

EXPLORE

Jurnal Sistem Informasi & Telematika (Telekomunikasi, Multimedia & Informatika)

Indra Kurniawan, Ahmad Faiq Abror

KOMPARASI METODE KOMBINASI SELEKSI FITUR DAN MACHINE LEARNING K-NEAREST NEIGHBOR PADA DATASET LABEL HOURS SOFTWARE EFFORT ESTIMATION

Fenty Ariani, Arnes Yuli Vandika, Handy Widjaya

IMPLEMENTASI ALAT PEMBERI PAKAN TERNAK MENGGUNAKAN IOT UNTUK OTOMATISASI PEMBERIAN PAKAN TERNAK

Robby Yuli Endra, Ahmad Cneus, Freddy Nur Affandi, Deni Hermawan

IMPLEMENTASI SISTEM KONTROL BERBASIS WEB PADA SMART ROOM DENGAN MENGGUNAKAN KONSEP INTERNET OF THINGS

Tri Susilowati, Suepto, Nungsiyati, Tomi Adi Kartika, Nur Zaman

PENERAPAN METODE SIMPLE ADDITIVE WEIGHTING (SAW) PADA AMRI SUPERMARKET BANJAR JAYA UNTUK PEMILIHAN KARYAWAN TERBAIK

Erlangga, Taqwan Thamrin, Panji Maulana, Nico Susanto

BUS TRACKER – SISTEM LACAK LOKASI CALON PENUMPANG, LOKASI BUS DAN PERKIRAAN WAKTU KEDATANGAN BUS

Stephen, Raymond, Handri Santoso

APLIKASI CONVOLUTION NEURAL NETWORK UNTUK MENDETEKSI JENIS-JENIS SAMPAH

Freddy Nur Afandi, Ramses Parulian Sinaga, Yuthsi Aprilinda, Fenty Ariani

IMPLEMENTASI FACE DETECTION PADA SMART CONFERENCE MENGGUNAKAN VIOLA JONES

Dani Yusuf, Freddy Nur Afandi

APLIKASI MONITORING BASE TRANSCEIVER STATION BERBASIS ANDROID MENGGUNAKAN METODE LOCATION BASED SERVICE

Dede Aprilia Haspita, Jimi Ali Baba

DECISION SUPPORT SYSTEM(SISTEM PENDUKUNG KEPUTUSAN) PENERIMAAN PESERTA DIDIK BARU

Reni Nursyanti, R.Yadi Rakhman Alamsyah, Surya Perdana

PERANCANGAN APLIKASI BERBASIS WEB UNTUK MEMBANTU PENGUJIAN KUALITAS KAIN TEKSTIL OTOMOTIF (STUDI KASUS PADA PT. ATEJA MULTI INDUSTRI)



Jurnal Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia, dan Informasi)
Volume 10, Nomor 2, Oktober 2019

NO	JUDUL PENELITIAN / NAMA PENULIS	HALAMAN
1.	KOMPARASI METODE KOMBINASI SELEKSI FITUR DAN MACHINE LEARNING K-NEAREST NEIGHBOR PADA DATASET LABEL HOURS SOFTWARE EFFORT ESTIMATION Indra Kurniawan, Ahmad Faiq Abror	83-89
2.	IMPLEMENTASI ALAT PEMBERI PAKAN TERNAK MENGGUNAKAN IOT UNTUK OTOMATISASI PEMBERIAN PAKAN TERNAK Fenty Ariani, Arnes Yuli Vandika, Handy Widjaya	90-97
3	IMPLEMENTASI SISTEM KONTROL BERBASIS WEB PADA SMART ROOM DENGAN MENGGUNAKAN KONSEP INTERNET OF THINGS Robby Yuli Endra , Ahmad Cucus, Freddy Nur Affandi, Deni Hermawan	98-106
4	PENERAPAN METODE SIMPLE ADDITIVE WEIGHTING (SAW) PADA AMRI SUPERMARKET BANJAR JAYA UNTUK PEMILIHAN KARYAWAN TERBAIK Tri Susilowati,Sucipto, Nungsiyati, Tomi Adi Kartika,Nur Zaman	107-115
5	BUS TRACKER – SISTEM LACAK LOKASI CALON PENUMPANG, LOKASI BUS DAN PERKIRAAN WAKTU KEDATANGAN BUS Erlangga,Taqwan Thamrin, Panji Maulana, Nico Susanto	116-121
6	APLIKASI CONVOLUTION NEURAL NETWORK UNTUK MENDETEKSI JENIS-JENIS SAMPAH Stephen, Raymond, Handri Santoso	122-132
7	IMPLEMENTASI FACE DETECTION PADA SMART CONFERENCE MENGGUNAKAN VIOLA JONES Freddy Nur Afandi, Ramses Parulian Sinaga, Yuthsi Aprilinda, Fenty Ariani	133-138
8	APLIKASI MONITORING BASE TRANSCIEVER STATION BERBASIS ANDROID MENGGUNAKAN METODE LOCATION BASED SERVICE Dani Yusuf, Freddy Nur Afandi	139-144
9	DECISION SUPPORT SYSTEM(SISTEM PENDUKUNG KEPUTUSAN) PENERIMAAN PESERTA DIDIK BARU Dede Aprilia Haspita, Jimi Ali Baba	145-152
10	PERANCANGAN APLIKASI BERBASIS WEB UNTUK MEMBANTU PENGUJIAN KUALITAS KAIN TEKSTIL OTOMOTIF (STUDI KASUS PADA PT. ATEJA MULTI INDUSTRI) Reni Nursyanti, R.Yadi Rakhman Alamsyah, Surya Perdana	153-159

Fakultas Ilmu Komputer
Universitas Bandar Lampung

JIST	Volume 10	Nomor 2	Halaman	Lampung Oktober 2019	ISSN 2087 – 2062 E-ISSN 2686-181X
------	-----------	---------	---------	-------------------------	--------------------------------------

**Jurnal Manajemen Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia & Informatika)**

Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Bandar Lampung

PENANGGUNG JAWAB

Rektor Universitas Bandar Lampung

Ketua Tim Redaksi:

Ahmad Cucus, S.Kom, M.Kom

Wakil Ketua Tim Redaksi:

Marzuki, S.Kom, M.Kom

TIM PENYUNTING :

PENYUNTING AHLI (MITRA BESTARI)

Prof. Mustofa Usman, Ph.D (Universitas Lampung)

Prof. Wamiliana, Ph.D (Universitas Lampung)

Akmal Junaidi, Ph.D (Universitas Lampung)

Handri Santoso, Ph.D (Institute Sains dan Teknologi Pradita)

Dr. Iing Lukman, M.Sc. (Universitas Malahayati)

Penyunting Pelaksana:

Robby Yuli Endra S.Kom., M.Kom

Yuthsi Aprilinda, S.Kom, M.Kom

Fenty Ariani, S.Kom., M.Kom

Pelaksana Teknis:

Wingky Kesuma, S.Kom

Shelvi, S.Kom

Alamat Penerbit/Redaksi:

Pusat Studi Teknologi Informasi - Fakultas Ilmu Komputer
Universitas Bandar Lampung
Gedung M Lantai 2 Pascasarjana
Jl. Zainal Abidin Pagar Alam no.89 Gedong Meneng Bandar Lampung
Email: explore@ubl.ac.id

PENGANTAR REDAKSI

Jurnal explore adalah jurnal yang diprakasai oleh program studi Informatika, Fakultas Ilmu Komputer Universitas Bandar Lampung, yang di kelola dan diterbitkan oleh Fakultas Ilmu Komputer / Pusat Sudi Teknologi Informasi.

Pada Edisi ini, explore menyajikan artikel/naskah dalam bidang teknologi informasi khususnya dalam pengembangan aplikasi, pengembangan machine learning dan pengetahuan lain dalma bidang rekayasa perangkat lunak, redaksi mengucapkan terima kasih dan selamat kepada penulis makalah ilmiah yang makalahnya kami terima dan di terbitkan dalam edisi ini, makalah ilmiah yang ada dalam jurnal ini memberikan kontribusi penting pada pengembangan ilmu dan teknologi.

Selain itu, sejumlah pakar yang terlibat dalam jurnal ini telah memberikan kontribusi yang sangat berharga dalam menilai makalah yang dimuat, oleh sebab itu, redaksi menyampaikan banyak terima kasih.

Pada kesempatan ini redaksi kembali mengundang dan memberikan kesempatan kepada para peneliti, di bidang pengembangan perangkat lunak untuk mempublikasikan hasil penelitiannya dalam jurnal ini.

Akhirnya redaksi berharap semoga makalah dalam jurnal ini bermanfaat bagi para pembaca khususnya bagi perkembangan ilmu dan teknologi dalam bidang perekaan perangkat lunak dan teknologi pada umumnya.

REDAKSI

KOMPARASI METODE KOMBINASI SELEKSI FITUR DAN MACHINE LEARNING K-NEAREST NEIGHBOR PADA DATASET LABEL HOURS SOFTWARE EFFORT ESTIMATION

Indra Kurniawan¹, Ahmad Faiq Abror²

Program Studi Rekayasa Perangkat Lunak

Politeknik Balekambang Jepara

Jl. Ponpes Balekambang Rt. 02/07 Jepara Jawa Tengah 59466

indrakurniawan.politbang@gmail.com¹ faiqabrор@gmail.com²

ABSTRACT

*The methods for Software Effort Estimation are divided into two, these methods are grouped into Non Machine Learning (non-ML) and Machine Learning (ML) methods [1]. The k-NN method has the disadvantage of being unable to tolerate irrelevant features and greatly affect the accuracy of k-NN. The k-NN method is also difficult to deal with missing data problems and feature categorization problems such as features that are not relevant, weight features that are not optimal, and the same features [2]. Whereas the dataset of Software Effort Estimation still has some serious challenges such as the characteristics of the data set, which are irrelevant features and the level of influence of each feature in the estimated data of the software effort [3]. This study compared the k-NN individual method with the combination of feature selection method with k-NN to find out which method was the best. The results showed that the Forward Selection (FS) method and Median Weighted Information Gain with k-Nearest Neighbor can overcome the problem of irrelevant features so as to increase accuracy in the RMSE Software Effort Estimation dataset, which is smaller in the Albrecht dataset of 5,953 using the Median method -WIG k-NN, the Miyazaki dataset is 55,421 and Kemerer is 123,081 using the FS k-NN method. The combination of kNN with Feature Selection is proven to be able to improve the estimation results better than kNN individuals. With the FS k-NN method being the best by winning in 2 datasets Miyazaki and Kemerer. **Keywords:** Software Effort Estimation, irrelevant feature, Forward Selection (FS), Median Weighted Information Gain, k-Nearest Neighbor.*

1. PENDAHULUAN

1.1 Latar Belakang

Estimasi usaha pengembangan perangkat lunak merupakan kegiatan dalam memperkirakan berapa banyak sumber daya yang dibutuhkan untuk menyelesaikan sebuah rencana proyek [4]. *Software Effort Estimation* adalah salah satu masalah tertua dan paling penting yang dihadapi dalam manajemen proyek perangkat lunak. Bisa merencanakan dengan benar adalah dasar untuk semua kegiatan manajemen proyek. Seseorang tidak dapat mengelola proyek tanpa mengetahui sumber daya apa yang dibutuhkan untuk mencapai tujuan proyek [5]. Dalam perkembangannya metode-metode untuk estimasi usaha pengembangan perangkat lunak terbagi menjadi dua, metode-metode tersebut di kelompokkan menjadi metode Non Machine Learning (non-ML) dan Machine Learning (ML) [1].

Beberapa metode non machine learning yang telah diusulkan untuk melakukan estimasi usaha pengembangan perangkat lunak diantaranya COCOMO [6], expert judgement [7], function point (FP) [8] dan use case point (UCP) [9]. Namun menurut hasil penelitian Nunes et al metode-metode non machine learning merupakan metode konvensional yang memiliki tingkat keakuratan yang relatif rendah [10].

Selain metode non machine learning yang banyak digunakan dalam *Software Effort Estimation* adalah machine learning. Hasil tinjauan literatur review Wen et all [2] menunjukkan bahwa hasil perbandingan antara model ML dan model non-ML dengan menggunakan model metrik MMRE. Sebagian besar percobaan menunjukkan bahwa model ML mengungguli model non-ML. Secara spesifik bahwa 66% (52 dari 79) hasil eksperimen menunjukkan keunggulan model ML sedangkan hanya 34% (27 dari 79) hasil

eksperimen menunjukkan keunggulan model non-ML.

Metode-metode machine learning yang telah digunakan untuk estimasi usaha pengembangan perangkat lunak diantaranya k-Nearest Neighbor (k-NN) [11], [12], [13], [14]. Artificial Neural Networks (ANN) atau Neural Network (NN) [11], [15], [16], Support Vector Machines [11], [17], Naive Bayes (NB) [18], Bayesian Networks (BN) [19], Decision Trees (DT) [12], Linear Regression (LR) [20].

Dari metode-metode ML tersebut hasil tinjauan literatur review yang dilakukan Wen et al [2] menunjukkan bahwa pendekatan CBR learning k-NN adalah teknik yang paling banyak digunakan, yaitu sebanyak 37%. Hasil perbandingan menunjukkan bahwa pendekatan CBR learning k-NN lebih akurat daripada model regresi. Pendekatan CBR dengan menggunakan algoritma k-NN mempunyai hasil yang lebih akurat dalam estimasi usaha perangkat lunak dibandingkan metode ML lainnya.

Menurut hasil tinjauan literatur review yang dilakukan Wen et al [2], algoritma k-NN merupakan metode yang mudah dimengerti dan menggunakan penalaran yang mirip dengan pemecahan masalah manusia, sehingga pengguna lebih mudah memahami. Algoritma k-NN juga memperlihatkan hasil yang bagus untuk kumpulan data kecil dan juga mampu mengatasi data yang outlier. Namun algoritma k-NN tidak dapat toleran terhadap fitur yang tidak relevan dan sangat mempengaruhi akurasi k-NN. MacDonell dan Shepperd berpendapat bahwa jika tidak ada teknik dominan yang dapat ditemukan maka menggabungkan beberapa teknik yang beragam dapat meningkatkan akurasi estimasi [21]. Penelitian yang dilakukan oleh Jørgensen [7] juga menemukan bahwa model gabungan menghasilkan estimasi yang lebih baik daripada model individual.

Dataset public *software effort* yang sering digunakan untuk estimasi usaha perangkat lunak diantaranya Albercht, Miyazaki, China, Desharnais, Maxwell, Kitchenham dan Cocomo. Label dalam dataset *software effort* terbagi menjadi dua, yaitu: hours dan mount.

Beberapa penelitian telah menunjukkan bahwa tingkat akurasi dalam *Software Effort Estimation* sangat bergantung pada fitur yang digunakan. Selain itu, pemilihan fitur yang digunakan telah menunjukkan pengaruh yang penting terhadap ketepatan estimasi [17].

Beberapa teknik dikembangkan untuk mengatasi masalah pengurangan fitur yang tidak relevan dan redundant fitur (fitur yang sama). Seleksi Fitur (*variable elimination*) membantu dalam memahami data, mengurangi kebutuhan komputasi, mengurangi efek dimensi dan meningkatkan kinerja prediksi [22].

Menurut Kabir, Shahjahan dan Murase [23] tujuan utama dari seleksi fitur adalah untuk menyederhanakan dan meningkatkan kualitas dataset dengan memilih fitur-fitur yang relevan. Seleksi fitur dilakukan dengan menghapus fitur yang kurang relevan dari kumpulan data asli tanpa mengorbankan kinerjanya.

Secara umum, pendekatan pemilihan fitur terbagi menjadi tiga kategori, yaitu: *Filter*, *Wrappers*, atau *Embedded algorithms* [24], [25]. Teknik *filter* dilakukan dengan menilai relevansi fitur dengan melihat sifat intrinsik data. Dalam banyak kasus, nilai relevansi fitur dihitung, dan fitur dengan skor rendah atau batasan yang ditentukan akan dihapus. Setelah itu, fitur pilihan ini disajikan sebagai data baru untuk algoritma klasifikasi/prediksi. Dalam metode filtering, sebuah fitur dapat dipilih berdasarkan beberapa kriteria yang telah ditetapkan sebelumnya [25].

Karena hal tersebut maka dalam penelitian ini peneliti akan melakukan komparasi kinerja metode individual k-NN dan metode seleksi fitur dengan algoritma k-NN pada *Software Effort Estimation*. Sehingga didapatkan diketahui model yang lebih akurat untuk *Software Effort Estimation*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka rumusan masalah penelitian ini yaitu: “Bagaimana komparasi kinerja antara metode individual k-NN dengan metode seleksi fitur dengan algoritma k-NN pada *Software Effort Estimation*?”

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka penelitian ini bertujuan untuk mengetahui kinerja terbaik antara metode individual k-NN dengan metode seleksi fitur dengan algoritma k-NN pada *Software Effort Estimation*.

1.4 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat

memberikan manfaat sebagai berikut:

- a. Dapat digunakan untuk melakukan estimasi berapa banyak sumber daya yang dibutuhkan untuk menyelesaikan sebuah proyek pengembangan perangkat lunak sehingga lebih akurat.
- b. Memberikan sumbangan dalam pengembangan dan penerapan model gabungan untuk meningkatkan kinerja algoritma k-Nearest Neighbor dalam menangani masalah data dengan fitur yang tidak relevan pada dataset *Software Effort Estimation*.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Beberapa penelitian terkait estimasi usaha perangkat lunak yang dijadikan state-of-the-art pada penelitian ini sebagai berikut:

Wen, Li, Lin, Hu dan Huang [2] yang melakukan Literature Review tentang metode-metode *Software Effort Estimation* yang digunakan dalam melakukan estimasi dari tahun 1991 sampai 2010. Dari 84 paper yang terdiri dari 59 paper yang terbit di jurnal, 24 paper yang terbit di conference atau proceeding dan satu paper dari chapter buku diketahui bahwa algoritma k-Nearest Neighbor (k-NN) merupakan yang paling banyak digunakan untuk melakukan estimasi usaha perangkat lunak yaitu sebanyak 37% dan disusul oleh Artificial Neural Networks (ANN) sebanyak 26%. Metode k-NN memperlihatkan hasil yang bagus untuk kumpulan data kecil dan juga mampu mengatasi data yang outlier. Namun metode k-NN tidak toleran terhadap fitur yang tidak relevan dan sangat mempengaruhi akurasi k-NN.

Aditya [11] yang melakukan perbandingan metode individual Machine Learning menggunakan dua dataset *Software Effort Estimation* yaitu Albercth dan Deshernanis. Metode yang dibandingkan yaitu *support vector machine*, *k-nearest neighbor*, *neural network* dengan parameter evaluasi yang digunakan adalah Root Mean Square Error (RMSE). Dari hasil komparasi RMSE anatara tiga model yang digunakan menunjukkan bahwa k-NN lebih unggul dibandingkan dengan SVM dan NN. Namun dengan penggunaan hanya 2 dataset *Software Effort Estimation* penelitian ini masih memiliki

kelemahan pada inkonsistensi hasil penelitian jika digunakan pada dataset lain.

Liu, Xiao dan Zhu [14] tentang *Software Effort Estimation* berfokus pada algoritma k-NN sebagai algoritma estimasi. Penulis mengkombinasikan algoritma k-NN dengan metode usulan untuk pemilihan fitur yaitu *localized neighborhood mutual information based feature selection (LFS)*. Metode LFS diusulkan untuk meminimalkan dampak informasi yang tidak dapat dijelaskan dalam pemilihan fitur. Metode validasi model eksperimen menggunakan *leave-one-out cross validation (LOOCV)* dengan kriteria evaluasi yang digunakan yaitu: Mean Absolute Residual (MAR) dan Mean Magnitude Relative Error (MMRE).

2.2 Landasan Teori

2.2.1 *Software Effort Estimation*

Menurut Presman estimasi sumber daya, biaya, dan waktu untuk usaha rekayasa pengembangan perangkat lunak membutuhkan pengalaman serta akses terhadap historis informasi yang baik. Caper Jones juga menyebutkan bahwa pendekatan estimasi yang baik dan historis data yang solid dapat menjadi harapan yang baik akan keberhasilan atas permasalahan estimasi yang tidak mungkin [26]. Steve McConnell menyebutkan bahwa estimasi yang bagus adalah estimasi yang memberikan pandangan yang cukup jelas tentang kenyataan proyek sehingga pemimpin proyek dapat mengambil keputusan yang baik tentang bagaimana mengendalikan proyek agar mencapai targetnya [5]

2.2.2 Data Mining

Data mining adalah proses menemukan pola dan pengetahuan dari sejumlah besar data. Sumber data dapat mencakup database, server data, Web, repositori informasi lain atau data yang dialirkan ke sistem secara dinamis. Sebagai sebuah proses penemuan pengetahuan, data mining biasanya melibatkan pembersihan data, integrasi data, pemilihan data, transformasi data, penemuan pola, evaluasi pola, dan menemukan pengetahuan. Salah satu domain dalam data mining adalah Machine learning. Metode machine learning melakukan pengolahan data dengan mesin atau computer [27].

2.2.3 k-Nearest Neighbor

k-Nearest Neighbor (k-NN) adalah suatu metode yang menggunakan algoritma

supervised dimana hasil dari query instance yang baru diklasifikasi berdasarkan mayoritas dari kategori pada k-NN. [28].

k-Nearest Neighbor (k-NN) adalah algoritma klasifikasi yang sederhana dan efektif. k-NN terdiri dari dua tahap, yaitu: pelatihan dan ujian yang dilakukan dengan memeriksa jarak dari setiap sampel uji ke sampel pelatihan untuk mengklasifikasikan setiap sampel uji. Nilai k harus dipilih berdasarkan dataset yang memiliki kelas sedangkan metrik jarak adalah parameter penting lainnya yang digunakan untuk menghitung dekat atau jauhnya tetangga dari sampel uji [29].

Tahapan untuk algoritma k-NN dapat penulis urutkan yaitu: 1) Tentukan nilai K, 2) Hitung jarak antara data baru dan semua data latih yang sudah ada dengan metrik jarak, 3) Tentukan K label data yang mempunyai jarak yang paling minimal, 4) Klasifikasikan data baru kedalam label data yang mayoritas

Ada berbagai metrik jarak pada k-NN namun metrik jarak Manhattan dan Euclidean merupakan matrik jarak yang paling banyak digunakan. Akurasi hasil dari kedua metrik jarak tersebut memperoleh hasil yang berbeda-beda [29].

3. METODE PENELITIAN

3.1 Persiapan Data

Penelitian ini menggunakan dataset *public* untuk *Software Effort Estimation* yang dapat diunduh melalui situs <http://openscience.us/repo/effort/>. Pada tahap pengolahan data, transformasi data dan pemilihan fitur pada masing-masing *dataset* dilakukan ke dalam bentuk yang dibutuhkan pada penelitian untuk metode yang diusulkan.

Tabel 1 Identifikasi Dataset

Dataset	Fitur	Jumlah Record	Effort Data			
			Satuan	Min	Max	Mean
Albrecht	8	24	Hours	0,5	105,2	22
Miyazaki	9	48	Hours	5,6	1.586	87
Kemerer	8	15	Hours	23,2	1.107	219

3.2 Metode yang Diusulkan

Peneliti mengusulkan untuk melakukan mengkomparasi metode machine learning k-Nearest Neighbor (k-NN) dan menambahkan algoritma Seleksi Fitur Median Weighted Information Gain (Median WIG) dan *Forward Selection* (FS).

Model yang diusulkan dalam penelitian ini mulai dari pengolahan dataset hingga menghasilkan model dan selanjutnya model akan diuji dengan menggunakan dataset testing. Dataset akan dilakukan pengujian dengan menggunakan algoritma berbeda secara bergantian. Hasil dari pengujian ini akan dilakukan komparasi algoritma mana yang paling akurat. Tahapan eksperimen dan pengujian metode dalam penelitian ini adalah sebagai berikut:

1. Melakukan pengujian metode individual k-Nearest Neighbor menggunakan 3 dataset public *Software Effort Estimation* berlabel hours yaitu: Albrecht, Miyazaki, Kemerer.
2. Melakukan pengujian metode seleksi fitur *Forward Selection* (FS) k-Nearest Neighbor menggunakan 3 dataset public *Software Effort Estimation* berlabel hours yaitu: Albrecht, Miyazaki, Kemerer.
3. Melakukan pengujian metode Median-WIG k-NN menggunakan 3 dataset public *Software Effort Estimation* berlabel hours yaitu: Albrecht, Miyazaki, Kemerer.

Tahapan-tahapan pengujian metode yang diusulkan dalam penelitian ini adalah sebagai berikut:

- a. Lakukan pengolahan data awal dengan transformasi setiap data pada dataset asli dengan fungsi Binary Sigmoid (Logsig).
 - b. Lakukan seleksi fitur dengan Median Weighted Information Gain (Median WIG) dan bentuk dataset baru dengan cara menentukan fitur yang relevan dengan kriteria (threshold) yaitu nilai fitur yang diatas median akan dipertahankan untuk digunakan dan fitur dibawah median akan dihilangkan.
4. Melakukan pembahasan komparasi hasil estimasi menggunakan nilai RMSE antara metode individual k-Nearest Neighbor, metode seleksi fitur *Forward Selection* (FS) dengan k-Nearest Neighbor dan metode seleksi fitur Median Weighted Information Gain (Median WIG) dengan k-Nearest Neighbor.

4. HASIL PENELITIAN

4.1 Pengujian Metode Individual k-Nearest Neighbor

Hasil perhitungan dengan *Rapidminer* metode individual k-NN menggunakan

parameter $k = 1$ dan perhitungan jarak Euclidean Distance diperoleh hasil perhitungan nilai RMSE pada public *Software Effort Estimation* berlabel hours sebagai berikut:

Tabel 2 Hasil Nilai RMSE Metode Individual k-Nearest Neighbor

Dataset	k-NN
Albrecht	6.583
Miyazaki	68.096
Kemerer	143.634

Dapat diketahui berdasarkan tabel diatas hasil perhitungan nilai RMSE yang diperoleh pada dataset Albrecht sebesar 6.583, dataset Miyazaki sebesar 68.096, dataset Kemerer sebesar 143.634.

4.2 Pengujian Metode Seleksi Fitur Forward Selection (FS) dengan Algoritma k-Nearest Neighbor

Hasil perhitungan dengan *Rapidminer* metode Forward Selection (FS) k-NN dengan parameter k-NN menggunakan $k = 1$ dan perhitungan jarak Euclidean distance diperoleh hasil perhitungan nilai RMSE pada 6 dataset *Software Effort Estimation* sebagai berikut:

Tabel 3 Keseluruhan Nilai RMSE metode FS k-NN

Dataset	FS k-NN
Albrecht	6.454
Miyazaki	55.421
Kemerer	123.081

Dapat diketahui bahwa hasil perhitungan metode FS k-NN pada tabel diatas menunjukkan nilai RMSE yang diperoleh untuk dataset Albrecht sebesar 1391.497, dataset Miyazaki sebesar 55.421, dataset Kemerer sebesar 123.081.

4.3 Pengujian Metode Seleksi Fitur Median WIG k-Nearest Neighbor

Langkah pertama lakukan pengolahan data dengan melakukan transformasi setiap data dengan fungsi Binary Sigmoid (Logsig). Setelah diperoleh data hasil transformasi setiap data dengan fungsi Binary Sigmoid (Logsig) lakukan perhitungan nilai entropy setiap data. Setelah diperoleh data hasil entropy lalu hitung nilai gain setiap data tersebut dan kemudian hitung median kumpulan data untuk nantinya dilakukan seleksi fitur pada dataset.

Pengujian Metode Median WIG k-Nearest Neighbor pada Dataset Albrecht diperoleh nilai median sebesar 0.197 sehingga terdapat 3 fitur yang akan dihilangkan dikarenakan nilai dibawah median (fitur tidak relevan) yaitu fitur Input, Inquiry dan File. Serta terdapat 4 fitur diatas median (fitur relevan) yaitu fitur Output, FPAdj, RawFPcounts, AdjFP serta 1 label Effort yang akan dipertahankan. Pada Dataset Miyazaki diperoleh nilai median sebesar 0.109 Sehingga terdapat 3 fitur yang akan dihilangkan dikarenakan nilai dibawah median (fitur tidak relevan) yaitu: KLOC, ESCRN, EFILE. Serta terdapat 4 fitur diatas median (fitur relevan) yaitu fitur SCRN, FORM, FILE, EFORM serta 1 label MM yang akan dipertahankan. Pada Dataset Kemerer diperoleh nilai median sebesar 0.221 Sehingga terdapat 3 fitur yang akan dihilangkan dikarenakan nilai dibawah median (fitur tidak relevan) yaitu: Language, Hardware, KSLOC. Serta terdapat 3 fitur diatas median (fitur relevan) yaitu: Duration, AdjFP, RAWFP serta 1 label EffortMM yang akan dipertahankan.

Pengujian dataset baru *Software Effort Estimation* dengan algoritma k-Nearest Neighbor menggunakan *rapidminer*. Hasil perhitungan dengan *Rapidminer* metode usulan Median-WIG k-NN dengan parameter k-NN menggunakan $k = 1$ dan perhitungan jarak Euclidean distance diperoleh hasil perhitungan nilai RMSE sebagai berikut:

Tabel 4 Keseluruhan Nilai RMSE metode Median-WIG k-NN

Dataset	Median WIG k-NN
Albrecht	5.958
Miyazaki	62.363
Kemerer	131.027

Dapat diketahui bahwa hasil perhitungan metode Median-WIG k-NN pada tabel diatas menunjukkan nilai RMSE yang diperoleh untuk dataset Albrecht sebesar 5.958, dataset Miyazaki sebesar 62.363, dataset Kemerer sebesar 131.027.

4.4 Pembahasan Komparasi Hasil Nilai RMSE

Pada tahap ini dilakukan perbandingan nilai RMSE dari seluruh metode dengan parameter k-NN menggunakan $k = 1$ dan perhitungan jarak Euclidean Distance. Berdasarkan hasil pengujian metode individual

k-Nearest Neighbor, seleksi fitur Forward Selection (FS) dan k-Nearest Neighbor, metode usulan seleksi fitur Median Weighted Information Gain (Median WIG) dan k-Nearest Neighbor yang keseluruhan hasil nilai RMSE dapat dilihat pada tabel dibawah ini:

Tabel 5 Komparasi Hasil Pengujian Keseluruhan Metode

Dataset	k-NN	FS k-NN	Median WIG k-NN
Albrecht	6.583	6.454	5.958
Miyazaki	68.096	55.421	62.363
Kemerer	143.634	123.081	131.027

Dari hasil pengujian yang telah di lakukan diatas menghasilkan nilai RMSE terkecil yaitu: pada dataset Albrecht sebesar 5.953 menggunakan metode Median-WIG k-NN, pada dataset Miyazaki sebesar 55.421 menggunakan metode FS k-NN, pada dataset Kemerer sebesar 123.081 menggunakan metode FS k-NN.

Hasil pengujian menunjukkan bahwa metode individual k-NN memperoleh nilai RMSE yang paling rendah dibandingkan dengan metode integrasi seleksi fitur dan k-NN, hal ini menunjukkan bahwa metode individual k-NN tidak dapat mengatasi data dengan fitur yang kurang relevan dan sangat mempengaruhi dari akurasi estimasi yang dilakukan.

Dari hasil komparasi seluruh pengujian tersebut diketahui bahwa metode seleksi fitur terbukti mampu meningkatkan akurasi estimasi yang ditunjukkan dengan penurunan nilai RMSE yang signifikan pada seluruh dataset. Sehingga terjadi peningkatan kinerja setelah menggunakan metode seleksi fitur. Secara keseluruhan dilihat dari RMSE seluruh dataset metode FS k-NN memiliki kinerja yang paling baik karena telah unggul pada 2 dataset.

Metode FS dapat meningkatkan akurasi secara signifikan algoritma k-NN pada *Software Effort Estimation* dengan unggul pada 2 dataset. Hal ini membuktikan fitur yang tidak relevan sangat mempengaruhi akurasi dalam melakukan estimasi. Dengan penggunaan metode yang diusulkan dengan menghilangkan fitur yang tidak relevan mampu meningkatkan akurasi estimasi yang dilihat dari penurunan nilai RMSE secara signifikan menjadi lebih kecil.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan maka kesimpulan dari peneliti ini yaitu:

Fitur yang tidak relevan sangat mempengaruhi akurasi dalam melakukan estimasi. Dengan penggunaan metode seleksi fitur dapat mengatasi masalah fitur yang tidak relevan, hal tersebut mampu meningkatkan akurasi estimasi yang dilihat dari penurunan nilai RMSE secara signifikan menjadi lebih kecil dari pada penggunaan metode individual k-NN.

Hasil komparasi menunjukkan bahwa metode Forward Selection (FS) dan Median Weighted Information Gain dengan k-Nearest Neighbor dapat mengatasi masalah fitur yang tidak relevan sehingga mampu meningkatkan akurasi pada dataset *Software Effort Estimation* dengan RMSE menjadi lebih kecil yaitu pada dataset Albrecht sebesar 5.953 menggunakan metode Median-WIG k-NN, pada dataset Miyazaki sebesar 55.421 dan Kemerer sebesar 123.081 menggunakan metode FS k-NN.

DAFTAR PUSTAKA

- [1] M. Shepperd and S. Macdonell, "Evaluating prediction systems in software project estimation," *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 820–827, 2012.
- [2] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, 2012.
- [3] A. Idri and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review," *Inf. Softw. Technol.*, 2014.
- [4] I. Sommerville, *Software Engineering Ninth Edition*, 9th ed. Boston: PEARSON, 2011.
- [5] A. Trendowicz and R. Jeffery, *Software Project Effort Estimation*. New York: Springer, 2014.
- [6] X. Huang, D. Ho, J. Ren, and L. F. Capretz, "Improving the COCOMO model using a neuro-fuzzy approach," *Appl. Soft Comput.*, vol. 7, pp. 29–40, 2007.
- [7] S. Grimstad and M. Jørgensen, "Inconsistency of expert judgment-based estimates of software development effort," *J. Syst. Softw.*, vol. 80, pp. 1770–1777, 2007.
- [8] G. R. Finnie and G. E. Wittig, "A Comparison of Software Effort Estimation

- Techniques : Using Function Points with Neural Networks , Case-Based Reasoning and Regression Models,” *J. Systems Softw.*, vol. 1212, no. 97, pp. 281–289, 1997.
- [9] S. M. Satapathy, B. P. Acharya, and S. K. Rath, “Early stage software effort estimation using random forest technique based on use case points,” *IET Softw.*, vol. 10, no. 1, pp. 10–17, 2016.
- [10] E. U. Points, N. J. Nunes, and L. Constantine, “i UCP : Software Project Size with,” *IEEE Softw.*, pp. 64–73, 2011.
- [11] E. K. Adhitya, R. Satria, and H. Subagyo, “Komparasi Metode Machine Learning dan Non Machine Learning untuk Estimasi Usaha Perangkat Lunak,” *J. Softw. Eng.*, vol. 1, no. 2, pp. 109–113, 2015.
- [12] A. Bakır, B. Turhan, and A. Bener, “A comparative study for estimating software development effort intervals,” *Softw. Qual J*, vol. 19, pp. 537–552, 2011.
- [13] V. Khatibi, B. Dayang, and N. Abang, “A PSO-based model to increase the accuracy of software development effort estimation,” *Softw. Qual J*, vol. 21, pp. 501–526, 2013.
- [14] Q. Liu, J. Xiao, and H. Zhu, “Feature selection for software effort estimation with localized neighborhood mutual information,” *Cluster Comput.*, no. 1, 2018.
- [15] V. S. Dave, “Comparison of Regression model , Feed-forward Neural Network and Radial Basis Neural Network for Software Development Effort Estimation,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 36, no. 5, pp. 1–5, 2011.
- [16] C. López-martín, “Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects,” *Appl. Soft Comput. J.*, pp. 1–16, 2014.
- [17] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, “GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation,” *Inf. Softw. Technol.*, vol. 52, no. 11, pp. 1155–1166, 2010.
- [18] J. Shivhare and S. K. Rath, “Software Effort Estimation using Machine Learning Techniques,” *ISEC*, 2014.
- [19] F. Zare, H. Khademi Zare, and M. S. Fallahnezhad, “Software effort estimation based on the optimal Bayesian belief network,” *Appl. Soft Comput.*, vol. 49, pp. 968–980, 2016.
- [20] R. Malhotra, A. Kur, and Y. Sigh, “Application of Machine Learning Methods for Software Effort Prediction,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 35, no. 3, pp. 1–6, 2010.
- [21] S. G. Macdonell and M. J. Shepperd, “Combining techniques to optimize effort predictions in software project management,” *J. Syst. Softw.* 66, vol. 66, pp. 91–98, 2003.
- [22] G. Chandrashekar and F. Sahin, “A survey on feature selection methods q,” *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.
- [23] M. Kabir and K. Murase, “A new hybrid ant colony optimization algorithm for feature selection,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3747–3763, 2012.
- [24] H. Liu, S. Member, L. Yu, and S. Member, “Algorithms for Classification and Clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, 2005.
- [25] A. Idri and S. Cherradi, “Improving Effort Estimation of Fuzzy Analogy using Feature Subset Selection,” 2016.
- [26] R. S. Pressman, *Software Engineering*, 7th ed. Boston: McGraw-Hill, 2010.
- [27] J. Han, M. Kamber, and J. Pei, *Data Mining : Concepts and Techniques*, 3rd ed. Waltham: Elsevier Inc., 2012.
- [28] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining : Practical Machine Learning Tools and Techniques*, 3 rd. Burlington,: Elsevier, 2011.
- [29] M. Kaya, “Classification of Pancreas Tumor Dataset Using Adaptive Weighted k Nearest Neighbor Algorithm,” *IEEE*, pp. 0–4, 2014.

PEDOMAN PENULISAN

1. Naskah belum pernah dipublikasikan atau dalam proses penyuntingan dalam jurnal ilmiah atau dalam media cetak lain.
2. Naskah diketik dengan spasi 1 pada kertas ukuran A4 dan spasi 2,5 sentimeter dengan huruf *Times New Roman* berukuran 11 point. Naskah diserahkan dalam bentuk cetakan sebanyak 2 eksemplar disertai *file* dalam CD atau dapat dikirim melalui *e-mail* kepada redaksi.
3. Naskah bebas dari tindakan plagiat.
4. Naskah dapat ditulis dalam bahasa Indonesia atau bahasa Inggris dengan jumlah isi 10–25 halaman A4 termasuk daftar pustaka.
5. Naskah berupa artikel hasil penelitian terdiri dari komponen: judul, nama penulis, abstrak, kata kunci, pendahuluan, metode, hasil, pembahasan, kesimpulan, daftar pustaka.
6. Daftar pustaka terdiri dari acuan primer (80%) dan sekunder (20%). Acuan primer berupa jurnal ilmiah nasional dan internasional, sedangkan acuan sekunder berupa buku teks.
7. Naskah berupa artikel konseptual terdiri dari komponen: judul, nama penulis, abstrak, kata kunci, pendahuluan, hasil, pembahasan, kesimpulan, daftar pustaka, dan ucapan terima kasih (jika ada).
8. Judul harus menggambarkan isi artikel secara lengkap, maksimal terdiri atas 12 kata dalam bahasa Indonesia atau 10 kata dalam bahasa Inggris.
9. Nama penulis disertai dengan asal lembaga tetapi tidak disertai dengan gelar. Penulis wajib menyertakan biodata penulis yang ditulis pada lembar terpisah, terdiri dari: alamat kantor, alamat, dan telepon rumah, Hp. dan *e-mail*.
10. Abstrak ditulis dalam bahasa Inggris atau bahasa Indonesia. Abstrak memuat ringkasan esensi hasil kajian secara keseluruhan secara singkat dan padat. Abstrak memuat latar belakang, tujuan, metode, hasil, dan kesimpulan. Abstrak diketik spasi tunggal dan ditulis dalam satu paragraf.
11. Kata kunci harus mencerminkan konsep atau variabel penelitian yang dikandung, terdiri atas 5–6 kata.
12. Pendahuluan menjelaskan hal-hal pokok yang dibahas, yang berisi tentang permasalahan penelitian, tujuan penelitian, dan rangkuman kajian teoritik yang relevan. Penyajian pendahuluan dalam artikel tidak mencantumkan judul.
13. Metode meliputi rancangan penelitian, populasi dan sampel, pengembangan instrumen penelitian, teknik pengumpulan data, dan teknik analisis data, yang diuraikan secara singkat.
14. Hasil menyajikan hasil analisis data yang sudah final bukan data mentah yang belum diolah.
15. Pembahasan merupakan penegasan secara eksplisit tentang interpretasi hasil analisis data, mengaitkan hasil temuan dengan teori atau penelitian terdahulu, serta implikasi hasil temuan dikaitkan dengan keadaan saat ini.
16. Pemaparan deskripsi dapat dilengkapi dengan gambar, foto, tabel, dan grafik yang semuanya mencantumkan judul, dan sumber acuan jika diperlukan.
17. Istilah dalam bahasa Inggris ditulis dalam huruf miring (*italic*).

Redaksi :
Pusat Studi Teknologi Informasi - Fakultas Ilmu Komputer
Universitas Bandar Lampung
Gedung M Lantai 2 Pascasarjana
Jl.Zainal Abidin Pagar Alam no.89 Gedong Meneng Bandar Lampung
Email: explore@ubl.ac.id

