# LAPORAN FINAL PROJECT

## DATA SCIENCE

# Prediksi Kelangsungan Hidup Penumpang Titanic: Peningkatan Performa Model Dengan Pemilihan Fitur dan Metode Penyeimbangan Data


**Disusun oleh:**

**Viga Laksa Hardjanto**

**PROGRAM STUDI MAGISTER ILMU KOMPUTER**

**DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS GADJAH MADA**

**YOGYAKARTA**


**2024**

# 1. Pendahuluan

## 1.1 Latar Belakang

Bidang pembelajaran mesin meningkatkan kemauan para ahli untuk menemukan informasi berguna dan sejumlah pengetahuan dari peristiwa masa lalu. Salah satu peristiwa paling terkenal dalam Sejarah dunia adalah tenggelamnya kapal Titanic. Penelitian ini bertujuan untuk menjelajahi dan menganalisis data penumpang kapal Titanic, yang merupakan salah satu tragedi paling terkenal dalam sejarah. Ketika kapal mewah ini tenggelam pada tanggal 15 April 1912 setelah bertabrakan dengan gunung es di Atlantik Utara, lebih dari 1500 orang kehilangan nyawa mereka. Dalam upaya untuk memahami lebih baik faktor-faktor yang mempengaruhi kelangsungan hidup dalam situasi darurat semacam ini, penelitian ini menggunakan metode machine learning. Data yang dikumpulkan dari penumpang Titanic meliputi berbagai informasi, seperti kelas tiket, usia, jenis kelamin, jumlah anggota keluarga, dan lainnya. Dengan menggunakan teknik machine learning, penelitian ini bertujuan untuk memanfaatkan data sejarah untuk membangun model prediksi yang dapat menentukan kemungkinan seseorang untuk selamat dalam kejadian serupa. Tujuan akhirnya adalah untuk mengidentifikasi pola dan hubungan antara berbagai variabel ini dengan kelangsungan hidup, sehingga memberikan wawasan yang berharga tentang faktor-faktor yang memengaruhi keselamatan individu dalam bencana laut. Dengan demikian, penelitian ini tidak hanya berpotensi memberikan informasi tentang tragedi Titanic itu sendiri, tetapi juga memberikan pandangan yang lebih luas tentang perilaku manusia dalam situasi darurat dan bagaimana faktor-faktor tertentu dapat mempengaruhi hasil akhir. Dengan menggunakan pendekatan machine learning untuk menganalisis data historis ini, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan bagi pemahaman kita tentang respons manusia dalam menghadapi bencana dan memberikan landasan bagi upaya-upaya pencegahan dan mitigasi di masa depan.

## 1.2 Rumusan Masalah

Membuat serta meningkatkan akurasi model machine learning menggunakan teknik seleksi fitur dan penanganan imbalance data.

## 1.3 Batasan Masalah

- Dataset yang digunakan berjudul titanic dataset yang diperoleh dari Kaggle
- Classifier yang digunakan dalam penelitian adalah Random Forest

## 1.4 Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan untuk mengembangkan sebuah model prediktif keselamatan penumpang transportasi laut. Secara khusus dapat dijelaskan sebagai berikut:

- Mengidentifikasi faktor resiko yang paling signifikan dalam keselamatan penumpang
- Membangun model prediktif yang lebih akurat

## 1.5 Manfaat Penelitian

Manfaat penelitian ini dapat dijelaskan sebagai berikut:

- Meningkatkan ketepatan prediksi keselamatan penumpang dengan mengidentifikasi subset fitur yang paling relevan, model prediktif yang dikembangkan memiliki potensi untuk memberikan hasil yang lebih akurat dalam menentukan kemungkinan seseorang selamat dari kejadian kecelakaan laut
- Seleksi fitur yang tepat tentunya dapat mengurangi jumlah fitur dalam pembangunan model prediktif sehingga mengurangi kompleksitas komputasi dan mempercepat daya komputasi yang dibutuhkan.
- Mengidentifikasi faktor-faktor penting yang dapat memberikan wawasan mendalam terkait kecelakaan laut

# 2. Landasan Teori

Dataset Titanic berisi informasi tentang penumpang kapal Titanic yang tenggelam pada tahun 1912. Dataset ini diperoleh dari situs *Kaggle* yang selanjutnya digunakan sebagai bahan untuk memahami faktor-faktor yang memengaruhi keselamatan penumpang di kapal Titanic. Dalam dataset Titanic terdapat kolom-kolom seperti:

- PassengerId: ID unik untuk setiap penumpang
- Survived: Menunjukkan apakah penumpang selamat (1) atau tidak (0)
- Pclass: Kelas tiket penumpang (1st, 2nd, 3rd)
- Name: Nama penumpang
- Sex: Jenis kelamin penumpang
- Age: Usia penumpang
- SibSp: Jumlah saudara atau pasangan di kapal
- Parch: Jumlah orang tua/anak di kapal
- Ticket: Nomor tiket

- Fare: Harga tiket
- Cabin: Nomor kabin
- Embarked: Pelabuhan keberangkatan (C = Cherbourg, Q = Queenstown, S = Southampton)

Random Forest Classifier merupakan salah satu algoritma dalam machine learning yang termasuk dalam kelompok ensemble learning. Ensemble learning adalah pendekatan di mana beberapa model yang lemah digabungkan untuk membentuk model yang lebih kuat. Random Forest Classifier memanfaatkan konsep ini dengan membangun sejumlah besar pohon keputusan secara acak selama proses pelatihan. Setiap pohon keputusan dihasilkan dari sampel acak dari data pelatihan dan fitur-fiturnya. Ketika datanya diprediksi, setiap pohon memberikan suara untuk kelas tertentu, dan kelas dengan suara terbanyak dipilih sebagai prediksi akhir model. Keunggulan utama dari Random Forest adalah kemampuannya untuk mengatasi masalah overfitting yang sering terjadi pada pohon keputusan tunggal. Selain itu, algoritma ini juga mampu menangani data yang tidak seimbang dengan baik dan dapat memberikan estimasi kepentingan fitur, yang memungkinkan untuk interpretasi hasil yang lebih baik. Dengan kombinasi dari berbagai pohon keputusan yang dihasilkan secara acak, Random Forest Classifier menjadi salah satu pilihan yang populer dan efektif dalam berbagai aplikasi, termasuk klasifikasi pada dataset yang kompleks dan besar.

Seleksi fitur dalam machine learning adalah proses penting untuk memilih subset fitur yang paling relevan dan informatif dari dataset asli. Tujuan utamanya adalah untuk meningkatkan kinerja model dengan mengurangi dimensi data, menghilangkan noise, dan memperbaiki generalisasi model. Terdapat berbagai metode seleksi fitur yang dapat diterapkan, termasuk metode filter, wrapper, dan embedded. Metode filter melakukan seleksi fitur berdasarkan statistik seperti korelasi, uji chisquare dan lainnya. Metode wrapper menggunakan algoritma pembungkus (*wrapper*) seperti forward selection, backward elimination, atau metode berbasis pencarian lainnya untuk mengevaluasi subset fitur dengan memperhitungkan kinerja model yang sebenarnya. Sementara itu, metode embedded melakukan seleksi fitur sebagai bagian dari proses pembelajaran model itu sendiri, seperti pada algoritma Random Forest atau regresi LASSO (Least Absolute Shrinkage and Selection Operator). Pentingnya seleksi fitur terutama terlihat dalam situasi di mana dataset memiliki banyak fitur atau dimensi yang tinggi, di mana penggunaan semua fitur dapat menyebabkan overfitting, memperlambat proses pembelajaran, atau menghasilkan model yang sulit diinterpretasi. Dengan memilih fitur-fitur yang paling

relevan dan informatif, seleksi fitur membantu meningkatkan efisiensi, interpretabilitas, dan kualitas prediksi model machine learning.

Dataset yang tidak seimbang (imbalanced dataset) merupakan kondisi di mana jumlah sampel atau observasi dalam satu kelas atau kategori lebih dominan dibandingkan dengan kelas atau kategori lainnya. Masalah utama yang timbul dari ketidakseimbangan dataset adalah bias yang mungkin terjadi dalam pembelajaran model, di mana model cenderung memprioritaskan kelas mayoritas dan mengabaikan kelas minoritas. Hal ini dapat mengakibatkan kinerja yang buruk dalam memprediksi kelas minoritas atau penting. Beberapa strategi yang umum digunakan untuk mengatasi ketidakseimbangan dataset termasuk resampling data, baik dengan oversampling (menggandakan sampel kelas minoritas) atau undersampling (mengurangi sampel kelas mayoritas), atau dengan menggunakan algoritma pembelajaran yang telah dimodifikasi untuk menangani ketidakseimbangan, seperti algoritma SMOTE (Synthetic Minority Over-sampling Technique) atau algoritma ensemble yang dikombinasikan dengan teknik pemodelan kelas minoritas. Pentingnya pengelolaan ketidakseimbangan dataset adalah untuk memastikan bahwa model yang dihasilkan mampu memberikan prediksi yang baik dan tidak bias terhadap kelas minoritas, sehingga memberikan keputusan atau solusi yang lebih seimbang dan akurat.

## 3. Desain dan Metodologi Pengembangan Model

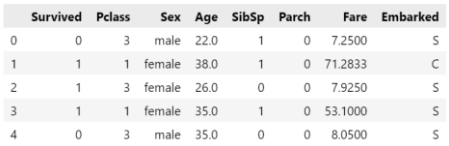
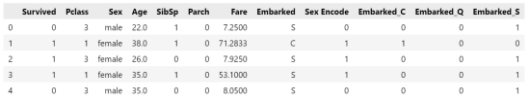
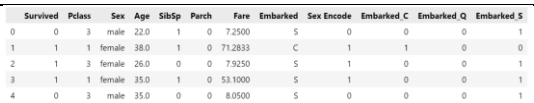
### 3.1 Pre-Processing

Pada tahap ini, Pre-processing wajib dilakukan dikarenakan perlunya penyesuaian beberapa hal. Dataset yang tersedia memiliki 12 kolom dengan tipe data yang beragam. Informasi setiap kolom disajikan pada gambar di bawah ini

```
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
```

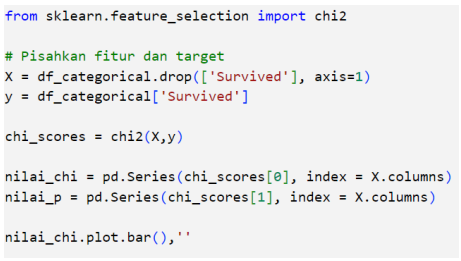
*Gambar 3. 1*

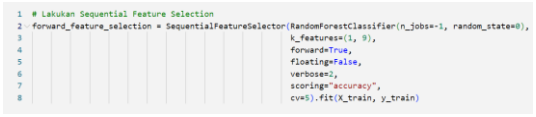
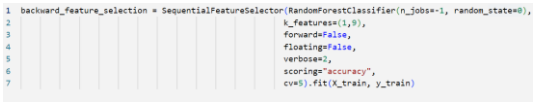
Dapat dilihat bahwa terdapat kolom yang memilki missing values yang cukup signifikan, yakni kolom age dan cabin. Namun perlu diingat bahwa tidak semua kolom diperlukan, adapun kolom-kolom yang tidak diperlukan dalam penelitian ini adalah PassengerId, Name, Ticket, Cabin. Detail tahap pre-processing dapat dilihat pada tabel di bawah ini

| Pre-Processing | Penjelasan |
|---|---|
|  | Menghapus kolom yang tidak diperlukan, yakni PassengerId, Name, Ticket, Cabin |
|  | Melakukan encode terhadap sejumlah kolom bertipe data kategorikal seperti sex dan embarked. One hot encoding diterapkan untuk pada kolom embarked. Adapun pada kolom sex, encode dilakukan dengan cara mapping, male bernilai 0 sedangkan female bernilai 1 |
|  | Mengisi nilai yang kosong pada kolom age dengan nilai modus |
|  | Hasil akhir tahap pre-processing |

### 3.2 Proses Seleksi Fitur

Seleksi fitur dilakukan menggunakan metode filter, wrapper, dan embedded. Lebih jelas tentang proses seleksi fitur dapat dilihat pada tabel di bawah ini.

| Seleksi Fitur | Penjelasan |
|---|---|
|  | Uji Chi Square digunakan untuk menentukan apakah ada hubungan yang signifikan antara variable-variable kategorikal. Dalam kasus ini yang berperan menjadi variable kategorikal adalah Survived,Pclass, Sex Encode, Embarked_C, Embarked_Q, Embarked_S |

| | |
|---|---|
|  | Korelasi digunakan untuk mendapatkan wawasan yang lebih baik tentang bagaimana variabel numerik berinteraksi dan saling mempengaruhi. Dalam kasus ini yang berpedan sebagai variable numerik adalah Age, SibSp,Parch,Fare |
|  | Implementasi forward selection di mana setiap fitur secara bertahap ditambahkan ke dalam model. Perlu diperhatikan untuk menambahkan seed dengan tujuan untuk mendapatkan hasil yang konsisten setiap kali program di jalankan |
|  | Implementasi backward selection merupakan kebalikan dari forward selection, yakni setiap fitur secara bertahap dihapus dari model |
|  | Proses exhaustive search melibatkan pembuatan dan evaluasi model untuk setiap kombinasi fitur yang mungkin. Exhaustive search feature selection memastikan bahwa tidak ada kombinasi fitur yang terlewatkan, dengan konsekuensi waktu eksekusi dan biaya komputasi yang mahal |

### 3.3 Train Test Split

Dilakukan pembagian dataset menjadi subset pelatihan (train) dan subset pengujian (test) dengan rasio 80% untuk pelatihan dan 20% untuk pengujian. Penggunaan nilai random state pada pembagian ini membantu dalam menjaga konsistensi hasil, sehingga melalui mekanisme ini dibagi dengan cara yang sama setiap kali proses pembagian dilakukan.

### 3.4 Imbalance Dataset

Data Titanic terdiri dari 891 baris yang selanjutnya dilakukan proses train test split data seperti yang telah dijelaskan pada bagian sebelumnya. Pada *data train*, terlihat bahwa persentase penumpang yang *survived* sebesar 37,6%, sementara yang *not survived* sebesar 62,4%. Diasumsikan bahwa dengan persentase tersebut data yang tersedia termasuk dalam kategori *imbalance dataset*. Untuk memastikan keberagaman data pada *data test*, dilakukan *balancing data* pada *data train*. Teknik yang digunakan adalah *random oversampling*, di mana jumlah sampel dari kelas minoritas ditingkatkan secara acak dengan mengambil sampel acak dari kelas minoritas hingga jumlahnya sebanding dengan kelas mayoritas. Hal ini bertujuan untuk meningkatkan keberagaman data dan memastikan model tidak bias terhadap kelas mayoritas.

## 4. Hasil dan Pembahasan

Berdasarkan eksperimen yang telah dilakukan didapatkan hasil seperti tabel di bawah ini.

*Tabel 4. 1 Perbandingan Paper Referensi dengan Eksperimen*

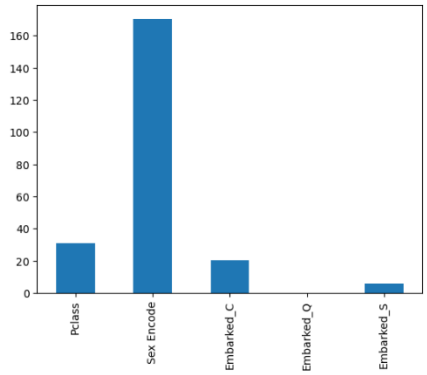
| Sumber | Model Yang Digunakan | Akurasi Terbaik (%) |
|---|---|---|
| Paper Referensi | Linear SVM | 80,3 |
| Eksperimen | Random Forest (Baseline) | 84,9 |

Random Forest digunakan sebagai model baseline sudah terdapat peningkatan dari sisi akurasi terhadap eksperimen dari paper referensi

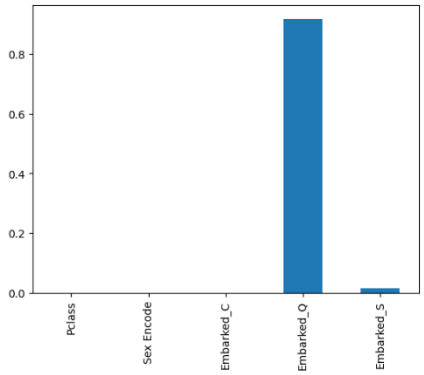
**Proses Seleksi Fitur**

**Metode Filter**

### a. Chi Square (data kategorikal)




*Gambar 4. 1 Nilai Chi*

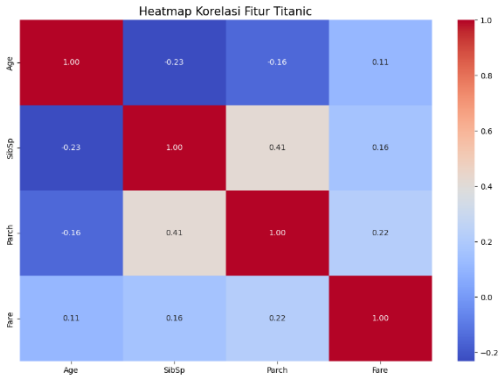



*Gambar 4. 2 Nilai p*

Gambar di atas menunjukkan bahwa fitur Embarked_Q dinilai paling tidak signifikan terhadap kinerja model, hal ini ditandai dengan hasil nilai chi yang sangat rendah pada Gambar

4.1 dan nilai p yang sangat tinggi pada Gambar 4.2 sehingga dengan metode ini fitur Embarked_Q dihapus.

### b. Correlation (data numerik)




*Gambar 4. 3*

Korelasi digunakan untuk melihat hubungan antar fitur dengan tipe data numerik. Gambar 4.3 menunjukkan bahwa pada setiap fitur numerik tidak terdapat nilai yang begitu signifikan, umumnya apabila terdapat nilai diatas 0,50 salah satu dari kedua kombinasi fitur dapat dihapus sehingga dengan ini data numerik tidak ada yang perlu dihapus.

## Metode Wrapper

### a. Forward Selection

| | feature_idx | cv_scores | avg_score | feature_names | ci_bound | std_dev | std_err |
|---|---|---|---|---|---|---|---|
| 1 | (5,) | [0.7482517482517482, 0.8181818181818182, 0.767... | 0.786526 | (Sex Encode,) | 0.035298 | 0.027463 | 0.013731 |
| 2 | (2, 5) | [0.7622377622377622, 0.8321678321678322, 0.767... | 0.794937 | (SibSp, Sex Encode) | 0.036725 | 0.028574 | 0.014287 |
| 3 | (2, 5, 8) | [0.7622377622377622, 0.8321678321678322, 0.760... | 0.793529 | (SibSp, Sex Encode, Embarked_S) | 0.038588 | 0.030023 | 0.015011 |
| 4 | (0, 2, 5, 8) | [0.7762237762237763, 0.8601398601398601, 0.767... | 0.796287 | (Pclass, SibSp, Sex Encode, Embarked_S) | 0.042689 | 0.033213 | 0.016607 |
| 5 | (0, 1, 2, 5, 8) | [0.7552447552447552, 0.8391608391608392, 0.788... | 0.803388 | (Pclass, Age, SibSp, Sex Encode, Embarked_S) | 0.043641 | 0.033954 | 0.016977 |
| 6 | (0, 1, 2, 4, 5, 8) | [0.7832167832167832, 0.7972027972027972, 0.802... | 0.800591 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.031324 | 0.024371 | 0.012186 |
| 7 | (0, 1, 2, 4, 5, 7, 8) | [0.7692307692307693, 0.8041958041958042, 0.809... | 0.799192 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.032246 | 0.025088 | 0.012544 |
| 8 | (0, 1, 2, 4, 5, 6, 7, 8) | [0.7552447552447552, 0.8181818181818182, 0.802... | 0.794967 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.033668 | 0.026195 | 0.013098 |
| 9 | (0, 1, 2, 3, 4, 5, 6, 7, 8) | [0.7622377622377622, 0.7972027972027972, 0.816... | 0.796395 | (Pclass, Age, SibSp, Parch, Fare, Sex Encode, ... | 0.032806 | 0.025524 | 0.012762 |

*Gambar 4. 4 Hasil Forward Selection*

Forward selection dimulai dengan model kosong dan secara bertahap menambahkan satu fitur pada setiap iterasi. Pada setiap langkah, model dievaluasi dan fitur yang memberikan peningkatan kinerja yang paling signifikan (misalnya, berdasarkan validasi silang) ditambahkan ke dalam model. Proses ini berlanjut hingga tidak ada peningkatan kinerja yang signifikan saat menambahkan fitur baru. Pada gambar 4.4 ditunjukkan bahwa hasil paling

maksimal menggunakan fitur Pclass, Age, SibSp, Sex Encode, Embarked_S dengan akurasi *data test* 82,1%.

## b. Backward Selection

| | feature_idx | cv_scores | avg_score | feature_names | ci_bound | std_dev | std_err |
|---|---|---|---|---|---|---|---|
| 9 | (0, 1, 2, 3, 4, 5, 6, 7, 8) | [0.7622377622377622, 0.7972027972027972, 0.816... | 0.796395 | (Pclass, Age, SibSp, Parch, Fare, Sex Encode, ... | 0.032806 | 0.025524 | 0.012762 |
| 8 | (0, 1, 2, 4, 5, 6, 7, 8) | [0.7552447552447552, 0.8181818181818182, 0.802... | 0.794967 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.033668 | 0.026195 | 0.013098 |
| 7 | (0, 1, 2, 4, 5, 7, 8) | [0.7692307692307693, 0.8041958041958042, 0.809... | 0.799192 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.032246 | 0.025088 | 0.012544 |
| 6 | (0, 1, 2, 4, 5, 7) | [0.7832167832167832, 0.8041958041958042, 0.816... | 0.80199 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.026764 | 0.020823 | 0.010412 |
| 5 | (0, 1, 4, 5, 7) | [0.8041958041958042, 0.8181818181818182, 0.795... | 0.811799 | (Pclass, Age, Fare, Sex Encode, Embarked_Q) | 0.028812 | 0.022417 | 0.011208 |
| 4 | (0, 1, 4, 5) | [0.7972027972027972, 0.8111888111888111, 0.816... | 0.814636 | (Pclass, Age, Fare, Sex Encode) | 0.031346 | 0.024388 | 0.012194 |
| 3 | (0, 4, 5) | [0.8391608391608392, 0.8181818181818182, 0.746... | 0.804708 | (Pclass, Fare, Sex Encode) | 0.041485 | 0.032277 | 0.016138 |
| 2 | (4, 5) | [0.8041958041958042, 0.8251748251748252, 0.718... | 0.787846 | (Fare, Sex Encode) | 0.047123 | 0.036663 | 0.018332 |
| 1 | (5,) | [0.7482517482517482, 0.8181818181818182, 0.767... | 0.786526 | (Sex Encode,) | 0.035298 | 0.027463 | 0.013731 |

*Gambar 4. 5 Hasil Backward Selection*

Backward selection dimulai dengan model yang berisi semua fitur yang tersedia. Pada setiap iterasi, satu fitur dihapus dari model. Kinerja model dievaluasi setelah menghapus fitur tersebut, dan fitur yang jika dihapus memberikan penurunan kinerja yang paling kecil (atau bahkan meningkatkan kinerja) dipertahankan. Proses ini berlanjut hingga tidak ada penurunan kinerja yang signifikan saat menghapus fitur lebih lanjut. Gambar 4.5 menunjukkan bahwa akurasi terbaik diperoleh ketika menggunakan fitur Pclass, Age, Fare, Sex Encode dengan akurasi setinggi 83,7%.

## c. Exhaustive Search

| | feature_idx | cv_scores | avg_score | feature_names | ci_bound | std_dev | std_err |
|---|---|---|---|---|---|---|---|
| 0 | (0,) | [0.651685393258427, 0.648876404494382] | 0.650281 | (Pclass,) | 0.006043 | 0.001404 | 0.001404 |
| 1 | (1,) | [0.584269662921348, 0.598314606741573] | 0.591292 | (Age,) | 0.030215 | 0.007022 | 0.007022 |
| 2 | (2,) | [0.609550561797752, 0.615168539325842] | 0.61236 | (SibSp,) | 0.012086 | 0.002809 | 0.002809 |
| 3 | (3,) | [0.609550561797752, 0.592696629213483] | 0.601124 | (Parch,) | 0.036258 | 0.008427 | 0.008427 |
| 4 | (4,) | [0.651685393258427, 0.676966292134831] | 0.664326 | (Fare,) | 0.054387 | 0.01264 | 0.01264 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 506 | (0, 1, 2, 4, 5, 6, 7, 8) | [0.792134831460674, 0.806179775280899] | 0.799157 | (Pclass, Age, SibSp, Fare, Sex Encode, Embarke... | 0.030215 | 0.007022 | 0.007022 |
| 507 | (0, 1, 3, 4, 5, 6, 7, 8) | [0.775280898876404, 0.792134831460674] | 0.78370B | (Pclass, Age, Parch, Fare, Sex Encode, Embarke... | 0.036258 | 0.008427 | 0.008427 |
| 508 | (0, 2, 3, 4, 5, 6, 7, 8) | [0.755617977528089, 0.761235955056179] | 0.758427 | (Pclass, SibSp, Parch, Fare, Sex Encode, Embar... | 0.012086 | 0.002809 | 0.002809 |
| 509 | (1, 2, 3, 4, 5, 6, 7, 8) | [0.778089887640449, 0.789325842696629] | 0.78370B | (Age, SibSp, Parch, Fare, Sex Encode, Embarked... | 0.024172 | 0.005618 | 0.005618 |
| 510 | (0, 1, 2, 3, 4, 5, 6, 7, 8) | [0.783707865168539, 0.808988764044938] | 0.796348 | (Pclass, Age, SibSp, Parch, Fare, Sex Encode, ... | 0.054387 | 0.01264 | 0.01264 |

*Gambar 4. 6 Hasil Exhaustive Search*

Proses exhaustive search dilakukan dengan pembentukan kombinasi fitur. Semua kombinasi yang mungkin dari fitur-fitur yang tersedia dibuat. Model dibuat untuk setiap kombinasi fitur dan dievaluasi menggunakan metrik kinerja yang relevan dalam kasus ini metrik evaluasi yang digunakan adalah akurasi. Selanjutnya, kombinasi fitur yang memberikan

kinerja terbaik sesuai dengan kriteria yang dipilih. Akurasi terbaik menggunakan metode ini adalah 84,9% dengan fitur Pclass, Age, SibSp, Fare, Sex Encode.

**Metode Embedded**

Metode embedded yang diterapkan pada ekperimen ini adalah lasso, seleksi fitur dengan metode Lasso melibatkan beberapa langkah yang sederhana. Pertama, metode dimulai dengan menentukan model regresi logistik. Kemudian, menerapkan metode Lasso dengan memberikan penalti pada koefisien fitur menggunakan norma L1. Ini akan mendorong beberapa koefisien menjadi nol, menghilangkan fitur yang tidak signifikan. Selanjutnya, menyesuaikan parameter penalti, yang disebut alpha, untuk mengontrol seberapa kuat penalti yang diberikan pada koefisien. Nilai alpha yang lebih besar akan menghasilkan model dengan lebih sedikit fitur. Pada eksperimen yang telah dilakukan metode lasso menghasilkan akurasi setinggi 79%.

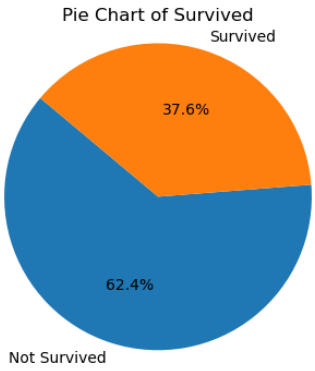
**Hasil Eksperimen Menggunakan Metode Seleksi Fitur**

*Tabel 4. 2 Hasil Eksperimen Menggunakan Metode Seleksi Fitur*

| Model | Model Yang Digunakan | Fitur Yang Digunakan | Akurasi Terbaik (%) |
|---|---|---|---|
| Base Line | Random Forest | Semua Fitur | 84,9 |
| Filter Chi Square & Correlation | Random Forest | Survived, Pclass, Age, SibSp, Parch, Fare, Sex Encode, Embarked_C, Embarked_S | 100 |
| Wrapper Forward Selection | Random Forest | Pclass, Age, SibSp, Sex Encode, Embarked_S | 82,1 |
| Wrapper Backward Selection | Random Forest | Pclass, Age, Fare, Sex Encode | 83,7 |
| Wrapper Exhaustive Search | Random Forest | Pclass, Age, SibSp, Fare, Sex Encode | 84,9 |
| Embedded Lasso | Random Forest | Pclass, Age, SibSp, Fare, Sex Encode | 79,8 |

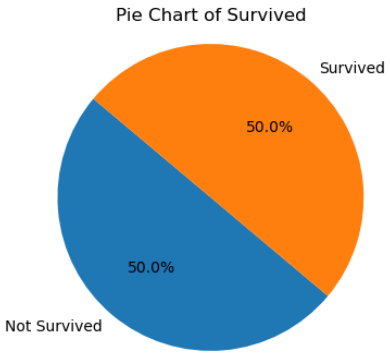
**Peningkatan Akurasi Menggunakan Metode Seleksi Fitur Dilanjutkan dengan Balancing Data**

Dataset displit menjadi *data train* dan *data test* menggunakan fungsi train_test_split. Ukuran *data train* sebesar 80% dari keseluruhan data sedangkan sisanya menjadi *data test*. Guna memastikan proses *balancing data* bersifat adil maka perlu dilakukan pemisahan sejak tahap ini. Setelah dilakukan proses split terlihat hasil distribusi *data train* seperti gambar di bawah ini




*Gambar 4. 7 Diagram Pie Penumpang Survived dan Not Survived*

Diagram *pie* menujukkan bahwa persentase penumpang *survived* sebesar 37,6% sedangkan *not survived* sebesar 62,4%. Asumsikan bahwa data ini adalah jenis data *imbalance*, oleh karena itu dilakukan proses *balancing data* dengan metode *random over sampling*. Hasil dari proses *balancing data* dapat dilihat pada gambar di bawah ini




*Gambar 4. 8 Diagram Pie Balanced*

Metode *oversampling* bertujuan untuk meningkatkan jumlah sampel dalam kelas minoritas sehingga memperkuat representasi kelas tersebut dalam model pembelajaran mesin. Adapun metode random oversampling melibatkan duplikasi acak sampel dari kelas minoritas sehingga meningkatkan jumlah sampel dalam kelas tersebut. Tabel di bawah ini menunjukkan kinerja model setelah dilakukan *balancing data*.

*Tabel 4. 3 Hasil Seleksi Fitur dan Balancing Data*

| Seleksi Fitur & Random Oversampling | Model Yang Digunakan | Fitur Yang Digunakan | Akurasi Terbaik (%) |
|---|---|---|---|
| Base Line | Random Forest | Semua Fitur | 84,3 |
| Filter Chi Square & Correlation | Random Forest | Survived, Pclass, Age, SibSp, Parch, Fare, Sex Encode, Embarked_C, Embarked_S | 83,2 |
| Wrapper Forward Selection | Random Forest | Pclass, Age, SibSp, Fare, Sex Encode | 83,7 |
| Wrapper Backward Selection | Random Forest | Pclass, Age, SibSp, Fare, Sex Encode | 83,7 |
| Wrapper Exhaustive Search | Random Forest | Pclass, Age, SibSp, Fare, Sex Encode | 83,7 |
| Embedded Lasso | Random Forest | Pclass, Age, SibSp, Fare, Sex Encode | 79,3 |

## 5. Kesimpulan

Berdasarkan eksperimen yang telah dilakukan didapatkan beberapa poin kesimpulan sebagai berikut:

- Proses *improvement* model dengan seleksi fitur secara keseluruhan dapat meningkatkan dan menurunkan kinerja model
- Kinerja terbaik dilakukan oleh *filter method* tanpa *balancing data* dengan akurasi 100%, namun model ini kurang bagus dalam melakukan generalisasi data dibuktikan dengan hasil akurasi data train mencapai 84,9%
- Proses *improvement* model dengan seleksi fitur ditambah dengan *balancing data* akurasinya cenderung lebih stabil
- Proses yang memakan waktu paling lama dan penggunaan sumber daya CPU terbanyak terjadi pada *exhaustive search* karena metode ini mencoba semua kemungkinan kombinasi fitur.

- Rata-rata akurasi dari seleksi fitur tanpa balancing data adalah 86,1% sedangkan dengan balancing data 82,7%
- Perlunya perlakuan yang berbeda dari tahap data *preprocessing* hingga *modelling* bergantung pada kondisi dataset untuk mendapatkan model dengan kinerja terbaik.

## 6. Lampiran

*Source code* eksperimen dapat dilihat pada link berikut ini :

1. Seleksi Fitur
   https://colab.research.google.com/drive/1TMNsWSObLAx4qyT3HpVbG2gAvCAiA5d4?usp=sharing
2. Seleksi Fitur & Balancing Data
   https://colab.research.google.com/drive/1uY9TPBW5w5HExRwY1DR0rFnIhZqWTlnf?usp=sharing