

## Analisis dan Perancangan *Machine Learning* Untuk Mendeteksi Kegagalan *Job* di *Apache Spark*

Eri Dariato

Fakultas Teknik, Prodi Teknik Informatika  
Universitas Dian Nusantara, Jakarta, Indonesia  
Email : [eri.dariato@undira.ac.id](mailto:eri.dariato@undira.ac.id)

### Article Information

#### Article history:

Received 5 March 2022  
Revised 10 April 2022  
Accepted 20 May 2022  
Available 25 June 2022

### Keywords

Database  
Artificial Intelligence  
Apache Spark  
Feature Engineering  
SQL

### Corresponding Author:

Eri Dariato,  
Fakultas Teknik,  
Prodi Teknik Informatika,  
Universitas Dian Nusantara,  
Email : [eri.dariato@undira.ac.id](mailto:eri.dariato@undira.ac.id)

### ABSTRACT

A collection of data stored in a database, so the longer the data, the bigger the data, because the data processed is very large, processing time in Apache Spark can take up to a dozen or tens of hours. Sometimes, the Apache Spark application even fails. Therefore, to minimize the waiting time that could have been avoided or reduced, artificial intelligence through Machine Learning will be used to detect whether an Apache Spark application will fail or run smoothly. Factors to determine this failure are called features and are generated through the feature engineering process. The purpose of this research is to design Machine Learning so that it is able to find out what features will determine the success or failure of the Apache Spark application. The research method used is the Prototyping process model.

**Keywords :** *Database, Artificial Intelligence, Apache Spark, Feature Engineering*

### ABSTRAK

Kumpulan data-data yang tersimpan disuatu database, sehingga semakin lama data yang semakin besar, karena data yang diproses berukuran sangat besar, waktu proses di *Apache Spark* bisa memakan waktu sampai belasan atau puluhan jam. Tidak jarang pula, akhirnya aplikasi *Apache Spark* tersebut bahkan mengalami kegagalan. Oleh karena itu, untuk meminimalisir adanya waktu tunggu yang sebenarnya bisa dihindari atau dikurangi, akan digunakan kecerdasan buatan melalui Machine Learning untuk mendeteksi apakah sebuah aplikasi *Apache Spark* akan mengalami kegagalan atau lancar. Faktor-faktor yang paling berpengaruh terhadap penentuan aplikasi tersebut disebut dengan feature dan dihasilkan melalui proses feature engineering. Tujuan dari penelitian ini adalah merancang Machine Learning sehingga mampu mengetahui feature-feature apa saja yang paling menentukan kesuksesan atau kegagalan aplikasi *Apache Spark*. Metode penelitian yang digunakan adalah model proses Prototyping. Hasil luaran penelitian ini adalah sebuah sistem pengelolaan basis.

**Kata Kunci :** *Database, Artificial Intelligence, Apache Spark, Feature Engineering*

Copyright@2022 Eri Dariato

This is an open access article under the [CC-BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.



## 1. Pendahuluan

Untuk mengatasi kebutuhan kecepatan dan pemrosesan data jumlah besar, diperkenalkan teknologi hadoop. Untuk pemrosesan digunakan Apache Spark. Apache Spark adalah sebuah mesin pengolah data yang cepat dalam melakukan tugas pemrosesan pada set data yang sangat besar. Spark juga memungkinkan pengguna untuk memproses dengan menggunakan bahasa pemrograman Java, Scala, atau Python. Kelebihan ini yang membuat Apache Spark menjadi kunci dalam dunia Big Data dan *Machine Learning*. Berbeda dengan hadoop *map reduce*, pemrosesan data nya membutuhkan akses ke disk untuk membaca dan menulis data. Sedangkan Apache Spark menggunakan memory untuk menyimpan data untuk mengurangi faktor I/O (Foundation, 2020) . Kepopuleran Apache Spark, membuat pengguna makin menaik, begitu pula dengan jumlah data yang diproses. Setiap kali ada permintaan eksekusi ke Apache Spark, permintaan ini disebut dengan aplikasi spark (*job*). Untuk ukuran aplikasi yang besar, eksekusi bisa memakan waktu satu jam, dua jam, atau lebih lama dari itu.



Gambar 1. Contoh Keluhan Job Apache Spark Lambat

Tentu kondisi ini tidak akan selalu ideal, kadang terdapat kendala yang dirasakan para pengguna. Kendala yang dimaksud adalah *job* yang dieksekusi menjadi sangat lambat atau bahkan akhirnya mengalami kegagalan (Ahmed, 2020). Proses SQL yang umumnya memiliki durasi lama (misalnya, diatas 12 jam), menjadi semakin lama durasinya, atau terpaksa terhenti karena faktor resource yang ternyata membutuhkan memori yang besar. Perhatikan gambar 1, untuk contoh keluhan *Job Apache Spark* yang melambat. Penyebab hal ini terjadi diantaranya adalah faktor *cluster* tempat instalasi Apache Spark, faktor efektifitas *SQL Query* yang dieksekusi, faktor alokasi sumber daya Apache Spark yang dijalankan, atau faktor-faktor lain. Khususnya jika faktor *cluster* lambat, maka bukan hanya satu dua pengguna yang merasakan, tapi bisa puluhan bahkan ratusan. Bukan hanya satu aplikasi yang terdampak, tapi banyak lagi aplikasi yang terganggu proses nya sehingga output yang diharapkan tidak bisa terselesaikan tepat waktu.

Ada kombinasi dari faktor-faktor diatas yang membuat *job* yang dieksekusi Apache Spark menjadi melambat atau gagal, membuat ide akan penggunaan Kecerdasan Buatan dalam bentuk *Machine Learning* untuk mengidentifikasi *job-job* bermasalah tersebut.

Rumusan masalah pada penelitian ini adalah apakah penggunaan *Machine Learning* dapat membantu menentukan apakah suatu *job* yang akan dieksekusi *Apache Spark* akan berjalan lancar atau bermasalah. Adapun Rumusan masalah pada penelitian ini terdapat tiga permasalahan yang akan diselesaikan yaitu :

1. Bagaimana mengidentifikasi *job* yang dieksekusi akan gagal sebagai dataset  
Identifikasi ini penting dilakukan sebelum penelitian *Machine Learning* berjalan, untuk dapat menentukan dataset yang akan digunakan. Pihak administrator umumnya memiliki *dashboard* tersendiri dan *logging* untuk keperluan ini?
2. Bagaimana membuat *Machine Learning* yang akan belajar dari dataset untuk kemudian mengidentifikasi *job-job* baru di Apache Spark, apakah kan gagal atau berhasil?
3. Faktor-faktor apa yang paling berpengaruh dalam *Machine Learning*?  
Jika dataset sudah ditentukan, maka selanjutnya akan ditentukan *feature* apa saja yang berpengaruh. Faktor-faktor apakah yang dipertimbangkan, dituangkan dalam bentuk *feature* yang nantinya akan diproses oleh model kecerdasan buatan (*Artificial Intelligence*) (Han, Jiawei, and Micheline Kamber, 2000). Lalu akan dilakukan iterasi *Machine Learning* sampai akhirnya dapat melakukan prediksi terhadap kondisi cluster. Apakah clusternya dalam kondisi baik ataukah bermasalah?

## 2. Kajian Terdahulu

### 2.1 Landasan Teori

#### a. Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. (Harianto Antonio, Novi Safriadi, 2012)

#### b. Basis Data

Basis Data adalah sekumpulan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi. (Ni Ketut Dewi Ari Jayanti, Ni Kadek Sumiari, 2018)

#### c. Apache Spark

Apache Spark merupakan mesin analitik terintegrasi untuk memproses data berukuran sangat besar. Spark menyediakan API untuk bahasa Java, Scala, Python dan R, dan mesin optimal yang mendukung eksekusi graph secara umum. Spark juga mendukung perangkat dengan level-tinggi seperti Spark SQL untuk SQL dan pemrosesan data terstruktur, Mllib untuk machine learning, GraphX untuk pemrosesan Graph, dan Streaming untuk komputasi inkremental dan pemrosesan streaming (Apache Spark Foundation, 2021)

#### d. *Feature Engineering*

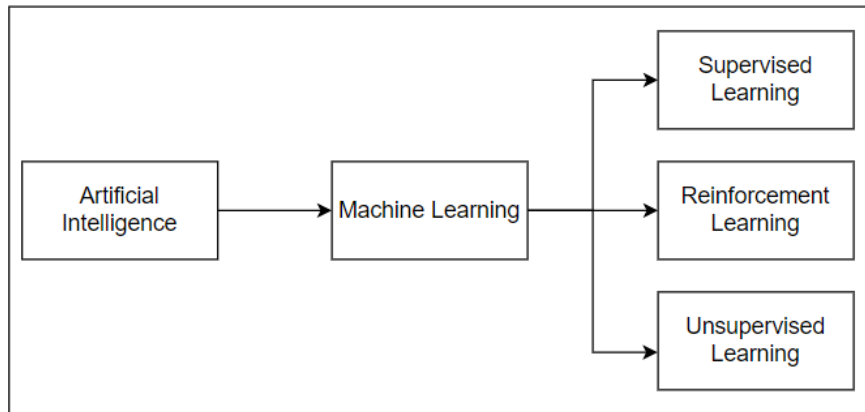
Feature Engineering adalah salah satu fitur utama dari machine learning untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. Feature Engineering juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi.

#### e. Kecerdasan Buatan

Machine learning dapat didefinisikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang (Goldberg & Holland, 1988). Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (training) dan pengujian (testing) (Huang, Zhu, & Siew, 2006).

Bidang machine learning berkaitan dengan pertanyaan tentang bagaimana membangun program komputer agar meningkat secara otomatis dengan berdasar dari pengalaman (Mitchell, 1997). Penelitian terkini mengungkapkan bahwa machine learning terbagi menjadi tiga kategori: Supervised Learning, Unsupervised Learning,

Reinforcement Learning (Somvanshi & Chavan, 2016). Skema keterkaitan artificial intelligence dan machine learning dapat dijelaskan dalam Gambar 1.



**Gambar 2 Skema Kecerdasan Buatan dan Machine Learning**

Teknik yang digunakan oleh *Supervised Learning* adalah metode klasifikasi di mana kumpulan data sepenuhnya diberikan label untuk mengklasifikasikan kelas yang tidak dikenal.

Pada teknik *Unsupervised Learning* sering disebut cluster dikarenakan tidak ada kebutuhan untuk pemberian label dalam kumpulan data dan hasilnya tidak mengidentifikasi contoh di kelas yang telah ditentukan. *Reinforcement Learning* biasanya berada antara *Supervised Learning* dan *Unsupervised Learning* (Board, 2017), teknik ini bekerja dalam lingkungan yang dinamis di mana konsepnya harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai (Das & Nene, 2017).

Metode supervised learning didasarkan pada kumpulan sampel data yang memiliki label. Kumpulan sampel digunakan untuk meringkas karakteristik distribusi ukuran perilaku dalam setiap jenis aplikasi sehingga membentuk model perilaku dari data (Amei, Huailin, Qingfeng, & Ling, 2011). Supervised learning dikelompokkan lebih lanjut dalam masalah klasifikasi dan regresi. Masalah klasifikasi adalah ketika variabel output berbentuk kategori, seperti merah atau biru atau penyakit dan tidak ada penyakit. Sedangkan masalah regresi adalah ketika variabel output adalah nilai riil, seperti dollar atau berat (Brownlee, 2016).

Supervised learning memiliki beberapa algoritma populer seperti Back-propagation (Negnevitsky, 2005), Linear regression, Random Forest, Support

Vector Machines (Brownlee, 2016), Naive Bayesian, Metode Rocchio, Decision Tree, k-Nearest Neighbor, Neural Network (Darujati & Gumelar, 2012), Logistic Regression, dan Neural Network (Lakshmi & Sheshasaayee, 2015). Kemudian beberapa algoritma untuk klasifikasi pun disebutkan dalam seperti Support Vector Machines (SVM), Normal Bayesian Classifier (NBC), K-Nearest Neighbor (KNN), Trees Gradient Boosted (GBT), Random Trees (RT), dan Artificial Neural Networks (ANN) (Židek, Pitel, & Hošovský, 2017).

Algoritma lainnya pun dibahas dalam (Athmaja, Hanumanthappa, & Kavitha, 2017) seperti Gaussian Mixture models, Hidden Markov Models, logistic regression, Kernel Regression, Deep neural networks, Deep belief networks, PCA, Kernel Perceptron.

## 2.2 Penelitian Terdahulu

Penelitian terdahulu tentang *Machine Learning* telah banyak dilakukan, namun sebagai bahan pertimbangan, perbandingan dan sumber referensi dalam penelitian ini maka penulis memilih beberapa penelitian terdahulu sebagai berikut:

**Tabel 1 Penelitian Terdahulu**

No	Nama	Judul	Tahun
1	Sunarya, Santoso, & Sentanu, 2015	Kecerdasan Buatan merupakan salah satu bidang dalam ilmu komputer yang ditujukan pada pembuatan software dan hardware yang dapat berfungsi sebagai sesuatu yang dapat berpikir seperti manusia	2015
2	Rahardja, Roihan, & others	Kecerdasan buatan banyak digunakan untuk memecahkan berbagai masalah seperti bisnis	2017
3	Russell & Norvig	Kecerdasan buatan juga banyak digunakan untuk memecahkan masalah seperti robotika, bahasa alami, matematika, game, persepsi, diagnosis medis, teknik, analisis keuangan, analisis sains, dan penalaran	2016
4	Sirait, E. R. E.	Implementasi Teknologi Big Data Di Lembaga Pemerintahan Indonesia.	2016
5	Windarto, A. P., Dewi, L. S., & Hartama, D.	Implementation of Artificial Intelligence in Predicting the Value of Indonesian Oil and Gas Exports With BP Algorithm	2017

### 3. Metodologi Penelitian

#### 3.1. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini dikumpulkan dari log sistem aplikasi Apache Spark di perusahaan XYZ. Selain itu, sebagai landasan teori dan referensi, data referensi SQL juga berasal dari ebook, jurnal, buku, observasi, dokumentasi platform, sistem platform dan berbagai informasi lain di Internet.

#### 3.2. Model Proses Prototyping

Model proses yang digunakan dalam penelitian ini adalah model proses prototyping, yaitu:

a. Komunikasi

Tahap komunikasi adalah sebuah tahapan dimana tim peneliti mengadakan komunikasi dengan responder atau nara sumber di tempat penelitian. Di tahap ini juga dilakukan pengumpulan yakni sebuah tahapan dimana tim peneliti mengambil data sumber atau data bukti dari sistem yang terkait.

b. Perencanaan cepat

Tahap perencanaan cepat adalah sebuah tahapan dimana dibuat perencanaan sumber daya dan fitur-fitur dalam aplikasi yang akan dirancang.

c. Pemodelan

Pemodelan menggunakan model desain pembelajaran ADDIE yang terdiri dari :

1. Analisis

Desain tahap analisis berfokus pada penentuan karakteristik dari dataset yang akan dijadikan bahan *learning* model. Juga ditentukan metrik-metrik yang akan terlibat di model.

2. Design

Tahap desain terkait dengan penentuan sasaran, konten, dan analisis yang terkait.

3. *Development*

Dalam tahan pengembangan dilakukan pembuatan dan penggabungan konten yang sudah dirancang pada tahapan desain.

4. Implementation

Fase ini, dibuat prosedur untuk implementasi variabel-variabel kedalam model *Machine Learning*

5. Evaluation

Tahap evaluasi melalui metrik untuk memastikan apakah masalah yang ada dapat diatasi dengan variabel-variabel dari hasil penelitian ini

## d. Konstruksi

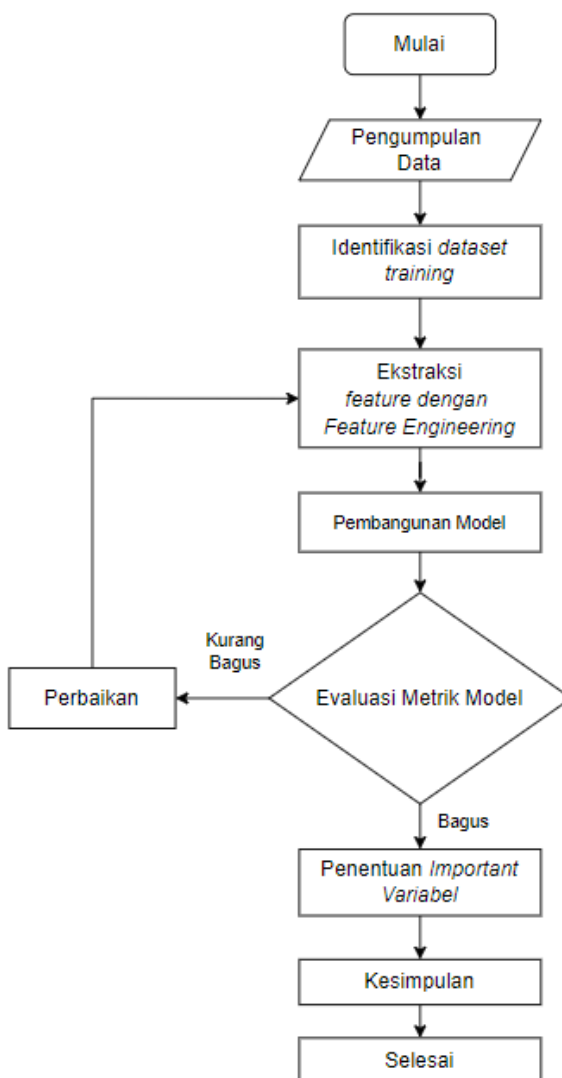
Tahap konstruksi variabel-variabel kedalam *feature engineering* sistem basis data

## e. Deployment

Pada tahap ini *Machine Learning* diuji coba pada kegiatan operasional organisasi.

### 3.3. Diagram Alir Penelitian

Tahapan proses yang akan dilakukan dalam penelitian ini digambarkan dalam diagram alir pada gambar 5 sebagai berikut :



Gambar 3 Diagram Alir Penelitian



## 4. Hasil Dan Pembahasan

### 4.1. Hasil

Penelitian ini bertujuan untuk mengidentifikasi *job* yang akan dieksekusi pada Apache Spark, apakah akan berakhir dengan lancar atau mengalami kegagalan. Untuk itu akan digunakan *Machine Learning* sebagai perangkat yang akan membantu implementasi identifikasi ini.

Dalam sebuah *Machine Learning*, akan ditentukan terlebih dahulu *dataset* yang akan dipelajari pola nya. . Jika kata kuncinya ditemukan di log job aplikasi spark, maka otomatis data tersebut akan menjadi dataset proses belajar model. Berikut adalah kata kunci yang digunakan untuk menyaring dataset dari keseluruhan *job* yang ada di Aplikasi Spark:

1. Reason[\:][\s](.\*)
2. ValueError[\:][\s](.\*)
3. TypeError[\:][\s](.\*)
4. (.)cancelled as part of cancellation of all jobs
5. (.)Web UI
6. java[\.](.\*)Exception[\:](.\*)
7. (.)spark.driver.maxResultSize(.\*)
8. Total size of serialized results(.\*)

```

1 regex_statement = f"""Reason[\:][\s](.*)|ValueError[\:][\s](.*)
2 |TypeError[\:][\s](.*)|(.)cancelled as part of cancellation of all jobs|(.)Web UI
3 |java[\.](.*)Exception[\:](.*)|(.)spark.driver.maxResultSize(.*)
4 |Total size of serialized results(.*)"""
5 def extract_reason(x):
6     search = re.search(regex_statement, str(x))
7     if search is not None :
8         return search.group(0)
9     else :
10        return None
11 reason_udf = udf(extract_reason)
12 df_spark_hr = df_spark_hr.withColumn(
13     'rgx_reason',
14     when(
15         df_spark_hr.stage_failurereason.isNotNull(),
16         reason_udf(f.col('stage_failurereason'))
17     ).otherwise(f.lit(None))
18 )

```

**Gambar 4** Proses Penentuan Dataset Belajar Model

Setelah memiliki dataset untuk proses belajar, berikutnya akan ditentukan parameter-parameter sebuah model. Untuk penelitian ini, parameter-parameter *Machine Learning* yang dijalankan adalah sebagai berikut :

Name	Value
Null_Value_Handling	median
feature_selection	True
imbalance_method	<class 'imblearn.combine._smote_tomek.SMOTETomek'>
n_estimators	20
n_feature	20
random_state	42
sampling_strategy	0.3
scale_pos_weight	2
test_size	0.2

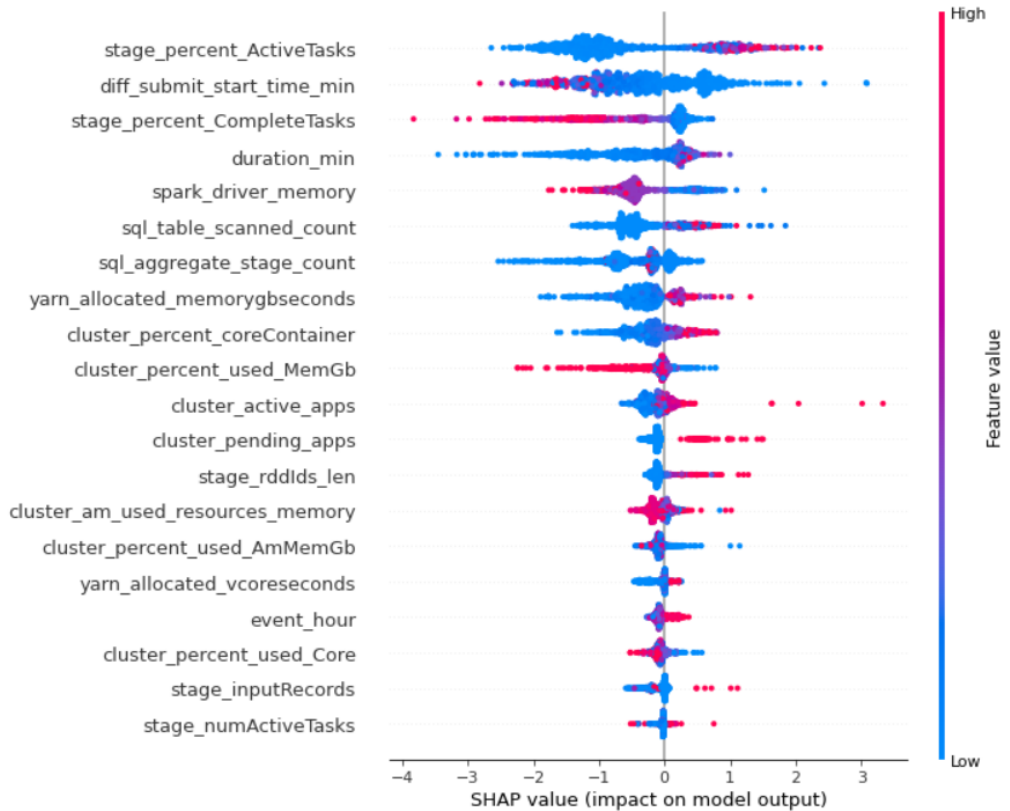
**Gambar 5 Parameter ML**

Definisi dari parameter *Machine Learning (ML)* yang digunakan :

**Tabel 2 Definisi Parameter Machine Learning (ML)**









No	Parameter	Value	Definisi Parameter
1	Null Value Handling	Median	<i>Treatment</i> terhadap nilai yang kosong yaitu dengan mengisi nilai median
2	Feature Selection	<i>True</i>	<i>Treatment</i> terhadap pemilihan <i>feature</i> yaitu dengan otomatis memilih
3	Imbalance method	SMOTETomek	Metode <i>Imbalance</i> yang dipilih
4	n_estimators	20	Jumlah estimators
5	n_feature	20	Jumlah <i>Important Variabel / Feature</i>
6	Random state	42	Nilai Urut Random yang digunakan
7	Sampling strategy	0.3	Strategi Sampling digunakan 3/10
8	Scale_pos_weight	2	Nilai besaran yang diskala kan
9	Test_size	0.2	Besar dataset yang akan diujikan

Sebagai data awal, jumlah *feature* yang akan dianalisa adalah sebanyak 83 *features* (lampiran 2). Untuk menentukan kontribusi dari masing-masing fitur kepada prediksi yang dilakukan, maka digunakan perhitungan nilai SHAP. Dari hasil iterasi yang dilakukan maka didapatkan SHAP value seperti dibawah ini



**Gambar 6 SHAP Value**

Sebagai bahan evaluasi model, digunakan beberapa metrik seperti dibawah ini :

Name	Value
ACCURACY 	0.994
AUC 	0.958
F1-Score 	0.8
F2-Score 	0.8
PRECISION 	0.8
RECALL 	0.8
SENSITIVITY 	0.8
SPECITIFITY 	0.997
fn 	3
fp 	3
tn 	931
tp 	12

**Gambar 7 Evaluation Metrics**

## 4.2. Pembahasan

Berdasarkan model yang di eksekusi dan nilai SHAP, maka faktor-faktor yang mempengaruhi (*feature important*) adalah sebagai berikut :

Tabel 3 Feature Important ML Model

No	Features	Tipe Data	Deskripsi Features
1	<b>Yarn_allocated_memorygbseconds</b>	Long Type	Lama eksekusi job sampai saat diambil log
2	<b>Cluster_percent_coreContainer</b>	Integer	Jumlah <i>core Container</i> yang digunakan oleh <i>job</i>
3	<b>Cluster_percent_used_memGb</b>	Long Type	Jumlah memori yang digunakan oleh job
4	<b>Cluster_percent_used_core</b>	Long Type	Jumlah <i>core</i> yang digunakan oleh job
5	<b>Stage_inputrecords</b>	Integer	Jumlah data (baris) yang digunakan sebagai input

6	<b>Cluster_active_apps</b>	Integer	Jumlah aplikasi ( <i>job</i> ) yang sedang aktif di <i>pool resource</i> yang sama
7	<b>Cluster_pending_apps</b>	Integer	Jumlah aplikasi ( <i>job</i> ) yang masih status <i>pending</i> di <i>pool resource</i> yang sama
8	<b>stage_rddIds_len</b>	Integer	Banyakny tahapan dari <i>RDD</i> yang akan dijalankan
9	<b>Cluster_am_used_resources_memory</b>	Long Type	Memori yang digunakan di <i>Application Master</i>
10	<b>Cluster_percent_used_ammemgb</b>	Long Type	Persentase memori yang digunakan di <i>Application Master</i>
11	<b>Yarn_allocated_vcoreseconds</b>	Integer	Waktu yang dialokasi untuk sebuah <i>job</i> Apache Spark dieksekusi
12	<b>Event_hour</b>	Integer	Waktu ambil <i>log</i> untuk analisa
13	<b>stage_percent_ActiveTasks</b>	Long Type	Persentase <i>task</i> yang sedang aktif ( <i>running</i> ) dibanding total keseluruhan task
14	<b>Diff_submit_start_time_min</b>	Long Type	Durasi waktu antara waktu <i>submission</i> dikurangi dengan waktu mulai <i>job</i> aplikasi Apache Spark
15	<b>stage_percent_CompleteTasks</b>	Long Type	Persentase tahapan yang sudah selesai
16	<b>stage_duration_min</b>	Long Type	Durasi tahapan eksekusi
17	<b>Spark_driver_memory</b>	Integer	Memori yang digunakan oleh <i>job</i> Apache Spark

18	<b>Sql_table_scanned_count</b>	Integer	Jumlah tabel yang terlibat di <i>job</i> Apache Spark
19	<b>Sql_aggregate_stage_count</b>	Integer	Jumlah tahapan yang terlibat di proses agregasi
20	<b>Stage_numactivetasks</b>	Integer	Jumlah tahapan yang sedang aktif

Berikut contoh datanya :

Tabel 4 Contoh Nilai Feature

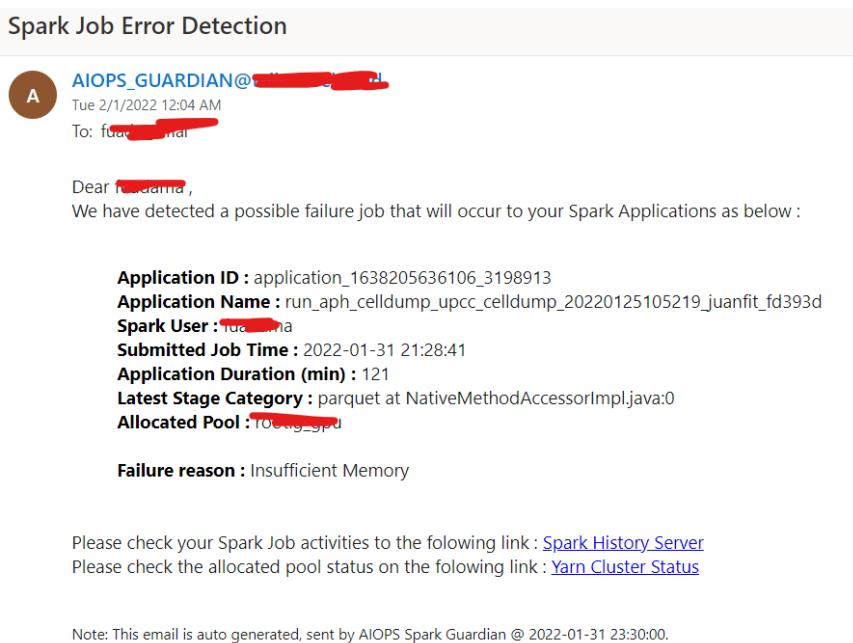
<b>No</b>	<b>Features</b>	<b>Contoh Nilai</b>
1	<b>Yarn_allocated_memorygbseconds</b>	29311242.92
2	<b>Cluster_percent_coreContainer</b>	25
3	<b>Cluster_percent_used_memGb</b>	86.21121212
4	<b>Cluster_percent_used_core</b>	51.84615385
5	<b>Stage_inputrecords</b>	0
6	<b>Cluster_active_apps</b>	2
7	<b>Cluster_pending_apps</b>	0
8	<b>stage_rddIds_len</b>	7
9	<b>Cluster_am_used_resources_memory</b>	11.72
10	<b>Cluster_percent_used_ammemgb</b>	0.529815695
11	<b>Yarn_allocated_vcoreseconds</b>	508539
12	<b>Event_hour</b>	5
13	<b>stage_percent_ActiveTasks</b>	3.343465046
14	<b>Diff_submit_start_time_min</b>	33.45
15	<b>stage_percent_CompleteTasks</b>	10.94224924
16	<b>stage_duration_min</b>	33.16666667
17	<b>Spark_driver_memory</b>	30
18	<b>Sql_table_scanned_count</b>	0
19	<b>Sql_aggregate_stage_count</b>	6
20	<b>Stage_numactivetasks</b>	7

Berdasarkan SHAP Value, ada 6 *feature* teratas yang paling berkontribusi kepada prediksi yang dilakukan yaitu Stage Percent Active Tasks, Diff submit start time min, Stage percent complete task, duration min, Spark Driver memory, Sql table scanned count. Faktor-faktor tersebut dapat diartikan jika makin lama suatu *job* berjalan,

banyaknya tabel (besar) yang terlibat dan masih tersisa banyak *stage* yang akan dieksekusi, membuat probabilitas suatu *job* gagal semakin besar.

Ditambah dengan faktor okupansi dari *cluster* dan alokasi memori yang ditentukan *job*, menjadi faktor penyumbang lain saat menentukan apakah *job* Aplikasi Spark akan menemui kegagalan ataukah lancar. Selain itu, faktor lain yang disorot juga kecepatan eksekusi suatu tahapan *job*, makin lama jeda antara waktu *submission* dikurangi dengan waktu mulai *job* aplikasi Apache Spark, mengindikasikan jika kondisi *cluster* juga tidak dalam performa yang optimal.

Sebagai tambahan, setelah model *Machine Learning* dieksekusi, dan ditentukan *job-job* yang berpotensi besar untuk gagal, selanjutnya akan dikeluarkan notifikasi ke pemilik *job* tersebut dalam bentuk email.



**Gambar 8** Contoh Email dari *Machine Learning* Model

## 5. Kesimpulan

### 5.1. Hasil

Berdasarkan penelitian dan hasil analisis yang dilakukan, dapat diperoleh kesimpulan sebagai berikut :

1. Untuk mengidentifikasi *job* yang gagal untuk selanjutnya dijadikan dataset proses *training* si model, digunakan kata kunci tertentu seperti
  - a. Reason[\:][\s](.\*)
  - b. ValueError[\:][\s](.\*)

- c. `TypeError[\\:][\\s](.*)`
  - d. `(.*)cancelled as part of cancellation of all jobs`
  - e. `(.*)Web UI`
  - f. `java[\\.](.*)Exception[\\:](.*)`
  - g. `(.*)spark.driver.maxResultSize(.*)`
  - h. `Total size of serialized results(.*)`
2. *Machine Learning* yang dibuat dalam penelitian ini melibatkan penentuan parameter ML, Metrik Evaluasi dan SHAP Value untuk menentukan prediksi dari Model
  3. Dari SHAP Value, ditentukan 20 *important variable* yang dapat dibagi menjadi beberapa kategori yaitu :
    - a. Waktu eksekusi mulai dan berjalan
    - b. Tabel yang terlibat
    - c. Proses stage yang terlibat (
    - d. Alokasi *resource* yang digunakan (memori dan vcores)
    - e. Okupansi *cluster*
    - f. Proses Aggregasi

Seluruh variabel yang dihasilkan dari penelitian ini akan dimasukkan kedalam model *Machine Learning*. Tujuan utamanya adalah untuk bisa menentukan apakah *job* Aplikasi Spark yang diproses akan berhasil atau gagal di tengah proses. Sebagai tambahan, setelah model *Machine Learning* dieksekusi, dan ditentukan *job-job* yang berpotensi besar untuk gagal, selanjutnya akan dikeluarkan notifikasi ke pemilik *job* tersebut dalam bentuk email.

## 5.2. Saran

1. Obyek penelitian ini diharapkan dapat diperluas pada lingkungan *Cloud*, karena perkembangan berbasis *Cloud* saat ini naik sangat signifikan, terutama pasca pandemi Covid 19
2. Bagi peneliti selanjutnya, saran yang dapat diberikan berkaitan dengan penelitian ini adalah penambahan analisis No SQL database. Teknologi No SQL berbeda dengan yang SQL mampu lakukan. Kekurangan dan kelebihan bisa menjadi bahasan tersendiri bagaimana sampai akhirnya eksekusi bisa berjalan lebih cepat dan optimal.
3. Penelitian selanjutnya juga diharapkan menggunakan lebih banyak sumber untuk mencari *features* yang relevan dengan *job* dari Apache Spark.



## 6. Ucapan Terima Kasih

Kami berharap semoga penelitian ini dapat digunakan sebagai informasi yang dapat menambah wawasan tentang cara merancang *feature engineering* yang efektif sebelum membangun model *machine learning*, serta berguna bagi pihak-pihak yang terkait dan dapat digunakan sebagai referensi untuk penelitian selanjutnya.

Penelitian ini dapat kami selesaikan berkat bantuan dari berbagai pihak yang mendukung, untuk itu tidak lupa kami ucapkan terima kasih kepada:

1. Rektor Universitas Dian Nusantara
2. Direktur LRPM Universitas Dian Nusantara
3. Dekan Fakultas Teknik Universitas Dian Nusantara
4. Kaprodi Teknik Informatika Universitas Dian Nusantara
5. Seluruh pihak yang tidak dapat kami sebutkan satu persatu yang telah membantu penelitian ini.

## 7. Pernyataan Penulis

Penulis menyatakan bahwa tidak ada konflik kepentingan terkait publikasi artikel ini. Penulis menegaskan bahwa data dan makalah bebas dari plagiarisme.

## Bibliografi

- Apache Hadoop*. (2022). Diambil kembali dari <http://hadoop.apache.org/>.
- Apache Spark*. (2022). Diambil kembali dari <https://spark.apache.org/>.
- Armbrust, M., Huai, Y., Liang, C., Xin, R., & Zaharia, M. (2015, April 13). *Deep Dive into Spark SQL's Catalyst Optimizer*. Diambil kembali dari <https://databricks.com:https://databricks.com/blog/2015/04/13/deep-dive-into-spark-sqls-catalyst-optimizer.html>
- Chanowich, E. d. (2001). *Query Optimization Advanced*.
- Goutam, S. (2021, Februari 12). *Apache Spark Logical And Physical Plans*. Diambil kembali dari <https://blog.clairvoyantsoft.com/:https://blog.clairvoyantsoft.com/spark-logical-and-physical-plans-469a0c061d9e>
- Han, J. a. (2000). *Data Mining Concepts & Techniques*. Morgan Kaufmann Publishers.
- Harianto Antonio, Novi Safriadi. (2012). Rancang Bangun Sistem Informasi Administrasi Informatika.

- Korth, H. d. (1991). *Database System Concepts*. Singapura: McGraw Hill.
- Leturgez, L. (2020, Juli 23). *Spark's Logical and Physical plans ... When, Why, How and Beyond*. Diambil kembali dari <http://www.medium.com:https://medium.com/datalex/sparks-logical-and-physical-plans-when-why-how-and-beyond-8cd1947b605a>
- Ni Ketut Dewi Ari Jayanti, Ni Kadek Sumiari. (2018). *Teori Basis Data*. Yogyakarta: Penerbit ANDI.
- Rahardja, U. R. (2017). Design of Business Intelligence in Learning Systems Using iLearning Media. *Universal Journal of Management*, 227-235.
- Russell, S. J. (2016). *Artificial Intelligence : a modern approach*. Malaysia: Pearson Education Limited.
- Sunarya, A. S. (2015). Sistem Pakar Untuk Mendiagnosa Gangguan Jaringan Lan. *CCIT*, 8(2), 1-11.
- Wahono, R. S. (2014, Januari 10). [romisatriawahono.net/2014/01/10/kontribusi-penelitian-dan-perbaikan-metode/](http://romisatriawahono.net/2014/01/10/kontribusi-penelitian-dan-perbaikan-metode/). Diambil kembali dari [romisatriawahono.net:https://romisatriawahono.net/2014/01/10/kontribusi-penelitian-dan-perbaikan-metode/](http://romisatriawahono.net:https://romisatriawahono.net/2014/01/10/kontribusi-penelitian-dan-perbaikan-metode/)
- Y. Bengio, A. C. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1798–1828