

Integrasi *Large Language Models* dalam Evaluasi dan Optimalisasi Arsitektur Perangkat Lunak: Studi Kasus Berbasis ISO/IEC/IEEE 42010

Yosi Briyan Saputro^{1*}, Moch. Darul Gusti Alief², Muhammad Ainul Yaqin³

^{1,2,3}Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim. Jl. Gajayana No. 50 Malang, Jawa Timur, Indonesia.

E-mail: 230605110055@student.uin-malang.ac.id

* Corresponding Author



<https://doi.org/>

ARTICLE INFO

Article history

Received: 24 January 2024

Revised: 30 January 2024

Accepted: 5 February 2024

Kata Kunci

Arsitektur Perangkat Lunak, ISO/IEC/IEEE 42010, Large Language Models, Evaluasi Arsitektur, GPT, Gemini.

Keywords

Software Architecture, ISO/IEC/IEEE 42010, Large Language Models, Architecture Evaluation, GPT, Gemini.

ABSTRACT

Arsitektur perangkat lunak sangat penting untuk mengembangkan sistem yang dapat dipelihara dan skalabel. Namun, masalah seperti keterbatasan dokumentasi dan kompleksitas desain sering kali menghambat penilaian arsitektural yang efektif. Tujuan dari penelitian ini adalah untuk mempelajari peran model bahasa besar (LLM) dalam menganalisis dokumentasi arsitektur perangkat lunak yang didasarkan pada standar ISO/IEC/IEEE 42010; penelitian ini menggunakan metode studi literatur dan eksperimen evaluasi otomatis terhadap sistem multi-lapisan, di mana model GPT berfungsi sebagai evaluator dan Gemini berfungsi sebagai validator. Hasil menunjukkan bahwa LLM dapat menemukan kesesuaian arsitektur dengan standar, menemukan masalah yang mungkin terjadi, dan memberikan saran optimalisasi berdasarkan praktik terbaik. Terlepas dari kenyataan bahwa validasi manual diperlukan untuk memastikan keakuratan evaluasi LLM, integrasi LLM menawarkan peluang yang signifikan untuk mempercepat proses analisis arsitektural berbasis data.

Software architecture is essential for developing maintainable and scalable systems. However, issues such as limited documentation and design complexity often hinder effective architectural assessment. The purpose of this study is to study the role of large language models (LLMs) in analyzing software architecture documentation based on the ISO/IEC/IEEE 42010 standard; this study uses literature study methods and automated evaluation experiments on multi-layer systems, where the GPT model serves as an evaluator and Gemini serves as a validator. The results show that LLMs can find architectural conformance to standards, find potential issues, and provide optimization suggestions based on best practices. Despite the fact that manual validation is required to ensure the accuracy of LLM evaluation, the integration of LLMs offers significant opportunities to accelerate the process of data-driven architectural analysis.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

PENDAHULUAN

Perkembangan sistem perangkat lunak modern yang semakin kompleks menuntut pendekatan desain arsitektural yang skalabel, dapat dipelihara, dan terdokumentasi dengan baik agar mudah dievaluasi. Banyak organisasi menghadapi masalah untuk memastikan bahwa arsitektur yang dibuat sesuai dengan prinsip dan standar yang berlaku. ISO/IEC/IEEE 42010 adalah standar internasional yang umum digunakan untuk menjelaskan arsitektur perangkat lunak (ISO/IEC/IEEE 15288, 2011). Standar ini menekankan pentingnya menentukan *stakeholder*, *concern*, *viewpoints*, *views*, *model*, dan *rationale* dalam dokumentasi arsitektural (ISO, 2015).

Sebaliknya, kemajuan teknologi dalam bidang kecerdasan buatan, terutama Large Language Models (LLM) seperti GPT dan Gemini, telah membuka kesempatan baru untuk membantu dalam proses evaluasi dan optimasi arsitektur perangkat lunak. LLM dengan kemampuan untuk memahami dan memproses bahasa alami dalam skala besar memungkinkan mereka untuk mengevaluasi

dokumentasi arsitektur secara otomatis, menemukan kesalahan, dan menyarankan solusi yang paling efektif (Schmid et al., 2025).

Tujuan dari penelitian ini adalah untuk mengevaluasi seberapa efektif LLM dapat digunakan sebagai alat bantu dalam proses evaluasi arsitektural, khususnya dengan mempertimbangkan kesesuaian dengan standar ISO/IEC/IEEE 42010. Selain itu, penelitian ini juga melibatkan proses validasi silang antara hasil evaluasi yang dihasilkan oleh GPT dengan Gemini (Kampelopoulos et al., 2025). Dalam langkah ini, hasil yang dihasilkan oleh LLM dinilai untuk keakuratan dan kegunaannya.

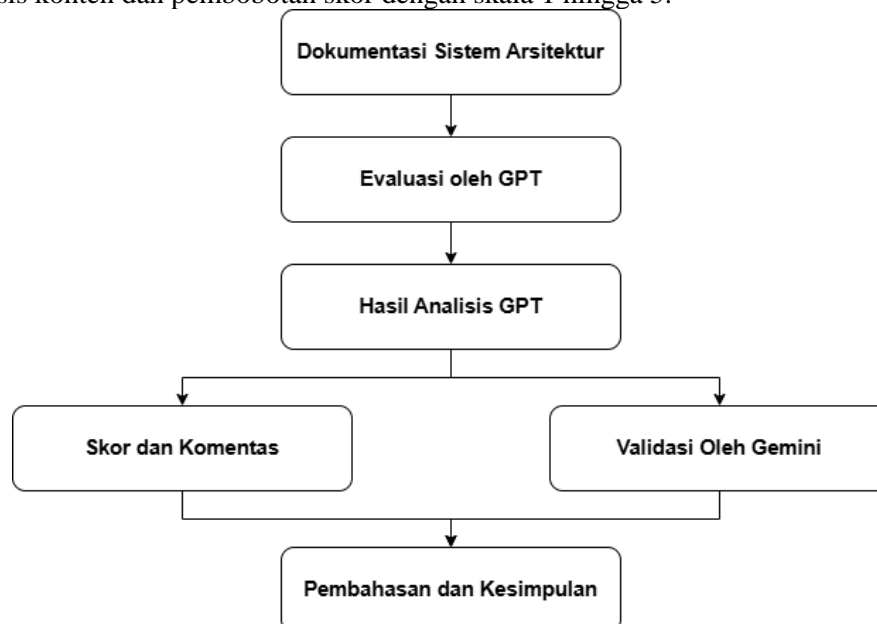
Penelitian ini penting karena menawarkan solusi baru untuk proses evaluasi arsitektur yang selama ini bergantung pada analisis manual yang memakan waktu dan tenaga (Muhammad et al., 2017). Diharapkan bahwa dengan memasukkan LLM ke dalam proses ini, penilaian kualitas arsitektur perangkat lunak akan lebih cepat dan lebih akurat.

METODE

Pendekatan eksploratif-kualitatif digunakan dalam penelitian ini, yang melibatkan studi literatur dan eksperimen sistem. Berdasarkan standar ISO/IEC/IEEE 42010, fokus utamanya adalah mengevaluasi fungsi Model Bahasa Besar (LLM) dalam proses analisis arsitektur perangkat lunak. Sebuah sistem perangkat lunak multi-lapisan yang dideskripsikan secara modular ke dalam tujuh lapisan arsitektur adalah subjek penelitian ini (Esposito et al., 2025).

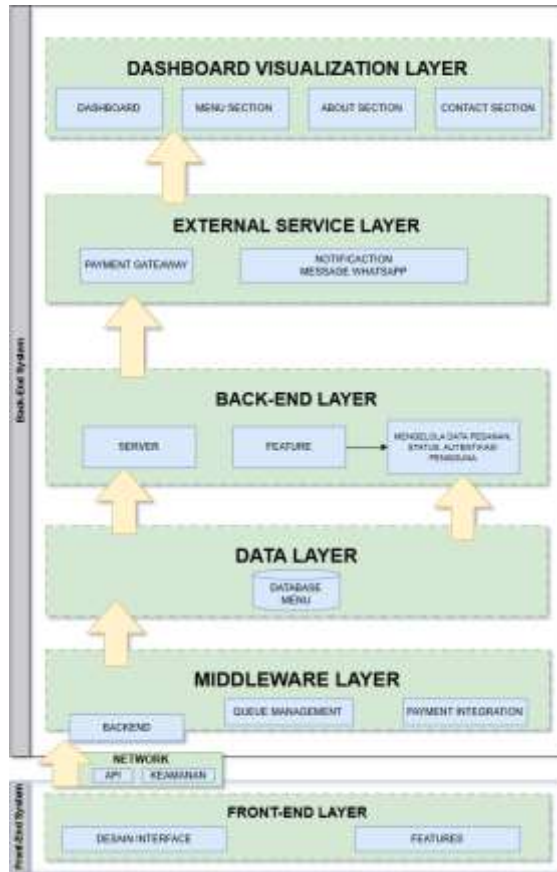
Pertama, dokumentasi arsitektur sistem dibuat berdasarkan elemen-elemen ISO 42010, seperti identifikasi pihak berwenang, masalah, pandangan, dan alasan. Setelah dokumentasi selesai, model GPT digunakan untuk mengevaluasi arsitektur berdasarkan tiga aspek utama: (1) kesesuaian dengan standar ISO/IEC/IEEE 42010, (2) identifikasi masalah potensial yang berkaitan dengan arsitektur, dan (3) saran untuk optimalisasi sistem.

Selanjutnya, hasil evaluasi GPT divalidasi dengan menggunakan Gemini sebagai LLM pembanding. Gemini memberikan umpan balik terhadap hasil analisis GPT dan memberikan penilaian berdasarkan skor efektivitas, konsistensi, dan akurasi analisis. Validasi dilakukan dengan menggunakan metode analisis konten dan pembobotan skor dengan skala 1 hingga 5.



Gambar 1. Diagram Alur Eksperimen Evaluasi LLM

Selain itu, untuk mendukung analisis, digunakan daftar checklist yang didasarkan pada ISO/IEC/IEEE 42010 serta template evaluasi yang didapat dari pustaka arsitektur sistem. Tiga laporan utama dibuat dari hasil pengujian: evaluasi kesesuaian dokumentasi, penemuan masalah arsitektural, dan saran berbasis praktik terbaik.



Gambar 2. Skema 7 Lapisan Arsitektur Sistem

HASIL DAN PEMBAHASAN

Studi ini menemukan tiga kategori evaluasi utama GPT: (1) evaluasi dokumentasi untuk standar ISO/IEC/IEEE 42010, (2) identifikasi masalah potensial arsitektural, dan (3) saran optimisasi. Gemini memvalidasi setiap evaluasi untuk menilai akurasi dan efektivitas hasil analisis GPT.

Hasil Evaluasi Dokumentasi Arsitektur

Menurut GPT, dokumentasi sudah sesuai standar. "Arsitektur terdiri dari tujuh lapisan yang modular dan terdokumentasi dengan baik," menurut model. Namun, Gemini menunjukkan bahwa GPT tidak dapat menemukan komponen yang hilang, seperti alasan keputusan desain dan metadata versi sistem. Gemini menemukan bahwa GPT menyimpulkan kesesuaian terlalu cepat tanpa melakukan pemetaan elemen ISO 42010.

Hasil Deteksi Masalah Arsitektural

GPT menyatakan bahwa sistem tidak mengalami masalah arsitektural yang signifikan, tetapi Gemini menilai bahwa GPT gagal menggunakan pendekatan sistematis untuk mengidentifikasi masalah seperti ketergantungan berlebihan atau komunikasi yang tidak efisien antara komponen. Selain itu, Gemini menyatakan bahwa tidak ada eksplorasi risiko atau evaluasi bottleneck.

Hasil Rekomendasi Optimalisasi

Dalam bagian ini, GPT menunjukkan performa terbaik. Model memberikan berbagai rekomendasi teknis yang relevan dan praktis, seperti:

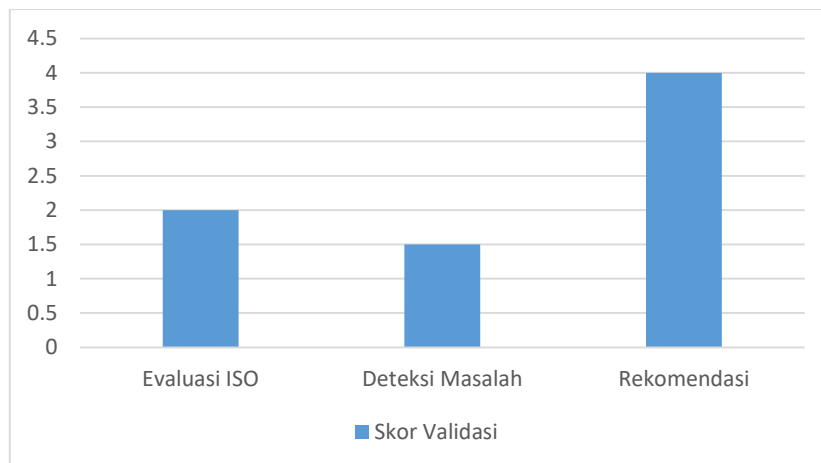
1. Implementasi Dependency Injection untuk mengurangi tight coupling
2. Penggunaan Message Queue untuk mendukung skala dan isolasi modul
3. Pendekatan Microservices untuk manajemen layanan independen
4. Penguatan keamanan dengan OAuth dan pendekatan Zero Trust Architecture

Tabel 1. Ringkasan Evaluasi & Validasi

Aspek Evaluasi	Penilaian GPT	Validasi Gemini	Catatan
Evaluasi terhadap standar ISO	Sesuai	Tidak konsisten	Tidak semua elemen ISO diidentifikasi eksplisit
Deteksi masalah struktural	Tidak ada	Tidak terverifikasi	Tidak menggunakan pendekatan teknis objektif
Rekomendasi optimisasi	Kuat	Relevan dan tepat	Perlu dikaitkan langsung dengan struktur sistem

Tabel 2. Komparasi Tanggapan GPT dan Validasi oleh Gemini

Aspek Evaluasi	Tanggapan GPT	Validasi Gemini
Dokumentasi Arsitektur	Dokumentasi sistem sudah memenuhi aspek viewpoints, stakeholders, dan concerns.	Tidak disebutkan konten views & rationale eksplisit. Belum memenuhi ISO 42010 sepenuhnya.
Deteksi Masalah	Tidak ditemukan pelanggaran prinsip arsitektur seperti tight coupling.	Tidak ada bukti konkrit atau metode sistematis untuk evaluasi struktur arsitektur.
Rekomendasi	Gunakan microservices, OAuth, Zero Trust Architecture, dan Dependency Injection.	Relevan dan praktis, tetapi perlu dikaitkan langsung dengan konteks sistem yang dievaluasi.



Gambar 3. Grafik Skor Validasi GPT oleh Gemini

Pembahasan

Penelitian menunjukkan bahwa LLM seperti GPT dapat membantu proses evaluasi arsitektur perangkat lunak, terutama dalam memberikan rekomendasi teknis berbasis praktik terbaik. Namun, GPT masih tidak memiliki kemampuan untuk mengevaluasi kesesuaian dokumentasi terhadap standar ISO/IEC/IEEE 42010 dan mengidentifikasi inefisiensi arsitektural. Validasi Gemini secara konsisten mengkritik kecenderungan GPT untuk memberikan kesimpulan positif tanpa analisis sistematis atau justifikasi berbasis data.

Dibandingkan dengan dua komponen lainnya, kinerja GPT lebih menonjol dalam rekomendasi optimisasi. Ada banyak artikel, dokumentasi, dan diskusi komunitas tentang praktik pengembangan perangkat lunak yang membentuk model yang dapat dijelaskan. Oleh karena itu, GPT memiliki kemampuan untuk mencantumkan berbagai pola solusi yang populer, seperti microservices, broker pesan, Zero Trust, dan ADR (Architectural Decision Records). Namun demikian, rekomendasi seringkali tidak terkait langsung dengan konteks sistem yang sedang dievaluasi, yang dapat menyebabkan overgeneralisasi.

Pada bagian deteksi masalah, GPT masih menunjukkan kelemahan dalam membedakan antara masalah yang tidak ditemukan dan tidak dapat ditemukan. Misalnya, GPT menyatakan bahwa "tidak ditemukan pelanggaran prinsip desain" dalam Uji 2, tetapi tidak memberikan dasar argumen seperti analisis dependensi, struktur modular, atau validasi pola komunikasi.

Karena tidak didukung oleh visualisasi arsitektural atau penilaian berbasis aturan, pernyataan tersebut dianggap sebagai asertif oleh Gemini.

1. Selain itu, hasil validasi Gemini menunjukkan bahwa tingkat keakuratan GPT berbeda-beda tergantung pada kedalaman diagnosis yang diperlukan:
2. GPT cukup kompeten dalam hal-hal yang bersifat standar dan encyklopedis seperti definisi teknologi atau prinsip desain.

Hasil ini mendukung gagasan bahwa LLM dapat berfungsi sebagai co-pilot dalam proses pengembangan perangkat lunak. Namun, LLM tidak dapat menggantikan critical thinking manusia secara keseluruhan. Evaluasi arsitektur membutuhkan pemahaman tentang konteks sistem, batasan domain, dan pertimbangan non-fungsional seperti keamanan, skalabilitas, dan maintainability, yang tidak selalu dijelaskan dalam dokumentasi tekstual.

Dari perspektif metodologis, metode eksperimen yang menggunakan kombinasi GPT sebagai evaluator awal dan Gemini sebagai validator akhir terbukti memberikan perspektif dua lapis: diagnosis awal dan penguatan atau refleksi kritis. Pendekatan ini dapat dikembangkan menjadi pipeline evaluasi semi-otomatis di mana GPT berfungsi sebagai asisten pengumpul temuan awal dan Gemini atau manusia berfungsi sebagai reviewer kritis.

Implikasi praktis dari pembahasan ini adalah:

1. Untuk drafting rekomendasi teknis atau refleksi awal desain sistem, GPT sangat cocok.
2. Namun, temuan harus divalidasi dengan standar formal seperti ISO/IEC/IEEE 42010 dan praktik teknis yang relevan dengan situasi proyek.
3. Ke depan, integrasi LLM dalam toolchain arsitektur perangkat lunak harus mempertimbangkan kombinasi pengawasan manusia dan otomasi.

Oleh karena itu, meskipun akurasi diagnosis dan pemetaan terhadap standar masih menjadi area eksplorasi yang perlu dikembangkan secara bertahap, penelitian ini menunjukkan bahwa kekuatan LLM ada pada kemampuan generalisasi dan saran berbasis corpus.

SIMPULAN

Studi ini menunjukkan bahwa Large Language Models (LLM) seperti GPT dan Gemini dapat memainkan peran penting dalam proses evaluasi dan optimasi arsitektur perangkat lunak, khususnya dalam kaitannya dengan kesesuaian dengan standar ISO/IEC/IEEE 42010. Berdasarkan tiga aspek evaluasi—kesesuaian dokumentasi, deteksi masalah arsitektural, dan rekomendasi optimisasi—ditunjukkan bahwa GPT memiliki kemampuan terbaik untuk memberikan rekomendasi praktis dan relevan yang berkaitan dengan arsitektur perangkat lunak.

Validasi model Gemini menunjukkan bahwa pengawasan manusia atau sistem penyeimbang sangat penting dalam interpretasi hasil analisis. Berkolaborasi dengan GPT sebagai analisis awal dan Gemini sebagai validator memberikan pendekatan yang lebih cepat namun tetap akurat.

Menurut temuan penelitian ini, metode pelatihan LLM yang lebih khusus pada arsitektur perangkat lunak harus dikembangkan dan diintegrasikan dengan alat otomatis seperti pengecek model atau penganalisis metrik arsitektur. Selain itu, dokumentasi arsitektur yang lebih terorganisir diperlukan agar hasil evaluasi LLM menjadi lebih relevan dan akurat untuk diterapkan di dunia kerja

UCAPAN TERIMA KASIH

Peneliti menyampaikan ucapan terima kasih kepada pihak yang sudah berkontribusi dalam pelaksanaan penelitian dan penyusunan artikel ini.

REFERENSI

Bucaioni, A., Weyssow, M., He, J., Lyu, Y., & Lo, D. (n.d.). *A Functional Software Reference Architecture for LLM-Integrated Systems*. 20200234.

- Bucaioni, A., Weyssow, M., He, J., Lyu, Y., & Lo, D. (2025). Artificial Intelligence for Software Architecture: Literature Review and the Road Ahead. *Proceedings of The ACM International Conference on the Foundations of Software Engineering (FSE '25)*, 1(1), 1–11. <http://arxiv.org/abs/2504.04334>
- Budi, C. S., & Bachtiar, A. M. (2018). Implementasi Arsitektur Microservices pada Backend Comrades. *Program Studi Teknik Informatika, Universitas Komputer Indonesia*, 1–6. <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-cahyantose-40046>
- Dzikria, I., & Maharani, N. A. (2024). Analisis Kebutuhan Arsitektur dan Desain Antarmuka Sistem Manajemen Risiko Berbasis Penelitian House of Risk pada Institusi Pendidikan. *Jurnal Teknik Informatika Dan Komputer*, 3, 31–39. <https://journal.uhamka.ac.id/index.php/jutikom>
- Effendi, A., & Yantri Komala Dewi. (2023). Analisis Bibliometrik Perancangan Arsitektur Dengan Kecerdasan Buatan. *SARGA: Journal of Architecture and Urbanism*, 17(1), 48–63. <https://doi.org/10.56444/sarga.v17i1.219>
- Esposito, M., Li, X., Moreschini, S., Ahmad, N., Cerny, T., Vaidhyanathan, K., Lenarduzzi, V., & Taibi, D. (2025). *Generative AI for Software Architecture. Applications, Trends, Challenges, and Future Directions*. <http://arxiv.org/abs/2503.13310>
- Ferdinand, J., Syahrina, A., & Musnansyah, A. (2022). PERANCANGAN ARSITEKTUR PERANGKAT LUNAK MICROSERVICES PADA APLIKASI OPEN LIBRARY UNIVERSITAS TELKOM MENGGUNAKAN gRPC. *Telkatika*, 1(2), 71–77.
- Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2023). *Large Language Models for Software Engineering: A Systematic Literature Review*. X(December), 1–79. <http://arxiv.org/abs/2308.10620>
- ISO/IEC/IEEE15288. (2011). INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering. *Internacional Standard ISO/IEC/IEEE 42010, First edit*, 104.
- ISO. (2015). *INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — iTeh STANDARD PREVIEW iTeh STANDARD PREVIEW. 2015*.
- Kampelopoulos, D., Tsanousa, A., Vrochidis, S., & Kompatsiaris, I. (2025). A review of LLMs and their applications in the architecture, engineering and construction industry. *Artificial Intelligence Review*, 58(8). <https://doi.org/10.1007/s10462-025-11241-7>
- Mohammad Harry Khomas Saputra, L. M. N. (2021). Penerapan Arsitektur Microservice Pada Sistem Tata Kelola Matakuliah Proyek Politeknik Pos Indonesia. *Pendidikan Kimia PPs UNM*, 13(3), 22–28.
- Muhammad, R., Ibnu Asror, S.T., M. T., & Indra Lukmana Sardi, S.T., M. . (2017). Desain dan Analisis Arsitektur Microservices Pada Sistem Informasi Akademik Perguruan Tinggi Dengan Pendekatan Architecture Tradeoff Analysis Method (ATAM) (Studi Kasus: iGracias Universitas Telkom). *E-Proceeding of Engineering*, Vol.4(2355–9365). <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/9192/9058>
- Round-robin, M., & Martin, R. (n.d.). *ISO / IEC / IEEE 42010 Architecture description*.
- Schmid, L., Hey, T., Armbruster, M., Corallo, S., Fuchß, D., Keim, J., Liu, H., & Kozirolek, A. (2025). *Software Architecture Meets LLMs: A Systematic Literature Review*. 1–15. <http://arxiv.org/abs/2505.16697>
- Van Heesch, U., Avgeriou, P., & Hilliard, R. (2012). A documentation framework for architecture decisions. *Journal of Systems and Software*, 85(4), 795–820. <https://doi.org/10.1016/j.jss.2011.10.017>