MODEL OPTIMASI *KNN*-PSORF DALAM MENANGANI *HIGH DIMENSIONAL* DATA BANJIR KOTA SAMARINDA

SKRIPSI

Diajukan Oleh:
Anggiq Karisma Aji Restu
2011102441089

PROGRAM STUDI S1TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
JULI 2024

MODEL OPTIMASI KNN-PSORF DALAM MENANGANI *HIGH DIMENSIONAL* DATA BANJIR KOTA SAMARINDA

SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar
Sarjana Komputer Fakultas Sains dan Teknologi
Universitas Muhammadiyah Kalimantan Timur

Diajukan Oleh:
Anggiq Karisma Aji Restu
2011102441089

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UIMVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
JULI 2024

LEMBAR PERSETUJUAN

Model Optimasi KNN-PSORF Dalam Menangani *High Dimensional* Data Banjir Kota Samarinda

SKRIPSI

Diajukan oleh:

Anggiq Karisma Aji Restu
2011102441089

Disetujui untuk diujikan
Pada tanggal 28 Juni 2024

Pembimbing

29/6/2024

Taghfirul Azhima Yoga Siswa, S.Kom, M.Kom
NIDN. 1118038805

Mengetahui,
Koordinator Skripsi

Abdul Rahim, S.Kom., M.Cs.
NIDN. 1115039601

LEMBAR PENGESAHAN

Model Optimasi KNN-PSORF Dalam Menangani *High Dimensional* Data Banjir Kota Samarinda

SKRIPSI

Diajukan oleh:

Anggiq Karisma Aji Restu

2011102441089

Diseminarkan dan Diujikan
Pada Tanggal ..3..Juli 2024

Penguji I	Penguji II
Wawan Joko Pranoto, S.Kom, M.Ti NIDN. 1102057701	Taghfirul Azhima Yoga Siswa, S.Kom, M.Kom NIDN. 1118038805

Mengetahui,
Ketua
Program Studi Teknik Informatika

Arbansyah, S.Kom., M. TI
NIDN.1118019203

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama	: Anggiq Karisma Aji Restu
NIM	: 2011102441089
Program Studi	: Teknik Informatika
Judul Penelitian	: Model Optimasi KNN-PSORF Dalam Menangani High Dimensional Data Banjir Kota Samarinda

Menyatakan bahwa Skripsi yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung resiko atau sanksi yang dijatuhkan kepada saya apa bila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam skripsi saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Samarinda, 14 juli 2024

Yang membuatan pernyataan

Anggiq Karisma Aji Restu

NIM: 2011102441127

ABSTRAK

Banjir adalah fenomena alam yang sering terjadi di Indonesia, termasuk di Kota Samarinda yang mengalami masalah banjir dalam tiga tahun terakhir dengan dampak ribuan rumah sebanyak 27.000 jiwa terkena banjir. Untuk memprediksi bencana banjir dibutuhkan teknologi machine learning menggunakan metode klasifikasi data mining. Namun, pada proses klasifikasi seringkali terjadi permasalahan yang berkaitan dengan data berdimensi tinggi ini dapat menyebabkan overfitting dan ketidakseimbangan kelas yang menyebabkan bias pada kelas yang dominan dengan mengabaikan kelas minoritas. Penelitian ini bertujuan untuk meningkatkan nilai akurasi klasifiikasi pada data banjir Kota Samarinda menggunakan algoritma K-Nearest Neighbor (KNN) yang dikombinasikan seleksi fitur relief dan optimasi Particle Swarm Optimization (PSO). Metode validasi yang digunakan adalah 10-fold cross-validation, sementara evaluasi kinerja model dilakukan menggunakan confusion matrix. Data yang digunakan diperoleh dari BPBD dan BMKG Kota Samarinda pada rentang tahun 2021-2023, dengan 19 fitur dan total 1095 record. Hasil seleksi fitur Relief didapatkan empat fitur penting, yaitu arah angin maksimum, kecepatan angin, kecepatan angin rata-rata, dan arah angin maksimum. Evaluasi rata-rata dengan nilai k=3, k=5, k=7, k=11, k=13, dan k=15 menunjukkan penerapan seleksi fitur Relief dan optimasi PSO, efektif dalam meningkatkan akurasi pada algoritma k-Nearest Neighbor pada data banjir dengan hasil akurasi KNN dan PSO memberikan peningkatan sebesar 2-5%, KNN dengan seleksi fitur Relief memberikan peningkatan sebesar 1-2% dan KNN dengan kombinasi Relief dan PSO memberikan peningkatan sebesar 2-5%. Kombinasi model KNN, Relief, PSO diharapkan dapat memberikan peforma yang optimal dalam klasifikasi data banjir Kota Samarinda.

Kata kunci: K-Nearest Neighbor, Relief, Banjir, 10-Fold Cross-Validation, klasifikasi

ABSTRACT

Floods are a natural phenomenon that frequently occurs in Indonesia, including in Samarinda City which has faced flood issues over the past three years, affecting thousands of homes and around 27,000 residents. Predicting flood disasters requires machine learning technology using data mining classification methods. However, classification processes often encounter issues related to high-dimensional data, which can lead to overfitting and class imbalance, thereby biasing dominant classes while neglecting minority classes. This research aims to enhance classification accuracy in Samarinda City's flood data using the K-Nearest Neighbor (KNN) algorithm combined with Relief feature selection and Particle Swarm Optimization (PSO) optimization. The validation method employed is 10-fold cross-validation, with performance evaluation using a confusion matrix. Data sourced from Samarinda City's Disaster Management Agency (BPBD) and Meteorology, Climatology, and Geophysics Agency (BMKG) spans from 2021 to 2023, comprising 19 features and a total of 1095 records. Relief feature selection identified four crucial features: maximum wind direction, wind speed, average wind speed, and maximum wind speed direction. Average evaluations with k values of 3, 5, 7, 11, 13, and 15 demonstrate that Relief feature selection and PSO optimization effectively enhance accuracy in the K-Nearest Neighbor algorithm for flood data, with KNN and PSO yielding improvements of 2-5%. Relief feature selection alone improves accuracy by 1-2%, while combining Relief with PSO provides a 2-5% enhancement. The combined KNN, Relief, PSO model is expected to deliver optimal performance in classifying Samarinda City's flood data.

Keywords: K-Nearest Neighbor, Relief, Flood, 10-Fold Cross-Validation, classification

PRAKATA

Dengan nama Allah Yang Maha Pengasih dan Penyayang, segala puji hanya bagi-Nya. Shalawat serta salam semoga tercurahkan kepada Rasulullah Muhammad SAW, yang telah membawa petunjuk serta rahmat bagi seluruh alam. Prakata ini dibuat sebagai ungkapan terima kasih dan penghargaan yang tulus dari penulis kepada semua pihak dalam penyelesaian skripsi ini, sehingga Peniliti dapat menyelesaikan skripsi ini dengan judul "Model Optimasi KNN-PSORF Dalam Menangani High Dimensional Data Banjir Kota Samarinda". Kami ingin mengucapkan terima kasih kepada banyak pihak yang telah membantu dan mendukung penyelesaian penelitian ini. Terutama kepada:

1. Bapak Agus Triyono, Ibu Natlisa Dwi Kristiani, dan Adik tercinta yang selalu memberikan doa serta dukungan kepada penulis.
2. Bapak Taghfirul Azhima Yoga Siswa, S.Kom, M.Kom Selaku Dosen Pembimbing, yang telah memberikan bimbingan, arahan, dan pengarahan yang diberikan selama proses studi akhir ini.
3. Bapak Wawan Joko Pranoto, S.Kom, M.Ti selaku Dosen Penguji seminar proposal penelitian dan sidang skripsi penulis.
4. Bapak Arbansyah, S.Kom., M.TI selaku ketua Program Studi S1 Teknik Informatika.
5. Prof. Ir. Sarjito, M.T., Ph.D., selaku Dekan Fakultas Sains & Teknologi Universitas Muhammadiyah Kalimantan Timur
6. Dr. Muhammad Musiyam, M.T selaku Rektor Universitas Muhammadiyah Kalimantan Timur.
7. Seluruh Bapak dan Ibu Dosen Program Studi Teknik Informatika Universitas Muhammadiyah Kalimantan Timur dan staff yang penulis banggakan dan hormati.
8. Winda Wahyuni yang selalu memberikan semangat, dukungan dan masukan dalam penyusunan laporan dari awal hingga penyelesaian studi akhir.
9. Teman-teman yang selalu mengganti nama grub yang beranggotakan 12 orang yang menjadi seperti keluarga sendiri dan tidak lupa selalu memberikan semangat dan dukungan selama menjalani proses perkuliahan, semoga kalian sukses semua saudara.

Kami sadar bahwa penelitian ini masih jauh dari kesempurnaan, oleh karena itu segala saran dan kritik membangun sangat kami harapkan guna perbaikan di masa yang akan datang.

Samarinda, 29 Juni 2024
Yang membuatan pernyataan

Anggiq Karisma Aji Restu
NIM: 2011102441089

DAFTAR ISI

DAFTAR TABEL

Tabel **Halaman**

DAFTAR GAMBAR

Gambar **Halaman**

DAFTAR LAMPIRAN

Lampiran **Halaman**

BAB I
PENDAHULUAN

1.1. Latar Belakang

Banjir adalah fenomena alam yang sering melanda indonesia. Menurut Data Informasi Bencana Indonesia (DIBI), dalam kurun waktu tiga tahun terakhir, tercatat sebanyak 4580 kejadian banjir di Indonesia dan Jumlah tertinggi terjadi pada tahun 2020, mencapai 1531 kejadian, menjadi yang terbanyak dalam hampir satu dekade terakhir (Databoks, 2023). Penyebab banjir disebab kan oleh aliran air dan curah hujan yang tinggi di suatu daerah, namun banjir juga dapat terjadi karena kondisi lingkungan seperti hilangnya lahan terbuka hijau. (Dilla Evitasari et al., 2023).

Kota Samarinda merupakan ibu kota dari Provinsi Kalimantan Timur yang saat ini sedang dilanda permasalahan banjir yang cukup parah. Banjir yang sering terjadi akhir-akhir ini sangat mengganggu aktivitas warga. Sebagian besar wilayah kota Samarinda yang bermasalah dengan banjir berlokasi di DAS Karangmumus (320 km2). Selain itu terdapat dua sub sistem lain yang juga mempunyai masalah banjir yaitu DAS Karang Asam Besar (9,65 km2) dan DAS Karang Asam Kecil (16,25 km2). (Purwanto, 2020). Pada tahun 2020 banjir terjadi pada 10 kecamatan, 4 kelurahan menyebabkan sebanyak 27.000 jiwa terkena dampak banjir yang merugikan masyarakat (Ernawati et al., 2021).

Klasifikasi banjir berdasarkan penyebabnya dapat membantu memperbaiki perkiraan frekuensi banjir, mendukung deteksi serta penafsiran untuk perubahan kejadian dan tingkat keparahan banjir (Tarasova et al., 2019). Oleh karena itu, perlu diadakan evaluasi perbaikan akurasinya dengan metode klasifikasi data *mining*. *Data mining* merupakan proses yang dilakukan dengan penggabungan teknik analisis data untuk memperoleh pola penting pada suatu data (Tarigan et al., 2022). Penerapan teknik *data mining* memiliki relevansi yang luas, termasuk dalam konteks bencana alam seperti banjir. Penggunaan *data mining* memegang peranan penting dalam menghubungkan teknologi dan penelitian, serta dapat mengenali pola asosiasi, melakukan klasifikasi, dan berinteraksi dengan algoritma pengklasifikasi untuk mendapatkan hasil yang bervariasi dari hasil yang buruk hingga mendapatkan hasil yang baik (Mian & Ghabban, 2022).

Ketidakseimbangan kelas *(class imbalance)* terjadi ketika sebagian besar data condong pada satu label kelas. Hal ini dapat terjadi dalam kedua klasifikasi kelas dua dan multi-kelas. Algoritma pembelajaran mesin menganggap bahwa data didistribusikan secara rata, jadi ketika ada kelas yang tidak seimbang, mesin akan lebih bias pada kelas yang dominan dengan mengabaikan kelas minoritas, sehingga pada kelas mayoritas lebih cenderung menunjukkan nilai akurasi yang lebih baik. Hal ini disebabkan oleh fakta bahwa fungsi pembelajaran mesin secara konsisten berusaha untuk mengoptimasi kuantitas, seperti tingkat *error,* tanpa mempertimbangkan distribusi data (Yoga Siswa, 2023).

Data berdimensi tinggi atau *High Dimensional* merupakan data yang mempunyai banyak atribut yang dapat digunakan dalam proses analisis. Misalnya, mempunyai puluhan bahkan ratusan atribut, maka data tersebut dapat diklasifikasikan sebagai data berdimensi tinggi (Hakimah et al., 2022). Data berdimensi tinggi dalam kumpulan data menyebabkan beberapa masalah dalam *Machine Learning*. Pertama, sulit bagi model pembelajaran untuk mencapai performa optimal karena semakin banyak fitur yang digunakan, semakin sulit bagi model pembelajaran mesin untuk memodelkan masalah tersebut. Kedua, jumlah data yang besar ini dapat menyebabkan *overfitting* karena banyaknya konfigurasi karakteristiknya meskipun data yang kita miliki sedikit. Ketiga, data dengan dimensi yang besar susah

untuk diproses secara komputasi (*computationally expensive*) baik dari segi memori maupun waktu (Ariyoga, 2022).

Algoritma *k-nearest neighbour* merupakan metode klasifikasi yang mengelompokkan data uji menjadi data latih berdasarkan jarak antara beberapa tetangga *(neighbor)* terdekat dari data uji tersebut, Algoritma *k-nearest neighbour* juga sering digunakan dalam penyelesaian data *mining* dalam klasifikasi (Sitepu & Manohar, 2022). Dari beberapa penelitian yang pernah dilakukan sebelumnya dalam klasifikasi data banjir, Algoritma *k-nearest neighbour* (KNN) tanpa seleksi fitur dinilai memiliki performa lebih unggul dalam klasifikasi data banjir dengan akurasi 94,91% dibandingkan dengan *Random Forest* 71,3 %, *Support Vector Machine* 52,71%*, Naive Bayes* 89,23%. (Gauhar et al., 2021; Hossain & Zeyad, 2023; Dilla Evitasari et al., 2023; Vafakhah et al., 2020; Daniel et al., 2023; Farhan & Setiaji, 2023).

Berdasarkan penelitian lain yang menerapkan algoritma KNN pada data banjir dengan dimensi data yang tinggi menunjukkan akurasi yang lebih rendah yaitu 88, 94%, Ditemukan adanya permasalahan pada penelitian dengan data *High Dimensional* yang dapat menurunkan akurasi(Cumel, David Zamri, Rahmaddeni, 2022). Kemudian, pada studi klasifikasi kesesuaian air, dimana algoritma KNN juga memiliki akurasi yang rendah, dan rendahnya akurasi KNN disebabkan karena bertemu dengan data berdimensi tinggi atau *high dimension* (Sopiatul Ulum et al., 2023). Pada dataset Banjir yang akan digunakan pada penelitian ini terdapat 19 fitur, dimana untuk jumlah fitur yang tinggi seringkali merujuk pada data berdimensi tinggi. Oleh karena itu, untuk mengatasi masalah tersebut dilakukan *feature selection* untuk mengidentifikasi fitur-fitur yang paling relevan, dengan tujuan untuk meningkatkan performa yang lebih baik dengan menggunakan *feature selection.*

Pendekatan yang digunakan dari penelitian sebelumnya dalam mengatasi dimensi tinggi menggunakan *feature selection Relief* terbukti bisa memberikan peningkatan akurasi sebesar 5-10%. Penelitian yang dilakukan memberikan peningkatan klasifikasi *Naive bayes* dan KNN dimana sebelum penerapan *feature selection* akurasi yang diperoleh sebesar 73,4% untuk *Naive bayes* dan 66,24% untuk KNN. Kemudian, setelah dilakukan penerapan *feature selection Relief* didapatkan peningkatan akurasi dimana akurasi *Naive bayes* menjadi 74,38% dan KNN menjadi 72,22% (Yahdin et al., 2021). Terdapat peningkatan yang signifikan pada akurasi algoritma KNN, yang sebelumnya memiliki akurasi sebesar 85,31% dan setelah penerapan *feature selection Relief* meningkat menjadi 95,63% (Yusra et al., 2021). Kemudian akurasi di atas 90% diperoleh setelah *feature selection Relief* dikombinasikan dengan KNN pada penelitian yang dilakukan oleh (Kemal Musthafa Rajabi et al., 2023; Abdulrazaq et al., 2021).

Kemudian dalam menghadapi ketidakseimbangan kelas pada dataset banjir dimana pada data yang diperoleh dari BMKG dan BPBD yang terjadi banjir berjumlah 49 data sedangkan yang tidak banjir berjumlah 841 data, maka pada penelitian ini juga akan menggunakan teknik *Synthetic Minority Over-samplingTechnique* (SMOTE) agar performa model yang dihasilkan dapat optimal. Berdasarkan penelitian sebelumnya, teknik SMOTE pernah digunakan dalam menangani ketidakseimbangan kelas pada dataset banjir dan dianggap dapat memberikan peningkatan akurasi terhadap model klasifikasi sebesar 0.21-10% (Nawi et al., 2020; Priscillia et al., 2022; Nursyahfitri et al., 2022; Razali et al., 2020; Dwi Astuti & Nova Lenti, 2021).

Berdasarkan beberapa penelitian sebelumnya yang pernah dilakukan, Optimasi *Particle Swarm Optimization* (PSO) pernah digunakan untuk mengklasifikasikan data banjir. penerapan PSO) menghasilkan hasil yang signifikan dalam memberikan peningkatan kinerja (optimasi) pada algoritma *Naïve Bayes* dan *K-Nearest Neighbor* (Yoga & Prihandoko, 2018). Penelitian lain yang menggunakan Optimasi PSO juga dapat mengoptimalkan peforma, dimana penerapan Optimasi tersebut dapat memberikan peningkatkan akurasi sebesar 3-11% (Dwiasnati & Yudo Devianto, 2022; Faldi et

al., 2023;Arora et al., 2021). sehingga PSO akan di gunakan sebagai metode optimasi dalam penelitian pada data banjir.

Data tersebut setelah dilakukan Pengolahan data, terdapat permasalahan imbalanced data dan Data High Dimensional. Peneliti berencana akan menggunakan algoritma KNN dimana dari penelitan sebelumnya, knn mampu memberikan hasil akurasi yang baik dalam mengklasifkasi berbagai macam data, terutama ketika dikombinasikan dengan teknik pemilihan fitur dan penanganan ketidakseimbangan kelas. Namun, sebagian besar penelitian sebelumnya menggunakan data dengan dimensi yang berbeda dan teknik pemilihan fitur yang beragam. selain dilakukan pengolahan data, dilakukan juga sebuah analisa melalui penelitian sebelumnya, bahwa belum ada penelitian yang sama dengan menggunakan Kombinasi model KNN-PSO-*Relief* (KNN-PSORF) dan teknik Oversamping SMOTE dalam menangani *High Dimensional* dan *Imbalaced Data.* Diharapakan *Kombinasi model KNN, PSO, Relief, SMOTE* dapat memberikan peforma optimal dalam klasifikasi data banjir Kota Samarinda.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah untuk penelitian ini dapat dirumuskan sebagai berikut:

1. Fitur Apa saja yang memiliki pengaruh penting Pada algoritma *k-nearest neighbors* (KNN) dengan menggunakan optimasi PSO, seleksi fitur *Relief* dan *oversampling* SMOTE dalam meningkatkan akurasi pada dataset banjir Kota Samarinda?
2. Seberapa besar peningkatan akurasi yang didapat Algoritma *k-nearest neighbors* (KNN) dalam Klasifikasi data banjir Kota Samarinda dengan menggunakan PSO sebagai optimasi, seleksi fitur *Relief* dalam menangani high dimensional dan *oversampling* SMOTE dalam menangani *imbalanced data* ?

1.3. Tujuan Penelitian

Berdasarkan latar belakang masalah yang telah diuraikan, tujuan utama dari penelitian ini adalah:

1. Menentukan atribut yang berpengaruh pada algortima *k-nearest neighbour* (KNN) terhadap dataset banjir Kota Samarinda.
2. Mengevaluasi hasil kinerja algoritma *k-nearest neighbors* (KNN) yang dievaluasi menggunakan metode *validasi cross-fold k-fold* dan *matrix confusion.*

1.4. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Penulis:
   a. Penelitian ini akan memberikan pengalaman dan pengetahuan praktis dalam mengembangkan metode klasifikasi pada data banjir dengan mengatasi masalah akurasi pada data berdimensi tinggi menggunakan algoritma KNN.
   b. Penelitian ini memberikan pengalaman dalam menerapkan teknik seleksi fitur dan optimasi pada algoritma klasifikasi untuk menghasilkan hasil penelitian yang lebih baik.
2. Peniliti selanjutnya:
   a. Hasil Penelitian ini diharapkan bisa memberikan kontribusi pada pengembangan ilmu pengetahuan, khususnya dalam bidang teknik data mining dan pengolahan data berdimensi tinggi, serta memberikan wawasan baru dalam penggunaan algoritma KNN dalam konteks klasifikasi banjir.
   b. Diharapkan bisa bermanfaat dan menjadi referensi bagi penelitian selanjutnya dalam bidang yang sama atau terkait, serta menjadi landasan untuk pengembangan penelitian lebih lanjut dalam upaya meningkatkan prediksi dan pengendalian banjir di daerah-daerah lain.

1.5. Batasan Masalah

Agar ruang lingkup permasalahan yang dibuat tidak meluas, maka peneliti membatasi penelitian sebagai berikut :

a. Data yang digunakan dalam penelitian ini adalah dataset banjir Kota Samarinda yang diperoleh dari BPBD (Badan Penanggulangan Bencana Daerah) Kota samarinda pada tahun 2021-2023.
b. Algoritma klasifikasi yang akan digunakan dalam penelitian kali ini adalah *K-Nearest Neighbor (KNN*) dengan tambahan metode seleksi fiturnya berupa *Relief*, metode *optimasi Particle Swarm Optimization (PSO)*.
c. Fitur-fitur yang digunakan dalam klasifikasi pada dataset banjir dalam penelitian ini meliputi (i) Temperatur-minimum, (ii) Temperatur-maksimum, (iii) Temperatur, (iv) Kelembaban, (v) Curah-hujan, (vi) Lamanya-penyinaran-matahari, (vii) Kecepatan-angin, (viii) Arah-angin-maksimum, (ix) Kecepatan-angin-rata-rata, (x) Arah-angin-terbanyak dan (xi) Terjadi-banjir yang akan dijadikan kelas atau target dari klasifikasi.

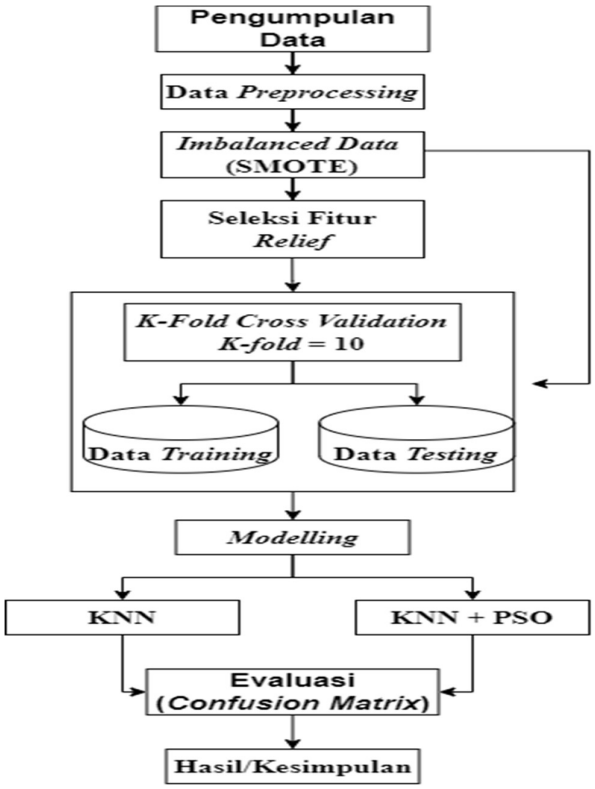
BAB II
METODOLOGI PENELITIAN

2.1. Objek Penelitian

Objek penelitian ini menggunakan data dari BPBD (Badan Penanggulangan Bencana Daerah) dan BMKG (Badan Meteorologi, Klimatologi, dan Geofisika) Kota Samarinda, Dataset ini berisikan data dari tahun 2021 hingga 2023. Adapaun lokasi Penelitian yang dilakukan pada BPBD (Badan Penanggulangan Bencana Daerah) Kota Samarinda yang beralamatkan di JL.Sentosa Dalam No.01, Sungai Pinang Dalam, Kecamatan Sungai Pinang, Kota Samarinda, Kalimantan Timur 75242 dan BMKG (Badan Meteorologi, Klimatologi, dan Geofisika) Kota Samarinda beralamatkan Jl. Pipit No.150, Bandara, Kec. Sungai Pinang, Kota Samarinda, Kalimantan Timur 75117.

2.2. Prosedur Penelitian

Setiap penelitian memilki beberapa langkah dalam tahap pelaksanaan penelitian, Dalam penelitian ini terdapat langkah-langkah untuk mencapai tujuan penelitian. Tahapan penelitian yang dilakukan dimulai dari pengumpulan dan analisis data hingga tahap terakhir yaitu evaluasi hasil. Berikut adalah bagan langkah-langkah yang akan dilakukan :



Gambar 2. 1 *Diagram* Alur Penelitian

2.2.1. Identifikasi Masalah

Identifikasi masalah dalam penelitian merupakan tahap penting yang memungkinkan peneliti untuk menetapkan fokus dan arah penelitian dengan tepat. Permasalahan utama dalam penelitian ini berkaitan dengan penentuan metode optimal untuk mengklasifikasikan data banjir di Kota Samarinda. Selain itu, dilakukan juga tinjauan literatur untuk mengidentifikasi kekosongan penelitian yang terkait dengan klasifikasi data banjir, yang dapat memberikan pandangan mendalam dan bahan yang relevan untuk penelitian ini.

2.2.2. Pengumpulan Data

Tahapan awal pada penelitian ini adalah melakukan pengumpulan data lalu dilanjutkan dengan proses data *preparation*, Penelitian ini menggunakan data Banjir Kota Samarinda dari tahun 2021-2023. Data yang didapatkan dari BPBD (Badan Penanggulangan Bencana Daerah) dan BMKG (Badan Meteorologi, Klimatologi, dan Geofisika) Kota Samarinda, menunjukkan bahwa data tersebut terdiri atas 1095 Dataset dengan 19 Atribut dan 1 Atribut sebagai label didalamnya. Pada dataset Banjir di Kota Samarinda dibentuk tabel yang dapat mempermudah dalam mengetahui berapa banyak fitur yang ada sebagai berikut :

Tabel 2. 1 Fitur *Dataset* Banjir BMKG dan BPBD Kota Samarinda

No	Fitur	Keterangan
1.	Tanggal	Waktu Kejadian
2.	(Tn)	Temperatur-minimum (°C)
3.	(Tx)	Temperatur-maksimum (°C)
4.	(Tavg)	Temperatur-rata-rata (°C)
5.	(RH_avg)	Kelembaban-rata-rata (%)
6.	(RR)	Curah-hujan (mm)
7.	(ss)	Lamanya-penyinaran-matahari (jam)
8.	(ff_x)	Kecepatan-angin-maksimum (m/s)
9	(ff_avg)	Kecepatan-angin-rata-rata (m/s)
10.	(ddd_x)	Arah-angin-maksimum (°)
11.	(ddd_car)	Arah-angin-terbanyak (°)
12.	Jam Kejadian	Jam Terjadinya Bencana
13.	Lokasi Wilayah	Wilayah Terjadinya Bencana
14.	Luas Area M2	Luas area yang terdampak
15.	Objek Terkena Bencana	Fasilitas Yang terdampak bencana
16.	Korban	Jumlah korban Yang terdampak bencana
17.	Kerugian	Nominal Kerugian yang dialami
18.	Keterangan	Detail kejadian bencana
19.	Terjadi Banjir	Ya/Tidak(class)

2.2.3. Data Pre-Procesing

Data yang didapat dari BPBD (Badan Penanggulangan Bencana Daerah) dan BMKG (Badan Meteorologi, Klimatologi, dan Geofisika) harus diolah lebih lanjut sebelum masuk ke tahap pemodelan untuk menghindari pengolahan data yang tidak diperlukan. Tahapan pemrosesan data ini meliputi integrasi data, seleksi, transformasi data, pembersihan data, dan penyeimbangan data sebagai berikut :

a. Data *Integration*

Pada *Data Integration* menggabungkan data dari berbagai sumber yang berbeda menjadi satu set data. Data yang akan digabungkan bersumber dari data Badan Penanggulangan Bencana Daerah (BPBD) dan Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Data ini mencakup tentang data banjir dan faktor pendukung penyebab banjir Kota Samarinda dari tahun 2021-2023. Proses integrasi data bertujuan untuk memberikan informasi yang lebih lengkap dan akurat terkait data banjir.

b. Data *Selection*

Data selection dilakukan untuk dilakukan pemilihan dan pengambilan data banjir Kota Samarinda yang diperoleh dari BPBD (Badan Penanggulangan Bencana Daerah) dan BMKG (Badan Meteorologi, Klimatologi, dan Geofisika), proses *data selection* akan difokuskan pada pemilihan data yang relevan dan penting untuk analisis banjir. kemudian untuk atribut yang dianggap kurang relevan maka akan dihapus. Tabel 2.1 merupakan data awal yang diperoleh dari BPBD (Badan Penanggulangan Bencana

Daerah) dan BMKG (Badan Meteorologi Klimatologi dan Geofisika) Kota Samarinda, yang terdiri dari 19 kolom. Setelah dilakukan analisis, sebanyak 7 kolom dianggap kurang relevan dan tidak digunakan dalam proses prediksi banjir. Oleh karena itu, dari 19 kolom awal, hanya 11 kolom yang dipilih sebagai fitur atau atribut, sedangkan 1 kolom lainnya dipilih sebagai target atau kelas seperti yang ada pada dalam tabel 2.2.

Tabel 2. 2 *Data Selection*

No	Atribut Awal	Atribut Hasil Seleksi	Keterangan
1	Tgl	Tanggal	Date
2	Tn	Temperatur-minimum	Atribut
3	Tx	Temperatur-maksimum	Atribut
4	Tavg	Temperatur rata-rata	Atribut
5	RH_avg	Kelembaban	Atribut
6	RR	Curah-hujan	Atribut
7	ss	Lamanya-penyinaran-matahari	Atribut
8	ff_x	Kecepatan-angin	Atribut
9	ddd_x	Arah-angin-maksimum	Atribut
10	ff_avg	Kecepatan-angin-rata-rata	Atribut
11	ddd_car	Arah-angin-terbanyak	Atribut
12	Terjadi Banjir	Terjadi-Banjir	Class/target

c. *Data Cleaning*

Data pada data banjir dari BPBD dan BMKG tahun 2021-2023 akan dilakukan serangkaian langkah untuk membersihkan dan mempersiapkan data agar siap digunakan dalam analisis atau pemodelan lebih lanjut. Data *cleaning* sendiri merupakan sebuah proses untuk memperbaiki atau membersihkan data yang tidak tepat, tidak lengkap, atau tidak konsisten. Pada tahap ini, dilakukan pembersihan data dengan menghapus entri yang memiliki nilai #N/A (tidak tersedia) atau data yang kosong, serta mengidentifikasi dan menghapus data yang duplikat.

```
#Penanganan data yang kosong
#Mengecek data kosong
kosong = df.isna().sum().sum()
print(df.isna().sum())
print("Jumlah data kosong: ", kosong)

#Menghitung perbandingan jumlah data sebelum dan sesudah menghapus data kosong
data_kotor = len(df)
data_bersih = df.dropna()

print(f"Jumlah data sebelum pembersihan data kosong: {data_kotor}")
print(f"Jumlah data setelah pembersihan data kosong: {len(data_bersih)}")
```

Gambar 2. 2 Proses *Data Cleaning*

Berikut adalah tabel 2.3 yang berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 3 *Parameter Data Cleaning*

Parameter	Keterangan
data.isna()	Fungsi ini digunakan untuk melakukan pengecekan, apakah terdapat nilai yang hilang (NaN) pada data.
len()	Fungsi yang digunakan untuk menghitung jumlah elemen dalam suatu objek/data.
dropna()	Fungsi ini digunakan untuk menghilangkan atau menghapus baris atau kolom yang mengandung nilai yang hilang (NaN).

```
# Menghitung jumlah nilai yang hilang di setiap kolom setelah pembersihan
missing_values_after = data_bersih.isnull().sum()
print("Jumlah nilai yang hilang setelah pembersihan:")
print(missing_values_after)
```

Gambar 2. 3 Pemeriksaan nilai yang hilang setelah penghapusan

Pada tabel 2.4 berikut berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 4 Pemeriksaan nilai yang hilang setelah penghapusan

Parameter	Keterangan
Missing_values_after	Digunakan untuk melakukan tindakan lanjutan setelah tahap penanganan nilai yang hilang selesai.
data.isnull()	Digunakan untuk mengidentifikasi nilai yang hilang dalam sebuah DataFrame atau struktur data lainnya.
sum	Digunakan untuk menghitung jumlah nilai-nilai dalam DataFrame atau Series.

d. Data*Transformation*

Berikut pada Data transformation dilakukan proses pengubahan atau pemrosesan data dari satu bentuk atau format ke bentuk atau format lain yang lebih sesuai atau berguna untuk analisis, pemodelan, atau aplikasi tertentu. Label *encoder* mengacu pada pengubahan nilai label ke dalam bentuk *numeric,* yang ditujukan untuk memudahkan mesin untuk dapat membaca data. Dengan label *encoder*, algoritma *machine learning* dapat dengan dalam mengambil keputusan terbaik dalam mengoperasikan data(Yoga Siswa, 2023). Tujuan lain dari data *transformation* adalah untuk menghasilkan data yang lebih mudah dimengerti, dan sesuai dengan kebutuhan penelitian yang akan dilakukan.

```
#Transformasi data
print(f"===Sebelum Transformasi Data:=== \n{data_bersih[['Arah-angin-terbanyak', 'terjadi-banjir']].head()} \n")#Proses Encoding
```

Gambar 2. 4 Sebelum *Transformasi Data*

Pada Gambar 2.5 merupakan Proses *Encoding*

```
#Transformasi data
print(f"===Sebelum Transformasi Data:=== \n{data_bersih[['Arah-angin-terbanyak', 'terjadi-banjir']].head()} \n")#Proses Encoding
ordinal = OrdinalEncoder()
labelEncoder = LabelEncoder()
data_transform = data_bersih['Arah-angin-terbanyak']

data_bersih['Arah-angin-terbanyak'] = labelEncoder.fit_transform(data_bersih['Arah-angin-terbanyak'])
data_bersih['terjadi-banjir'] = data_bersih['terjadi-banjir'].replace({'banjir':1, 'tidak banjir':0})

print(f"\n===Setelah Transformasi Data:=== \n{data_bersih[['Arah-angin-terbanyak', 'terjadi-banjir']].head()}")
```

Gambar 2. 5 Proses *Encoding*

Berikut adalah tabel 2.5 yang berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 5 *Parameter Data Transformation*

Parameter	Keterangan
OrdinalEncoder	Berfungsi mengubah fitur kategorikal menjadi representasi numerik
LabelEncoder()	Berguna untuk melakukan pengkodean label pada data.
Data_transform	Berfungsi menyimpan fitur atau atribut dari dataset yang diubah menjadi representasi *numerik.*
Fit_transfrom	menyesuaikan *encoder* dengan data dan langsung mengubah data menjadi nilai numerik.
Replace	metode Pandas yang digunakan untuk menggantikan nilai spesifik dalam DataFrame.
Head	digunakan untuk menampilkan beberapa baris pertama dari DataFrame tersebut

e. Data *Balancing*

Data *balancing* merupakan proses yang dilakukan untuk menyeimbangkan distribusi kelas atau label pada dataset. Hal ini sering kali diperlukan dalam konteks masalah klasifikasi di mana terdapat ketidakseimbangan yang signifikan antara jumlah sampel yang termasuk dalam setiap kelas atau label. Pada penelitian ini menggunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE) dalam menyeimbangkan data yang tidak seimbang (*imbalance data*). Metode SMOTE digunakan untuk menghasilkan sampel sintetis dari kelas minoritas, sehingga meningkatkan representasi kelas minoritas dalam dataset.

```
#Memisahkan variabel atribut dan target
X = data_bersih.drop(['Tanggal', 'terjadi-banjir'], axis=1)
y = data_bersih['terjadi-banjir']

#Jumlah data pada variable kelas sebelum diterapkan oversampling SMOTE
before = y.value_counts()

#Penerapan oversampling SMOTE
smote = SMOTE(random_state=42)
X_res, y_res = smote.fit_resample(X, y)

#Jumlah data pada variable kelas setelah diterapkan oversampling SMOTE
after = y_res.value_counts()

#Visualisasi perbandingan sebelum dan sesudah diterapkan oversampling SMOTE
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

ax[0].bar(before.index, before.values)
ax[0].set_title('Sebelum SMOTE')
ax[0].set_xlabel('Kelas')
ax[0].set_ylabel('Jumlah')

ax[1].bar(after.index, after.values)
ax[1].set_title('Setelah SMOTE')
ax[1].set_xlabel('Kelas')
ax[1].set_ylabel('Jumlah')

plt.show()
```

Gambar 2. 6 *Data Balancing*

Berikut adalah tabel 2.6 yang berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 6 *Parameter Data Balancing*

Parameter	Keterangan
drop()	Fungsi drop digunakan untuk melakukan penghapusan terhadap kolom atau atribut yang diinginkan.
value_counts()	Fungsi ini digunakan untuk menghitung jumlah setiap nilai pada atribut yang diinginkan.
SMOTE()	Untuk membuat sebuah objek SMOTE yang digunakan untuk *oversampling*.
fit_resample	Metode yang digunakan oleh objek SMOTE yang telah dibuat sebelumnya untuk menerapkan teknik *oversampling* pada data
plt.subplots	Membuat grid subplot dengan 1 baris dan 2 kolom
figsize	Menentukan ukuran dari keseluruhan figur. Lebar 12 inci dan tinggi 6 inci.
Plt.show	Menampilkan plot yang telah dibuat

2.2.4. Pembagian Data

Dalam penelitian klasifikasi menggunakan algoritma KNN, Selanjutnya dataset yang akan digunakan dibagi menjadi dua bagian utama: data latih dan data uji. Untuk memastikan evaluasi model yang akurat dan konsisten, peneliti mengadopsi teknik *K-Fold Cross Validation*. pengujian menggunakan *k-fold cross validation*, dengan seiring bertambahnya nilai K (tetangga terdekat) maka akan mempengaruhi nilai akurasi (Nabila et al., 2021). Teknik ini diimplementasikan melalui *library sklearn.model_selection* dengan menggunakan fungsi *cross_val_score* di lingkungan Python. Dengan pendekatan ini, peneliti dapat memvalidasi kinerja model secara menyeluruh dengan membagi dataset dan menghitung skor rata-rata dari hasil pengujian. Teknik *K-Fold Cross Validation* yang digunakan dengan membagi data menjadi 10 kelompok, yang berarti data akan dibagi menjadi 10 bagian.

```
# K-fold cross validation
kf = KFold(n_splits=10, shuffle=True, random_state=42)

for k in k_values:
    print(f"\nk = {k}")

    fold_scores = []
    fold_confusion_matrices = []
    kFold = 1

    for train_index, test_index in kf.split(X_res, y_res):
        X_train, X_test = X_res.iloc[train_index], X_res.iloc[test_index]
        y_train, y_test = y_res.iloc[train_index], y_res.iloc[test_index]
```

Gambar 2. 7 Pembagian Data

Berikut adalah tabel 2.7 yang berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 7 *Parameter* Pembagian Data

Parameter	Keterangan
KFold	Metode validasi silang untuk membagi data menjadi beberapa lipatan yang kemudian akan menggunakanya secara bergantian sebagai *dataset training* dan *testing*.
n_splits=5	Menentukan jumlah lipatan yang akan dibuat.
shuffle=True	Menginisiasikan apakah ingin menyatukan kembali data sebelum membaginya menjadi lipatan atau tidak.
random_state=42	Nilai yang digunakan untuk melakukan kontrol terhadap proses penyatuan kembali agar hasil dapat direproduksi secara konsisten
Train_index & *test_index*	Setiap iterasi dari loop yang terjadi akan mendapatkan indeks yang dihasilkan untuk data *training* dan *testing*.
kf.split(X)	Metode yang mengembalikan generator yang menghasilkan indeks untuk setiap lipatan, yang dimana setiap iterasi akan menghasilkan indeks untuk dataset *training* dan *testing*.
X.iloc[train_index], X.iloc[test_index]	Menggunakan indeks yang telah dihasilkan untuk memilih baris yang sesuai dari atribut yang ada pada 'X' untuk dataset *training* dan *testing*.
y.iloc[train_index], y.iloc[test_index]	Menggunakan indeks yang telah dihasilkan untuk memilih baris yang sesuai dari atribut yang ada pada 'y' untuk dataset *training* dan *testing*.

2.2.5. Modelling

Pada tahap ini, akan diuraikan mengenai model yang digunakan dalam penelitian ini. Model klasifikasi yang digunakan adalah *K-Nearest Neighbor* (KNN) dengan *Relief* sebagai teknik seleksi fitur dan *Particle Swarm Optimization* (PSO) sebagai metode optimasi. Sebelum dataset diproses, pendekatan *Synthetic Minority Oversampling Technique* (SMOTE) akan digunakan untuk menangani ketidak seimbangan data(*Imbalanced data*). akhir dari penelitian akan melakukan perbandingan dengan menggunakan seleksi fitur *Relief* dan tanpa menggunakan seleksi fitur *Relief*. berikut proses pemodelannya.

a. Permodelan Algoritma KNN

Tahap awal Menyiapkan dataset banjir yang telah melalui tahap data balancing menggunakan metode *oversampling* SMOTE dengan detailnya seperti yang dijelaskan pada gambar 2.5 lalu dibagi menjadi data *training* dan data *testing* yang dikhususkan untuk model dalam mempelajari pola data dengan menggunakan teknik *10-Fold Cross-Validation.*

Selanjutnya permodelan ini akan menggunakan *K-Nearest Neighbor* (KNN) dalam mencari hasil evaluasi *confusion matrix* dan juga prediksi nilai akurasi.

```
# Inisiasi algoritma k-nearest neighbors
knn = KNeighborsClassifier(n_neighbors=k, metric='euclidean')

# Melatih model KNN
knn.fit(X_train, y_train)

# Predict on the test set
y_pred = knn.predict(X_test)
```

Gambar 2. 8 Persiapan *Insialisasi Model KNN*

Pada tabel 2.8 berisikan tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 8 *Parameter* Persiapan *Insialisasi Model KNN*

Parameter	Keterangan
KneighborsClassifier()	kelas dari library scikit-learn yang digunakan untuk membuat model klasifikasi menggunakan algoritma *K-Nearest Neighbors* (KNN).
n_neighbors=5	Parameter ini menetapkan jumlah tetangga terdekat yang akan digunakan dalam algoritma KNN.
knn.fit()	metode yang digunakan untuk melatih model KNN pada data pelatihan.
X_train	variabel yang menyimpan data fitur yang digunakan untuk melatih model.
y_train	variabel yang menyimpan label atau target yang sesuai dengan data fitur di *X_train.*
predict()	metode yang digunakan untuk membuat prediksi menggunakan model KNN yang telah dilatih.

Adapun rumus yang biasa digunakan dalam melakukan perhitungan algoritma *K-Nearest Neighbor* (KNN) adalah sebagai berikut :

$$\sqrt{\sum_{i=1}^{p} \quad (\alpha_{k-} b_k)^2}$$

(2. 1)

Sumber : (Jatmiko Indriyanto, 2021)

Keterangan :

α_K : Sampel data

b_k : Data uji atau Data testing

P : Dimensi data

i : variable data

b. Permodelan Algoritma KNN – PSO

Pemodelan ini akan menerapkan *Particle Swarm Optimization* (PSO) untuk mengoptimalkan parameter-parameter algoritma klasifikasi dengan tujuan meningkatkan performa keseluruhan model. Tahapan dalam pemodelan ini dimulai dengan menginstal pustaka *pyswarm* Berikut adalah penerapan PSO pada model yang akan dioptimalkan.

```
!pip install pyswarm
```

Gambar 2. 9 *Menginstall PSO*

Setelah melakukan pengginstalan, selanjutnya dilakukan pengimporan modul PSO dari *pyswarm* dan penerapan PSO

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
from pyswarm import pso  # Make sure you have the PSO library installed

# Function to optimize KNN
def optimize_knn(x, k):
    leaf_size = int(x[0])
    p = int(x[1])

    clf = KNeighborsClassifier(n_neighbors=k, leaf_size=leaf_size, p=p)
    scores = cross_val_score(clf, X_res, y_res, cv=10)

    return -scores.mean()  # Maximizing accuracy by minimizing the negative score

# Bounds for KNN hyperparameters
lb = [10, 1]  # Lower bounds for leaf_size and p
ub = [50, 2]  # Upper bounds for leaf_size and p

# List of odd k values to optimize
k_values = [3, 5, 7, 9, 11, 13, 15]

optimal_parameters = {}

for k in k_values:
    # Perform PSO optimization for each k
    xopt, fopt = pso(optimize_knn, lb, ub, args=(k,), swarmsize=50, maxiter=30)
```

Gambar 2. 10 *Mengimpor* dan *menerapkan PSO* Pada *Model*

Berikut adalah tabel 2.9 yang berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 9 Parameter *Mengimpor* dan *menerapkan PSO* Pada *Model*

Parameter	Keterangan
optimize_knn(x)	Sebuah fungsi objektif yang akan dioptimalkan yang menerima vector *'x'* berisikan parameter yang akan dioptimalkan.
cross_val_score()	Menginisialisasi model *K-Nearest Neighbort* berdasarkan parameter yang diberikan. Dalam fungsi tersebut melakukan *cross validation* dengan *5-fold cross-validation*. Kemudian mengembalikan rata-rata negative dari skor *cross validation*.
lb	Batas bawah untuk setiap parameter dalam vektor *'x'*.
ub	Batas atas untuk setiap parameter dalam vektor *'x'*.
xopt, fopt	Berfungsi untuk mendapatkan kombinasi parameter terbaik yang meminimalkan fungsi *'optimize_rf'*.
swarmsize=50	Mengindikasikan bahwa terdapat 50 partikel dalam *swarm* yang akan dievaluasi pada setiap iterasi.
maxiter=30	Jumlah iterasi maksimal yang akan dijalankan oleh PSO.

c. Permodelan Algoritma KNN –RELIEF

Dalam penelitian ini, algoritma *Relief* akan digunakan untuk seleksi fitur. Pada tahap seleksi fitur, fitur-fitur yang ada akan diurutkan berdasarkan pengaruhnya terhadap hasil prediksi, dimulai dari fitur yang memiliki pengaruh terbesar hingga fitur yang memiliki pengaruh terkecil atau bahkan tidak berpengaruh sama sekali. Tahapannya meliputi mengimpor modul *Relief* dari library *sklearn* dan modul pandas. Modul pandas digunakan untuk mengubah dataset menjadi bentuk dataframe. Setelah itu, fitur-fitur akan dirangking berdasarkan skor yang diperoleh dari algoritma *Relief*, berikut penerapannya.

```
!pip install ReliefF
```

Gambar 2. 11 *Menginstall Relief*

Berikut adalah tabel 2.10 yang berisi tentang tiap parameter yang digunakan dalam kode tersebut beserta penjelasan fungsi tiap parameter.

Tabel 2. 10 Parameter *Relief*

Parameter	Keterangan
SelectKBest()	Fungsi dari *library sklearn* yang digunakan untuk memilih fitur-fitur terbaik berdasarkan nilai kriteria tertentu.
f_classif	Fungsi yang digunakan untuk mendapatkan skor relief yang akan digunakan sebagai kriteria untuk pemilihan fitur.
fit_transform	Kombinasi dari fit dan transform. Pertama, fit menyesuaikan selektor dengan data, kemudian transform menerapkan seleksi fitur pada data.
X_res	Artibut hasil dari proses *oversampling* SMOTE.
Y_rres	Label atau target yang telah diproses dala
*Fs.top_features_*	Mendapatkan indeks fitur yang dipilih berdasarkan skor tertinggi.
selected_features_names	Menyimpan nama dari fitur yang dipilih.
pd.DataFrame	Digunakan untuk membuat *dataframe* berdasarkan skor dan *p-value* yang dihasilkan sebelumnya.
feature_weights	Menyimpan bobot atau skor pentingnya dari fitur-fitur yang dipilih berdasarkan hasil perhitungan algoritma *Relief*.
*fs.feature_importances_*	Atribut ini berisi bobot atau skor pentingnya setiap fitur yang dihitung oleh algoritma ReliefF.
selected_features_indices	Indeks fitur yang dipilih.

Adapun rumus Langkah-langkah untuk melakukan seleksi fitur *Relief* sebagai berikut:

a. Inisialisasi nilai awal seluruh bobot fitur = 0 dan menentukanjumlah iterasi.
b. Memilih sebuah data yang akan dijadikan sebagai titik acak atau titik pusat.
c. Mencari miss dan hitterdekat dengan cara menghitung jarak antara titik pusat dengan data yang memiliki kelas yang sama. Jarak terdekat antara titik pusat dan data pada kelas positif disebut hit.Sedangkan, jarak terdekat antara titik pusat dengan data yang pada kelas negatifdisebut miss.
d. Lakukan update bobot untuk setiap fitur. Fitur dengan data kategori dihitung menggunakan Persamaan 2.2.

$$diff(A, Ri, HM) = \{0; value(A, Ri) = value(A, HM) \\ 1; otherwise \quad (2.2)$$

e. Sedangkan, fitur dengan data numerik dihitung menggunakan persamaan 2.3.

$$diff(A, Ri, HM) = \frac{\text{value(A,Ri)} - \text{value(A,HM)}|}{\max(A) - \min(A)} \quad (2.3)$$

f. Sehingga rumus perbaruan bobot dihitung menggunakan persamaan 2.4

$$W[A] = W[A] - diff(A, Ri, H)m + diff(A, Ri, M)m \quad (2.4)$$

Selanjutnya, dilakukan iterasi yang dimulai dari langkah 1 hingga bobot fitur yang baru telah didapat.

c. Permodelan Algoritma KNN - RELIEF –PSO

Dalam penelitian ini, akan mengkombinasi tiga algoritma yang masing-masing telah diselesaikan tahapannya, algoritma tersebut yakni KNN, *Relief*, dan PSO. Tahap awal dimulai dengan mengimpor pustaka, termasuk dari pustaka *sklearn*, modul *SelectKBes*t dan *Relief* dari modul seleksi fitur untuk memilih fitur, serta *KNeighborsClassifier* untuk membuat model KNN, dan modul PSO dari *library pyswarm* untuk mengoptimalkan PSO. Selanjutnya, akan dilakukan proses persiapan dan pembagian data menggunakan *kfold cross validation*, diikuti dengan penerapan algoritma seleksi fitur untuk mendapatkan fitur yang relevan. Langkah berikutnya adalah menerapkan optimasi menggunakan PSO untuk menyesuaikan parameter-parameter algoritma klasifikasi dengan tujuan meningkatkan performa keseluruhan model. Setelah itu, kami melatih model klasifikasi KNN dengan parameter yang telah dioptimalkan.

d. Perbandingan Hasil

Hasil akhirnya akan dilakukan perbandingan mengenai penerapan *Feature Selection* dan tanpa *Feature Selection* untuk melihat seberapa optimal *KNN* bekerja dalam menemukan hasil optimal baik dengan menggunakan *Oversampling* dan optimasi maupun tidak sama sekali.

2.2.5. Evaluasi

Dalam penelitian ini, kinerja model KNN akan dievaluasi dengan membandingkan beberapa model KNN sebelumnya, baik yang menggunakan teknik seleksi fitur maupun yang tidak. Evaluasi model dilakukan untuk mendapatkan model terbaik melalui pengujian terhadap data uji berdasarkan *Confusion Matrix.* Evaluasi dilakukan menggunakan accuracy sebagai metrik, yang dapat dihitung dengan persamaan berikut :

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} X100\% \quad (2.5)$$

Keterangan:

TP (*True Positive*) : jumlah data yang berlabel yes diklasifikasikan sebagai benar oleh model.

TN (*True Negative*) : jumlah data yang berlabel no diklasifikasikan sebagai salah oleh model.

FP *(False Positive)* : jumlah data yang berlabel yes seharusnya salah.

FN (*False Negative*) : jumlah data yang berlabel no diklasifikasikan padahal seharusnya benar.

BAB III
HASIL DAN PEMBAHASAN

3.1. Hasil Penilitian

Pada penelitian ini, dilakukan analisis terhadap data banjir di Kota Samarinda, Data yang digunakan dalam penelitian ini mencakup berbagai variabel yang mempengaruhi kejadian banjir di Kota Samarinda, data tersebut terdiri atas 1095 Dataset dengan 19 Atribut dan 1 Atribut sebagai label didalamnya. Data ini memiliki karakteristik berdimensi tinggi yang memerlukan teknik seleksi fitur untuk meningkatkan akurasi model prediksi. Proses seleksi fitur dilakukan menggunakan algoritma *Relief* yang dioptimalkan dengan PSO. *Relief* berfungsi untuk menilai kepentingan setiap fitur berdasarkan kemampuan mereka dalam membedakan kelas target, dalam hal ini kejadian banjir dan tidak banjir. Hasil seleksi fitur menunjukkan bahwa dari sekian banyak variabel, terdapat 9 fitur utama yang paling signifikan dalam memprediksi kejadian banjir di Samarinda. Setelah seleksi fitur, model dioptimalkan menggunakan PSO untuk menentukan parameter terbaik yang memaksimalkan kinerja model.

3.1.1. Pengumpulan Data

Penelitian ini menggunakan total data banjir sebanyak 1095 *record*. yang didapatkan dari BPBD dan BMKG Kota Samarinda tahun 2021-2023 dalam dataset tersebut, terdapat 49 *record* yang menunjukkan adanya kejadian banjir, sementara 841 *record* menunjukkan tidak adanya kejadian banjir dengan label kelas 0 menunjukkan tidak terjadi banjir, sedangkan label kelas 1 menunjukkan terjadi banjir.

Data yang diperoleh dari BMKG yang memiliki 11 fitur sedangkan data yang diperoleh dari BPBD memiliki 9 fitur. Data yang didapatkan dari BMKG meliputi tanggal, temperatur maksimum (Tx), temperatur minimum (Tn), temperatur rata-rata (Tavg), kelembaban rata-rata (RH-avg), curah hujan (RR), lamanya penyinaran matahari (ss), kecepatan angin maksimum (ff_x), arah angin maksimum (ddd_x), kecepatan angin rata-rata (ff_avg), dan arah angin terbanyak (ddd_car). Sedangkan data yang didapatkan dari BPBD meliputi tanggal, jam kejadian, jenis bencana, lokasi wilayah, luas area m2, objek terkena bencana, korban, kerugian, dan keterangan. hasil penggabungan data dari pendekatan *excel* tersebut dapat dilihat dalam tabel berikut :

Tabel 3. 1 Data Yang Diperoleh Dari BMKG

Tanggal	Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_car
01-01-2021	23	33,2	26,5	88	1,8	3,3	4	280	2	W
02-01-2021	23,2	30,8	27,1	88	7	6,4	2	140	1	C
03-01-2021	24,8	32,7	27,3	84	2	1,2	5	290	2	NW
04-01-2021	24,6	31,8	28,1	84	2,7	5,4	3	300	2	NW
05-01-2021	24,6	31,4	27,4	83	10,5	1,8	4	300	2	W
…	…	…	…	…	…	…	…	…	…	…
27-12-2023	24,2	32	27,6	83	0,3	2,8	4	60	2	NE
28-12-2023	24	32	28	82	1	6,4	5	70	2	C
29-12-2023	23,7	32,7	28,7	77	3,5	5,9	5	60	2	NE
30-12-2023	24,2	32,6	28,3	82		10,4	4	60	1	NE
31-12-2023	24,6	32,4	28,3	84	0	6,9	4	90	2	E

Selain data dari BMKG, terdapat data dari bpbd yang ditunjukkan pada Tabel 3,2, sebagi berikut.

Tabel 3. 2 *Data Yang Diperoleh Dari BPBD*

NO	TANGGAL	JAM KEJADIAN	JENIS BENCANA	LOKASI/ WILAYAH KELURAHAN/ KECAMATAN	LUAS AREA M²	JUMLAH OBYEK YANG TERKENA BENCANA	Korban				JUMLAH		KERUGIAN (RP)	KETERANGAN
							KL	KS	KH	KM	KK	JIWA		
1	03 Januari 2021	-	Banjir	Jl. Irigasi RT. 50 Kel. Rawa Makmur Kec. Palaran (Dataran Rendah) Wilayah Handil Bakti RT. 1, RT. 2, RT 3 (Dataran Rendah)	± -	Jalan mejadi Susah Untuk Di lalui Dan Menganggu aktivitas warga	-		-	-	-	-	-	Genangan Air Penyebab Air Sungai Mahakam pasang Dan Lokasi Banjir adalah Dataran Rendah
2	Selasa, 05 Januari 2021	-	Pohon Tumbang	Jl. Kesehatan Dalam Kel. Temindung Permai	-	Jalan mejadi Susah Untuk Di lalui Dan Menganggu aktivitas warga	-	-	-	-	-	-	-	Penyebab : Hujan Deras dan Angin Kencang
3	Kamis. 07 Januari 2021	-	TANAH LONGSOR	Jl Wiraswasta Gang Bukit Indah Rt.15 Kel.Sidodadi	-	1 Tiang listrik roboh	-	-	-	-	-	-	-	Akibat hujan intentias deras dan angin kencang
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19	Rabu, 13 Desember 2023	Pukul 21.33 Wita	Pohon Tumbang	Jl. Gunung Tabur Kel. Gunung Kelua Kec. Samarinda Ulu		Akses jalan tertutup								Penyebab Hujan deras disertai angin kencang Upaya: Melakukkan Pemangkasan
20	Jum'at, 15 Desember 2023	Pukul 15.00 Wita	Pohon Tumbang	Jl. Balai Kota Samarinda Kel. Bugis Kec. Samarinda Kota		Dampak: - Mengenai kanopi Parkiran Bus Pemkot Samarinda								Dampak: Angin Kencang Upaya: Melakukkan Pemangkasan

3.1.2. Data Preparation

a. Data Integration

Data yang didapatkan dari BPBD dan BMKG selanjutnya akan digabungkan menjadi satu untuk mempermudah pengolahan data sehingga data yang didapatkan lebih lengkap untuk mengetahui informasi mengenai penyebab terjadinya bencana banjir. Setelah kedua data digabungkan maka didapatkan 19 atribut dan 1 atribut yang dijadikan sebagai kelas.

Proses digabungkannya kedua data yang telah didapatkan menggunakan perangkat lunak *Microsoft Office Excel*. Adapun data dari BPBD yang digabungkan yaitu tanggal, jam kejadian, jenis bencana, lokasi wilayah, luas area, objek terkena bencana, korban, kerugian, dan Keterangan. Kemudian pada data dari BMKG yang memilki fitur temperatur minimum, temperatur maksimum, temperatur rata-rata, kelembaban rata-rata, curah hujan, lamanya penyinaran matahari, kecepatan angina maksimum, arah angin saat kecepatan maksimum, kecepatan angin rata-rata, dan arah angin terbanyak.

Tabel 3. 3 Hasil Data *Integration*

No	Atribut	Tipe Data	Keterangan
1	Tanggal	*date*	Tanggal Kejadian
2	Jam Kejadian	*String*	Jam kejadian
3	Jenis Bencana	*string*	Bencana alam yang terjadi
4	Lokasi wilayah	*string*	Tempat terjadinya banjir
5	Luas Area M2	*numeric*	Luas area yang terdampak
6	Objek terkena bencana	*string*	Kerugian fasilitas yang terdampak bencana
7	Korban	*numeric*	Jumlah korban terdampak bencana
8	Kerugian	*numeric*	Nominal kerugian
9	Keterangan	*numeric*	Detail kejadian bencana
10	Tn	*numeric*	Temperatur minimum (°C)
11	Tx	*numeric*	Temperatur maksimum (°C)
12	Tavg	*numeric*	Temperatur rata-rata (°C)
13	RH_avg	*numeric*	Kelembaban rata-rata (%)
14	RR	*numeric*	Curah hujan (mm)
15	Ss	*numeric*	Lamanya penyinaran matahari (jam)
16	ff_x	*numeric*	Kecepatan angin maksimum (m/s)
17	ddd_x	*numeric*	Arah angin saat kecepatan maksimum (°)
18	ff_avg	*numeric*	Kecepatan angin rata-rata (m/s)
19	ddd_car	*string*	Arah angin terbanyak (°)

b. Data Selection

Tahap ini dilakukan proses pemilihan atribut-atribut yang telah dipilih secara manual pada Data Banjir dari BMKG dan BPBD yang diperoleh sebanyak 19 Atribut yang terpilih dan 1 atribut sebagai target atau kelas.

	Tanggal	Temperatur-minimum	Temperatur-maksimum	Temperature-rata-rata	Kelembaban	Curah-hujan	Lama-penyinaran-matahari	Kecepatan-angin	Arah-angin-maksimum	Kecepatan-angin-rata-rata	Arah-angin-terbanyak	terjadi-banjir
0	01-01-2021	23.0	33.2	26.5	88.0	1.8	3.3	4.0	280.0	2.0	W	tidak banjir
1	02-01-2021	23.2	30.8	27.1	88.0	7.0	6.4	2.0	140.0	1.0	C	tidak banjir
2	03-01-2021	24.8	32.7	27.3	84.0	2.0	1.2	5.0	290.0	2.0	NW	banjir
3	04-01-2021	24.6	31.8	28.1	84.0	2.7	5.4	3.0	300.0	2.0	NW	tidak banjir
4	05-01-2021	24.6	31.4	27.4	83.0	10.5	1.8	4.0	300.0	2.0	W	tidak banjir
...	...	...	...	...	...	...	...	...	...	...	...	...
1090	27-12-2023	24.2	32.0	27.6	83.0	0.3	2.8	4.0	60.0	2.0	NE	tidak banjir
1091	28-12-2023	24.0	32.0	28.0	82.0	1.0	6.4	5.0	70.0	2.0	C	tidak banjir
1092	29-12-2023	23.7	32.7	28.7	77.0	3.5	5.9	5.0	60.0	2.0	NE	tidak banjir
1093	30-12-2023	24.2	32.6	28.3	82.0	NaN	10.4	4.0	60.0	1.0	NE	tidak banjir
1094	31-12-2023	24.6	32.4	28.3	84.0	0.0	6.9	4.0	90.0	2.0	E	tidak banjir

1095 rows × 12 columns

Gambar 3. 1 Hasil *Data Selection*

c. Data Cleaning

Pada tahap ini akan menggunakan bahasa pemrograman *Python* serta *library Pandas*, dengan fungsi *dropn*a() untuk menghapus baris yang memiliki nilai kosong pada dataset Banjir dengan jumlah awal 1095 *record.*

	Tanggal	Temperatur-minimum	Temperatur-maksimum	Temperature-rata-rata	Kelembaban	Curah-hujan	Lama-penyinaran-matahari	Kecepatan-angin	Arah-angin-maksimum	Kecepatan-angin-rata-rata	Arah-angin-terbanyak	terjadi-banjir
0	01-01-2021	23.0	33.2	26.5	88.0	1.8	3.3	4.0	280.0	2.0	W	tidak banjir
1	02-01-2021	23.2	30.8	27.1	88.0	7.0	6.4	2.0	140.0	1.0	C	tidak banjir
2	03-01-2021	24.8	32.7	27.3	84.0	2.0	1.2	5.0	290.0	2.0	NW	banjir
3	04-01-2021	24.6	31.8	28.1	84.0	2.7	5.4	3.0	300.0	2.0	NW	tidak banjir
4	05-01-2021	24.6	31.4	27.4	83.0	10.5	1.8	4.0	300.0	2.0	W	tidak banjir
...	...	...	...	...	...	...	...	...	...	...	...	...
1090	27-12-2023	24.2	32.0	27.6	83.0	0.3	2.8	4.0	60.0	2.0	NE	tidak banjir
1091	28-12-2023	24.0	32.0	28.0	82.0	1.0	6.4	5.0	70.0	2.0	C	tidak banjir
1092	29-12-2023	23.7	32.7	28.7	77.0	3.5	5.9	5.0	60.0	2.0	NE	tidak banjir
1093	30-12-2023	24.2	32.6	28.3	82.0	NaN	10.4	4.0	60.0	1.0	NE	tidak banjir
1094	31-12-2023	24.6	32.4	28.3	84.0	0.0	6.9	4.0	90.0	2.0	E	tidak banjir

1095 rows × 12 columns

Gambar 3. 2 *Dataset* sebelum dibersihkan

Pada tampilan data, Sebelum proses pembersihan data dilakukan, terdapat nilai-nilai kosong pada data. Oleh karena itu, diperlukan penghapusan data yang tidak lengkap. Jumlah total data sebelum pembersihan adalah 1095 baris. Setelah melalui proses *data cleaning*, jumlahnya berkurang menjadi 890 baris. Dengan demikian, terdapat 205 baris data kosong yang telah dihapus selama proses pembersihan data.

```
Tanggal                       0
Temperatur-minimum           73
Temperatur-maksimum          10
Temperature-rata-rata         6
Kelembaban                    7
Curah-hujan                 133
Lama-penyinaran-matahari      8
Kecepatan-angin               2
Arah-angin-maksimum           2
Kecepatan-angin-rata-rata     2
Arah-angin-terbanyak          2
terjadi-banjir                0
dtype: int64
```

Gambar 3. 3 Nilai kosong pada tiap *atribut* sebelum *data cleaning*

Pada gambar 3.3 merupakan nilai kosong pada tiap atribut sebelum *data cleaning*, dimana curah hujan yang terbesar nilai kosongnya, sebesar 133. Selanjutnya akan dilakukan proses pemeriksaan seperti yang ada pada gambar 3.4.

```
Jumlah data sebelum pembersihan data kosong: 1095
Jumlah data setelah pembersihan data kosong: 890
```

Gambar 3. 4 Jumlah Sebelum dan sesudah *Data Cleaning*

Setelah dilakukan pemeriksaan pada data yang ada, berikut hasil dari *cleaning data*.

No	Tanggal	Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_car	terjadi_banjir
1	01-01-2021	23	33,2	26,5	88	1,8	3,3	4	280	2	8	0
2	02-01-2021	23,2	30,8	27,1	88	7	6,4	2	140	1	0	0
3	03-01-2021	24,8	32,7	27,3	84	2	1,2	5	290	2	4	1
4	04-01-2021	24,6	31,8	28,1	84	2,7	5,4	3	300	2	4	0
5	05-01-2021	24,6	31,4	27,4	83	10,5	1,8	4	300	2	8	0
....	...	...	...	...	...	...	...	...	...	...	...	...
876	26-12-2023	24,0	30	26,9	89	50	7,0	3,0	260	1	6	0
877	27-12-2023	24,2	32	27,6	83	0,3	2,8	4,0	60	2	3	0
878	28-12-2023	24,0	32,0	28,0	82	1,0	6,4	5	70	2	0	0
879	29-12-2023	23,7	32,7	28,7	77	3,5	5,9	5,0	60,0	2,0	3	0
890	31-12-2023	24,6	32,4	28,3	84	0	6,9	4	90	2	1	0

Gambar 3. 5 Hasil Data *Cleaning*

Gambar 3.5 merupakan hasil proses data cleaning, selanjutnya setelah proses pembersihan data seperti yang ditunjukkan pada Gambar 3.5, jumlah data dengan nilai kosong telah dihilangkan, dan data yang tersisa setelah proses pembersihan berjumlah 890 *record.* Kemudian dilakukan pemeriksaan kembali, untuk memastikan bahwa tidak ada lagi data kosong pada data banjir, dengan hasil pemeriksaan sebagai yang ditunjukkan pada gambar 3.6.

```
Jumlah nilai yang hilang setelah pembersihan:
Tanggal                     0
Temperatur-minimum          0
Temperatur-maksimum         0
Temperature-rata-rata       0
Kelembaban                  0
Curah-hujan                 0
Lama-penyinaran-matahari    0
Kecepatan-angin             0
Arah-angin-maksimum         0
Kecepatan-angin-rata-rata   0
Arah-angin-terbanyak        0
terjadi-banjir              0
dtype: int64
```

Gambar 3. 6 Jumlah Nilai Kosong Tiap Kolom Setelah Pembersihan

d. Data Transformation

Pada tahap transformasi data, dilakukan pengubahan terhadap data kategorikal menjadi numerik. Data yang diubah pada tahap ini meliputi 'Arah-angin-terbanyak' dan 'terjadi_banjir'. Dalam proses ini, pengubahan data kategorikal menjadi numerik untuk variabel 'Arah-angin-terbanyak' menggunakan *library* `*LabelEncoder()*` dari *sklearn*, sedangkan pengubahan untuk variabel 'terjadi_banjir' menggunakan fungsi `*replace()*` dari Python.

Tabel 3. 4 *Dataset* Sebelum *ditranformasi*

No	Arah-angin-terbanyak	terjadi-banjir
0	W	Tidak Banjir
1	C	Tidak Banjir
2	NW	Banjir
3	NW	Tidak Banjir
4	W	Tidak Banjir
…		
1089	SE	Tidak Banjir
1090	NE	Tidak Banjir
1091	C	Tidak Banjir
1092	NE	Tidak Banjir
1094	E	Tidak Banjir

Pada Tabel 3.4 merupakan tampilan data pada kolom atribut 'Arah-angin-terbanyak' dan 'terjadi_banjir' sebelum transformasi data menunjukkan bahwa data berbentuk kategorikal.

Tabel 3. 5 *Dataset* Setelah *ditranformasi*

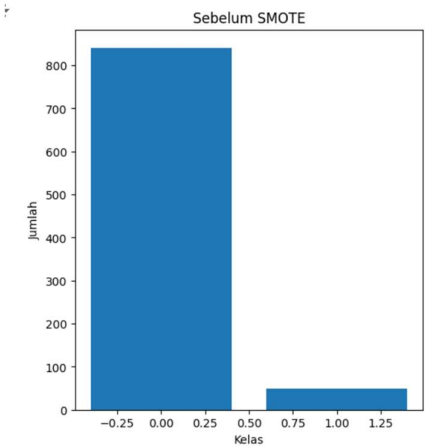
No	**Arah-angin-terbanyak**	**Terjadi-banjir**
0	8	0
1	0	0
2	4	1
3	4	0
4	8	0
…	…	…

No	Arah-angin-terbanyak	Terjadi-banjir
1090	6	0
1091	3	0
1092	0	0
1093	3	0
1094	1	0

Pada Tabel 3.5 merupakan tampilan data pada kolom atribut 'Arah-angin-terbanyak' dan 'terjadi_banjir' setelah *transformasi* data menunjukkan bahwa data yang sebelumnya berbentuk kategorikal telah diubah menjadi *numeric* untuk memudahkan proses klasifikasi.

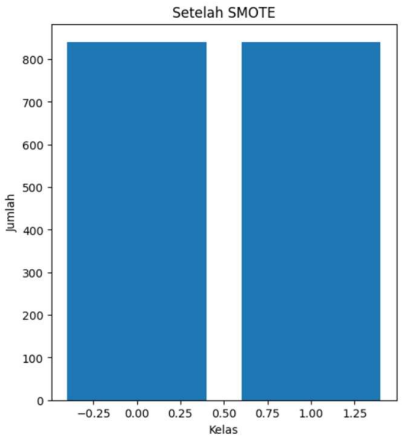
e. Data Balancing

Proses penyeimbangan data dilakukan menggunakan modul Python *`imblearn.over_sampling`* dengan mengimpor fungsi SMOTE (*Sampling Strategy*) untuk melakukan *oversampling.* Teknik *oversampling* ini diterapkan untuk menyamakan jumlah sampel antara kelas minoritas dan kelas mayoritas dalam dataset banjir yang memiliki ketidakseimbangan kelas. Dengan menerapkan SMOTE, ketidakseimbangan kelas pada data banjir dapat diatasi secara efektif.



Gambar 3. 7 Jumlah Kelas Sebelum Penerapan *SMOTE*

Pada Gambar 3.7 terdapat perbedaan jumlah kelas dimana kategori Tidak Banjir (0) berjumlah 841 data dan kategori terjadi-banjir (1) berjumlah 49 data.



Gambar 3. 8 Jumlah Kelas Sesudah Penerapan *SMOTE*

Pada Gambar 3.8 Setelah *oversampling* SMOTE diterapkan, jumlah sampel di kedua kelas menjadi seimbang, masing-masing dengan 841 sampel. Dengan demikian, setelah penerapan SMOTE, total jumlah data meningkat menjadi 1682 *record.*

3.1.3. Hasil Permodelan KNN Dengan *Pyhton*

Dalam penelitian ini, tahap pemodelan menggunakan algoritma KNN dengan nilai k=3, k=5, k=7, k=11, k=13, dan k=15, serta pembagian data menggunakan 10-*fold cross-validation*. Evaluasi dilakukan menggunakan *Confusion Matrix*. Hasil evaluasi rata-rata *Confusion Matrix* untuk seluruh *fold* dari masing-masing nilai k dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Hasil Evaluasi *Confusion Matrix*

Nilai K	TN	FP	FN	TP	*Mean Accuracy*
K=3	66	1.9	18	82	88,23%
K=5	62	2.6	22	82	85,38%
K=7	60	3.2	24	81	84,07%
K=9	58	3.6	26	80	82,52%
K=11	57	4.4	27	80	81,27%
K=13	56	4.9	28	79	80,62%
K=15	54	5.1	30	79	79,31%

Dapat dilihat pada tabel 3.6 bahwa hasil akurasi rata-rata dari masing-masing nilai K pada keseluruhan *fold* dengan total nilai TN = 66 adalah yang paling tinggi yang didapatkan dari k=3, nilai FP = 5.1 adalah yang paling tinggi didapatkan pada k=15, nilai FN = 28 memiliki nilai tinggi yang didapatkan pada K=15 ,dan nilai TP = 82 adalah nilai yang tinggi pada k=7 dan k=5. bisa dilihat juga bahwa akurasi rata-rata terbaik yang didapatkan dari keseluruhan *fold* adalah k=3 sebesar 88,23%, sedangkan untuk akurasi terendah adalah k=15 dengan nilai akurasi rata-rata 79,31%.

3.1.4. Hasil Permodelan KNN + PSO

PSO bertindak sebagai metode optimasi untuk mengoptimalkan parameter-parameter algoritma klasifikasi dengan tujuan meningkatkan kinerja keseluruhan model. Dalam konteks algoritma KNN, parameter-parameter yang dioptimalkan meliputi jumlah jumlah tetangga (*n_neighbors*), kedalaman maksimum pohon (*max_depth*), jumlah minimum sampel untuk memisah internal node (*min_samples_split*), dan jumlah minimum sampel di *leaf node (min_samples_leaf)*. Pengoptimalan ini dilakukan untuk memastikan model mencapai kinerja yang optimal dalam hal akurasi, sehingga model dapat menghasilkan prediksi yang lebih akurat dan refektif.

***Tabel 3. 7** Hasil E*valuasi *k-nearest neighbors (KNN) + PSO*

Nilai K	TN	FP	FN	TP	*Mean Accuracy*
K=3	84	15	0.6	69	90,85%
K=5	83	17	0.8	67	89,42%
K=7	84	20	0.5	64	87,70%
K=9	83	22	0.9	62	86,50%
K=11	83	22	0.8	62	86,39
K=13	83	23	1.3	61	85,55%
K=15	83	24	60	1.4	84,84%

Dapat dilihat pada tabel 3.7 bahwa hasil akurasi rata-rata dari masing-masing nilai K pada keseluruhan fold mengalami kenaikan akurasi.

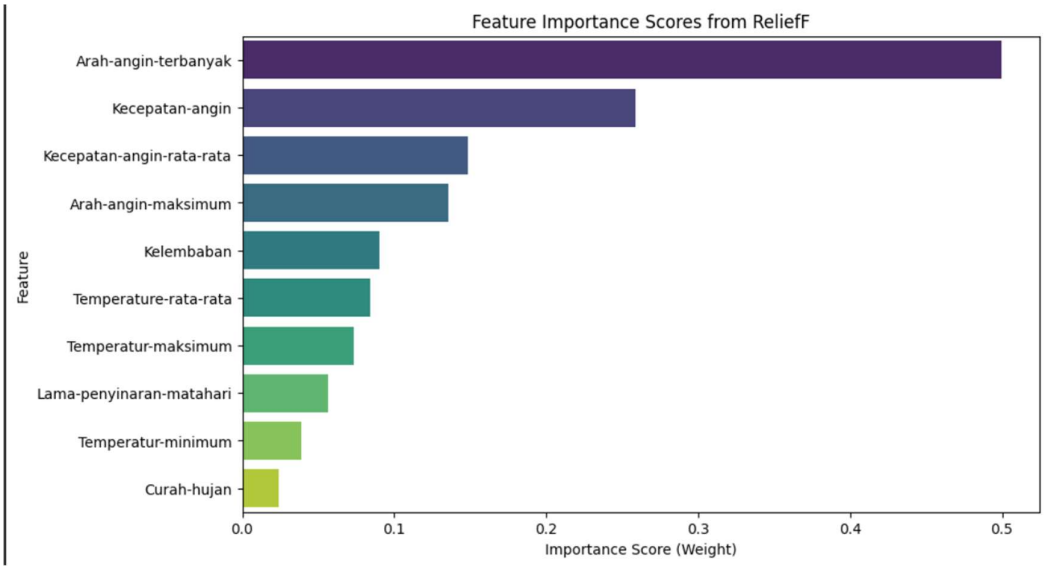
3.1.5. Hasil Permodelan KNN + *Relief*

Pemodelan ini menggunakan algoritma KNN dan seleksi fitur menggunakan *Relief*. *Relief* sebagai metode seleksi fitur yang akan melakukan perangkingan fitur-fitur dalam dataset berdasarkan skor kepentingan *(importance score),* yang menunjukkan kemampuan fitur dalam membedakan antara kelas-kelas, di mana fitur dengan skor kepentingan yang lebih tinggi memiliki pengaruh yang besar terhadap hasil klasifikasi. Sebaliknya, fitur dengan skor kepentingan yang rendah memiliki pengaruh yang lebih kecil terhadap klasifikasi dan mungkin kurang relevan untuk model tersebut.

```
Fitur terbaik beserta bobotnya:
                    Feature    Weight
0       Arah-angin-terbanyak  0.499401
1            Kecepatan-angin  0.259026
2  Kecepatan-angin-rata-rata  0.148464
3        Arah-angin-maksimum  0.135455
4                 Kelembaban  0.090396
5      Temperature-rata-rata  0.084353
6         Temperatur-maksimum  0.073162
7   Lama-penyinaran-matahari  0.056167
8         Temperatur-minimum  0.039113
9                 Curah-hujan  0.024114
```

Gambar 3. 9 Hasil Perangkingan *relief* berdasarkan *(importance score)*

Pada Gambar 3.10 menjelaskan Grafik setiap fitur berdasarkan skor kepentingan (*importance score)*
dari Fitur yang memiliki Skor terbanyak hingga terendah.



Gambar 3. 10 Grafik *scores* dari *Relief*

Implementasi *Relief* dilakukan melalui pendekatan bahasa pemrograman *Python*. Berdasarkan Gambar 3.9 maka dilakukan penentuan atribut yang akan digunakan yang dapat dilihat pada tabel 3.8.

Tabel 3. 8 Penentuan Atribut yang Digunakan

No	Atribut	Skor	Hasil
1.	Arah-angin-terbanyak	0.499401	Digunakan
2.	Kecepatan-angin	0.259026	Digunakan
3.	Kecepatan-angin-rata-rata	0.148464	Digunakan
4.	Arah-angin-maksimum	0.135455	Digunakan
5.	Kelembapan	0.090396	Tidak Digunakan
6.	Temperature-rata-rata	0.084353	Tidak Digunakan
7.	Temperature-maksimum	0.073162	Tidak Digunakan
8.	Lama-penyiraman-matahari	0.056167	Tidak Digunakan
9.	Temperature-minimum	0.039113	Tidak Digunakan
10.	Curah-hujan	0.024114	Tidak Digunakan

Dapat dilihat pada tabel 3.8 dari 10 atribut, yang akan di ambil adalah 4 atribut dengan nilai Skor tertinggi. 4 artibut tersebut yaitu Arah-angin-terbanyak, kecepatan-angin, Kecepatan-angin-rata-rata dan Arah-angin-maksimum.

Setelah itu, dilakukan evaluasi ulang menggunakan *Confusion Matrix* dan melihat kembali hasil akurasi rata-rata pada tiap nilai K dengan pembagian data yang sama pada nilai K nya, yaitu 10-*fold* untuk melihat apakah ada perbedaan akurasi setelah melakukan seleksi fitur. Hasil evaluasi *Confusion Matrix* dapat dilihat pada tabel 3.9

Tabel 3. 9 Evaluasi *Confusion Matrix* Setelah Seleksi Fitur

Nilai K	TN	FP	FN	TP	*Mean Accuracy*
K=3	74	5.6	9.8	78	89.59%
K=5	72	6.1	79	12	87.69%
K=7	70	6.4	78	14	85.67%
K=9	69	6.9	77	16	84.90%
K=11	68	8.1	76	16	82.64%
K=13	**66**	9.8	74	18	81.09%
K=15	65	11	73	19	80.20%

Pada Tabel 3.9, terdapat perubahan dalam hal peningkatan akurasi rata-rata pada masing-masing nilai K pada keseluruhan fold setelah menerapkan seleksi fitur *Relief.*

3.1.6. Hasil Permodelan KNN + *Relief* + PSO

Pada pemodelan ini, akan menerapkan optimasi PSO terhadap model klasifikasi untuk meningkatkan performa model KNN yang telah diintegrasikan dengan seleksi fitur *Relief.* Optimasi ini mencakup penyesuaian berbagai parameter, termasuk jumlah tetangga terdekat (*n_neighbors*), ukuran *leaf (leaf_size),* dan parameter jarak (p) dalam algoritma KNN.

Tabel 3. 10 Evaluasi *KNN+ Relief + PSO*

Nilai K	TN	FP	FN	TP	*Mean Accuracy*
K=3	66	1.9	18	82	90.84%
K=5	62	2.6	22	82	89.95%
K=7	60	3.2	24	81	87.75%
K=9	58	3.6	26	80	86.68%
K=11	57	4.4	27	80	85.55%
K=13	56	4.9	28	79	83.23%
K=15	54	5.1	30	79	82.11%

Pada Tabel 3.1 penerapan KNN dengan seleksi fitur *Relief* dan optimasi PSO mengalami peningkatan akurasi rata-rata pada masing-masing nilai K pada keseluruhan *fold.*

3.1.7. Perbandingan Hasil

Berikut adalah perbandingan hasil dari berbagai model klasifikasi yang telah diterapkan dalam penelitian ini. Penelitian ini mengevaluasi performa model *k-nearest neighbors* (KNN) dalam beberapa konfigurasi: KNN dasar tanpa optimasi, KNN yang dioptimalkan dengan *Particle Swarm Optimization* (PSO), KNN yang menggunakan seleksi fitur dengan algoritma *Relief,* dan kombinasi KNN yang menggunakan seleksi fitur *Relief* serta optimasi PSO. Pada tabel 3.11 akan menyajikan hasil akurasi dari masing-masing model untuk memberikan gambaran yang jelas tentang efektivitas setiap pendekatan dalam meningkatkan kinerja klasifikasi.

Tabel 3. 11 Perbandingan hasil akurasi dari setiap model KNN

Nilai K	***KNN***	**KNN+PSO**	**Status**	***KNN+Relief***	**status**	**KNN+Relief+PSO**	**Status**
K=3	88,23%	90,85%	+2.62%	89.59%	1,36%	90.84%	2,61%
K=5	85,38%	89,42%	+4.04%	87.69%	2,31%	89.95%	4,57%
K=7	84,07%	87,70%	+3,63%	85.67%	1,60%	87.75%	3,68%
K=9	82,52%	86,50%	+ 3.98%	84.90%	2,38%	86.68%	4,16%
K=11	81,27%	86,39	+ 5.12%	82.64%	1,37%	85.55%	4,28%
K=13	80,62%	85,55%	+ 4.93%	81.09%	0,47%	83.23%	2,61%
K=15	79,31%	84,84%	+ 5.53%	80.20%	0,89%	82.11%	2,80%

Dalam penelitian ini, berbagai metode optimasi diterapkan pada algoritma KNN untuk meningkatkan performa model. Hasil perbandingan akurasi pada table 3.11 menunjukkan bahwa pada K=3,k=5,k=7,k=9,k=11,k=13,k=15 Secara keseluruhan, model KNN dasar mengalami peningkatan akurasi setelah dioptimasi dengan PSO, dibandingkan dengan penggunaan seleksi fitur *Relief* atau kombinasi keduanya. Peningkatan terbesar terlihat pada KNN+PSO, terutama pada nilai K yang lebih tinggi.

3.2. Pembahasan

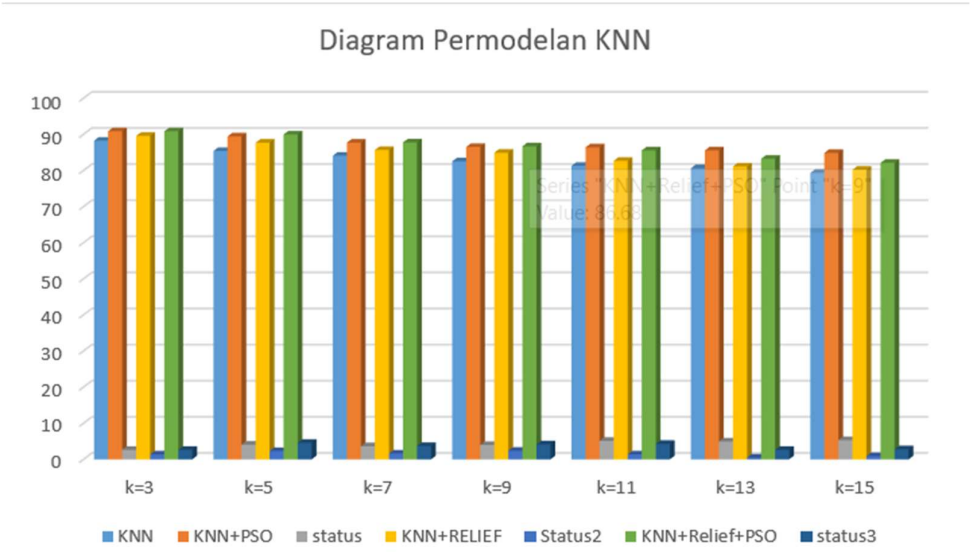
Penelitian ini menggunakan data banjir Kota Samarinda yang diperoleh dari BPBD dan BMKG untuk periode 2021-2023. Data yang terkumpul kemudian melalui beberapa tahapan pengolahan, termasuk pengumpulan data, persiapan data (data *preparation*), transformasi data (data *transformation*), pembersihan data (data *cleaning*), penyeimbangan data (data *balancing*), pembagian data, dan

pemodelan yang melibatkan beberapa model klasifikasi untuk mengevaluasi hasilnya. Teknik *oversampling* SMOTE digunakan untuk penyeimbangan data, dan pembagian data training dan testing dilakukan menggunakan K-*fold Cross Validation* dengan K=10.

Setelah dilakukan Analisis, menunjukkan bahwa PSO secara signifikan meningkatkan akurasi algoritma KNN melalui optimasi parameter yang lebih tepat, sementara seleksi fitur dengan *Relief* juga memberikan peningkatan akurasi. Kombinasi *Relief* dan PSO memperlihatkan sinergi dalam pengoptimalan parameter dan pemilihan fitur yang relevan, menghasilkan peningkatan akurasi terbesar kedua setelah KNN dan PSO. Metode kombinasi ini menegaskan pentingnya penyesuaian parameter dan seleksi fitur untuk memperbaiki performa model secara signifikan, terutama pada nilai k yang lebih tinggi.

Peningkatan akurasi oleh PSO dapat dijelaskan oleh beberapa faktor utama. Pertama, optimasi parameter dengan PSO memungkinkan penyesuaian yang sangat tepat terhadap parameter model seperti jumlah tetangga terdekat (k), ukuran leaf, dan parameter jarak, dengan menggunakan pendekatan berbasis populasi untuk eksplorasi ruang parameter secara efektif. Kedua, PSO membantu mengatasi overfitting dengan mencari keseimbangan optimal antara bias dan varians, sehingga model dapat menangkap pola yang lebih relevan dalam data tanpa terlalu terpengaruh oleh noise. Ketiga, Meskipun seleksi fitur Relief meningkatkan akurasi dengan memilih fitur yang paling relevan dan mengurangi noise, sebagian peningkatannya tidak sebesar yang dicapai oleh PSO. Hal ini dikarenakan Relief hanya memilih fitur tanpa menyesuaikan parameter model, sementara PSO memberikan manfaat lebih efektif melalui optimasi parameter langsung. kombinasi antara seleksi fitur Relief dan optimasi PSO tetap memberikan manfaat, namun tidak selalu lebih baik daripada optimasi parameter langsung dengan PSO.

Dengan demikian, kemampuan PSO dalam menyesuaikan parameter model dan mengatasi *overfitting* menjelaskan mengapa model KNN yang dioptimalkan dengan PSO secara konsisten memberikan peningkatan akurasi tertinggi dibandingkan model KNN dasar, model KNN dengan seleksi fitur *Relief*, dan model KNN dengan kombinasi seleksi fitur *Relief* dan PSO.



Gambar 3. 11 Diagram Permodelan KNN

3.2.1. Fitur Terpilih

Dalam penelitian ini, seleksi fitur menggunakan algoritma *Relief* diterapkan pada data banjir Kota Samarinda untuk meningkatkan performa model klasifikasi *k-Nearest Neighbors* (KNN), dengan fitur-fitur terpilih meliputi arah-angin-terbanyak (0.499401), kecepatan-angin (0.259026), kecepatan-angin-rata-rata (0.148464), dan arah-angin-maksimum (0.135455). Pada model yang menggabungkan seleksi fitur dan optimasi parameter menunjukkan performa prediksi banjir yang lebih baik. Peningkatan terbesar pada model KNN dengan seleksi fitur *Relief* terlihat pada K=9, dengan akurasi meningkat dari 82.52% menjadi 84,90 %, mengindikasikan bahwa pemilihan fitur yang relevan berhasil meningkatkan performa model klasifikasi dalam memprediksi kejadian banjir di Kota Samarinda.

Penelitian sebelumnya juga mendukung hasil ini, seperti penelitian oleh (Intan & Sari, 2023) yang menggunakan seleksi fitur *gain ratio* mengidentifikasi 'Kelembaban', 'Temperatur-minimum', dan 'Temperatur-maksimum' sebagai fitur paling berpengaruh, menunjukkan peningkatan akurasi algoritma klasifikasi KNN dengan peningkatan sebesar 5.95%. Sementara itu, penelitian oleh (Dilla Evitasari et al., 2023) yang menerapkan seleksi fitur menggunakan *algoritma Genetik* (GA) pada data banjir Kota Samarinda menemukan bahwa 'Kelembapan', 'Lama-penyinaran-matahari', dan 'Kecepatan-angin' memiliki pengaruh paling signifikan terhadap prediksi banjir, meningkatkan akurasi algoritma klasifikasi SVM sebesar 13.45%.

3.2.2. Peningkatan Akurasi

Dalam penelitian ini, algoritma KNN diterapkan untuk klasifikasi data banjir dengan menggunakan metode *oversampling* SMOTE dalam proses pra-pemrosesan data. Hasil menunjukkan bahwa model dasar KNN mencapai berbagai tingkat akurasi pada nilai k yang berbeda, dengan peningkatan akurasi yang signifikan setelah menerapkan seleksi fitur *Relief* dan optimasi PSO. KNN PSO memberikan peningkatan sebesar 2-5%, dengan seleksi fitur *Relief* peningkatan sebesar 1-2% dan KNN dengan kombinasi *Relief* dan PSO peningkatan sebesar 2-5%. Penggunaan algoritma KNN dalam klasifikasi data banjir yang menggunakan metode *oversampling* SMOTE tanpa seleksi fitur dan optimasi menunjukkan peningkatan akurasi yang signifikan setelah penerapan optimasi. Model KNN yang menggabungkan metode *oversampling* SMOTE dan seleksi fitur *Relief* juga memperlihatkan peningkatan akurasi yang berarti setelah optimasi. Namun, model KNN dengan seleksi fitur memiliki kecepatan pemrosesan yang lebih cepat, meskipun peningkatan akurasinya tidak sebesar yang dicapai oleh optimasi PSO. Hal ini mengindikasikan bahwa PSO memberikan pengaturan parameter yang lebih tepat dan peningkatan performa keseluruhan model dibandingkan dengan model yang hanya menggunakan seleksi fitur *Relief.* Hasil Penelitian ini ditujukan kepada Badan Penanggulangan Bencana Daerah (BPBD), Dinas Pekerjaan Umum dan Penataan Ruang (PUPR), Pemerintah Daerah Kota Samarinda, serta institusi penelitian dan akademik yang berkontribusi terhadap pengembangan ilmu pengetahuan dan teknologi untuk penanggulangan bencana. Diharapkan bahwa hasil dari penilitian ini dapat memberikan informasi yang berguna dan pengetahuan yang bermanfaat yang berguna untuk pengambilan keputusan memprediksi bencana banjir sehingga dapat membuat keputusan yang lebih tepat dalam meningkatkan kesiapsiagaan dan respons terhadap banjir, serta membantu dalam perencanaan dan mitigasi bencana banjir di Kota Samarinda.

BAB IV
KESIMPULAN DAN SARAN

4.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa poin penting sebagai berikut:

a. Hasil dari penerapan seleksi fitur Relief yang diterapkan pada data banjir Kota Samarinda mengidentifikasi empat fitur dengan pengaruh signifikan berdasarkan peringkatnya, yaitu Arah-angin-terbanyak, Kecepatan-angin, Kecepatan-angin-rata-rata, dan Arah-angin-maksimum.
b. Setelah melalui tahap seleksi fitur dan penerapan PSO, disimpulkan bahwa penerapan *Relief* efektif dalam meningkatkan akurasi dari algoritma *K-Nearest Neighbor* pada data banjir di Kota Samarinda dengan hasil akurasi KNN PSO memberikan peningkatan sebesar 2-5%, dengan seleksi fitur *Relief* peningkatan sebesar 1-2% dan KNN dengan kombinasi *Relief* dan PSO peningkatan sebesar 2-5%.

4.2. Saran

Saran untuk penelitian selanjutnya, diantaranya:

a. Meskipun seleksi fitur *Relief* telah memberikan peningkatan akurasi yang berarti, penelitian selanjutnya dapat mengeksplorasi teknik seleksi fitur lainnya dengan menggunakan metode lain pada algoritma *K-Nearest Neighbor* untuk data nilai Banjir di kota Samarinda. seperti ANOVA (*Analysis of Variance*), *Adaboost, Chi-square, Recursive Feature Elimination (RFE*), *Principal Component Analysis (PCA*) dan metode lainnya yang relevan dan informatif, sehingga dapat lebih meningkatkan akurasi dan efisiensi model klasifikasi KNN.
b. Pada penelitian selanjutnya, diharapkan untuk melakukan Melakukan studi komparatif dengan algoritma klasifikasi lain seperti *Random Forest, Support Vector Machine (SVM), Naïve Bayes, dan Decision Tree* untuk melihat bagaimana kombinasi seleksi fitur dan optimasi parameter dapat diterapkan pada algoritma lain dan membandingkan hasilnya.

DAFTAR RUJUKAN

Abdulrazaq, M. B., Mahmood, M. R., Zeebaree, S. R. M., Abdulwahab, M. H., Zebari, R. R., & Sallow, A. B. (2021). An Analytical Appraisal for Supervised Classifiers' Performance on Facial Expression Recognition Based on Relief-F Feature Selection. *Journal of Physics: Conference Series*, *1804*(1). https://doi.org/10.1088/1742-6596/1804/1/012055

Ariyoga, D. (2022). *Perbandingan Metode Seleksi Fitur Filter, Wrapper, Dan Embedded Pada Klasifikasi Data Nirs Mangga Menggunakan Random Forest Dan Support Vector Machine …*. https://dspace.uii.ac.id/handle/123456789/38955

Arora, A., Arabameri, A., Pandey, M., Siddiqui, M. A., Shukla, U. K., Bui, D. T., Mishra, V. N., & Bhardwaj, A. (2021). Optimization of state-of-the-art fuzzy-metaheuristic ANFIS-based machine learning models for flood susceptibility prediction mapping in the Middle Ganga Plain, India. *Science of the Total Environment*, *750*(August). https://doi.org/10.1016/j.scitotenv.2020.141565

Cumel, David Zamri, Rahmaddeni, S. (2022). Perbandingan Metode Data Mining untuk Prediksi Banjir Dengan Algoritma Naïve Bayes dan KNN. *SENTIMAS: Seminar Nasional Penelitian Dan …*, 40–48. https://journal.irpi.or.id/index.php/sentimas/article/view/353%0Ahttps://journal.irpi.or.id/index.php/sentimas/article/download/353/132

Daniel, I., Hartono, H., & Situmorang, Z. (2023). Analysis of Machine Learning Algorithms in Predicting the Flood Status of Jakarta City. *International Conference on Information Science and Technology Innovation (ICoSTEC)*, *2*(1), 82–87. https://doi.org/10.35842/icostec.v2i1.42

Databoks. (2023). *BNPB: Tren Banjir di Indonesia Cenderung Menurun dalam Tiga Tahun Terakhir*. https://databoks.katadata.co.id/datapublish/2023/02/20/bnpb-tren-banjir-di-indonesia-cenderung-menurun-dalam-tiga-tahun-terakhir

Dilla Evitasari, Y., Pranoto, W. J., & Adzmi Verdikha, N. (2023). EVALUASI SUPPORT VECTOR MACHINE DENGAN OPTIMASI METODE GENETIC ALGORITHM PADA KLASIFIKASI BANJIR KOTA SAMARINDA Evaluation Support Vector Machine With Optimization Genetic Algorithm Method On Flood Classification In Samarinda. *Jurnal Sains Komputer Dan Teknologi Informasi*, *6*(1), 49–53.

Dwi Astuti, F., & Nova Lenti, F. (2021). Implementasi SMOTE untuk mengatasi Imbalance Class pada Klasifikasi Car Evolution menggunakan K-NN. *Jurnal JUPITER*, *13*(1), 89–98.

Dwiasnati, S., & Yudo Devianto. (2022). Optimization of Flood Prediction using SVM Algorithm to determine Flood Prone Areas. *Journal of Systems Engineering and Information Technology (JOSEIT)*, *1*(2), 40–46. https://doi.org/10.29207/joseit.v1i2.1995

Faldi, F., NurHalisha, T., Pranoto, W. J., & ... (2023). The application of particle swarm optimization (PSO) to improve the accuracy of the naive bayes algorithm in predicting floods in the city of Samarinda. *Journal of Intelligent …*, *6*(3), 138–146. http://idss.iocspublisher.org/index.php/jidss/article/view/148%0Ahttps://idss.iocspublisher.org/index.php/jidss/article/download/148/99

Farhan, N. M., & Setiaji, B. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, *12*(2), 284–301. http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135

Gauhar, N., Das, S., & Moury, K. S. (2021). Prediction of Flood in Bangladesh using k-Nearest Neighbors Algorithm. *International Conference on Robotics, Electrical and Signal Processing Techniques*, 357–361. https://doi.org/10.1109/ICREST51555.2021.9331199

Hakimah, M., Prabiantissa, C. N., Rozi, N. F., Yamani, L. N., & Puspitasari, I. (2022). Determination of Relevant Feature Combinations for Detection Stunting Status of Toddlers. *2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022*, 324–329. https://doi.org/10.1109/ISRITI56927.2022.10053069

Hossain, M. S., & Zeyad, M. (2023). Prediction of Flood in Bangladesh Using Different Classifier Model. *AIUB Journal of Science and Engineering*, *22*(1), 45–52. https://doi.org/10.53799/ajse.v22i1.365

Intan, S., & Sari, P. (2023). ANALISIS PENGARUH GAIN RATIO UNTUK ALGORITMA K-NEAREST

NEIGHBOR PADA KLASIFIKASI DATA BANJIR DI KOTA SAMARINDA Analysis Of The Effect Of Gain Ratio For Algorithms K-Nearest Neighbor On Classsification Flood Data In Samarinda City. *Jurnal Sains Komputer Dan …*, *6*(1), 54–59. https://journal.umpr.ac.id/index.php/jsakti/article/view/5472%0Ahttps://journal.umpr.ac.id/index.php/jsakti/article/download/5472/3664

Kemal Musthafa Rajabi, Witanti, W., & Rezki Yuniarti. (2023). Penerapan Algoritma K-Nearest Neighbor (KNN) Dengan Fitur Relief-F Dalam Penentuan Status Stunting. *INNOVATIVE: Journal Of Social Science Research*, *3*, 3555–3568.

Members, E. B., Parman, D. H., Tarakan, U. B., Saputro, H., Fahrizal, Y., Yogyakarta, U. M., & Roseshita, R. D. (2021). Jurnal Bencana 2021. *Journal of Community Engagement in Health*, *4*(Peningkatan Pengetahuan Siswa Terhadap Mitigasi Bencana di SD Muhammadiyah 4 Samarinda), 393–399.

Mian, T. S., & Ghabban, F. (2022). Competitive Advantage: A Study of Saudi SMEs to Adopt Data Mining for Effective Decision Making. *Journal of Data Analysis and Information Processing*, *10*(03), 155–169. https://doi.org/10.4236/jdaip.2022.103010

Nabila, S. P., Ulinnuha, N., Yusuf, A., Informasi, S., Wonosari, J., & Timur, J. (2021). *MODEL PREDIKSI KELULUSAN TEPAT WAKTU DENGAN METODE FUZZY C-MEANS DAN K-NEAREST NEIGHBORS*. *6*(1), 39–47.

Nawi, N. M., Makhtar, M., Salikon, M. Z., & Afip, Z. A. (2020). A comparative analysis of classification techniques on predicting flood risk. *Indonesian Journal of Electrical Engineering and Computer Science*, *18*(3), 1342–1350. https://doi.org/10.11591/ijeecs.v18.i3.pp1342-1350

Nursyahfitri, R., Rozikin, C., & Adam, R. I. (2022). Penerapan Metode SMOTE dalam Klasifikasi Daerah Rawan Banjir di Karawang Menggunakan Algoritma Naive Bayes. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, *10*(4), 339. https://doi.org/10.26418/justin.v10i4.46935

Priscillia, S., Schillaci, C., & Lipani, A. (2022). Arti fi cial Intelligence in Geosciences Flood susceptibility assessment using arti fi cial neural networks in Indonesia. *Artificial Intelligence in Geosciences*, *2*(April), 215–222.

Purwanto, P. (2020). Analisis Sistem Pengendalian Banjir Sungai Pampang Daerah Aliran Hulu Sungai Karangmumus. *Jurnal Kacapuri : Jurnal Keilmuan Teknik Sipil*, *3*(2), 44. https://doi.org/10.31602/jk.v3i2.4066

Razali, N., Ismail, S., & Mustapha, A. (2020). Machine learning approach for flood risks prediction. *IAES International Journal of Artificial Intelligence*, *9*(1), 73–80. https://doi.org/10.11591/ijai.v9.i1.pp73-80

Sitepu, R., & Manohar, M. (2022). Implementasi Algoritma K-Nearest Neigbor Untuk Klasifikasi Pengajuan Kredit. *Jurnal Sistem Informasi, Teknik Informatika Dan Teknologi Pendidikan*, *1*(2), 49–56. https://doi.org/10.55338/justikpen.v1i2.6

Sopiatul Ulum, Alifa, R. F., Rizkika, P., & Rozikin, C. (2023). Perbandingan Performa Algoritma KNN dan SVM dalam Klasifikasi Kelayakan Air Minum. *Generation Journal*, *7*(2), 141–146. https://doi.org/10.29407/gj.v7i2.20270

Tarasova, L., Merz, R., Kiss, A., Basso, S., Blöschl, G., Merz, B., Viglione, A., Plötner, S., Guse, B., Schumann, A., Fischer, S., Ahrens, B., Anwar, F., Bárdossy, A., Bühler, P., Haberlandt, U., Kreibich, H., Krug, A., Lun, D., … Wietzke, L. (2019). Causative classification of river flood events. *Wiley Interdisciplinary Reviews: Water*, *6*(4), 1–23. https://doi.org/10.1002/wat2.1353

Tarigan, P. M. S., Hardinata, J. T., Qurniawan, H., Safii, M., & Winanjaya, R. (2022). Implementasi Data Mining Menggunakan Algoritma Apriori Dalam Menentukan Persediaan Barang. *Jurnal Janitra Informatika Dan Sistem Informasi*, *2*(1), 9–19. https://doi.org/10.25008/janitra.v2i1.142

Vafakhah, M., Mohammad Hasani Loor, S., Pourghasemi, H., & Katebikord, A. (2020). Comparing performance of random forest and adaptive neuro-fuzzy inference system data mining models for flood susceptibility mapping. *Arabian Journal of Geosciences*, *13*(11), 1–16. https://doi.org/10.1007/s12517-020-05363-1

Yahdin, S., Desiani, A., Gofar, N., & Agustin, K. (2021). Application of the Relief-f Algorithm for Feature Selection in the Prediction of the Relevance Education Background with the Graduate Employment of the Universitas Sriwijaya. *Computer Engineering and Applications Journal*, *10*(2), 71–80.

https://doi.org/10.18495/comengapp.
v10i2.369

Yoga Siswa, T. A. (2023). *DATA MINING: MENGUPAS TUNTAS ANALISIS DATA DENGAN METODE KLASIFIKASI HINGGA DEPLOYMENT APLIKASI MENGGUNAKAN PYTHON* (T. A. Yoga Siswa (ed.)). UMKT PRESS.

Yoga, T. A., & Prihandoko. (2018). Penerapan Optimasi Berbasis Particle Swarm Optimization (Pso) Algoritma Naïve Bayes Dan K-Nearest Neighbor Sebagai Perbandingan Untuk Mencari Kinerja Terbaik Dalam Mendeteksi Kanker Payudara. *Jurnal Bangkit Indonesia*, *7*(2), 1. http://journal.universitasmulia.ac.id/index.php/metik/
article/view/62

Yusra, R. N., Sitompul, O. S., & Sawaluddin. (2021). Kombinasi K-Nearest Neighbor (KNN) dan Relief-F Untuk Meningkatkan Akurasi Pada Klasifikasi Data. *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan*, *1*, 0–5.

DAFTAR RIWAYAT HIDUP

Saya Anggiq Karisma Aji Restu, juga dikenal sebagai Anggiq. Saya lahir pada tanggal 3 Juli 2000 di Banyuwangi, Jawa Timur. Saya adalah anak pertama dari pasangan Agus Triyono dan Natalisa Dwi Kristiani. belajar di SDN 004 Bontang dari 2007 hingga 2013, SMPN 4 Bontang dari 2013 hingga 2016, dan SMKN 3 Jurusan Geologi Pertambangan dari 2016 hingga 2019. Sebelumnya Pada tahun 2019–2020, penulis pernah magang di BPN(Badan Pertanahan Nasional) Kota Bontang selama 3 bulan dan pernah bekerja posisi maintenance di salah satu perusahaan BUMN yaitu Pertamina dan Gas Kota Bontang selama 6 bulan. Pada tahun 2020, penulis menjadi mahasiswa Universitas Muhammadiyah Kalimantan Timur, Fakultas Sains dan Teknologi, jurusan Teknik Informatika. Saat menjadi mahasiswa, penulis juga melakukan magang selama 3 bulan di Disdukcapil dilaksanakan pada semester 7. Demikian deskripsi riwayat hidup yang penulis sampaikan jika terdapat kesalahan atau kekurangan mohon dimaafkan karena kesempurnaan hanya milik Sang Maha pencipta, maka penulis mengharapkan kritik dan saran mengenai skripsi ini.

LAMPIRAN

Lampiran 1 Codingan

- Mengimport modul yang dibutuhkan

```
from pyswarm import pso
from sklearn.neighbors import KNeighborsClassifier
from skrebate import ReliefF
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import cross_val_score,KFold
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score, f1_score, recall_score
from sklearn.metrics import precision_score, classification_report, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd
```

- Mengimport file csv

```
#Melakukan import terhadap dataset banjir
data = pd.read_csv('banjir.csv')
df = pd.DataFrame(data)
```

- Membuat model KNN dan melakukan 10-fold, serta confusion matrix

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import confusion_matrix, accuracy_score, classification_repo
import numpy as np
import pandas as pd

# Asumsi: X_res dan y_res telah didefinisikan sebelumnya
# Membuat array k dengan nilai k dari 3 sampai 15
k_values = np.arange(3, 16)

# Variable untuk menyimpan hasil akurasi dan confusion matrices
all_fold_scores = {}
all_confusion_matrices = {}

# K-fold cross validation
kf = KFold(n_splits=10, shuffle=True, random_state=42)

for k in k_values:
    print(f"\nk = {k}")

    fold_scores = []
    fold_confusion_matrices = []
    kFold = 1

    for train_index, test_index in kf.split(X_res, y_res):
        X_train, X_test = X_res.iloc[train_index], X_res.iloc[test_index]
        y_train, y_test = y_res.iloc[train_index], y_res.iloc[test_index]
```

```
        # Inisiasi algoritma k-nearest neighbors
        knn = KNeighborsClassifier(n_neighbors=k, metric='euclidean')

        # Melatih model KNN
        knn.fit(X_train, y_train)

        # Predict on the test set
        y_pred = knn.predict(X_test)

        # Mencari nilai confusion matrix
        cMetrics = confusion_matrix(y_test, y_pred)
        fold_confusion_matrices.append(cMetrics)
        print(f"Fold {kFold} : Confusion Matrix")
        print(cMetrics)

        # Mendapatkan nilai akurasi
        fold_accuracy = accuracy_score(y_test, y_pred)
        fold_scores.append(fold_accuracy)

        # Menampilkan nilai akurasi
        print(f"Fold {kFold}: Accuracy: {fold_accuracy:.2f}")

        # Report
        print(f"{kFold} Report\n{classification_report(y_test, y_pred)}")

        # Update nilai kFold
        kFold += 1

    # Menyimpan hasil rata-rata akurasi dan confusion matrices untuk setiap nilai k
    all_fold_scores[k] = np.mean(fold_scores)
    all_confusion_matrices[k] = fold_confusion_matrices

# Menampilkan hasil akurasi untuk setiap nilai k
for k, avg_accuracy in all_fold_scores.items():
    print(f"k = {k}: Rata-rata Akurasi = {avg_accuracy * 100:.2f}%")
```

```
k = 3
Fold 1 : Confusion Matrix
[[68 17]
```

● Membuat fungsi untuk melakukan perhitungan Relief

```python
# Import pustaka yang diperlukan
from skrebate import ReliefF
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Inisiasi ReliefF dengan parameter yang sesuai
relief = ReliefF(n_neighbors=15)  # Parameter n_neighbors dapat disesuaikan

# Melakukan fit transform pada data
X_transformed = relief.fit_transform(X.values, y.values)

# Mendapatkan indeks fitur yang dipilih berdasarkan ranking
selected_features_indices = relief.top_features_

# Mendapatkan nama fitur yang dipilih
selected_features_names = X.columns[selected_features_indices]

# Mendapatkan bobot fitur
feature_weights = relief.feature_importances_[selected_features_indices]

# Membuat DataFrame dengan nama fitur dan bobotnya
feature_importances = pd.DataFrame({
    'Feature': selected_features_names,
    'Weight': feature_weights
})

# Mengurutkan fitur berdasarkan bobot
feature_importances = feature_importances.sort_values(by='Weight', ascending=False)

# Menampilkan fitur terbaik beserta bobotnya
print("Fitur terbaik beserta bobotnya:")
print(feature_importances)

# Membuat grafik bobot fitur
plt.figure(figsize=(10, 6))
sns.barplot(data=feature_importances, x='Weight', y='Feature', palette='viridis')
plt.title('Feature Importance Scores from ReliefF')
plt.xlabel('Importance Score (Weight)')
plt.ylabel('Feature')
plt.show()
```

- Membuat ulang dataframe serta penerapan ulang KNN, 10-fold, Confusion Matrix dan fitur terpilih

```python
# Fitur terbaik dari seleksi fitur ReliefF
top_features = ['Arah-angin-terbanyak', 'Kecepatan-angin', 'Kecepatan-angin-rata-rata', 'Arah-angin-maksimum']

# Memisahkan variabel atribut dan target berdasarkan fitur yang dipilih
X_res_selected = X_res[top_features]
y_res = y_res

# Membuat array k dengan nilai k dari 3 sampai 15
k_values = np.arange(3, 16)

# Variable untuk menyimpan hasil akurasi dan confusion matrices
all_fold_scores = {}
all_confusion_matrices = {}

# K-fold cross validation
kf = KFold(n_splits=10, shuffle=True, random_state=42)

for k in k_values:
    print(f"\nk = {k}")

    fold_scores = []
    fold_confusion_matrices = []
    kFold = 1

    for train_index, test_index in kf.split(X_res_selected, y_res):
        X_train, X_test = X_res_selected.iloc[train_index], X_res_selected.iloc[test_index]
        y_train, y_test = y_res.iloc[train_index], y_res.iloc[test_index]

        # Inisiasi algoritma k-nearest neighbors
        knn = KNeighborsClassifier(n_neighbors=k, metric='euclidean')

        # Melatih model KNN
        knn.fit(X_train, y_train)

        # Predict on the test set
        y_pred = knn.predict(X_test)

        # Mencari nilai confusion matrix
        cMetrics = confusion_matrix(y_test, y_pred)
        fold_confusion_matrices.append(cMetrics)
        print(f"Fold {kFold} : Confusion Matrix")
        print(cMetrics)

        # Mendapatkan nilai akurasi
        fold_accuracy = accuracy_score(y_test, y_pred)
        fold_scores.append(fold_accuracy)

        # Menampilkan nilai akurasi
        print(f"Fold {kFold}: Accuracy: {fold_accuracy:.2f}")

        # Report
        print(f"{kFold} Report\n{classification_report(y_test, y_pred)}")

        # Update nilai kFold
        kFold += 1

    # Menyimpan hasil rata-rata akurasi dan confusion matrices untuk setiap nilai k
    all_fold_scores[k] = np.mean(fold_scores)
    all_confusion_matrices[k] = fold_confusion_matrices

# Menampilkan hasil akurasi untuk setiap nilai k
for k, avg_accuracy in all_fold_scores.items():
    print(f"k = {k}: Rata-rata Akurasi = {avg_accuracy * 100:.2f}%")
```

• Penerapan PSO

```
# Function to optimize KNN
def optimize_knn(x, k):
    leaf_size = int(x[0])
    p = int(x[1])

    clf = KNeighborsClassifier(n_neighbors=k, leaf_size=leaf_size, p=p)
    scores = cross_val_score(clf, X_res, y_res, cv=10)

    return -scores.mean()  # Maximizing accuracy by minimizing the negative score

# Bounds for KNN hyperparameters
lb = [10, 1]  # Lower bounds for leaf_size and p
ub = [50, 2]  # Upper bounds for leaf_size and p

# List of odd k values to optimize
k_values = [3, 5, 7, 9, 11, 13, 15]

optimal_parameters = {}

for k in k_values:
    # Perform PSO optimization for each k
    xopt, fopt = pso(optimize_knn, lb, ub, args=(k,), swarmsize=50, maxiter=30)

    # Extract optimal parameters for this k
    optimal_leaf_size = int(xopt[0])
    optimal_p = int(xopt[1])
    optimal_score = -fopt

    # Store the optimal parameters and corresponding score for this k
    optimal_parameters[k] = {
        'optimal_leaf_size': optimal_leaf_size,
        'optimal_p': optimal_p,
        'optimal_score': optimal_score
    }

    # Print the optimal parameters and the corresponding score
    print(f"Optimal parameters for k = {k}:")
    print(f"Optimal leaf size: {optimal_leaf_size}")
    print(f"Optimal p: {optimal_p}")
    print(f"Optimal score: {optimal_score}")
    print("")
```

```
# Display all optimal parameters
for k, params in optimal_parameters.items():
    print(f"k = {k}:")
    print(f"  Optimal leaf size: {params['optimal_leaf_size']}")
    print(f"  Optimal p: {params['optimal_p']}")
    print(f"  Optimal score: {params['optimal_score']}")
```

- Membuat ulang dataframe serta penerapan PSO, 10-fold, Confusion Matrix dan fitur terpilih

```
# Memisahkan variabel atribut dan target
X_res = X_res
y_res = y_res

# Optimal parameters from PSO
optimal_k_values = [3, 5, 7, 9, 11, 13, 15]
optimal_leaf_sizes = [10, 34, 11, 49, 12, 21, 11]
optimal_p = 1

# Variable untuk menyimpan hasil akurasi dan confusion matrices untuk setiap nilai k
all_fold_scores = {}
all_confusion_matrices = {}

# K-fold cross validation
kf = KFold(n_splits=10, shuffle=True, random_state=42)

for i, k in enumerate(optimal_k_values):
    fold_scores = []
    fold_confusion_matrices = []
    leaf_size = optimal_leaf_sizes[i]

    kFold = 1
    for train_index, test_index in kf.split(X_res, y_res):
        X_train, X_test = X_res.iloc[train_index], X_res.iloc[test_index]
        y_train, y_test = y_res.iloc[train_index], y_res.iloc[test_index]

        # Inisiasi algoritma k-nearest neighbors dengan parameter hasil optimasi PSO
        knn = KNeighborsClassifier(n_neighbors=k, leaf_size=leaf_size, p=optimal_p, metric='minkowski')

        # Melatih model KNN
        knn.fit(X_train, y_train)

        # Predict on the test set
        y_pred = knn.predict(X_test)

        # Mencari nilai confusion matrix
        cMetrics = confusion_matrix(y_test, y_pred)
        fold_confusion_matrices.append(cMetrics)
        print(f"Fold {kFold}, k = {k} : Confusion Matrix")
        print(cMetrics)

        # Mendapatkan nilai akurasi
        fold_accuracy = accuracy_score(y_test, y_pred)
        fold_scores.append(fold_accuracy)

        # Menampilkan nilai akurasi
        print(f"Fold {kFold}, k = {k}: Accuracy: {fold_accuracy:.2f}")

        # Report
        print(f"Fold {kFold}, k = {k} Report\n{classification_report(y_test, y_pred)}")

        # Update nilai kFold
        kFold += 1

    # Menyimpan hasil rata-rata akurasi dan confusion matrices untuk setiap nilai k
    all_fold_scores[k] = np.mean(fold_scores)
    all_confusion_matrices[k] = fold_confusion_matrices

# Menampilkan hasil akurasi rata-rata untuk setiap nilai k
for k, avg_accuracy in all_fold_scores.items():
    print(f"k = {k}: Rata-rata Akurasi = {avg_accuracy * 100:.2f}%")
```

Lampiran 2 Surat Pengantar Pengambilan Data BMKG

UMKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Muhammadiyah Kalimantan Timur
Berkarakter | Berwawasan | Berkemajuan
Telp. 0541-748511 Fax. 0541-766832
Website http://informatika.umkt.ac.id
email: informatika@umkt.ac.id

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Nomor : 003-011/FST.1/A.7/C/2024
Lampiran : -
Perihal : **Permohonan Pengambilan Data**

Kepada Yth.
Kepala Badan Meteorologi, Klimatologi dan Geofisika (BMKG)
di -
Tempat

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Aamiin.

Sehubungan untuk memenuhi Tugas Akhir/Skripsi Tahun Akademik 2023/2024, maka dengan ini kami mengajukan permohonan untuk melakukan pengambilan data di Meteorologi, Klimatologi dan Geofisika (BMKG) Kota Samarinda. Adapun data yang diminta yaitu data parameter (temperatur maksimum, temperatur minimum, temperatur rata-rata, kelembaban rata-rata, curah hujan, lamanya penyinaran matahari, kecepatan angin maksimum, arah angin maksimum, kecepatan angin rata-rata, dan arah angin terbanyak), dengan nama mahasiswa sebagai berikut:

No	Nama	NIM	Program Studi
1	Raenald Syaputra	2011102441040	Teknik Informatika
2	Ilham Taufiq	2011102441152	Teknik Informatika
3	Anggiq Karisma Aji Restu	2011102441089	Teknik Informatika
4	Vito Junivan Rivaldo	2011102441019	Teknik Informatika
5	Faldy Alfareza Pambudi	2011102441097	Teknik Informatika

Demikian surat permohonan ini dibuat. Atas perhatian dan kerjasamanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 7 Ramadhan 1445 H
18 Maret 2024 M

Ketua Program Studi S1 Teknik Informatika

Fitriyah, S.Kom., M.TI
NIDN. 1118019203

Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda

Lampiran 3 Surat Pengantar Pengambilan Data BPBD

UMKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Muhammadiyah Kalimantan Timur
Berkarakter | Berwawasan | Berkemajuan

Telp. 0541-748511 Fax. 0541-766832
Website http://informatika.umkt.ac.id
email: informatika@umkt.ac.id

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Nomor : 003-008/FST.1/A.7/C/2024
Lampiran : -
Perihal : **Permohonan Pengambilan Data**

Kepada Yth.
Kepala Badan Penanggulangan Bencana Daerah Kota Samarinda
di -
Tempat

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Aamiin.

Sehubungan untuk memenuhi Tugas Akhir/Skripsi Tahun Akademik 2023/2024, maka dengan ini kami mengajukan permohonan untuk melakukan pengambilan data di Badan Penanggulangan Bencana Daerah Kota Samarinda. Adapun data yang diminta yaitu berupa data banjir tahun 2021-2023 Kota Samarinda, dengan nama mahasiswa sebagai berikut:

No	Nama	NIM	Program Studi
1	Raenald Syaputra	2011102441040	Teknik Informatika
2	Ilham Taufiq	2011102441152	Teknik Informatika
3	Anggiq Karisma Aji Restu	2011102441089	Teknik Informatika
4	Faldy Alfareza Pambudi	2011102441097	Teknik Informatika
5	Vito Junivan Rivaldo	2011102441019	Teknik Informatika

Demikian surat permohonan ini dibuat. Atas perhatian dan kerjasamanya kami mengucapkan terima kasih.

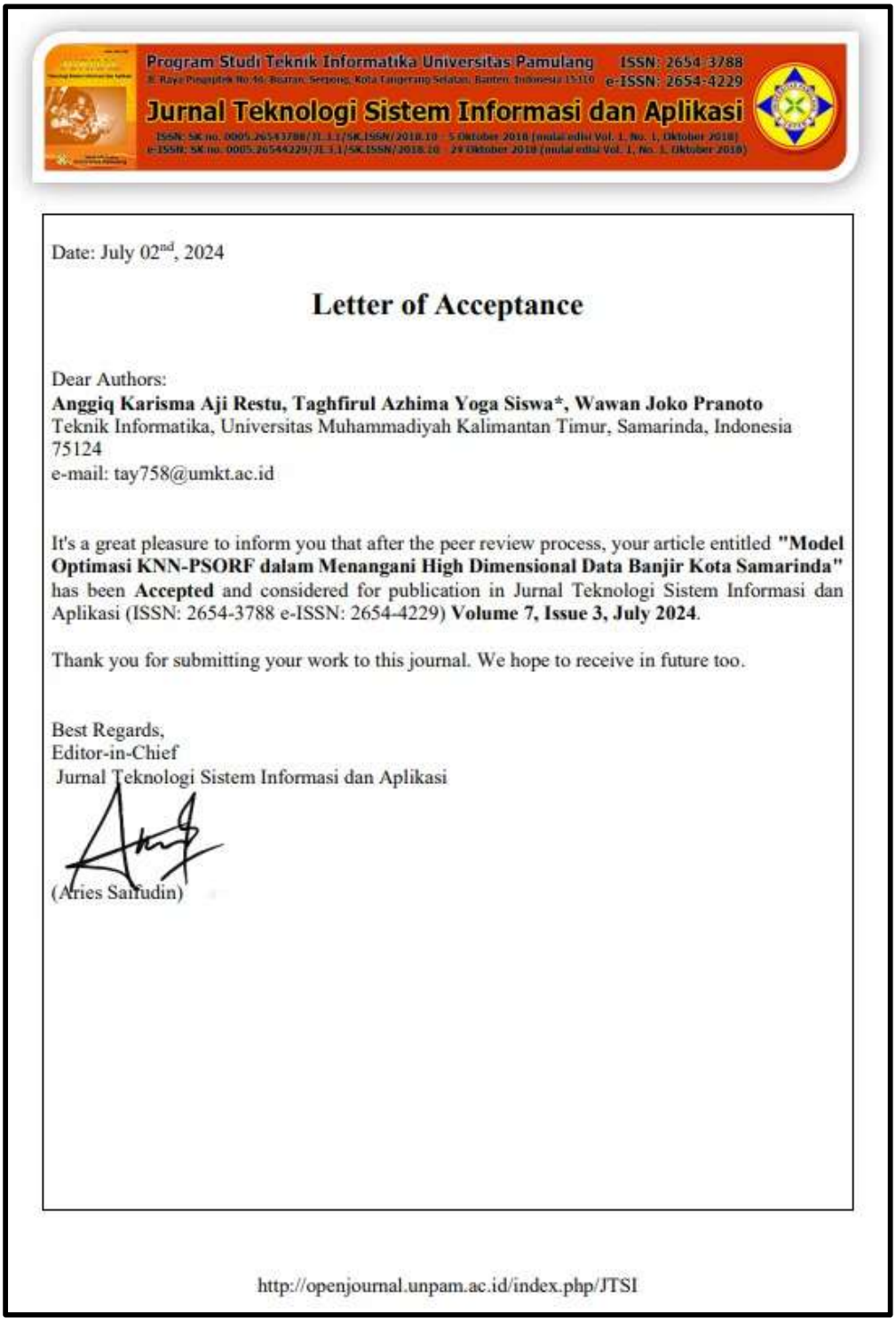
Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 7 Ramadhan 1445 H
18 Maret 2024 M

Ketua Program Studi S1 Teknik Informatika

Ansyah, S.Kom., M.TI
NIDN. 1118019203

Lampiran 4 Surat *Letter of Acceptance Jurnal*



Program Studi Teknik Informatika Universitas Pamulang ISSN: 2654-3788
e-ISSN: 2654-4229
Jurnal Teknologi Sistem Informasi dan Aplikasi

Date: July 02nd, 2024

Letter of Acceptance

Dear Authors:
Anggiq Karisma Aji Restu, Taghfirul Azhima Yoga Siswa*, Wawan Joko Pranoto
Teknik Informatika, Universitas Muhammadiyah Kalimantan Timur, Samarinda, Indonesia 75124
e-mail: tay758@umkt.ac.id

It's a great pleasure to inform you that after the peer review process, your article entitled **"Model Optimasi KNN-PSORF dalam Menangani High Dimensional Data Banjir Kota Samarinda"** has been **Accepted** and considered for publication in Jurnal Teknologi Sistem Informasi dan Aplikasi (ISSN: 2654-3788 e-ISSN: 2654-4229) **Volume 7, Issue 3, July 2024**.

Thank you for submitting your work to this journal. We hope to receive in future too.

Best Regards,
Editor-in-Chief
Jurnal Teknologi Sistem Informasi dan Aplikasi

(Aries Saifudin)

Lampiran 5 Lampiran Kartu bimbingan

KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama	:	Anggiq Karisma Aji Restu
NIM	:	2011102441089
Nama Dosen Pembimbing	:	Taghfirul Azhima Yoga Siswa, S.Kom, M.Kom
Judul Penelitian	:	Model Optimasi KNN-PSORF dalam menangani High Dimensional Data Banjir Kota Samarinda

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	22-01-2024	Pembahasan tahapan dalam Penelitian Skripsi	
2.	7-02-2024	Pertemuan Kedua, mencari Paper/artikel rujukan Penelitian	
3.	16-02-2024	Pertemuan Ketiga, Review survei Paper untuk mencari permasalahan pada klasifikasi data mining	
4.	17-02-2024	Pertemuan Keempat, Review Technical Paper dan road maps Penelitian	
5.	22-02-2024	Pertemuan Kelima, mencari Paper dan artikel sesuai rujukan terkait objek	
6	29-02-2024	Pertemuan Keenam, Penentuan judul Penelitian	
7	13-03-2024	Pertemuan Ketujuh, Pembuatan Canvas Penelitian	
8	15-03-2024	Pertemuan Kedelapan, Pengajuan surat Permohonan data untuk Penelitian	
9	18-03-2024	Pertemuan Kesembilan, Revisi Proposal bab 1, bab 2 dan Perbaikan Penulisan	
10	24,04-2024	Pertemuan Kesepuluh, Revisi Proposal bab 1, bab 2 dan Perbaikan Penulisan	

11.	10 Mei 2024	Pertemuan Kesebelas, bimbingan Pembahasan penulisan Bab 3 dan bab 4	
12.	17 mei 2024	Pertemuan kedua belas, Revisi naskah skripsi bab 3 dan bab 4	
13.	27 mei 2024	Konsultasi Perbaikan Penulisan hasil Penelitian dan artikel Jurnal	
14.	23-06-2024	Konsultasi dan Perbaikan Penyusunan Penulisan artikel Jurnal	

Dosen Pembimbing

Taghfirul Azhima Yoga Siswa, S.Kom, M.Kom
NIDN. 1118038805

Mengetahui
Ketua Program Studi

Irfansyah, S.Kom, M. TI.
NIDN.1118019203

SKRIPSI ANGGIQ KARISMA AJI RESTU

by Teknik Informatika Universitas Muhammadiyah Kalimantan Timur

Submission date: 12-Jul-2024 12:59PM (UTC+0800)
Submission ID: 2415566973
File name: Model_Optimasi_KNNPSORF_-_ANGGIQ_KARISMA_AJI_RESTU.docx (759.6K)
Word count: 8174
Character count: 52503

SKRIPSI ANGGIQ KARISMA AJI RESTU

ORIGINALITY REPORT

26%	22%	15%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.umkt.ac.id Internet Source	9%
2	Raenald Syaputra, Taghfirul Azhima Yoga Siswa, Wawan Joko Pranoto. "Model Optimasi SVM Dengan PSO-GA dan SMOTE Dalam Menangani High Dimensional dan Imbalance Data Banjir", Teknika, 2024 Publication	2%
3	Ari Ahmad Dhani, Taghfirul Azhima Yoga Siswa, Wawan Joko Pranoto. "Perbaikan Akurasi Random Forest Dengan ANOVA Dan SMOTE Pada Klasifikasi Data Stunting", Teknika, 2024 Publication	2%
4	repository.ub.ac.id Internet Source	2%
5	journal.umpr.ac.id Internet Source	1%
6	ejournal.ikado.ac.id Internet Source	1%