EXPERT JURNAL MANAJEMEN SISTEM INFORMASI DAN TEKNOLOGI

Estimasi Proyek Aplikasi Online Shop dengan COCOMO II Menggunakan Pendekatan Algoritma SPRINT

Nisya Awalliya, Yulison Herry Chrisnanto*, Rezki Yuniarti
Informatika, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani, Cimahi, Indonesia
nisyaawalliya20@if.unjani.ac.id, * yhc@if.unjani.ac.id, rezki.yuniarti@lecture.unjani.ac.id

ABSTRACT – Software Project Estimation for an Online Shop Application Using COCOMO II with the SPRINT Algorithm. Effort estimation is an important factor in successful software development. COCOMO II is an estimation model that can estimate application project effort using scale factors and cost drivers. However, the accuracy of this model is considered low, so this research aims to improve the estimation accuracy of the COCOMO II model by applying the SPRINT algorithm approach to the research object of the online shop application project. The COCOMO II model is used to calculate the estimation of project time, personnel, and costs. Meanwhile, the SPRINT algorithm is used to predict the priority of module work based on the COCOMO II estimation results. This study compares the accuracy of effort estimation with previous research using the COCOMO II model with the C4.5 algorithm approach which is the predecessor of the SPRINT algorithm. The results show that the estimation of application projects using the COCOMO II model with the SPRINT algorithm approach produces 100% accuracy, more accurate than the COCOMO II model with the C4.5 algorithm approach which only produces 90% accuracy. This research proves that the use of the SPRINT algorithm can further improve the speed and accuracy of prediction compared to the use of the C4.5 algorithm.

Keywords: COCOMO II; Effort Estimation; SPRINT Algorithm; Online Shop

ABSTRAK – Estimasi upaya merupakan faktor penting dalam keberhasilan pengembangan perangkat lunak. COCOMO II adalah model estimasi yang dapat memperkirakan upaya proyek aplikasi dengan menggunakan scale factor dan juga cost driver. Namun demikian, akurasi model ini dianggap rendah, sehingga penelitian ini bertujuan untuk meningkatkan akurasi estimasi model COCOMO II dengan menerapkan pendekatan algoritma SPRINT pada objek penelitian proyek aplikasi online shop. Model COCOMO II digunakan untuk menghitung estimasi waktu, personel, dan biaya proyek. Sementara itu, algoritma SPRINT digunakan untuk memprediksi prioritas pengerjaan modul berdasarkan hasil estimasi COCOMO II. Penelitian ini membandingkan akurasi estimasi upaya dengan penelitian sebelumnya yang menggunakan model COCOMO II dengan pendekatan algoritma C4.5 yang merupakan pendahulu dari algoritma SPRINT. Hasil penelitian menunjukkan bahwa estimasi proyek aplikasi menggunakan model COCOMO II menggunakan pendekatan Algoritma SPRINT menghasilkan akurasi 100%, lebih tinggi dibandingkan model COCOMO II menggunakan pendekatan algoritma C4.5 yang hanya mencapai 90%. Penelitian ini membuktikan bahwa penggunaan algoritma SPRINT dapat lebih meningkatkan kecepatan dan akurasi prediksi dibandingkan penggunaan algoritma C4.5.

Kata Kunci: Algoritma SPRINT; COCOMO II; Estimasi Upaya; Online Shop

1. PENDAHULUAN

Dalam beberapa tahun terakhir, penggunaan aplikasi *Online shop* sebagai media perantara proses jual beli telah meningkat dengan pesat [1]. Aplikasi *online shop* merupakan sebuah aplikasi yang dimiliki oleh sebuah merek dagang untuk menjual barang dagangannya secara daring. Saat ini, telah banyak merek dagang yang menggunakan aplikasi *online shop* untuk memperluas target pasar dan menjaring pembeli dengan lebih mudah [1]. Oleh sebab itu, aplikasi yang digunakan harus memiliki kualitas yang memadai, sehingga baik penjual ataupun pelanggan memiliki pengalaman dalam menjalankan aplikasi *online shop* tersebut [2].

Dari segi pengembangannya, perangkat lunak yang berkualitas adalah perangkat lunak yang diselesaikan sesuai dengan waktu, anggaran, dan kualitas minimum yang telah ditetapkan pada saat proses *requirement* [3]. Estimasi upaya menjadi salah satu faktor penting dalam membangun kualitas perangkat lunak [4]. Kekurangan dan kelebihan estimasi dapat mengakibatkan penyalahgunaan anggaran, kualitas aplikasi yang rendah, dan bahkan kegagalan proyek [5]. Oleh karena itu, penggunaan metode yang tepat sangat penting dalam menentukan estimasi untuk mencapai akurasi estimasi yang lebih baik dan menghindari terjadinya masalah[5].

COCOMO II merupakan model estimasi upaya yang telah banyak digunakan dan mampu dengan cepat memproyeksikan upaya, jadwal, personel, dan biaya yang diperlukan untuk menyelesaikan proyek pengembangan perangkat lunak [6]. Namun, akurasi dan validitas estimasi dari model COCOMO II dalam proyek pengembangan perangkat lunak berskala besar masih dianggap kurang

akurat [7]. Proyek berskala besar cenderung memiliki durasi pengerjaan yang lebih lama, sehingga selama waktu itu kebutuhan dan spesifikasi dari perangkat lunak dapat berubah secara signifikan [6]. COCOMO II memiliki pendekatan yang lebih statis sehingga akan lebih sulit untuk menyesuaikan hasil estimasi terhadap perubahan-perubahan tersebut [8]. Oleh sebab itu, banyak penelitian yang menggunakan COCOMO II sebagai model estimasi upaya yang dikolaborasikan dengan berbagai algoritma untuk mengoptimalkan akurasi estimasi, seperti Algoritma C4.5 [9], *Random Forest* [10], Fuzzy [2], Neuro Fuzzy [11], dan lainnya.

Pada penelitian [9] yang membahas estimasi upaya proyek aplikasi *Point Of Sales (POS)* menggunakan COCOMO II sebagai model estimasi upaya dan menerapkan pendekatan algoritma C4.5 untuk mendapatkan estimasi dengan akurasi prioritas yang lebih baik [9]. Hasil dari penelitian ini menunjukkan bahwa penggunaan COCOMO II dengan menerapkan pendekatan algoritma C4.5 untuk memprediksi prioritas modul pada estimasi upaya aplikasi POS mencapai akurasi sebesar 90% yang dapat digunakan sebagai dasar acuan pengembangan proyek aplikasi POS selanjutnya [9].

Untuk meningkatkan akurasi estimasi upaya proyek pengembangan aplikasi, penggunaan model COCOMO II dengan pendekatan algoritma C4.5 memberikan kontribusi yang begitu signifikan [9]. Namun demikian, Algoritma *Scalable Parallelizable Induction of Decision Tree (SPRINT)* yang merupakan algoritma perbaikan dari C4.5, memiliki kecepatan, skalabilitas, fleksibilitas dan akurasi prediksi yang lebih baik dibandingkan dengan algoritma C4.5 [12].

Algoritma SPRINT merupakan algoritma klasifikasi paralel yang memungkinkan untuk kolaborasi dalam membangun model klasifikasi yang akurat dari *dataset* berukuran besar dan kompleks [13]. Algoritma ini mengatasi dua tantangan utama dalam klasifikasi paralel yaitu, penempatan data dan penyeimbangan beban kerja, serta tidak memerlukan struktur data terpusat, membuat algoritma ini lebih optimal dan fleksibel [12].

Dengan kemampuan untuk terus menerus memproses dan menganalisis data baru, model yang dihasilkan dapat diperbarui secara berkala, sehingga Algoritma SPRINT dapat membantu model COCOMO II dalam membuat estimasi yang lebih dinamis dan adaptif, sesuai dengan kebutuhan perangkat lunak.

Oleh karena itu, pada penelitian ini akan dilakukan estimasi upaya berupa jadwal, personel, dan biaya serta prioritas pengerjaan modul yang bertujuan untuk membangun kualitas dan mengurangi potensi kegagalan proyek akibat estimasi yang buruk pada proyek pengembangan perangkat lunak *online shop*. Proses estimasi dihitung menggunakan model COCOMO II dan menerapkan pendekatan Algoritma SPRINT untuk meningkatkan akurasi prediksi prioritas modul, dengan faktor estimasi berasal dari aplikasi yang *online shop* yang telah ada yaitu, aplikasi *Adorable Projects*.

2. TINJAUAN PUSTAKA

Estimasi upaya pada proyek perangkat lunak merupakan sebuah proses memprediksi waktu, personel, dan biaya dalam perancangan proyek perangkat lunak [14]. Hal ini melibatkan analisis berbagai faktor seperti ruang lingkup proyek, persyaratan, kompleksitas, kemampuan tim, metodologi pengembangan, dan potensi risiko untuk menentukan keseluruhan biaya pengembangan produk perangkat lunak [15].

Estimasi upaya perangkat lunak memerlukan pengamatan dan uji coba yang sangat teliti untuk menentukan metode pengembangan yang tepat [16]. Teknik estimasi upaya perangkat lunak memungkinkan prediksi usaha dan biaya proyek, serta perkiraan jumlah tenaga kerja dan waktu yang dibutuhkan untuk menyelesaikan proyek [16].

COCOMO II merupakan model untuk mengestimasi upaya yang dibutuhkan dalam proses pengembangan perangkat lunak, baik itu dalam segi biaya, personel, maupun waktu pengembangannya [17]. Dalam model ini, seluruh perangkat lunak dibagi menjadi beberapa modul yang berbeda [16]. Beberapa sub-model bagian dari COCOMO II, di antaranya adalah *application-composition, early-design, reuse,* dan *post-architecture* [15].

Post-Architecture merupakan sub-model yang paling rinci dan menggunakan arsitektur sistem yang telah ada sebelumnya, sehingga struktur sub-sistemnya dapat diketahui [16]. Estimasi yang dihasilkan dari *post-architecture* didasarkan pada Rumus 1.

$$PM = A.Size^{E}.\prod_{i=1}^{17} EMi \qquad \text{(Rumus 1)}$$

Dengan A merupakan konstanta yang memiliki nilai 2,940, lalu *Size* didefinisikan sebagai *Kilo Source Line of Code (KSLOC)* [8]. E merupakan nilai eksponen pada Rumus 2. Lalu, terakhir EMi merupakan tujuh belas faktor pendorong estimasi, yang juga disebut sebagai *Effort Multiplier (EM),* seperti yang ditunjukkan pada Tabel 2.

$$E = B + 0.01\sum_{I=1}^{5} SF \qquad \text{(Rumus 2)}$$

Dengan B merupakan konstanta yang memiliki nilai 0,91, dan SF merupakan variabel *Scale factor (SF)* berkisar dari *Very Low (VL)* hingga *Extra High (EH)* yang nilainya ditunjukkan pada Tabel 1. Proses perhitungan estimasi jadwal, personel, dan biaya pun dilakukan dengan menggunakan beberapa Rumus. Rumus 3 merupakan Rumus untuk mengukur estimasi jadwal.

$$TDEV = C \times PM^{F} \qquad \text{(Rumus 3)}$$

Dengan *C* merupakan konstanta yang memiliki nilai 3,67 dan *F* merupakan konstanta yang memiliki nilai 0,28. Selanjutnya untuk menghitung estimasi jumlah personel dan estimasi biaya, dilakukan dengan menggunakan Rumus 4 dan 5 secara berurutan.

DOI http://dx.doi.org/10.36448/expert.v14i1.3770
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 1
June 30, 2024 – pg. 64

$$P = \frac{PM}{TDEV} \quad \text{(Rumus 4)}$$

$$Cost = P \times Avg\ Labor\ Cost \quad \text{(Rumus 5)}$$

Dengan *PM* dan *TDEV* didapatkan dari perhitungan Rumus 1 dan Rumus 3, dan a*vg labor cost* merupakan rata-rata gaji *programmer* di Indonesia per bulan.

Seperti telah disebutkan pada Rumus 1, 2, 3, 4, 5, estimasi COCOMO II memerlukan 3 faktor estimasi, yaitu KSLOC, *scale factor*, dan *effort multipliers*. KSLOC merupakan ribuan baris kode sumber yang dihitung tanpa menyertakan baris komentar dan baris kosong. *Scale factor* merupakan faktor-faktor yang mempengaruhi seberapa besar usaha tambahan yang dibutuhkan saat ukuran proyek meningkat. Faktor-faktor ini bernilai dari *Very Low (VL), Low (L), Nominal (N), High (H), Very High (VH)* hingga *Extra High (EH),* seperti pada Tabel 1 penelitian maupun pengujian yang telah dilakukan.

Sedangkan *effort multipliers* merupakan faktor-faktor yang mempengaruhi estimasi upaya *(effort)* yang dibutuhkan untuk menyelesaikan proyek perangkat lunak. *Effort multiplierss* ini memperhitungkan berbagai aspek dari proyek, termasuk atribut produk, *platform,* personil, dan proyek itu sendiri. Setiap *effort multipliers* memiliki nilai yang menggambarkan seberapa besar pengaruhnya terhadap usaha yang diperlukan, dengan *range* nilai setiap faktor sama seperti *scale factor*, seperti pada Tabel 2.

Tabel 1 *Scale Factor*

Drivers	**Simbol**	**VL**	**L**	**N**	**H**	**VH**	**EH**
PREC	SF_1	6.20	4.96	3.72	2.48	1.24	0.0
FLEX	SF_2	5.07	4.05	3.04	2.03	1.01	0.0
RSEL	SF_3	7.07	5.65	4.24	2.83	1.41	0.0
TEAM	SF_4	5.48	4.38	3.29	2.19	1.10	0.0
PMAT	SF_5	7.80	6.24	4.68	3.12	1.56	0.0

Tabel 2 *Effort Multipliers*

Drivers	**Simbol**	**VL**	**L**	**N**	**H**	**VH**	**EH**
RELY	EM_1	0.82	0.92	1	1.10	1.26	-
DATA	EM_2	-	0.90	1	1.14	1.28	-
CPLX	EM_3	0.73	0.87	1	1.17	1.34	1.74
RUSE	EM_4	-	0.95	1	1.07	1.15	1.24
DOCU	EM_5	0.81	0.91	1	1.11	1.23	-
TIME	EM_6	-	-	1	1.11	1.29	1.63
STOR	EM_7	-	-	1	1.05	1.17	1.46
PVOL	EM_8	-	0.87	1	1.15	1.30	-
ACAP	EM_9	1.42	1.19	1	0.85	0.71	-
PCAP	EM_{10}	1.34	1.15	1	0.88	0.76	-
PCON	EM_{11}	1.29	1.12	1	0.90	0.81	-
APEX	EM_{12}	1.22	1.10	1	0.88	0.81	-
PLEX	EM_{13}	1.19	1.09	1	0.91	0.85	-
LTEX	EM_{14}	1.20	1.09	1	0.91	0.84	-
TOOL	EM_{15}	1.17	1.09	1	0.90	0.78	-
SITE	EM_{16}	1.22	1.09	1	0.93	0.86	0.80
SCED	EM_{17}	1.43	1.14	1	1.00	1.00	-

Algoritma SPRINT atau Algoritma *Scalable Parallelizable Induction of Decision Tree* yang dikembangkan pada tahun 1996 oleh Shafer, Agrawal, dan Mehta, memungkinkan banyak prosesor yang berbeda untuk bekerja bersama pada kumpulan data yang sangat besar, dengan tujuan utama untuk menghindari kendala memori dengan menghilangkan hubungan antara ukuran memori utama dan dimensi kumpulan data pelatihan [11].

Algoritma ini tergolong ke dalam algoritma yang cepat dan dapat diskalakan [13]. Melalui pemrosesan paralel, algoritma SPRINT membagi pekerjaan menjadi tugas-tugas kecil yang dapat dijalankan secara bersamaan oleh banyak prosesor, memungkinkan pembagian data dan pembentukan *node* pohon keputusan secara cepat dan tepat [13].

Algoritma SPRINT adalah algoritma induksi pohon keputusan yang dirancang untuk bekerja secara paralel pada sistem komputasi dengan banyak prosesor [18]. Algoritma ini tidak memiliki rumus matematis sederhana seperti algoritma lain yang lebih analitik [18]. Algoritma ini terdiri dari serangkaian langkah dan struktur data yang memungkinkan pemrosesan paralel yang ringkas [12].

Penelitian terdahulu [10] menggunakan *Random Forest Regression* sebagai metode dan diimplementasikan dengan menggunakan *dataset* NASA93. Hasil penelitian ini menunjukkan bahwa *Random Forest Regression* memiliki

DOI http://dx.doi.org/10.36448/expert.v14i1.3770
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 1
June 30, 2024 – pg. 65

MMRE sebesar 54%, MMRE *Support Vector Regression* sebesar 73%, dan MMRE Metode *Bee Colony* 115%. Hasil dari penelitian ini menunjukkan bahwa penggunaan *Random Forest Regression* dalam meningkatkan akurasi estimasi COCOMO II lebih baik dibandingkan dengan *Support Vector Regression* dan *Bee Colony Method*.

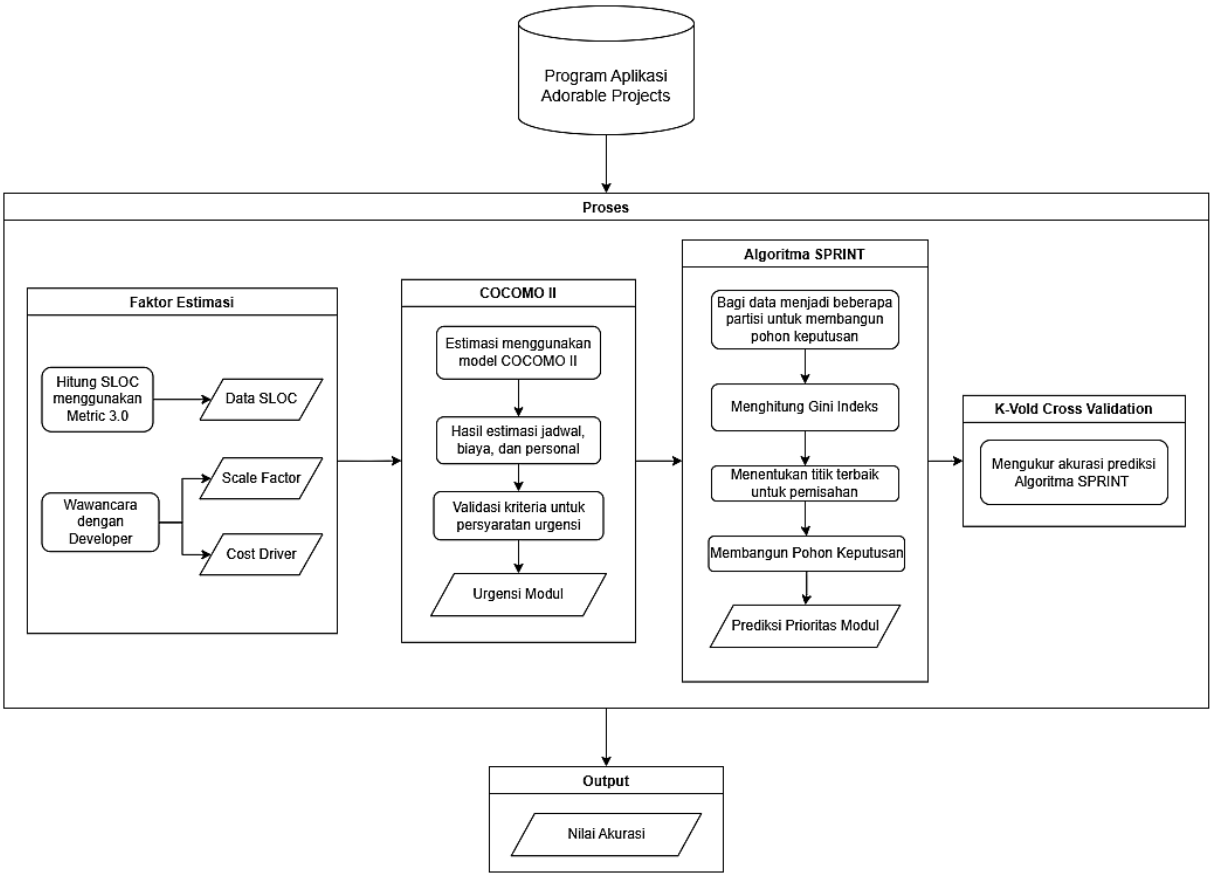
Penelitian [2] membuktikan bahwa dengan menggunakan analogi berbasis Fuzzy yang digabungkan dengan model COCOMO, estimasi perangkat lunak menjadi lebih baik. Hal ini terbukti dalam percobaan yang dilakukan yang menunjukkan bahwa metode Fuzzy menghasilkan MMRE sebesar 32.651%, metode Neuro-Fuzzy menghasilkan MMRE sebesar 56.46%, sedangkan model penggabungan Fuzzy-COCOMO menghasilkan MMRE 59.34%.

Pada penelitian [9] yang membahas estimasi upaya proyek aplikasi *Point of Sales* (POS) menggunakan COCOMO II sebagai model estimasi upaya dan menerapkan pendekatan Algoritma C4.5 untuk mendapatkan estimasi dengan akurasi prioritas yang lebih baik [9]. Hasil dari penelitian ini menunjukkan bahwa penggunaan COCOMO II dengan menerapkan pendekatan algoritma C4.5 untuk memprediksi prioritas modul pada estimasi upaya aplikasi POS mencapai akurasi sebesar 90% [9].

Untuk meningkatkan akurasi estimasi upaya dalam proyek pengembangan perangkat lunak, penggunaan model COCOMO II yang disertai dengan pendekatan Algoritma C4.5 sangat berpengaruh besar [10]. Namun, Algoritma SPRINT yang merupakan algoritma perbaikan dari Algoritma C4.5, memiliki kecepatan, skalabilitas, fleksibilitas dan akurasi yang lebih baik [14]. Dengan begitu, penggabungan COCOMO II dengan Algoritma SPRINT memiliki potensi untuk menghadirkan estimasi upaya yang lebih akurat dalam proyek pengembangan perangkat lunak, membuka jalan bagi penelitian lebih lanjut dalam memanfaatkan teknologi terkini untuk meningkatkan manajemen upaya secara menyeluruh.

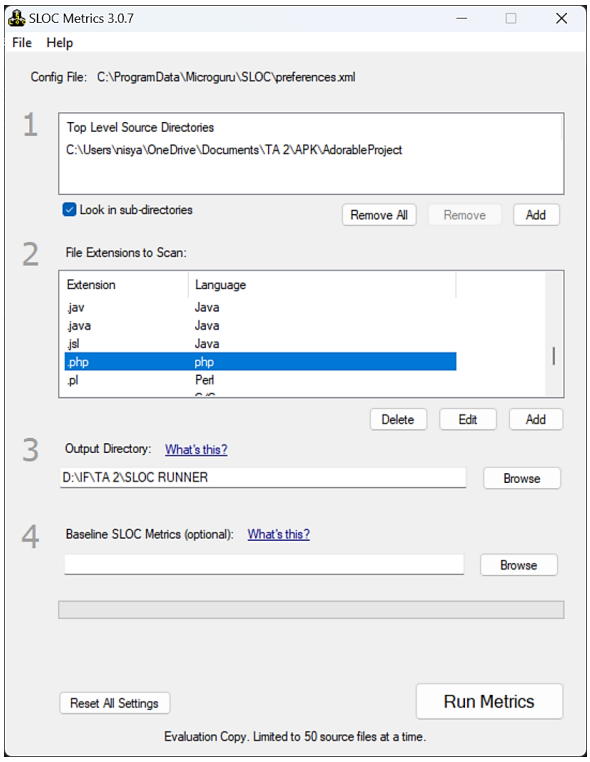
3. METODOLOGI

Penelitian dilakukan dengan menghitung data *Line of Code (LOC)* proyek aplikasi *online shop* dengan menggunakan aplikasi SLOC Metric 3.0, dilanjutkan dengan analisis estimasi dengan menggunakan model COCOMO II, lalu model COCOMO diklasifikasikan menggunakan Algoritma SPRINT untuk mendapatkan hasil estimasi akhir. Alur penelitian digambarkan seperti diagram pada Gambar 1. Aplikasi *online shop* yang digunakan adalah aplikasi berbasis website, Adorable Projects (AP). AP merupakan sebuah merek dagang yang menjual berbagai jenis alas kaki, tas, dan aksesoris lainnya.



Gambar 1 Alur Penelitian

Faktor estimasi yang digunakan untuk estimasi proyek aplikasi AP dengan menggunakan model COCOMO II adalah data SLOC dari aplikasi AP, *Scale factor*, dan juga *Effort multipliers*. SLOC berasal dari *Line of Code* (LOC) modul-modul aplikasi AP, yaitu modul *address,* modul *admin,* modul *cart,* modul *category,* modul *checkout,* modul *customer,* modul *order,* modul *product,* serta modul *shop*. Untuk menghasilkan SLOC, data dari modul-modul aplukasi AP ini diolah menggunakan aplikasi SLOC *Metric* 3.0 seperti terlampir pada Gambar 2.



Gambar 2 Aplikasi SLOC Metric 3.0

Semua *Line of Code* dari modul aplikasi AP dihitung menggunakan aplikasi Metric 3.0 untuk kemudian akan digunakan sebagai *input* utama dalam perhitungan estimasi COCOMO II. Tabel 3 dan Tabel 4 berurutan menunjukkan data SLOC dari modul *address* dan total SLOC pada setiap modil aplikasi AP.

Tabel 3 Data SLOC Modul *Address*

File	***SLOC***
AddressFormat.php	426
Address.php	352
AbstractAddressHandler.php	38
AbstractCustomerAddressHandler.php	24
AddressChecksumCore.php	22
AddressFormatter.php	16
EditOrderAddressHandler.php	118
EditCartAddressHandler.php	111
EditCustomerAddressHandler.php	98
EditManufacturerAddressHandler.php	75
AddCustomerAddressHandler.php	50
AddManufacturerAddressHandler.php	46
BulkDeleteAddressHandler.php	29
SetRequiredFieldsForAddressHandler.php	23
DeleteAddressHandler.php	19
GetCustomerAddressForEditingHandler.php	59
GetManufacturerAddressEditingHandler.php	32
GetRequiredFieldsForAddressHandler.php	15
Total	**1533**

Tabel 4 Total SLOC Setiap Modul Aplikasi AP

Modul	**Total SLOC**
Address	1659
Admin	17612
Cart	7461
Category	3084
Checkout	1328
Customer	2672
Order	10574
Product	17834
Shop	2475

Selanjutnya, untuk mendapatkan *scale factor* dan *effort multipliers*, harus dilakukan pengambilan data melalui wawancara dengan pengembang aplikasi AP. Setiap pertanyaan didasarkan dari faktor-faktor pada *scale factor* dan *effort multipliers*. Tabel 5 dan Tabel 6 berurutan menunjukkan *scale factor* dan *effort multipliers* untuk setiap modul aplikasi AP.

Tabel 5 *Scale Factor* Modul Aplikasi AP

Drivers	**Modul**								
	Product	***Admin***	***Order***	***Cart***	***Category***	***Customer***	***Address***	***Shop***	***Checkout***
PREC	H	H	H	H	H	H	H	H	H
FLEX	H	H	L	N	N	N	N	L	N
RSEL	N	N	H	H	H	H	H	N	H
TEAM	H	H	H	H	N	N	N	N	N
PMAT	H	VH	H	H	H	H	H	H	H
∑SF	**14,06**	**12,5**	**14,67**	**13,66**	**14,76**	**14,76**	**14,76**	**15,98**	**14,76**

DOI http://dx.doi.org/10.36448/expert.v14i1.3770
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 1
June 30, 2024 – pg. 67

Tabel 6 *Effort Multipliers* Modul Aplikasi AP

Drivers	Modul								
	Product	***Admin***	***Order***	***Cart***	***Category***	***Customer***	***Address***	***Shop***	***Checkout***
RELY	N	N	N	N	N	N	N	N	N
DATA	H	L	N	N	L	H	H	N	L
CPLX	VH	VH	H	L	L	L	L	L	L
RUSE	VH	H	H	H	H	H	H	H	H
DOCU	N	N	N	N	N	N	N	N	N
TIME	H	VH	H	N	N	N	N	H	N
STOR	VH	H	VH	N	N	N	N	N	N
PVOL	L	L	L	L	L	L	L	L	L
ACAP	H	VH	H	H	L	L	L	N	L
PCAP	VH	VH	H	H	H	H	H	H	H
PCON	H	N	N	N	N	N	N	N	N
APEX	VH	VH	VH	VH	VH	VH	VH	VH	VH
PLEX	VH	VH	VH	VH	VH	VH	VH	VH	VH
LTEX	N	H	N	N	L	L	L	N	L
TOOL	VH	VH	VH	VH	VH	VH	VH	VH	VH
SITE	L	L	L	L	L	L	L	L	L
SCED	VH	VH	N	N	N	N	N	N	N
$\prod_{i=1}^{17}$ EMi	**0,676**	**0,480**	**0,704**	**0,403**	**0,553**	**0,701**	**0,701**	**0,463**	**0,487**

Setelah SLOC, *Scale factor*, dan *Effort multipliers* telah dihasilkan, hal selanjutnya yang dapat dilakukan adalah memulai proses estimasi menggunakan COCOMO II. Untuk memulai perhitungan estimasi jadwal, personel, dan biaya, pertama dilakukan perhitungan estimasi upaya dengan menggunakan Rumus 1. Berikut sampel perhitungan untuk estimasi upaya pada modul *product* aplikasi AP.

$$PM = A.Size^{E}.\prod_{i=1}^{17} EMi$$
$$PM = 2{,}94 \times 17{,}834^{1{,}0506} \times 0{,}67655160$$
$$PM = 2{,}94 \times 20{,}63297 \times 0{,}67655160$$
$$PM = 40{,}9775$$

Selanjutnya, hasil estimasi upaya ini akan digunakan untuk menghitung estimasi jadwal pada modul *product* dengan menggunakan Rumus 3. Berikut merupakan sampel perhitungan untuk estimasi jadwal pada modul *product*.

$$TDEV = C \times PM^{F}$$
$$TDEV = 3{,}67 \times 40{,}9775^{0{,}28}$$
$$TDEV = 3{,}67 \times 2{,}8282$$
$$TDEV = 10{,}3795$$

Hasil estimasi jadwal ini akan digunakan untuk menghitung estimasi personel pada modul *product* dengan menggunakan Rumus 4. Berikut merupakan sampel perhitungan untuk estimasi personel pada modul *product*.

$$P = \frac{PM}{TDEV} = \frac{40{,}9775}{10{,}3795} = 3.9479$$

Hasil estimasi personel ini akan digunakan untuk menghitung estimasi biaya pada modul *product* dengan menggunakan Rumus 5. Berikut merupakan sampel perhitungan untuk estimasi biaya pada modul *product*.

$$Cost = P \times Avg\ Labor\ Cost$$
$$Cost = 3{,}94790 \times 5.300.000$$
$$Cost = 20.923.877$$

Sehingga dapat disimpulkan bahwa estimasi upaya pada proyek aplikasi *Online shop* yang akan dilakukan selanjutnya untuk estimasi jadwal, personel, dan biaya pada modul *product* akan menghabiskan waktu selama kurang lebih 10 bulan, membutuhkan tenaga kerja sebanyak kurang lebih 4 orang, dan biaya sebesar kurang lebih Rp 20.923.877 berdasarkan perhitungan dari aplikasi AP.

Setelah proses sebelumnya selesai, selanjutnya dilakukan validasi kriteria jadwal, personel, dan biaya pada setiap modul berdasarkan kondisi yang ditetapkan seperti pada Tabel 7. Kondisi ini didasarkan pada upaya yang telah dilakukan pada proyek Aplikasi AP dengan data didapat dari hasil wawancara kepada pengembang aplikasi AP. Proses ini dilakukan untuk menentukan prioritas pengerjaan setiap modul.

Tabel 7 Kriteria Validasi

No	Kondisi	Kriteria
1	Jadwal ≤ 6 Bulan	Y
2	Jadwal > 6 Bulan	X
3	Personel ≤ 2 Orang	Y
4	Personel > 2 Orang	X
5	Biaya ≤ Rp 7.000.000,00	Y
6	Biaya > Rp 7.000.000,00	X

Kriteria *Y* dan *X* memiliki arti di mana jika semua upaya menghasilkan kriteria *Y,* maka urgensi pengerjaan

DOI http://dx.doi.org/10.36448/expert.v14i1.3770
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 1
June 30, 2024 – pg. 68

modul tersebut semakin rendah, sedangkan jika semua upaya menghasilkan kriteria *X,* maka urgensi pengerjaan modul tersebut menjadi semakin tinggi.

Selanjutnya untuk proses estimasi menggunakan Algoritma SPRINT, langkah pertama yang dilakukan adalah dengan partisi data yang bertujuan untuk membagi *dataset* besar menjadi beberapa *subset* yang lebih kecil, sehingga setiap *subset* dapat diproses secara paralel. Proses ini membantu mempercepat proses pembelajaran dan mengurangi kompleksitas komputasi. Metode pembagian data yang digunakan pada penelitian ini adalah dengan menggunakan pembagian berdasarkan atribut, pada hal ini atribut yang digunakan adalah atribut prioritas.

Selanjutnya dilakukan perhitungan Gini indeks untuk menentukan *split point* terbaik, yang kemudian akan digunakan sebagai *node* dari pohon keputusan yang akan dibangun. Hasil dari pohon keputusan yang dibangun kemudian dievaluasi menggunakan *K-Vold Cross Validation.*

4. HASIL DAN PEMBAHASAN

Berdasarkan perhitungan estimasi jadwal, personel, dan biaya yang diperoleh dari model COCOMO II, maka pada Tabel 8 ditunjukkan data estimasi jadwal dalam bulan, personel, dan biaya untuk setiap modul aplikasi AP.

Tabel 8 Estimasi Jadwal, Biaya, dan Personel

Modul	Jadwal (Bulan)	Personel	Biaya (Rp/Bulan)
Address	5,0369	0,6150	3.259.693
Admin	9,2825	2,9622	15.699.848
Cart	6.6912	1.2766	6.766.079
Category	5,6644	0,8318	4.408.591
Customer	4,4137	0,4379	2.320.945
Checkout	5,8004	0,8841	4.685.909
Order	8,7215	2,5235	13.374.306
Product	10,3796	3,9479	20.923.877
Shop	5,2490	0,6839	3.624.463

Hasil estimasi jadwal, personel, dan biaya ini kemudian divalidasi dengan menggunakan kriteria validasi pada Tabel 7 untuk mengetahui kesesuaian hasil dari jadwal, personel, dan biaya proyek. Sehingga didapat data seperti pada Tabel 9 yang digunakan sebagai bahan untuk menentukan urgensi dan keputusan proyek.

Dari Tabel 9 di atas, didapatkan modul yang memiliki urgensi pengerjaan paling tinggi adalah modul admin, order dan *product*, karena pada jadwal, personel, dan biaya modul ini memiliki kriteria X. Sedangkan untuk modul yang memiliki urgensi pengerjaan rendah adalah modul *address, cart, category, checkout, customer,* dan *shop*.

Hasil estimasi jadwal, personel, dan biaya yang telah divalidasi akan digunakan sebagai data *training sample* pada algoritma SPRINT. Proses klasifikasi algoritma SPRINT dimulai dengan membagi data menjadi *subset* yang lebih kecil untuk mendapatkan *node* root pada pohon keputusan. Metode pembagian data yang digunakan pada penelitian ini adalah dengan menggunakan pembagian berdasarkan atribut, pada hal ini atribut yang digunakan adalah atribut urgensi. Pada atribut ini terdapat dua jenis data, yaitu YA dengan jumlah *instance* sebanyak tiga buah dan TIDAK dengan jumlah *instance* sebanyak 4 buah.

Tabel 9 Persyaratan Urgensi

Modul	Konfir Jadwal	Konfir Personel	Konfir Biaya	Urgensi
Address	Y	Y	Y	TIDAK
Admin	X	X	X	YA
Cart	X	Y	Y	TIDAK
Category	Y	Y	Y	TIDAK
Checkout	Y	Y	Y	TIDAK
Customer	Y	Y	Y	TIDAK
Order	X	X	X	YA
Product	X	X	X	YA
Shop	Y	Y	Y	TIDAK

Jadi, pada atribut urgensi terdapat proporsi YA = 3/9 = 1/3 dan proporsi TIDAK = 6/9 = 2/3. Sehingga, selanjutnya dapat dilakukan perhitungan indeks *Gini* sebelum proses pemisahan, proses perhitungan dilakukan seperti di bawah ini.

$$Gini(S) = 1 - (p_{YA}^2 + p_{TIDAK}^2)$$
$$Gini(S) = 1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right)$$
$$Gini(S) = 1 - \left(\left(\frac{1}{9}\right) + \left(\frac{4}{9}\right)\right)$$
$$Gini(S) = 1 - \left(\frac{5}{9}\right)$$
$$Gini(S) = 1 - 0{,}556$$
$$Gini(S) = 0.4444$$

Dari perhitungan di atas, maka didapatkan indeks Gini sebelum proses pemisahan untuk *node root* adalah sebesar 0,4444. Selanjutnya dilakukan perhitungan indeks Gini setelah pemisahan, yang dilakukan pada semua atribut yang ada, yaitu jadwal, personel, dan biaya. Pada ketiga atribut ini, terdapat dua jenis data berbeda yaitu *X* dan *Y*.

Pada atribut jadwal, terdapat total 4 *instance* untuk jadwal = *X,* dengan 3 YA dan 1 TIDAK, maka proporsi YA = 3/4 dan proporsi TIDAK = 1/4. Sehingga, indeks *gini* setelah pemisahan untuk jadwal = X adalah 0.375. Sedangkan pada jadwal = *Y* terdapat 5 *instance* dengan proporsi 0 YA dan 5 TIDAK, sehingga karena pada salah satu simpul memiliki proporsi 0, maka indeks Gini setelah pemisahan untuk jadwal = *Y* adalah 0. Selanjutnya, dilakukan perhitungan indeks Gini keseluruhan setelah pemisahan dengan menggunakan rumus sebagai berikut:

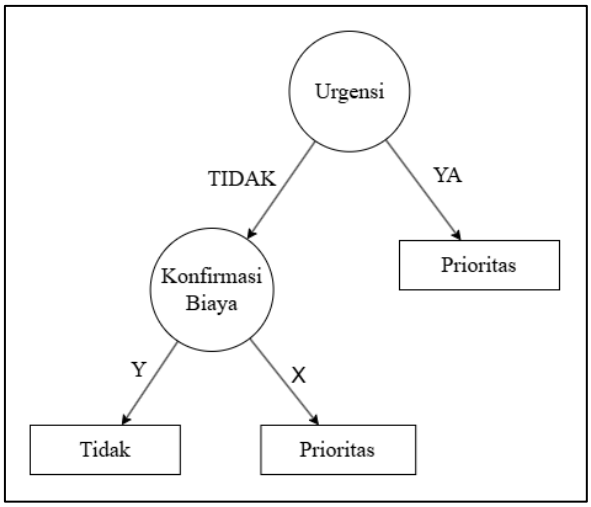
$$Gini_{Split} = \frac{4}{9} \times Gini(X) + \frac{5}{9} \times Gini(Y)$$
$$Gini_{Split} = \frac{4}{9} \times 0{,}375 + \frac{5}{9} \times 0$$
$$Gini_{Split} = \frac{4}{9} \times 0{,}375$$
$$Gini_{Split} = 0{,}1667$$

DOI http://dx.doi.org/10.36448/expert.v14i1.3770
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 1
June 30, 2024 – pg. 69

Setelah semua atribut selesai dihitung, maka didapatkan indeks Gini *split* pada atribut jadwal sebesar 0,3428, sedangkan atribut personel dan biaya memiliki indeks Gini *split* sebesar 0. Sehingga, atribut personel atau biaya merupakan atribut yang paling cocok untuk pemisahan pertama dalam pembangunan pohon keputusan.

Pada penelitian ini, atribut biaya diambil sebagai pemisahan pertama, didasarkan pada pertimbangan bahwa biaya merupakan salah satu komponen paling penting dalam proyek pengembangan aplikasi. Sehingga hasil perhitungan Gini indeks ini menghasilkan pohon keputusan seperti pada Gambar 3.

Pohon keputusan pada Gambar 3 berfungsi sebagai acuan untuk proses penentuan prioritas pengerjaan modul aplikasi AP berdasarkan dua faktor, yaitu urgensi dan konfirmasi biaya. Jika modul memenuhi persyaratan urgensi, maka pengerjaan modul tersebut menjadi prioritas. Namun, jika tidak memenuhi, langkah berikutnya adalah mengonfirmasi biaya, jika biaya bernilai *Y* maka modul tersebut tidak menjadi prioritas. Sebaliknya, jika biaya bernilai *X*, modul pengerjaan tersebut akan menjadi prioritas.



Gambar 3 Pohon Keputusan Prediksi Prioritas

Berdasarkan Gambar 3, maka didapat keputusan proyek untuk prioritas pengerjaan modul proyek aplikasi AP, keputusan proyek ini ditunjukkan seperti pada Tabel 10.

Tabel 10 Keputusan Proyek

Modul	Konfirmasi Jadwal	Konfirmasi Personel	Konfirmasi Biaya	Urgensi	Prioritas
Address	Y	Y	Y	TIDAK	Rendah
Admin	X	X	X	YA	Tinggi
Cart	X	Y	Y	TIDAK	Tinggi
Category	Y	Y	Y	TIDAK	Rendah
Checkout	Y	Y	Y	TIDAK	Rendah
Customer	Y	Y	Y	TIDAK	Rendah
Order	X	X	X	YA	Tinggi
Product	X	X	X	YA	Tinggi
Shop	Y	Y	Y	TIDAK	Rendah

Pada Tabel 9 di atas, modul *chart* yang sebelumnya memiliki urgensi TIDAK yang artinya urgensi pengerjaan modul ini adalah rendah, berubah menjadi tinggi setelah dilakukan prediksi menggunakan Algoritma SPRINT. Sehingga, modul-modul yang memiliki prioritas pengerjaan tinggi sekarang berubah dari hanya admin, order, dan *product* menjadi admin, *chart*, *order*, dan *product*.

Hasil dari pohon keputusan yang telah dibuat ini kemudian dievaluasi akurasinya dengan menggunakan metode *K-Fold Cross Validation*. Evaluasi dilakukan dengan $k = 3$, dan hasil evaluasi menunjukkan nilai akurasi untuk setiap *fold* mencapai 100%, sehingga rata-rata akurasi dari pohon keputusan yang telah dibuat adalah 100% dengan presisi dan *recall* sebesar 66,67%.

5. KESIMPULAN

Akurasi estimasi sumber daya merupakan faktor penting dalam keberhasilan pengembangan perangkat lunak. Penelitian ini bertujuan untuk meningkatkan akurasi estimasi sumber daya proyek aplikasi *online shop* dengan menggunakan kombinasi model COCOMO II dengan pendekatan Algoritma SPRINT. Dengan kemampuan untuk terus menerus memproses dan menganalisis data baru, model yang dihasilkan dari algoritma SPRINT dapat diperbarui secara berkala, sehingga algoritma ini dapat membantu model COCOMO II dalam membuat estimasi yang lebih dinamis dan adaptif, sesuai dengan kebutuhan perangkat lunak. Model COCOMO II digunakan untuk menghitung estimasi waktu, personel, dan biaya proyek. Sedangkan Algoritma SPRINT digunakan untuk memprediksi prioritas pengerjaan modul berdasarkan hasil estimasi COCOMO II.

Penelitian ini membandingkan akurasi estimasi upaya dengan penelitian sebelumnya yang menggunakan model COCOMO II dengan pendekatan Algoritma C4.5 yang merupakan pendahulu dari algoritma SPRINT pada proyek aplikasi *Point of Sales*. Estimasi proyek aplikasi menggunakan model COCOMO II dengan pendekatan Algoritma SPRINT menghasilkan akurasi 100%, lebih tinggi dibandingkan model COCOMO II dengan pendekatan algoritma C4.5 yang hanya mencapai 90%.

DOI http://dx.doi.org/10.36448/expert.v14i1.3770
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 1
June 30, 2024 – pg. 70

Penelitian ini dapat membuktikan bahwa penggunaan Algoritma SPRINT dapat meningkatkan kecepatan dan akurasi prediksi dibandingkan dengan C4.5. Sehingga, hasil dari penelitian ini dapat dijadikan sebagai acuan untuk estimasi proyek perangkat lunak di masa yang akan datang.

Penelitian ini menggunakan satu proyek aplikasi *online shop*, yaitu *Adorable Projects* sebagai subjek penelitian. Untuk meningkatkan validitas dan generalisasi hasil, disarankan pada penelitian selanjutnya untuk menerapkan model COCOMO II dengan pendekatan algoritma SPRINT pada beberapa proyek aplikasi *online shop* sekaligus. Dengan menerapkan model ini pada berbagai proyek aplikasi online shop, hal ini akan memberikan gambaran yang lebih komprehensif mengenai kinerja model COCOMO II dengan Algoritma SPRINT dan memperkuat temuan penelitian dengan meningkatkan generalisasi hasil ke berbagai situasi nyata dalam pengembangan perangkat lunak.

6. UCAPAN TERIMA KASIH

Penyelesaian penelitian ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, ucapan terima kasih yang sebesar-besarnya disampaikan kepada *Adorable Projects* yang telah mengizinkan penelitian ini dilakukan di perusahaan mereka dan mengizinkan penggunaan program aplikasinya. Kesempatan untuk mengakses program aplikasi yang sedang berjalan ini sangat berharga dalam memahami aspek praktis pengembangan aplikasi *online shop* dan mengumpulkan data aktual untuk penelitian ini.

DAFTAR PUSTAKA

[1] Badan Pusat Statistik, *Statistik eCommerce 2022/2023*. 2023.

[2] D. Manikavelan and R. Ponnusamy, "Software quality analysis based on cost and error using fuzzy combined COCOMO model," *J Ambient Intell Humaniz Comput,* 2020, doi: 10.1007/s12652-020-01783-9.

[3] D. Laurie Hughes, Yogesh K. Dwived, Antonis C. Simintiras, and Nripendra P. Rana, *Success and Failure of IS/IT Projects,* 1st ed. Springer Cham, 2016. doi: https://doi.org/10.1007/978-3-319-23000-9.

[4] M. Y. Kotowaroo and R. K. Sungkur, "Success and Failure Factors Affecting Software Development Projects from IT Professionals' Perspective," in *Soft Computing for Security Applications,* G. Ranganathan, X. Fernando, and S. Piramuthu, Eds., Singapore: Springer Nature Singapore, 2023, pp. 757–772.

[5] P. Chatzipetrou, E. Papatheocharous, L. Angelis, and A. S. Andreou, "A multivariate statistical framework for the analysis of software effort phase distribution," *Inf Softw Technol*, vol. 59, pp. 149–169, Mar. 2015, doi: 10.1016/j.infsof.2014.11.004.

[6] P. S. Kumar, H. S. Behera, K. Anisha Kumari, J. Nayak, and B. Naik, "Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades," *Computer Science Review*, vol. 38. Elsevier Ireland Ltd, 2020. doi: 10.1016/j.cosrev.2020.100288.

[7] S. Sabrjoo, M. Khalili, and M. Nazari, "Comparison of the accuracy of effort estimation methods," in 2015 *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, 2015, pp. 724–728. doi: 10.1109/KBEI.2015.7436134.

[8] Muneeb Ullah, Rahman Ali, Abdullah, Mukhtiar Ahmad, Tufail Khan, and Farhan Ul Mulk, "Software Cost Estimation – A Comparative Study of COCOMO-II and Bailey-Basili Models," 2019 *International Conference on Advances in the Emerging Computing Technologies (AECT).,* 2019, doi: https://doi.org/10.1109/AECT47998.2020.9194166.

[9] Kadinar Novel, Sfenrianto Sfenrianto, Windu Gata, Kaman Nainggolan, and Mochamad Wahyudi, "Specify of Estimation Using C4.5 Algorithm of Software Project Estimation at the Point of Sales Application with Cocomo II," in 2018 *4th International Conference on Science and Technology (ICST*), IEEE, 2018. doi: 10.1109/ICSTC.2018.8528638.

[10] I. C. Suherman, R. Sarno, and Sholiq, "Implementation of random forest regression for COCOMO II effort estimation," in *Proceedings - 2020 International Seminar on Application for Technology of Information and Communication: IT Challenges for Sustainability, Scalability, and Security in the Age of Digital Disruption, iSemantic 2020,* Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 476–481. doi: 10.1109/iSemantic50169.2020.9234269.

[11] I. Kaur, G. S. Narula, R. Wason, V. Jain, and A. Baliyan, "Neuro fuzzy—COCOMO II model for software cost estimation," *International Journal of Information Technology,* vol. 10, no. 2, pp. 181–187, 2018, doi: 10.1007/s41870-018-0083-6.

[12] Liu Caili, "Research on Classification Algorithm in Data Mining," in 2019 *International Conference on Advanced Manufacturing, Computation and Optimization (AMCO 2019),* The Academy of Engineering and Education (AEE), 2019. doi: 10.35532/jces.v1.012.

[13] Z. ÇETİNKAYA and F. HORASAN, "Decision Trees in Large Data Sets," *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi,* vol. 13, no. 1, pp. 140–151, Jan. 2021, doi: 10.29137/umagd.763490.

[14] P. Sharma and J. Singh, "Systematic Literature Review on Software Effort Estimation Using Machine Learning Approaches," in *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS),* 2017, pp. 43–47. doi: 10.1109/ICNGCIS.2017.33.

[15] Ian Sommerville, *Software Engineering 7*. New York, Harlow, England: Pearson/Addison-Wesley, 2004. Accessed: May 09, 2024. [Online]. Available: https://ifs.host.cs.st-andrews.ac.uk/Books/SE7/SampleChapters/ch26.pdf

[16] S. M. R. Chirra and H. Reza, "A Survey on Software Cost Estimation Techniques," *Journal of Software Engineering and Applications,* vol. 12, no. 06, pp. 226–248, 2019, doi: 10.4236/jsea.2019.126014.

[17] Barry W. Boehm et al., *Software Cost Estimation with COCOMO II*. Prentice Hall, 2000.

[18] V. Sheth, U. Tripathi, and A. Sharma, "A Comparative Analysis of Machine Learning Algorithms for Classification Purpose," in *Procedia Computer Science,* Elsevier B.V., 2022, pp. 422–431. doi: 10.1016/j.procs.2022.12.044.