exponential Mechanism, Privacy Tradeoff

Abstract

Differential Privacy is proposed to provide a solution to the problem of data security that can ensure that data utility and data privacy can be maintained and controlled properly. Differential Privacy is growing rapidly and is considered a promising method and has been used by various parties such as Apple and Google in their security systems. In this study, an experimental test was conducted on the aggregate query 'count' to study the performance of one of the differential privacy mechanisms, namely Exponential Mechanism on the Student's Alcohol Consumptions dataset. The Student's Alcohol Consumptions dataset is relevant to previous research. Where this dataset contains sensitive information related to high school students' alcohol consumption so that their privacy must be maintained. Furthermore, the optimal value of ε (epsilon) is determined by testing the relative error value on the use of the value of ε (epsilon) and analyzing its effect on the privacy tradeoff. The experimental study conducted shows that the value of ε = 10 in the Student's Alcohol Consumptions dataset maintains the privacy tradeoff well (with an average error in each category = 22%) so that the dataset meets ε -differential privacy.

Keywords: Differential Privacy, Exponential Mechanism, Privacy Tradeoff

1. Pendahuluan

Latar Belakang

Data merupakan kunci yang sangat penting pada era digitalisasi seperti sekarang ini, dimana keberadaan data sangat dibutuhkan oleh banyak pihak untuk kepentingan yang berbeda-beda. Analisis data memiliki peran yang sangat besar bagi pihak-pihak terkait untuk mengumpulkan serta memproses informasi yang berguna bagi kepentingan pihak-pihak tersebut. Sebagai contoh, beberapa pihak membutuhkan data untuk kepentingan penelitian, suatu perusahaan perlu mengenal pola perilaku customers & users mereka, ataupun untuk kebutuhan rekomendasi produk. Itulah mengapa kebutuhan akan suatu informasi terus meningkat dari waktu ke waktu. Akan tetapi, tidak jarang data berisikan suatu informasi pribadi yang dapat dikatakan sebagai informasi sensitif. Tentunya hal ini dapat menempatkan suatu individu dalam resiko berbahaya dimana informasi pribadinya dapat sewaktu-waktu digunakan untuk hal yang tidak seharusnya, mengingat bahwa pada era digitalisasi ini data merupakan suatu asset yang sangat penting bagi tiap individu dan harus dijaga keamanannya. Pada kasus ini, tidak bisa disangkal bahwa data sangat dibutuhkan namun keamanan dari data tersebut tetap harus diutamakan.

Meskipun terdapat beberapa teknik pengamanan data yang telah diusulkan, namun kelemahan-kelemahan pada teknik yang telah diusulkan masih ada. Contohnya adalah penggunaan metode privasi tradisional seperti enkripsi dan *anonymity*. Pada metode enkripsi dibuat data tidak dapat diakses sama sekali kecuali oleh orang yang memiliki kunci [1] . Pada penggunaan metode *anonymity* yang masih rentan terhadap serangan privasi seperti *de-identification* dan *record linkage* [2]. Contoh serangan privasi pada *anonymity* adalah pada kasus Netflix

Prize pada tahun 2006. Arvind Narayanan dan Vitaly Shmatikov (2007) melalui papernya yang berjudul "*Robust De-anonymization of Large Datasets (How To Break Anonymity of the Netflix Prize Dataset)*", menerapkan metode *de-anonymization* ke data Netflix Prize yang berisi peringkat film anonim dari 500.000 pelanggan Netflix. Berdasarkan penelitian tersebut, pihak ketiga (atau adversary) yang memiliki informasi tambahan dari IMDb (*Internet Movie Database* / Basis Data Film Internet) dapat dengan mudah mengidentifikasi informasi dari pelanggan Netflix meskipun data tersebut sudah di anonimisasi [3]. Oleh karena itu, dibutuhkan suatu teknik pengamanan data yang dapat menjamin bahwa utilitas data dan privasi data dapat dijaga dan dikontrol dengan baik. Pada tahun 2006, Cynthia Dwork mengusulkan metode pendekatan baru dalam menjaga privasi data yang disebut *Differential Privacy* [4].

Differential Privacy adalah pendekatan baru yang memungkinkan seseorang untuk mengumpulkan, mengolah, ataupun menganalisis data sekaligus menjamin perlindungan privasi terhadap informasi individu yang ada pada dataset [5]. Sehingga ada atau tidaknya suatu informasi individu pada dataset, tidak akan mempengaruhi hasil analisis data. *Differential Privacy* belakangan ini masih menjadi topik penelitian yang popular dalam bidang privasi data serta penggunaannya telah berkembang pesat. Apple dan Google adalah dua perusahaan besar yang telah mengaplikasikan metode *differential privacy* pada sistem keamanannya. *Differential Privacy* dianggap dapat menjamin perlindungan informasi sensitif dari data suatu individu, terlepas dari latar belakang informasi yang diketahui oleh pihak ketiga (*attacker*) [2].

Differential Privacy adalah metode pertama yang dapat membuktikan *privacy guarantee* secara matematis dan dapat mengukur potensi dari *privacy loss (*s). *Privacy loss* merupakan parameter yang membatasi berapa banyak hilangnya privasi pada suatu data. Parameter s merepresentasikan tradeoff antara privasi dan akurasi data, dimana parameter s menentukan berapa banyak noise yang akan ditambahkan ke dataset. Jika value s lebih kecil, maka jaminan privasi lebih besar namun akurasi dari hasil ouput makin kecil. Jika value s lebih besar, maka akurasi dari hasil output makin besar namun jaminan privasi semakin kecil. Sehingga menentukan s yang tepat merupakan pilihan yang sangat penting ketika mengimplementasi differential privacy [5].

Penelitian [2] mempelajari penerapan *Differential Privacy* menggunakan *Laplace Mechanism* pada sampel dataset terpilih. Berdasarkan penelitian tersebut disebutkan bahwa penerapan *Laplace Mechanism* hanya terbatas pada tipe data numerik saja, serta diperlukannya suatu pendekatan yang dapat mempermudah dalam menentukan nilai s(epsilon) yang optimal. Disebutkan juga bahwa penggunaan nilai s(epsilon) yang sama dapat menghasilkan privacy guarantee yang berbeda tergantung dari jenis kueri data yang dilakukan dan atribut data yang digunakan.

Exponential Mechanism adalah salah satu mekanisme fundamental dari differential privacy yang dapat digunakan untuk tipe data non-numerik. Oleh karena itu, pada penelitian ini penulis melakukan eksperimen untuk uji performansi differential privacy pada dataset Student's Alcohol Consumptions dengan menggunakan Exponential Mechanism, kemudian menganalisis nilai s(epsilon) yang optimal agar dataset dapat mencapai s-differential privacy hingga privacy tradeoff dapat terkontrol dengan baik.

Topik dan Batasannya

Mengacu pada latar belakang yang telah disampaikan, rumusan masalah dalam penelitian ini adalah "Berapakah nilai s (epsilon) yang optimal pada dataset *Student's Alcohol Consumptions* agar dapat mencapai s-*differential privacy* hingga *privacy tradeoff* dapat terkontrol dengan baik?".

Adapun batasan masalah pada penelitian ini antara lain :

1. Mekanisme yang digunakan adalah *Exponential Mechanism*.
2. Kueri agregat yang digunakan untuk uji eksperimen adalah kueri agregat 'count'.
3. Parameter yang digunakan dalam menentukan *privacy tradeoff* adalah nilai s (epsilon) dan nilai *relative error*.

Tujuan

Mengacu pada topik dan batasan yang telah disampaikan, tujuan dalam penelitian ini adalah "Menentukan nilai s (epsilon) yang optimal pada dataset *Student's Alcohol Consumptions* agar dataset dapat *mencapai* s-differential privacy hingga privacy tradeoff dapat terkontrol dengan baik".

Hal ini bisa diilustrasikan pada Tabel 1.

Tabel 1. Keterkaitan antara tujuan, pengujian dan kesimpulan

No	Tujuan	Pengujian	Kesimpulan
1	Menentukan nilai s (epsilon) yang optimal pada dataset	Menghitung nilai s (epsilon) yang optimal melalui	Penggunaan nilai s sangat mempengaruhi keseimbangan antara *noise* dan utilitas

	Student's Alcohol Consumptions agar dapat mencapai s-differential privacy hingga privacy tradeoff dapat terkontrol dengan baik	implementasi *Exponential Mechanism* dan hasil perhitungan *error* terhadap nilai s yang berbeda yang digunakan pada dataset	data. Setelah dilakukan pengujian, didapatkan hasil bahwa nilai s = 10 memiliki rata-rata *error* paling rendah. Sehingga nilai s = 10 dianggap paling optimal untuk diimplementasikan terhadap dataset *Student's Alcohol Consumptions* dalam memenuhi s -*differential privacy*

Organisasi Tulisan

Adapun organisasi tulisan pada penelitian ini adalah sebagai berikut :

1. Pendahuluan : Pada bagian ini dijelaskan penjelasan mengenai Latar Belakang, identifikasi topik beserta batasannya, tujuan, serta metode penelitian.
2. Studi Terkait : Pada bagian ini dijelaskan semua teori/studi literatur yang mendukung terkait topik Tugas Akhir. Serta dijelaskan juga metric pengukuran dan data yang digunakan pada permasalahan topik Tugas Akhir.
3. Sistem yang dibangun : Bagian ini menjelaskan rancangan dan sistem yang dibuat & dipakai dalam memenuhi tujuan Tugas Akhir.
4. Evaluasi : Bagian ini berisi dua sub-bagian, yaitu Hasil Pengujian dan Analisis Hasil Pengujian. Pengujian dan analisis yang dilakukan selaras dengan tujuan TA sebagaimana dinyatakan dalam Pendahuluan.

2. Studi Terkait

2.1 Differential Privacy

Memasuki era digitalisasi dan era *big data*, tentunya *data privacy* menjadi topik yang paling sering dibicarakan. Banyaknya perusahan dan instansi besar seperti lembaga pemerintahan, perusahaan Big Tech (Apple, Google), *e-commerce*, membutuhkan data untuk mengenal pola interaksi dan perilaku dari para penggunanya. Namun tidak jarang hal ini bisa saja melanggar privasi dari para pengguna [6]. Data yang dikumpulkan tersebut dapat dijual, ditukar ataupun dibagikan ke pihak ketiga. Misalnya *e-commerce* membagikan data pelanggan mereka ke pihak ketiga seperti agen periklanan [2]. Meskipun telah banyak metode yang diusulkan untuk menjaga privasi orang-orang yang membagikan data mereka, belakangan ini *differential privacy* menjadi teknik pengamanan data yang paling banyak diadopsi oleh banyak perusahaan [7]. *Differential Privacy* dianggap dapat menjamin perlindungan informasi sensitif dari data suatu individu, terlepas dari latar belakang informasi yang diketahui oleh pihak ketiga (*attacker*) [2].

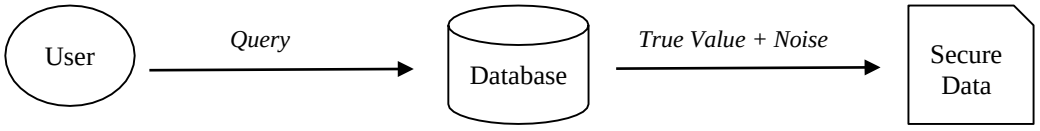
2.1.1 Pengertian

Differential Privacy adalah definisi matematis yang kuat dari privasi dalam konteks analisis statistik dan pembelajaran mesin. Penggunaan *differential privacy* memungkinkan dilakukannya pengumpulan, analisis dan berbagai perkiraan statistik (perhitungan rata-rata, table kontingensi, dan sintesis data) berdasarkan personal data yang digunakan, sekaligus melindungi privasi individu yang ada pada data [8]. Lebih spesifiknya lagi, *differential privacy* adalah definisi matematis dari privasi dimana *privacy risk* dapat diukur. Metode ini mempertimbangkan level maksimum dari *privacy loss* yang disebut *privacy loss parameter*, serta memanipulasi dataset agar data mencapai level privasi sekaligus mempertahankan utilitas dan akurasi dari data [5]. *Differential Privacy* memiliki manfaat yang jelas karena dianggap kuat terhadap berbagai serangan privasi. Selain itu penggunaannya juga transparan, karena data masih dapat dikumpulkan dan dibagikan untuk kepentingan analisis tanpa memiliki dampak negatif terhadap privasi individu di dalam data. Hal ini meningkatkan potensi untuk mengakses data yang sebelumnya tidak dapat dibagikan secara luas seperti data sensitif, data medis ataupun data keuangan [5]. Ide utama dari *differential privacy* adalah partisipasi informasi individu pada data tidak mempengaruhi data privasi dari individu tersebut [9]. Ada atau tidaknya partisipasi individu pada data juga tidak akan mempengaruhi hasil analisis data. Sehingga, data analis tidak dapat mempelajari hal apapun terkait informasi privasi individu baik sebelum melakukan analisis data ataupun setelahnya. Hal ini dapat terjadi karena *differential privacy* mengandalkan *random noise* untuk ditambahkan kedalam dataset agar analisis data yang dihasilkan menjadi berisik (*noisy data*) dan (jika memungkinkan) sulit untuk melanggar data privasi [10].

2.1.2 Cara Kerja *Differential Privacy*

Differential privacy bekerja dengan menambahkan *random noise* kedalam dataset pada saat melakukan analisis data, yang mana hal ini menutup kemungkinan untuk mempelajari informasi individu yang ada pada dataset berdasarkan hasil analisis. Namun perlu digaris bawahi bahwa hasil analisis data setelah penambahan *noise* bukanlah 100% akurat, melainkan perkiraan yang mendekati akurasi asli dan nilai yang sebenarnya. Hal ini memungkinkan hasil analisis data akan terus berubah selama beberapa kali karena *noise* yang digunakan adalah *random* [6]. Dengan demikian, *differential privacy* memberikan jaminan privasi (*privacy guarantee*) resiko yang sama pada tiap individu terlepas apakah informasi mereka ada pada analisis ataupun tidak [5].

Secara singkat dapat dilihat pada diagram berikut :



Gambar 1. Alur Differential Privacy

Differential privacy bergantung pada probabilitas. Maka, *noise* yang ditambahkan kedalam dataset haruslah acak. *Random noise* yang dibutuhkan ini harus diperoleh dari distribusi probabilitas [9]. Probabilitas yang digunakan juga harus bergantung terhadap jenis mekanisme dan tipe data yang akan digunakan. Sebagai contoh jika tipe data yang akan digunakan adalah data dengan tipe data numerik, maka salah satu mekanisme yang cocok untuk digunakan adalah *Laplace Mechanism*. Dimana *Laplace Mechanism* menambahkan sejumlah *noise* (ke perhitungan data asli) yang diperoleh dari fungsi probabilitas distribusi *laplace*. Jumlah *noise* yang ditambahkan akan berujung kepada kondisi dimana user harus bisa menyeimbangkan *privacy* dan utilitas data (kondisi ini disebut sebagai *privacy tradeoff*). Menambahkan terlalu banyak *noise* akan membuat data anonim, namun utilitas data akan berkurang. Dalam *differential privacy*, *privacy tradeoff* ini dikontrol oleh parameter *privacy loss* yang dilambangkan dengan s (epsilon) [11]. Apabila privacy tradeoff ini dapat dikontrol dengan baik, maka data akan memenuhi s-differential privacy.

2.1.3 Komponen Differential Privacy

Differential privacy identik dengan penambahan noise pada suatu dataset sehingga pihak ketiga tidak dapat mengidentifikasi apakah data suatu individu digunakan untuk analisis atau tidak. Ada beberapa komponen yang perlu diketahui dalam *differential privacy* :

A. Utilitas Data & Privasi Data

Fokus utama pada *differential privacy* adalah tentang bagaimana menyeimbangkan *privacy tradeoff* antara privasi data dan utilitas data. Jika *privacy loss parameter* diatur untuk meningkatkan utilitas data, maka privasi data akan menurun (dimana jumlah *noise* yang ditambahkan adalah sedikit). Jika *privacy loss parameter* diatur untuk meningkatkan privasi data, maka utilitas data akan menurun (dimana jumlah *noise* yang ditambahkan adalah banyak) [5]. Sehingga sangat penting untuk menentukan s yang tepat.

B. Privacy Guarantee

Privacy guarantee menjamin bahwa siapapun yang melihat hasil dari *differential privacy* analisis, pada dasarnya akan membuat kesimpulan yang sama tentang informasi pribadi individual siapapun. Baik apakah informasi pribadi tersebut ada pada analisis ataupun tidak [8]. Untuk membuat *privacy guarantee* menjadi lebih kuat, maka dibutuhkan nilai *privacy loss* yang sangat kecil [10]. Agar *privacy guarantee* ini menjadi lebih terjamin, maka kita dapat memaksakan *maximum privacy loss* atau bisa juga disebut *privacy budget*. Perlu untuk diingat, apabila *privacy budget* terus bertambah maka *privacy guarantee* memburuk [10].

C. Privacy Loss

Privacy loss menjelaskan bahwa sesuatu yang dapat dipelajari/didapatkan pada hasil *differential privacy* terhadap informasi individu adalah terbatas dan dapat diukur [8]. Batas atas (*upper bound*) dari *privacy loss* diukur dengan menggunakan *privacy loss parameter* yang dilambangkan dengan s (epsilon). Parameter ini melambangkan privacy tradeoff serta menentukan jumlah noise yang akan

ditambahkan kedalam data. Sehingga pemilihan nilai ε (epsilon) yang tepat perlu dipertimbangkan dengan baik ketika mengimplementasikan differential privacy [5]. Parameter ini mengukur kuantitas dari query pada data yang digunakan terhadap differential privacy, sehingga dapat mengontrol proporsi output dari fungsi probabilitas.

D. Sensitivity

Parameter *sensitivity* mengukur seberapa banyak *noise* yang dibutuhkan pada mekanisme *differential privacy*. Secara kasar, sensitivitas dari suatu fungsi mencerminkan berapa banyak jumlah output dari fungsi yang akan berubah ketika inputnya juga berubah. Parameter *sensitivity* yang selalu digunakan pada mekanisme *differential privacy* ada dua, yaitu *Global Sensitivity* dan *Local Sensitivity* [2] [12].

- *Global Sensitivity* : Parameter ini menunjukkan seberapa jauh perbedaan maksimal yang dihasilkan oleh kueri data setelah dan sebelum ditambahkan oleh *noise*. *Global sensitivity* bekerja dengan baik pada saat melakukan kueri seperti *count* atau *sum* yang memiliki sensitivitas rendah. Sebagai contoh, untuk kueri *count* memiliki G(S) = 1 karena menambahkan atau menghapus satu *row* dari dataset akan merubah hitungan count >= 1. Namun, pada kueri lain seperti *median* atau *mean* maka G(S) yang digunakan pasti bernilai lebih besar [2] [10].
- *Local Sensitivity* : Parameter ini mengukur sensitivitas untuk kumpulan data lokal, dimana perubahan yang terjadi terbatas pada data lokal (bukan dari seluruh kumpulan data). Untuk kueri data seperti *count*, *local sensitivity* identik dengan *global sensitivity*. Perbedaan antara dua *sensitivity* ini adalah *global sensitivity* adalah sensitivitas minimum yang diperlukan kueri untuk mencakup semua kemungkinan pada dataset, sedangkan *local sensitivity* adalah sensitivitas minimum yang diperlukan kueri untuk mencakup satu kumpulan data tertentu pada dataset [13].

2.2 Exponential Mechanism

2.2.1 Pengertian

Mekanisme eksponensial (yang diusulkan oleh Frank McSherry and Kunal Talwar pada tahun 2007) adalah salah satu metode *ε-differential privacy* untuk memilih suatu elemen di dalam himpunan data [14]. Beberapa mekanisme *differential privacy* seperti *Laplace Mechanism* atau *Gaussian Mechanism* berfokus untuk menjawab kueri numerik. Walaupun demikian, diketahui bahwa fungsi kueri tidak selalu menghasilkan hasil numerik. Oleh karena itu, untuk dapat menjawab kueri non-numerik kita dapat menggunakan *Exponential Mechanism*. *Exponential Mechanism* bekerja dengan mengeluarkan elemen terbaik dari suatu set data dengan tetap menjaga privasi data [2][12].

Penentuan elemen terbaik yang akan dioutput ditentukan melalui fungsi penilaian skor (*scoring function*) untuk tiap elemen pada suatu set data. Namun, biasanya hasil ouput dari *exponential mechanism* tidak selalu merupakan elemen yang memiliki skor tinggi. Hal ini terjadi karena, mekanisme ini memenuhi *differential privacy* dengan cara memperkirakan skor maksimal untuk tiap elemen data [12].

2.2.2 Cara Kerja Exponential Mechanism

Mekanisme eksponensial menambahkan noise lalu mengoutput elemen r terbaik pada set data D melalui fungsi probabilitas [15] :

$$\Pr\ [A(D) = r]\ \text{a exp}\ (\frac{\varepsilon u(D,r)}{2\Delta u}) \quad (1)$$

dimana,

$\Pr[A(D) = r]$: Probabilitas elemen r
ε : Parameter epsilon (*privacy loss parameter*)
$u(D, r)$: *Quality scoring function*
Δu : *Sensitivity* dari *utility scoring function*

Exponential Mechanism dianggap telah memenuhi ε-*differential privacy* apabila sudah memenuhi Δu. Untuk mendefinisikan Δu atau *sensitivity* dari *utility scoring function*, maka perhitungan digunakan menggunakan parameter *global sensitivity* [16] *:*

$$GS(u)\ = \max r \in R, D, D' \colon \|D - D'\|\ \leq 1\ |u(D,r) - u(D',r)| \quad (2)$$

Mekanisme eksponensial memenuhi ϵ-*differential privacy* [12]:

- Analis memilih satu set R dari beberapa kemungkinan output

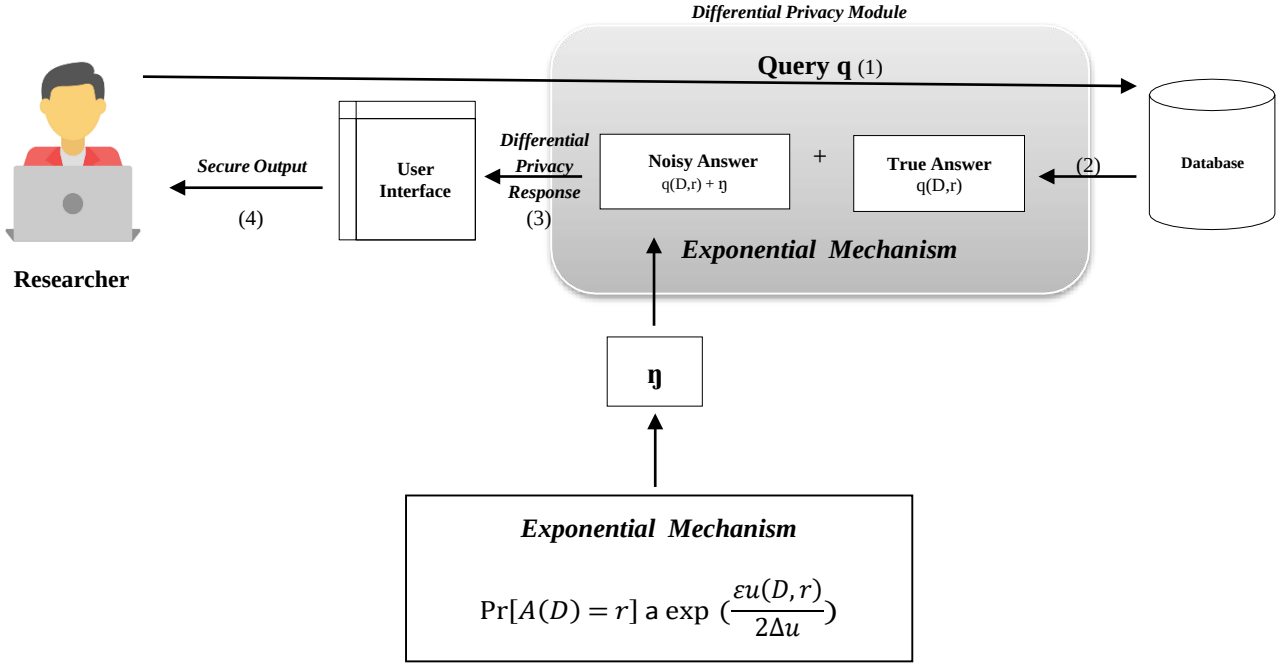
- Analis menentukan fungsi penilaian u: D × R → **R** dengan sensitivitas global Δu
- Output mekanisme eksponensial r∈R dengan probabilitas sebanding dengan:

$$exp\ (\ \epsilon u(x, r)\ /\ 2\Delta u\ )$$

3. Sistem yang Dibangun

3.1 Rancangan Sistem

Studi eksperimen yang dilakukan pada Tugas Akhir ini adalah menganalisis bagaimana performansi dari mekanisme *diffrerential privacy* dapat menghasilkan output dataset yang lebih *secure*. Penulis menambahkan *noise* ke dalam dataset pada *aggregate queries* dengan menggunakan *Exponential Mechanism* sebagai metode *differential privacy* untuk menjaga data privasi pada dataset. *User interface* yang digunakan pada penelitian berbentuk *web framework* sederhana yang berfungsi sebagai tampilan visual dari hasil output kueri agregat. Dimana *user* dapat melakukan kueri data terhadap *database*, kemudian hasil daripada kueri data tersebut merupakan data *secure* yang dimana *noise* telah ditambahkan ke perhitungan data asli. Secara sederhana dapat dilihat pada diagram berikut:



Gambar 2. Rancangan Sistem

Untuk menentukan nilai s yang optimal pada dataset *Student's Alcohol Consumptions*, penulis mengukur nilai relative error pada tiap parameter s yang digunakan kemudian menganalisis pengaruhnya terhadap *privacy tradeoff*. Sehingga kemudian dapat dianalisis berapakah nilai optimal s untuk dataset *Student's Alcohol Consumptions* agar dapat memenuhi s-differential privacy.

3.2 Software & Hardware Programming

A. User Interface

Pembuatan *user interface* pada penelitian ini dalam bentuk *web framework* sederhana dibantu oleh *framework* Django. Django adalah *framework open source full-stack* yang membantu dalam membangun sebuah web dengan didukung oleh Bahasa pemrograman Python. Django memiliki *template*, *libraries*, dan API yang memudahkan pengembangan aplikasi dengan cepat.

B. Software & Programming Language

Bahasa pemrograman yang digunakan pada penelitian ini adalah Python karena Bahasa Python mudah dipelajari serta diimplementasikan. Dan juga, python memiliki *library* yang lengkap dan sangat mendukung. Selain itu, penulis juga menggunakan Google Colaboratory sebagai *coding environment* untuk mengeksekusi eksperimen dan visualisasi data.

C. Hardware

Semua eksperimen dan analisis yang dikerjakan oleh penulis dilakukan menggunakan Sistem Operasi Windows 10 Home Single Languange 64-bit (10.0, Build 19041) dengan processor Intel®Core™i5-8250U CPU @ 1.60GHz (8CPUs), ~1,8Ghz.

3.3 Dataset

Dataset yang dipilih adalah dataset *Student's Alcohol Consumptions* yang didapat dari Kaggle: *Your Machine Learning and Data Science Community*. Data ini berisi banyak informasi sosial, gender dan studi yang menarik tentang siswa pada sekolah menengah. Dataset ini memiliki 649 row data serta memiliki 33 atribut :

Atribut	Dekripsi
School	student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
Sex	student's sex (binary: 'F' - female or 'M' - male)
Age	student's age (numeric: from 15 to 22)
Address	student's home address type (binary: 'U' - urban or 'R' - rural)
Famsize	family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
PStatus	parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
Medu	mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
Fedu	father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
Mjob	mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
Fjob	father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
Reason	reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
Guardian	student's guardian (nominal: 'mother', 'father' or 'other')
Traveltime	home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
Studytime	weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
Failures	number of past class failures (numeric: n if 1<=n<3, else 4)
Schoolsup	extra educational support (binary: yes or no)
Famsup	family educational support (binary: yes or no)
Paid	extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)

Activities	extra-curricular activities (binary: yes or no)
Nursery	attended nursery school (binary: yes or no)
Higher	wants to take higher education (binary: yes or no)
Internet	Internet access at home (binary: yes or no)
Romantic	with a romantic relationship (binary: yes or no)
Famrel	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
Freetime	free time after school (numeric: from 1 - very low to 5 - very high)
Goout	going out with friends (numeric: from 1 - very low to 5 - very high)
Dalc	workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
Walc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
Health	current health status (numeric: from 1 - very bad to 5 - very good)
Absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)
G3	final grade (numeric: from 0 to 20, output target)

3.4 Eksperimen

Pada studi eksperimen yang dilakukan, penulis akan menguji salah satu mekanisme *differential privacy*, yaitu *Exponential Mechanism* dengan mengimplementasikannya kepada sample dataset terpilih yaitu dataset *Student's Alcohol Consumptions* dan akan berfokus kepada kueri agregat *count*.

Pertama-tama, penulis mengimport dataset dalam format .csv ke dalam kodingan dalam Bahasa python. Uji eksperimen ini dilakukan menggunakan Google Colaboratory. Setelah dataset di-import, beberapa library yang dibutuhkan yaitu matplotlib, pandas dan numpy di *load*. Selanjutnya, *dataframe* akan membaca file dataset dalam format .csv. Karena penulis berfokus ke kueri agregat, maka pada eksperimen kali ini penulis ingin menguji performasi dari *exponential mechanism* dengan menghitung (*count*) jumlah dari kategori 'Mjob', lalu menghitung presentase kemunculan tiap elemen beserta *score* dari tiap-tiap elemennya. Mengacu kepada fungsi kualitas :

$$q: D \times r \rightarrow R$$

Maka, kategori 'Mjob' dianggap sebagai set data (D), sedangkan options adalah elemen pada set (r). Hasil output akan ditampilkan pada hasil 4.1.1

Kode 3.4.1 Hitung Score & Presentase Kemunculan di Dataset Tiap Elemen

```
options = df['Mjob'].unique()
def score(data, option):
    return data.value_counts()[option]/1000
for option in options :
  print (option + " : " + str(score(df['Mjob'], option)*1000/649*100)  + "%" + " : " + str(score(df['Mjob'], option)))
```

Selanjutnya, setelah score pada tiap elemen dihitung, dilakukan implementasi *exponential mechanism*. Implementasi *exponential mechanism* ini menambahkan *noise* ke perhitungan data asli. *Noise* yang akan ditambahkan, didapat dari fungsi probabilitas :

$$\Pr[A(D) = r] \text{ a } \exp\left(\varepsilon u(D, r)/2\Delta u\right)$$

Kode 3.4.2 Implementasi Exponential Mechanism

```
def exponential(x, R, u, sensitivity, epsilon):
    scores = [u(x, r) for r in R]
    probabilities = [np.exp(epsilon * score / (2 * sensitivity)) for score in scores]
    probabilities = probabilities / np.linalg.norm(probabilities, ord=1)
    return np.random.choice(R, 1, p=probabilities)[0]
exponential(df['Mjob'], options, score, 1, 10)def score(data, option):
    return data.value_counts()[option]/1000
for option in options :
  print (option + " : " + str(score(df['Mjob'], option)*1000/649*100)  + "%" + " : " + str(score(df['Mjob'], option)))
```

Hasil daripada kode 3.4.2 dapat dilihat pada hasil 4.1.2. Pada eksperimen ini, penulis menggunakan nilai s = 10 serta nilai sensitivity = 1. (*Aggregat count* hanya memiliki sensitivity = 1 karena menambahkan ataupun menghapus 1 *row* dari dataset, akan merubah hitungan count >=1 [10]). Selanjutnya, akan ditampilkan hasil perhitungan *count* dari kategori 'Mjob' setelah ditambah *noise* dan perhitungan *count* asli tanpa *noise*. Hasil akan ditampilkan pada hasil 4.1.3

Kode 3.4.3 Hasil Real Count & Noise Count

```
#Hasil count menggunakan exponential
r_exp = [exponential(df['Mjob'], options, score, 1, 10) for i in range(649)]
print ("Exponential Mechanism Count Result")
r_exp_ = pd.Series(r_exp).value_counts()
print (r_exp_)

print ("")

#Hasil real count tanpa exponential
r_real = df['Mjob']
print ("Real Count Result")
r_real_ = pd.Series(r_real).value_counts()
print (r_real_)
```

Melalui percobaan ini, dapat dilihat perbedaan hasil *count* setelah dan sebelum diimplementasikannya *exponential mechanism*. Untuk mengukur utilitas dari output, penulis mengukur utility data dengan menghitung nilai *absolute-error* dan *relative-error*.
Absolute-error : Selisih antara *true count* dan *noisy count*
Relative-error : Hasil *absolute-error* dibagi *true count*

Kode 3.4.4 Perhitungan Error

```
# First cell-by-cell errors.
errortab=np.abs(r_exp_-r_real_)
print("Cell-by-cell absolute error:")
print(errortab)
print()

# Then relative errors.
pcterrortab=errortab/r_real_
print("Cell-by-cell relative error:")
print(pcterrortab)
print()

# Now the total error and the relative error.
errorsum=errortab.sum().sum()
print("Total L1 Error: {0:1d}".format(errorsum))
truesum=r_real_.sum().sum()
errorrel=100*errorsum/truesum
print("Relative L1 Error: {:4.2f}%".format(errorrel))
```

Hasil dari kode 3.4.4 dapat dilihat pada hasil 4.1.4. Terakhir, untuk memilih hasil *error* yang paling kecil dari tiap-tiap nilai s yang digunakan agar privacy serta utilitas data seimbang, penulis melakukan beberapa kueri sebanyak n-kali terhadap seluruh kategori (atribut pada dataset). Hasil dapat dilihat pada hasil 4.1.5.

4. Evaluasi

4.1 Hasil Pengujian

Hasil 4.1.1 Hitung Score & Presentase Kemunculan di Dataset Tiap Elemen

at_home : 20.801232665639446% : 0.135
health : 7.395993836671804% : 0.048
other : 39.753466872110934% : 0.258
services : 20.955315870570107% : 0.136
teacher : 11.093990755007704% : 0.072

Hasil 4.1.2 Implementasi Exponential Mechanism

"Services"

Hasil 4.1.3 Hasil Real Count & Noise Count

Exponential Mechanism Count Result
other 211
services 141
at_home 128
teacher 103
health 66
dtype: int64

Real Count Result
other 258
services 136
at_home 135
teacher 72
health 48
Name: Mjob, dtype: int64

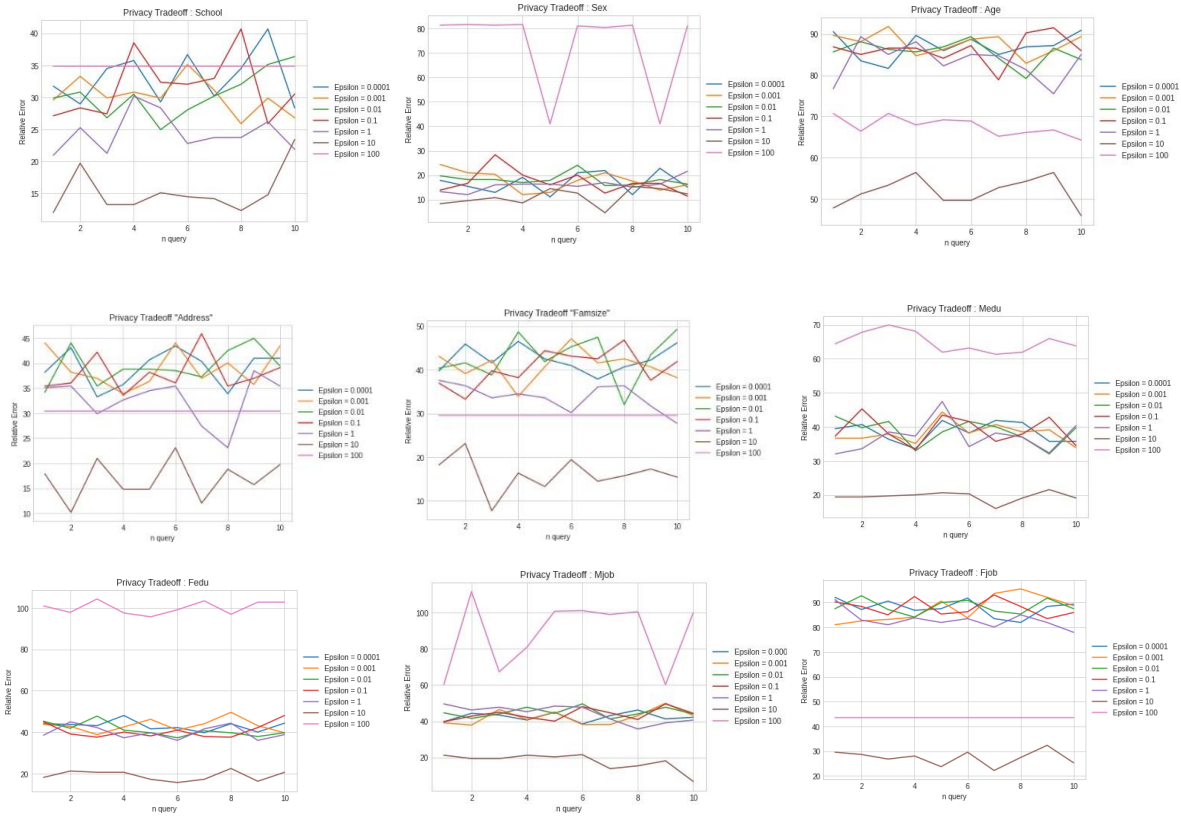
Hasil 4.1.4 Perhitungan Error

Cell-by-cell absolute error:
other 47
services 5
at_home 7
teacher 31
health 18
dtype: int64

Cell-by-cell relative error:
other 0.182171
services 0.036765
at_home 0.051852
teacher 0.430556
health 0.375000
dtype: float64

Total L1 Error: 108
Relative L1 Error: 16.64%

Hasil 4.1.5 Hasil error setelah n-queries



Gambar 3. Hasil Grafik Perbandingan Privacy Tradeoff Pada Tiap Kategori. (Relative error dengan satuan persen (%) , dan n-query merujuk ke query ke-n (kueri ke-1, ke-2, dst..) yang dilakukan).

Tabel 2. Hasil rata-rata error per-epsilon pada tiap kategori setelah 10 kali kueri

		Kategori															
		School	Sex	Age	Address	Famsize	PStatus	Medu	Fedu	Mjob	Fjob	Reason	Guardian	Traveltime	Studytime	Failures	Schoolsup
E	0,0001	33	16.918	87.025	39.076	42.434	73.312	38.49	43.112	42.464	87.949	39.073	71.741	78.861	59.323	118.522	76.98
p	0,001	30.23	17.69	87.706	38.983	40.925	72.726	38.12	43.174	42.217	87.549	37.658	73.837	78.243	59.353	118.367	77.995
s	0,01	30.477	18.119	85.578	39.416	42.897	77.351	38.642	41.109	44.931	88.381	40.279	74.359	76.178	58.366	118.026	82.619
i	0,1	31.588	17.225	86.316	37.906	40.462	74.453	38.982	40.678	43.759	87.888	36.795	71.679	79.17	57.997	115.749	76.488
l	1	24.437	15.931	83.328	32.758	33.774	62.805	37.104	39.97	44.191	82.958	35.961	65.269	72.821	53.467	111.156	65.424
o	10	15.254	11.124	51.71	16.795	16.085	7.734	19.537	19.046	17.749	27.365	7.641	14.146	19.692	12.974	13.683	5.793
n	100	34.82	73.191	67.611	30.35	29.58	12.33	64.838	100.122	88.136	43.45	78.215	29.89	50.112	85.131	15.41	10.48

		Kategori															
		Famsup	Paid	Activities	Nursery	Higher	Internet	Romantic	Famrel	Freetime	Goout	Dalc	Walc	Health	Absences	G1	G2	G3
E	0,0001	25.948	85.978	3.052	60.032	78.121	53.774	25.024	73.16	50.631	31.432	98.522	42.065	35.499	123.052	79.045	74.081	74.885
p	0,001	21.295	88.105	4.099	60.124	79.845	52.387	24.93	73.867	52.572	32.788	98.182	42.711	35.407	126.195	77.903	73.219	75.993
s	0,01	21.233	88.629	4.529	59.816	82.003	52.233	27.456	72.172	51.864	31.433	99.847	42.866	37.228	125.053	78.583	74.637	74.235
i	0,1	21.757	87.211	4.807	59.108	76.98	53.375	26.84	74.485	52.358	33.559	99.723	42.004	36.857	123.236	76.579	69.398	75.808
l	1	19.969	73.252	4.53	49.428	65.918	45.456	21.417	67.335	47.55	29.707	92.325	38.459	34.114	122.433	74.083	72.636	71.001
o	10	13.251	1.603	2.434	14.946	6.503	16.917	15.163	24.531	17.289	14.577	22.959	14.883	12.789	100.34	63.144	59.108	58.522
n	100	57.793	6.01	41.355	19.72	10.63	23.27	40.481	66.395	92.357	111.895	30.51	96.641	82.033	70.214	54.023	56.408	69.554

		Rata-Rata Error
E	0,0001	59
p	0,001	60
s	0,01	60
i	0,1	59
l	1	54
o	10	22
n	100	53

4.2 Analisis Hasil Pengujian

Pada ***hasil 4.1.1*** dapat dilihat bahwa elemen/option 'other' memiliki presentase kemunculan dan *score* paling tinggi yaitu 39.75% dan *score* 0.258. Namun, pada ***hasil 4.1.2***, *best element* yang di ouput oleh sistem adalah 'services' yang memiliki presentase kemunculan sebesar 20.95% dan *score* 0.136. Hal ini terjadi karena jika mengacu kepada cara kerja *exponential mechanism*, *best element* yang di output didapat dari hasil probabilitas. Sehingga hasil outputnya, tidak selalu merupakan elemen yang memiliki presentase kemunculan maupun *score* yang paling tinggi.

Pada ***hasil 4.1.3***, tabel *Exponential Mechanism Count Result* menampilkan hasil *count* dimana perhitungan aslinya telah ditambah dengan *noise*. Sedangkan tabel *real count* merupakan jumlah asli dari kategori 'Mjob'. Pada ***hasil 4.1.4,*** *Cell-by-cell absolute error* merupakan selisih antara hasil yang didapat oleh *noisy count* dan *real count*. *Relative error* sendiri adalah hasil dari *absolute-error* dibagi dengan *real count*. Lalu dapat dilihat juga hasil *total error* = 168 dan *total relative error* = 16.64%. Pada hakikatnya, mekanisme *differential privacy* menggunakan *exponential mechanism* dianggap sudah memenuhi ε-*differential privacy*.

Namun, akurasi dari hasil output tetap harus dipertanyakan. ε(epsilon) sendiri merupakan parameter kontrol sebagai penyeimbang antara *noise* dan utilitas data. Oleh karena itu, pemilihan ε yang tepat merupakan hal yang sangat penting. Pada ***hasil 4.1.5***, dilakukan uji terhadap seluruh kategori pada dataset, kemudian menentukan nilai ε optimal dengan melakukan query sebanyak 10-kali untuk melihat perhitungan error antara tiap-tiap ε yang digunakan. Untuk diketahui, disini dipilih nilai ε yang paling kecil adalah 0,0001 sedangkan nilai ε yang paling besar adalah 100.

Berdasarkan percobaan yang dilakukan, hasil grafik menunjukkan bahwa penggunaan parameter ε yang sama pada satu dataset memiliki pola grafik yang berbeda-beda. Hal ini disebabkan oleh tipe data pada kategori yang ada pada dataset berbeda-beda dan tiap kategori memiliki jumlah & *score* option yang berbeda-beda pula. Seperti halnya pada kategori Sex dan kategori Mjob, dimana kategori Sex memiliki dua option yaitu Female (F) dan Male (M), sedangkan kategori Mjob memiliki lima option yaitu at_home, health, other, services dan juga teacher. Sehingga walaupun parameter ε yang diuji coba adalah sama pada satu dataset, hasil error akan menunjukkan hasil yang fluktuatif tergantung dari jenis kueri dan tipe data yang digunakan. Hasil perhitungan *error* yang diperoleh dari grafik pada tiap kategori dan parameter ε yang berbeda, penulis analisis melalui tabel untuk kemudian dihitung rata-rata *error* pada tiap parameter ε. Karena grafik menunjukkan hasil yang berbeda-beda, maka penulis menentukan nilai ε optimal dengan memilih hasil ε secara global. Dimana berdasarkan table pada ***result 4.1.5***, penggunaan $\varepsilon = 0{,}001$ dan $\varepsilon = 0{,}01$ memiliki rata-rata error paling tinggi yaitu 60%. Sedangkan penggunaan $\varepsilon = 10$ memiliki rata-rata error yang paling rendah diantara seluruh parameter ε yang digunakan selama 10 kali kueri yaitu = 22%.

Oleh karena itu, berdasarkan hasil pengujian, penulis memilih menggunakan $\varepsilon = 10$ karena memiliki tingkat error yang cukup rendah, sehingga privacy tradeoff dapat terkontrol dengan baik. Dari percobaan ini dapat disimpulkan bahwa penggunaan parameter ε yang sama pada satu dataset, akan memiliki privacy tradeoff yang berbeda-beda, tergantung oleh tipe data dan jenis kueri data yang dilakukan. Sehingga penggunaan parameter $\varepsilon = 10$ tidak dapat dijadikan acuan ketika akan melakukan uji terhadap kueri yang lebih kompleks pada dataset yang sama. Dimana hasil yang didapat ini merupakan hasil dari kueri agregat 'count' pada satu kategori saja. Untuk itu sebagai contoh, jika jenis kueri yang ingin dilakukan adalah 'mean' atau 'count' pada dua kategori yang berbeda, maka hasil privacy tradeoff pasti akan berbeda sehingga parameter ε yang optimal belum tentu = 10. Maka dari itu, penggunaan parameter ε harus ditetapkan dengan cermat sesuai dengan tipe data ataupun jenis kueri yang ingin dilakukan. Pemilihan parameter ε yang optimal, akan mempengaruhi privacy tradeoff dari suatu data. Semakin tinggi nilai error yang dihasilkan, maka utilitas data semakin rendah. Begitu pula sebaliknya.

5. Kesimpulan

Exponential Mechanism bekerja terhadap kueri non-numerik/*categorical* dengan mengeluarkan elemen terbaik dari suatu set data. Pada implementasi yang telah dilakukan pada kueri agregat, hasil *count* menggunakan *exponential mechanism* memiliki hasil perhitungan yang sudah tercampur *noise*, dimana *noise* diperoleh dari perhitungan *score* dan fungsi probabilitas. Harus digaris bawahi bahwa penggunaan nilai ε sangat mempengaruhi keseimbangan antara *noise* dan utilitas data, sehingga harus dipilih dengan sangat hati-hati serta harus disesuaikan dengan sampel data yang akan digunakan agar *privacy tradeoff* dapat dikontrol dengan baik. Berdasarkan pengujian yang telah dilakukan pada kueri agregat '*count*', penggunaan nilai ε = 10 memiliki rata-rata *error* paling rendah untuk seluruh kategori yang ada pada dataset, yaitu = 22%. Oleh karena itu, ε = 10 dianggap paling optimal untuk diimplementasikan pada dataset Student's Alcohol Consumptions, sehingga dataset memenuhi ε-*differential privacy.*

Pada penelitian ini, parameter yang digunakan untuk menganalisis data agar mencapai ε-*differential privacy* adalah parameter ε dan nilai *relative error*, sehingga untuk penelitian berikutnya penggunaan parameter lain dapat dipertimbangkan untuk menghasilkan *privacy tradeoff* yang lebih baik pula, seperti pengujian terhadap berbagai serangan privasi yang dapat dilakukan sehingga dapat membuktikan apakah *differential privacy* kebal terhadap serangan privasi seperti *de-identification* atau *record linkage*. Pengujian menggunakan jenis kueri yang lebih kompleks juga dapat dilakukan untuk mengidentifikasi lebih jauh berapakah nilai parameter ε yang jauh lebih optimal untuk diimplementasikan terhadap dataset.

REFERENSI

[1] Zhu, T., 2018. Explainer: what is differential privacy and how can it protect your data?. [online] The Conversation. Available at: <https://theconversation.com/explainer-what-is-differential-privacy-and-how-can-it-protect-your-data-90686> [Accessed 17 September 2021].

[2] Seleshi, B. and ASSEFFA, S., 2017. A Case Study on Differential Privacy. MASTER'S THESIS, UMEÅ UNIVERSITY, Department of Computing Science, p.87.

[3] Narayanan, A. and Shmatikov, V., 2007. Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset). The University of Texas at Austin,.

[4] Dwork, C., 2006. Differential Privacy. Automata, Languages and Programming, Proceedings of the 33rd international conference on Automata, Languages and Programming - Volume Part II, pp.1-12.

[5] Ghandi, R. and Jayanti, A., 2020. Technology Factsheet: Differential Privacy. Harvard University, Harvard Kennedy School, the Belfer Center for Science and International Affairs,.

[6] Tyagi, N., 2021. What is Differential Privacy and How does it Work? | Analytics Steps. [online] Analyticssteps.com. Available at: <https://www.analyticssteps.com/blogs/what-differential-privacy-and-how-does-it-work> [Accessed 6 September 2021].

[7] Kaptchuk, G., M. Redmiles, E. and Cummings, R., 2021. People want data privacy but don't always know what they're getting -- GCN. [online] GCN. Available at: <https://gcn.com/Articles/2020/10/21/differential-privacy-understanding.aspx?Page=1> [Accessed 6 September 2021].

[8] Wood, A., Altman, M., Bembenek, A., Bun, M., Gaboardi, M., Honaker, J., Nissim, K., O'Brien, D., Steinke, T. and Vadhan, S., 2018. Differential Privacy: A Primer for a Non-Technical Audience. SSRN Electronic Journal,.

[9] Loman, J., 2018. Comparing the Performance of the Laplace and Staircase Mechanisms in Differential Privacy. Bachelor Thesis, Computer Science, Radboud University,.

[10] Elamurugaiyan, A., 2018. A Brief Introduction to Differential Privacy. [online] Medium. Available at: <https://medium.com/georgian-impact-blog/a-brief-introduction-to-differential-privacy-eacf8722283b> [Accessed 6 September 2021].

[11] Sartor, N., 2019. Explaining Differential Privacy in 3 Levels of Difficulty | Aircloak. [online] Aircloak. Available at: <https://aircloak.com/explaining-differential-privacy/> [Accessed 6 September 2021].

[12] Near, J. and Abuah, C., 2021. Programming Differential Privacy.

[13] Fathima, S., 2020. "Sensitivity" in Differential Privacy. [online] Medium. Available at: <https://becominghuman.ai/query-sensitivity-types-and-effects-on-differential-privacy-mechanism-c94fd14b9837> [Accessed 6 September 2021].

[14] Ji, Z., 2017. Learning Information from Data while Preserving Differential Privacy. UNIVERSITY OF CALIFORNIA, SAN DIEGO,.

[15] Xiong, L., 2018. CS573 Data Privacy and Security. [online] Mathcs.emory.edu. Available at: <http://www.mathcs.emory.edu/~lxiong/cs573_f16/share/slides/04_DP.pdf> [Accessed 17 September 2021].

[16] Maffei, M., 2016. Differential Privacy (Part II). [online] Cs.ioc.ee. Available at: <http://www.cs.ioc.ee/ewscs/2016/maffei/maffei-slides-lecture2.pdf> [Accessed 17 September 2021].

[17] Programming Differential Privacy [Source Code]. https://github.com/uvm-plaid/programming-dp

[18] Differential Privacy Demonstration Materials [Source Code]. https://github.com/mwerevu/dpdemo