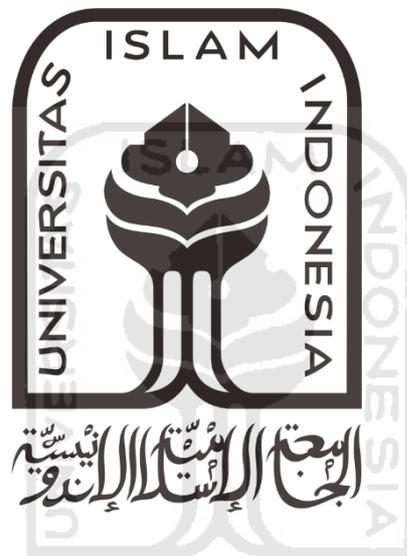


**PERAMALAN ENERGI *PHOTOVOLTAIC* DENGAN
MENGUNAKAN ALGORITMA *NAÏVE BAYES CLASSIFIER*
DAN *K-NEAREST NEIGHBORS***

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

Muhammad Ikhsan

15524028

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta
2020**

LEMBAR PENGESAHAN

PERAMALAN ENERGI *PHOTOVOLTAIC* DENGAN MENGGUNAKAN ALGORITMA

NAÏVE BAYES CLASSIFIER DAN *K-NEAREST NEIGHBORS*

TUGAS AKHIR
ISLAM
Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Muhammad Ikhsan
15524028

Yogyakarta, 20 Juli 2020

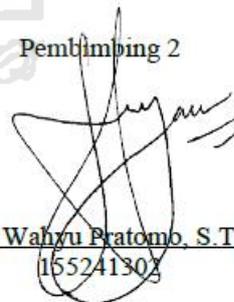
Menyetujui,

Pembimbing 1



Dzata Farahiyah, S.T., M.Sc.
155220509

Pembimbing 2



Setyawan Wahyu Pratomo, S.T., M.T.
155241303

LEMBAR PENGESAHAN

SKRIPSI

PERAMALAN ENERGI *PHOTOVOLTAIC* DENGAN MENGGUNAKAN ALGORITMA

NAÏVE BAYES CLASSIFIER DAN *K-NEAREST NEIGHBORS*

Dipersiapkan dan disusun oleh:

Muhammad Ikhsan

15524028

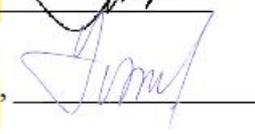
Telah dipertahankan di depan dewan penguji

Pada tanggal: 12 Agustus 2020

Susunan dewan penguji

Ketua Penguji : Dzata Farahiyah, S.T., M.Sc., 

Anggota Penguji 1: Dr. Eng. Hendra Setiawan, S.T., M.T., 

Anggota Penguji 2: Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D., 

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 12 Agustus 2020

Ketua Program Studi Teknik Elektro



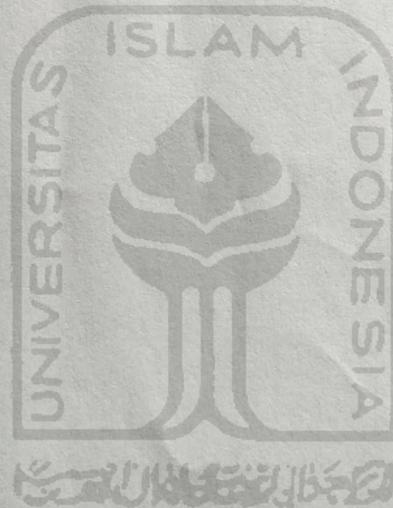
Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D.

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.



Yogyakarta, 20 Juli 2020



Muhammad Ikhsan

KATA PENGANTAR

Assalamu 'alaikum Wr. Wb

Segala puji dan syukur penulis panjatkan Kehadirat Allah SWT yang telah melimpahkan rahmat, karunia serta hidayah-Nya kepada hamba-Nya selama masih dalam iman dan ikhsan.

Atas petunjuk dan ridho-Nya, Skripsi yang berjudul "*PERAMALAN ENERGI PHOTOVOLTAIC DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES CLASSIFIER DAN K-NEAREST NEIGHBORS* " ini dapat diselesaikan dengan baik dan lancar. Skripsi ini wajib ditempuh oleh mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia sebagai salah satu syarat untuk menyelesaikan jenjang studi Strata 1.

Kelancaran dalam mempersiapkan dan menyelesaikan Skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dengan rasa hormat dan terima kasih yang sebesar-besarnya penulis haturkan kepada:

1. Bapak Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D., selaku Ketua Jurusan Teknik Elektro, Universitas Islam Indonesia.
2. Ibu Dzata Farahiyah, S.T., M.Sc., selaku Dosen Pembimbing I, yang telah memberi bantuan dan pengarahan hingga terselesaikan laporan Skripsi ini.
3. Bapak Setyawan Wahyu Pratomo, S.T., M.T., selaku Dosen Pembimbing II, yang telah memberi bantuan dan pengarahan hingga terselesaikan laporan Skripsi ini
4. Seluruh Dosen Jurusan Teknik Elektro, terima kasih atas bimbingan selama menempuh kuliah dari semester pertama hingga akhir di Jurusan Teknik Elektro.
5. Bapak Samsul dan Ibu Hanny serta keluargaku, terimakasih atas bimbingan, usaha, do'a dan kasih sayangnya.
6. Teman-teman mahasiswa Jurusan Teknik Elektro Universitas Islam Indonesia angkatan 2015.
7. Teman-teman yang membantu saya dalam mengerjakan skripsi (Dimas, Afif, Ihsan, Bhayu, Ragil, Zikra, Gading, dan Raka) yang telah mendukung dan mendo'akan saya untuk menyelesaikan skripsi ini terima kasih.
8. Semua pihak yang telah memberikan masukan, dorongan dan semangat dalam menyelesaikan skripsi ini.

Wassalamu 'alaikum Wr. Wb.

ARTI LAMBANG DAN SINGKATAN

AI	: <i>Artificial Intelligence</i>
BMKG	: Badan Meteorologi, Klimatologi, dan Geofisika
FN	: <i>False Negative</i>
FP	: <i>False Positive</i>
KNN	: <i>K-Nearest Neighbors</i>
LMA	: <i>Levenberg-Marquardt Algorithm</i>
ML	: <i>Machine Learning</i>
MW	: <i>megawatt</i>
N	: <i>Negative</i>
NASA	: <i>National Aeronautics and Space Administration</i>
NBC	: <i>Naïve Bayes Classifier</i>
NMAE	: <i>Normalized Mean Absolute Error</i>
P	: <i>Positive</i>
PEH	: <i>Pembangkitan Energi Harian</i>
PV	: <i>Photovoltaic</i>
RRMH	: <i>Rata-rata Radiasi Matahari Harian</i>
RNN	: <i>Recurrent Neural Network</i>
RSPSH	: <i>Rata-rata Suhu Panel Surya Harian</i>
TDPSMH	: <i>Total Durasi Pancaran Sinar Matahari Harian</i>
TN	: <i>True Negative</i>
TP	: <i>True Positive</i>
h	: <i>hour</i>
kWh	: <i>kilowatt hour</i>
kWh/m ² /day	: <i>kilowatt hour per square meter per day</i>
°C	: <i>Celcius</i>

ABSTRAK

Energi surya yang melimpah diseluruh dunia dapat menjadi sumber energi yang ekonomis dalam penerapannya. Disebabkan hal itu, pemanfaatan energi surya terus meningkat sepanjang tahun. Banyak faktor yang mempengaruhi pembangkitan energi listrik melalui panel surya. Faktor-faktor tersebut meliputi kondisi atmosfer, lintasan matahari, kondisi cuaca, tutupan awan, karakteristik dari panel surya yang digunakan, serta faktor eksternal lainnya. Karena banyak faktor yang mempengaruhi produksi energi yang dihasilkan, maka penting untuk melakukan peramalan terhadap produksi energi yang dihasilkan oleh *Photovoltaic*. Hal ini dapat membantu dalam mengelola perencanaan dan manajemen sistem operasi serta meningkatkan efisiensi dari sistem energi surya. Pada penelitian ini, metode *Machine Learning* digunakan untuk memprediksi suatu label klasifikasi dari produksi energi yang dihasilkan oleh *Photovoltaic*. Fitur atau atribut yang digunakan untuk memprediksi produksi energi adalah Rata-rata Radiasi Matahari Harian, Rata-rata Suhu Panel Surya Harian, dan Total Durasi Pancaran Sinar Matahari Harian. Algoritma *Naïve Bayes Classifier* dan *K-Nearest Neighbors* digunakan untuk melatih model *machine learning* dalam beberapa skenario pengujian. Berdasarkan hasil penelitian, didapati bahwa model yang dilatih dengan menggunakan *K-Nearest Neighbors* memiliki performa yang lebih baik secara keseluruhan daripada model *Naïve Bayes Classifier*.

Kata Kunci: Panel Surya, *Machine Learning*, *Naïve Bayes Classifier*, *K-Nearest Neighbors*.



DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ARTI LAMBANG DAN SINGKATAN	v
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 Studi Literatur	4
2.2 Tinjauan Teori.....	5
2.2.1 <i>Photovoltaic</i>	5
2.2.2 <i>Machine Learning</i>	6
2.2.3 <i>Naïve Bayes Classifier (NBC)</i>	6
2.2.4 <i>K-Nearest Neighbors (KNN)</i>	7
2.2.5 <i>Confusion Matrix</i>	9
2.2.6 Akurasi dan Error	9
2.2.7 Presisi	10

2.2.8 <i>Recall</i>	11
2.2.9 <i>F-Score</i>	11
BAB 3 METODOLOGI	12
3.1 Alur Penelitian	12
3.1.1 Studi Literatur	12
3.1.2 Pengumpulan dan Pengolahan Data Penelitian	12
3.1.3 Pembuatan Program	15
3.1.4 Pembagian Data Latih dan Data Uji	15
3.1.5 Analisis Hasil Kinerja	16
3.1.6 Pembuatan Laporan	16
3.2 Perancangan Sistem	16
3.2.1 Proses Penyiapan Data	17
3.2.2 Proses Pelatihan Model	17
3.2.3 Proses Pengujian & Evaluasi	18
BAB 4 HASIL DAN PEMBAHASAN	19
4.1 Simulasi <i>Naïve Bayes</i>	19
4.1.1 Akurasi dan Eror Pada Model <i>Naïve Bayes</i>	19
4.1.2 Presisi Pada Model <i>Naïve Bayes</i>	20
4.1.3 <i>Recall</i> Pada Model <i>Naïve Bayes</i>	20
4.1.4 <i>F-Score</i> Pada Model <i>Naïve Bayes</i>	21
4.2 Simulasi <i>K-Nearest Neighbors</i>	22
4.2.1 Akurasi dan Eror Model <i>K-Nearest Neighbors</i>	22
4.2.2 Presisi Model <i>K-Nearest Neighbors</i>	23
4.2.3 <i>Recall</i> Model <i>K-Nearest Neighbors</i>	23
4.2.4 <i>F-Score</i> Model <i>K-Nearest Neighbors</i>	24
4.3 Perbandingan Performa Antar Algoritma	25
4.4 Pengujian <i>K-Fold Cross Validation</i>	27
BAB 5 KESIMPULAN DAN SARAN	28

5.1 Kesimpulan	28
5.2 Saran	28
DAFTAR PUSTAKA	29
LAMPIRAN	1



DAFTAR GAMBAR

Gambar 3.1 Bagun Alur Penelitian	12
Gambar 3.2 Struktur <i>Dataset</i>	13
Gambar 3.3 Blok Diagram Sistem	17
Gambar 4.1 Hasil Grafik Nilai Akurasi dan Error Pada Model <i>Naïve Bayes</i>	19
Gambar 4.2 Hasil Grafik Nilai Presisi Pada Model <i>Naïve Bayes</i>	20
Gambar 4.3 Hasil Grafik Nilai <i>Recall</i> Pada Model <i>Naïve Bayes</i>	21
Gambar 4.4 Hasil Grafik Nilai <i>F-Score</i> Pada Model <i>Naïve Bayes</i>	21
Gambar 4.5 Hasil Grafik Nilai Akurasi dan Error Pada Model KNN.....	22
Gambar 4.6 Hasil Grafik Nilai Presisi Pada Model KNN.....	23
Gambar 4.7 Hasil Grafik Nilai <i>Recall</i> Pada Model KNN	24
Gambar 4.8 Hasil Grafik Nilai <i>F-Score</i> Pada Model KNN	24
Gambar 4.9 Hasil Grafik Nilai Tiap Parameter Pengujian Model	25



DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i>	9
Tabel 2.2 <i>Confusion Matrix</i> 3×3	9
Tabel 3.1 Data <i>Input</i> dan <i>Output</i>	13
Tabel 3.2 Label Klasifikasi	15
Tabel 3.3 <i>Label Encoding</i>	15
Tabel 3.4 Skenario Pengujian.....	16
Tabel 3.5 Data Latih dan Data Uji	16
Tabel 3.6 Pengujian Singkat Nilai K.....	18
Tabel 4.1 Akurasi <i>K-Cross Fold Validation</i>	27



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Energi surya merupakan salah satu sumber energi paling terjangkau dan sumber energi terbarukan yang bersih di dunia. Energi surya yang melimpah diseluruh dunia dapat menjadi sumber energi yang ekonomis dalam penerapannya. Disebabkan kelebihan yang dimilikinya, pemanfaatan energi surya terus meningkat sepanjang tahun. Untuk mengkonversi energi surya menjadi energi listrik salah satunya adalah dengan menggunakan perangkat *Photovoltaic* (PV) atau *PV Module*.

Dalam penerapannya, variabel meteorologis yang berubah seiring ruang dan waktu membuat produksi listrik dengan memanfaatkan energi surya menjadi tidak pasti. Banyak faktor yang mempengaruhi pembangkitan energi listrik melalui panel surya. Faktor-faktor tersebut meliputi kondisi atmosfer, lintasan matahari, kondisi cuaca, tutupan awan, karakteristik dari panel surya yang digunakan, dan faktor eksternal lainnya. Karena banyak faktor yang mempengaruhi produksi energi yang dihasilkan, maka penting untuk melakukan peramalan terhadap produksi energi PV. Peramalan produksi energi PV dapat membantu dalam mengelola perencanaan dan membantu mengetahui produksi energi PV berdasarkan data meteorologis pada suatu daerah yang belum terpasang perangkat PV/sistem *monitoring* PV. Akhir-akhir ini, banyak penelitian yang melakukan peramalan terhadap produksi energi dari sistem surya. Diantaranya menggunakan metode *Machine Learning* (ML) [1].

Machine learning merupakan teknik yang memungkinkan mesin belajar langsung dari contoh, data, dan pengalaman. Berbeda dengan pendekatan pemrograman tradisional yang mana mesin diberikan perintah satu persatu, *machine learning* dapat membuat keputusan sendiri terhadap suatu permasalahan setelah ia “belajar” dari contoh atau data yang diberikan. *Machine learning* termasuk dari bagian *Artificial Intelligence* (AI) yang mendukung sistem cerdas yang dapat mempelajari fungsi tertentu setelah diberikan data untuk dipelajari [2].

Oleh karena itu, pada penelitian ini penulis berupaya untuk melakukan peramalan dari produksi energi yang dihasilkan perangkat *photovoltaic* dengan menggunakan algoritma *Naive Bayes Classifier* (NBC) dan *K-Nearest Neighbors* (KNN) yang mana merupakan algoritma klasifikasi dari *machine learning*. Peramalan produksi energi PV dilakukan untuk membantu dalam mengelola perencanaan dan mengetahui produksi energi PV berdasarkan data meteorologis pada suatu daerah yang belum terpasang perangkat PV/sistem *monitoring* PV. Modul PV yang dijadikan studi terletak di Gedung K.H. Mas Mansur milik Fakultas Teknologi Industri Universitas

Islam Indonesia. Setelah melakukan peramalan, dilakukan analisa terkait performa dari algoritma yang digunakan dalam beberapa skenario pengujian.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan sebelumnya maka didapati beberapa rumusan masalah, yaitu:

1. Bagaimana kinerja dari model peramalan energi *photovoltaic* menggunakan algoritma *Naïve Bayes Classifier*?
2. Bagaimana kinerja dari model peramalan energi *photovoltaic* menggunakan algoritma *K-Nearest Neighbors*?

1.3 Batasan Masalah

Batasan Masalah berisi hal-hal yang membatasi lingkup penelitian.

1. Perangkat *photovoltaic* yang dijadikan studi terletak di Gedung K.H. Mas Mansur milik Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Fitur yang digunakan untuk meramalkan produksi energi *photovoltaic* adalah Rata-rata Suhu Panel Surya Harian (RSPSH), Total Durasi Pancaran Sinar Matahari Harian (TDPSMH), dan Rata-rata Radiasi Matahari Harian (RRMH).
3. Data yang digunakan untuk RSPSH diperoleh dari *database online* milik SolaX Power (<https://solax-portal.com>, Akses pada Maret 2020).
4. Data yang digunakan untuk TDPSMH diperoleh dari *database online* milik Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) (<http://dataonline.bmkg.go.id>, Akses pada Maret 2020).
5. Data yang digunakan untuk RRMH diperoleh dari *database online* milik *National Aeronautics and Space Administration* (NASA) (<https://power.larc.nasa.gov>, Akses pada Maret 2020).
6. Penetapan *range* untuk label berdasarkan pada *dataset*.
7. *Dataset* yang digunakan mempunyai jumlah label kelas yang sama banyak.
8. Rasio skenario pembagian data latih dan data uji yang digunakan, yaitu 75:25, 80:20, dan 90:10.
9. Pembagian data uji dan data latih dilakukan secara acak dengan menggunakan bantuan *scikit-learn library*.

10. Perhitungan komputasi algoritma *Naïve Bayes Classifier* dan *K-Nearest Neighbors* menggunakan bantuan *scikit-learn library*.
11. Parameter K yang ditetapkan bernilai 5 (K=5).
12. Parameter untuk mengevaluasi kinerja model adalah akurasi, presisi, *recall*, dan *f-score*.

1.4 Tujuan Penelitian

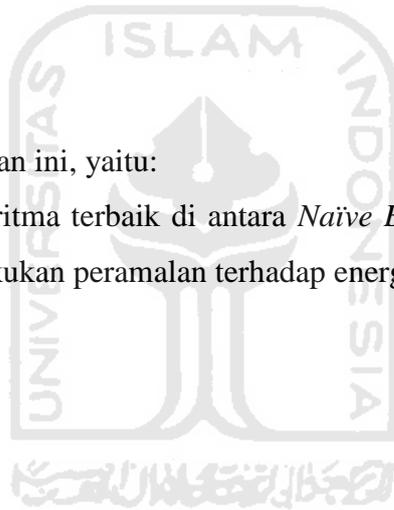
Tujuan dari penelitian ini dibagi menjadi beberapa hal, yaitu:

1. Untuk mengetahui kinerja dari model peramalan energi *photovoltaic* dengan menggunakan algoritma *Naïve Bayes Classifier*.
2. Untuk mengetahui kinerja dari model peramalan energi *photovoltaic* dengan menggunakan algoritma *K-Nearest Neighbors*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini, yaitu:

1. Dapat mengetahui algoritma terbaik di antara *Naïve Bayes Classifier* dan *K-Nearest Neighbors*, dalam melakukan peramalan terhadap energi *photovoltaic*.



BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian yang dilakukan oleh Bayindir, Yesilbudak, Colak, dan Genc [3], melakukan peramalan terhadap produksi energi listrik yang dihasilkan oleh sistem PV di Van, Turki pada tahun 2013-2014. Hal ini dilakukan untuk membantu memanfaatkan potensi maksimal yang dapat dihasilkan dan meningkatkan efisiensinya. Pada penelitian ini, metode pengklasifikasian *Naïve Bayes Classifier* (NBC) digunakan untuk membantu meramalkan total energi listrik harian yang dihasilkan oleh PV. Pada *dataset* terdapat 4 fitur atau atribut yang digunakan untuk melatih model, yaitu rata-rata temperatur harian, durasi total pancaran sinar matahari harian, radiasi matahari harian, dan total pembangkitan energi PV harian. Setiap fitur dibagi menjadi lima label kelas (“Sangat Rendah”, “Rendah”, “Sedang”, “Tinggi”, dan “Sangat Tinggi”) untuk mempermudah proses klasifikasi. Dari hasil penelitian, diperoleh *recall* peramalan sebesar 98,305 % dan akurasi sebesar 82,191%.

Penelitian yang dilakukan oleh Nam dan Hur [4], mengusulkan pendekatan peramalan probabilistik untuk produksi PV di Korea Selatan berdasarkan data yang dikumpulkan dari Januari 2015-Desember 2016 dengan menggunakan model peramalan *hybrid spatio-temporal*. Penelitian ini menggunakan 2 metode estimasi untuk meramal *output* dari sistem *photovoltaic*, yaitu *Naïve Bayes Classifier* dan metode *Kriging* untuk membantu melakukan pemodelan spasial pada titik yang diinginkan. Data yang digunakan berasal dari tahun 2015 untuk meramalkan produksi PV selama setahun berikutnya (2016). Untuk menguji metode *hybrid* ini digunakan *Normalized Mean Absolute Error* (NMAE) untuk melihat performa model. Hasil penelitian menunjukkan bahwa prediksi dari model memiliki nilai NMAE kurang dari 10%.

Pada penelitian selanjutnya, menurut Awan, Khan, dan Aslam [1], menyatakan bahwa banyak faktor yang mempengaruhi pembangkitan energi listrik melalui *photovoltaic*. Faktor-faktor tersebut meliputi kondisi atmosfer, lintasan matahari, kondisi cuaca, tutupan awan, dan sifat fisik dari *solar cell* yang mengubah energi matahari menjadi energi listrik. Pemahaman yang akurat dan benar tentang faktor-faktor ini menjamin model peramalan pembangkitan energi listrik dengan sel surya yang andal. Penelitian ini mengusulkan solusi untuk memasukkan parameter yang berpengaruh pada pemodelan *Recurrent Neural Network* (RNN). RNN selanjutnya dioptimalkan menggunakan *Levenberg-Marquardt Algorithm* (LMA) untuk mendapatkan akurasi perkiraan yang lebih baik. Sistem PV yang dijadikan studi terletak di Lahore, Pakistan dengan

menggunakan data pada tahun 2014. Hasil dari penelitian menunjukkan bahwa peramalan memiliki akurasi yang memuaskan sebesar 97,89%

Pada penelitian yang dilakukan oleh Putri, Suparti, dan Rahmawati [5], melakukan klasifikasi terhadap status kerja penduduk angkatan kerja di Demak pada tahun 2012 dengan menggunakan algoritma *Naïve Bayes Classifier* dan *K-Nearest Neighbors*. Fitur atau atribut yang digunakan untuk menentukan status seseorang menganggur atau tidak adalah jenis kelamin, status dalam rumah tangga, status perkawinan, pendidikan, dan usia. Dari hasil penelitian, didapati akurasi yang diperoleh oleh *Naïve Bayes Classifier* sebesar 94,09% dan akurasi yang dihasilkan oleh *K-Nearest Neighbors* sebesar 96,06% .

Pada penelitian yang dilakukan oleh Hussein, Hajimirza, dan Balog [6], menyatakan bahwa prediksi *output* daya PV penting untuk mengamankan operasi *grid*, penjadwalan, dan manajemen energi *grid*. Salah satu elemen kunci dalam prediksi keluaran PV adalah analisis deret waktu terutama di lokasi di mana pengukuran radiasi matahari atau parameter cuaca lainnya belum direkam. Dalam penelitian ini, beberapa metode prediksi deret waktu termasuk metode statistik dan *neural network* yang didasarkan pada kecerdasan buatan dibandingkan untuk memprediksi output daya PV. Peramalan tenaga surya dilakukan setiap jam untuk memverifikasi efektivitas model. Data yang digunakan dalam penelitian ini terdiri dari 3640 jam data operasi yang diambil dari stasiun PV yang terhubung ke jaringan 20 MW di Cina. Studi perbandingan menunjukkan bahwa model *neural network* lebih akurat daripada model statistik ketika digunakan untuk prediksi deret waktu dari *output* daya PV dan memerlukan lebih sedikit waktu komputasi.

2.2 Tinjauan Teori

2.2.1 Photovoltaic

Photovoltaic (PV) adalah bidang teknologi dan penelitian terkait dengan perangkat yang secara langsung mengkonversi sinar matahari menjadi listrik. Sel surya adalah unsur dasar dalam pembuatan sistem *photovoltaic*. Sel surya terbuat dari bahan semikonduktor, seperti silikon. Sejumlah sel surya saling terhubung secara elektrik dan dipasang dalam struktur pendukung tunggal disebut dengan ‘modul PV’ [7].

Menurut [8], ada beberapa faktor yang mempengaruhi produksi dari panel PV, yaitu resistensi beban, intensitas cahaya matahari, temperatur dari sel, tutupan cahaya matahari, dan jenis bahan sel yang digunakan. Pada penelitian ini, beberapa parameter penentu tersebut digunakan untuk meramalkan produksi dari PV dengan menggunakan metode *Machine Learning*.

Adapun Sistem PV yang dijadikan studi, terletak di Gedung K.H. Mas Mansur milik Fakultas Teknologi Industri Universitas Islam Indonesia dengan kapasitas pembangkitan 2×5000 watt solar power generation.

2.2.2 Machine Learning

Machine Learning (ML) atau pembelajaran mesin merupakan teknik yang populer akhir-akhir ini karena banyak digunakan untuk menirukan atau menggantikan perilaku manusia dalam menyelesaikan masalah. Termasuk ke dalam cabang aplikasi dari kecerdasan buatan atau *Artificial Intelligence* (AI), *machine learning* berfokus pada pengembangan sistem yang mampu belajar sendiri tanpa harus berulang kali diprogram oleh manusia. Teknik ini memungkinkan mesin untuk belajar langsung dari contoh dan pengalaman dalam bentuk data [2].

Sekumpulan data yang digunakan untuk mengajar atau melatih model *machine learning* disebut dengan Data latih atau *train set* kemudian untuk menguji suatu model *machine learning* yang telah dilatih, digunakan sebuah *dataset* baru yang disebut dengan Data uji atau *test set*. *Dataset* yang digunakan untuk menguji model tidak boleh diambil dari data latih dikarenakan model dapat mengingat data yang digunakan untuk melatihnya sehingga model yang dihasilkan memiliki performa yang buruk.

2.2.3 Naïve Bayes Classifier (NBC)

Naïve Bayes Classifier atau yang selanjutnya disebut dengan *Naïve Bayes* merupakan metode pengklasifikasian statistik yang didasari oleh teorema *Bayes*. Teorema *Bayes* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya dengan asumsi bahwa setiap fitur atau atribut penentu keputusan bersifat bebas (*independence*) sehingga keberadaan setiap fitur tidak ada kaitannya dengan keberadaan fitur yang lain. Kondisi antar fitur diasumsikan saling bebas, oleh karena itu disebut dengan *Naïve* [9]. Teorema *Bayes* ditunjukkan oleh persamaan (2.1).

$$P(A|B) = \frac{(P(B|A) \times P(A))}{P(B)} \quad (2.1)$$

Dimana:

$P(A|B)$: Probabilitas label A berdasarkan fitur B (*posterior probability*)

$P(B|A)$: Probabilitas fitur B berdasarkan pada label A (*likelihood*)

$P(A)$: Probabilitas label A (*prior probability*)

$P(B)$: Probabilitas fitur B (*evidence*)

Penentuan hasil prediksi label untuk suatu sampel data dilakukan dengan membandingkan *posterior probability* masing-masing label, label yang memiliki *posterior probability* paling tinggi akan dijadikan hasil prediksi. Pada penelitian ini ada 3 label yang digunakan, yaitu “Rendah”, “Sedang”, dan “Tinggi”. Adapun untuk fitur yang digunakan adalah RSPSH, TDPSMH, dan RRMH.

Penerapan persamaan (2.1) dengan menggunakan fitur dan label yang digunakan pada penelitian ini, menghasilkan persamaan (2.2), (2.3), dan (2.4) untuk menghitung masing-masing *posterior probability* label.

$$P(\text{Rendah}|X_1, X_2, X_3) = \frac{(P(X_1, X_2, X_3|\text{Rendah}) \times P(\text{Rendah}))}{P(X_1, X_2, X_3)} \quad (2.2)$$

$$P(\text{Sedang}|X_1, X_2, X_3) = \frac{(P(X_1, X_2, X_3|\text{Sedang}) \times P(\text{Sedang}))}{P(X_1, X_2, X_3)} \quad (2.3)$$

$$P(\text{Tinggi}|X_1, X_2, X_3) = \frac{(P(X_1, X_2, X_3|\text{Tinggi}) \times P(\text{Tinggi}))}{P(X_1, X_2, X_3)} \quad (2.4)$$

Dimana:

X_1 : RSPSH

X_2 : TDPSMH

X_3 : RRMH

Perhitungan *posterior probability* pada penelitian ini, dilakukan dengan menggunakan bantuan *scikit-learn library*. Pada *scikit-learn library* terdapat algoritma komputasi yang mempermudah melakukan perhitungan pada jumlah data yang banyak.

2.2.4 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) merupakan metode klasifikasi yang mengelompokkan data ke kelompok yang memiliki sifat termirip dengannya. Algoritma klasifikasi ini disebut juga algoritma malas karena tidak mempelajari cara mengkategorikan data, melainkan hanya mengingat data yang sudah ada. Tujuan dari algoritma ini adalah mengklasifikasi objek baru berdasarkan atribut dan sampel latih. Pengklasifikasian tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori [10].

Pada KNN, parameter K merupakan jumlah dari tetangga terdekat. Algoritma ini menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. Pada saat

diberikan sampel uji, maka algoritma akan menemukan sejumlah K objek (titik latih) yang paling dekat dengan titik uji. Dekat atau jauhnya tetangga pada penelitian ini, dihitung berdasarkan jarak *Manhattan*. Klasifikasi dilakukan menggunakan menentukan mayoritas label di antara K objek yang telah ditentukan, label mayoritas pada sejumlah tetangga K akan dijadikan label hasil prediksi [11]. Perhitungan jarak pada penelitian ini dirumuskan oleh persamaan (2.5).

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2.5)$$

Dimana:

$d(x, y)$: Jarak

x_i : Sampel data latih

y_i : Sampel data uji

i : Variabel data

n : Dimensi data

Berdasarkan fitur yang digunakan pada penelitian ini, maka perhitungan jarak antara suatu sampel data latih dan data uji diperlihatkan oleh persamaan (2.6).

$$d(x_i, y_i) = |A_{x_i} - A_{y_i}| + |B_{x_i} - B_{y_i}| + |C_{x_i} - C_{y_i}| \quad (2.6)$$

Dimana:

$d(x_i, y_i)$: Jarak antara suatu data uji dan data latih

x_i : Data latih

y_i : Data uji

i : Variabel data

A: RSPSH

B: TDPSMH

C: RRMH

Perhitungan jarak pada penelitian ini, dilakukan dengan menggunakan bantuan *scikit-learn library* yang mana di dalamnya terdapat algoritma komputasi yang mempermudah melakukan perhitungan pada jumlah data yang banyak.

2.2.5 Confusion Matrix

Confusion Matrix adalah tabel yang sering digunakan untuk menggambarkan kinerja model klasifikasi (*classifier*) pada satu set data uji yang nilai sebenarnya diketahui. Pada *machine learning* dan khususnya klasifikasi statistik, *confusion matrix* juga dikenal sebagai *error matrix*. *Confusion matrix* seperti yang ditampilkan pada Tabel 2.1 dapat membantu dalam visualisasi kinerja suatu algoritma. Tabel ini menampilkan hasil prediksi pada masalah klasifikasi, jumlah prediksi benar dan salah dirangkum dengan nilai-nilai dan dipecah kepada masing-masing label. Hal ini dapat memberikan wawasan dalam mengetahui jenis kesalahan yang dibuat oleh *classifier*.

Tabel 2.1 *Confusion Matrix*

		<i>Predict Label</i>	
		<i>Positive (P)</i>	<i>Negative (N)</i>
<i>Actual Label</i>	P	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
	N	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Dimana:

True Positive: Merupakan data positif dan diprediksi benar

True Negative: Merupakan data negatif dan diprediksi benar

False Positive: Merupakan data negatif namun diprediksi sebagai data positif (error tipe 1)

False Negative: Merupakan data positif namun diprediksi sebagai data negatif (error tipe 2)

Penelitian ini menggunakan 3 label klasifikasi sehingga Tabel 2.1 berubah menjadi sedikit lebih rumit seperti yang ditunjukkan oleh Tabel 2.2.

Tabel 2.2 *Confusion Matrix 3 × 3*

		<i>Predict Label</i>		
		Label A	Label B	Label C
<i>Actual Label</i>	Label A	AA	AB	AC
	Label B	BA	BB	BC
	Label C	CA	CB	CC

2.2.6 Akurasi dan Error

Akurasi merupakan jumlah prediksi yang benar dibagi dengan keseluruhan sampel data. Akurasi dapat diperoleh menggunakan persamaan (2.7).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.7)$$

Dikarenakan penelitian ini menggunakan 3 label sehingga *confusion matrix* menjadi 3×3 , maka penerapan persamaan (2.7) berdasarkan pada tabel Tabel 2.2, menghasilkan persamaan (2.8).

$$\text{Akurasi} = \frac{AA + BB + CC}{AA + AB + AB + BA + BB + BC + CA + CB + CC} \quad (2.8)$$

Error merupakan seberapa banyak prediksi yang salah, dinyatakan dengan persamaan (2.9).

$$\text{Error} = \frac{FP + FN}{TP + TN + FP + FN} \quad (2.9)$$

Kemudian, persamaan (2.9) berubah mengikuti Tabel 2.2, sehingga menjadi persamaan (2.10).

$$\text{Error} = \frac{AB + AC + BA + BC + CA + CB}{AA + AB + AB + BA + BB + BC + CA + CB + CC} \quad (2.10)$$

Atau, persamaan (2.10) dapat ditulis menjadi persamaan (2.11).

$$\text{Error} = 1 - \text{Akurasi} \quad (2.11)$$

2.2.7 Presisi

Presisi merupakan rasio berapa banyak prediksi yang benar-benar positif dari semua kelas positif yang diprediksi dengan benar. Nilai presisi dapat dihitung menggunakan persamaan (2.12).

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.12)$$

Dikarenakan setiap label memiliki prediksi positif, maka untuk menghitung presisi dari *classifier* secara keseluruhan harus menghitung terlebih dahulu presisi per labelnya mengikuti Tabel 2.2 menggunakan persamaan (2.13).

$$\text{Presisi } i = \frac{A_i}{A_i + B_i + C_i} \quad (2.13)$$

Setelah itu, menghitung presisi total menggunakan persamaan (2.14).

$$\text{Presisi} = \frac{\text{Presisi A} + \text{Presisi B} + \text{Presisi C}}{3} \quad (2.14)$$

2.2.8 Recall

Recall atau *sensitivity* merupakan seberapa banyak prediksi yang benar dari semua kelas positif. Nilai *recall* dapat dihitung dengan menggunakan persamaan (2.15).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.15)$$

Dikarenakan setiap label memiliki prediksi positif, maka untuk menghitung *recall* dari *classifier* secara keseluruhan harus menghitung terlebih dahulu *recall* per labelnya mengikuti tabel menggunakan persamaan (2.16).

$$\text{Recall } i = \frac{iA}{iA + iB + iC} \quad (2.16)$$

Maka, *recall* dari *classifier* diperoleh menggunakan persamaan (2.17).

$$\text{Recall} = \frac{\text{Recall A} + \text{Recall B} + \text{Recall C}}{3} \quad (2.17)$$

2.2.9 F-Score

Presisi dan *recall* merupakan pengukuran kinerja model yang sama pentingnya. Namun, keduanya sedikit bertolak belakang. *Recall* bisa ditingkatkan semaksimal mungkin dengan cara memperbanyak prediksi sampel pada kelas positif yang mengakibatkan FP juga bertambah. Hal ini akan membuat presisi semakin turun karena tujuan utama dari evaluasi model menggunakan presisi adalah mengurangi jumlah FP. Berlaku juga sebaliknya untuk *recall* yang dipengaruhi oleh FN. Oleh karena itu, dibutuhkan suatu cara untuk mendapatkan model yang seimbang. *F-Score* digunakan untuk mencari nilai tengah dari presisi dan *recall* yang mana merupakan rata-rata harmonik dari keduanya [12]. *F-Score* dapat dihitung dengan menggunakan persamaan (2.18).

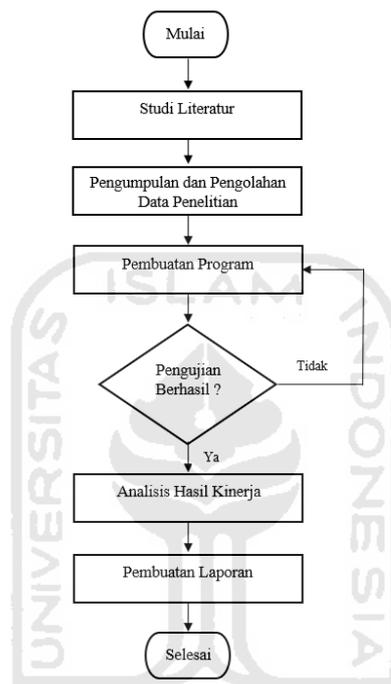
$$F - \text{Score} = 2 \times \frac{\text{presisi} \times \text{recall}}{\text{presisi} + \text{recall}} \quad (2.18)$$

BAB 3

METODOLOGI

3.1 Alur Penelitian

Secara garis besar langkah-langkah atau tahapan yang dilakukan oleh penulis agar penelitian menjadi lebih terstruktur diilustrasikan oleh bagan alur seperti pada Gambar 3.1 beserta penjelasan mengenai setiap tahapannya.



Gambar 3.1 Bagan Alur Penelitian

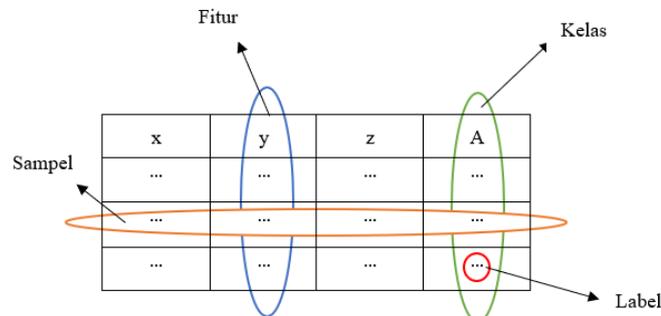
3.1.1 Studi Literatur

Pada tahapan ini penulis mengumpulkan dan mempelajari berbagai sumber literatur akademis seperti *paper*, buku, dan jurnal terkait topik yang sedang diteliti. Berkaitan dengan pembahasan mengenai *Photovoltaic*, *Machine Learning*, *Naïve Bayes Classifier*, dan *K-Nearest Neighbors* serta metode yang digunakan pada peneliti sebelumnya dalam melakukan penelitian yang sejenis.

3.1.2 Pengumpulan dan Pengolahan Data Penelitian

Setelah menentukan permasalahan dan membaca studi literatur, maka langkah selanjutnya adalah menyiapkan *dataset*. Metode *machine learning* melakukan pembelajaran dengan menggunakan data yang diberikan untuk membuat sebuah keputusan atau memberikan prediksi.

Oleh karena itu, data merupakan hal yang sangat penting. Jenis pembelajaran yang digunakan pada penelitian ini adalah klasifikasi (*classification*) dimana variabel *output* yang coba diprediksi berupa kategori. *Dataset* pada *machine learning* terdiri dari beberapa bagian seperti yang ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Struktur *Dataset*

Baris pada *dataset* disebut dengan sampel atau *data point* dan kolom pada *dataset* disebut dengan fitur atau atribut sedangkan kolom terakhir pada *dataset* disebut sebagai kelas. Fitur merupakan properti individual yang terukur atau karakteristik dari fenomena objek yang diamati (*input*) sedangkan kelas merupakan suatu hal yang coba diprediksi (*output*) [12]. Berdasarkan [6] ada beberapa faktor yang mempengaruhi produksi energi dari panel surya seperti suhu dari panel surya, tutupan cahaya matahari, pancaran radiasi yang diterima dan material penyusun sel yang digunakan. Pada penelitian ini, beberapa faktor dipilih untuk dijadikan fitur (*input*) yang mencoba memprediksi produksi energi listrik dari panel surya seperti yang diperlihatkan oleh Tabel 3.1. Data yang digunakan pada penelitian ini diambil dalam rentang waktu September 2017-Juli 2019. Setelah melakukan pengolahan, data sampel yang diperoleh menjadi 183 sampel data.

Tabel 3.1 Data *Input* dan *Output*

Atribut Data	Keterangan
Rata-rata Suhu Panel Surya Harian	Fitur (<i>input</i>)
Total Durasi Pancaran Sinar Matahari Harian	Fitur (<i>input</i>)
Rata-rata Radiasi Matahari Harian	Fitur (<i>input</i>)
Pembangkitan Energi Harian	Kelas (<i>output</i>)

- **Rata-Rata Suhu Panel Surya Harian (RSPSH)**

Merupakan suhu rata-rata panel surya dari jam 06.00-18.00 yang mana merupakan jam produksi energi dari panel surya. Satuan yang digunakan untuk RSPSH adalah °C (*Celcius*). Data diperoleh dari *database online* milik Solax Power (<https://solax-portal.com>, Akses pada Maret

2020). Situs ini merupakan situs penunjang dari produk panel surya yang terpasang di Gedung K.H. Mas Mansur milik Fakultas Teknologi Industri Universitas Islam Indonesia. Situs ini berfungsi untuk memonitor kinerja dari panel surya yang terpasang dan merekam data historisnya.

- **Total Durasi Pancaran Sinar Matahari Harian (TDPSMH)**

Data diambil dari situs *database online* milik Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) (<http://dataonline.bmkg.go.id>, Akses pada Maret 2020). Situs ini menyimpan data iklim dan aktivitas lempeng bumi untuk wilayah Indonesia yang direkam melalui stasiun BMKG yang tersebar di seluruh Indonesia. Satuan yang digunakan untuk TDPSMH adalah h (*hour*).

- **Rata-Rata Radiasi Matahari Harian (RRMH)**

Data diperoleh dari *database online* milik *National Aeronautics and Space Administration* (NASA) (<https://power.larc.nasa.gov>, Akses pada Maret 2020). Situs ini menyediakan *dataset* aktifitas matahari dan meteorologi dari penelitian NASA yang ditujukan untuk mendukung energi terbarukan, membangun efisiensi energi, dan kebutuhan pertanian. Satuan yang digunakan untuk RRMH adalah kWh/m²/day (*kilowatt hour per square meter per day*).

- **Pembangkitan Energi Harian (PEH)**

Merupakan data pembangkitan energi listrik oleh panel surya. Satuan yang digunakan untuk PEH adalah kWh (*kilowatt hour*). Data diperoleh dari situs yang sama dengan RSPSH (<https://solax-portal.com>, Akses pada Maret 2020). Pembangkitan energi harian akan digunakan sebagai target klasifikasi pada penelitian ini. Namun, sebelum itu PEH akan terlebih dahulu dibagi ke dalam beberapa label.

- **Labelling**

Data PEH yang diperoleh dari situs *monitoring* panel surya merupakan data ordinal, sehingga perlu dilakukan penglabelan untuk mempermudah proses klasifikasi. Label yang digunakan adalah “Rendah”, “Sedang”, dan “Tinggi” dengan menetapkan *range* tertentu untuk setiap labelnya.

Ada 2 tipe *dataset* yang ditemui dalam masalah pengklasifikasian, yaitu *dataset* yang tidak seimbang (*imbalanced dataset*) dan *dataset* yang seimbang (*balanced dataset*). Pada penelitian ini, *dataset* yang disiapkan adalah *balanced dataset*. Maksudnya jumlah pengamatan sama banyak disetiap labelnya seperti yang ditunjukkan pada Tabel 3.2. Pembagian *range* untuk masing-masing label pada penelitian ini berdasarkan pada *dataset* yang dikumpulkan, dimana nilai tertinggi yang diperoleh dikurangi nilai terendahnya kemudian dibagi 3 untuk mendapatkan *range* masing-masing label. Cara seperti ini juga dilakukan Bayindir, Yesibuldak, Colak, dan Genc dalam penelitiannya [3], dimana mereka membagi target klasifikasi menjadi 5 label berdasarkan dataset yang mereka punya.

Tabel 3.2 Label Klasifikasi

Label	Range (kWh/m ²)	Jumlah
Rendah	0,1 - 8,3	61
Sedang	8,31 - 16,5	61
Tinggi	16,51 - 24,7	61

Algoritma *machine learning* dapat menghasilkan prediksi lebih cepat ketika diberikan format data numerik. Oleh karena itu, *label encoding* digunakan untuk mengubah label menjadi bentuk numerik sehingga dapat diproses lebih cepat oleh mesin. Proses *encoding* dilakukan dengan menggunakan bantuan perangkat lunak *Microsoft Excel*. Adapun label produksi energi PV yang telah dirubah menjadi bentuk numerik ditampilkan oleh Tabel 3.3.

Tabel 3.3 Label Encoding

Label	Hasil Encoding
Rendah	0
Sedang	1
Tinggi	2

3.1.3 Pembuatan Program

Pembuatan program dilakukan dengan menggunakan bantuan perangkat lunak (*software*) *spyder*. *Software* ini dipilih karena memiliki *interface* yang ramah bagi pemula dan gampang diintegrasikan dengan *virtual environment anaconda*. *Anaconda environment* dibutuhkan untuk meng-*install* paket *library* seperti *pandas* dan *scikit-learn* yang digunakan pada penelitian ini. *Pandas* merupakan *software library* yang ditulis untuk bahasa pemrograman *python* yang memiliki fungsi untuk memanipulasi dan analisis data. Sedangkan *scikit-learn* merupakan *library* yang di dalamnya terdapat berbagai macam algoritma pembelajaran untuk *machine learning* [12].

3.1.4 Pembagian Data Latih dan Data Uji

Dataset yang telah diolah kemudian dibagi menjadi data latih dan data uji. Data latih merupakan data yang akan digunakan untuk membangun model sedangkan data uji digunakan untuk menguji seberapa baik model bekerja. Data uji harus dipisahkan dari data yang akan dilatih dikarenakan model dapat mengingat data yang digunakan untuk melatihnya sehingga prediksi akan selalu bernilai benar untuk data yang telah digunakan sebelumnya.

Pada penelitian ini proporsi pembagian data latih dan data uji dilakukan dengan 3 macam skenario pengujian seperti yang ditunjukkan oleh Tabel 3.4.

Tabel 3.4 Skenario Pengujian

Pengujian	Data Latih	Data Uji
Skenario 1	75%	25%
Skenario 2	80%	20%
Skenario 3	90%	10%

Berdasarkan pengujian yang ditampilkan pada Tabel 3.4, maka menghasilkan jumlah dari data latih dan data uji untuk masing-masing skenario seperti yang ditampilkan pada Tabel 3.5. Pembagian data latih dan data uji dilakukan secara acak dengan menggunakan bantuan *scikit-learn library*.

Tabel 3.5 Data Latih dan Data Uji

Pengujian	Data Latih	Data Uji	Jumlah
Skenario 1	137	46	183
Skenario 2	146	37	183
Skenario 3	164	19	183

3.1.5 Analisis Hasil Kinerja

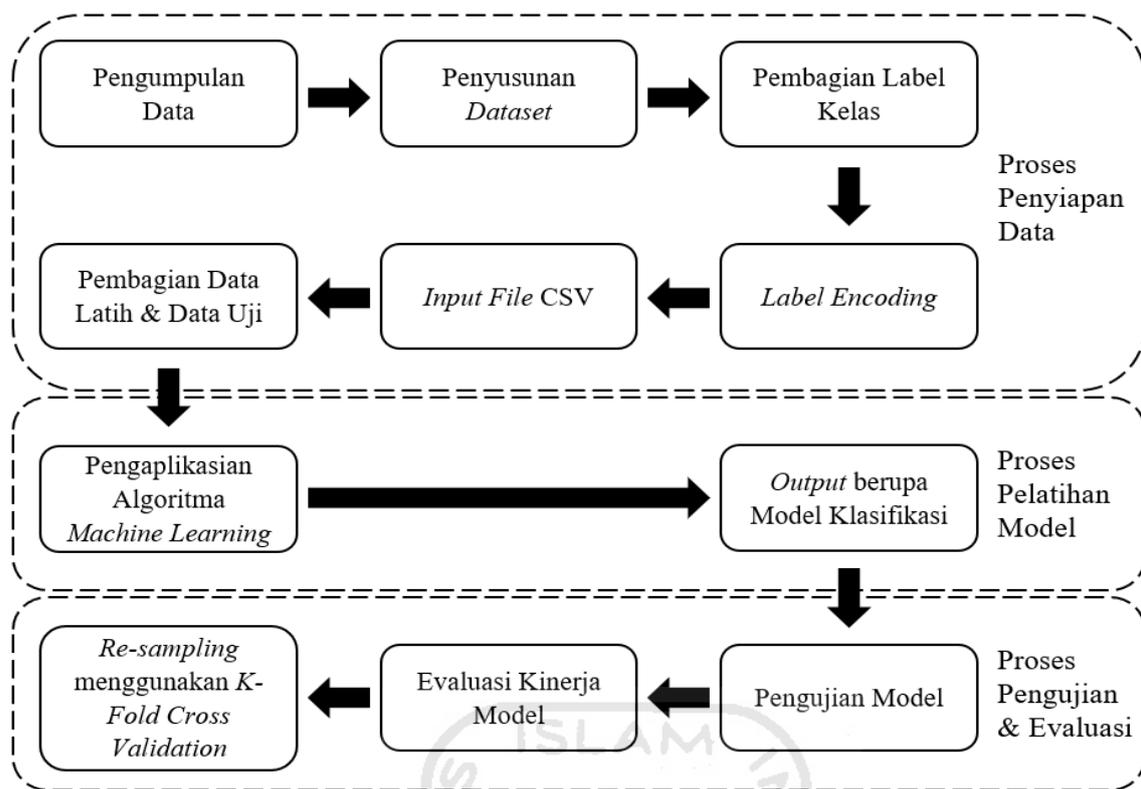
Analisa kinerja dari simulasi yang dilakukan pada penelitian ini dibagi menjadi 2 bagian. Pertama, membandingkan performa masing-masing skenario pada setiap algoritma. Kedua, membandingkan performa antar algoritma secara keseluruhan. Parameter yang digunakan untuk menganalisa performa kedua bagian tersebut adalah akurasi, presisi, *recall*, dan *f-score*.

3.1.6 Pembuatan Laporan

Pada tahap ini penelitian telah selesai dilakukan dan penulis membuat laporan yang berisi kegiatan selama penelitian dari awal studi literatur hingga akhir yang berisi analisa dan hasil kesimpulan dari penelitian.

3.2 Perancangan Sistem

Perancangan sistem dilakukan agar sistem berjalan dengan sistematis dan tidak terjadi eror ketika dilakukan pengujian. Adapun blok diagram dari sistem ditunjukkan oleh Gambar 3.3 beserta penjelasan mengenai setiap tahapannya.



Gambar 3.3 Blok Diagram Sistem

3.2.1 Proses Penyiapan Data

Fase pertama pada sistem berisi proses penyiapan *dataset* sebelum dimasukkan ke dalam algoritma *machine learning* untuk dilatih. Dimulai dengan mengumpulkan masing-masing atribut data dari sumber data yang telah dilampirkan sebelumnya untuk membuat suatu dataset yang terdiri dari 3 *input* dan 1 *output*. *Output* atau kelas dibagi menjadi 3 label kategori untuk mempermudah proses klasifikasi. Kemudian masing-masing label tersebut diubah menjadi bentuk numerik untuk mempercepat proses komputasi dengan bantuan perangkat lunak *Microsoft Excel*. Pada proses terakhir di fase ini, *dataset* dibagi menjadi 2 bagian, yaitu data latih dan data uji berdasarkan beberapa skenario pengujian.

3.2.2 Proses Pelatihan Model

Data latih yang telah dipisah selanjutnya akan digunakan untuk membangun model klasifikasi dengan mengaplikasikannya ke dalam algoritma *machine learning*. Algoritma yang digunakan pada penelitian adalah *Naïve Bayes* dan KNN.

- **Penentuan nilai parameter K pada algoritma KNN**

Nilai K yang digunakan pada penelitian ini adalah 5 (K=5) seperti yang disebutkan pada batasan masalah penelitian dikarenakan nilai tersebut merupakan nilai *default* yang diberikan pada

library scikit-learn. Sebagai perbandingan, maka dilakukan pengujian singkat dengan menggunakan nilai parameter K 1-5 dengan hasil yang ditunjukkan oleh Tabel 3.6. Dimana seperti yang terlihat bahwa akurasi yang dihasilkan ketika K bernilai 5 lebih tinggi ketika dibandingkan dengan hasil keempat nilai lainnya.

Tabel 3.6 Pengujian Singkat Nilai K

Nilai K	Akurasi
1	70%
2	80%
3	78%
4	80%
5	83%

3.2.3 Proses Pengujian & Evaluasi

Pada tahapan ini, model klasifikasi yang telah dibangun menggunakan data latih akan diuji hasil prediksi label kelasnya ketika diberikan fitur (*input*) data uji. Hasil prediksi kemudian dibandingkan dengan nilai label data uji sesungguhnya untuk melihat bagaimana kinerja model klasifikasi. Parameter yang digunakan untuk menganalisis kinerja model adalah akurasi, presisi, *recall*, dan *f-score*.

Proses terakhir pada penelitian ini adalah melakukan *re-sampling* untuk melihat apakah model tergeneralisasi dengan baik atau tidak (terjadi *overfitting & underfitting*). *Overfitting* adalah suatu keadaan dimana data yang digunakan untuk pelatihan itu adalah yang "terbaik". Sehingga apabila dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi (hasil yang dibuat tidak sesuai yang diharapkan). Sedangkan *Underfitting* merupakan keadaan dimana model pelatihan data yang dibuat tidak mewakili keseluruhan data yang akan digunakan nantinya. Sehingga menghasilkan performa yang buruk dalam pelatihan data. Oleh karena itu, pada penelitian ini akan digunakan teknik *K-Fold Cross Validation*.

K-Fold Cross Validation merupakan metode statistik untuk mengevaluasi kinerja generalisasi yang lebih stabil dan menyeluruh daripada menggunakan pemisahan *dataset* menjadi satu set pelatihan dan pengujian. Dalam *K-Fold Cross Validation*, data akan dibagi beberapa bagian berulang kali (iterasi) dan kemudian model dilatih. Nilai dari parameter K menentukan seberapa banyak *dataset* akan dibagi dan seberapa banyak iterasi proses *training* dan *testing*. Tidak ada aturan khusus dalam penentuan nilai K. Namun, biasanya nilai K yang ditetapkan adalah K=5 atau K=10 [12]. Pada penelitian ini, nilai K yang digunakan adalah 10 (K=10).

BAB 4

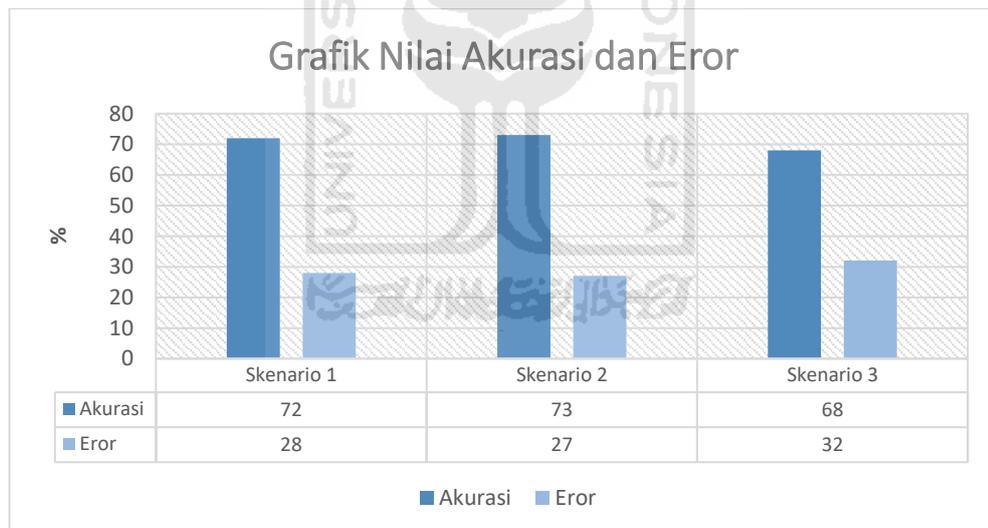
HASIL DAN PEMBAHASAN

4.1 Simulasi *Naïve Bayes*

Pada bagian ini, model *machine learning* telah selesai dilatih menggunakan algoritma *Naïve Bayes*. Prediksi yang dihasilkan kemudian dianalisis menggunakan *confusion matrix*. Pada *confusion matrix* terdapat beberapa informasi yang bisa dimanfaatkan untuk mengevaluasi kinerja model. Diantaranya adalah akurasi, eror, presisi, *recall*, dan *f-score*. Hasil dari setiap skenario dianalisa berdasarkan informasi yang didapat dari *confusion matrix*.

4.1.1 Akurasi dan Eror Pada Model *Naïve Bayes*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter akurasi dan eror dapat dilihat pada Gambar 4.1.



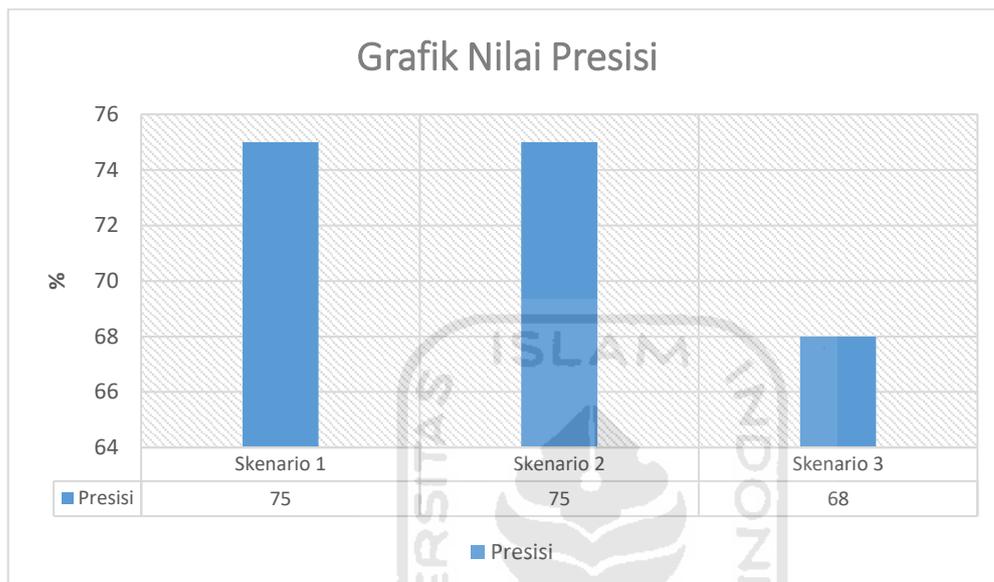
Gambar 4.1 Hasil Grafik Nilai Akurasi dan Eror Pada Model *Naïve Bayes*

Dari hasil Gambar 4.1 dapat dilihat, akurasi dari masing-masing skenario tidak jauh berbeda satu sama lain. Pada kasus data ini, model *naïve bayes* memiliki akurasi di kisaran 70%. Pada skenario 1 akurasi bernilai 72%, kemudian pada skenario 2 akurasi sedikit bertambah sehingga menjadi 73% dan turun pada skenario 3 menjadi 68%. Hasil akurasi yang tidak linear pada ketiga skenario menunjukkan bahwa model yang dilatih kurang bisa merepresentasikan data dengan baik. Salah satu cara yang digunakan untuk mengoptimalkan model yang kurang bagus adalah dengan menambahkan data sampel.

Dikarenakan nilai eror diperoleh melalui persamaan (2.11), maka hasil eror pada model berbanding terbalik dengan akurasi untuk kasus data ini.

4.1.2 Presisi Pada Model *Naïve Bayes*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter presisi dapat dilihat pada Gambar 4.2.

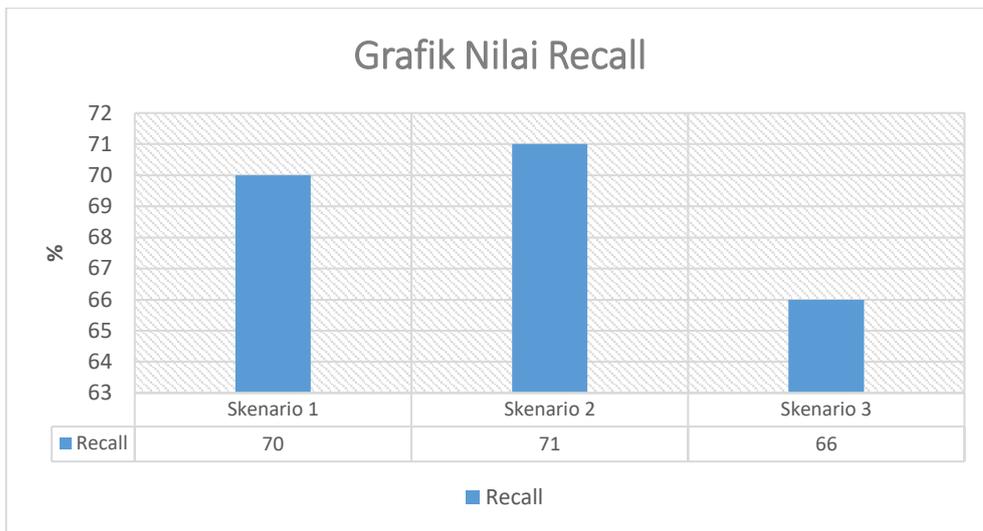


Gambar 4.2 Hasil Grafik Nilai Presisi Pada Model *Naïve Bayes*

Dari hasil Gambar 4.2 dapat dilihat, presisi dari model skenario 1 dan 2 sama baiknya sedangkan skenario 3 memiliki presisi yang lebih rendah. Pada skenario 1 dan 2, presisi bernilai 75% kemudian turun menjadi 68% pada skenario 3. Hasil ini menunjukkan bahwa pada skenario 3 dengan rasio pembagian data uji dan latih sebesar 90:10, prediksi banyak menghasilkan *false positive* sehingga presisi turun cukup jauh [12].

4.1.3 Recall Pada Model *Naïve Bayes*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *recall* dapat dilihat pada Gambar 4.3.

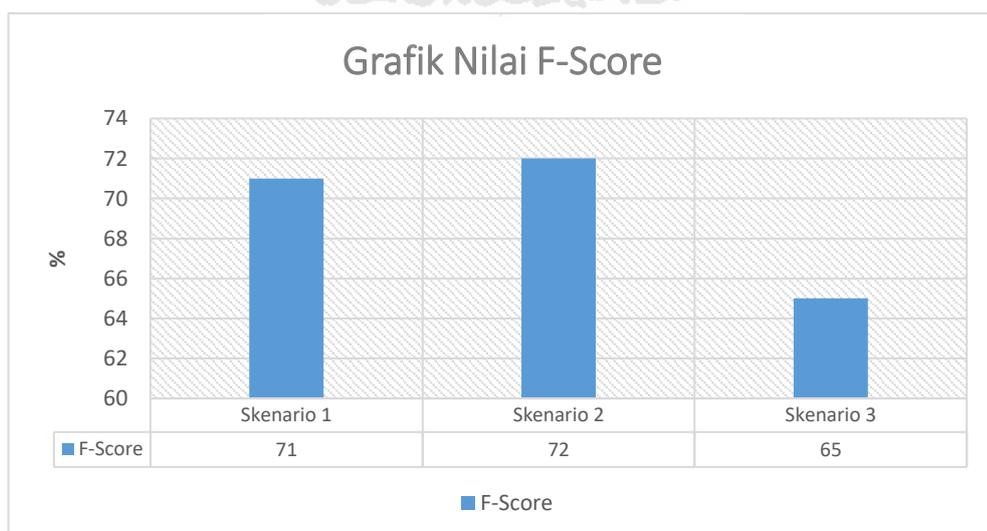


Gambar 4.3 Hasil Grafik Nilai *Recall* Pada Model *Naïve Bayes*

Dari hasil Gambar 4.3 dapat dilihat, Skenario 2 memiliki nilai *recall* terbaik. Pada skenario 1 *recall* bernilai 70% kemudian naik menjadi 71% pada skenario 2 dan turun pada skenario 3 hingga 66%. Hasil ini menunjukkan model skenario 3 tidak dilatih dengan terlalu baik, karena selain banyak menghasilkan prediksi *false positive* seperti diungkapkan hasil presisi, juga menghasilkan *false negative* sehingga nilai *recall* sedikit lebih rendah dari presisi.

4.1.4 *F-Score* Pada Model *Naïve Bayes*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *f-score* dapat dilihat pada Gambar 4.4



Gambar 4.4 Hasil Grafik Nilai *F-Score* Pada Model *Naïve Bayes*

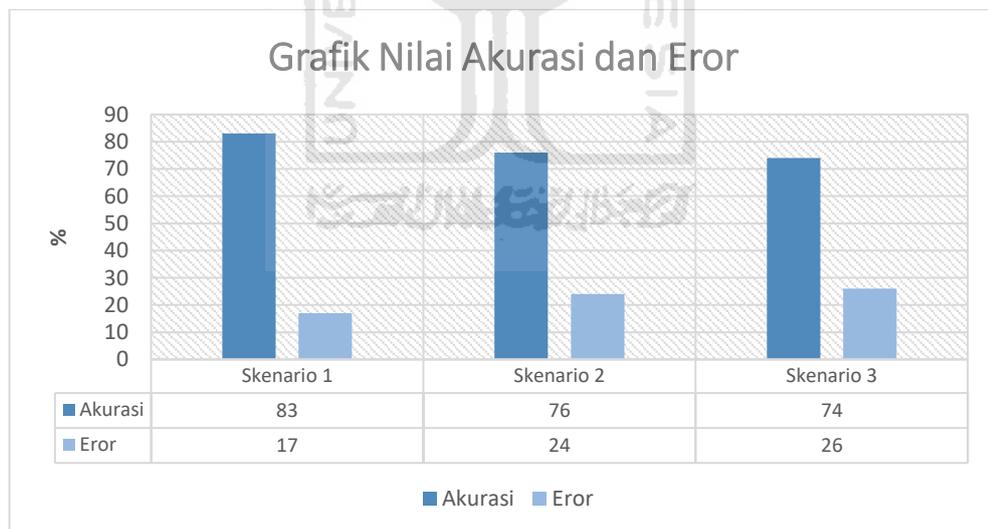
Dari hasil Gambar 4.4 dapat dilihat, Skenario 2 memiliki nilai *f-score* terbaik. Pada skenario 1 *f-score* bernilai 71% kemudian naik menjadi 72% pada skenario 2 dan turun menjadi 65%. Hasil *f-score* hampir mirip dengan presisi dan *recall* karena merupakan rata-rata harmonik dari keduanya.

4.2 Simulasi *K-Nearest Neighbors*

Pada bagian ini, model *machine learning* telah selesai dilatih menggunakan algoritma KNN. Prediksi yang dihasilkan kemudian dianalisis menggunakan *matrix evaluation* atau disebut dengan *confusion matrix*. Pada *confusion matrix* terdapat beberapa informasi yang bisa dimanfaatkan untuk mengevaluasi kinerja model. Diantaranya adalah akurasi, eror, presisi, *recall*, dan *f-Score*. Hasil dari setiap skenario dianalisa berdasarkan informasi yang didapat dari *confusion matrix*.

4.2.1 Akurasi dan Eror Model *K-Nearest Neighbors*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter akurasi dan eror dapat dilihat pada Gambar 4.5.



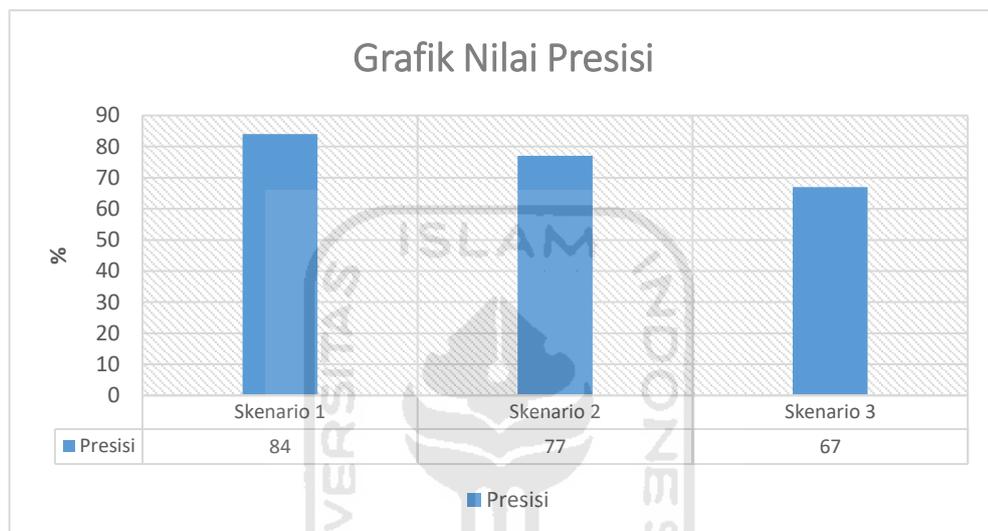
Gambar 4.5 Hasil Grafik Nilai Akurasi dan Eror Pada Model KNN

Dari hasil Gambar 4.5 dapat dilihat, bahwa pada kasus data yang digunakan penelitian ini, skenario pembagian data latih dan uji 1 memiliki akurasi terbaik dan akurasi akan semakin turun ketika data uji yang diberikan berkurang. Pada skenario 1 akurasi bernilai 83% kemudian turun menjadi 76% hingga bernilai 74% pada skenario 3. Skenario 1 dengan rasio pembagian data uji dan latih sebesar 75:25, dianggap paling optimal dalam membagi *dataset* pada *machine learning*.

Sedangkan untuk eror, dikarenakan nilai diperoleh melalui persamaan (2.11), maka hasil eror pada model berbanding terbalik dengan akurasi untuk kasus data ini. Pada skenario 1 terjadi eror sebesar 17% kemudian bertambah pada skenario 2 menjadi 24% dan skenario 3 memiliki eror paling besar dengan nilai 26%.

4.2.2 Presisi Model *K-Nearest Neighbors*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter presisi dapat dilihat pada Gambar 4.6.

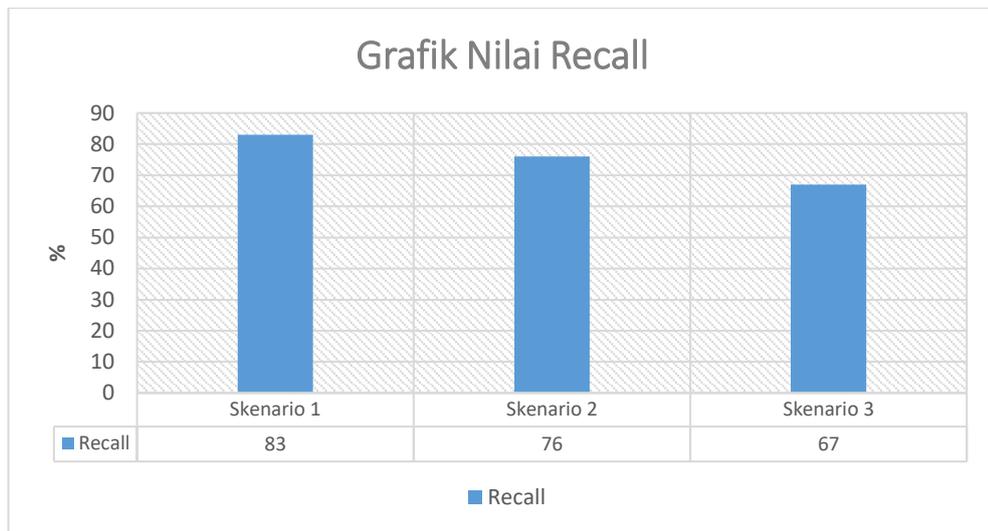


Gambar 4.6 Hasil Grafik Nilai Presisi Pada Model KNN

Dari hasil Gambar 4.6 dapat dilihat, skenario 1 memiliki nilai presisi yang paling tinggi sebesar 84% kemudian turun pada skenario 2 menjadi 77%. Skenario 3 memiliki nilai presisi yang jauh lebih rendah dari skenario 1 dengan nilai presisi 67%. Hasil ini menggambarkan bahwa pada saat rasio pembagian data latih dan uji menjadi 90:10 (skenario 3) model banyak menghasilkan prediksi *false positive* sehingga nilai presisi turun.

4.2.3 Recall Model *K-Nearest Neighbors*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *recall* dapat dilihat pada Gambar 4.7.

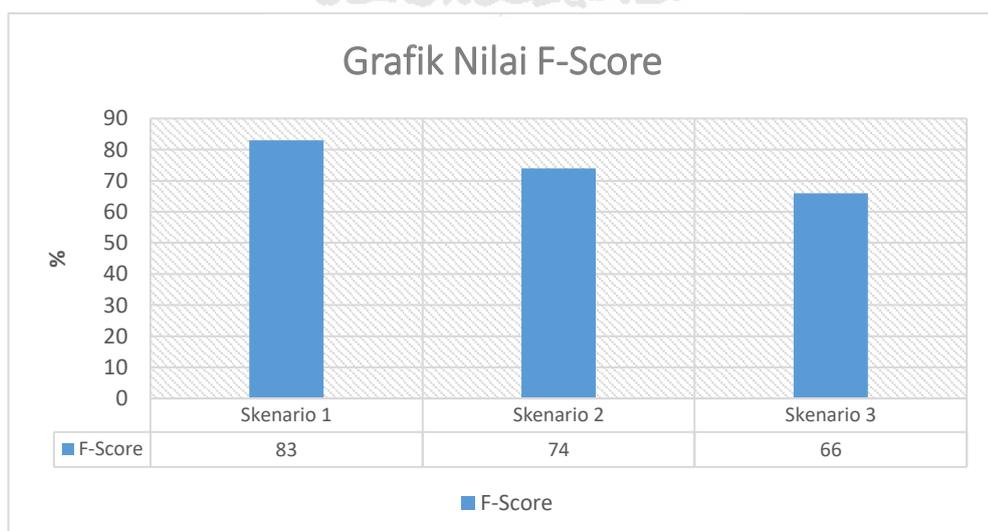


Gambar 4.7 Hasil Grafik Nilai *Recall* Pada Model KNN

Dari hasil Gambar 4.7 dapat dilihat, Skenario 1 memiliki nilai *recall* paling baik sebesar 83% kemudian turun pada skenario 2 menjadi 76% hingga bernilai 67% pada skenario 3. Sama seperti presisi, nilai *recall* pada masing-masing skenario semakin berkurang seiring berkurangnya sampel data uji. Hasil ini menunjukkan pada skenario 3 prediksi banyak menghasilkan *false negative* sehingga nilai *recall* turun cukup jauh dari skenario 1.

4.2.4 *F-Score* Model *K-Nearest Neighbors*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *f-score* dapat dilihat pada Gambar 4.8.

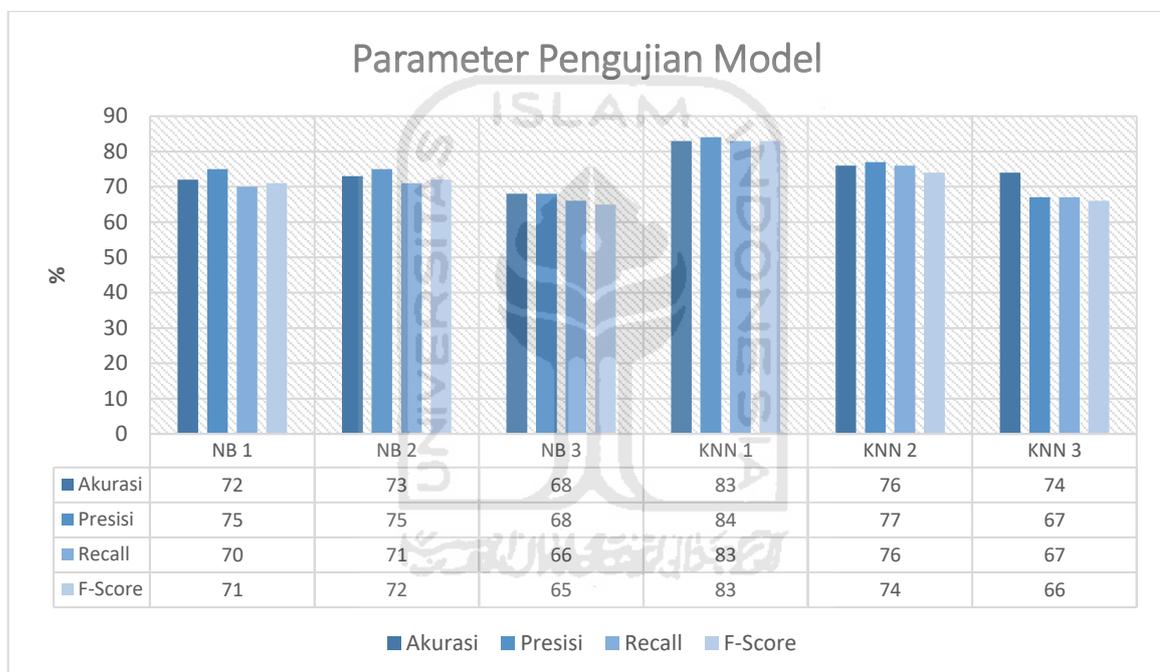


Gambar 4.8 Hasil Grafik Nilai *F-Score* Pada Model KNN

Dari hasil Gambar 4.8 dapat dilihat, bahwa nilai *f-score* terbaik dimiliki oleh skenario 1 dan semakin turun seiringnya berkurangnya data uji yang digunakan. Pada skenario 1 *f-score* bernilai 83% kemudian turun menjadi 74% dan bernilai 66% pada skenario 66. Hasil dari *f-score* tidak berbeda jauh dari presisi dan *recall* karena merupakan rata-rata harmonik dari keduanya.

4.3 Perbandingan Performa Antar Algoritma

Perbandingan hasil model klasifikasi dari setiap skenario pengujian yang telah dilatih menggunakan algoritma *Naïve Bayes* dan KNN dilakukan untuk melihat kinerja dari masing-masing model. Untuk menganalisis masing-masing model, maka digunakan bantuan dari *confusion matrix* dengan parameter berupa akurasi, presisi, *recall*, dan *f-score*. Hasil simulasi dari parameter tersebut dapat dilihat pada Gambar 4.9.



Gambar 4.9 Hasil Grafik Nilai Tiap Parameter Pengujian Model

Pada pembuatan model *machine learning*, secara umum rasio pembagian untuk data latih berkisar pada 40% - 80% dari keseluruhan *dataset* namun, hal tersebut masih bisa tergantung oleh karakteristik dari masing-masing *dataset* [13]. *Dataset* yang digunakan pada penelitian ini terbilang cukup kecil dimana hanya memiliki 183 sampel. Oleh karena itu, pada penelitian ini penulis ingin mengetahui hasil dari berbagai skenario pengujian rasio pembagian data latih dan data uji dan pengaruhnya terhadap *dataset* yang kecil. Setelah melakukan pengujian model, didapati bahwa ketika data uji semakin sedikit maka model akan menghasilkan persentase yang rendah dari setiap parameter. Hal ini disebabkan karena pengaruh dari prediksi model yang salah terhadap data uji baik itu *false positive* maupun *false negative* akan sangat mempengaruhi dari

persentase parameter. Sebagai contoh, pada pengujian skenario 3 algoritma *Naïve Bayes*, model menghasilkan prediksi salah sebanyak 6 sampel dari 19 total sampel data uji yang mana berarti terjadi eror klasifikasi sekitar kurang dari 1/3 total data uji yang diberikan, hal ini sangat mempengaruhi persentase dari parameter pengujian. Sebagai perbandingan, skenario 1 *Naïve Bayes* menghasilkan prediksi salah sebanyak 13 sampel dari 46 data uji yang diberikan. Skenario 1 *Naïve Bayes* memiliki prediksi salah yang lebih banyak namun, menghasilkan model yang lebih baik. Oleh karena itu, jumlah sampel pada *dataset* perlu diperhatikan seperti yang diungkapkan Xu pada penelitiannya [14], dimana variasi dari *correct classification rate* akan berkurang seiring meningkatnya jumlah *dataset* yang menunjukkan bahwa pada *dataset* yang besar skenario pembagian rasio *dataset* tidak terlalu berpengaruh.

Sedangkan untuk kasus *dataset* kecil seperti penelitian ini, rasio pembagian perlu dipertimbangkan, tetapi tentu saja akan lebih baik jika mempebaharui *dataset* dengan menambahkan sampel baru atau fitur baru sehingga model bisa memiliki bahan pembelajaran yang lebih banyak lagi. Performa model yang rendah ketika diberikan “bahan pembelajaran” yang lebih banyak juga terjadi pada penelitian milik Nugraha [15], dimana pada penelitiannya rasio pembagian data latih dan data uji sebesar 60:40 menghasilkan performa yang lebih rendah daripada skenario dengan rasio 40:60. Hal ini disebabkan oleh *dataset* yang kecil/kurang banyak.

Dari hasil Gambar 4.9 dapat dilihat, pembagian data latih dan uji menjadi 80:20 (skenario 2) merupakan skenario pengujian terbaik untuk algoritma *Naïve Bayes* pada kasus data penelitian ini. Skenario 2 memiliki nilai yang stabil di setiap parameter pengujian. Namun, hasil ini memiliki akurasi dan *recall* yang lebih rendah dari milik model yang dijadikan acuan pada studi literatur (“Akurasi: 82,191% “ dan “*Recall*: 98,305%”) [3] dan (“Akurasi: 94,09%”) [5]. Hal ini disebabkan oleh kurangnya data penelitian. *Naïve Bayes* memiliki kelebihan dalam pengujian menggunakan *dataset* yang besar. Memiliki lebih banyak data yang berkualitas dapat membantu membangun model klasifikasi (*classifier*) menjadi lebih baik [12].

Pada pelatihan model menggunakan algoritma KNN, Skenario 1 memiliki kinerja yang stabil pada setiap parameternya kemudian performa terus turun ketika data uji semakin sedikit. Hasil dari akurasi yang dihasilkan oleh skenario 1 lebih baik daripada milik salah satu model acuan pada studi literatur namun, tertinggal pada nilai *recall* (“Akurasi: 82,191% “ dan “*Recall*: 98,305%”) [3]. Ketika dibandingkan dengan hasil dari penelitian milik putri, suparti, dan rahmawati (“Akurasi: 96,09%”) [5], model yang dihasilkan memiliki akurasi yang lebih rendah.

Hasil pengujian menunjukkan bahwa untuk kasus data penelitian ini, secara keseluruhan model yang dilatih dengan menggunakan algoritma KNN lebih baik dari pada *Naïve Bayes* pada masing-masing skenario pengujian. Perlu dilakukan optimisasi tambahan pada model yang dihasilkan oleh *Naïve Bayes* seperti menambahkan fitur baru dan sampel data.

4.4 Pengujian *K-Fold Cross Validation*

Setelah melakukan *re-sampling* menggunakan *K-Fold Cross Validation*, maka didapati hasil pengujian dari setiap algoritma seperti yang ditampilkan Tabel 4.1. Dari hasil ini, bisa dikatakan bahwa terjadi *overfitting* pada model yang telah dilatih sebelumnya ketika dilakukan pembagian data latih dan uji secara *random* menggunakan *scikit-learn library*. Model yang dihasilkan terlalu spesifik untuk data tertentu sehingga memiliki akurasi yang lumayan tinggi, yang mana jika diberikan *input* data baru maka performa model akan turun. Salah satu cara yang umum digunakan untuk mengatasi permasalahan ini adalah dengan melakukan *re-sampling*. Hasil pengujian menunjukkan bahwa akurasi yang dihasilkan setelah melakukan *re-sampling* lebih rendah dibandingkan dengan pengujian yang membagi data latih dan uji secara acak.

Pada hasil pengujian, didapati bahwa skenario 3 KNN memiliki akurasi tertinggi di antara 6 skenario pengujian yang dilakukan dengan nilai akurasi sebesar 74% dan Skenario 1 *Naïve Bayes* merupakan model dengan akurasi terendah dengan nilai akurasi sebesar 55%. Dari hasil ini juga didapati bahwa akurasi akan meningkat ketika diberikan data latih yang lebih banyak.

Tabel 4.1 Akurasi *K-Cross Fold Validation*

Iterasi	Akurasi <i>K-Cross Fold Validation</i>					
	NB 1	NB 2	NB 3	KNN 1	KNN 2	KNN 3
1	43%	67%	71%	57%	80%	82%
2	50%	53%	47%	57%	67%	65%
3	64%	47%	47%	86%	60%	65%
4	64%	73%	59%	64%	80%	94%
5	64%	67%	63%	64%	60%	69%
6	71%	53%	56%	71%	60%	63%
7	71%	71%	81%	86%	79%	88%
8	46%	50%	69%	62%	71%	69%
9	38%	43%	44%	62%	64%	63%
10	38%	43%	44%	54%	57%	56%
Rata-rata	55%	57%	58%	66%	68%	71%

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah penulis lakukan, kesimpulan yang dapat diambil penulis, yaitu:

1. Pada kasus data penelitian ini, model yang dilatih dengan menggunakan algoritma KNN memiliki kinerja yang lebih baik daripada yang dilatih menggunakan algoritma *Naïve Bayes*.
2. Penurunan performa model secara keseluruhan ketika pembagian data latih dan data uji mengikuti skenario pengujian 3 (90:10).
3. Dalam melakukan prediksi terhadap energi yang dapat dihasilkan oleh panel surya penting untuk memilih fitur atau atribut yang sesuai.

5.2 Saran

Setelah melakukan penelitian, penulis memiliki beberapa saran untuk penelitian selanjutnya agar mendapatkan hasil yang lebih baik:

1. Untuk melakukan prediksi terhadap energi listrik yang dihasilkan dapat menambahkan sampel data dan juga fitur-fitur baru sehingga model memiliki “bahan pembelajaran” yang lebih banyak lagi.
2. Melakukan tahap lanjutan setelah mengevaluasi model, yaitu *Model Deployment* sehingga hasil prediksi bisa diakses secara *online*.

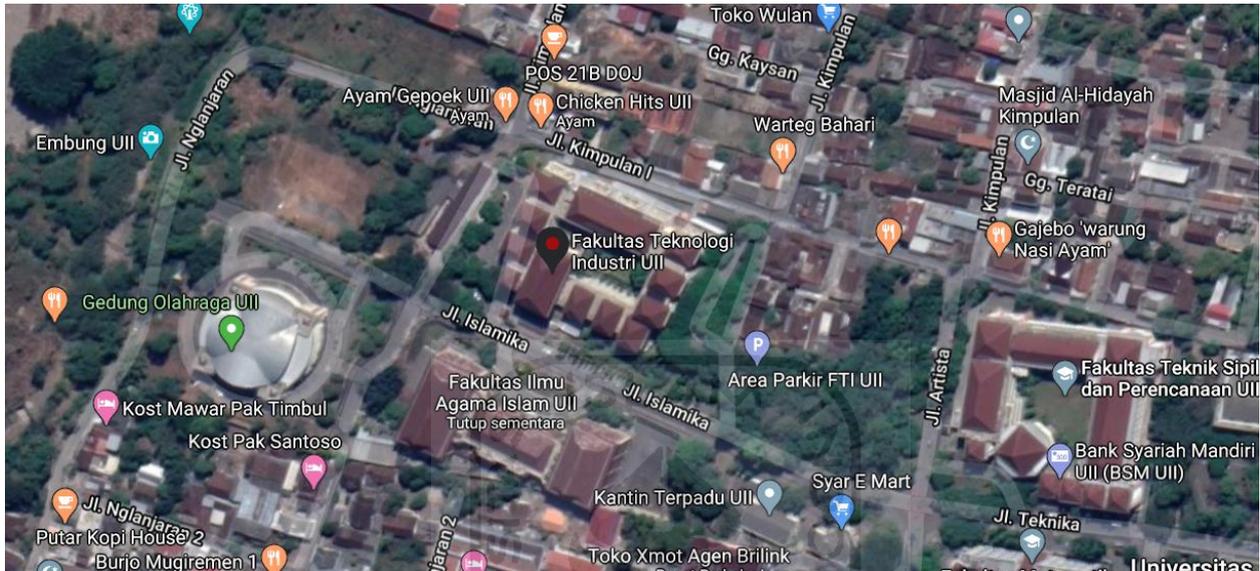
DAFTAR PUSTAKA

- [1] S. M. Awan, Z. A. Khan, and M. Aslam, "Solar Generation Forecasting by Recurrent Neural Networks optimized by Levenberg-Marquardt Algorithm," *IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 1, pp. 276–281, 2018.
- [2] T. R. Society, *Machine learning: the power and promise of computers that learn by example*. 2017.
- [3] R. Bayindir, M. Yesilbudak, M. Colak, and N. Genc, "A novel application of naive bayes classifier in photovoltaic energy prediction," *Proc. - 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017*, vol. 2018-Janua, pp. 523–527, 2018.
- [4] S. Nam and J. Hur, "Probabilistic Forecasting Model of Solar Power Outputs Based on the Na ve Bayes Classifier and Kriging Models," *Energies*, vol. 11, no. 11, 2018.
- [5] R. E. Putri, Suparti, and R. Rahmawati, "Perbandingan Metode Klasifikasi Naive Bayes dan K-Nearest Neighbors Pada Analisis Data Status Kerja di Kabupaten Demak Tahun 2012," vol. 3, pp. 831–838, 2014.
- [6] H. Sharadga, S. Hajimirza, and R. S. Balog, "Time series forecasting of solar power generation for large-scale photovoltaic plants," *Renew. Energy*, 2020.
- [7] E. Commission, *Photovoltaic solar energy*. 2009.
- [8] M. Wasfi and S. Member, "Solar Energy and Photovoltaic Systems," no. June, 2014.
- [9] R. Nadia, F. Nhita, F. Informatika, U. Telkom, and M. Online, "ANALISIS DAN IMPLEMENTASI ALGORITMA NAÏVE BAYES CLASSIFIER TERHADAPA PEMILIHAN GUBERNUR JAWA BARAT 2018 PADA MEDIA ONLINE," vol. 5, no. 1, pp. 1678–1700, 2018.
- [10] J. Wira and G. Putra, "Pengenalan Pembelajaran Mesin dan Deep Learning Jan Wira Gotama Putra Pengenalan Konsep Pembelajaran Mesin dan Deep Learning," no. March, 2018.
- [11] A. R. Alfarisi, H. Tjandra, and I. Arieshanti, "Perbandingan Performa antara Imputasi Metode Konvensional dan Imputasi dengan Algoritma," vol. 2, no. 1, pp. 1–4, 2013.
- [12] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python*. 2016.
- [13] K. K. Dobbin and R. M. Simon, "Optimally splitting cases for training and testing high dimensional classifiers," *BMC Med. Genomics*, vol. 4, no. 1, p. 31, 2011.
- [14] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set : A Comparative Study of Cross - Validation , Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *J. Anal. Test.*, no. 0123456789, 2018.
- [15] P. A. Nugraha, "Perbandingan Metode Probabilistik Naive Bayesian Classifier dan Jaringan Syaraf Tiruan Learning Vector Quantization dalam Kasus Klasifikasi Penyakit Kandungan," vol. 2, no. 2, pp. 20–33, 2013.

LAMPIRAN

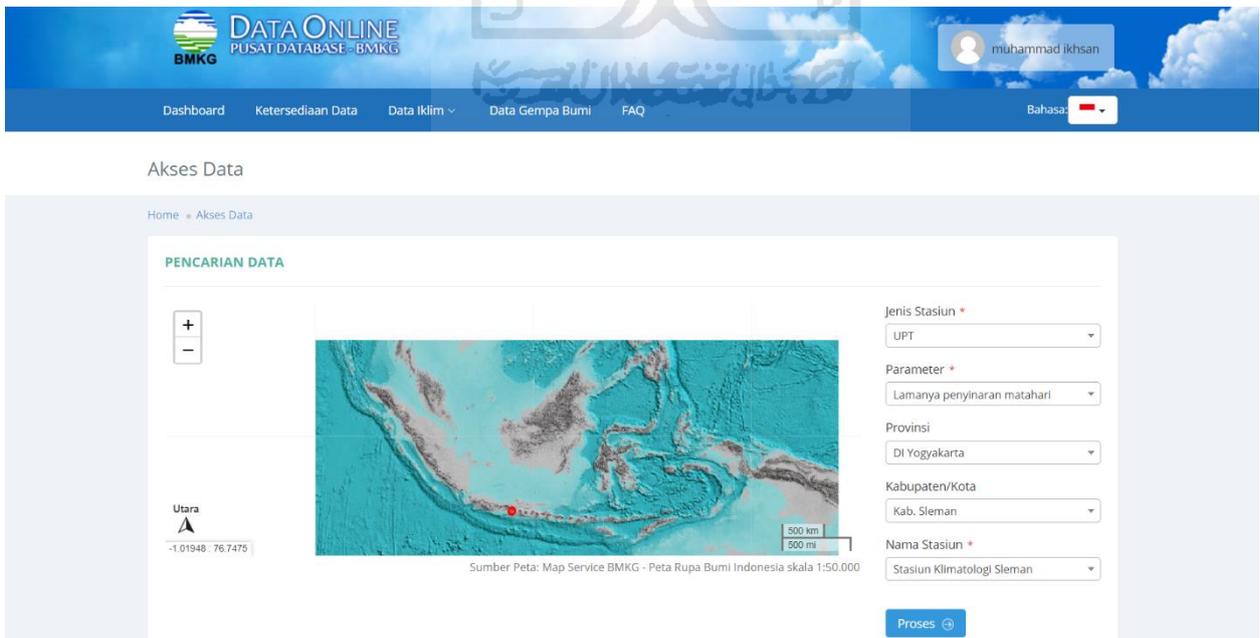
Lampiran 1 – Lokasi Sistem PV yang Dijadikan Studi

Alamat: Gedung K.H. Mas Mansyur, Kampus Terpadu Universitas Islam Indonesia, Jalan Kaliurang KM 14,5, Umbulmartani, Ngemplak, Krawitan, Umbulmartani, Kec. Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584



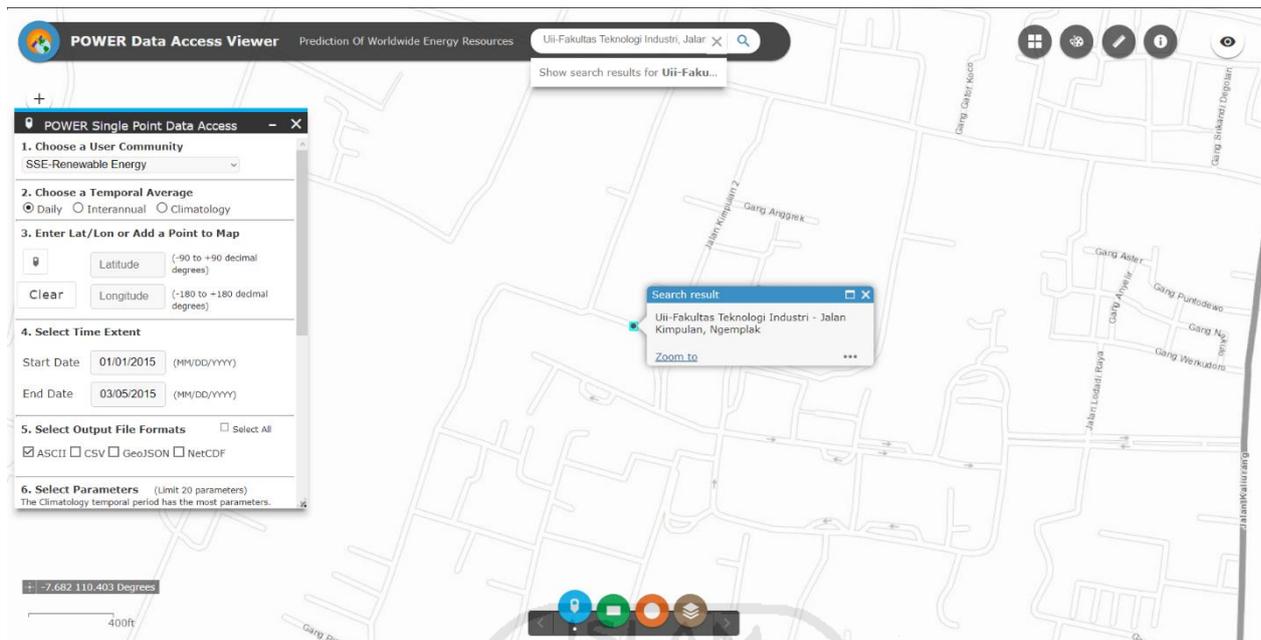
Lampiran 2 – Database Online BMKG

Alamat web: <http://dataonline.bmkg.go.id>, Akses pada Maret 2020



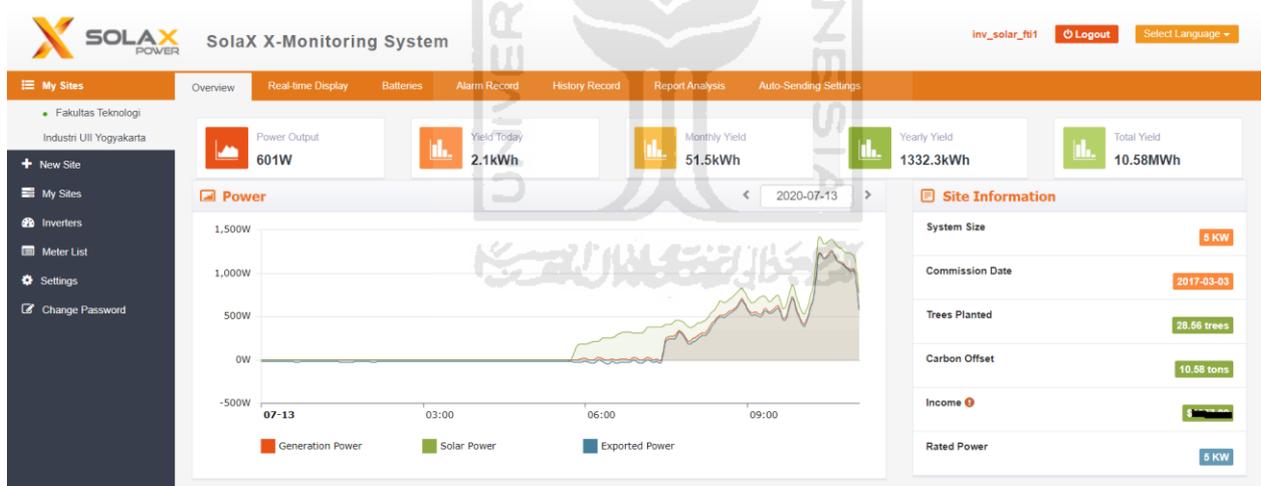
Lampiran 3 - Database Online NASA

Alamat web: <https://power.larc.nasa.gov>, Akses pada Maret 2020



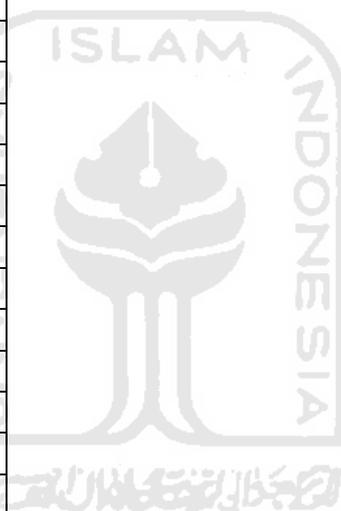
Lampiran 4 – Situs Monitoring SolaX Portal

Alamat web: <https://solax-portal.com>, Akses pada Maret 2020



Lampiran 5 - Dataset

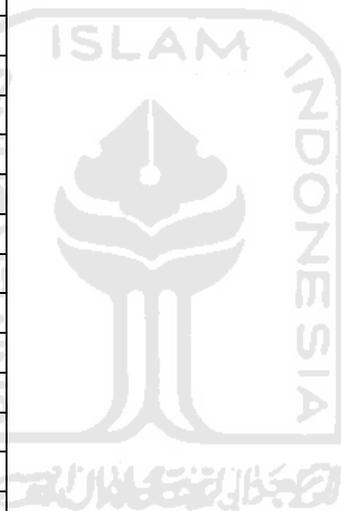
Dataset			
RSPSH	TDPSMH	RRMH	Kelas
33	1,9	4,43	2
35	3,9	4,87	2
32	9,7	5,42	0
21	10,2	7,18	2
35	4,4	5,36	0
21	4	4,25	1
33	8,4	5,56	0
34	6,6	4,3	1
34	5,1	4,7	1
22	4,5	3,76	2
35	8,6	4,49	2
34	8,4	5,75	2
34	2,9	1,8	0
22	4,2	6	1
34	5,6	4,97	1
34	9,6	5,56	0
35	5,2	6,83	2
22	4	3,66	1
35	5,2	4,46	1
35	6,6	5,1	1
35	6,9	5	2
33	10,9	5,53	0
35	7,2	4,43	1
34	4,3	3,92	1
22	6,6	6,47	2
34	5,9	5,19	1
35	4,7	3,21	0
34	4,6	4,71	1
34	5,3	5,38	1
33	10	6,22	0
22	6,1	6,33	2
22	0,4	1,95	0
35	6,9	5,75	0
34	4,1	3,98	1
35	6,5	6,56	2
36	5,1	5,13	1
22	0,1	3,35	0
33	8,3	5,79	0
22	10	6,87	2
35	8,5	5,47	0
22	7,3	6,54	2
34	4,9	4,16	1
35	8,2	5,43	0



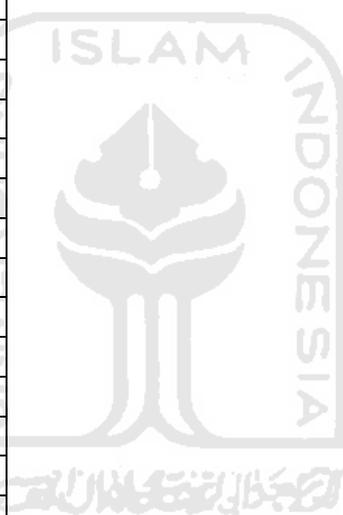
35	5,3	5,1	2
22	6,7	4,11	1
22	5	5,1	1
22	4,8	2,71	1
34	7,5	4,23	2
35	8,8	5,81	0
34	3,4	2	0
34	5,2	3,62	1
33	1	3,2	0
34	4	3,72	2
22	8	5,65	2
36	5,6	5,6	1
22	6,5	6,66	2
35	5,3	6,28	2
35	8,7	5,55	0
34	5,1	4,7	2
34	8,2	5,82	2
36	7,8	5,16	0
33	10,4	5,54	0
33	10,1	5,7	0
35	9	5,72	0
36	7,7	4	1
35	9,6	5,97	2
33	3,5	3,78	0
34	4,8	4,73	2
33	8,5	5,28	0
35	4	3,5	1
35	5,1	4,45	1
32	10	5,63	0
36	7,2	4,27	1
35	9,4	5,78	2
35	5,7	5,37	2
34	8,1	5,64	2
33	0,3	4,81	0
34	3	5,5	2
35	8,2	5,19	0
22	4	3,8	2
35	7,4	5,3	2
35	4,1	5,84	1
36	8,9	6,1	0
33	4,2	5,13	1
33	9,9	5,66	0
34	8,4	6,46	2
34	6,3	4,79	1
22	2,2	4,52	1
22	10,2	6,99	2
33	5,5	3,77	1
36	5,6	5,62	0



35	5,4	4,66	1
35	6,6	4,79	1
34	11,6	4,81	2
34	4,8	3,26	1
22	9,3	6,68	2
33	10,6	5,4	0
34	7,2	6,32	2
22	4,4	6,2	2
33	0,3	4,5	2
22	6,8	4,66	1
32	10,8	5,77	0
22	6,8	5,8	2
35	7,1	4,68	1
33	5	5,15	2
35	3,7	4,41	0
34	8,1	5,56	0
33	3,3	4,19	0
22	5,4	4,59	1
34	9	5,74	0
34	0,3	3,25	0
22	6,8	5	1
29	7,3	4,16	1
32	9,5	5,69	0
34	8,9	5,35	0
22	8	5,69	2
34	2,2	5,11	2
34	4,1	4,44	1
35	6,2	4,32	0
22	4	4,63	1
34	7,8	6,26	2
35	6,2	4,81	1
35	5,1	5,18	0
22	4	1,72	1
35	10,4	5,62	0
35	4,2	5,16	1
22	2,2	5,65	2
33	9,9	5,65	0
33	8,6	5,5	0
32	10,5	5,53	0
34	3	5,16	2
35	8,9	5,84	0
35	4,2	3,54	1
35	6,1	7,1	2
33	1	4,73	2
32	6,2	5,16	1
36	3,5	5,99	0
22	5,4	6,46	2
22	8,2	6,91	2

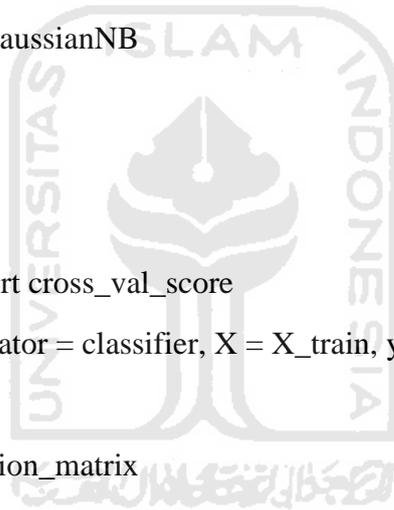


32	1,9	3	0
34	3,7	3,67	0
35	4,2	5,49	1
33	9,7	5,84	0
22	10,1	6,71	2
34	1,9	4,9	2
35	7,8	5,31	0
35	6,1	3,18	1
33	9,9	5,92	0
34	2,2	4,7	1
32	8	5,88	0
33	5,5	4,19	1
34	4,3	4,42	1
22	4	3,45	1
22	6,3	4,73	2
22	8,6	6,19	2
21	1,4	6,21	2
33	5,6	4,19	1
32	8,2	5,74	0
35	8,7	5,1	0
35	7,9	4,51	1
22	0,4	5,47	2
36	9,1	5,86	2
35	10,5	6,1	2
34	5,3	3,59	1
34	9,2	5,81	0
35	7,7	4,8	1
34	5,6	4,1	1
34	9,6	3,67	2
34	6,1	5,19	2
32	5,8	5,1	1
22	6,7	3,94	1
22	4,9	4,88	1
33	1,6	1,14	0
22	4,1	5,31	1
22	0,4	6,88	2
22	7,9	7,22	2
34	3,6	4,46	2
33	0,2	4,38	1
36	7,7	6,1	0
33	9	5,67	0
35	5,5	5,14	0
35	7,9	5,55	0
35	8,4	5,28	2



Lampiran 2 – Program Naïve Bayes

```
import pandas as pd
dataset = pd.read_csv('Dataset 2.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print (accuracies.mean()*100)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
from sklearn.metrics import classification_report
cr = classification_report(y_test, y_pred)
print(cr)
```



Lampiran 3 – Program K-Nearest Neighbors

```
import pandas as pd
dataset = pd.read_csv('Dataset 2.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 6, metric = 'manhattan')
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print (accuracies.mean()*100)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
from sklearn.metrics import classification_report
cr = classification_report(y_test, y_pred)
print(cr)
```

Lampiran 4 – Confusion Matrix

- NB skenario 1:

$$\begin{bmatrix} 13 & 5 & 0 \\ 2 & 14 & 1 \\ 5 & 0 & 6 \end{bmatrix}$$

- NB skenario 2:

$$\begin{bmatrix} 11 & 5 & 0 \\ 1 & 12 & 1 \\ 2 & 1 & 4 \end{bmatrix}$$

- NB skenario 3:

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 7 & 1 \\ 3 & 0 & 2 \end{bmatrix}$$

- KNN skenario 1:

$$\begin{bmatrix} 13 & 4 & 1 \\ 1 & 16 & 0 \\ 1 & 1 & 9 \end{bmatrix}$$

- KNN skenario 2:

$$\begin{bmatrix} 9 & 5 & 2 \\ 0 & 14 & 0 \\ 1 & 1 & 5 \end{bmatrix}$$

- KNN skenario 3:

$$\begin{bmatrix} 2 & 1 & 2 \\ 0 & 9 & 0 \\ 2 & 0 & 3 \end{bmatrix}$$



Lampiran 5 – Prediksi dan Data Uji

Naïve Bayes						KNN					
Skenario 1		Skenario 2		Skenario 3		Skenario 1		Skenario 2		Skenario 3	
Data Uji	Prediksi	Data Uji	Prediksi	Data Uji	Prediksi	Data Uji	Prediksi	Data Uji	Prediksi	Data Uji	Prediksi
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2
1	2	1	2	1	2	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0
2	0	2	1	2	0	2	1	2	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
2	0	2	0	2	0	2	2	2	1	2	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0
2	0	2	0	2	0	2	0	2	0	2	0
2	2	2	2			2	2	2	2		
0	0	0	0			0	0	0	0		
2	2	2	2			2	2	2	2		
0	0	0	0			0	0	0	0		
0	0	0	0			0	0	0	0		
0	1	0	1			0	1	0	1		
0	1	0	1			0	0	0	0		
0	0	0	0			0	1	0	1		
0	1	0	1			0	1	0	1		
1	1	1	1			1	1	1	1		
1	1	1	1			1	1	1	1		
0	0	0	0			0	0	0	0		
0	0	0	0			0	0	0	0		
0	1	0	1			0	1	0	1		
0	0	0	0			0	0	0	0		
1	1	1	1			1	1	1	1		
1	0	1	0			1	1	1	1		
1	1	1	1			1	1	1	1		
1	1					1	1				
0	0					0	0				
0	0					0	0				
1	1					1	1				
1	0					1	1				
2	2					2	2				
2	2					2	2				
2	0					2	2				
2	0					2	2				