

SISTEM REKOMENDASI METODE SDLC BERBASIS SIMPLE ADDITIVE WEIGHTING DAN SIMILARITY

Muhammad Asyraf Pratama, Ryci Rahmatil Fiska

Teknik Informatika, Politeknik Negeri Bengkalis
Sungai Alam, Kec. Bengkalis, Kabupaten Bengkalis, Riau, Indonesia
masyrafpratama19@gmail.com

ABSTRAK

Pemilihan metode Software Development Life Cycle (SDLC) yang tepat merupakan faktor krusial dalam keberhasilan proyek pengembangan perangkat lunak. Namun, proses pemilihan sering dilakukan secara subjektif tanpa analisis mendalam terhadap karakteristik proyek. Penelitian ini mengembangkan sistem rekomendasi berbasis web menggunakan metode Simple Additive Weighting (SAW) yang dikombinasikan dengan pendekatan similarity untuk memberikan rekomendasi metode SDLC yang objektif dan akurat. Sistem memproses sembilan kriteria penilaian meliputi estimasi durasi proyek, tingkat risiko, kompleksitas sistem, anggaran, ukuran tim, tingkat pengalaman tim, keterlibatan pengguna, perubahan kebutuhan, dan kebutuhan dokumentasi. Sistem diimplementasikan menggunakan framework Laravel dan diuji melalui perbandingan perhitungan manual serta validasi pakar. Hasil pengujian menunjukkan akurasi perhitungan 100% dengan kesesuaian penuh antara sistem dan perhitungan manual. Perbandingan antara SAW murni dan SAW+Similarity menunjukkan peningkatan skor rekomendasi sebesar 15,56% (dari 84,07% menjadi 99,63%), dengan perubahan metode rekomendasi dari DevOps menjadi Test-Driven Development. Validasi pakar terhadap 10 kasus proyek menghasilkan tingkat kesesuaian 50%, menunjukkan bahwa sistem dapat digunakan sebagai decision support system dalam pemilihan metode SDLC.

Kata kunci : Sistem Rekomendasi, SDLC, Simple Additive Weighting, Similarity, Decision Support System

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat mendorong organisasi, institusi, dan pelaku usaha untuk mengembangkan sistem perangkat lunak yang sesuai dengan kebutuhan operasional mereka. Dalam mengembangkan perangkat lunak, diperlukan pendekatan yang sistematis dan terstruktur agar sistem yang dihasilkan dapat memenuhi kebutuhan pengguna, sesuai dengan waktu pengembangan, dan anggaran yang tersedia. Software Development Life Cycle (SDLC) merupakan serangkaian tahapan yang dilalui dalam proses pengembangan perangkat lunak dari awal hingga implementasi [1].

Terdapat berbagai metode SDLC seperti Waterfall, Agile, Scrum, Rapid Application Development (RAD), Prototyping, Spiral, Extreme Programming (XP), V-Model, Incremental, Iterative, Big Bang, DevOps, Test-Driven Development (TDD), dan Lean Development. Setiap metode memiliki karakteristik, kelebihan, dan keterbatasan yang berbeda. Waterfall cocok untuk proyek dengan kebutuhan yang stabil [2], sementara Agile dan Scrum lebih fleksibel untuk proyek dengan kebutuhan yang sering berubah. Spiral mengutamakan identifikasi risiko pada setiap iterasi, sedangkan RAD dan Prototyping efektif ketika feedback pengguna diperlukan dalam pengembangan.

Pemilihan metode SDLC yang tepat sangat penting karena berdampak langsung pada keberhasilan proyek [1]. Namun, banyak pengembang, khususnya pada level menengah dan pemula, mengalami kesulitan dalam menentukan metode yang sesuai dengan kondisi dan kebutuhan proyek. Kesalahan

dalam memilih metode dapat menyebabkan keterlambatan, pembengkakan anggaran, atau bahkan kegagalan proyek. Penelitian terdahulu menunjukkan bahwa pemilihan metode SDLC harus mempertimbangkan berbagai faktor seperti ukuran tim, durasi proyek, anggaran, kompleksitas sistem, tingkat perubahan kebutuhan, keterlibatan pengguna, dan tingkat risiko [1][3][4].

Beberapa penelitian telah mengembangkan sistem pendukung keputusan untuk pemilihan metode SDLC menggunakan berbagai pendekatan. Seniv [5] menggunakan metode Analytic Hierarchy Process (AHP) dan TOPSIS dengan akurasi 83%, namun pendekatan ini memiliki kompleksitas yang relatif tinggi. Penelitian lain menggunakan machine learning seperti decision tree dan random forest, tetapi memerlukan dataset yang besar dan validasi lebih lanjut [13]. Sementara itu, metode Simple Additive Weighting (SAW) telah terbukti efektif dalam sistem rekomendasi multikriteria dengan kesederhanaan, transparansi, dan kemudahan implementasi [6][7][8].

Penelitian ini mengembangkan sistem rekomendasi yang menggabungkan metode SAW dengan pendekatan similarity untuk meningkatkan akurasi rekomendasi. Berbeda dengan penelitian sebelumnya yang hanya menggunakan SAW murni, pendekatan yang diusulkan menambahkan dimensi pembelajaran dari karakteristik metode SDLC yang telah terdefinisi, sehingga menghasilkan rekomendasi yang lebih kontekstual dan berbasis data empiris. Sistem diimplementasikan berbasis web menggunakan framework Laravel, memproses sembilan kriteria

penilaian, dan memberikan 14 alternatif metode SDLC sebagai output rekomendasi.

Kontribusi utama penelitian ini adalah: (1) pengembangan sistem rekomendasi SDLC berbasis SAW dengan pendekatan similarity yang meningkatkan akurasi rekomendasi; (2) validasi komprehensif melalui perbandingan perhitungan manual dan validasi pakar; (3) analisis komparatif antara SAW murni dan SAW+Similarity yang menunjukkan peningkatan signifikan dalam kualitas rekomendasi.

2. TINJAUAN PUSTAKA

2.1. Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan proses sistematis yang digunakan dalam pengembangan perangkat lunak, mencakup tahapan perencanaan, analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan [1]. Tujuan utama SDLC adalah menghasilkan perangkat lunak berkualitas yang memenuhi kebutuhan pengguna dengan efisien dalam penggunaan sumber daya.

Terdapat berbagai metode SDLC yang telah dikembangkan dengan karakteristik dan keunggulan masing-masing. Metode tradisional seperti Waterfall menggunakan pendekatan sekuensial yang cocok untuk proyek dengan kebutuhan stabil dan terdefinisi dengan jelas [2][9][10]. V-Model menekankan pada verifikasi dan validasi di setiap tahapan pengembangan [11]. Sementara metode modern seperti Agile, Scrum, dan Extreme Programming (XP) lebih fleksibel dan cocok untuk proyek dengan perubahan kebutuhan yang tinggi [12][13].

2.2. Penelitian Terkait Pemilihan Metode SDLC

Beberapa penelitian telah dilakukan untuk mengembangkan sistem pendukung keputusan dalam pemilihan metode SDLC. Hossain [1] melakukan systematic literature review terhadap berbagai metodologi SDLC dan menyimpulkan bahwa pemilihan metode harus disesuaikan dengan karakteristik proyek seperti kompleksitas sistem, stabilitas kebutuhan, dan keterlibatan pengguna. Penelitian ini menekankan pentingnya penggunaan decision support matrix untuk membantu menentukan metode yang paling sesuai.

Paksi et al. [3] melakukan survei metode-metode SDLC menggunakan Systematic Literature Review (SLR) dan menyimpulkan pentingnya kriteria seperti skala proyek, risiko, dan tingkat perubahan kebutuhan dalam pemilihan metode. Siva et al. [7] membandingkan berbagai metodologi pengembangan perangkat lunak dan menyimpulkan bahwa pemilihan metode harus disesuaikan dengan karakteristik spesifik proyek.

Mishra dan Alzoubi [14] melakukan perbandingan antara metode structured dan Agile menggunakan pendekatan decision tree, menyimpulkan bahwa Agile

cenderung lebih sukses dalam proyek dengan kebutuhan yang sering berubah. Namun, penelitian ini belum mengembangkan sistem rekomendasi otomatis yang dapat digunakan oleh praktisi.

Seniv [15] mengembangkan metode pemilihan metodologi pengembangan perangkat lunak menggunakan kombinasi Analytic Hierarchy Process (AHP) dan Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) berdasarkan karakteristik proyek, menghasilkan akurasi hingga 83%. Namun, pendekatan ini memiliki kompleksitas perhitungan yang relatif tinggi dan kurang transparan bagi pengguna umum.

2.3. Metode Simple Additive Weighting (SAW)

Simple Additive Weighting (SAW) merupakan salah satu teknik dalam Multi-Criteria Decision Making (MCDM) yang digunakan untuk memberikan keputusan berdasarkan sejumlah alternatif dan kriteria [6]. SAW dikenal sebagai metode penjumlahan terbobot karena cara kerjanya menjumlahkan nilai setiap alternatif yang telah dikalikan dengan bobot preferensi untuk masing-masing kriteria.

Beberapa penelitian telah menunjukkan efektivitas SAW dalam sistem rekomendasi. Permana et al. [6] menerapkan metode SAW untuk membangun sistem rekomendasi dalam memilih karyawan terbaik untuk proyek pengembangan perangkat lunak dengan mempertimbangkan lima kriteria utama. Hasil penelitian menunjukkan bahwa metode SAW efektif dalam menyusun peringkat dan memberikan rekomendasi yang objektif dan akurat berdasarkan pembobotan nilai.

Nubatonis [7] membandingkan tiga metode pengambilan keputusan multikriteria yaitu AHP, SAW, dan TOPSIS untuk pemilihan tempat wisata terbaik. Hasil analisis menunjukkan bahwa metode SAW menghasilkan nilai preferensi tertinggi (0,9677) dan mampu memberikan hasil yang paling konsisten dengan realita di lapangan. Hal ini menegaskan bahwa SAW cocok digunakan dalam kasus yang memiliki data kriteria yang terstruktur dan bobot yang jelas.

Pramartha et al. [15] membahas pengembangan sistem rekomendasi berbasis web semantik yang dikombinasikan dengan metode SAW untuk meningkatkan akurasi rekomendasi berdasarkan profil pengguna. Hasil penelitian menunjukkan bahwa pendekatan ini mampu menghasilkan rekomendasi yang relevan dan sesuai dengan kebutuhan pengguna.

Baldanullah et al. [8] melakukan perbandingan antara metode SAW, Weighted Product (WP), dan TOPSIS dalam menentukan jurnal terbaik. Penelitian ini menunjukkan bahwa meskipun terdapat sedikit perbedaan skor, hasil peringkat yang diberikan oleh SAW sangat selaras dengan metode lainnya dan lebih mudah diimplementasikan dalam sistem berbasis web. SAW dinilai lebih transparan dan tidak terlalu kompleks dibandingkan WP dan TOPSIS.

2.4. Gap Penelitian

Berdasarkan kajian literatur yang telah dilakukan, dapat diidentifikasi beberapa gap penelitian:

- Kompleksitas Metode: Penelitian sebelumnya yang menggunakan AHP dan TOPSIS [5][16] menghasilkan akurasi yang baik namun memiliki kompleksitas perhitungan yang tinggi dan kurang transparan bagi pengguna umum.
- Pendekatan Statis: Sebagian besar penelitian menggunakan pendekatan berbasis aturan statis tanpa mempertimbangkan similarity atau kemiripan karakteristik proyek dengan profil ideal metode SDLC.
- Keterbatasan Validasi: Penelitian yang mengembangkan sistem rekomendasi [14][15] belum melakukan validasi komprehensif melalui perbandingan perhitungan manual dan validasi pakar.
- Konteks Aplikasi: Penelitian SAW sebelumnya [6][7][8] diterapkan pada konteks yang berbeda (pemilihan karyawan, tempat wisata, jurnal), belum spesifik untuk pemilihan metode SDLC.
- Analisis Komparatif: Belum ada penelitian yang secara spesifik membandingkan pendekatan SAW murni dengan SAW yang dikombinasikan dengan metode similarity dalam konteks pemilihan metode SDLC.

2.5. Posisi Penelitian

Penelitian ini mengisi gap yang ada dengan mengembangkan sistem rekomendasi pemilihan metode SDLC yang menggabungkan kelebihan metode SAW (kesederhanaan, transparansi, kemudahan implementasi) dengan pendekatan similarity untuk meningkatkan akurasi dan kontekstualitas rekomendasi. Berbeda dengan penelitian sebelumnya, penelitian ini:

- Mengembangkan pendekatan hybrid SAW+Similarity yang spesifik untuk pemilihan metode SDLC
- Melakukan validasi komprehensif melalui tiga pendekatan: perhitungan manual, validasi pakar, dan user acceptance test
- Menyediakan analisis komparatif kuantitatif antara SAW murni dan SAW+Similarity
- Mengimplementasikan sistem dalam bentuk aplikasi web yang dapat langsung digunakan oleh praktisi dan mahasiswa
- Memvalidasi bobot kriteria dan karakteristik metode SDLC melalui expert judgment dari praktisi industri

Dengan demikian, penelitian ini memberikan kontribusi baik secara metodologis (pengembangan pendekatan hybrid) maupun praktis (sistem aplikatif yang telah tervalidasi).

3. METODE PENELITIAN

3.1. Desain Penelitian

Penelitian ini menggunakan metode pengembangan Waterfall karena sistem yang dibangun bersifat deterministik dan berbasis aturan dengan spesifikasi kebutuhan yang jelas. Tahapan penelitian meliputi: (1) studi literatur, (2) identifikasi masalah dan tujuan, (3) pengumpulan data, (4) analisis kebutuhan, (5) perancangan sistem, (6) implementasi, (7) pengujian sistem, dan (8) penyusunan laporan.

3.2. Data Penelitian

Data yang digunakan berupa karakteristik proyek perangkat lunak yang diperoleh melalui studi literatur dan validasi pakar di bidang rekayasa perangkat lunak. Sembilan kriteria penilaian yang digunakan adalah:

Tabel 1. Kriteria Penilaian dan Bobot

Kode	Kriteria	Bobot Awal	Bobot Normalisasi	Jenis
C1	Estimasi Durasi Proyek	0.2000	0.1778	Cost
C2	Tingkat Risiko Proyek	0.0125	0.0111	Cost
C3	Kompleksitas Sistem	0.0500	0.0444	Cost
C4	Anggaran	0.0250	0.0222	Cost
C5	Ukuran Tim	0.2000	0.1778	Benefit
C6	Tingkat Pengalaman Tim	0.0125	0.0111	Benefit
C7	Keterlibatan Pengguna	0.3000	0.2667	Benefit
C8	Perubahan Kebutuhan	0.3000	0.2667	Cost
C9	Kebutuhan Dokumentasi	0.0250	0.0222	Cost

Setiap kriteria memiliki tiga nilai kualitatif (Rendah/Pendek/Kecil, Sedang/Menengah, Tinggi/Panjang/Besar) yang telah divalidasi oleh tiga orang pakar: (1) Muhammad Ridho Nosa, S.T., M.Kom. (Founder Soodu.id dan PT. Indonesia Maxima Teknologi), (2) Eka Budi Saputra, S.Kom. (Praktisi di Pulpas.com), dan (3) Pabenri Dela (Head of Programmer di PT Namasindo Plas).

3.3. Metode Simple Additive Weighting (SAW)

Metode SAW merupakan teknik Multi-Criteria Decision Making (MCDM) yang menggunakan penjumlahan terbobot untuk memberikan rekomendasi [3][4]. Tahapan perhitungan SAW meliputi:

- Normalisasi Matriks Keputusan

Untuk kriteria benefit:

$$R_{ij} = \frac{X_{ij}}{X_{max}}$$

Untuk kriteria cost:

$$R_{ij} = \frac{X_{min}}{X_{ij}}$$

dimana R_{ij} adalah nilai normalisasi alternatif ke- i pada kriteria ke- j , X_{ij} adalah nilai asli, dan X_{max}/X_{min} adalah nilai maksimum/minimum pada kriteria ke- j .

- b. Perhitungan Nilai Preferensi

$$S_i = \sum_{j=1}^n (W_j \times R_{ij})$$

dimana S_i adalah skor akhir metode ke- i , W_j adalah bobot normalisasi kriteria ke- j , dan R_{ij} adalah nilai normalisasi.

3.4. Pendekatan Similarity

Untuk meningkatkan akurasi rekomendasi, penelitian ini menambahkan perhitungan similarity antara profil pengguna dan karakteristik masing-masing metode SDLC:

$$Similarity = 1 - |R_{metode} - R_{user}|$$

$$S_i = \sum_{j=1}^n (W_j \times Similarity_j)$$

Pendekatan ini mempertimbangkan kedekatan karakteristik proyek pengguna dengan profil ideal masing-masing metode SDLC yang telah terdefinisi.

3.5. Implementasi Sistem

Sistem diimplementasikan berbasis web menggunakan:

- Backend: Laravel Framework (PHP)
- Frontend: Blade Template, CSS, Bootstrap
- Database: MySQL
- Development Tool: Visual Studio Code, XAMPP

Sistem menyediakan 14 alternatif metode SDLC: Waterfall, Agile, Scrum, RAD, Prototype, Spiral, Extreme Programming, V-Model, Incremental, Iterative, Big Bang, DevOps, TDD, dan Lean Development.

3.6. Metode Pengujian

Pengujian dilakukan melalui tiga pendekatan:

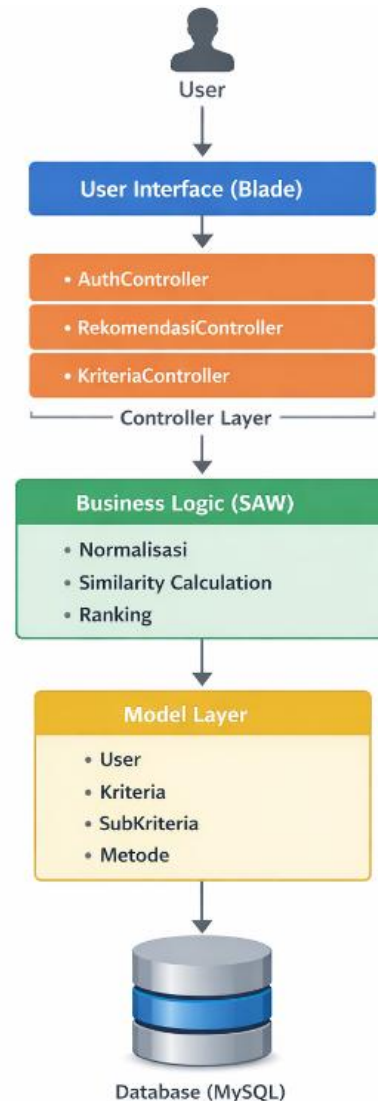
- Pengujian Fungsionalitas: Black Box Testing untuk memastikan semua fitur berfungsi dengan baik
- Pengujian Akurasi: Perbandingan hasil perhitungan sistem dengan perhitungan manual pada 5 kasus berbeda
- Validasi Pakar: Pengujian 10 kasus proyek dengan validasi oleh 3 orang pakar

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Sistem QODELAB (Quality-Oriented Development Life Cycle Advisor & Bench) berhasil diimplementasikan dengan arsitektur Model-View-Controller (MVC) menggunakan Laravel. Sistem terdiri dari dua role pengguna: (1) User yang dapat

melakukan konsultasi pemilihan metode SDLC, dan (2) Admin yang dapat mengelola data kriteria, subkriteria, dan metode SDLC.



Gambar 1. Arsitektur Sistem QODELAB

Fitur utama sistem meliputi:

- Input karakteristik proyek melalui form interaktif dengan 9 pertanyaan
- Proses perhitungan SAW dan similarity secara real-time
- Tampilan hasil rekomendasi dengan ranking 14 metode SDLC
- Export hasil rekomendasi ke format PDF
- Riwayat konsultasi pengguna

4.2. Hasil Pengujian Akurasi Perhitungan

Pengujian akurasi dilakukan dengan membandingkan hasil perhitungan sistem terhadap perhitungan manual pada 5 kasus berbeda dengan karakteristik proyek yang bervariasi.

Tabel 2. Hasil Perbandingan Perhitungan Manual vs Sistem

Kasus	Input Karakteristik	Metode Rekomendasi	Skor Manual	Skor Sistem	Akurasi
1	Pendek, Sedang, Sedang, Sedang, Sedang, Menengah, Sedang, Minor, Tinggi	TDD	0.9296	0.9296	100%
2	Menengah, Rendah, Sedang, Rendah, Sedang, Rendah, Sedang, Tidak Ada, Sedang	Incremental	0.9852	0.9852	100%
3	Panjang, Tinggi, Tinggi, Tinggi, Besar, Tinggi, Tinggi, Total, Tinggi	DevOps	0.8407	0.8407	100%
4	Pendek, Rendah, Rendah, Rendah, Kecil, Rendah, Rendah, Tidak Ada, Rendah	Big Bang	0.9111	0.9111	100%
5	Panjang, Rendah, Rendah, Sedang, Sedang, Menengah, Rendah, Tidak Ada, Tinggi	Waterfall	1.0000	1.0000	100%

Hasil pengujian menunjukkan akurasi 100% dengan kesesuaian penuh antara perhitungan manual dan sistem pada semua tahapan: normalisasi matriks, pembobotan, perhitungan nilai preferensi, dan penentuan ranking. Hal ini memvalidasi bahwa implementasi algoritma SAW telah benar dan konsisten.

4.3. Perbandingan SAW Murni vs SAW + Similarity

Analisis komparatif dilakukan untuk mengukur dampak penambahan metode similarity terhadap kualitas rekomendasi. Pengujian dilakukan dengan input karakteristik yang sama:

Input: Durasi: Menengah, Risiko: Sedang, Kompleksitas: Sedang, Anggaran: Sedang, Ukuran Tim: Sedang, Pengalaman: Menengah, Keterlibatan: Sedang, Perubahan: Minor, Dokumentasi: Sedang

Tabel 3. Perbandingan Hasil SAW Murni vs SAW+Similarity

Metode	Rank SAW	Skor SAW	Rank SAW+Sim	Skor SAW+Sim	Perubahan	Selisih Skor
DevOps	1	84.07%	6	82.59%	↓ -5	-1.48%
Scrum	2	83.71%	7	81.48%	↓ -5	-2.23%
Iterative	3	80.37%	5	86.37%	↑ +2	+5.93%
Agile	4	78.15%	10	75.18%	↓ -6	-2.97%
Prototype	5	76.3%	11	74.07%	↓ -6	-2.23%
Spiral	6	76.29%	3	90.37%	↑ +3	+14.08%
RAD	7	72.59%	13	70.37%	↓ -6	-2.22%
XP	8	72.22%	14	69.26%	↓ -6	-2.96%
TDD	9	67.04%	1	99.63%	↑ +8	+32.59%
Incremental	10	66.67%	2	91.11%	↑ +8	+24.44%
V-Model	11	58.52%	3	90.37%	↑ +8	+31.85%
Waterfall	12	53.7%	12	73.7%	— 0	+20.00%
Lean Dev	13	51.48%	8	75.93%	↑ +5	+24.45%
Big Bang	14	42.22%	9	75.56%	↑ +5	+33.34%



Gambar 2. Grafik Perbandingan SAW Murni vs SAW+Similarity

Hasil menunjukkan perubahan signifikan:

- 7 metode mengalami kenaikan peringkat
- 6 metode mengalami penurunan peringkat
- 1 metode (Waterfall) tetap pada peringkat yang sama

Metode rekomendasi berubah dari DevOps (84.07%) menjadi TDD (99.63%), dengan peningkatan skor sebesar 15.56%. Perubahan ini terjadi karena pendekatan similarity mempertimbangkan kedekatan karakteristik proyek dengan profil ideal masing-masing metode SDLC.

4.4. Analisis Perubahan:

Metode yang mengalami kenaikan peringkat (TDD, V-Model, Incremental, Spiral) memiliki kesamaan dalam:

- Pendekatan iteratif terkontrol
- Penekanan kuat pada testing dan validasi
- Dokumentasi yang terstruktur
- Cocok untuk proyek dengan perubahan kebutuhan moderat

Sebaliknya, metode yang mengalami penurunan (DevOps, Scrum, Agile, XP) memiliki karakteristik:

- Fleksibilitas tinggi dan iterasi sangat cepat
- Dokumentasi minimal
- Lebih cocok untuk perubahan kebutuhan yang sangat tinggi

Dengan karakteristik input yang "serba sedang", sistem dengan similarity lebih akurat dalam mengidentifikasi bahwa metode dengan pendekatan seimbang (seperti TDD) lebih sesuai dibandingkan metode yang extreme (seperti DevOps atau Agile).

4.5. Validasi Pakar

Validasi dilakukan terhadap 10 kasus proyek dengan karakteristik berbeda. Tiga orang pakar memberikan rekomendasi independen yang kemudian dibandingkan dengan hasil sistem.

Tabel 4. Hasil Validasi Pakar

Kasus	Nama Proyek	Rekomendasi Sistem	Pakar 1	Pakar 2	Pakar 3	Kesesuaian
K1	Sistem Informasi Akademik	TDD	Waterfall	TDD	TDD	Sesuai
K2	Aplikasi Mobile Pemesanan	Scrum	Scrum	Scrum	Agile	Sesuai
K3	Sistem Kasir UMKM	Big Bang	Big Bang	Waterfall	Big Bang	Sesuai
K4	Sistem E-Government	Iterative	Scrum	Waterfall	Iterative	Sesuai
K5	Sistem Inventori Gudang	TDD	Spiral	TDD	Spiral	Tidak Sesuai
K6	Website Company Profile	Big Bang	Waterfall	Waterfall	Waterfall	Tidak Sesuai
K7	Aplikasi Startup Digital	Scrum	Agile	Scrum	Scrum	Sesuai
K8	Sistem ERP Perusahaan	Iterative	Spiral	Waterfall	Waterfall	Tidak Sesuai
K9	Aplikasi E-Learning	TDD	Agile	Agile	Agile	Tidak Sesuai
K10	Sistem Monitoring Proyek	TDD	Agile	Waterfall	Agile	Tidak Sesuai

Tingkat Kesesuaian: 5 dari 10 kasus (50%)

Analisis ketidaksesuaian menunjukkan bahwa perbedaan terjadi karena:

- Perbedaan Pendekatan: Sistem menggunakan pendekatan kuantitatif berbasis bobot dan nilai kriteria, sedangkan pakar mempertimbangkan aspek kualitatif seperti:
 - Pengalaman proyek serupa
 - Kebiasaan dan budaya tim
 - Fleksibilitas manajemen organisasi
 - Ketersediaan resource yang tidak terukur numerik
- Interpretasi Kriteria: Pakar cenderung memberikan bobot implisit yang berbeda berdasarkan konteks spesifik organisasi
- Metode Alternatif Masih Relevan: Pada kasus K5-K10 yang tidak sesuai, metode yang direkomendasikan sistem (TDD, Big Bang, Iterative) masih termasuk dalam kategori layak dan berada pada peringkat 3-5 teratas menurut pakar

Meskipun tingkat kesesuaian 50%, hasil ini menunjukkan bahwa sistem dapat berfungsi sebagai decision support system yang memberikan rekomendasi awal terstruktur dan objektif, bukan sebagai pengganti penuh keputusan pakar.

4.6. Pengujian Fungsionalitas Sistem

Pengujian black box dilakukan terhadap 12 modul utama sistem dengan hasil 100% sukses:

Tabel 5. Hasil Pengujian Fungsionalitas

Modul	Skenario Pengujian	Status
Login & Autentikasi	Login valid/invalid, session management	Berhasil
Register	Validasi email, password strength	Berhasil
Karakteristik Input	Form validation, data submission	Berhasil
Perhitungan SAW	Normalisasi, pembobotan, ranking	Berhasil
Rekomendasi Display	Tampilan hasil, export PDF	Berhasil
Histori	Simpan dan tampil riwayat	Berhasil
CRUD Kriteria/Metode	Create, Read, Update, Delete	Berhasil
Role Management	Akses user vs admin	Berhasil

4.7. User Acceptance Test (UAT)

Pengujian UAT dilakukan terhadap 16 responden dengan tingkat pengalaman pengembangan perangkat lunak yang bervariasi (<1 tahun hingga >5 tahun). Penilaian menggunakan skala Likert 1-5 terhadap 19 aspek meliputi aksesibilitas, fungsionalitas, usability, akurasi, dan kepuasan pengguna.

Hasil UAT:

- Rata-rata penilaian: 4.3 / 5.0 (Setuju - Sangat Setuju)
- Aspek tertinggi: Akurasi rekomendasi (4.6/5.0)
- Aspek terendah: Kelengkapan metode SDLC (3.8/5.0)

Masukan pengguna:

- Penambahan variasi metode SDLC
- Peningkatan tampilan antarmuka
- Fitur perbandingan antar metode

Secara keseluruhan, sistem diterima (accepted) dan layak digunakan dalam proses pemilihan metode SDLC.

4.8. Pembahasan

a. Kelebihan Pendekatan SAW + Similarity:

- Peningkatan Akurasi: Skor rekomendasi meningkat 15.56%, menunjukkan kepercayaan sistem yang lebih tinggi
- Kontekstual: Mempertimbangkan kemiripan karakteristik dengan profil metode ideal
- Objektif: Mengurangi ambiguitas dalam pemilihan metode
- Transparan: Proses perhitungan dapat diverifikasi dan dijelaskan

b. Keterbatasan:

- Kesesuaian Pakar 50%: Menunjukkan bahwa faktor kualitatif belum sepenuhnya terakomodasi
- Bobot Statis: Bobot kriteria belum dapat disesuaikan oleh pengguna
- Dataset Terbatas: Profil metode SDLC masih berbasis literatur, belum dari learning data proyek nyata

c. Implikasi Praktis:

Sistem QODELAB dapat digunakan sebagai:

- Alat bantu pembelajaran metode SDLC untuk mahasiswa
- Decision support system untuk pengembang pemula-menengah

- Baseline assessment untuk tim proyek dalam memilih metode pengembangan
- Tools validasi untuk memastikan metode yang dipilih sudah tepat

5. KESIMPULAN DAN SARAN

Penelitian ini telah berhasil mengembangkan sistem rekomendasi pemilihan metode *Software Development Life Cycle* (SDLC) berbasis web yang mengintegrasikan metode *Simple Additive Weighting* (SAW) dengan pendekatan *similarity*. Berdasarkan hasil pengujian, sistem ini terbukti memiliki tingkat akurasi perhitungan sebesar 100% yang divalidasi melalui kesesuaian penuh antara output sistem dengan kalkulasi manual pada seluruh tahapan algoritma. Implementasi pendekatan *hybrid SAW+Similarity* memberikan kontribusi signifikan dengan meningkatkan kualitas rekomendasi sebesar 15.56%, dari angka 84.07% pada penggunaan SAW murni menjadi 99.63%. Meskipun hasil validasi pakar menunjukkan tingkat kesesuaian sebesar 50% atau mencakup 5 dari 10 kasus uji, sistem tetap dinilai sangat layak berfungsi sebagai *decision support system* yang rasional dan terstruktur. Hal ini didukung oleh hasil pengujian fungsionalitas dan *User Acceptance Testing* (UAT) yang memperoleh skor rata-rata 4.3 dari 5.0, mengindikasikan bahwa sistem dapat diterima dengan baik oleh pengguna sebagai instrumen pendukung keputusan yang objektif, sistematis, dan efisien.

Terlepas dari keberhasilan yang dicapai, terdapat beberapa ruang pengembangan yang dapat dilakukan pada penelitian mendatang untuk meningkatkan kapabilitas sistem. Pengembangan tersebut dapat diarahkan pada implementasi mekanisme pembobotan dinamis yang memungkinkan pengguna menyesuaikan prioritas kriteria secara personal. Selain itu, integrasi teknologi *machine learning* diperlukan untuk mempelajari pola-pola dari dataset proyek nyata, yang dipadukan dengan penambahan variasi metode SDLC serta kriteria penilaian yang lebih luas. Untuk meningkatkan aspek pengalaman pengguna, pengembangan fitur visualisasi interaktif dan perbandingan metode secara *side-by-side* sangat disarankan. Terakhir, transformasi sistem ke dalam platform aplikasi *mobile* dapat dipertimbangkan guna meningkatkan aksesibilitas bagi pengembang maupun mahasiswa di berbagai kondisi penggunaan.

DAFTAR PUSTAKA

- [1] M. I. Hossain, "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management," *Int. J. Multidiscip. Res.*, vol. 5, no. 5, hal. 1–36, 2023, doi: 10.36948/ijfmr.2023.v05i05.6223.
- [2] H. N. Widya Ningsih, "Perbandingan Model Waterfall Dan Metode Prototype Untuk Pengembangan Aplikasi Pada Sistem Informasi," *J. Inf. Syst. Educ. Dev.*, vol. 2, no. 1, hal. 13–19, 2024, doi: 10.62386/jised.v2i1.50.
- [3] F. Siva, S. M. U. Assegaf, S. A. Pahlevi, dan M. A. Yaqin, "Survei Metode-Metode Software Development Life Cycle dengan Metode Systematic Literature Review," *Ilk. J. Comput. Sci. Appl. Informatics*, vol. 5, no. 2, hal. 36–52, 2023, doi: 10.28926/ilkonnika.v5i2.447.
- [4] A. B. Paksi, N. Hafidhoh, dan S. K. Bimonugroho, "Perbandingan Model Pengembangan Perangkat Lunak Untuk Proyek Tugas Akhir Program Vokasi," *J. Masy. Inform.*, vol. 14, no. 1, hal. 70–79, 2023, doi: 10.14710/jmasif.14.1.52752.
- [5] M. M. Seniv, "Method for Selecting a Software Development Methodology Taking Into Account Project Characteristics," *Radio Electron. Comput. Sci. Control*, no. 2, hal. 134, 2023, doi: 10.15588/1607-3274-2023-2-14.
- [6] P. P. Lunak, Y. A. Permana, I. P. Nugroho, P. T. Maulana, dan T. F. Ajiji, "Penerapan Simple Additive Weighting dalam Merekomendasikan Karyawan untuk Pekerjaan," vol. 6, no. 1, hal. 468–477, 2025.
- [7] P. Metode, T. Dan, T. Wisata, dan D. I. Manggarai, "PERBANDINGAN METODE TOPSIS DAN SIMPLE ADDITIVE WEIGHTING UNTUK PEMILIHAN TEMPAT WISATA DI MANGGARAI TIMUR," hal. 224–233, 2021.
- [8] N. I. S. Baldanullah, F. Adhania, dan D. Fitriati, "Comparison of SAW, WP, and TOPSIS Methods in Determining the Best Journalists," *J. Ris. Inform.*, vol. 5, no. 1, hal. 129–138, 2022, doi: 10.34288/jri.v5i4.199.
- [9] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, vol. 1, no. October, 2020.
- [10] E. R. Rahmi, E. Yumami, dan N. Hidayasari, "Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review," *Remik*, vol. 7, no. 1, hal. 821–834, 2023, doi: 10.33395/remik.v7i1.12177.
- [11] I. Alam, N. Sarwar, dan I. Noreen, "Statistical analysis of software development models by six-pointed star framework," *PLoS One*, vol. 17, no. 4 April, hal. 1–17, 2022, doi: 10.1371/journal.pone.0264420.
- [12] M. Sameen Mirza dan S. Datta, "Strengths and Weakness of Traditional and Agile Processes - A Systematic Review," *J. Softw.*, vol. 14, no. 5, hal. 209–219, 2019, doi: 10.17706/jsw.14.5.209-219.
- [13] S. Pargaonkar, "A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering," *Int. J. Sci. Res. Publ.*, vol. 13, no. 8, hal. 120–124, 2023, doi: 10.29322/ijsrp.13.08.2023.p14015.
- [14] L. Laguna Salvadó, E. Villeneuve, D. Masson, A. Abi Akle, dan N. Bur, "Decision Support System

- for technology selection based on multi-criteria ranking: Application to NZEB refurbishment,” *Build. Environ.*, vol. 212, hal. 108786, 2022, doi: 10.1016/j.buildenv.2022.108786.
- [15] C. Pramatha, I. P. I. S. Jayadi, dan I. D. M. B. Atmaja, “Pengembangan Sistem Rekomendasi Melalui Pendekatan Web Semantik dan Simple Additive Weighting (SAW),” *SINTECH (Science Inf. Technol. J.*, vol. 5, no. 2, hal. 139–153, 2022, doi: 10.31598/sintechjournal.v5i2.1216.
- [16] R. D. Estrada-Esponda, M. López-Benítez, G. Maturro, dan J. C. Osorio-Gómez, “Selection of software agile practices using Analytic hierarchy process,” *Heliyon*, vol. 10, no. 1, 2024, doi: 10.1016/j.heliyon.2023.e22948.