OPTIMASI METODE *AVERAGING ENSEMBLE MODEL* DENGAN *PARTICLE SWARM OPTIMIZATION UNTUK* ESTIMASI *SOFTWARE EFFORT*

THESIS

Oleh .
ACHMAD FAHREZA ALIF PAHLEVI
NIM. 230605210013

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

OPTIMASI METODE *AVERAGING ENSEMBLE MODEL* DENGAN *PARTICLE SWARM OPTIMIZATION UNTUK* ESTIMASI *SOFTWARE EFFORT*

THESIS

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)

Oleh:
ACHMAD FAHREZA ALIF PAHLEVI
NIM. 230605210013

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

HALAMAN PERSETUJUAN

OPTIMASI METODE AVERAGING ENSEMBLE MODEL DENGAN PARTICLE SWARM OPTIMIZATION UNTUK ESTIMASI SOFTWARE EFFORT

THESIS

Oleh:
ACHMAD FAHREZA ALIF PAHLEVI
NIM. 230605210013

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 5 Desember 2024

Pembimbing I,	Pembimbing II,
Dr.Ir.Mokhammad Amin Hariyadi,M.T	Dr. Agung Teguh Wibowo Almais, M.T.
NIP. 19670118 200601 1 001	NIP. 19860301 202121 1 016

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

ii

HALAMAN PENGESAHAN

OPTIMASI METODE AVERAGING ENSEMBLE MODEL DENGAN PARTICLE SWARM OPTIMIZATION UNTUK ESTIMASI SOFTWARE EFFORT

THESIS

Oleh:
ACHMAD FAHREZA ALIF PAHLEVI
NIM. 230605210013

Telah Dipertahankan di Depan Dewan Penguji Thesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 5 Desember 2024

Susunan Dewan Penguji

Ketua Penguji	: Dr. Cahyo Crysdian NIP. 19740424 200901 1 008	( )
Anggota Penguji I	: Dr. M. Ainul Yaqin, M.Kom NIP. 19761013 200604 1 004	( )
Pembimbing I	: Dr.Ir.Mokhammad Amin Hariyadi,M.T NIP. 19670018 200501 1 001	( )
Pembimbing II	: Dr. Agung Teguh Wibowo Almais M.T NIP. 19860301 202121 1 016	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

iii

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Achmad Fahreza Alif Pahlevi
NIM	: 230605210013
Fakultas / Prodi	: Sains dan Teknologi / Magister Informatika
Judul Skripsi	: Optimasi metode Averaging Ensemble model dengan Particle Swarm Optimization untuk Estimasi Sofrtware Effort.

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang,
Yang membuat pernyataan,

Achmad Fahreza Alif Pahlevi
NIM. 230605210013

iv

MOTTO

Dalam keadaan apapun, jangan menyesali apa yang telah kita lakukan.

"Under any circumstances, do not regret what we have done."

HALAMAN PERSEMBAHAN

Saya persembahkan karya ini kepada:
Ayah saya,

Achmad Faisal

Yang telah mendukung dan menyemangati saya hingga sampai titik ini

Bunda saya,

Yuliatiningsih

Yang telah mendukung dan menyemangati saya hingga sampai titik ini

Saudara saya,

Achmad Harsya Bachtiar Sani

Yang telah mendukung dan menyemangati saya hingga sampai titik ini

Semua teman-teman seperjuangan,

Magister Informatika Angkatan 8

Semua Tim Departemen Direksi dan Developer,

Ekata Tech Indonesia

Sahabat-sahabat seperjuangan saya,

Galan Ramadhan Harya Galib, Aldian Faizzul Anwar, Abdurrozzaq Ashshidiqi Zuhri, Revaldi Rahmatmulya, Fahrendra Khoirul Ihtada, Aji Bagas Prakasa, Muhammad Haris Ibrahim, dan sahabat-sahabat lainnya yang tidak bisa saya sebutkan satu persatu.

Semoga kita semua selalu diberi kemudahan oleh Allah SWT

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Segala puji hanya milik Allah Subhanahu Wa Ta'ala atas segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan Thesis yang berjudul "Optimasi metode Averaging Ensemble model dengan Particle Swarm Optimization untuk Estimasi Sofrtware Effort". Semoga shalawat dan salam senantiasa terlimpah kepada Nabi Muhammad Sallalahu 'Alaihi wa Sallam. Dan semoga kita semua mendapat syafaatnya di hari kiamat nanti, Aamiin.

Penulis mengucapkan rasa syukur dan terima kasih yang tak terhingga kepada semua pihak-pihak yang selalu memberikan bantuan dan motivasi kepada penulis untuk dapat menyelesaikan Thesis ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr.Ir. Mokhammad Amin Hariyadi, M.T selaku dosen pembimbing I dan Dr. Agung Teguh Wibowo Almais, M.T selaku dosen pembimbing II yang telah memberikan bantuan dan arahan kepada penulis, sehingga bisa menuntaskan Thesis ini.

5. Dr. Cahyo Crysdian dan dan Dr. M. Ainul Yaqin, M.Kom selaku dosen penguji yang telah menguji serta memberikan masukan sehingga penulis dapat menuntaskan Thesis dengan baik.
6. Segenap Dosen, Admin, Laboran dan Mahasiswa Program Studi Magister Informatika yang telah memberikan banyak dukungan dan bimbingan selama pengerjaan Thesis ini.
7. Bunda, Ayah, serta saudara saya yang selalu memberikan dukungan dan motivasi untuk terus berusaha, dan doa yang tak putus-putusnya selalu disampaikan agar dapat menuntaskan Thesis ini dengan lancar dan baik.

Akhir kata, penulis mengakui bahwa penulisan pada Thesis ini masih banyak kekurangan. Saya berharap semoga Thesis ini diterima sebagai amal ibadah yang tulus dan bermanfaat di sisi Allah Subhanahu Wa Ta'ala. Semoga karya ini menjadi bagian dari kontribusi yang tak terputus dalam rangka memperkuat dan mengembangkan ilmu pengetahuan, serta melaksanakan tugas sebagai hamba Allah yang berkomitmen.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 5 Desember 2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Pahlevi, Achmad. 2024. **Optimasi metode Averaging Ensemble model dengan Particle Swarm Optimization untuk Estimasi Sofrtware Effort.** Thesis. Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr.Ir.Mokhammad Amin Hariyadi, M.T (II) Dr.Agung Teguh Wibowo Almais, M.T.

.

Kata kunci: Software Effort Estimation, Averaging Ensemble Model, XGBoost, RF

Kemajuan pesat teknologi informasi dan komunikasi (TIK) telah mengubah cara operasional bisnis, dengan aplikasi perangkat lunak menjadi pilar utama industri modern. Namun, estimasi biaya pengembangan perangkat lunak tetap menjadi tantangan kritis bagi perusahaan teknologi di era persaingan ini. Estimasi biaya yang akurat dan responsif sangat penting untuk menjamin keberhasilan bisnis. Penelitian ini mengusulkan penggunaan Averaging Ensemble Model untuk meningkatkan akurasi estimasi usaha perangkat lunak (Software Effort Estimation, SEE) dengan memanfaatkan data penggerak biaya COCOMO. Model ini menggabungkan prediksi dari Random Forest (RF) dan XGBoost untuk mencapai kinerja yang lebih unggul. Analisis perbandingan menggunakan dataset NASA93 dan China menunjukkan bahwa Averaging Ensemble Model secara konsisten mengungguli model RF dan XGBoost secara terpisah, dengan nilai Root Mean Square Error (RMSE) masing-masing sebesar 0,0313 dan 0,0155. Model ini menunjukkan prediksi yang stabil dan mendekati nilai aktual, menegaskan keandalannya dan efektivitasnya. Penelitian di masa depan dapat mengeksplorasi penambahan model dasar lainnya serta penerapan pendekatan ini pada dataset yang lebih beragam untuk meningkatkan ketahanan dan generalisasi. Penelitian ini menegaskan potensi ensemble learning dalam menghadapi kompleksitas estimasi biaya perangkat lunak.

ABSTRACT

Pahlevi, Achmad. 2024. **Optimization of the Averaging Ensemble Model Using Particle Swarm Optimization for Software Effort Estimation.** Thesis. Master's Program of Informatics Engineering Faculty of Science and TechnologyUniversitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr.Ir.Mokhammad Amin Hariyadi, M.T (II) Dr.Agung Teguh Wibowo Almais, M.T.

The rapid advancement of information and communication technology (ICT) has transformed business operations, with software applications becoming a cornerstone of modern industries. However, estimating software development costs remains a critical challenge for technology companies in this competitive era. Accurate and responsive cost estimation is crucial to ensuring business success. This study proposes the use of an Averaging Ensemble Model to enhance the accuracy of Software Effort Estimation (SEE) by leveraging COCOMO cost driver data. The model combines predictions from Random Forest (RF) and XGBoost to achieve superior performance. Comparative analysis using the NASA93 and China datasets demonstrates that the Averaging Ensemble Model consistently outperforms RF and XGBoost as standalone models, with Root Mean Square Error (RMSE) values of 0.0313 and 0.0155, respectively. The model exhibits stable predictions closely aligned with actual values, affirming its reliability and effectiveness. Future research may explore adding other base models and applying this approach to more diverse datasets to enhance robustness and generalizability. This study highlights the potential of ensemble learning in addressing the complexities of software cost estimation.

Key words: Software Effort Estimation, Averaging Ensemble Model, XGBoost, RF.

البحث مستخلص

فهلفي، أحمد. 2024. تحسين نموذج التجميع المتوسط باستخدام تحسين سرب الجسيمات لتقدير جهد البرمجيات. أطروحة. برنامج الماجستير في هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة العلوم الإسلامية الحكومية مولانا مالك إبراهيم مالانغ.
المشرفون: (1) الدكتور المهندس محمد أمين هريادي، M.T. (2) الدكتور أجونغ تجوه وييوه أليس، .M.T

الكلمات المفتاحية: تقدير جهد البرمجيات، نموذج التجميع المتوسط، XGBoost، .RF.

التقدم السريع في تكنولوجيا المعلومات والاتصالات (ICT) قد غير من طريقة عمل الشركات، حيث أصبحت التطبيقات البرمجية حجر الزاوية في الصناعات الحديثة. ومع ذلك، لا يزال تقدير تكاليف تطوير البرمجيات تحديًا كبيرًا للشركات التكنولوجية في هذا العصر التنافسي. يعتبر التقدير الدقيق والاستجابة السريعة للتكاليف أمرًا حاسمًا لضمان نجاح الأعمال. تقترح هذه الدراسة استخدام نموذج التجميع المتوسط (Averaging Ensemble Model) لتحسين دقة تقدير جهد البرمجيات (Software Effort Estimation - SEE) من خلال الاستفادة من بيانات محركات التكلفة لنموذج . يدمج النموذج التنبؤات الناتجة عن Random Forest (RF) وXGBoost لتحقيق أداء متفوق. يوضح التحليل المقارن باستخدام مجموعتي بيانات NASA93 وChina أن نموذج التجميع المتوسط يتفوق باستمرار على نماذج RF وXGBoost المستقلة، حيث بلغت قيم Root Mean Square Error (RMSE) 0.0313 و0.0155 على التوالي. يعرض النموذج تنبؤات مستقرة وقريبة جدًا من القيم الفعلية، مما يؤكد موثوقيته وفعاليته. يمكن للبحوث المستقبلية استكشاف إضافة نماذج أساسية أخرى وتطبيق هذا النهج على مجموعات بيانات أكثر تنوعًا لتعزيز المرونة وقابلية التعميم. تسلط هذه الدراسة الضوء على إمكانات التعلم التجميعي في مواجهة تعقيدات تقدير تكاليف البرمجيات.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi (TIK) telah mengubah sektor bisnis secara drastis selama beberapa tahun terakhir. Era digital yang berkembang pesat telah merubah paradigma dalam cara perusahaan beroperasi (Karna, Gotovac, and Vicković 2020). Aplikasi perangkat lunak telah menjadi fondasi bagi berbagai sektor industri, memungkinkan mereka untuk mencapai efisiensi, memfasilitasi transaksi bisnis yang lebih kompleks, meningkatkan pengalaman pelanggan, serta mengelola dan menganalisis data dengan lebih efektif. Selain itu, aplikasi perangkat lunak juga memungkinkan komunikasi yang lebih efisien dan terintegrasi di seluruh perusahaan, dari kepentingan internal hingga eksternal (Zakaria et al. 2021)

Namun, di balik perkembangan ini, muncul tantangan kritis yang dihadapi oleh perusahaan-perusahaan teknologi. Salah satu tantangan utama adalah dalam hal menentukan estimasi biaya untuk pembuatan aplikasi perangkat lunak (Hoc et al. 2023). Dalam era persaingan yang semakin ketat, perusahaan harus dapat memberikan estimasi biaya yang akurat dan responsif kepada pelanggan mereka. Ini merupakan faktor yang sangat penting dalam menentukan keberhasilan bisnis teknologi informasi modern (Ali et al. 2023). Inilah tantangan yang harus diatasi untuk tetap bersaing di era ini.

Dalam upaya untuk mengatasi tantangan ini, penggunaan metode *Ensemble Learning* sebagai pendekatan matematis untuk menghitung estimasi *Software Effort*

muncul sebagai solusi yang menjanjikan. Dengan memanfaatkan data-data *Cost Driver* COCOMO, sistem *Machine Learning* memiliki potensi untuk memberikan estimasi harga yang lebih cepat, lebih akurat, dan lebih responsif terhadap kebutuhan pelanggan (Priya Varshini et al. 2021). Dengan demikian, proses estimasi dapat dipercepat, memberikan keunggulan yang dibutuhkan dalam pasar yang berubah dengan cepat dan penuh persaingan. dalam berbagai aspek pengembangan perangkat lunak. Ini termasuk manajemen proyek yang lebih efisien, alokasi sumber daya yang lebih baik, dan perencanaan bisnis yang lebih tepat (Rhmann, Pandey, and Ansari 2022).

Penelitian ini memiliki implikasi yang jauh lebih luas. Penggunaan metode *Ensemble Learning* untuk estimasi *Software Effort* perangkat lunak memiliki potensi untuk membuka jalan bagi pengembangan model prediksi yang lebih kompleks ini selaras dengan konsep Islam, sesuai dengan akad wakalah yang disertai adanya upah (ujrah) atau lengkapnya akad wakalah bil ujrah sesuai dengan ketentuan yang dan menjalankan pross pengiriman sesuai amanah yang dan tanggung jawab yang diberikan. Jadi transaksi pembuatan software telah berjalan sesuai amanah dan tanggung jawab yang baik, maka transaksi pembuatan software telah sesuai dengan akad wakalah bil ujrah. Adapun yang dijadikan dasar hukum wakalah adalah Firman Allah .

QS. Al-Baqarah (2). 254

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اَنْفِقُوْا مِمَّا رَزَقْنٰكُمْ مِّنْ قَبْلِ اَنْ يَّأْتِيَ يَوْمٌ لَّا بَيْعٌ فِيْهِ وَلَا خُلَّةٌ وَّلَا شَفَاعَةٌۗ وَالْكٰفِرُوْنَ هُمُ الظّٰلِمُوْنَ

Artinya . ”*Hai orang-orang yang beriman, belanjakanlah (di jalan Allah) sebagian dari rezeki yang telah Kami berikan kepadamu sebelum datang hari yang*

pada hari itu tidak ada lagi jual beli dan tidak ada lagi syafa'at. Dan orang-orang kafir itulah orang-orang yang zalim". (QS.Al-Baqarah ayat 254).

Ayat QS. Al-Baqarah (2). 254 berbicara tentang pentingnya mengeluarkan sebagian dari rezeki yang telah diberikan oleh Allah sebelum datangnya hari ketika tidak ada lagi jual beli, persahabatan, dan syafaat. Ayat ini memiliki beberapa tafsir yang dapat dikaitkan dengan penelitian tentang prediksi estimasi biaya pembuatan aplikasi. Ayat ini memerintahkan orang-orang beriman untuk mengeluarkan sebagian dari rezeki yang telah diberikan oleh Allah sebagai bentuk ketaatan kepada-Nya Dalam konteks penelitian, hal ini dapat dihubungkan dengan pentingnya mengalokasikan sumber daya yang tepat untuk memperkirakan biaya pembuatan aplikasi dengan akurat dan efektif.

Tafsir Al-Qurtubi menyatakan bahwa mengeluarkan harta di jalan Allah dapat menjadi wajib atau anjuran, tergantung pada keadaan yang menyertainya Dalam penelitian, hal ini dapat diartikan bahwa penggunaan metode estimasi biaya yang tepat harus disesuaikan dengan keadaan proyek yang sedang berjalan. Ayat ini juga memerintahkan untuk mengeluarkan harta di jalan Allah selama masih memiliki kemampuan melakukannya sebagai bekal di hari akhir kelak Dalam penelitian, hal ini dapat dihubungkan dengan pentingnya mengalokasikan sumber daya dengan bijak untuk memperkirakan biaya pembuatan aplikasi yang dapat memberikan manfaat jangka panjang.

Dalam konteks penelitian tentang prediksi estimasi biaya pembuatan aplikasi, ayat QS. Al-Baqarah (2). 254 dapat diartikan sebagai pengingat akan pentingnya mengalokasikan sumber daya dengan bijak dan mengeluarkan sebagian dari rezeki yang telah diberikan oleh Allah untuk memperkirakan biaya pembuatan

aplikasi dengan akurat dan efektif. Terdapat juga hadits mengenai pentingnya melakukan pekerjaan secara sempurna.

ءيش لك ىلع ناسحلْا بتك اللَّ نإ…

Artinya. ”*Sesungguhnya Allah SWT mewajibkan (kepada kita) untuk berbuat yang optimal dalam segala sesuatu…*”.

Berdasarkan kutipan hadits diatas, dapat dikatakan bahwa kinerja merupakan prestasi yang dicapai seseorang dalam melaksanakan tugas-tugas atau pekerjaannya sesuai dengan standar dan kriteria yang ditetapkan untuk suatu pekerjaan (Zarkasyi 2016). Ini sesuai dengan proses optimasi yang digunakan di penelitian ini Dimana tujuan dari penelitian ini adalah mendapatkan model paling optimal untuk memprediksi SEE.

Dengan dasar masalah yang telah diuraikan, penelitian ini memiliki tujuan yang sangat penting, yakni mengembangkan model machine learning yang bukan hanya efisien dan akurat dalam menghitung estimasi *Software Effort* perangkat lunak, tetapi juga mendorong inovasi serta membuka peluang baru dalam industri teknologi informasi yang terus berkembang dan berubah secara dinamis (Gautam and Singh 2022).

1.2 Pernyataan Masalah

Seberapa Efektif penggunaan metode *Averaging Ensemble Model* dengan optimasi PSO dalam mengestimasi *Software Effort*?

1.3 Batasan Masalah

Berikut batasan masalah yang digunakan dalam penelitian ini.

1. Data yang digunakan di penelitian ini yaitu China *Software Effort Estimation Dataset* dan Promise Nasa93 *Dataset*
2. Data hanya digunakan untuk memprediksi *Software Effort Estimation* dari masing-masing *Dataset*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah Membandingkan hasil prediksi model *Averaging Ensemble Learning dengan optimasi PSO* dengan *Random Forest* dan *Extreme Gradient Boosting* untuk menentukan estimasi *Software Effort.*

1.5 Manfaat Penelitian

Hasil dari penelitian ini adalah output performa dari metode *Averaging Ensemble Learning dengan optimasi PSO* untuk memprediksi estimasi biaya pembuatan *software*. Kelompok-kelompok yang bisa mendapat manfaat dari hasil penelitian ini antara lain.

1. Komunitas *Software Engineering* dengan tujuan musyawarah; dan pihak-pihak terkait.
2. Peneliti *Data Mining* untuk tujuan rujukan penelitian.

BAB II
STUDI PUSTAKA

2.1 Software Effort Estimation (SEE)

Software Effort Estimation (SEE) adalah proses untuk memperkirakan biaya, usaha, dan sumber daya yang dibutuhkan untuk mengembangkan perangkat lunak. Beberapa metode yang telah dikembangkan untuk SCE antara lain COCOMO, *Function Point Analysis* (FPA), *Use Case Point* (UCP), *Machine Learning* (ML) dan *Artificial Intelligence* (AI) (Rajper and Shaikh 2016). Dalam memilih metode estimasi biaya yang tepat, perlu mempertimbangkan karakteristik proyek, data yang tersedia, dan tujuan estimasi. Metode yang lebih kompleks seperti COCOMO dan ML/AI mungkin memerlukan lebih banyak data dan pengetahuan ahli, sedangkan metode yang lebih sederhana seperti FPA dan UCP mungkin lebih mudah digunakan tetapi kurang akurat (Chirra and Reza 2019). Faktor-faktor yang mempengaruhi estimasi biaya antara lain ketidakjelasan lingkup, kompleksitas design, dan ukuran proyek. Untuk mengurangi risiko perubahan estimasi biaya, perlu dilakukan pembaruan estimasi secara berkala berdasarkan informasi yang tersedia.

Beberapa penelitian terdahulu telah dilakukan untuk mengembangkan metode estimasi biaya yang lebih akurat dan efektif. Sebagai contoh, pada penelitian Shekhar (2018) dilakukan systematic literature review untuk mengevaluasi teknik-teknik estimasi biaya pengembangan perangkat lunak. Hasil dari penelitian ini menunjukkan bahwa teknik-teknik seperti COCOMO, FPA, dan UCP masih menjadi teknik yang paling banyak digunakan dalam SCE. Namun,

penelitian ini juga menunjukkan bahwa teknik-teknik ML dan AI seperti SVR dan ANN memiliki potensi untuk memberikan hasil yang lebih akurat.

Pada penelitian oleh Chirra and Reza (2019) dilakukan survey terhadap berbagai pendekatan estimasi biaya pengembangan perangkat lunak. Hasil dari penelitian ini menunjukkan bahwa pendekatan yang paling banyak digunakan adalah COCOMO, FPA, dan UCP. Namun, penelitian ini juga menunjukkan bahwa pendekatan ML dan AI seperti SVR dan ANN memiliki potensi untuk memberikan hasil yang lebih akurat. Kemudian riset oleh Almohimeed et al. (2023) menggunakan *Random Forest* sebagai salah satu metode dalam kombinasi dengan SVR, LR, dan KNN untuk memprediksi SEE. Hasil penelitian ini menunjukkan bahwa model kombinasi ini menghasilkan akurasi sebesar 92.08, presisi 92.07, recall 92.08, dan F1-Score sebesar 92.01.

Dari hasil penelitian-penelitian di atas, dapat disimpulkan bahwa terdapat berbagai metode yang dapat digunakan untuk melakukan estimasi biaya pengembangan perangkat lunak, dan pemilihan metode yang tepat perlu mempertimbangkan karakteristik proyek, data yang tersedia, dan tujuan estimasi. Oleh karena itu, perlu dilakukan penelitian lebih lanjut untuk mengembangkan metode estimasi biaya yang lebih akurat dan efektif.

2.2 Particle Swarm Optimization

Penelitian oleh Li et al. (2024) memperkenalkan metode PSO-RANP untuk prediksi node localization di wireless sensor network (WSN). Dalam konteks WSN, node localization merupakan tantangan penting untuk menentukan posisi node yang akurat. PSO-RANP, yang menggabungkan metode Particle Swarm Optimization

(PSO) dan Relative Angle of Neighbor Position (RANP), digunakan untuk meningkatkan akurasi prediksi lokasi node. Hasil penelitian menunjukkan bahwa metode PSO-RANP mampu mengurangi Root Mean Square *Error* (RMSE) hingga 20% lebih rendah dibandingkan dengan metode lain yang digunakan untuk masalah serupa. Dengan performa yang lebih baik ini, PSO-RANP memberikan solusi yang lebih efisien dalam mengoptimalkan akurasi node localization pada jaringan sensor nirkabel, menjadikannya alternatif yang unggul dalam skenario di mana akurasi posisi node sangat penting.

Penelitian oleh Abdo, Abdelkader, and Abdel-Hamid (2024) memperkenalkan metode SA-PSO-GK++ untuk clustering data medis. Metode ini merupakan kombinasi dari *Simulated Annealing* (SA), *Particle Swarm Optimization* (PSO), dan GK++ (Generalized K-means++) yang dirancang untuk meningkatkan akurasi dalam pengelompokan data medis yang kompleks. Hasil penelitian menunjukkan bahwa metode PSO-GB++ berhasil mencapai performa clustering yang lebih baik berdasarkan beberapa metrik evaluasi. *Specific Error Deviation* (SED) tercatat sebesar W+465*, Normalized Mutual Information* (NMI) juga sebesar W-465, dan tingkat kesalahan atau *Error Rate* sebesar W+465. Metode ini menunjukkan potensi yang besar dalam menganalisis dan mengelompokkan data medis, membantu dalam meningkatkan efisiensi dan akurasi di berbagai aplikasi medis, seperti diagnosis penyakit atau analisis genetik.

Penelitian yang dilakukan oleh Srivastava, Dwivedi, and Singh (2023) menggunakan metode PSO-XGBoost untuk memprediksi *SEE*. PSO (Particle Swarm Optimization) dikombinasikan dengan XGBoost (*Extreme Gradient*

Boosting) bertujuan untuk meningkatkan akurasi prediksi melalui optimasi *hyperparameter* dengan PSO, sementara XGBoost dikenal sebagai algoritma boosting yang sangat efisien untuk tugas regresi maupun klasifikasi. Hasil dari penelitian ini menunjukkan bahwa model PSO-XGBoost mencapai nilai *Mean Absolute Error* (MAE) sebesar 0.0714, *Mean Squared Error* (MSE) sebesar 0.0309, dan Root *Mean Squared Error* (RMSE) sebesar 0.0175. Nilai-nilai kesalahan yang rendah ini menunjukkan bahwa model PSO-XGBoost mampu memberikan prediksi yang cukup akurat.

2.3 Random Forest

Penelitian yang menggunakan metode *Random Forest* untuk prediksi *Software Effort Estimation* (SEE) menunjukkan performa yang kompetitif. (Priya Varshini et al. 2021) menerapkan *Random Forest* pada *Dataset* PROMISE Nasa93 dan berhasil mencapai *Mean Absolute Error* (MAE) sebesar 0.484 dan *Mean Squared Error* (MSE) sebesar 0.436. Hasil ini menunjukkan bahwa *Random Forest* mampu menghasilkan prediksi yang cukup akurat dalam konteks SEE.

Selain itu, penelitian lain oleh (Matlock et al. 2018) menggunakan *Random Forest* yang dikombinasikan dengan KNN Residual pada *Dataset* yang berbeda, menghasilkan korelasi sebesar 0.7504 dan NMSE sebesar 35.26. Metode ini menunjukkan bahwa penggabungan *Random Forest* dengan metode lain dapat meningkatkan akurasi prediksi SEE.

Kemudian riset oleh Almohimeed et al. (2023) menggunakan *Random Forest* sebagai salah satu metode dalam kombinasi dengan SVR, LR, dan KNN untuk memprediksi SEE. Hasil penelitian ini menunjukkan bahwa model kombinasi

ini menghasilkan akurasi sebesar 92.08, presisi 92.07, recall 92.08, dan F1-Score sebesar 92.01. *Random Forest*, sebagai bagian dari *Ensemble* ini, berkontribusi terhadap hasil yang sangat akurat dan presisi dalam memprediksi SEE.

Secara keseluruhan, *Random Forest* terbukti efektif dalam prediksi SEE, terutama karena kemampuannya dalam menangani data yang bervariasi dan mencegah overfitting. Namun, untuk hasil yang lebih optimal, tuning *hyperparameter* atau penggabungan dengan metode lain seperti yang ditunjukkan dalam penelitian Matlock et al. (2018) dapat meningkatkan performa lebih lanjut.

2.4 *Extreme* Gradient Boostring (XGBoost)

Penelitian yang menggunakan metode *Gradient Boosting* untuk prediksi *Software Effort Estimation* (SEE) menunjukkan hasil yang signifikan. Gautam and Singh (2022) menerapkan *Gradient Boosting* pada *Dataset* PROMISE Nasa93 dan berhasil mencatatkan *Mean Absolute Error* (MAE) sebesar 0.476 serta *Mean Squared Error* (MSE) sebesar 0.414. Hasil ini menunjukkan bahwa *Gradient Boosting* memiliki kemampuan yang cukup baik dalam memprediksi SEE dibandingkan beberapa metode lainnya. Namun, hasil penelitian ini menunjukkan bahwa *Gradient Boosting* masih memiliki ruang untuk perbaikan jika dibandingkan dengan beberapa metode lain seperti *Random Forest* dan Stacking *Ensemble* yang menghasilkan *Error* lebih rendah.

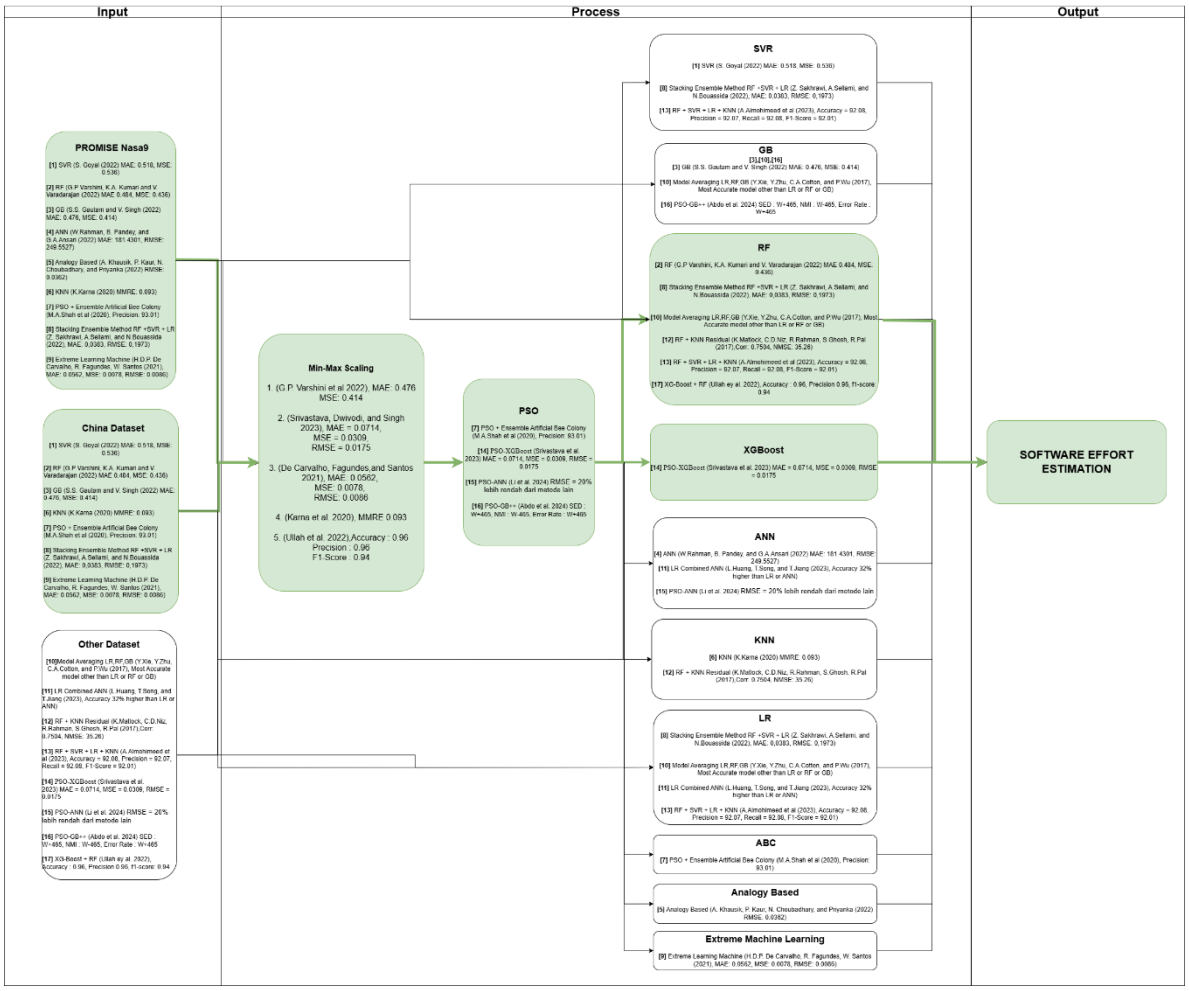
Kemudian riset oleh Srivastava et al. (2023) menggunakan metode PSO-XGBoost untuk memprediksi *SEE*. PSO (Particle Swarm Optimization) dikombinasikan dengan XGBoost (*Extreme Gradient Boosting*) bertujuan untuk meningkatkan akurasi prediksi melalui optimasi *hyperparameter* dengan PSO,

sementara XGBoost dikenal sebagai algoritma boosting yang sangat efisien untuk tugas regresi maupun klasifikasi. Hasil dari penelitian ini menunjukkan bahwa model PSO-XGBoost mencapai nilai *Mean Absolute Error* (MAE) sebesar 0.0714, *Mean Squared Error* (MSE) sebesar 0.0309, dan Root *Mean Squared Error* (RMSE) sebesar 0.0175. Nilai-nilai kesalahan yang rendah ini menunjukkan bahwa model PSO-XGBoost mampu memberikan prediksi yang akurat.

Penggunaan PSO dalam optimasi *hyperparameter* XGBoost memberikan kontribusi signifikan dalam meningkatkan performa prediktif model, terutama dalam kasus data yang kompleks seperti *cryptocurrency*, yang sangat dipengaruhi oleh volatilitas pasar. Kombinasi metode ini menunjukkan potensi yang besar dalam meningkatkan efisiensi prediksi di berbagai kebutuhan yang memerlukan estimasi yang akurat.

2.5 Kerangka Teori

Dalam melakukan penelitian tentang prediksi biaya pembuatan *software* dengan metode *Model Averaging Ensemble* perlu mengacu pada jurnal-jurnal sebelumnya yang telah melakukan sebagai kerangka teori seperti yang terlihat pada Gambar 2.1 .



Gambar 2.1 Kerangka Teori

Pada gambar 2.1 menampilkan ringkasan penelitian yang menggunakan berbagai metode untuk memprediksi *Software Effort Estimation* (SEE) berdasarkan *Dataset* PROMISE Nasa93 dan *Dataset* lainnya. Pada *Dataset* PROMISE Nasa93, metode SVR yang digunakan oleh Goyal (2022) menghasilkan MAE sebesar 0.518 dan MSE sebesar 0.536. Priya Varshini et al. (2021) menggunakan *Random Forest* (RF) dengan hasil MAE 0.484 dan MSE 0.436. Metode *Gradient Boosting* (GB) oleh Gautam and Singh (2022)menghasilkan MAE 0.476 dan MSE 0.414,

sementara ANN oleh Rahman et al. (2024) mencatat MAE yang sangat tinggi sebesar 181.4301 dan RMSE 249.5527.

Beberapa penelitian lain menggunakan pendekatan yang berbeda seperti Analogy Based oleh Khausik et al. (2022) dengan RMSE sebesar 0.0362, K-Nearest Neighbors (KNN) oleh Karna et al. (2020) dengan MMRE 0.093, serta PSO dan *Ensemble* Artificial Bee Colony oleh Shah et al. (2020) yang menghasilkan presisi sebesar 93.01. Metode Stacking *Ensemble* yang menggabungkan RF, SVR, dan LR oleh Sakhrawi et al. (2022) mencatat MAE 0.0383 dan RMSE 0.1973. Hasil lain yang signifikan diperoleh oleh De Carvalho, Fagundes, and Santos (2021)dengan metode *Extreme* Learning Machine (ELM) yang mencatat MAE 0.0562 dan RMSE 0.0086.

Pada *Dataset* lainnya, Xie et al. (2019) menggunakan Model *Averaging* dengan gabungan LR, RF, dan GB, menghasilkan model yang paling akurat dibandingkan metode tunggal. Zhang et al. (2019) menggabungkan LR dan ANN, yang menghasilkan akurasi 32% lebih tinggi dibandingkan penggunaan metode LR atau ANN secara terpisah. Matlock et al. (2018) memadukan RF dengan KNN Residual, mencapai korelasi sebesar 0.7504 dan NMSE 35.26, sedangkan Almohimeed et al. (2023) menggunakan gabungan RF, SVR, LR, dan KNN dengan hasil akurasi 92.08%, presisi 92.07%, recall 92.08%, dan F1-Score 92.01. Seluruh penelitian ini mengarah pada pengembangan prediksi SEE yang lebih akurat, khususnya dengan menggunakan metode *Averaging Ensemble*, yang berpotensi memberikan hasil prediksi yang lebih stabil dan unggul dibandingkan metode tunggal. Untuk lebih jelasnya dapat dilihat di tabel 2.1 .

Tabel 2.1 Daftar Referensi

No	Peneliti (Tahun)	Metode dan Studi Kasus	Metode Penelitian	Hasil Penelitian
1.	(S. Goyal 2022)	Metode SVR digunakan untuk memprediksi SEE	SVR	MAE. 0.518 MSE .0.536
2.	(G.P. Varshini et al 2022)	Metode *Random Forest* digunakan untuk memprediksi SEE	*Random Forest*	MAE. 0.476 MSE. 0.414
3.	(Gautam and Singh 2022)	Metode *Gradient Boosting* digunakan untuk memprediksi SEE	*Gradient Boosting*	MAE. 0.476 MSE. 0.414
4.	(Rhmann et al. 2022)	Metode ANN digunakan untuk memprediksi SEE	ANN	MAE. 181.4301 RMSE. 249.5527
5.	(Kaushik et al. 2022)	Metode Analogy Based digunakan untuk memprediksi SEE	Analogy Based	RMSE. 0.0362
6.	(Karna et al. 2020)	Metode KNN digunakan untuk memprediksi SEE	KNN	MMRE. 0.093
7.	(Shah et al. 2020)	Metode PSO + *Ensemble* Artificial Bee Colony digunakan untuk memprediksi SEE	PSO + *Ensemble* Artificial Bee Colony	Precision. 93.01
8.	(Sakhrawi et al. 2022)	Metode Stacking *Ensemble* Method RF +SVR + LR digunakan untuk memprediksi SEE	RF +SVR + LR	MAE. 0,0383, RMSE. 0,1973
9.	(De Carvalho et al. 2021)	Metode *Extreme* Learning Machine digunakan untuk memprediksi SEE	*Extreme* Learning Machine	MAE. 0.0562, MSE. 0.0078, RMSE. 0.0086
10.	(Xie et al. 2019)	Metode Model *Averaging* LR,RF,GB digunakan untuk memprediksi SEE	LR,RF,GB	Most Accurate model other than LR or RF or GB
11.	(Huang et al. 2023)	Metode LR Combined ANN digunakan untuk memprediksi SEE	LR + ANN	Accuracy 32% higher than LR or ANN)
12.	(Matlock et al. 2018)	Metode RF + KNN Residual digunakan untuk memprediksi SEE	RF + KNN Residual	Corr. 0.7504, NMSE. 35.26
13.	(Almohimeed et al. 2023)	Metode RF + SVR + LR + KNN digunakan untuk memprediksi SEE	RF + SVR + LR + KNN	Accuracy = 92.08, Precision = 92.07, Recall = 92.08, F1-Score = 92.01
14.	(Srivastava et al. 2023)	Metode PSO-XGBoost digunakan untuk prediksi *SEE*	PSO-XGBoost	MAE = 0.0714 MSE = 0.0309 RMSE = 0.0375
15.	(Li et al. 2024)	Metode PSO-RANP digunakan untuk memprediksi node localization di wireless sensor network	PSO-RANP	RMSE = 20% lebih rendah dari metode lain
16.	(Abdo et al. 2024)	Metode PSO-GB++ digunakan untuk clustering data medis	PSO-GB++	SED . W+465 NMI . W-465 *Error Rate* . W+465

17.	(Ullah et al. 2022)	Metode RF-XGB digunakan untuk data Loss detection Smart Grids	RF + XGBosst	Accuracy . 0.96 Precision . 0.96 F1-Score . 0.94

Berdasarkan Tabel 2.1 pada Penelitian terkait prediksi *Software Effort Estimation* (SEE) telah banyak menggunakan berbagai metode. Penelitian oleh S. Goyal (2022) menggunakan metode Support Vector Regression (SVR) dengan hasil MAE sebesar 0.518 dan MSE 0.536. Penelitian oleh Priya Varshini et al. (2021) dan (Gautam and Singh 2022) mengaplikasikan metode *Random Forest* dan *Gradient Boosting* yang sama-sama menghasilkan MAE sebesar 0.476 dan MSE sebesar 0.414. Penelitian oleh (Rhmann et al. 2022) yang menggunakan metode Artificial Neural Network (ANN) menghasilkan nilai MAE yang cukup tinggi, yaitu 181.4301 dan RMSE sebesar 249.5527, yang menunjukkan performa model kurang optimal.

Penelitian oleh Kaushik et al. (2022) dengan metode Analogy Based memperoleh RMSE 0.0362, sedangkan Karna et al. (2020) menggunakan metode K-Nearest Neighbors (KNN) dengan hasil MMRE 0.093. Shah et al. (2020) menerapkan kombinasi PSO dan *Ensemble* Artificial Bee Colony dengan hasil presisi mencapai 93.01. Penelitian lain oleh Sakhrawi et al. (2022) yang menggabungkan *Random Forest*, SVR, dan Linear Regression melalui metode Stacking *Ensemble* menghasilkan MAE 0.0383 dan RMSE 0.1973.

Penelitian oleh De Carvalho et al. (2021) menggunakan metode *Extreme* Learning Machine dengan hasil MAE 0.0562, MSE 0.0078, dan RMSE 0.0086. Xie et al. (2019) menyajikan metode Model *Averaging* antara Linear Regression, *Random Forest*, dan *Gradient Boosting* yang menghasilkan model paling akurat

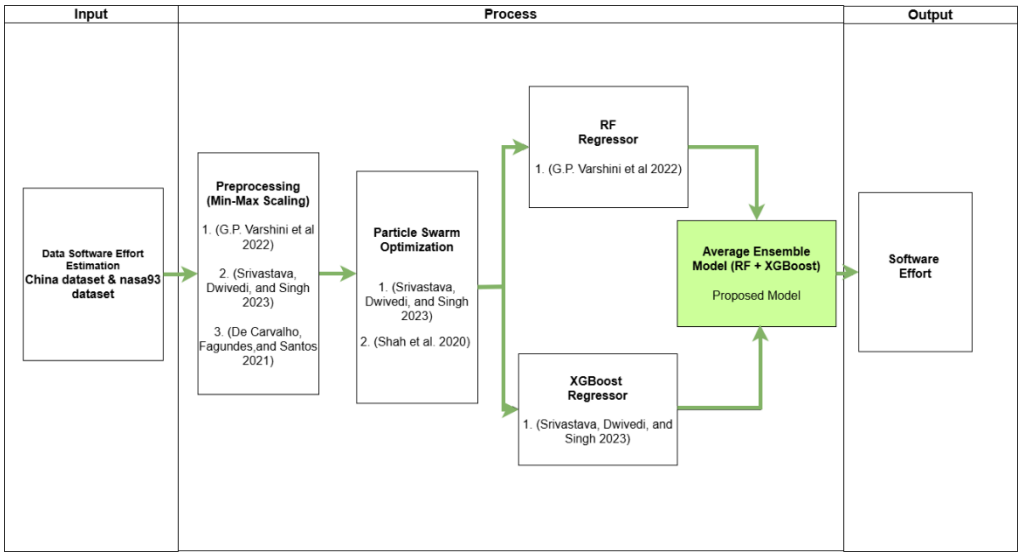
dibandingkan metode tunggal. Sementara itu, Zhang et al. (2019) menggabungkan Logistic Regression dengan ANN yang memberikan peningkatan akurasi sebesar 32% dibanding metode individu. Penelitian oleh Matlock et al. (2018) memadukan *Random Forest* dengan KNN Residual dan mencapai korelasi sebesar 0.7504 dan NMSE 35.26. Penelitian oleh Almohimeed et al. (2023) yang menggabungkan *Random Forest*, SVR, Linear Regression, dan KNN memberikan hasil akurasi sebesar 92.08%, presisi 92.07%, recall 92.08%, dan F1-score 92.01.

Karena alasan diatas maka penulis engembangkan model prediksi SEE yang lebih akurat dengan menggunakan metode *Averaging Ensemble* Learning, yang diharapkan dapat menggabungkan kekuatan berbagai model untuk menghasilkan prediksi yang lebih baik.

BAB III
METODOLOGI PENELITIAN

Pada bab ini akan membahas secara rinci tentang metode *Model Averaging Ensemble* yang digunakan untuk prediksi SEE. Selain itu, bab ini juga akan membahas tentang kerangka konsep, desain dan implementasi sistem yang digunakan dalam penelitian ini. Penelitian ini juga akan membahas tentang pengujian model dengan menggunakan data testing dan pengukuran performa model dengan menggunakan *Root Mean Squeared Error* (RMSE) (Abdollahpour, Kosari-Moghaddam, and Bannayan 2020).

3.1 Kerangka Konsep



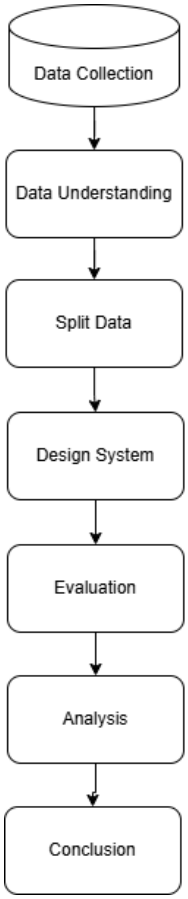
Gambar 3. 1 Kerangka Konsep

Gambar 3.1 menjelaskan kerangka konsep yang dijalankan pada penelitian ini, dimana input adalah *Software Effort Estimation* (SEE) *Dataset* China dan Nasa93, *Dataset* ini berifi informasi untuk estimasi SEE. Kemudian langkah selanjutnya, adalah Proses dimulai dari *Preprocessing* dengan menggunakan Min-

Max Scaling Dimana Proses ini menormalisasi data agar berada pada rentang 0 -1 (Rais 2022). Setelah melalui tahap *Preprocessing,* Langkah selanjutnya adalah menggunakan *Particle Swarm Optimization* (PSO) untuk mengoptimalkan parameter model. Terdapat dua model utama yaitu *Random Forest* (RF) *Regressor* dan *Extreme Gradient Boosting (XGBoost) Regressor.* Setelah model RF dan XGBoost menghasilkan prediksi, metode *Ensemble Averaging* Model digunakan untuk menggabungkan hasil prediksi kedua model tersebut, berdasarkan penelitian oleh Zhang et al. (2019) metode *Ensemble* menghasilkan akurasi 32% lebih tinggi dibandingkan model individu.

3.2 Desain Penelitian

Beberapa tahapan dalam penelitian ini adalah melakukan beberapa tahapan secara urut sehingga proses desain hingga implementasi dapat dilakukan dimulai dari hal yang mendasar hingga hasil yang optimal. Alur proses desain penelitian dapat dilihat pada Gambar 3.2.



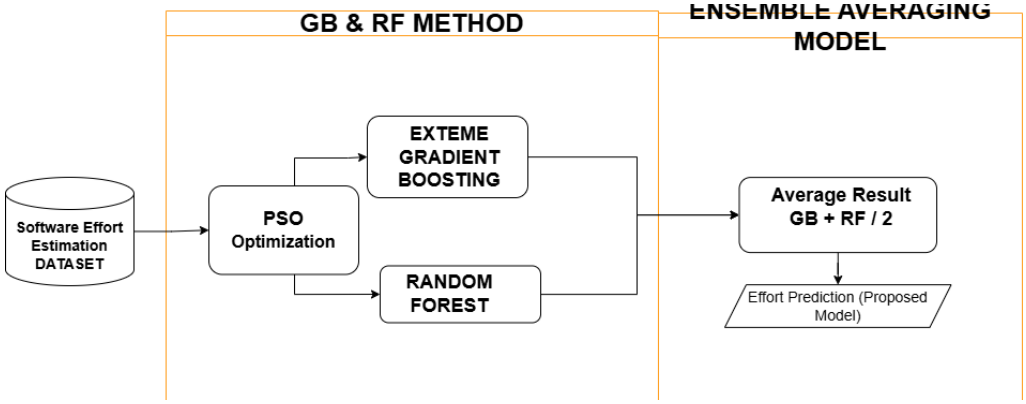
Gambar 3. 2 Desain Penelitian

Pada Gambar 3.2, Desain Penelitian menggambarkan alur proses dari pengembangan model prediksi biaya menggunakan metode *Extreme Gradient Boosting* (XGB) dan *Random Forest* (RF). Tahap pertama adalah Pengumpulan Data (*Data Collection*), di mana data yang relevan dikumpulkan dari berbagai sumber. Setelah data terkumpul, dilanjutkan dengan Preprocessing Data, yaitu tahap pembersihan dan persiapan data agar siap digunakan oleh model, seperti menangani nilai yang hilang, normalisasi, dan konversi data menjadi format yang sesuai. Selanjutnya, dilakukan Perancangan Sistem (Design Sistem) menggunakan dua metode utama, yaitu *Extreme Gradient Boosting* (XGB) dan *Random Forest* (RF).

Kedua model ini akan digunakan secara bersamaan untuk membangun sistem prediksi biaya. Setelah sistem dirancang, model diuji untuk menghitung kesalahan prediksi melalui metrik Root *Mean Squared Error* (RMSE). RMSE digunakan untuk mengevaluasi tingkat akurasi model dengan melihat seberapa besar deviasi antara nilai prediksi dan nilai sebenarnya. Tahap terakhir adalah Kesimpulan (Conclusion), di mana hasil analisis dan evaluasi disimpulkan berdasarkan performa model yang sudah dibangun, dengan fokus pada akurasi dan kesalahan prediksi yang dihasilkan oleh metode XGBoost dan RF.

3.3 Design System

Design system akan menjelaskan bagaimana sistem melakukan prediksi *Software Effort Estimation* (SEE) menggunakan metode *Ensemble Averaging Model* dan menghitung akurasi dari *Ensemble Averaging Model*. Alur proses *design* sistem dalam penelitian ini dapat dilihat pada Gambar 3.4 .



Gambar 3. 3 Design Sistem

Pada gambar 3.2 menunjukkan alur metode prediksi biaya menggunakan kombinasi *Extreme Gradient Boosting* (XGB) dan *Random Forest* (RF) yang dioptimalkan dengan Particle Swarm Optimization (PSO). Pada tahap Training, data latih digunakan untuk melatih dua model utama, yaitu *Extreme Gradient*

Boosting dan *Random Forest*. Sebelum pelatihan, model dioptimalkan dengan PSO untuk menemukan parameter terbaik yang dapat meningkatkan performa kedua model tersebut. Setelah pelatihan selesai, pada tahap Testing, data uji diterapkan pada kedua model yang telah dilatih untuk menghasilkan prediksi biaya, yaitu *Cost* Prediction (XGB) dari Exteme *Gradient Boosting* dan *Cost* Prediction (RF) dari *Random Forest*. Pada tahap akhir, kedua hasil prediksi ini digabungkan dalam model *Ensemble Averaging*, di mana hasil prediksi dari *Gradient Boosting* dan *Random Forest* dirata-ratakan dengan menggunakan formula Model *Averaging* yaitu (XGB + RF) / 2.

Hasil rata-rata dari kedua prediksi ini kemudian digunakan sebagai prediksi akhir untuk biaya, yang akhirnya dievaluasi menggunakan metrik Root *Mean Squared Error* (RMSE) untuk mengukur seberapa akurat model dalam melakukan prediksi. Pendekatan *Ensemble* ini memungkinkan model untuk memanfaatkan kelebihan dari kedua metode, yakni XGB dan RF, sehingga dapat meningkatkan akurasi prediksi dibandingkan jika hanya menggunakan salah satu metode saja.

3.4 Data Collection

Data Collection merupakan tahapan yang penting dalam penelitian ini, oleh karena itu dibutuhkan beberapa tahapan untuk mendapatkan kebutuhan data yang tepat diolah dimulai dari pemahaman tentang proses bisnis (*business process undesrtanding*) dari obyek yang akan diteliti, pemahaman data yang akan dikumpulkan (*data Understanding*) dan kesiapan data yang akan diolah (*data preparation*). Dengan demikian *Data Collection* dapat didapat secara tepat untuk dilaksanakan proses selanjutnya.

3.4.1 Business Process *Understanding*

Tahap ini merupakan tahap awal sebelum mempersiapkan data yang akan diolah yaitu memahami proses bisnis yang ada. Proses SEE merupakan langkah penting dalam pengembangan dan penentuan *Software Effort* yang ditawarkan kepada klien. Dalam studi kasus ini, peneliti akan fokus pada penggunaan model prediksi berdasarkan *Dataset* telah dikumpulkan menggunakan metode *Ensemble Averaging Model*.

3.4.2 Data Understanding

Setelah mengetahui proses bisnis, berikutnya adalah memahami data yang terkait dengan proses bisnisnya. Setiap parameter data berpengaruh terhadap perhitungan *Software Effort Estimation* (SEE) yang sudah ditetapkan di setiap *Dataset*.

3.4.3 Data Preparation

Penelitian ini menggunakan Data Sekunder Dimana data Promise Nasa93 dan China. *Software Effort Estimation Dataset* adalah data yang digunakan penelitian sebelumnya tentang SEE. Data Sekunder didapat dari public *Dataset* resmi pada halaman *web* resmi opensciene (https.//openscience.us/repo/effort/cocomo/nasa93.html) untuk nasa93 dan (https.//zenodo.org/records/268446) untuk China *Dataset*. Berikut parameter *Dataset* yang akan digunakan pada penelitian ini ditunjukkan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Detail Variabel *Dataset* Nasa93

No.	Feature	Deskripsi	Type Data	Selection Feature	Mean	Std Dev	Min	Max
1	*Data*	*Size of database*	Float	Data	1.004	0.073	0.94	1.16
2	*Rely*	*Requirements for software dependability*	Float	Rely	1.036	0.193	0.75	1.4
3	*Cplx*	*Complexity of the Product*	Float	Cplx	1.091	0.203	0.7	1.65
4	*Time*	*Time limit for execution*	Float	Time	1.046	0.17	0.7	1.66
5	*Stor*	*Primary storage limitation*	Float	Stor	1.008	0.121	0.87	1.34
6	*Vrtl*	*Volatility of virtual machines*	Float	Vrtl	1.045	0.158	0.71	1.72
7	*Acap*	*Capability of analysts*	Float	Acap	0.947	0.057	0.82	1.04
8	*Pcap*	*Skills of programmers*	Float	Pcap	0.919	0.069	0.79	1.04
9	*Aexp*	*Application-related experience*	Float	Aexp	0.947	0.046	0.85	1.04
10	*Vexp*	*Work with virtual machines*	Float	Vexp	1.005	0.09	0.9	1.24
11	*Lexp*	*Fluency in a programming language*	Float	Lexp	1.001	0.032	0.95	1.12
12	*Modp*	*Applying state-of-the-art programming techniques*	Float	Modp	1.004	0.059	0.91	1.12
13	*Tool*	*Time for necessary development*	Float	Tool	1.049	0.076	1.02	1.24
14	*Seed*	*The application of software*	Float	Seed	1.047	0.076	0.83	1.29
15	*Tool*	*The application of software*	Float	Tool	1.057	0.089	0.83	1.29
16	*Effort*	*Physical exertion measured in person-months*	Float	Effort	683.32	812.582.5	59	14400
17	*Loc*	*Lines of code*	Float	Loc	77.21	168.509	1.98	1150

Tabel 3.2 Detail Variabel Dataset China

No.	Feature	Deskripsi	Type Data	Mean	Std Dev	Min	Max
1	*AFP*	*Function Points(FP) adjustments*	Integer	487	1059	9	17518
2	*Input*	*input of FP*	Integer	167	486	0	9404
3	*Output*	*External output of FP*	Integer	114	221	0	2455
4	*Enquiry*	*FP of external output enquiry*	Integer	62	105	0	952
5	*File*	*FP of internal logical files*	Integer	91	210	0	2955
6	*Added*	*FP for additional functions*	Integer	260	830	0	13580
7	*Interface*	*Added FP to the external interface*	Integer	24	85	0	1572
8	*Deleted*	*FP of modified functions*	Integer	12	124	0	2657
9	*Changed*	*FP of modified functions*	Integer	85	291	0	5193
10	*PDR_UFP*	*Delivery rate of productivity (unadjusted FP)*	Double	13	14	0.4	101
11	*PDR_AFP*	*Delivery rate of productivity (adjusted FP)*	Double	12	12	0.3	83.8
12	*NPDU_UFP*	*Delivery rate of productivity (unadjusted FP)*	Double	1	1	1	4
13	*NPDR_AFP*	*Delivery rate for normalized productivity (adjusted FP)*	Double	14	15	0.4	108
14	*Dev.Type*	*Type of development*	Numerical Only {0}	0	0	0	0
15	*Resource*	*Type of team*	Discrete	12	12	0.3	83.8
16	*N_effort*	*Normalized effort*	Integer	4278	7071	31	54620
17	*Duration*	*Time spent on the project overall*	Integer	9	1	8	84
18	*Effort*	*An overview of the work report*	Integer	3921	6481	26	54260

Adapun jumlah data yang diolah sebesar 93 data untuk proyek yang telah diambil dari data Promise Nasa93 dan *Dataset* China dengan jumlah data sebanyak 498 yang telah digunakan di beberapa penelitian sebelumnya. Dengan memperhatikan Table 3.1 yang memperlihatkan 17 atribut *Dataset* nasa93 yang selanjutnya ditentukan attribut target yaitu atribut *Effort* untuk digunakan sebagai variabel dependen sedangkan Tabel 3.2 memperlihatkan 18 atribut dari Dari

Dataset China. Hasil dari evaluasi ini menjadi data sekunder yang dianalisis dan dimodelkan guna memprediksi SEE, Data ini menjadi landasan utama dalam pengembangan model prediksi biaya proyek aplikasi perangkat lunak dengan metode *Model Averaging Ensemble*.

3.5 Preprocessing

Proses Data awalnya berbentuk csv dengan value evaluasi parameter-parameter yang disebutkan di tabel 3.1 dan 3.2. Sebelum melakukan prediksi dengan algoritma *Model Averaging Ensemble*, data mentah akan diolah terlebih dahulu supaya siap untuk digunakan. Ada dua tahap dalam pengolahan data ini, yaitu.

3.5.1 Memuat *Dataset*

Data-data aplikasi dan estimasi harga yang telah terkumpul akan dimuat ke dalam bentuk *DataFrame* pandas.

3.5.2 Preprocessing Data

Proses preprocessing data akan dilakukan untuk memastikan bahwa data siap untuk digunakan dalam pembuatan model prediksi. Proses ini mencakup beberapa langkah sebagai berikut.

a) Penanganan *Missing Value*

Memeriksa *Dataset* untuk memastikan bahwa tidak ada data yang hilang (*missing value*). Jika ada baris data yang mengandung *missing value*, baris tersebut akan dihapus untuk menjaga integritas *Dataset*.

b) Normalisasi

Dalam tahapan ini, penulis melakukan Data scaling dengan tujuan membuat semua nilai data berada dalam rentang yang sama sehingga diharapkan data yang digunakan akan diproses dengan lebih baik. Teknik yang digunakan penulis adalah *scaling min-max*. Rumus dari *Min-Max Scaling* dapat dilihat di persamaan 3.1.

$$z = \frac{x - \min(x)}{[\max(x) - \min(x)]} \tag{3.1}$$

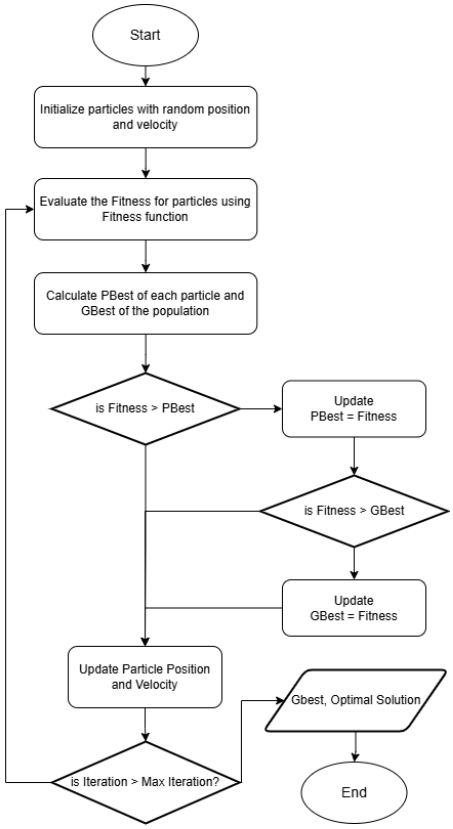
Keterangan.
z = hasil *scaling*,
x = nilai x (asli),
min(x) = nilai minimal untuk variabel x,
max(x) = nilai maksimal untuk variabel x.

c) Pembagian Data

Dataset akan dibagi berdasarkan persentase 80:20 untuk digunakan sebagai data latih (*training* data) sebesar 80% dan data uji (*testing* data) sebesar 20%.

3.6 Particle Swarm Optimization

Particle Swarm Optimization (PSO) adalah metode optimasi yang diciptakan berdasarkan perilaku sekumpulan burung. Prinsip dasarnya adalah setiap individu (partikel) dalam koloni memiliki lokalisasi spasial yang unik. Partikel-partikel ini saling berinteraksi dan berkomunikasi untuk mencari Solusi yang optimal (Gopal, Sultani, and Bansal 2020). Flowchart dari PSO dapat dilihat di gambar 3.3.



Gambar 3. 4 Flowchart PSO

Gambar 3.3 menggambarkan langkah-langkah dari algoritma Particle Swarm Optimization (PSO), yang merupakan algoritma optimasi berbasis populasi. Proses dimulai dengan inisialisasi partikel, di mana sekelompok partikel ditempatkan secara acak di ruang pencarian. Setiap partikel diinisialisasi dengan posisi dan kecepatan acak. Posisi partikel menggambarkan solusi potensial terhadap masalah yang sedang dicari solusinya, sementara kecepatan menentukan arah dan jarak pergerakan partikel dalam iterasi berikutnya. Setelah inisialisasi, setiap partikel dievaluasi menggunakan fungsi fitness yang mengukur kualitas solusi yang diwakili oleh partikel tersebut.

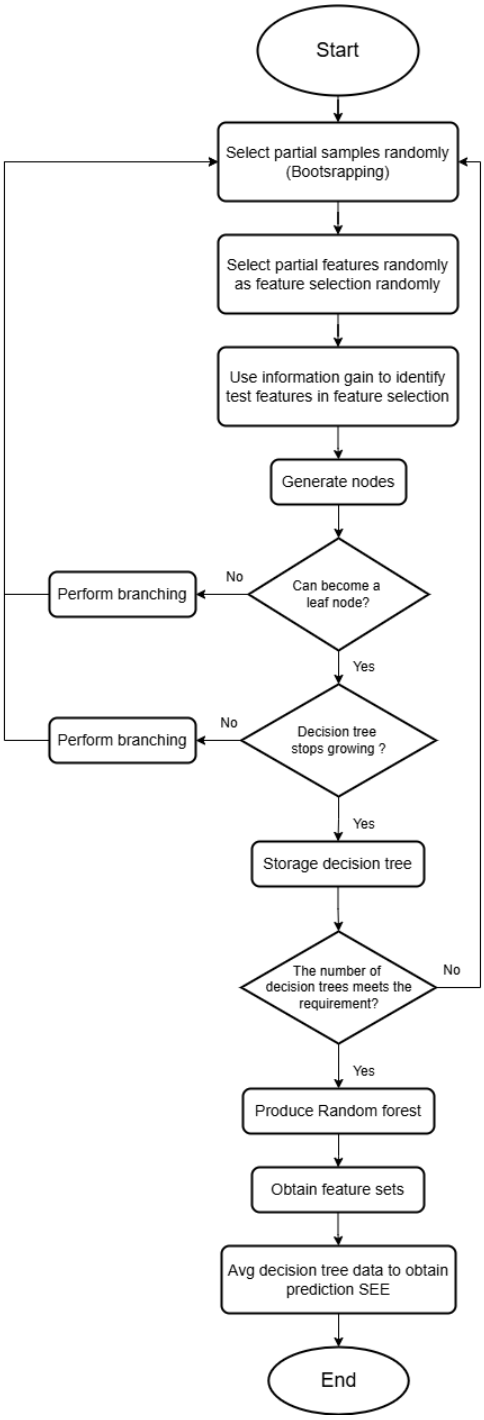
Langkah selanjutnya adalah menghitung dua parameter penting, yaitu PBest (*Personal Best*), yang menyimpan posisi terbaik yang pernah dicapai oleh setiap

partikel, dan GBest (*Global Best*), yang menyimpan posisi terbaik di antara seluruh partikel. Setelah ini, setiap partikel membandingkan nilai fitness saat ini dengan nilai PBest-nya. Jika fitness saat ini lebih baik dari PBest, maka PBest diperbarui dengan posisi terbaru. Demikian juga, nilai fitness partikel dibandingkan dengan GBest, dan jika fitness saat ini lebih baik dari GBest, maka GBest juga diperbarui.

Posisi dan kecepatan partikel kemudian diperbarui berdasarkan kombinasi dari posisi saat ini, PBest, dan GBest. Hal ini memungkinkan partikel bergerak menuju solusi optimal dengan kecepatan yang dikendalikan oleh perbedaan antara posisi saat ini dengan PBest dan GBest. Proses ini diulangi hingga kondisi penghentian tercapai, baik karena telah mencapai jumlah iterasi maksimum yang telah ditentukan, atau karena telah ditemukan solusi yang optimal atau mendekati optimal. Pada akhir proses, solusi terbaik atau optimal yang ditemukan oleh partikel akan menjadi hasil akhir dari algoritma ini (Raval and Pandya 2022).

3.7 *Random Forest*

Random Forest didasarkan pada prinsip decision tree dan dapat digunakan untuk menyelesaikan masalah klasifikasi maupun prediksi. *Random Forest* terdiri dari kumpulan beberapa decision tree, di mana masing-masing pohon menghasilkan keputusan, dan hasil akhir ditentukan berdasarkan pada majority vote (mayoritas keputusan tersebut) atau menghitung nilai rata-rata pada masalah regresi (Charoenkwan et al. 2021). Flowchart dari *Random Forest* dapat dilihat di gambar 3.4 .



Gambar 3. 5 Flowchart *Random Forest*

Gambar 3.4 menggambarkan proses pembentukan *Random Forest*, yang merupakan salah satu metode *Ensemble* learning dalam machine learning. Proses dimulai dengan memilih sebagian sampel secara acak dari *Dataset* menggunakan

teknik bootstrapping. *Bootstrapping* adalah teknik pengambilan sampel, di mana beberapa sampel dari *Dataset* dapat dipilih lebih dari satu kali, sedangkan sampel lainnya mungkin tidak dipilih sama sekali. Setelah itu, sebagian fitur juga dipilih secara acak untuk setiap pohon keputusan, dan proses ini disebut feature selection.

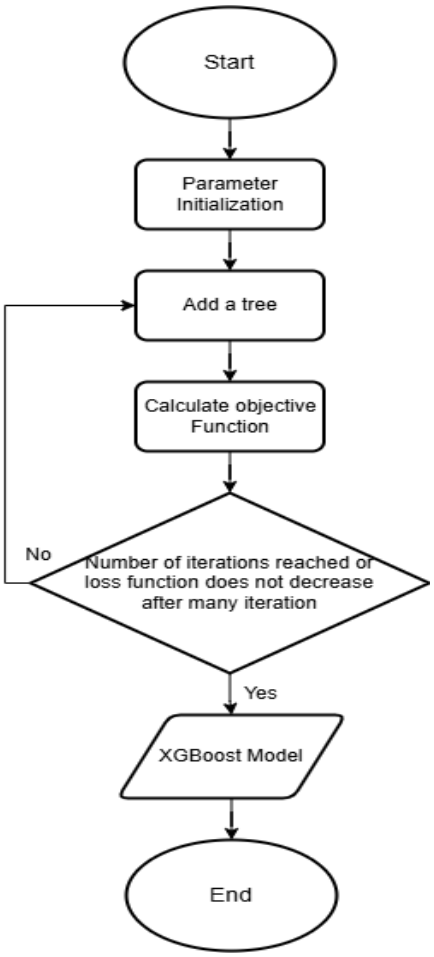
Selanjutnya, information gain digunakan untuk mengidentifikasi fitur mana yang akan digunakan sebagai kriteria pengujian dalam pembuatan node. Node kemudian dihasilkan, dan proses branching dimulai. Pada setiap node, akan ditentukan apakah node tersebut dapat menjadi leaf node atau tidak. Jika node tersebut tidak dapat menjadi *leaf node*, maka proses branching dilanjutkan. Jika pohon keputusan telah berhenti tumbuh, pohon tersebut akan disimpan sebagai bagian dari *Random Forest*.

Setelah pohon keputusan tersimpan, sistem akan memeriksa apakah jumlah pohon yang diperlukan untuk membentuk *Random Forest* telah terpenuhi. Jika jumlah pohon telah mencapai batas yang ditentukan, *Random Forest* akan diproduksi. Setelah *Random Forest* terbentuk, sistem akan menghasilkan himpunan fitur dari proses tersebut. Terakhir yaitu *Averaging* hasil semua pohon Keputusan untuk mendapatkan hasil prediksi.

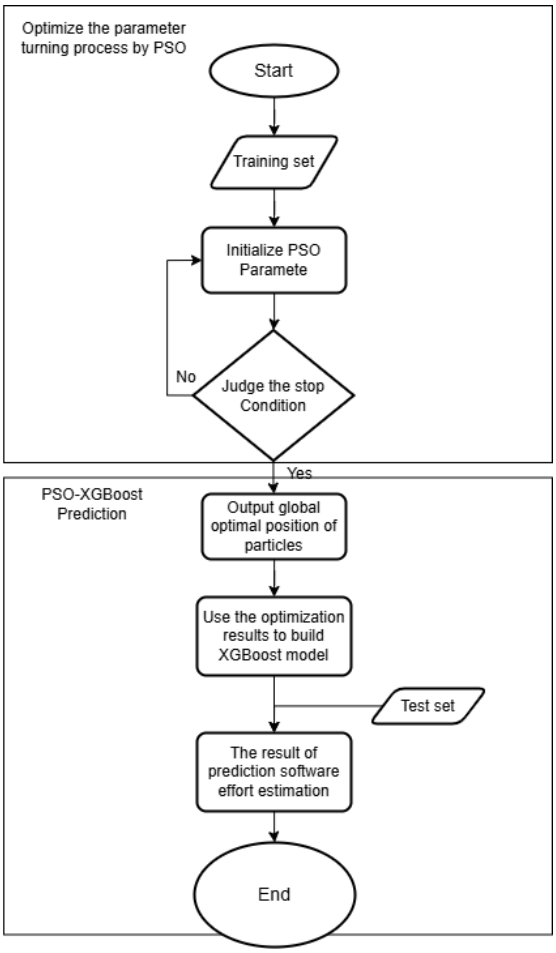
3.8 *Extreme Gradient Boosting*

XGBoost merupakan pengembangan dari algoritma *Gradient Boosting Decision Trees* (GBDT), algoritma ini dirancang untuk meningkatkan kecepatan dan kinerja model dengan mengoptimalkan proses pelatihan pohon keputusan secara bertahap (Montero-Manso et al. 2020). XGBoost menggunakan pendekatan boosting, di mana model dibangun secara berurutan, dan setiap model baru

berusaha untuk mengoreksi kesalahan dari model sebelumnya (Herni Yulianti, Oni Soesanto, and Yuana Sukmawaty 2022). Berikut adalah flowchart *Extreme* Gradient Boostring (XGB) .



*Gambar 3. 6 Fl*owchart General XGB



Gambar 3. 7 Flowchart Proses PSO XGB

Gambar 3.5 Menggambarkan alur kerja XGBoost dalam membangun model prediksi. Proses dimulai dengan inisialisasi parameter, di mana parameter penting seperti learning rate, jumlah pohon, dan regulasi disetel. Setelah itu, model akan menambahkan satu pohon keputusan baru ke dalam model secara iteratif.

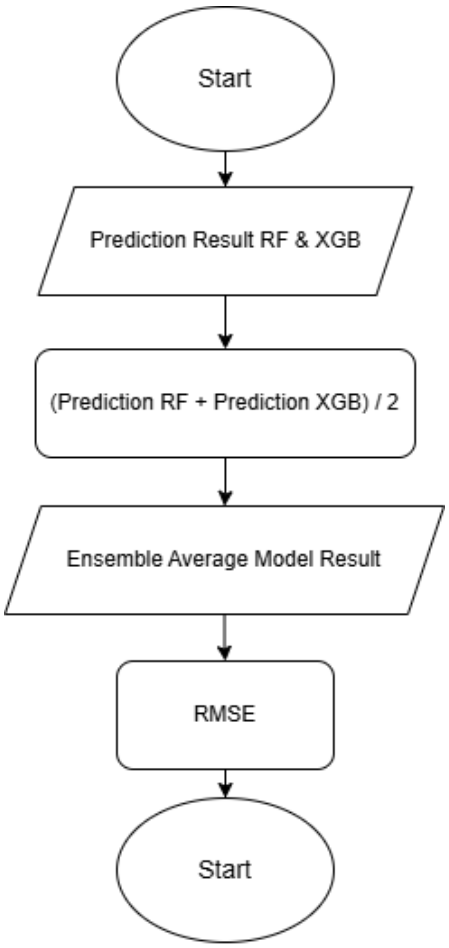
Setelah penambahan pohon, dilakukan perhitungan fungsi objektif, yang bertujuan untuk mengukur seberapa baik model saat ini bekerja dalam hal meminimalkan *Error* atau loss. Jika model belum mencapai jumlah iterasi yang ditentukan atau jika fungsi loss masih bisa dikurangi, maka proses akan mengulangi penambahan pohon dan perhitungan fungsi objektif (Fang et al. 2022).

Proses ini akan terus berlanjut sampai jumlah iterasi maksimal tercapai atau fungsi loss berhenti menurun secara signifikan. Setelah kriteria ini terpenuhi, alur berakhir dan model XGBoost final yang telah dilatih siap untuk digunakan dalam membuat prediksi pada data SEE.

Kemudian Gambar 3.6 adalah Gambaran jika penggunaan Particle Swarm Optimization diterapkan di model XGBoost Dimana sebelum pemilihan parameter dilakukan proses optimasi dengan PSO menggunakan data Promise Nasa93 yang digunakan dalam penelitian ini. Dan parameter paling optimal akan digunakan untuk membangan model XGBoost untuk memprediksi SEE.

3.9 Ensemble Averaging Model

Setelah melakukan training 2 model sebelumnya yaitu *Random Forest* (RF) dan *Extreme Gradient Boosting* (XGB), di Langkah ini peneliti melakukan metode *Ensemble Averaging* Model dengan mengambil rata rata prediksi dari kedua model RF dan XGB. Untuk flowchart dari *Ensemble Averaging* Model dapat dilihat di Gambar 3.7



Gambar 3. 8 *Ensemble Averaging* model flowchart

Gambar 3.7 adalah *Flowchart* dari *Ensemble Averaging* model Dimana data Prediksi dari RF dan XGB akan digunakan sebagai input, lalu melalui tahap proses dengan menggunakan rumus .

$$Ensemble\ Avg\ Model\ Prediction = \frac{(RF_{prediction} + XGB_{predction})}{2}$$

Setelah itu output prediksi *Ensemble Averaging* Model akan dievaluasi dengan menggunakan

3.10 Evaluasi Performa Model Prediksi

Dalam penelitian ini untuk mengukur performa prediksi menggunakan *Root Mean Absolute Error* (RMSE) sebagai hasil akhir dari pengukuran metode prediksi.

3.10.1 RMSE (Root *Mean Absolute Error*)

Untuk mengukur nilai keakuratan metode yang digunakan dalam penelitian ini menggunakan RMSE agar diketahui rata-rata selisih mutlak nilai sebenarnya (aktual) dengan nilai prediksi kemudian di gunakan fungsi pengakaran (Yulianto, Mahmudy, and Soebroto 2020). Semakin kecil nilai RMSE, Semakin baik model tersebut dalam melakukan prediksi (Chai, Draxler, and Prediction 2014). dituangkan dalam rumus RMSE sebagai berikut .

$$RMSE = \left(\frac{\sum(y_i - \hat{y}_i)}{n}\right)^{1/2} \quad (3.5)$$

Keterangan .
n = ukuran sampel
y*i* = nilai data aktual ke-*i*
y ̂_i = nilai data prediksi ke-*i*

3.11 Eksperimen

Dalam penelitian ini untuk eksperimen yang telah dilakukan penulis yaitu menggunakan *Dataset* nasa93 dengan jumlah *Dataset* sejumlah 50 data yang digunakan untuk eksperimen, sesuai dengan desain sistem di gambar 3.2 maka data awal yang akan dipakai ditampilkan di tabel 3.3.

Tabel 3. 3 Dataset eksperimen

49	48	47	46	.	4	3	2	1	0	**Index**
1.00	1.40	1.00	1.00	·	1.15	1.15	1.15	1.15	1.15	**rely**
1.08	1.00	1.00	1.00	·	0.94	0.94	0.94	0.94	0.94	**data**
1.15	1.65	1.15	1.00	·	1.15	1.15	1.15	1.15	1.15	**cplx**
1.30	1.11	1.00	1.00	·	1.00	1.00	1.00	1.00	1.00	**time**
1.00	1.06	1.06	1.00	·	1.00	1.00	1.00	1.00	1.00	**stor**
1.00	0.87	1.00	0.87	·	0.87	0.87	0.87	0.87	0.87	**virt**
1.07	0.87	1.00	1.00	·	0.87	0.87	0.87	0.87	0.87	**turn**
0.86	1.00	0.86	0.86	·	1.00	1.00	1.00	1.00	1.00	**acap**
0.91	0.91	0.91	0.82	·	1.00	1.00	1.00	1.00	1.00	**aexp**
0.86	1.00	1.00	0.70	·	1.00	1.00	1.00	1.00	1.00	**pcap**
1.0	1.0	1.0	1.1	·	1.0	1.0	1.0	1.0	1.0	**vexp**
0.95	1.00	0.95	0.95	·	0.95	0.95	0.95	0.95	0.95	**lexp**
1.10	1.10	0.91	0.91	·	0.91	0.91	0.91	0.91	0.91	**modp**
1.10	0.91	1.00	1.00	·	1.00	1.00	1.00	1.00	1.00	**tool**
1.04	1.00	1.04	1.00	·	1.08	1.08	1.08	1.08	1.08	**sced**
78.0	21.0	47.5	190.0	·	9.7	8.2	7.7	24.6	25.9	**equivphys**
571.4	107.0	252.0	420.0	·	25.2	36.0	31.2	117.6	117.6	**act_effort**

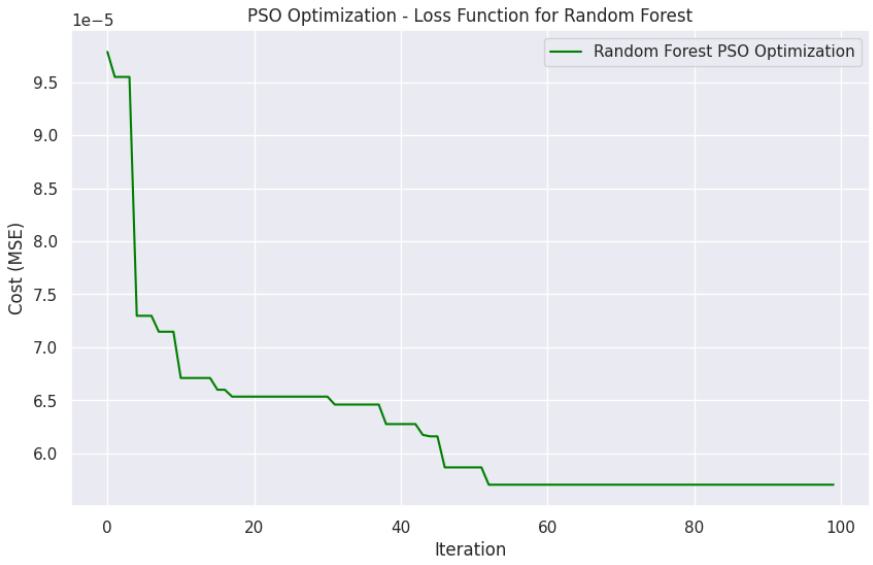
Langkah selanjutnya yaitu melakukan Preprocessing dengan menggunakan Min-Max Scaler, untuk hasil *Dataset*nya dapat dilihat di tabel 3.4 .

Tabel 3. 4 Dataset Experiment setelah Min-Max Scaling

Index	0	1	2	3	4	.	46	47	48
rely	0.519231	0.519231	0.519231	0.519231	0.519231	.	0.230769	0.230769	1.000000
data	0.000000	0.000000	0.000000	0.000000	0.000000	.	0.272727	0.272727	0.272727
cplx	0.3750	0.3750	0.3750	0.3750	0.3750	.	0.1875	0.3750	1.0000
time	0.000000	0.000000	0.000000	0.000000	0.000000	.	0.000000	0.000000	0.166667
stor	0.000000	0.000000	0.000000	0.000000	0.000000	.	0.000000	0.107143	0.107143
virt	0.000000	0.000000	0.000000	0.000000	0.000000	.	0.000000	0.464286	0.000000
turn	0.00	0.00	0.00	0.00	0.00	.	0.65	0.65	0.00
acap	1.000000	1.000000	1.000000	1.000000	1.000000	.	0.517241	0.517241	1.000000
aexp	1.0	1.0	1.0	1.0	1.0	.	0.0	0.5	0.5
pcap	1.000000	1.000000	1.000000	1.000000	1.000000	.	0.000000	1.000000	1.000000
vexp	0.0	0.0	0.0	0.0	0.0	.	1.0	0.0	0.0
lexp	0.000000	0.000000	0.000000	0.000000	0.000000	.	0.000000	0.000000	0.263158
modp	0.321429	0.321429	0.321429	0.321429	0.321429	.	0.321429	0.321429	1.000000
tool	0.414634	0.414634	0.414634	0.414634	0.414634	.	0.414634	0.414634	0.195122
sced	1.0	1.0	1.0	1.0	1.0	.	0.0	0.5	0.0
equivphys	0.056321	0.053232	0.013070	0.014259	0.017823	.	0.446293	0.107652	0.044677
act_effort	0.045660	0.045660	0.009533	0.011540	0.007025	.	0.172102	0.101856	0.041228

49	0.230769	0.636364	0.3750	0.454545	0.000000	0.464286	1.00	0.517241	0.5	0.533333	0.0	0.000000	1.000000	0.658537	0.5	0.180133	0.235407

Setelah itu, *Dataset* akan di *split,* menjadi dua yakni data training dan data testing, dengan perbandingan rasio sebesar 80% untuk data training dan 20% untuk data testing. Lalu Langkah selanjutnya adalah menggunakan optimasi Particle Swarm Optimization (PSO) sebagaimana yang telah digambarkan di Gambar 3.3 dengan menggunakan dua metode yaitu *Random Forest* (RF) sebagaimana yang telah digambarkan di Gambar 3.4 dan metode *Extreme Gradient Boosting* (XGB) sebagaimana digambarkan di gambar 3.5 dan 3.6. untuk Grafik *loss function* training dari Metode RF dan XGB dengan Optimasi PSO dapat dilihat di gambar 3.8 dan 3.9 .

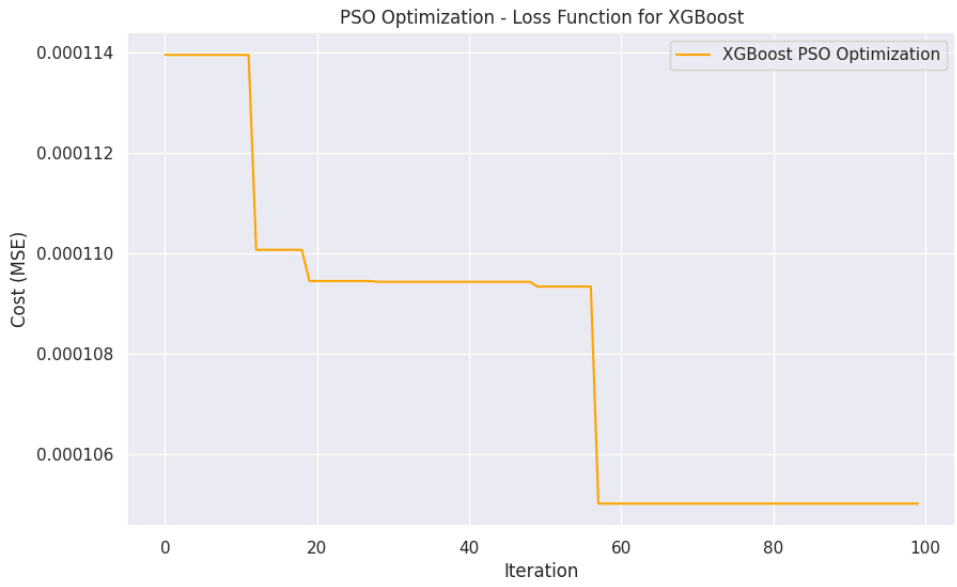


Gambar 3. 9 Loss function PSO-RF untuk Experiment

Gambar 3.8 memperlihatkan proses optimisasi fungsi loss dari model *Random Forest* menggunakan algoritma Particle Swarm Optimization (PSO) selama 100 iterasi. Grafik menunjukkan bagaimana nilai *Mean Squared Error*

(MSE), yang mewakili kesalahan prediksi model, secara progresif menurun. Pada tahap awal (sekitar 20 iterasi pertama), terlihat penurunan yang signifikan pada MSE, dari sekitar 9.5 hingga 7.0, menandakan bahwa PSO berhasil dengan cepat menemukan parameter yang lebih baik untuk meningkatkan performa model. Setelah itu, meskipun masih ada penurunan, kecepatannya melambat, dan grafik menunjukkan tren stabil dari iterasi ke-50 hingga ke-100, di mana MSE bertahan di bawah 6.0.

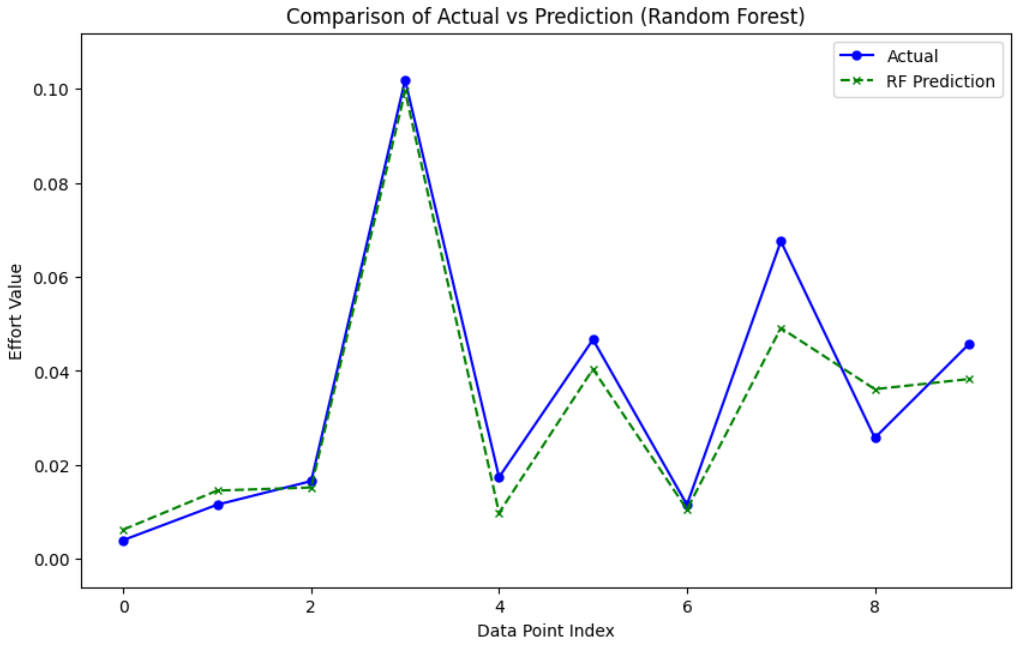
Stabilitas pada bagian akhir grafik menunjukkan bahwa proses optimisasi telah mendekati titik konvergensi, di mana penyesuaian parameter lebih lanjut tidak lagi memberikan peningkatan signifikan pada performa model. Dengan demikian, grafik ini menunjukkan efektivitas PSO dalam menemukan parameter optimal untuk model *Random Forest*, dengan peningkatan performa yang drastis di awal, dan stabil di akhir iterasi.



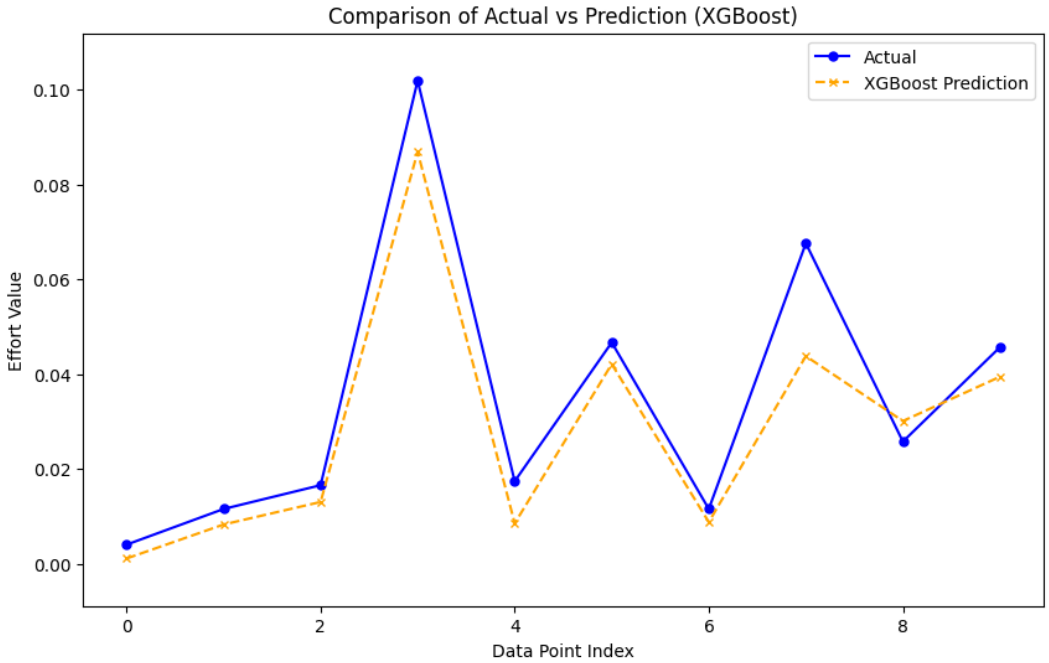
Gambar 3. 10 Loss function PSO-XGB untuk Experiment

Gambar 3.9 menunjukkan proses optimisasi fungsi loss dari model XGBoost menggunakan algoritma Particle Swarm Optimization (PSO) selama 100 iterasi. Pada awal grafik, terlihat bahwa nilai *Mean Squared Error* (MSE) dimulai dari sekitar 0.000114. Pada sekitar iterasi ke-10, terjadi penurunan tajam hingga MSE mencapai 0.000110, yang menunjukkan adanya perbaikan performa model dalam waktu singkat.

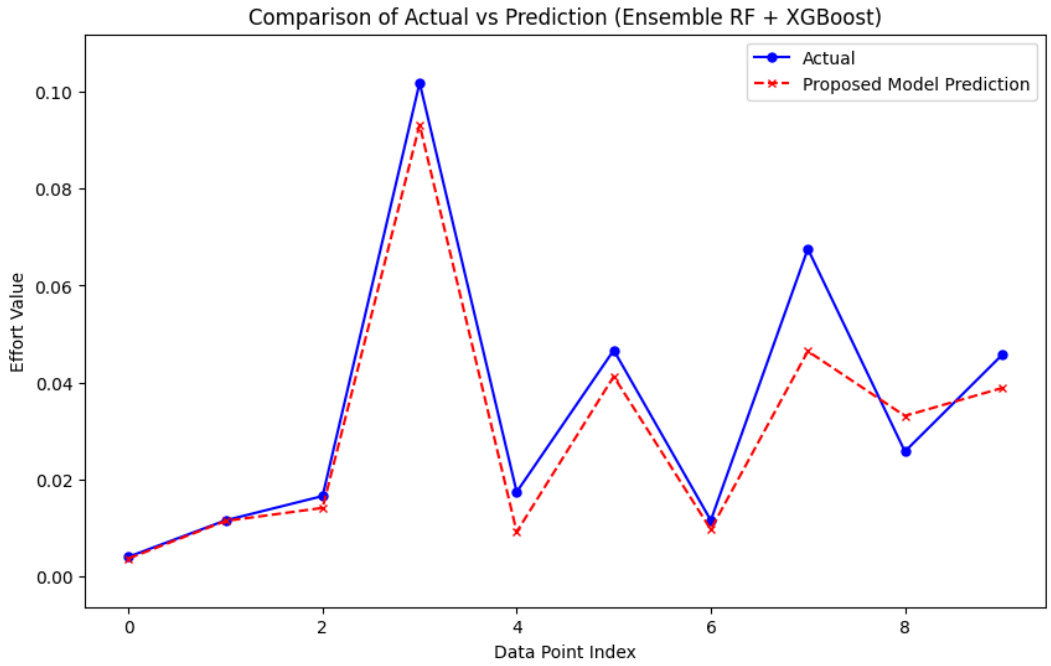
Setelah iterasi tersebut, grafik mengalami fase yang lebih stabil, dengan sedikit fluktuasi hingga iterasi ke-50. Pada titik ini, PSO berhasil menurunkan nilai MSE lebih lanjut ke angka 0.000106 di sekitar iterasi ke-60. Setelah iterasi ini, grafik tetap stabil hingga iterasi ke-100, menandakan bahwa proses optimisasi telah mencapai titik konvergensi. Keseluruhan grafik menunjukkan bahwa optimisasi PSO pada model XGBoost lebih lambat pada awalnya dibandingkan dengan *Random Forest*, tetapi akhirnya berhasil menemukan parameter yang optimal dengan penurunan yang signifikan pada MSE. Proses ini menggambarkan bagaimana PSO dapat membantu model XGBoost mencapai performa yang lebih baik dalam memprediksi data.untuk hasil prediksi model dengan parameter terbaik optimasi PSO dapat dilihat di gambar 3.10 untuk *Random Forest*, 3.11 untuk *Extreme Gradient Boosting*, dan 3.12 untuk *Ensemble* RF+XGB .



Gambar 3. 11 Perbandingan prediksi RF dan data Aktual unutk Eksperimen



Gambar 3. 12 Perbandingan prediksi XGB dan data Aktual unutk Eksperimen



Gambar 3. 13 Perbandingan prediksi Ensemble RF+XGB dan data Aktual unutk Eksperimen

Dari ketiga grafik yang ditampilkan di gambar 3.10, 3.11 dan 3.12, penulis dapat menganalisis performa prediksi dari tiga model berbeda. *Random Forest*, XGBoost, dan model *Ensemble* yang menggabungkan keduanya. Pada grafik pertama, prediksi *Random Forest* (garis hijau putus-putus) menunjukkan kecenderungan mengikuti pola data aktual (garis biru), namun terdapat beberapa deviasi terutama pada puncak data. Grafik kedua menunjukkan prediksi model XGBoost (garis oranye putus-putus) yang juga mengikuti tren data aktual dengan sedikit perbedaan pada beberapa titik data. Grafik ketiga menampilkan prediksi dari model *Ensemble* (garis merah putus-putus), yang tampaknya memberikan hasil yang lebih sesuai dengan data aktual dibandingkan dua model sebelumnya. Model *Ensemble* ini tampak lebih stabil dan akurat dalam mengikuti fluktuasi nilai aktual, menunjukkan bahwa penggabungan dua model ini dapat meningkatkan akurasi

prediksi secara keseluruhan. Ini menunjukkan bahwa penggunaan teknik *Ensemble* dapat menjadi pendekatan yang efektif untuk meningkatkan kualitas prediksi dalam konteks *Software Effort Estimation* (SEE). Untuk detail hasil prediksi dapat dilihat di tabel 3.5 .

Tabel 3. 5 Hasil Prediksi dan Deviasi tiap Model Eksperimen

Index Data	**Actual Effort**	***Ensemble* RF+XGB Prediction**	**RF Prediction**	**XGB Prediction**	**Deviation (RF+XGB)**	**Deviation (RF)**	**Deviation (XGB)**
32	0.004014	0.003664	0.006235	0.001093	**0.000350**	-0.002221	0.002921
12	0.011540	0.011409	0.014550	0.008269	**0.000131**	-0.003010	0.003272
25	0.016558	0.014110	0.015199	0.013020	0.002448	**0.001359**	0.003538
47	0.101856	0.093186	0.099496	0.086875	**0.008671**	0.002360	0.014982
33	0.017394	0.009159	0.009804	0.008514	**0.008235**	0.007590	0.008880
22	0.046663	0.041241	0.040377	0.042106	0.005422	0.006286	**0.004558**
3	0.011540	0.009654	0.010526	0.008783	**0.001886**	0.001015	0.002757
30	0.067570	0.046449	0.049123	0.043774	0.021121	**0.018447**	0.023795
26	0.025757	0.033103	0.036111	0.030094	**-0.007346**	-0.010355	-0.004337
1	0.045660	0.038847	0.038263	0.039431	**0.006813**	0.007397	0.006229

Dari tabel 3.5 diatas perbandingan antara nilai aktual, prediksi model *Ensemble* (*Ensemble Random Forest* dan XGBoost), prediksi *Random Forest*, serta prediksi XGBoost, dapat dilihat deviasi (selisih) antara nilai aktual dan prediksi dari setiap model. Analisis deviasi ini memberikan wawasan mengenai akurasi masing-masing model.

Secara umum, model *Ensemble* cenderung memberikan prediksi yang lebih mendekati nilai aktual dibandingkan dengan masing-masing model individual (*Random Forest* dan XGBoost), terlihat dari banyaknya nilai deviasi terkecil yang dihasilkan oleh model *Ensemble*. Misalnya, pada data pertama dengan nilai aktual 0.004014, model *Ensemble* memiliki deviasi terkecil sebesar 0.000350 dibandingkan dengan model *Random Forest* (-0.002221) dan XGBoost (0.002921). Begitu juga pada data kedua, deviasi model *Ensemble* hanya 0.000131, yang lebih kecil dibandingkan dengan *Random Forest* (-0.003010) dan XGBoost (0.003272).

Meskipun demikian, ada beberapa kasus di mana salah satu model individual (*Random Forest* atau XGBoost) memiliki deviasi yang lebih baik dibandingkan model *Ensemble*. Misalnya, pada data ketiga, prediksi *Random Forest* memberikan deviasi terbaik sebesar 0.001359 dibandingkan dengan model *Ensemble* (0.002448) dan XGBoost (0.003538). Dari analisis ini secara keseluruhan, penggunaan pendekatan *Ensemble* memberikan prediksi yang lebih akurat dalam berbagai skenario. Hal ini juga dibuktikan dengan evaluasi menggunakan *Root Mean Squared Error (*RMSE) di tabel 3.6 .

Tabel 3. 6 Evaluasi RMSE untuk Eksperimen

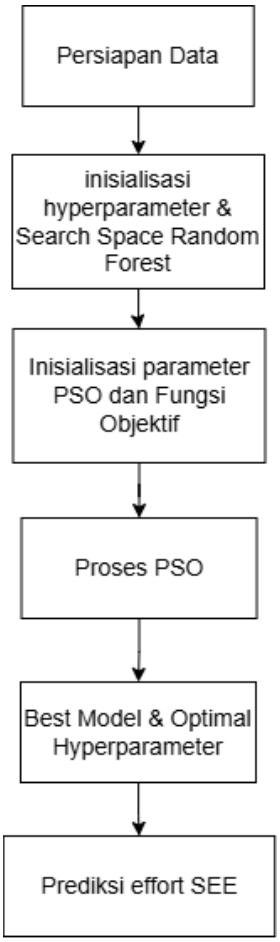
Model	**RMSE**
Random Forest (PSO optimized)	0.00853
XGBoost (PSO optimized)	0.00993
***Ensemble* (RF + XGBoost) PSO Optimized**	**0.00788**

Dari hasil analisis model tabel 3.6, dapat dilihat bahwa model *Ensemble* yang menggabungkan *Random Forest* dan XGBoost yang dioptimalkan menggunakan PSO menghasilkan nilai Root Mean Square *Error* (RMSE) terendah, yaitu 0.00788. Hal ini menunjukkan bahwa model *Ensemble* memiliki performa yang lebih baik dibandingkan dengan model individual lainnya. Model *Random Forest* yang juga dioptimalkan dengan PSO memiliki RMSE sebesar 0.00853, sedangkan model XGBoost dengan optimasi PSO menghasilkan RMSE 0.00993. Perbedaan nilai RMSE ini menunjukkan bahwa meskipun *Random Forest* memiliki kinerja yang baik, kombinasi kedua model dalam bentuk *Ensemble* memberikan peningkatan akurasi yang signifikan dalam memprediksi estimasi usaha perangkat lunak. Keberhasilan pendekatan *Ensemble* ini mengindikasikan bahwa pemanfaatan informasi dari kedua model dapat memberikan hasil yang lebih akurat dibandingkan dengan penggunaan salah satu model secara terpisah.

BAB IV
PSO-*RANDOM FOREST*

4.1. Desain Metode PSO - *Random Forest*

Pada bab ini akan membahas secara rinci tentang metode *Random Forest* dengan Optimasi menggunakan PSO yang digunakan untuk prediksi SEE dengan dataset Nasa93 dan China Dataset. Strategi yang digunakan digambarkan di gambar 4.1.



Gambar 4. 1 Desain metode PSO-RF

Berdasarkan gambar 4.1 dimulai dengan mendefinisikan masalah estimasi software effort dan menentukan metode optimasi yang digunakan, yaitu *Random Forest* dengan optimasi PSO.

4.1.1. Persiapan Data

Pada tahap ini akan melakukan *feature extraction* dataset untuk *Software Effort Estimation* (SEE). Kemudian membagi dataset menjadi data latih (*training set*) dan data uji (*test set*). Langkah selanjutnya yaitu *Data preprocessing* menggunakan *Min-Max Scaling*.

4.1.2. Inisialisasi *Hyperparameter* dan Search Space

hyperparameter yang dioptimalkan yaitu n_estimators, max_depth, min_samples_split, dan random_state. lower bounds dan upper bounds untuk masing-masing *hyperparameter* ditetapkan di tabel 4.1.

Tabel 4. 1, *Hyperparameter Random Forest*

Hyperparameter	***Lower Bound***	***Upper Bound***	**Deskripsi**
n_estimators	50	300	Jumlah pohon dalam *Random Forest*. Semakin banyak pohon, semakin baik akurasinya, tetapi dengan biaya waktu yang lebih tinggi.
max_depth	2	30	Kedalaman maksimum setiap pohon. Semakin besar nilai ini, semakin kompleks pohonnya, tetapi bisa menyebabkan overfitting.
Min_samples_split	2	10	Jumlah sampel minimum untuk memecah node. Nilai lebih tinggi mencegah pohon menjadi terlalu kompleks.

4.1.3. Inisialisasi Fungsi Objektif

Fungsi objektif yang digunakan untuk menghitung *loss function* dari model adalah nilai *Mean Squared Error* (MSE).

$$\text{MSE} \quad = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{4.1}$$

Keterangan :

MSE = *Mean Squared Error*

n = jumlah data

Yi = nilai aktual

Y^I = nilai prediksi

Model *Random Forest* Regressor dilatih menggunakan data latih, dan nilai MSE dihitung berdasarkan prediksi data uji.

4.1.4. Inisialisasi PSO

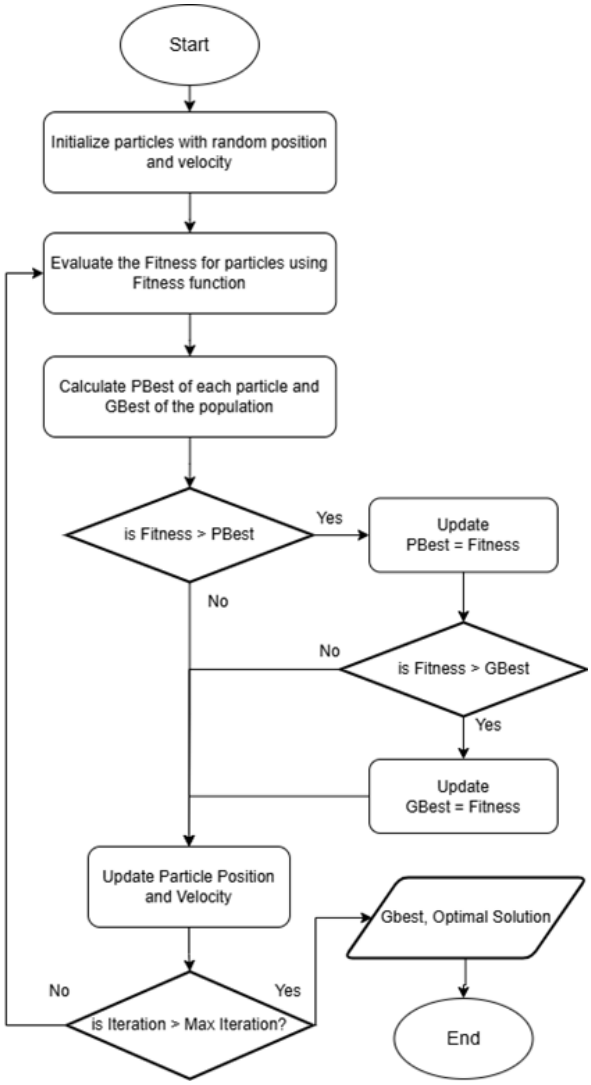
Menentukan parameter PSO yaitu jumlah partikel (n_particles), dimensi (dimensions), dan parameter eksplorasi seperti w, c1, dan c2. Untuk parameter PSO dapat dilihat di tabel 4.2.

Tabel 4. 2. Parameter PSO

Parameter	**Nilai**
Jumlah partikel (n_particles)	30
Dimensi (dimensions)	4
Inersia (w)	0.9
Koefisien Kognitif (c1)	0.5
Koefisien Sosial (c2)	0.3
Jumlah iterasi	100

4.1.5. Proses Optimasi PSO

Pada tiap iterasinya setiap partikel menghasilkan kombinasi *hyperparameter* berdasarkan ruang pencarian. Nilai MSE dihitung menggunakan fungsi objektif. Posisi terbaik partikel dan populasi diperbarui. Ulangi proses hingga iterasi mencapai batas maksimum atau konvergensi tercapai. Untuk alur PSO dapat dilihat Digambar 4.2.



Gambar 4. 2. Alur PSO

4.1.6. Model Optimal

Pada tahap ini *hyperparameter* terbaik (rf_best_pos) digunakan untuk membangun model *Random Forest Regressor*. Model dilatih (*training*) kembali dengan data latih.

4.1.7. Evaluasi Model

Menggunakan model dengan *hyperparameter* optimal untuk memprediksi data uji. Hitung nilai evaluasi, menggunakan *Root Mean Squared Error* (RMSE), untuk menilai kinerja model.

4.2. Implementasi

Training model dilakukan sesuai Desain metode PSO-RF di gambar 4.1. Model PSO-*Random Forest* dioptimasi menggunakan parameter yang dapat dilihat di tabel 4.3.

Tabel 4. 3. Rekapitulasi pengaturan *hyperparameter* RF dan parameter PSO

Hyperparameter	**Nilai**
Hyperparameter Random Forest	*n_estimators,max_depth, min_samples_split*
Lower Bounds	50, 2, 2
Upper Bounds	300, 30, 10
Fungsi Objektif	*Mean Squared Error (MSE)*
Jumlah Partikel (*n_particles*)	30
Dimensi (*dimensions*)	4
Inersia (w)	0.9
Koefisien Kognitif (c1)	0.5
Koefisien Sosial (c2)	0.3
Jumlah iterasi (iters)	100

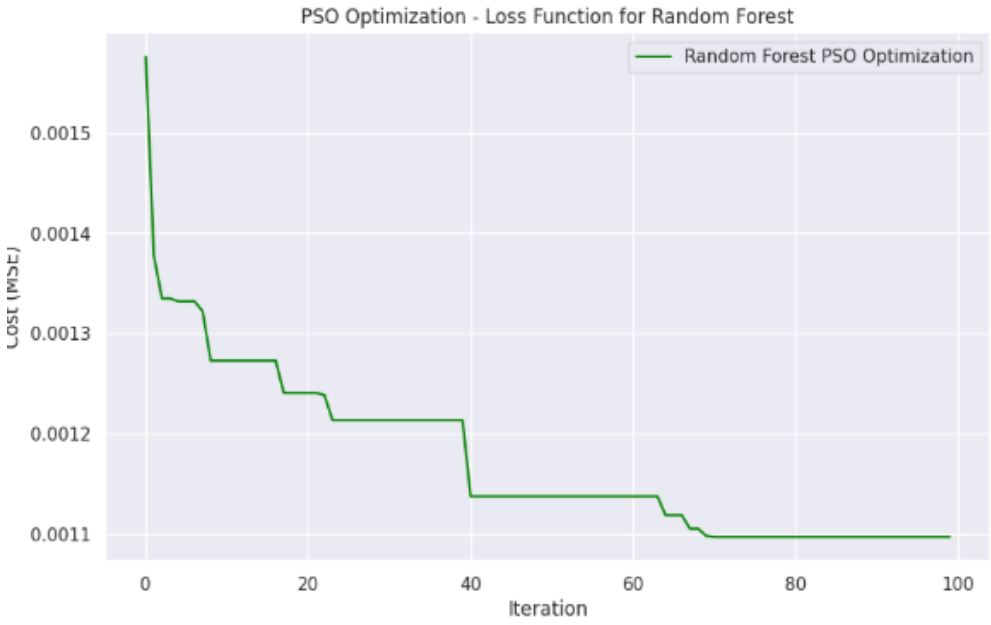
4.3. Uji Coba

Training Model dilakukan menggunakan rasio data pelatihan dan pengujian sebesar 80:20, yaksi 80% untuk data pelatihan dan 20% untuk data uji dari hasil

preprocessing yang selanjutnya digunakan untuk proses pengujian model. Selain itu, proses *training* dilakukan untuk setiap iterasi PSO yang akan memvarisasikan *hyperparameter* dari model *Random Forest.*

4.3.1. Hasil Optimasi PSO-RF nasa93

Hasil dari optimasi *model* PSO-RF dengan dataset nasa93 dapat dilihat di gambar 4.4.



Gambar 4. 3. Visualisasi proses optimasi PSO untuk model PSO-RF

Gambar 4.3. menunjukkan proses optimasi model *Random Forest* (RF) menggunakan Particle Swarm Optimization (PSO) pada data NASA93 Dataset untuk estimasi Software Effort. Sumbu horizontal merepresentasikan jumlah iterasi (hingga 100 iterasi), sedangkan sumbu vertikal menunjukkan nilai cost yang digunakan, yaitu *Mean Squared Error* (MSE).

Dari grafik ini, terlihat bahwa pada awal iterasi (0–10 iterasi), nilai MSE mengalami penurunan yang signifikan. Hal ini menunjukkan bahwa PSO berhasil menemukan kombinasi *hyperparameter* awal yang lebih optimal dibandingkan inisialisasi awal. Selanjutnya, penurunan MSE mulai melambat pada iterasi 20–50, menandakan bahwa algoritma PSO mulai mendekati nilai konvergensi. Setelah iterasi ke-60, nilai MSE menjadi stabil di sekitar 0.0011, yang mengindikasikan bahwa kombinasi *hyperparameter* terbaik telah ditemukan oleh PSO untuk model *Random Forest*. Hasil dari training model PSO-RF nasa93 dapat dilihat di tabel 4.4.

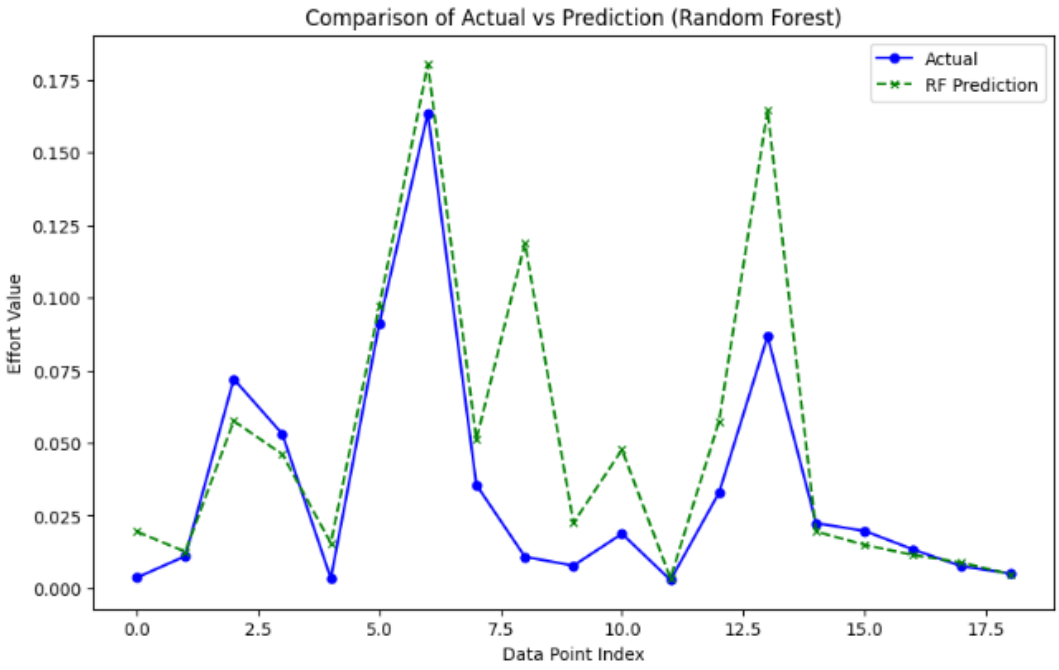
Tabel 4. 4. Hasil prediksi model PSO-RF vs Aktual untuk dataset nasa93

Random Forest Prediction **(act_effort)**	***Actual Data*** **(act_effort)**
0.01948356493	0.003608611904
0.01253558558	0.01102089581
0.05755657155	0.07200156048
0.04608866504	0.05310511301
0.01551639361	0.003364786775
…	…
0.01947988617	0.02238314681
0.01490527251	0.01970107039
0.01140270111	0.01331285202
0.008796027162	0.007509813961
0.004786651892	0.005071562675

Tabel 4.4 menunjukkan perbandingan hasil prediksi dari model PSO-RF jika dibandingkan dengan data Aktual untuk dataset nasa93. Hasil menunjukkan model menghasilkan prediksi yang cukup mendekati nilai aktual untuk beberapa data, seperti pada kasus dengan nilai aktual 0.0110 yang diprediksi sebesar 0.0125, serta nilai aktual 0.0531 yang diprediksi sebesar 0.0460. Namun, terdapat beberapa

kasus dengan error yang lebih besar, misalnya nilai aktual 0.0036 diprediksi menjadi 0.0194, menunjukkan bahwa model terkadang mengalami kesulitan menangkap pola data tertentu.

Selanjutnya, data di bagian akhir menunjukkan bahwa prediksi menjadi semakin dekat dengan nilai aktual. Misalnya, pada nilai aktual 0.0197, model memprediksi sebesar 0.0149, atau nilai aktual 0.0075 diprediksi sebesar 0.0087, yang mengindikasikan bahwa model mampu menangkap pola untuk sebagian besar data uji. Untuk visualisasi hasil prediksi vs aktual dari model PSO-RF ini dapat dilihat di gambar 4.4.



Gambar 4. 4. Visualisasi komparasi data prediksi PSO-RF vs aktual untuk nasa93

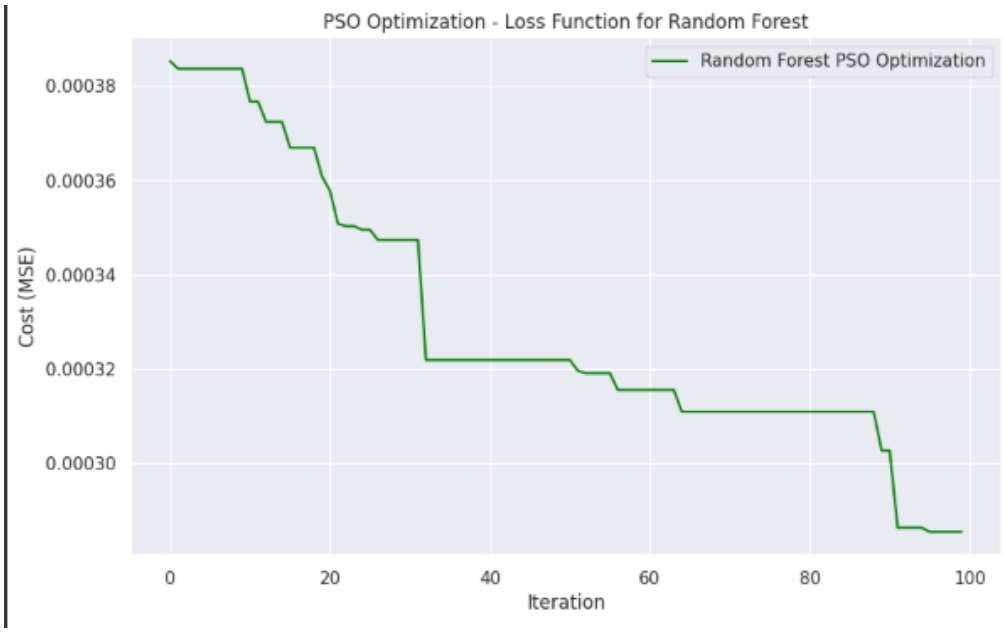
Lankah terakhir untuk melakukan evaluasi model yaitu dengan menghitung *Root Mean Squared Error* (RMSE) menggunakan data prediksi PSO-RF dan data aktual untuk nasa93 dataset. Hasil RMSE dapat dilihat di tabel 4.5.

Tabel 4. 5 Hasil RMSE model PSO-RF untuk nasa93

Model	Dataset	RMSE
PSO-*Random Forest*	Nasa93	0.0329

4.3.2. Hasil Optimasi PSO-RF China

Hasil dari optimasi *model* PSO-RF dengan dataset nasa93 dapat dilihat di gambar 4.5.



Gambar 4. 5. Visualisasi proses optimasi PSO untuk model PSO-RF

Gambar 4.4 menunjukkan proses optimasi model *Random Forest* (RF) menggunakan Particle Swarm Optimization (PSO) untuk pelatihan pada dataset China Dataset dalam tugas prediksi Software Effort Estimation. Sumbu horizontal

menggambarkan iterasi optimasi (dari 0 hingga 100 iterasi), sementara sumbu vertikal menunjukkan nilai cost, yang diukur dalam bentuk *Mean Squared Error* (MSE).

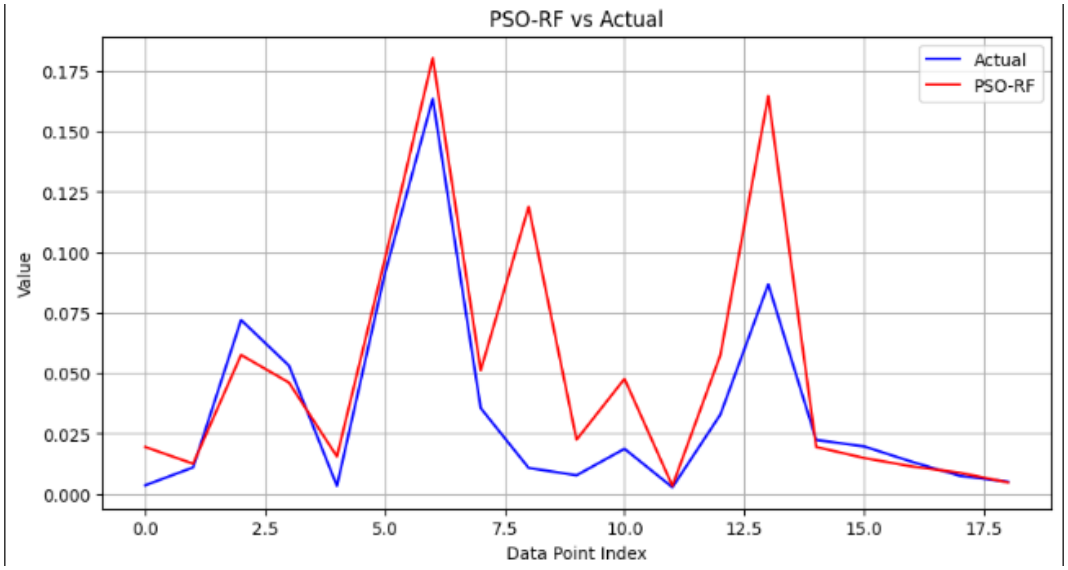
Pada iterasi awal, nilai MSE berada di sekitar 0.00038, yang menandakan performa model belum optimal dengan kombinasi *hyperparameter* awal. Selama 20 iterasi pertama, nilai MSE mengalami penurunan bertahap, mencerminkan bahwa algoritma PSO sedang mengeksplorasi ruang solusi untuk menemukan kombinasi *hyperparameter* yang lebih baik. Setelah iterasi ke-20, terjadi penurunan signifikan dalam MSE hingga mencapai 0.00032 di sekitar iterasi ke-30. Selanjutnya, proses optimasi melambat, dan penurunan MSE menjadi lebih kecil hingga mendekati konvergensi pada iterasi ke-80, dengan nilai MSE akhir di sekitar 0.00030. Hasil dari training model PSO-RF China dapat dilihat di tabel 4.5.

Tabel 4. 6 Hasil prediksi model PSO-RF vs Aktual untuk dataset China

***Random Forest Prediction* (effort)**	***Actual Data* (effort)**
0.0121351967	0.01329816463
0.04526187467	0.04225739092
0.2086664743	0.2400263765
0.03448132055	0.03221965784
0.5040969638	0.4614243323
…	…
0.01873259882	0.01923288273
0.01067007952	0.01271201964
0.01851312154	0.02029527054
0.3991802801	0.4116569586
0.01043939941	0.00981792871

Tabel 4.6 menunjukkan perbandingan hasil prediksi dari model PSO-RF jika dibandingkan dengan data Aktual untuk dataset China. Hasil menunjukkan model memiliki kemampuan yang cukup baik untuk memprediksi pola pada dataset ini, meskipun terdapat beberapa perbedaan yang signifikan. Dalam beberapa kasus, model menghasilkan prediksi yang sangat dekat dengan nilai aktual, seperti nilai aktual 0.0132 yang diprediksi 0.0121, atau nilai aktual 0.0322 yang diprediksi 0.0344. Hal ini menunjukkan kemampuan model untuk menangkap pola secara akurat pada data tertentu.

Namun, terdapat beberapa kasus di mana perbedaan antara nilai prediksi dan aktual cukup besar, seperti nilai aktual 0.2400 yang diprediksi 0.2086, atau nilai aktual 0.4614 yang diprediksi 0.5040. Kesalahan ini mengindikasikan bahwa model mungkin kesulitan menangkap kompleksitas hubungan tertentu dalam data, terutama pada data dengan nilai effort yang lebih besar. Pada data akhir, prediksi model tetap konsisten mendekati nilai aktual. Sebagai contoh, nilai aktual 0.0192 diprediksi 0.0187, dan nilai aktual 0.4116 diprediksi 0.3991. Hal ini mencerminkan kemampuan model untuk menghasilkan estimasi yang relatif akurat pada sebagian besar data uji. Untuk visualisasi hasil prediksi vs aktual dari model PSO-RF ini dapat dilihat di gambar 4.6.



Gambar 4. 6. Visualisasi komparasi data prediksi PSO-RF vs aktual untuk nasa93

Lankah terakhir untuk melakukan evaluasi model yaitu dengan menghitung *Root Mean Squared Error* (RMSE) menggunakan data prediksi PSO-RF dan data aktual untuk nasa93 dataset. Hasil RMSE dapat dilihat di tabel .

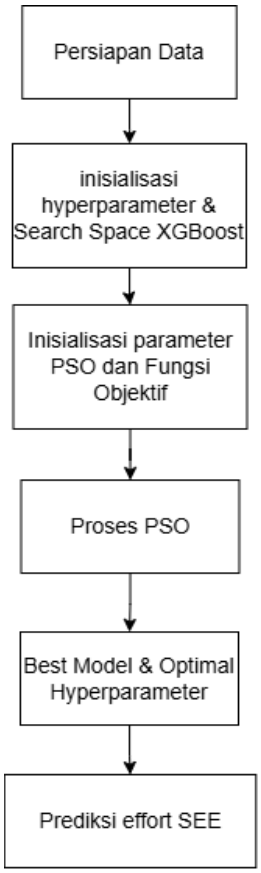
Tabel 4. 7. Hasil RMSE model PSO-RF untuk nasa93

Model	**Dataset**	**RMSE**
PSO-*Random Forest*	China	0.0167

BAB V
PSO-*EXTREME GRADIENT BOOSTING*

5.1. Desain Metode PSO – *Extreme Gradient Boosting*

Pada bab ini akan membahas secara rinci tentang metode *Extreme Gradient Boosting (XGBoost)* dengan Optimasi menggunakan *Particle Swarm Optimization* yang digunakan untuk prediksi SEE dengan dataset Nasa93 dan China Dataset. Strategi yang digunakan digambarkan di gambar 5.1.



Gambar 5. 1 Desain metode PSO-XGBoost

Berdasarkan gambar 4.1 dimulai dengan mendefinisikan masalah estimasi software effort dan menentukan metode optimasi yang digunakan, yaitu XGBoost dengan optimasi PSO.

5.1.1. Persiapan Data

Pada tahap ini akan melakukan *feature extraction* dataset untuk *Software Effort Estimation* (SEE). Kemudian membagi dataset menjadi data latih (*training set*) dan data uji (*test set*). Langkah selanjutnya yaitu *Data preprocessing* menggunakan *Min-Max Scaling*.

5.1.2. Inisialisasi *Hyperparameter* dan Search Space

hyperparameter yang dioptimalkan yaitu n_estimators, max_depth, min_samples_split, dan random_state. lower bounds dan upper bounds untuk masing-masing *hyperparameter* ditetapkan di tabel 5.1.

Tabel 5. 1 *Hyperparameter Random Forest*

Hyperparameter	***Lower Bound***	***Upper Bound***	**Deskripsi**
n_estimators	50	300	Jumlah pohon dalam XGBoost *tree*. Semakin banyak pohon, semakin baik akurasinya, tetapi dengan biaya waktu yang lebih tinggi.
max_depth	3	8	Kedalaman maksimum setiap pohon. Semakin besar nilai ini, semakin kompleks pohonnya, tetapi bisa menyebabkan overfitting.
Learning_rate	0.05	0.3	Laju pembelajaran untuk menyesuaikan bobot/*weight*.

5.1.3. Inisialisasi Fungsi Objektif

Fungsi objektif yang digunakan untuk menghitung *loss function* dari model adalah nilai *Mean Squared Error* (MSE).

$$\text{MSE} \quad = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{5.1}$$

Keterangan :

MSE = *Mean Squared Error*

n = jumlah data

Yi = nilai aktual

Y^I = nilai prediksi

Model *Random Forest* Regressor dilatih menggunakan data latih, dan nilai MSE dihitung berdasarkan prediksi data uji.

5.1.4. Inisialisasi PSO

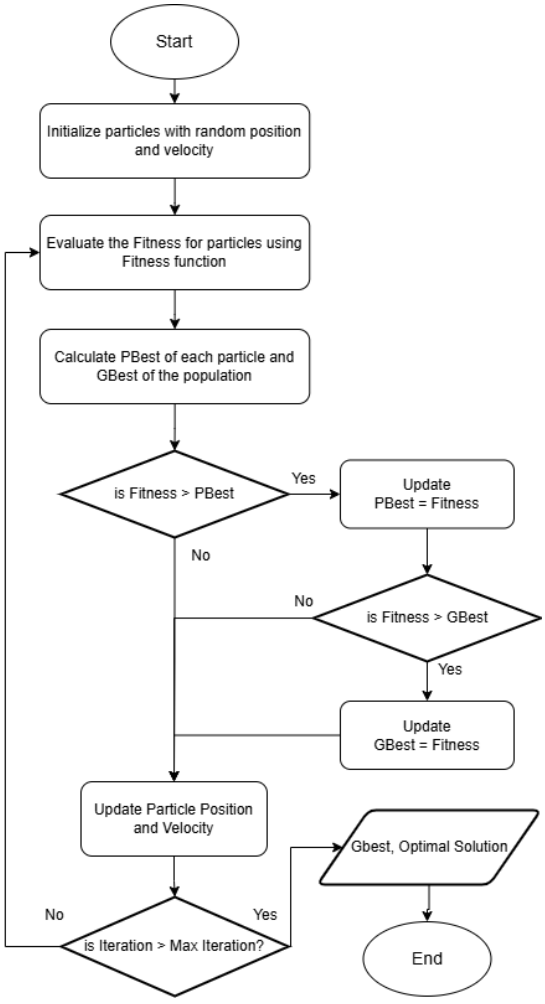
Menentukan parameter PSO yaitu jumlah partikel (n_particles), dimensi (dimensions), dan parameter eksplorasi seperti w, c1, dan c2. Untuk parameter PSO dapat dilihat di tabel 5.2.

Tabel 5. 2 Parameter PSO

Parameter	**Nilai**
Jumlah partikel (n_particles)	30
Dimensi (dimensions)	4
Inersia (w)	0.9
Koefisien Kognitif (c1)	0.5
Koefisien Sosial (c2)	0.3
Jumlah iterasi	100

5.1.5. Proses Optimasi PSO

Pada tiap iterasinya setiap partikel menghasilkan kombinasi *hyperparameter* berdasarkan ruang pencarian. Nilai MSE dihitung menggunakan fungsi objektif. Posisi terbaik partikel dan populasi diperbarui. Ulangi proses hingga iterasi mencapai batas maksimum atau konvergensi tercapai. Untuk alur PSO dapat dilihat Digambar 4.3.



Gambar 5. 2. Alur PSO

5.1.6. Model Optimal

Menggunakan *hyperparameter* terbaik (xgboost_best_pos) untuk membangun model *XGBoost Regressor*. Model dilatih kembali dengan data latih (*Training set*).

5.1.7. Evaluasi Model

Menggunakan model dengan *hyperparameter* optimal untuk memprediksi data uji. Hitung nilai evaluasi, menggunakan *Mean Squared Error* (MSE), untuk menilai kinerja model.

5.2. Implementasi

Training model dilakukan, Model PSO-*Extreme Gradient Boosting* dioptimasi menggunakan parameter yang dapat dilihat di tabel 5.3.

Tabel 5. 3 Rekapitulasi pengaturan *hyperparameter* XGBoost dan parameter PSO

Hyperparameter	**Nilai**
Hyperparameter XGB	*n_estimators,max_depth, learning_rate*
Lower Bounds	100, 3, 0.05
Upper Bounds	250, 8, 0.3
Fungsi Objektif	*Mean Squared Error (MSE)*
Jumlah Partikel (*n_particles*)	30
Dimensi (*dimensions*)	4
Inersia (w)	0.9
Koefisien Kognitif (c1)	0.5
Koefisien Sosial (c2)	0.3
Jumlah iterasi (iters)	100

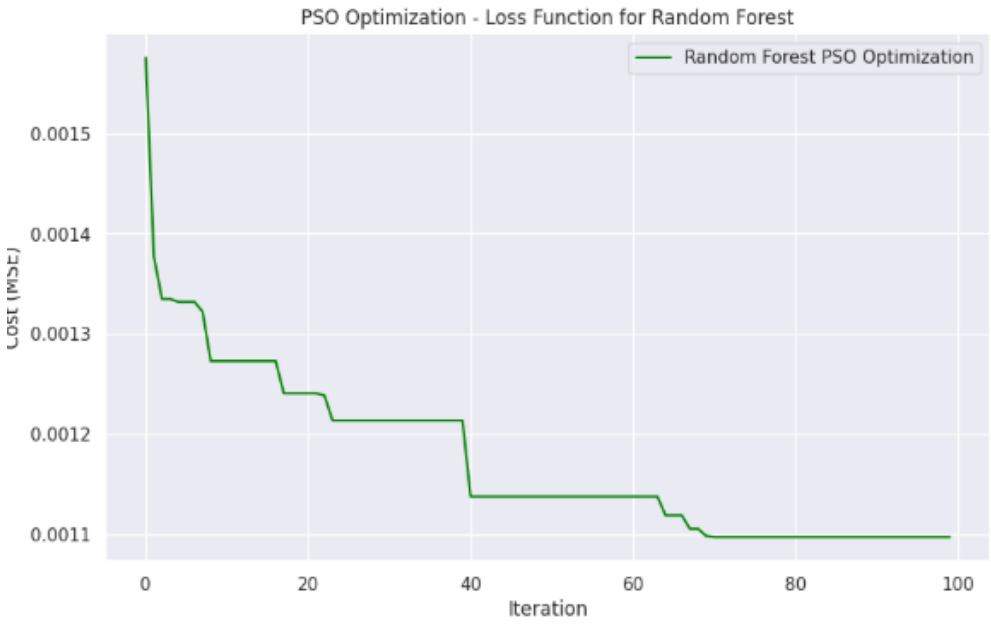
5.3. Uji Coba

Training Model dilakukan menggunakan rasio data pelatihan dan pengujian sebesar 80:20, yaksi 80% untuk data pelatihan dan 20% untuk data uji dari hasil *preprocessing* yang selanjutnya digunakan untuk proses pengujian model. Selain

itu, proses *training* dilakukan untuk setiap iterasi PSO yang akan memvarisasikan *hyperparameter* dari model *Random Forest.*

5.3.1. Hasil Optimasi PSO-XGBoost NASA93

Hasil dari optimasi *model* PSO-XGBoost dengan dataset nasa93 dapat dilihat di gambar 5.3.



Gambar 5. 3. Visualisasi proses optimasi PSO untuk model PSO-XGBoost untuk dataset nasa93

Gambar 5.3. menunjukkan proses optimasi model *Extreme Gradient Boosting (XGBoost)* menggunakan Particle Swarm Optimization (PSO) pada data NASA93 Dataset untuk estimasi Software Effort. Sumbu horizontal merepresentasikan jumlah iterasi (hingga 100 iterasi), sedangkan sumbu vertikal menunjukkan nilai cost yang digunakan, yaitu *Mean Squared Error* (MSE).

Dari grafik ini, terlihat bahwa pada awal iterasi (0–10 iterasi), nilai MSE mengalami penurunan yang signifikan. Hal ini menunjukkan bahwa PSO berhasil

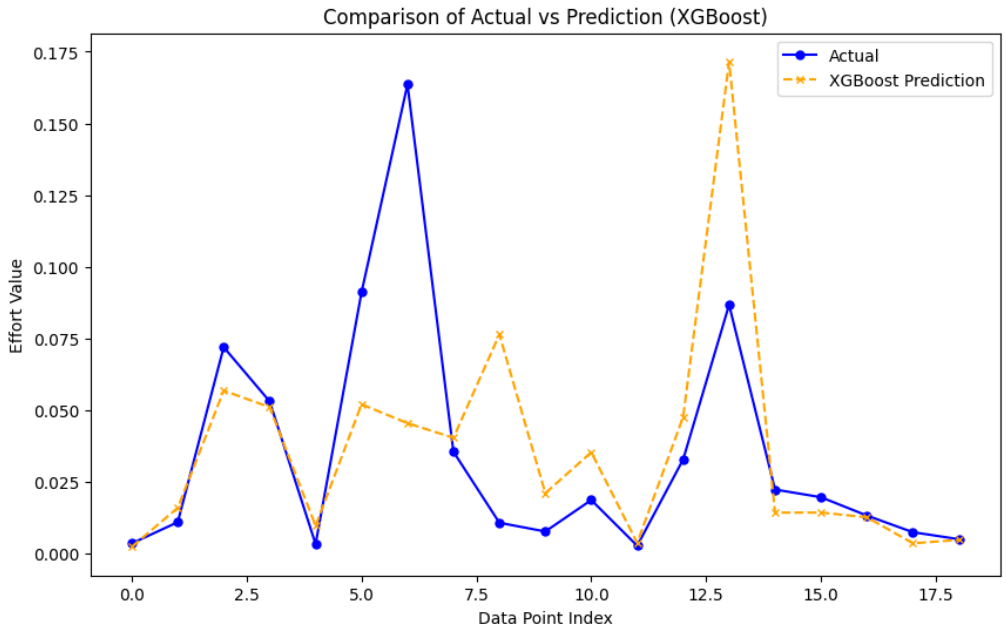
menemukan kombinasi *hyperparameter* awal yang lebih optimal dibandingkan inisialisasi awal. Selanjutnya, penurunan MSE mulai melambat pada iterasi 20–50, menandakan bahwa algoritma PSO mulai mendekati nilai konvergensi. Setelah iterasi ke-60, nilai MSE menjadi stabil di sekitar 0.0011, yang mengindikasikan bahwa kombinasi *hyperparameter* terbaik telah ditemukan oleh PSO untuk model *Random Forest*. Hasil dari training model PSO-XGboost nasa93 dapat dilihat di tabel 5.4.

Tabel 5. 4. Hasil prediksi model PSO-XGBoost vs Aktual untuk dataset nasa93

XGBoost Prediction **(act_effort)**	***Actual Data*** **(act_effort)**
0.002429616638	0.003608611904
0.01601126976	0.01102089581
0.05686497688	0.07200156048
0.05109214783	0.05310511301
0.009728044271	0.003364786775
…	…
0.01435205154	0.02238314681
0.01435205154	0.01970107039
0.01270530187	0.01331285202
0.00364282215	0.007509813961
0.004829321057	0.005071562675

Tabel 5.4 menunjukkan perbandingan hasil prediksi dari model PSO-XGBoost jika dibandingkan dengan data Aktual untuk dataset nasa93. Hasil menunjukkan beberapa prediksi menyimpang secara signifikan dari nilai aktual. Sebagai contoh, untuk data ke-1, model memprediksi 0.0024, sementara nilai aktualnya adalah 0.0036, menunjukkan selisih yang relatif kecil. Namun, pada data ke-3, prediksi model adalah 0.0569, yang berbeda cukup signifikan dari nilai aktual

0.0720. Di sisi lain, terdapat juga beberapa kasus di mana prediksi mendekati nilai aktual, seperti pada data ke-4 (prediksi 0.0511 dan aktual 0.0531), yang menunjukkan kinerja yang baik pada titik tersebut. Untuk visualisasi hasil prediksi vs aktual dari model PSO-XGBoost ini dapat dilihat di gambar 4.4.



Gambar 5. 4 Visualisasi komparasi data prediksi PSO-XGBoost vs aktual untuk nasa93

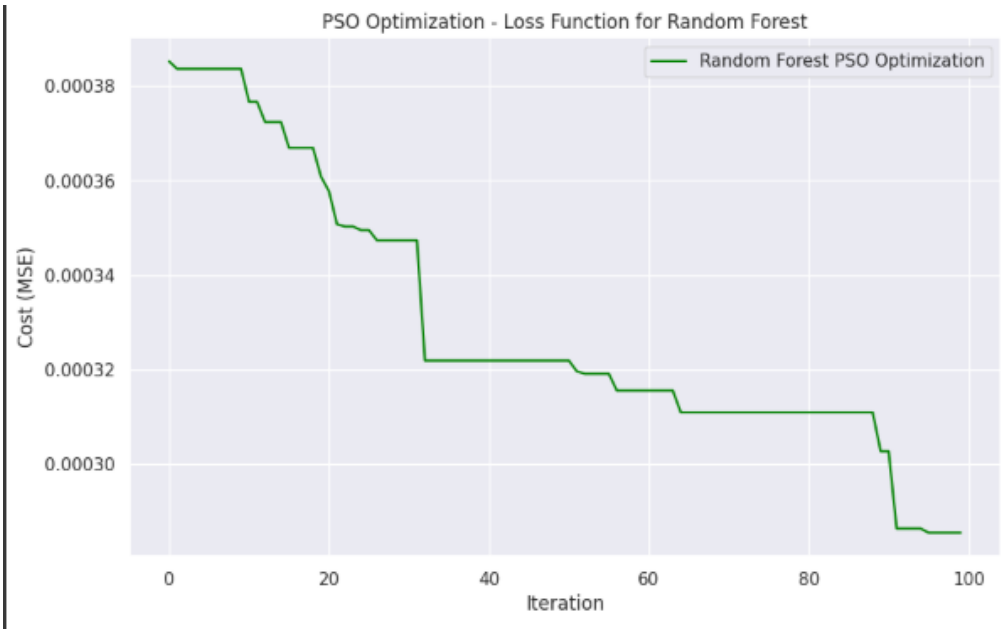
Lankah terakhir untuk melakukan evaluasi model yaitu dengan menghitung *Root Mean Squared Error* (RMSE) menggunakan data prediksi PSO-XGBoost dan data aktual untuk nasa93 dataset. Hasil RMSE dapat dilihat di tabel .

Tabel 5. 5. Hasil RMSE model PSO-XGBoost untuk nasa93

Model	**Dataset**	**RMSE**
PSO-XGBoost	Nasa93	0.0384

5.3.2. Hasil Optimasi PSO-XGBoost China

Hasil dari optimasi *model* PSO-XGBoost dengan dataset China dapat dilihat di gambar 5.5.



Gambar 5.5. Visualisasi proses optimasi PSO untuk model PSO-XGBoost untuk dataset China

Gambar 5.5 menunjukkan proses optimasi model *Extreme Gradient Boosting (XGBoost)* menggunakan *Particle Swarm Optimization* (PSO) untuk pelatihan pada dataset China Dataset dalam tugas prediksi Software Effort Estimation. Sumbu horizontal menggambarkan iterasi optimasi (dari 0 hingga 100 iterasi), sementara sumbu vertikal menunjukkan nilai cost, yang diukur dalam bentuk *Mean Squared Error* (MSE). Pada iterasi awal, nilai MSE berada di sekitar 0.00038, yang menandakan performa model belum optimal dengan kombinasi *hyperparameter* awal. Selama 20 iterasi pertama, nilai MSE mengalami penurunan bertahap, mencerminkan bahwa algoritma PSO sedang mengeksplorasi ruang solusi untuk menemukan kombinasi *hyperparameter* yang lebih baik. Setelah iterasi ke-

20, terjadi penurunan signifikan dalam MSE hingga mencapai 0.00032 di sekitar iterasi ke-30. Selanjutnya, proses optimasi melambat, dan penurunan MSE menjadi lebih kecil hingga mendekati konvergensi pada iterasi ke-80, dengan nilai MSE akhir di sekitar 0.00030. Hasil dari training model PSO-XGboost China dapat dilihat di tabel 5.6.

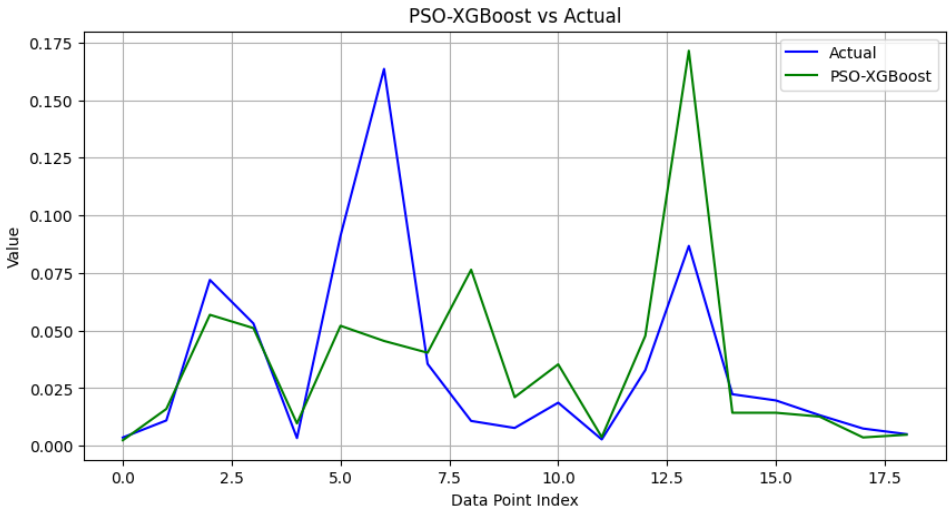
Tabel 5. 6 Hasil prediksi model PSO-XGBoost vs Aktual untuk dataset China

***XGBoost Prediction* (effort)**	***Actual Data* (effort)**
0.01410262659	0.01329816463
0.04564544559	0.04225739092
0.2051182538	0.2400263765
0.03631743416	0.03221965784
0.421416074	0.4614243323
…	…
0.02304049209	0.01923288273
0.01006335765	0.01271201964
0.01989921555	0.02029527054
0.3974200785	0.4116569586
0.0108309621	0.00981792871

Tabel 5.6 menunjukkan perbandingan hasil prediksi dari model PSO-XGBoost jika dibandingkan dengan data Aktual untuk dataset China. Hasil menunjukkan model mampu menangkap tren nilai aktual dengan cukup baik, meskipun terdapat beberapa perbedaan antara prediksi dan data aktual. Sebagai contoh, untuk data pertama, prediksi adalah 0.0141, yang sangat dekat dengan nilai aktual 0.0133, menunjukkan akurasi yang tinggi. Namun, pada data ke-3, terdapat deviasi yang lebih besar, di mana prediksi model adalah 0.2051 sementara nilai aktual adalah 0.2400. Demikian pula, pada data ke-5, prediksi model 0.4214

berbeda dari nilai aktual 0.4614, meskipun pola keseluruhan tetap mengikuti arah data aktual.

Pada data lain, model menunjukkan kinerja yang lebih stabil, seperti pada data ke-9 (prediksi 0.3974 dan aktual 0.4117), dengan selisih yang relatif kecil. Hal ini menunjukkan bahwa meskipun terdapat kesalahan prediksi, model secara keseluruhan dapat memberikan estimasi yang cukup mendekati nilai aktual. Untuk visualisasi hasil prediksi vs aktual dari model PSO-XGBoost ini dapat dilihat di gambar 5.6.



Gambar 5. 6. Visualisasi komparasi data prediksi PSO-XGBoost vs aktual untuk China

Langkah terakhir untuk melakukan evaluasi model yaitu dengan menghitung *Root Mean Squared Error* (RMSE) menggunakan data prediksi PSO-XGBoost dan data aktual untuk China dataset. Hasil RMSE dapat dilihat di tabel .

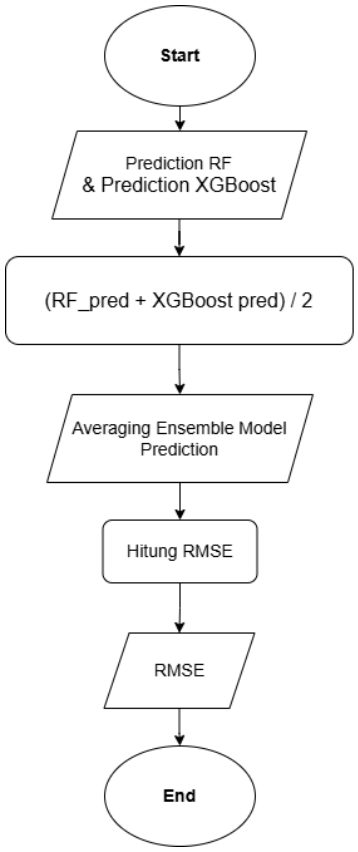
Tabel 4. 8 Hasil RMSE model PSO-RF untuk nasa93

Model	**Dataset**	**RMSE**
PSO-XGBoost	China	0.0199

BAB VI
AVERAGING ENSEMBLE MODEL

6.1. Desain Metode *Averaging Ensemble Model*

Setelah mendapatkan hasil prediksi Software Effort Estimation (SEE) dari model PSO-RF dan PSO-XGBoost untuk dataset Nasa93 dan China Dataset, Langkah selanjutnya yaitu melakukan proses *Averaging Prediction* untuk *Averaging Ensemble Model* dengan tahapan di gambar 6.1.



Gambar 6. 1 Langkah Langkah *Averaging Ensemble Model*

Gambar 6.1 menjelaskan Langkah-langkah penggabungan hasil prediksi dari dua model yang berbeda, yaitu *Random Forest* (RF) dan XGBoost (XGB), serta

mengevaluasi performa gabungan model tersebut menggunakan *Root Mean Squared Error* (RMSE).

6.2. Uji Coba

Pada tahap ini peneliti penggabungan hasil prediksi dari dua model yang berbeda, yaitu *Random Forest* (RF) dan *Extreme Gradient Boosting* (XGB) yang telah dioptimasi dengan *Particle Swarm Optimization* (PSO) dengan nama *Averaging Ensemble Model Prediction.*

6.2.1. Hasil uji Coba *Averaging Ensemble Model* nasa93

Hasil uji coba *Averaging Ensemble Model* untuk dataset nasa93 dapat diilihat di tabel 6.1.

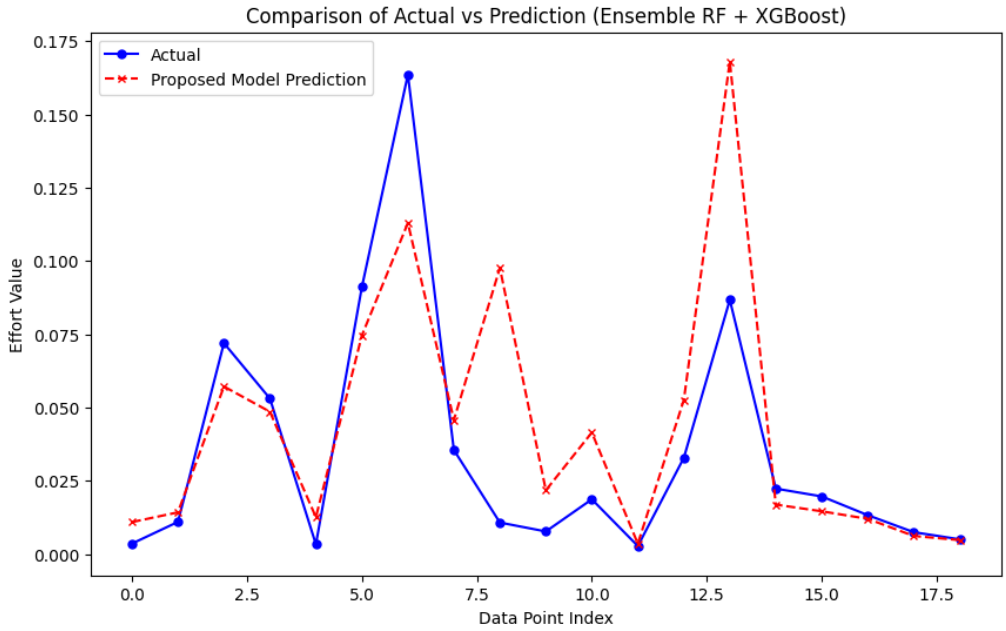
Tabel 6. 1 Hasil prediksi *Averaging Ensemble Model* vs Actual data nasa93

Averaging Ensemble Model **(act_effort)**	***Actual Data*** **(act_effort)**
0.01095659079	0.003608611904
0.01427342767	0.01102089581
0.05721077422	0.07200156048
0.04859040643	0.05310511301
0.01262221894	0.003364786775
…	…
0.01691596885	0.02238314681
0.01462866203	0.01970107039
0.01205400149	0.01331285202
0.006219424656	0.007509813961
0.004807986474	0.005071562675

Tabel 6.1 menunjukkan perbandingan hasil prediksi dari *Averaging Ensemble Model* jika dibandingkan dengan data Aktual untuk dataset nasa93. Hasil menunjukkan model ini cenderung mengalami kesalahan prediksi yang bervariasi,

dengan beberapa prediksi cukup mendekati nilai aktual dan lainnya cukup jauh. Misalnya, pada data pertama, prediksi model 0.01096 lebih tinggi dibandingkan dengan nilai aktual 0.00361, menunjukkan deviasi yang cukup besar. Hal serupa terjadi pada data kedua, di mana prediksi model 0.01427 lebih tinggi dari nilai aktual 0.01102. Namun, pada data ketiga, perbedaan antara prediksi dan nilai aktual lebih kecil, dengan prediksi model 0.0572 dan nilai aktual 0.0720.

Pada beberapa titik lain, *model Averaging Ensemble* juga menunjukkan kesalahan yang relatif kecil, seperti pada data ke-8, dengan prediksi 0.01463 dan nilai aktual 0.01970. Meskipun ada kesalahan pada beberapa data, model ini tampaknya cukup baik dalam menangkap tren umum dari data aktual, meskipun dengan sedikit pergeseran pada estimasi. Untuk visualisasi hasil prediksi vs aktual dari model *Averaging Ensemble Model* ini dapat dilihat di gambar 6.2.



Gambar 6. 2. Visualisasi komparasi data prediksi *Averaging Ensemble Model* vs aktual untuk nasa93

Lankah terakhir untuk melakukan evaluasi model yaitu dengan menghitung *Root Mean Squared Error* (RMSE) menggunakan data prediksi *Averaging Ensemble Model* dan data aktual untuk nasa93 dataset. Hasil RMSE dapat dilihat di tabel 5.5.

Tabel 5. 7. Hasil RMSE model *Averaging Ensemble Model* untuk nasa93

Model	**Dataset**	**RMSE**
Averaging Ensemble Model	Nasa93	0.0313

6.2.1. Hasil uji Coba *Averaging Ensemble Model* China

Hasil uji coba *Averaging Ensemble Model* untuk dataset China dapat diilihat di tabel 6.2.

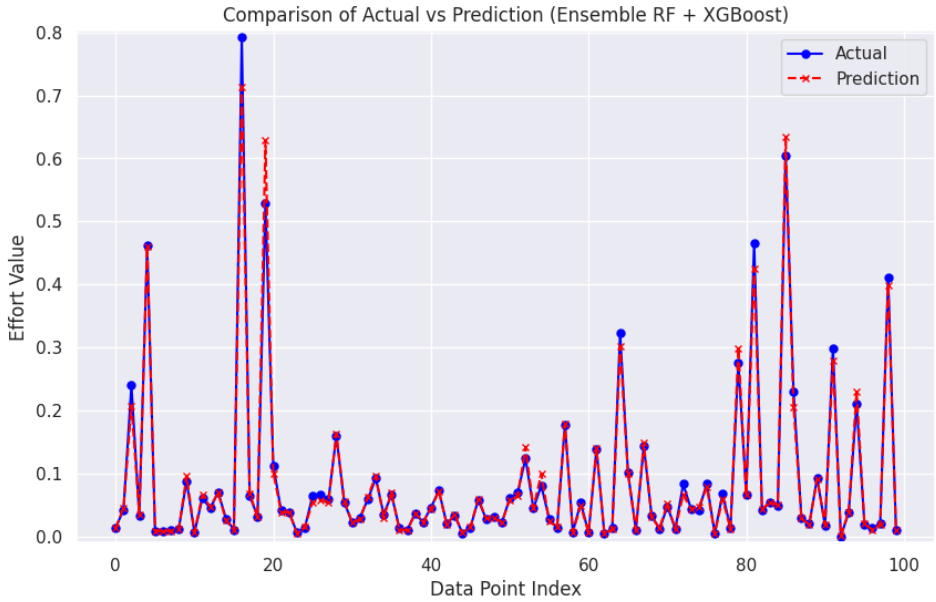
Tabel 6. 2. Hasil prediksi *Averaging Ensemble Model* vs Actual data China

***Averaging Ensemble Model* (effort)**	***Actual Data* (effort)**
0.01311891165	0.01329816463
0.04545366013	0.04225739092
0.2068923641	0.2400263765
0.03539937736	0.03221965784
0.4627565189	0.4614243323
…	…
0.02088654545	0.01923288273
0.01036671859	0.01271201964
0.01920616854	0.02029527054
0.3983001793	0.4116569586
0.01063518075	0.00981792871

Tabel 6.1 menunjukkan perbandingan hasil prediksi dari *Averaging Ensemble Model* jika dibandingkan dengan data Aktual untuk dataset China. Hasil menunjukkan model ini memiliki kinerja yang cukup baik dalam memprediksi nilai

dengan deviasi yang relatif kecil pada sebagian besar data. Sebagai contoh, pada data pertama, prediksi model 0.0131 sangat dekat dengan nilai aktual 0.0133, menunjukkan akurasi yang tinggi. Begitu juga pada data kedua, dengan prediksi 0.0455 dan nilai aktual 0.0423, perbedaan antara keduanya relatif kecil. Namun, pada data ketiga, model mengalami kesalahan yang lebih besar, di mana prediksi 0.2069 cukup jauh dari nilai aktual 0.2400.

Pada beberapa titik lainnya, seperti data keempat (prediksi 0.0354 dan aktual 0.0322) dan data kelima (prediksi 0.4628 dan aktual 0.4614), kesalahan yang terjadi masih cukup dapat diterima dan tidak terlalu signifikan. Namun, terdapat juga beberapa titik dengan kesalahan prediksi yang lebih tinggi, seperti pada data kesembilan, di mana prediksi 0.3983 sedikit lebih rendah dari nilai aktual 0.4117. Untuk visualisasi hasil prediksi vs aktual dari model *Averaging Ensemble Model* ini dapat dilihat di gambar 6.3.



Gambar 6. 3. Visualisasi komparasi data prediksi *Averaging Ensemble Model* vs aktual untuk China

Lankah terakhir untuk melakukan evaluasi model yaitu dengan menghitung *Root Mean Squared Error* (RMSE) menggunakan data prediksi *Averaging Ensemble Model* dan data aktual untuk nasa93 dataset. Hasil RMSE dapat dilihat di tabel 6.3.

Tabel 6. 3. Hasil RMSE model *Averaging Ensemble Model* untuk China

Model	**Dataset**	**RMSE**
Averaging Ensemble Model	China	0.0156

BAB VII
PEMBAHASAN

Berdasarkan data perbandingan antara nilai aktual dan prediksi dari tiga model *Proposed Model*, *Random Forest* (RF), dan XGBoost dapat dilihat bahwa *Proposed Model* menunjukkan kinerja yang lebih unggul secara keseluruhan dalam memprediksi nilai aktual, meskipun ada beberapa titik di mana model lain memiliki akurasi yang lebih baik. Pada banyak data, *Proposed Model* berhasil memberikan prediksi yang lebih mendekati nilai aktual dibandingkan dengan model RF dan XGBoost. Hasil perbandingan prediksi ketiga model dapat dilihat di tabel 7.1 untuk nasa93 dataset dan tabel 7.2 untuk China dataset.

Tabel 7. 1. Perbandingan prediksi 3 model vs aktual untuk nasa93dataset

Random Forest Prediction **(act_effort)**	***XGBoost Prediction*** **(act_effort)**	***Averaging Ensemble Model*** **(act_effort)**	***Actual Data*** **(act_effort)**
0.01948356493	0.01410262659	0.01311891165	0.003608611904
0.01253558558	0.04564544559	0.04545366013	0.01102089581
0.05755657155	0.2051182538	0.2068923641	0.07200156048
0.04608866504	0.03631743416	0.03539937736	0.05310511301
0.01551639361	0.421416074	0.4627565189	0.003364786775
…	…	…	…
0.01947988617	0.02304049209	0.02088654545	0.02238314681
0.01490527251	0.01006335765	0.01036671859	0.01970107039
0.01140270111	0.01989921555	0.01920616854	0.01331285202
0.008796027162	0.3974200785	0.3983001793	0.007509813961
0.004786651892	0.0108309621	0.01063518075	0.005071562675

Dari tabel 7.1 dapat dilihat sebagai contoh, pada data pertama, prediksi *Proposed Model* (0.01096) lebih mendekati nilai aktual (0.0036) dibandingkan

dengan RF (0.0195) dan XGBoost (0.0024), meskipun terdapat sedikit perbedaan. Pada titik data lainnya, seperti data ketiga (aktual: 0.0720), *Proposed Model* (0.0572) tampil lebih akurat dibandingkan kedua model lainnya, di mana prediksi RF dan XGBoost memiliki selisih yang sedikit lebih besar. Selain itu, pada data ke-9 (aktual: 0.0108), meskipun ketiga model mengalami kesalahan yang signifikan, *Proposed Model* (0.0976) tetap menunjukkan hasil yang lebih mendekati dibandingkan RF dan XGBoost yang memiliki deviasi lebih besar.

Lebih lanjut, pada data ke-13 dan ke-15, *Proposed Model* (0.1680 dan 0.0169) memperlihatkan prediksi yang lebih akurat daripada kedua model lainnya. Meskipun model XGBoost sering kali memberikan prediksi yang mendekati nilai aktual, *Proposed Model* secara keseluruhan menunjukkan hasil yang lebih stabil dan lebih dekat dengan nilai aktual dalam berbagai titik data.

Tabel 7. 2 Perbandingan prediksi 3 model vs aktual untuk China dataset

***Random Forest Prediction* (effort)**	***XGBoost Prediction* (effort)**	***Averaging Ensemble Model* (effort)**	***Actual Data* (effort)**
0.0121351967	0.002429616638	0.01095659079	0.01329816463
0.04526187467	0.01601126976	0.01427342767	0.04225739092
0.2086664743	0.05686497688	0.05721077422	0.2400263765
0.03448132055	0.05109214783	0.04859040643	0.03221965784
0.5040969638	0.009728044271	0.01262221894	0.4614243323
…	…	…	…
0.01873259882	0.01435205154	0.01691596885	0.01923288273
0.01067007952	0.01435205154	0.01462866203	0.01271201964
0.01851312154	0.01270530187	0.01205400149	0.02029527054
0.3991802801	0.00364282215	0.006219424656	0.4116569586
0.01043939941	0.004829321057	0.004807986474	0.00981792871

Selanjutnya untuk dataset China pada tabel 7.2 menunjukkan bahwa *Averaging Ensemble Model* menunjukkan performa yang lebih konsisten dalam memberikan prediksi yang lebih mendekati nilai aktual. Sebagai contoh, pada data pertama, *Averaging Ensemble Model* (0.01096) memiliki prediksi yang lebih akurat dibandingkan dengan prediksi XGBoost (0.0024) dan *Random Forest* (0.0121), yang sedikit lebih jauh dari nilai aktual (0.0133). Ini menunjukkan bahwa model Averaging Ensemble dapat menghasilkan estimasi yang lebih andal dan stabil pada sebagian besar data.

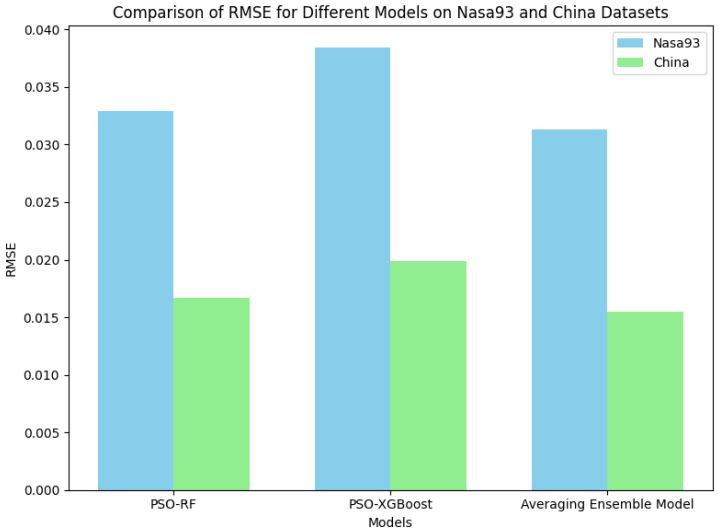
Pada beberapa titik lainnya, seperti data ketiga (aktual: 0.2400), ketiga model menunjukkan prediksi yang lebih jauh dari nilai aktual, namun *Averaging Ensemble Model* (0.0572) sedikit lebih dekat dibandingkan dengan *Random Forest* (0.2087) dan XGBoost (0.0569), meskipun kesalahan prediksi masih cukup besar. Hal serupa terjadi pada data kelima, di mana *Averaging Ensemble Model* (0.0126) memberikan prediksi yang lebih mendekati nilai aktual (0.4614) dibandingkan dengan *Random Forest* (0.5041) dan XGBoost (0.0097). selanjutnya untuk perbandingan *Root Mean Squared Error* (RMSE) dapat dilihat di tabel 7.3.

Tabel 7. 3, Perbandingan RMSE tiap model

Model	***RMSE***	Dataset
PSO-RF	0.0329	Nasa93
PSO-XGBoost	0.0384	
Averaging Ensemble Model	**0.0313**	
PSO-RF	0.0167	China
PSO-XGBoost	0.0199	
Averaging Ensemble Model	**0.0155**	

Berdasarkan tabel 7.3. dapat dilihat perbandingan Root Mean Square Error (RMSE) untuk tiga model PSO-RF, PSO-XGBoost, dan Averaging Ensemble Model dalam dua dataset yang berbeda, yaitu Nasa93 dan China. Pada dataset **Nasa93**, ***Averaging Ensemble Model*** menunjukkan RMSE terendah, yaitu 0.0313, dibandingkan dengan PSO-RF (0.0329) dan PSO-XGBoost (0.0384). Hal ini mengindikasikan bahwa pendekatan ensemble, yang menggabungkan kekuatan dari berbagai model, memberikan hasil prediksi yang lebih akurat dan lebih stabil untuk dataset ini.

Pada dataset **China**, kinerja model **Averaging Ensemble** juga lebih baik, dengan RMSE sebesar 0.0155, yang lebih rendah dibandingkan dengan PSO-RF (0.0167) dan PSO-XGBoost (0.0199). Sekali lagi, model Averaging Ensemble memberikan prediksi yang lebih mendekati nilai aktual, menunjukkan bahwa *Proposed Model* lebih efektif dalam menangani dataset ini. Untuk Grafik perbandingan dapat dilihat di gambar 7.1.



Gambar 7. 1. Grafik perbandingan RMSE tiap model

Secara keseluruhan, hasil ini menunjukkan bahwa Averaging Ensemble Model lebih unggul dibandingkan dengan model PSO-RF dan PSO-XGBoost, baik pada dataset Nasa93 maupun China, dengan kemampuan untuk menghasilkan prediksi yang lebih akurat, tercermin dari RMSE yang lebih rendah. Hal ini menegaskan bahwa menggunakan metode ensemble dapat mengurangi kesalahan prediksi dan meningkatkan akurasi dibandingkan dengan model tunggal (RF dan XGBoost).

Dalam Hukum Islam, jual beli aplikasi atau software ada kesamaan dengan jual beli salam atau bisa disebut dengan jual beli barang pesanan. Dimana penjual menjual sesuatu yang tidak dilihat zatnya, hanya ditentukan dengan sifat barang itu ada didalam pengakuan (tanggungan) si penjual. Dalam hukum Islam juga diperbolehkan jual beli online dengan mengikuti seluruh rukun dan syarat sebelum melakukan transaksi tersebut yang sudah diatur berdasarkan sumber hukum yang ada yaitu

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِذَا تَدَايَنْتُمْ بِدَيْنٍ اِلٰٓى اَجَلٍ مُّسَمًّى فَاكْتُبُوْهُۗ

”*Hai orang-orang yang beriman, apabila kamu bermuamalah tidak secara tunai untuk waktu yang ditentukan, hendaklah kalian menuliskannya*” (Al-Baqarah ayat 282).

Ayat tersebut dapat menjadi landasan hukum jual beli online dalam Islam. Selain itu, jual beli yang tidak tunai heknaknya segera ditulis agar. terhindar dari kesalahpahaman atau mencegah terjadinya kelupaan dari salah satu pihak. Jual beli salam menurut Islam terdapat beberapa rukun yang harus terpenuhi, diantaranya: 1)

sighat, yaitu ijab dan qabul; 2) aqiddani, yaitu orang yang melakukan transaksi jual beli, dalam hal ini penjual dan pembeli, dan 3) objek barang yang ingin di transaksi terkait harga dan barang yang dipesan. Adapun syarat yang harus dipenuhi yaitu: 1) uang dibayarkan terlebih dahulu; 2) barang menjadi utang bagi penjual; 3) barang diberikan sesuai dengan waktu yang sudah disepakati; 4) barang yang sudah dijanjikan harus ada, jika belum ada maka transaksi jual beli tidak sah; 5) kejelasan barang sangat diperlukan seperti ukuran, takaran dan jumlah, ketiga komponen tersebut memang sudah lumrah dan berlaku bagi proses jual beli, dan 6) sifat-sifat barang diketahui dengan jelas agar tidak menjadi perselisihan dikemudian hari.

Oleh karena itu, transaksi pembuatan software telah dilakukan sesuai dengan hukum Islam. Firman Allah SWT adalah dasar hukum wakalah

وَلَا تَأْكُلُوٓاْ أَمْوَٰلَكُم بَيْنَكُم بِٱلْبَٰطِلِ وَتُدْلُواْ بِهَآ إِلَى ٱلْحُكَّامِ لِتَأْكُلُواْ فَرِيقًا مِّنْ أَمْوَٰلِ ٱلنَّاسِ بِٱلْإِثْمِ وَأَنتُمْ تَعْلَمُونَ

”*Dan janganlah sebahagian kamu memakan harta sebahagian yang lain di antara kamu dengan jalan yang bathil dan (janganlah) kamu membawa (urusan) harta itu kepada hakim, supaya kamu dapat memakan sebahagian daripada harta benda orang lain itu dengan (jalan berbuat) dosa, padahal kamu mengetahui*”. (QS:Al-Baqarah ayat 188).

Penelitian ini sejalan dengan ajaran Islam yang mendorong transparansi, keadilan, dan kejujuran dalam segala aspek kehidupan, termasuk dalam urusan bisnis dan ekonomi. QS. Al-Baqarah (2): 188 menegaskan agar umat Islam tidak memakan harta sesama dengan cara yang tidak benar, seperti melakukan penipuan atau praktik bisnis yang tidak etis. Allah SWT melarang keras tindakan

menyimpang dalam berbisnis dan mengingatkan bahwa membawa urusan harta ke hadapan hakim dengan niat untuk mendapatkan keuntungan yang tidak halal merupakan perbuatan dosa.

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا لَا تَأْكُلُوْٓا اَمْوَالَكُمْ بَيْنَكُمْ بِالْبَاطِلِ اِلَّآ اَنْ تَكُوْنَ تِجَارَةً عَنْ تَرَاضٍ مِّنْكُمْۗ وَلَا تَقْتُلُوْٓا اَنْفُسَكُمْۗ اِنَّ اللّٰهَ كَانَ بِكُمْ رَحِيْمًا

"*Wahai orang-orang yang beriman, janganlah kamu memakan harta sesamamu dengan cara yang batil (tidak benar), kecuali berupa perniagaan atas dasar suka sama suka di antara kamu. Janganlah kamu membunuh dirimu. Sesungguhnya Allah adalah Maha Penyayang kepadamu*.". (QS. An-Nisa (2): 29).

Tawar menawar dalam Islam diperbolehkan selama dilakukan sesuai dengan prinsip-prinsip syariat Islam. Aktivitas tawar menawar dalam transaksi jual beli dianggap halal atau diperbolehkan, dengan ketentuan tidak melanggar aturan-aturan syariah, seperti larangan riba, dan dilakukan dengan prinsip kejujuran, transparansi, dan kesepakatan yang saling ridha antara penjual dan pembeli. Dalam Al-Quran, terdapat ayat yang menunjukkan bahwa tawar menawar dalam jual beli diperbolehkan, seperti QS. An-Nisa ayat 29 yang berbunyi "Hai orang-orang yang beriman, janganlah kamu saling memakan harta sesamamu dengan jalan yang batil, kecuali dengan jalan perniagaan yang berlaku dengan suka sama suka di antara kamu." Selain itu, dalam hadis, Rasulullah SAW juga mencontohkan tawar menawar dalam berbagai transaksi jual beli. Oleh karena itu, tawar menawar dalam Islam diperbolehkan asalkan memenuhi ketentuan-ketentuan tersebut.

Dalam konteks penggunaan *metode Averaging Ensemble Learning* untuk estimasi *Software Effort*, prinsip kejujuran dan keadilan yang diajarkan dalam QS. Al-Baqarah (2): 188 dapat diaplikasikan. Penelitian ini mendorong transparansi dalam penetapan harga aplikasi perangkat lunak, menghindari praktik-praktik yang dapat merugikan pihak lain, dan memastikan bahwa estimasi harga didasarkan pada data yang akurat dan metodologi yang etis. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengembangan metode Ensemble, tetapi juga menggambarkan implementasi nilai-nilai moral dan etika Islam dalam praktik bisnis di era digital. Selain itu terdapat juga firman Allah

وَيْلٌ لِّلْمُطَفِّفِيْنَۙ الَّذِيْنَ اِذَا اكْتَالُوْا عَلَى النَّاسِ يَسْتَوْفُوْنَ وَاِذَا كَالُوْهُمْ اَوْ وَّزَنُوْهُمْ يُخْسِرُوْنَۗ

" Celakalah bagi orang-orang yang curang (dalam menakar dan menimbang)! (yaitu) orang-orang yang apabila menerima takaran dari orang lain mereka minta dicukupkan, dan apabila mereka menakar atau menimbang (untuk orang lain), mereka mengurangi.". (Al-Mutaffifin ayat 1 - 3).

Penelitian ini mencoba menciptakan sebuah paradigma yang sejalan dengan nilai-nilai moral dan etika yang ditegaskan dalam QS. Al-Mutaffifin (83): 1-3. Ayat ini mengecam keras perilaku curang, terutama dalam hal menakar dan menimbang. Allah SWT menyatakan kecelakaan bagi orang-orang yang curang, yang ketika mereka menerima takaran dari orang lain, mereka meminta agar takaran tersebut cukup, tetapi ketika mereka yang menakar atau menimbang untuk orang lain, mereka justru mengurangi takaran tersebut.

BAB VIII
KESIMPULAN DAN SARAN

8.1. Kesimpulan

Berdasarkan hasil perbandingan prediksi antara model *Proposed Model*, *Random Forest* (RF), dan XGBoost pada kedua dataset, yaitu Nasa93 dan China, dapat disimpulkan bahwa ***Averaging Ensemble Model*** secara konsisten menunjukkan kinerja yang lebih unggul berdasarkan perbandingan Root Mean Square Error (RMSE), Averaging Ensemble Model juga menunjukkan hasil yang lebih baik dengan RMSE yang lebih rendah pada kedua dataset, baik untuk **Nasa93 dengan RMSE sebesar 0.0313** maupun **China dengan RMSE sebesar 0.0155**.

Meskipun ada beberapa titik data di mana model lain memberikan prediksi yang sedikit lebih akurat, Averaging Ensemble Model secara keseluruhan memberikan prediksi yang lebih mendekati nilai aktual. Hal ini terlihat pada banyak data di mana *Proposed Model* memiliki nilai prediksi yang lebih stabil dan lebih dekat dengan nilai aktual dibandingkan dengan RF dan XGBoost dengan perolehan RMSE paling rendah sebesar 0.0313 untuk Nasa93 dan 0.0155 untuk China. Oleh karena itu, dapat disimpulkan bahwa **Averaging Ensemble Model** adalah pendekatan yang lebih efektif dan dapat diandalkan untuk menghasilkan estimasi yang lebih akurat dibandingkan dengan model PSO-RF dan PSO-XGBoost.

8.2. Saran

Berdasarkan hasil yang diperoleh, ada beberapa saran yang dapat dipertimbangkan untuk penelitian selanjutnya.

A. Untuk meningkatkan kinerja model Averaging Ensemble, dapat dipertimbangkan untuk menambahkan lebih banyak model dasar (base models) selain *Random Forest* (RF) dan XGBoost, seperti *Support Vector Machines* (SVM) *atau Gradient Boosting Machines* (GBM), yang mungkin dapat memberikan hasil yang lebih akurat dan stabil.
B. Penelitian lebih lanjut dapat dilakukan dengan menggunakan dataset *Software Effort Estimation* (SEE) yang lebih beragam untuk menguji sejauh mana Averaging Ensemble Model dapat mengatasi berbagai jenis data.

DAFTAR PUSTAKA

Abdo, Amani, Omnia Abdelkader, and Laila Abdel-Hamid. 2024. "SA-PSO-GK++: A New Hybrid Clustering Approach for Analyzing Medical Data." *IEEE Access* 12(December 2023):12501–16. doi: 10.1109/ACCESS.2024.3350442.

Abdollahpour, Shamsollah, Armaghan Kosari-Moghaddam, and Mohammad Bannayan. 2020. "Prediction of Wheat Moisture Content at Harvest Time through ANN and SVR Modeling Techniques." *Information Processing in Agriculture* 7(4):500–510. doi: 10.1016/j.inpa.2020.01.003.

Ali, Syed Sarmad, Jian Ren, Kui Zhang, Ji Wu, and Chao Liu. 2023. "Heterogeneous Ensemble Model to Optimize Software Effort Estimation Accuracy." *IEEE Access* 11(March):27759–92. doi: 10.1109/ACCESS.2023.3256533.

De Carvalho, Halcyon Davys Pereira, Roberta Fagundes, and Wylliams Santos. 2021. "Extreme Learning Machine Applied to Software Development Effort Estimation." *IEEE Access* 9:92676–87. doi: 10.1109/ACCESS.2021.3091313.

Chai, T., R. R. Draxler, and Climate Prediction. 2014. "Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? – Arguments against Avoiding RMSE in the Literature." (2005):1247–50. doi: 10.5194/gmd-7-1247-2014.

Charoenkwan, Phasit, Wararat Chiangjong, Chanin Nantasenamat, Md Mehedi Hasan, Balachandran Manavalan, and Watshara Shoombuatong. 2021. "StackIL6: A Stacking Ensemble Model for Improving the Prediction of IL-6 Inducing Peptides." *Briefings in Bioinformatics* 22(6):1–13. doi: 10.1093/bib/bbab172.

Chirra, Sai Mohan Reddy, and Hassan Reza. 2019. "A Survey on Software Cost Estimation Techniques." *Journal of Software Engineering and Applications* 12(06):226–48. doi: 10.4236/jsea.2019.126014.

Fang, Jian, Hongbin Wang, Fan Yang, Kuang Yin, Xiang Lin, and Min Zhang. 2022. "A Failure Prediction Method of Power Distribution Network Based on PSO and XGBoost." *Australian Journal of Electrical and Electronics Engineering* 19(4):371–78. doi: 10.1080/1448837X.2022.2072447.

Gautam, Swarnima Singh, and Vrijendra Singh. 2022. "Adaptive Discretization Using Golden Section to Aid Outlier Detection for Software Development Effort Estimation." *IEEE Access* 10(August):90369–87. doi: 10.1109/ACCESS.2022.3200149.

Gopal, Anshul, Mohammad Mahdi Sultani, and Jagdish Chand Bansal. 2020. "On Stability Analysis of Particle Swarm Optimization Algorithm." *Arabian Journal for Science and Engineering* 45(4):2385–94. doi: 10.1007/s13369-019-03991-8.

Herni Yulianti, Sri Elina, Oni Soesanto, and Yuana Sukmawaty. 2022. "Penerapan Metode Extreme Gradient Boosting (XGBOOST) Pada Klasifikasi Nasabah Kartu Kredit." *Journal of Mathematics: Theory and Applications* 4(1):21–26. doi: 10.31605/jomta.v4i1.1792.

Hoc, Huynh Thai, Radek Silhavy, Zdenka Prokopova, and Petr Silhavy. 2023. "Comparing Stacking Ensemble and Deep Learning for Software Project Effort Estimation." *IEEE Access* 11(June):60590–604. doi: 10.1109/ACCESS.2023.3286372.

Karna, Hrvoje, Sven Gotovac, and Linda Vicković. 2020. "Data Mining Approach to Effort Modeling on Agile Software Projects." *Informatica (Slovenia)* 44(2):231–39. doi: 10.31449/inf.v44i2.2759.

Kaushik, Anupama, Prabhjot Kaur, Nisha Choudhary, and Priyanka. 2022. "Stacking Regularization in Analogy-Based Software Effort Estimation." *Soft Computing* 26(3):1197–1216. doi: 10.1007/s00500-021-06564-w.

Li, Nana, Lei Liu, Dongyao Zou, and Xing Liu. 2024. "Node Localization Algorithm for Irregular Regions Based on Particle Swarm Optimization Algorithm and Reliable Anchor Node Pairs." *IEEE Access* 12(March):37470–82. doi: 10.1109/ACCESS.2024.3374518.

Matlock, Kevin, Carlos De Niz, Raziur Rahman, Souparno Ghosh, and Ranadip Pal. 2018. "Investigation of Model Stacking for Drug Sensitivity Prediction." *BMC Bioinformatics* 19(Suppl 3). doi: 10.1186/s12859-018-2060-2.

Montero-Manso, Pablo, George Athanasopoulos, Rob J. Hyndman, and Thiyanga S. Talagala. 2020. "FFORMA: Feature-Based Forecast Model Averaging." *International Journal of Forecasting* 36(1):86–92. doi: 10.1016/j.ijforecast.2019.02.011.

Priya Varshini, A. G., K. Anitha Kumari, D. Janani, and S. Soundariya. 2021. "Comparative Analysis of Machine Learning and Deep Learning Algorithms for Software Effort Estimation." *Journal of Physics: Conference Series* 1767(1). doi: 10.1088/1742-6596/1767/1/012019.

Rahman, Mizanur, Hasan Sarwar, Abdul Kader, Teresa Goncalves, and Ting Tin Tin. 2024. "Review and Empirical Analysis of Machine Learning-Based Software Effort Estimation." *IEEE Access* 12(May):85661–80. doi: 10.1109/ACCESS.2024.3404879.

Rais, Zulkifli. 2022. "Analisis Support Vector Regression (Svr) Dengan Kernel Radial Basis Function (Rbf) Untuk Memprediksi Laju Inflasi Di Indonesia." *VARIANSI: Journal of Statistics and Its Application on Teaching and Research* 4(1):30–38. doi: 10.35580/variansiunm13.

Rajper, Samina, and Zubair A. Shaikh. 2016. "Software Development Cost Estimation: A Survey." *Indian Journal of Science and Technology* 9(31):177–205. doi: 10.17485/ijst/2016/v9i31/93058.

Raval, Pranav D., and Ashit S. Pandya. 2022. “A Hybrid PSO-ANN-Based Fault Classification System for EHV Transmission Lines.” *IETE Journal of Research* 68(4):3086–99. doi: 10.1080/03772063.2020.1754299.

Rhmann, Wasiur, Babita Pandey, and Gufran Ahmad Ansari. 2022. “Software Effort Estimation Using Ensemble of Hybrid Search-Based Algorithms Based on Metaheuristic Algorithms.” *Innovations in Systems and Software Engineering* 18(2):309–19. doi: 10.1007/s11334-020-00377-0.

Shah, Muhammad Arif, Dayang Norhayati Abang Jawawi, Mohd Adham Isa, Muhammad Younas, Abdelzahir Abdelmaboud, and Fauzi Sholichin. 2020. “Ensembling Artificial Bee Colony with Analogy-Based Estimation to Improve Software Development Effort Prediction.” *IEEE Access* 8:58402–15. doi: 10.1109/ACCESS.2020.2980236.

Shekhar, Shivangi. 2018. “Review of Various Software Cost Estimation Techniques.” (May 2016). doi: 10.5120/ijca2016909867.

Srivastava, Vibha, Vijay Kumar Dwivedi, and Ashutosh Kumar Singh. 2023. “Cryptocurrency Price Prediction Using Enhanced PSO with Extreme Gradient Boosting Algorithm.” *Cybernetics and Information Technologies* 23(2):170–87. doi: 10.2478/cait-2023-0020.

Ullah, Ashraf, Nadeem Javaid, Muhammad Umar Javed, Pamir, Byung Seo Kim, and Saeed Ali Bahaj. 2022. “Adaptive Data Balancing Method Using Stacking Ensemble Model and Its Application to Non-Technical Loss Detection in Smart Grids.” *IEEE Access* 10(November):133244–55. doi: 10.1109/ACCESS.2022.3230952.

Xie, Yuying, Yeying Zhu, Cecilia A. Cotton, and Pan Wu. 2019. “A Model Averaging Approach for Estimating Propensity Scores by Optimizing Balance.” *Statistical Methods in Medical Research* 28(1):84–101. doi: 10.1177/0962280217715487.

Yulianto, Fendy, Wayan Firdaus Mahmudy, and Arief Andy Soebroto. 2020. “Comparison of Regression, Support Vector Regression (SVR), and SVR-Particle Swarm Optimization (PSO) for Rainfall Forecasting.” *Journal of Information Technology and Computer Science* 5(3):235–47. doi: 10.25126/jitecs.20205374.

Zakaria, Noor Azura, Amelia Ritahani Ismail, Nadzurah Zainal Abidin, Nur Hidayah Mohd Khalid, and Afrujaan Yakath Ali. 2021. “Optimization of COCOMO Model Using Particle Swarm Optimization.” *International Journal of Advances in Intelligent Informatics* 7(2):177–87. doi: 10.26555/ijain.v7i2.583.

Zarkasyi, Ahmad. 2016. “Manajemen Kinerja Dalam Tafsir Al-Qur’an Dan Hadist Pendekatan Filsafat Tematik.” *Jurnal Qolamuna* 2(1):133–50.

Zhang, Bing, Huihui Ren, Guoyan Huang, Yongqiang Cheng, and Changzhen Hu.

2019. “Predicting Blood Pressure from Physiological Index Data Using the SVR Algorithm 08 Information and Computing Sciences 0801 Artificial Intelligence and Image Processing.” *BMC Bioinformatics* 20(1):1–15.