

## Diskusi.1

✓ Selesai: Lihat

Lakukan: Kirim balasan: 1

Jatuh tempo: Minggu, 19 Oktober 2025, 23:59

menampilkan balasan dalam bentuk bertingkat

Setelan v

### Diskusi.1

Rabu, 28 Mei 2025, 10:16

Selamat datang di sesi diskusi studi kasus pertama pada mata kuliah Rekayasa Perangkat Lunak! Mari kita simak sebuah situasi nyata yang menguji kreativitas dan kemampuan problem solving Anda.

#### Studi Kasus: Tantangan Dokumentasi pada Hasil Proyek Vendor

PT. ABC mengontrak sebuah vendor untuk mengembangkan sebuah program penting yang mendukung operasional perusahaan. Setelah proyek selesai, vendor menyerahkan source code program kepada PT. ABC, namun tanpa disertai dokumentasi teknis yang memadai.

Setahun kemudian, seiring dengan perubahan kebutuhan pasar, PT. ABC memutuskan untuk melakukan modifikasi pada salah satu fitur program. Namun, tim internal mengalami kendala besar karena:

- Tidak adanya dokumentasi yang menjelaskan struktur, logika, dan arsitektur program.
- Kesulitan memahami dan mengidentifikasi bagian-bagian kode yang perlu diubah.

#### Pertanyaan Diskusi:

1. Menurut Anda, apa saja dampak negatif yang mungkin timbul akibat tidak adanya dokumentasi dari vendor?
2. Bagaimana hal ini dapat mempengaruhi operasional dan biaya perusahaan dalam jangka panjang?
3. Solusi apa yang anda sarankan untuk situasi seperti ini ?

Silakan bagikan ide dan solusi inovatif kalian untuk mengatasi masalah ini. Diskusi ini diharapkan tidak hanya memberikan jawaban atas permasalahan saat ini, tetapi juga membuka wawasan tentang best practices dalam pengelolaan proyek perangkat lunak. Selamat berdiskusi dan berkreasi!

*Catatan: Saya sarankan, teman-teman mahasiswa menjawab langsung pada tempat yang disediakan, TIDAK mengupload jawaban berupa file, termasuk TIDAK mengupload jawaban dengan GAMBAR ya. Terima kasih!*

**Re: Diskusi.1**oleh [052471106 MUHAMAD FUJI SUBEKTI](#) - Senin, 6 Oktober 2025, 10:14**1. Dampak negatif yang mungkin timbul akibat tidak adanya dokumentasi dari vendor****a. Kebingungan dan Ketidakpastian:**

Tanpa dokumentasi yang jelas, tim internal PT. ABC akan kesulitan memahami struktur dan logika program. Hal ini dapat menyebabkan kebingungan dalam proses modifikasi, yang berpotensi mengarah pada kesalahan pengkodean.

**b. Waktu dan Biaya Tambahan:**

Ketika tim harus menghabiskan waktu untuk memahami kode yang tidak terdokumentasi, ini akan meningkatkan waktu pengerjaan proyek. Akibatnya, biaya yang dikeluarkan untuk melakukan modifikasi bisa jauh lebih tinggi daripada yang direncanakan.

**c. Risiko Kualitas:**

Ketidakjelasan dalam kode dapat menyebabkan implementasi perubahan yang tidak sesuai dengan harapan, berpotensi merusak fitur lain atau mengurangi kualitas keseluruhan aplikasi.

**d. Kesulitan dalam Pemeliharaan:**

Tanpa dokumentasi, pemeliharaan jangka panjang menjadi sulit. Tim baru yang bergabung tidak akan memiliki panduan untuk memahami dan meneruskan pekerjaan yang telah dilakukan sebelumnya.

**2. Bagaimana hal ini dapat mempengaruhi operasional dan biaya perusahaan dalam jangka panjang**

**Operasional:** Ketidakmampuan untuk melakukan modifikasi yang diperlukan dengan cepat dapat menghambat respons perusahaan terhadap perubahan pasar. Hal ini dapat mempengaruhi daya saing PT. ABC.

**Biaya:** Biaya tambahan akan muncul akibat waktu yang terbuang, kemungkinan gaji karyawan yang lebih tinggi untuk melakukan pekerjaan yang lebih kompleks, dan potensi kehilangan pendapatan akibat keterlambatan dalam meluncurkan fitur baru.

**3. Solusi untuk situasi seperti tersebut****a. Membangun Dokumentasi yang Memadai:**

Segara setelah proyek berakhir, PT. ABC harus meminta vendor untuk menyusun dokumentasi yang jelas tentang struktur, logika, dan arsitektur program. Ini termasuk diagram alir, penjelasan kode, dan panduan penggunaan.

**b. Implementasi Best Practices dalam Pengelolaan Proyek:**

Di masa depan, PT. ABC perlu menetapkan standar dokumentasi sebagai bagian dari kontrak dengan vendor. Hal ini harus mencakup persyaratan untuk dokumentasi teknis yang lengkap dan penggunaan alat dokumentasi yang baik.

**c. Pelatihan Internal:**

Mengadakan pelatihan bagi tim internal tentang cara membaca dan memahami kode, serta teknik dokumentasi yang baik, akan memperkuat kemampuan tim dalam menangani situasi serupa di masa mendatang.

**d. Menggunakan Alat Otomatisasi:**

Pertimbangkan untuk menggunakan alat otomatisasi yang dapat membantu menghasilkan dokumentasi dari kode secara otomatis, sehingga mengurangi beban manual dan memastikan dokumentasi selalu terbaru.

**e. Membangun Kultur Dokumentasi:**

Mendorong budaya dokumentasi di dalam tim, di mana setiap anggota merasa bertanggung jawab untuk mendokumentasikan pekerjaan mereka, akan sangat membantu dalam menciptakan basis pengetahuan yang kuat.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:10

Terima kasih atas analisisnya. Jawaban Anda sudah menunjukkan pemahaman yang baik mengenai pentingnya dokumentasi vendor jangan lupa tambahkan referensi biar jawaban terlihat ilmiah

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [054416248 PANDU WIJAYA](#) - Senin, 6 Oktober 2025, 21:33

##### 1. Dampak Negatif Akibat Tidak Adanya Dokumentasi Teknis dari Vendor

Beberapa dampak negatif yang dapat timbul:

- Sulit Memelihara dan Mengembangkan Aplikasi: Tanpa dokumentasi, tim internal kesulitan memahami alur program, struktur kode, dan dependensi sistem.
- Risiko Bug dan Error Saat Modifikasi: Perubahan kode tanpa pemahaman menyeluruh bisa memunculkan bug baru atau merusak fungsi lain.
- Ketergantungan Pada Vendor: Jika hanya vendor yang memahami sistem, maka perusahaan sangat tergantung pada mereka, bahkan setelah kontrak selesai.
- Produktivitas Tim Menurun: Tim perlu waktu ekstra untuk memahami kode dari awal, yang bisa memperlambat delivery.
- Kesulitan Saat Audit atau Sertifikasi: Tidak adanya dokumentasi menyulitkan saat dilakukan audit TI atau sertifikasi standar ISO/ITIL.

##### 2. Dampak Terhadap Operasional dan Biaya Perusahaan dalam Jangka Panjang

- Biaya Tambahan untuk Reverse Engineering: Perusahaan harus mengalokasikan sumber daya untuk membaca dan memahami kode secara manual, yang memakan waktu dan biaya.
- Waktu Pengembangan yang Lebih Lama: Update fitur atau debugging menjadi lambat, yang bisa mengganggu operasional.
- Risiko Keamanan: Tanpa dokumentasi, sulit mengidentifikasi potensi celah keamanan.
- Kehilangan Pengetahuan (Knowledge Loss): Jika tim internal yang memahami sebagian kode keluar, maka pengetahuan bisa hilang sepenuhnya.

##### 3. Solusi dan Best Practices yang Disarankan

###### A. Solusi untuk Situasi Saat Ini

- Audit dan Re-documentation: Lakukan audit source code oleh tim internal atau pihak ketiga untuk membuat dokumentasi teknis baru (manual reverse engineering).
- Code Refactoring Bertahap: Sambil memahami sistem, refactor bagian-bagian kode agar lebih mudah dibaca dan dikelola.
- Libatkan Eks-Vendor Jika Perlu: Jika masih memungkinkan, minta vendor lama untuk membuat dokumentasi, meskipun dengan biaya tambahan.

###### B. Best Practices untuk Ke Depan

1. Kontrak Harus Mewajibkan Dokumentasi: Pastikan dalam kontrak proyek, dokumentasi teknis lengkap adalah deliverable wajib.
2. Code Review Berkala: Lakukan review rutin selama pengembangan untuk memastikan dokumentasi berkembang bersama kode.
3. Gunakan Tools Otomatisasi Dokumentasi:  
Seperti Doxygen, Sphinx, atau Swagger untuk dokumentasi API.
4. Training dan Knowledge Transfer: Jadwalkan sesi knowledge transfer di akhir proyek, termasuk pelatihan untuk tim internal.
5. Source Code Management yang Baik: Gunakan version control seperti Git + dokumentasi commit yang jelas.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:11

Pendapat yang baik! Dokumentasi sangat penting untuk menjaga kontinuitas kerja dan efisiensi biaya.  
tambahkan referensi biar jawaban terlihat ilmiah

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [053883097 SYAHRUL HASANUDIN](#) - Senin, 6 Oktober 2025, 22:35

Assalamulaikum..

Berikut tanggapan saya tentang pertanyaan diskusi pada tuton sesi pertama.

1. Berikut ini ada beberapa poin dampak negatif yang timbul jika tidak adanya dokumentasi teknis yang memadai:

##### 1) Penurunan Kualitas dan Kestabilan Sistem

- Kesalahan modifikasi : Modifikasi yang dilakukan pada satu bagian dapat secara tidak sengaja merusak fungsi di bagian lain yang tidak terkait.
- Pelatihan yang mahal : Setiap karyawan baru yang bergabung dengan tim pemeliharaan akan memerlukan kurva pembelajaran yang sangat curam, membuang waktu senioritas untuk menjelaskan kode, bukan untuk mengembangkan fitur baru.

##### 2) Ketergantungan pada Vendor

- Kehilangan kendali : PT. ABC kehilangan kemampuan untuk memelihara atau mengembangkan sistem secara mandiri.
- Risiko yang berlanjut : Jika vendor tersebut bangkrut atau tim pengembang intinya berganti, tidak ada orang lain yang memiliki pengetahuan mendalam tentang kode tersebut. Program tersebut menjadi terbengkalai yang berisiko tidak dapat dipertahankan.

##### 3) Peningkatan Waktu dan Biaya Pemeliharaan

- Peningkatan biaya modifikasi : Waktu yang seharusnya digunakan untuk coding malah terbuang untuk proses Reverse Engineering (menganalisis kode yang sudah ada untuk memahami fungsinya), yang merupakan proses yang mahal dan rentan kesalahan.
- Waktu troubleshooting lebih lama : Tanpa penjelasan logika, tim internal harus menghabiskan waktu berjam-jam atau berhari-hari hanya untuk melacak bagaimana suatu fungsi bekerja.

2. Pengaruh jangka panjang pada operasional dan biayanya pada kasus tersebut, yaitu:

## 1) Sumber Daya Internal

- Pengaruh operasional : Moral tim TI turun karena merasa tidak berdaya dan fokus teralihkan dari pengembangan strategis ke pemadam kebakaran (firefighting).
- Pengaruh biaya jangka panjang : Biaya tenaga kerja (gaji tim internal) menjadi tidak efisien karena mereka menghabiskan sebagian besar waktu untuk *deciphering* kode.

## 2) Agilitas Bisnis

- Pengaruh operasional : Kemampuan adaptasi terhadap perubahan pasar menjadi sangat lambat.
- Pengaruh biaya jangka panjang : Kehilangan peluang pasar akibat ketidakmampuan sistem merespons cepat.

## 3) Total Biaya Kepemilikan (TCO)

- Pengaruh operasional : TCO sistem menjadi tinggi karena biaya pemeliharaan (*maintenance*) dan support melonjak.
- Pengaruh biaya jangka panjang : Setiap upgrade ke teknologi baru menjadi jauh lebih mahal dan berisiko.

3. Berikut solusi - solusi yang dapat mencegah terjadinya kasus di atas untuk tahun ke depannya lagi:

1) Memanfaatkan Code Analysis Tools : Menggunakan tool otomatis (seperti static code analyzer) yang dapat membantu memetakan dependensi, struktur kelas, dan diagram alir kode.

2) Dokumentasi Minimal : Mengalokasikan sumber daya internal (atau menyewa technical writer eksternal) untuk membuat dokumentasi minimal berdasarkan reverse engineering yang dilakukan saat ini. Fokus pada bagian yang akan dimodifikasi (untuk memecahkan masalah mendesak) dan arsitektur level tinggi (untuk orientasi tim).

3) Wajib dokumentasi : Kontrak vendor harus secara eksplisit mencantumkan serah terima dokumentasi yang komprehensif sebagai syarat mutlak penerimaan dan pembayaran terakhir.

4) Integrasi dokumentasi dalam proses pengembangan : Dokumentasi tidak boleh menjadi pekerjaan di akhir proyek. PT. ABC harus mewajibkan vendor untuk menyimpan dokumentasi dalam format yang sama dengan kode (e.g., markdown files di repositori GIT) dan menjadikannya bagian dari setiap commit (Documentation as Code).

5) Penggunaan Self-Documenting Code : Standar coding yang ketat, termasuk penamaan variabel dan fungsi yang jelas (self-documenting code) dan penggunaan comments yang memadai di dalam source code itu sendiri. Meskipun bukan pengganti dokumentasi eksternal, ini sangat meringankan beban reverse engineering.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.1**

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:12

Bagus! Anda mampu mengaitkan antara dokumentasi dan efisiensi operasional secara tepat.tambahkan referensi biar jawaban terliht ilmiah

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.1**

oleh [TRIANA PUTRI WAHYUNI 053839459](#) - Senin, 6 Oktober 2025, 23:42

Apabila sebuah proyek perangkat lunak diserahkan tanpa dokumentasi, dampaknya cukup merugikan bagi perusahaan. Tim internal PT. ABC akan mengalami kesulitan memahami struktur dan logika program, karena tidak ada penjelasan yang menjelaskan cara kerja setiap bagian kode. Kondisi ini membuat proses perbaikan atau pengembangan fitur baru menjadi lebih lambat dan berisiko menimbulkan kesalahan yang tidak diinginkan. Dalam jangka panjang, hal tersebut dapat menghambat operasional perusahaan karena sistem menjadi sulit untuk dikembangkan sesuai kebutuhan yang terus berubah. Selain itu, perusahaan mungkin perlu mengeluarkan biaya tambahan untuk mempelajari ulang kode program atau bahkan melibatkan pihak luar untuk membantu melakukan dokumentasi ulang. Tentu saja hal ini tidak efisien dan dapat mengganggu produktivitas tim.

Sebagai langkah perbaikan, sebaiknya perusahaan menetapkan kebijakan agar setiap vendor wajib menyerahkan dokumentasi teknis yang lengkap bersamaan dengan hasil proyek. Untuk situasi yang sudah terjadi, PT. ABC dapat mulai membuat dokumentasi baru dengan menelusuri kembali kode yang ada, misalnya melalui proses reverse engineering. Perusahaan juga dapat membentuk tim khusus yang bertugas memperbarui dokumentasi setiap kali terjadi perubahan sistem. Dengan demikian, pengelolaan perangkat lunak di masa mendatang akan lebih terarah, efisien, dan berkelanjutan.

#### Referensi

Putra, A. R., & Hidayat, A. (2020). Analisis Pengaruh Dokumentasi terhadap Keberhasilan Pengembangan Sistem Informasi. Jurnal Informatika dan Rekayasa Perangkat Lunak.

Dicoding Indonesia. (2023). Pentingnya Dokumentasi dalam Pengembangan Software.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:13

Penjelasan yang jelas dan logis. Anda telah menguraikan dampak jangka panjang dengan baik.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [054518659 DHAFFA NIZHAR ADINDHA PUTRA](#) - Selasa, 7 Oktober 2025, 01:09

1. dampak negatif yang mungkin timbul akibat tidak adanya dokumentasi dari vendor
  - a. Tantangan Pemeliharaan. Tim internal mungkin kesulitan memahami alur logika dan struktur sistem jika tidak ada dokumentasi. Ini dapat menyebabkan kesalahan saat melakukan perubahan.
  - b. Perubahan yang dilakukan tanpa dokumentasi dapat menyebabkan sistem menjadi lambat, tidak efisien, atau bug baru yang sulit dilacak.
  - c. tidak adanya dokumentasi dapat membuat menurunkan produktivitas tim karena membutuhkan waktu yang lebih dan biaya lebih untuk memahami kode dari awal.
2. Bagaimana hal ini dapat mempengaruhi operasional dan biaya perusahaan dalam jangka panjang
  - a. tidak adanya dokumentasi dapat mengganggu operasi dan biaya perusahaan dalam jangka Panjang. Karena jumlah Waktu dan sumber daya yang dibutuhkan untuk memahami kode jauh menjadi lebih besar, biaya pemeliharaan akan meningkat.
  - b. ada peningkatan kemungkinan downtime, karena kesalahan kecil dalam modifikasi dapat menyebabkan sistem gagal berfungsi dan mengganggu aktivitas operasional
  - c. perbaikan atau debugging karyawan yang bergantung pada sistem mengalami penurunan produktivitas. karena tim baru yang tergabung akan memilih untuk mengembangkan kode daripada mempelajari kode lama, hal tersebut akan membuat keterlambatan adanya fitur baru
  - d. organisasi dapat mengeluarkan biaya tambahan untuk audit atau reverse engineering dalam membangun sistem baru dari awal jika sistem tidak dapat diubah
3. Solusi untuk situasi yang dialami PT ABC

PT. ABC harus mewajibkan setiap vendor yang bekerja sama harus melampirkan dokumentasi dengan hasil kerja proyek yang sudah direncanakan. Setelah apa yang terjadi dengan PT. ABC, di masa akan datang harus membuat kontrak untuk dokumentasi wajib dari vendor yang bekerja sama, dengan kriteria dokumentasi yang lengkap setiap pengerjaan dan hasil yang baik

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:14

Analisis yang kuat. Dokumentasi vendor memang menjadi dasar bagi pengawasan dan evaluasi kerja sama bisnis. tambahkan referensi biar jawaban terlihat ilmiah

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AHMAD MINAN NAJIB 050480425](#) - Selasa, 7 Oktober 2025, 10:32

Assalamualikum wr wb

Izin menjawab pertanyaan diskusi 1

Saya akan menyampaikan pemahaman saya mengenai studi kasus PT. ABC ini, dengan fokus pada analisis masalah dan solusi praktis. Sebagai penggemar rekayasa perangkat lunak, saya melihat ini sebagai pelajaran berharga tentang pentingnya dokumentasi dalam siklus hidup software. Langsung saja saya akan menjawab pertanyaan diatas sebagai berikut:

##### 1. Dampak Negatif yang Mungkin Timbul Akibat Tidak Adanya Dokumentasi dari Vendor

Kurangnya dokumentasi teknis seperti yang dialami PT. ABC bisa menimbulkan berbagai dampak negatif, baik secara teknis maupun organisasional. Seperti:

- Kesulitan Pemahaman dan Maintenance Kode: Tanpa dokumentasi yang menjelaskan arsitektur, struktur modul, atau logika bisnis, tim internal harus "reverse engineering" kode secara manual. Ini memakan waktu lama dan rawan kesalahan, seperti salah mengidentifikasi dependensi antar modul, yang bisa menyebabkan bug baru atau kerusakan fitur existing.
- Penurunan Produktivitas Tim: Developer internal menghabiskan waktu berbulan-bulan hanya untuk memetakan kode, alih-alih fokus pada pengembangan fitur baru. Hal ini bisa menimbulkan frustrasi, turnover karyawan yang tinggi, dan hilangnya pengetahuan institusional jika tim berganti.
- Risiko Keamanan dan Kepatuhan: Dokumentasi yang absen membuat sulit untuk mengaudit kode terhadap vulnerability (misalnya, SQL injection atau data leaks). Jika program menangani data sensitif, ini bisa melanggar regulasi seperti GDPR atau PDPA di Indonesia, berpotensi menimbulkan tuntutan hukum.

- Ketergantungan Berlebih pada Vendor: PT. ABC mungkin terpaksa kembali ke vendor untuk bantuan, yang bisa menimbulkan konflik kontrak atau biaya tambahan yang tidak direncanakan.

##### 2. Pengaruh pada Operasional dan Biaya Perusahaan dalam Jangka Panjang

Dalam jangka panjang, masalah ini bisa merusak fondasi operasional PT. ABC dan meningkatkan biaya secara eksponensial:

- Gangguan Operasional: Modifikasi fitur yang tertunda bisa membuat perusahaan kehilangan peluang pasar, seperti gagal merespons perubahan kebutuhan pelanggan. Misalnya, jika program ini mendukung e-commerce, delay bisa berarti penurunan revenue hingga 20-30% per kuartal, berdasarkan studi kasus serupa dari Gartner.
- Peningkatan Biaya Maintenance: Tanpa dokumentasi, biaya perawatan software bisa melonjak 2-5 kali lipat. Menurut IEEE, proyek tanpa dokumentasi baik memerlukan 40% lebih banyak usaha untuk modifikasi. PT. ABC mungkin harus merekrut konsultan eksternal atau bahkan membangun ulang bagian kode, yang bisa menelan biaya jutaan rupiah.
- Dampak Strategis Jangka Panjang: Ketergantungan pada kode "hitam" (black box) menghambat inovasi dan



skalabilitas. Perusahaan bisa kesulitan mengintegrasikan sistem baru (seperti AI atau cloud migration), yang pada akhirnya menurunkan daya saing. Secara finansial, ini bisa menambah opportunity cost, di mana dana yang seharusnya untuk ekspansi malah terkuras untuk "pembersihan" warisan kode.

### 3. Solusi yang Disarankan untuk Situasi Seperti Ini

Untuk mengatasi masalah saat ini di PT. ABC dan mencegahnya di masa depan, ada beberapa pendekatan multi-layer yang menggabungkan solusi jangka pendek dan best practices jangka panjang. Berikut solusinya:

- Solusi Jangka Pendek (Remediasi Segera):

- Lakukan Code Review dan Dokumentasi Internal: Bentuk tim khusus (misalnya, 2-3 developer senior) untuk melakukan static code analysis menggunakan tools seperti SonarQube atau Understand. Hasilnya, buat dokumentasi dasar: diagram arsitektur (dengan UML via tools seperti Lucidchart), API docs (menggunakan Swagger jika ada endpoint), dan komentar inline pada kode kritis. Estimasi waktu: 1-3 bulan, tergantung ukuran proyek.

- Gunakan AI-Assisted Tools untuk Reverse Engineering: Inovasi modern seperti GitHub Copilot atau Sourcegraph bisa membantu menganalisis kode secara otomatis, menghasilkan pseudocode atau diagram flow. Ini bisa mempercepat proses hingga 50% dibanding manual.

- Solusi Jangka Menengah (Modifikasi dan Transfer Pengetahuan):

- Refactor Kode Secara Bertahap: Identifikasi fitur yang perlu diubah dan refactor hanya bagian itu, sambil menambahkan unit tests (dengan framework seperti JUnit atau pytest) untuk memastikan stabilitas. Libatkan vendor lama jika kontrak memungkinkan, tapi dengan klausul penalti untuk dokumentasi di masa depan.

- Training dan Onboarding Tim: Adakan workshop internal untuk mentransfer pengetahuan, termasuk sesi "code walkthrough" mingguan. Ini membangun ownership tim dan mengurangi ketergantungan eksternal.

- Best Practices Jangka Panjang (Pencegahan di Proyek Mendatang):

- Integrasikan Dokumentasi dalam Kontrak Vendor: Selalu sertakan klausul wajib seperti "Deliverable Documentation Standards" yang mencakup SRS (Software Requirements Specification), design docs, dan user manuals. Gunakan template dari IEEE 829 untuk standarisasi. Tambahkan milestone pembayaran yang bergantung pada kualitas dokumentasi (misalnya, 20% pembayaran final setelah review docs).

- Adopsi Agile dengan Dokumentasi Berkelanjutan: Pindah ke metodologi seperti DevOps atau Agile, di mana dokumentasi dibuat iteratively (e.g., via Confluence atau Notion). Tools seperti Doxygen atau Sphinx bisa auto-generate docs dari komentar kode.

- Bangun Knowledge Base Internal: Implementasikan sistem seperti wiki perusahaan atau repository Git dengan branch khusus untuk docs. Inovasi: Gunakan low-code platforms seperti OutSystems untuk prototipe cepat yang self-documenting, mengurangi risiko di proyek vendor selanjutnya.

- Audit Rutin dan Vendor Scoring: Lakukan pre-qualification vendor berdasarkan track record dokumentasi, dan lakukan post-project audit untuk evaluasi.

Dengan solusi ini, PT. ABC tidak hanya menyelesaikan masalah saat ini tapi juga membangun ekosistem software yang lebih resilient. Ini sejalan dengan prinsip SOLID dan Clean Code dari Robert C. Martin, di mana dokumentasi adalah kunci untuk maintainability.

Referensi

Modul MSIM4303 modul 1

Software engineering : a practitioner's approach. Pressman,R.S.

Rekayasa perangkat lunak terstruktur dan berorientasi objek. Sukamto

Sekian dan terimakasih.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:14

Penjelasan yang baik, menggambarkan pemahaman mendalam terhadap konsep manajemen vendor

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



**Re: Diskusi.1**

oleh [MUHAMMAD RIZQI PRATAMA 053580633](#) - Selasa, 7 Oktober 2025, 11:42

1. Dampak yang timbulkan jika tidak adanya dokumentasi dari vendor, PT ABC akan kebingungan dalam memodifikasi dan memahami struktur dan logika program. Sehingga PT ABC kesulitan dalam memelihara dan mengembangkan sesuai dengan kebutuhan pasar. Resiko Bug sistem lebih besar dan membutuhkan biaya, waktu lebih, dimana Tim IT pada PT ABC harus bekerja lebih ekstra karena mereka perlu memahami struktur dan logika program.
2. Ketergantungan PT ABC dengan vendor, dalam hal ini PT ABC kesulitan dalam mengembangkan program secara mandiri, mereka akan mengeluarkan biaya dan waktu yang lebih karena harus bergantung dengan vendor. Jika PT ABC ingin melakukan perbaikan dan pengembangan secara mandiri, mereka membutuhkan biaya dan waktu yang lebih kepada Tim IT karena kesulitan memahami struktur dan logika program. Hal tersebut mengakibatkan PT ABC lambat dalam penyesuaian terhadap permintaan Pasar, karena terjadinya hambatan operasional internal.
3. Solusi untuk PT ABC, segera bentuk Tim IT khusus untuk memahami struktur, logika, dan arsitektur program, lakukan Dokumentasi Program yang menjelaskan secara terperinci. Evaluasi kontrak kerjasama terhadap vendor, PT ABC harus lebih teliti dan harus wajib melampirkan dokumentasi program untuk pengembangan dan perbaikan di masa yang akan datang.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.1**

oleh [AJAY SUPRIADI 04001670](#) - Selasa, 7 Oktober 2025, 16:16

Sangat baik! Anda berhasil menjelaskan hubungan antara dokumentasi dan pengendalian biaya perusahaan. tambahkan referensi biar jawaban terlihat ilmiah

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.1**

oleh [HASAN RAPI 053782613](#) - Selasa, 7 Oktober 2025, 17:16

Assalamualaikum izin menanggapi diskusinya bapak/ibu, tutor..

### 1. Dampak Negatif Tidak adanya Dokumentasi

Tidak adanya dokumentasi dalam suatu proyek dapat menimbulkan berbagai masalah serius yang mana diantaranya:

- Tim Internal akan kesulitan memahami arsitektur, logika bisnis dan alur kerja kode program yang dibuat oleh vendor. akibatnya proses maintenance dan pengembangan lanjutan akan lebih sulit dan berisiko tinggi (pressman & maxim, 2015)
- perusahaan menjadi sangat tergantung pada vendor awal karena hanya mereka yang memahami sistem tersebut,
- sulit mendapat acuan apabila terjadi perubahan sistem pada proyek tersebut

### 2. Dampak Terhadap Operasional dan Biaya Jangka Panjang

- Dalam jangka panjang, ketiadaan dokumentasi menyebabkan biaya operasional meningkat karena tim harus meluangkan waktu ekstra untuk memahami sistem sebelum melakukan perubahan. Hal ini menurunkan produktivitas dan meningkatkan risiko gangguan operasional (Sommerville, 2016).
- Menurut Rosa dan Shalahuddin (2018), dokumentasi yang baik menjadi landasan utama agar sistem dapat dikembangkan kembali secara efisien dan berkelanjutan. Tanpa dokumentasi, perusahaan juga kehilangan kelincahan dalam menyesuaikan diri terhadap perubahan kebutuhan bisnis.

### 3. Solusi dan Rekomendasi

Beberapa langkah yang dapat diambil antara lain:

- Melakukan reverse engineering terhadap kode program untuk menyusun dokumentasi baru
- melaksanakan kode review agar tim internal memahami struktur dan fungsi kode.
- menggunakan alat bantu otomatis seperti doxygen atau javadoc untuk menghasilkan dokumentasi dari kode.
- menetapkan standar dokumentasi wajib
- membangun budaya dokumentasi di lingkungan kerja serta menerapkan pendekatan dev opsional agar dokumentasi selalu diperbarui seiring pengembangan sistem

### Kesimpulan

Dokumentasi adalah aset penting dalam pengelolaan proyek, tanpanya perusahaan akan menghadapi kesulitan teknis, ketergantungan kepada vendor, serta pemborosan biaya di masa depan, oleh karena itu budaya dan kebijakan dokumentasi wajib diterapkan sebagai bagian dari praktik terbaik (best practice).

### Referensi

- Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
- Sutabri, T. (2012). *Analisis Sistem Informasi*. Yogyakarta: Andi.
- Rosa, A. S., & Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Rabu, 8 Oktober 2025, 05:16

Jawaban Anda menunjukkan pemahaman yang matang tentang pentingnya dokumentasi vendor dalam menjaga kesinambungan sistem

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [053929596 FEIZA RAHMAH ASSYIFA](#) - Selasa, 7 Oktober 2025, 18:39

Nama: Feiza Rahmah Assyifa

NIM: 053929596

Prodi: Sistem Informasi

Selamat malam Ajay Supriadi, S.Kom., M.Kom. Izinkan saya memberikan tanggapan atas soal pada diskusi sesi 1 ini.

PT. ABC mengontrak sebuah vendor untuk mengembangkan sebuah program penting yang mendukung operasional perusahaan. Setelah proyek selesai, vendor menyerahkan source code program kepada PT. ABC, namun tanpa disertai dokumentasi teknis yang memadai.

Setahun kemudian, seiring dengan perubahan kebutuhan pasar, PT. ABC memutuskan untuk melakukan modifikasi pada salah satu fitur program. Namun, tim internal mengalami kendala besar karena:

- Tidak adanya dokumentasi yang menjelaskan struktur, logika, dan arsitektur program.
- Kesulitan memahami dan mengidentifikasi bagian-bagian kode yang perlu diubah.

Pertanyaan Diskusi:

Dalam kasus ini, perusahaan PT. ABC bekerja sama dengan vendor untuk mengembangkan sebuah program penting. Setelah proyek rampung, vendor hanya menyerahkan source code tanpa disertai dokumentasi teknis. Ketika setahun kemudian perusahaan ingin melakukan perubahan pada fitur program, tim internal mengalami kesulitan karena tidak tersedia informasi mengenai struktur, logika, dan arsitektur sistem, sehingga bagian kode yang perlu dimodifikasi pun sulit dipahami.

1. Menurut Anda, apa saja dampak negatif yang mungkin timbul akibat tidak adanya dokumentasi dari vendor?

- Risiko ketergantungan (Vendor Lock-in):

PT. ABC menjadi terikat secara total pada vendor awal untuk setiap kebutuhan pemeliharaan (maintenance), perbaikan (bug fixing), atau pengembangan fitur baru.

Hal ini menempatkan PT. ABC pada posisi tawar yang lemah, di mana vendor dapat menentukan biaya dan jangka waktu yang tidak efisien

- Peningkatan technical debt dan kualitas kode menurun:

Tim internal atau vendor baru harus melakukan analisis balik (reverse-engineering) kode untuk memahami logika dan arsitektur, sebuah proses yang memakan waktu dan biaya besar.

Modifikasi yang dilakukan tanpa pemahaman menyeluruh berpotensi menimbulkan bug baru pada komponen yang tidak terkait (efek domino), sehingga meningkatkan beban teknis (technical debt) program.

- Hambatan transfer pengetahuan (Knowledge Transfer):

Proses alih pengetahuan ke tim internal atau pihak ketiga menjadi mustahil dilakukan secara efisien, menghambat pengembangan kompetensi internal perusahaan.

2. Bagaimana hal ini dapat mempengaruhi operasional dan biaya perusahaan dalam jangka panjang?

Melonjaknya biaya kepemilikan (Total Cost of Ownership/TCO):

- Biaya yang dikeluarkan untuk pemeliharaan dan modifikasi melampaui anggaran normal karena harus mencakup biaya waktu yang digunakan untuk dekonstruksi kode (code deciphering).
- Dalam jangka waktu 3-5 tahun, biaya ini dapat mencapai nilai yang jauh lebih tinggi daripada biaya pengembangan program awal.

Hambatan Agilitas Bisnis (Business Agility):

- Kemampuan perusahaan untuk beradaptasi terhadap perubahan regulasi atau dinamika pasar menjadi sangat lambat. Waktu yang dibutuhkan untuk modifikasi fitur krusial (seperti kasus pada PT. ABC) menjadi tidak kompetitif.
- Ini berpotensi mengakibatkan hilangnya peluang pendapatan atau penalti kepatuhan (compliance).

Ancaman Re-platforming:

- Jika kompleksitas kode tanpa dokumentasi menjadi tidak terkendali, perusahaan mungkin terpaksa mengambil keputusan ekstrem untuk mengembangkan ulang program dari awal (re-platforming), yang merupakan kerugian modal yang signifikan.

3. Solusi apa yang anda sarankan untuk situasi seperti ini ?

PT. ABC perlu menerapkan solusi terstruktur untuk mengatasi kendala saat ini dan mencegah terulang di masa depan:

A. Tindakan jangka pendek

1. Audit dan Reverse-Engineering Kode:

- Mengontrak konsultan atau tim pengembang senior yang kompeten untuk melakukan audit kode secara menyeluruh, memetakan arsitektur, dan menghasilkan dokumentasi minimum yang esensial (khususnya untuk bagian yang akan dimodifikasi).

2. Sesi Transfer Pengetahuan Terstruktur:

- Jika memungkinkan, negosiasikan dengan vendor awal untuk sesi knowledge transfer intensif, dengan kompensasi yang jelas. Hasil sesi harus didokumentasikan dalam format standar.

## B. Tindakan angka panjang

## 1. Penetapan Klausul Kontrak Wajib:

- Setiap perjanjian kerja sama (MoU) dengan vendor harus secara eksplisit mencantumkan Dokumentasi Teknis sebagai deliverable utama dan syarat mutlak penerimaan proyek.
- Dokumentasi yang diminta harus spesifik (misalnya, Data Model Diagram, API Specification, System Architecture Diagram).

## 2. Penerapan Mekanisme Retensi Pembayaran:

- Tahan porsi tertentu dari pembayaran akhir (Retensi Proyek) hingga dokumentasi telah diverifikasi dan disetujui oleh tim teknis internal PT. ABC.

## 3. Standardisasi Internal:

- Mewajibkan standar code commenting dan penggunaan alat otomatisasi (tools) untuk menghasilkan dokumentasi dari kode sumber (inline documentation), memastikan proses dokumentasi terintegrasi dalam siklus pengembangan (SDLC).

## Sumber referensi:

- Badr Interactive. (2025). 5 Risiko Proyek Software Gagal dan Cara Mencegahnya di Tahap Awal. Diakses dari <https://badr.co.id/project-management/5-risiko-proyek-software-gagal-dan-cara-mencegahnya-di-tahap-awal/>
- Pollux Integra. (2025). 10 Kasus Kegagalan Proyek Pengembangan Software: Pembelajaran Berharga dari Kesalahan Masa Lalu. Diakses dari <https://polluxintegra.co.id/kasus-kegagalan-proyek-pengembangan-software/>
- Kompasiana. (2025). Pentingnya Dokumentasi dalam Rekayasa Perangkat Lunak: Menjaga Kejelasan dan Konsistensi Proyek. Diakses dari <https://www.kompasiana.com/iqbalharish4725/68599cfa34777c6b25245202>
- UTI-TTIS. (2025). Pentingnya Dokumentasi dalam Rekayasa Perangkat Lunak untuk Proyek Sukses. Diakses dari <https://csirt.teknokrat.ac.id/pentingnya-dokumentasi-dalam-rekayasa-perangkat-lunak-untuk-proyek-sukses/>
- Wicaksono, S. R. (2021). Dokumentasi Perangkat Lunak. Malang: CV. Seribu Bintang. e-ISBN: 978-623-8396-04-7. Diakses dari [https://www.academia.edu/123658861/Dokumentasi\\_Perangkat\\_Lunak](https://www.academia.edu/123658861/Dokumentasi_Perangkat_Lunak)

Tautan permanen Tampilkan induk Balas

**Re: Diskusi.1**oleh [AJAY SUPRIADI 04001670](#) - Rabu, 8 Oktober 2025, 05:17

Jawaban Anda menggambarkan pemahaman yang kuat tentang hubungan antara dokumentasi dan efisiensi biaya

Tautan permanen Tampilkan induk Balas

**Re: Diskusi.1**oleh [RISKI SEPTIADI MURTI WAHYU 052218499](#) - Selasa, 7 Oktober 2025, 22:37

## 1. Dampak Negatif Tidak Adanya Dokumentasi Dari Vendor

Ketika dokumentasi tidak tersedia, ada beberapa dampak negatif yang bisa terjadi, yaitu:

## 1) Kesulitan Dalam Pemeliharaan Dan Perubahan:

Tim internal kesulitan memahami struktur kode, cara kerja bisnis, serta hubungan antar bagian sistem. Hal ini membuat setiap perubahan sulit dilakukan dan berisiko menyebabkan masalah baru.

## 2) Bergantung Pada Vendor Awal:

Perusahaan menjadi tergantung pada vendor yang membangun sistem, karena hanya mereka yang paham bagaimana kode itu bekerja. Ini mengurangi kemampuan perusahaan untuk mengelola sistem secara mandiri.

## 3) Waktu Dan Biaya Pengembangan Ulang Meningkat:

Tanpa dokumentasi, tim harus menganalisis ulang sistem untuk memahami cara kerjanya. Proses ini membutuhkan waktu dan biaya yang lebih besar.

#### 4) Risiko Terhadap Kinerja Sistem:

Jika perubahan dilakukan tanpa pemahaman yang cukup, bisa mengganggu sistem utama dan berdampak langsung pada operasional perusahaan.

### 2. Dampak terhadap Operasional dan Biaya Perusahaan dalam Jangka Panjang

#### 1) Penurunan Efisiensi Operasional:

Perusahaan kesulitan untuk cepat menyesuaikan diri dengan perubahan kebutuhan pasar karena proses mengubah sistem menjadi lebih lambat. Hal ini dapat menyebabkan kehilangan peluang bisnis yang ada.

#### 2) Peningkatan Biaya Pemeliharaan:

Setiap kali ingin memperbaiki sistem atau menambahkan fitur baru, perusahaan harus menghabiskan banyak uang untuk menganalisis kode, atau bahkan harus menghubungi vendor lama kembali.

#### 3) Risiko Teknologi Usang:

Karena sistem sulit diubah, maka sistem tersebut bisa tidak kompatibel dengan teknologi terbaru. Akhirnya, perusahaan mungkin harus membangun sistem dari awal.

#### 4) Menurunnya Keandalan dan Keamanan Sistem:

Tanpa ada dokumentasi yang lengkap, sulit memastikan bahwa standar keamanan, pengendalian akses, dan integritas data tetap terjaga ketika sistem mengalami perubahan.

### 3. Solusi dan Cara Terbaik yang Direkomendasikan

Untuk menghadapi situasi seperti ini dan mencegah terulang di masa depan, beberapa solusi yang inovatif dan bisa diterapkan adalah sebagai berikut:

#### 1) Solusi Jangka Pendek

##### a. Melakukan Reverse Engineering:

Gunakan alat bantu untuk menganalisis struktur program, diagram alur, serta ketergantungan modul. Hasil dari analisis ini bisa menjadi dasar pembuatan dokumentasi yang baru.

##### b. Membentuk Tim Dokumentasi Internal:

Tentukan tim khusus untuk membuat dokumen teknis berdasarkan hasil pengamatan terhadap kode dan wawancara dengan vendor (jika masih bisa dilakukan).

#### 2) Solusi Jangka Panjang (Praktik Terbaik dalam RPL)

##### a. Menentukan Standar Dokumentasi Proyek:

Dalam setiap kontrak kerja dengan vendor, harus diatur bahwa hasil proyek wajib disertai dengan dokumentasi lengkap, seperti desain sistem, manual pengguna, diagram arsitektur, dan lainnya.

##### b. Menggunakan Manajemen Konfigurasi Perangkat Lunak (SCM):

Semua komponen program, seperti kode sumber, dokumen, spesifikasi, dan laporan perubahan, harus dikelola dengan sistem manajemen versi yang terpusat seperti Git. Hal ini agar setiap perubahan dapat dicatat dan di dokumentasikan dengan baik.

##### c. Menerapkan Prinsip Kemudahan Pemeliharaan dalam Desain Sistem:

Pastikan perangkat lunak dikembangkan dengan prinsip modularitas, penggunaan nama yang jelas, serta komentar dalam kode yang informatif.

d. Membuat Knowledge Base Internal:

Setiap proyek yang selesai harus disimpan bersama dokumennya dalam repositori pengetahuan internal perusahaan, sehingga bisa diakses oleh tim di masa depan.

Sumber:

1. BMP STSI4202 Rekasaya Perangkat Lunak.,
2. [https://repository.unkris.ac.id/id/eprint/1978/2/Compressed-Final%20Layout%20Naskah%20Pengantar%20E-Business\\_Revisi.pdf](https://repository.unkris.ac.id/id/eprint/1978/2/Compressed-Final%20Layout%20Naskah%20Pengantar%20E-Business_Revisi.pdf).,
3. Kenneth C. Laudon, Carol Guercio Traver. E-Commerce 2021: Business, Technology, Society. Pearson. 2020.,
4. [https://www.suryainternusa.com/files/L7myrxf7gj89y3femrlb/2018.05.15-AR%202017-Surya\\_Semesta\\_Internusa\\_Final.pdf](https://www.suryainternusa.com/files/L7myrxf7gj89y3femrlb/2018.05.15-AR%202017-Surya_Semesta_Internusa_Final.pdf).

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Rabu, 8 Oktober 2025, 05:18

Sangat baik! Anda sudah mampu mengaitkan dampak operasional dengan risiko jangka panjang secara runtut

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [RATIH NIKEN PRATIWI RAMADHANI 053714635](#) - Rabu, 8 Oktober 2025, 01:37

Assalamualaikum.

Izinkan saya menjawab diskusi 1 ini,jika ada kesalahan mohon dikoreksi.

Terima Kasih

 [rekayasa perangkat lunak diskusi 1.docx](#)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Rabu, 8 Oktober 2025, 05:19

jawaban diskusi langsung di kerjakan di bagian komentar jangan di microsoft word, kalo terus berlanjut nilainya saya pastikan dibawah 75

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Rabu, 8 Oktober 2025, 05:20

Catatan: Saya sarankan, teman-teman mahasiswa menjawab langsung pada tempat yang disediakan, TIDAK mengupload jawaban berupa file, termasuk TIDAK mengupload jawaban dengan GAMBAR ya. Terima kasih!

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



**Re: Diskusi.1**oleh [VEYSTI DICNORYA BERNANTI 052338378](#) - Rabu, 8 Oktober 2025, 10:11

Selamat Siang

Izin menjawab diskusi

1. Dampak negatif yang akan timbul akibat tidak adanya dokumentasi dari vendor adalah :

a. Mengalami kesulitan Pemeliharaan.

Tanpa dokumentasi tim internal sulit memahami struktur program, alur logika dan fungsi tiap modul, akibatnya perbaikan bug atau pembaruan fitur menjadi sangat lambat dan beresiko tinggi menimbulkan error baru

b. Ketergantungan pada vendor

PT. ABC menjadi bergantung pada vendor lama untuk setiap perubahan kecil sekalipun. Ini menciptakan vendor lock-in — situasi di mana perusahaan sulit berpindah ke penyedia lain atau mengelola sistemnya sendiri.

c. Penurunan Efisiensi dan Produktivitas

Waktu dan tenaga banyak terbuang untuk memahami ulang program. Hal ini menunda proyek lain dan mengurangi produktivitas tim IT.

d. Kerugian Finansial

Tanpa dokumentasi, biaya untuk reverse engineering (membaca ulang dan menganalisis kode) sangat besar. Selain itu, downtime operasional juga dapat menyebabkan kerugian bisnis.

2. Dampak terhadap operasional dan biaya perusahaan dalam jangka panjang adalah sering terjadi gangguan saat modifikasi, sulit menyesuaikan program dengan perubahan bisnis, ketergantungan tinggi pada individu tertentu yang memahami kode sehingga sulit untuk regenerasi staff IT yang baru, Biaya pemeliharaan meningkat hingga 50–70% karena harus melakukan analisis ulang sistem. Setiap update atau patch memerlukan waktu lebih lama, sehingga perusahaan kalah cepat dalam menyesuaikan diri dengan pasar dan Jika sistem mengalami gangguan yang berdampak pada layanan pelanggan, reputasi perusahaan bisa menurun.

3. Solusi yang dapat saya sarankan :

a. Tim internal atau pihak ketiga melakukan analisis terhadap kode yang ada untuk membuat dokumentasi baru (misalnya diagram arsitektur, flowchart, API documentation).

Jika memungkinkan, adakan pelatihan atau konsultasi dari vendor untuk menjelaskan sistem yang telah dibuat.

b. Buat tim khusus yang bertugas menulis ulang dokumentasi teknis dan standard operating procedure (SOP) untuk pengelolaan program.

c. Wajibkan Dokumentasi dalam Kontrak Vendor

Setiap proyek harus mencakup deliverables dokumentasi (arsitektur, flow program, struktur database, API, dan user manual). Gunakan klausul penalti jika vendor tidak memenuhi kewajiban dokumentasi.

d. Lakukan project review dan audit dokumentasi sebelum proyek diserahkan ke fase produksi.

ide dan solusi inovatif yang dapat saya bagikan untuk mengatasi permasalahan ini :

Membangun Knowledge Base Digital:

Gunakan platform seperti Confluence, Notion, atau SharePoint untuk menyimpan seluruh dokumentasi proyek, pelatihan, dan hasil pembelajaran agar mudah diakses oleh semua tim.

AI-based Code Understanding Tools:

Terapkan teknologi seperti GitHub Copilot atau Sourcegraph Cody untuk membantu tim memahami struktur kode lama dengan cepat.

Sumber Referensi :

Rosa Ariani Sukanto. (2025). Rekayasa Perangkat Lunak Modul 1. Universitas Terbuka

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach. McGraw-Hill.

IEEE Std 829-2008: Software and System Test Documentation Standard.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Kamis, 9 Oktober 2025, 18:52

sangat bagus, Anda telah menekankan pentingnya transparansi informasi dari pihak vendor.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [RIANDI SOLIHIN 051676705](#) - Rabu, 8 Oktober 2025, 20:13

Izin menjawab diskusi tutor

##### 1. Dampak Negatif

- Sulit melakukan pemeliharaan dan pengembangan lanjutan karena kode sulit dipahami.
- Terjadi ketergantungan pada vendor (vendor lock-in).
- Risiko kesalahan, bug, dan keamanan meningkat.
- Penundaan proyek dan menurunnya produktivitas tim internal.

##### 2. Dampak terhadap Operasional dan Biaya

- Biaya pemeliharaan meningkat karena waktu analisis kode lebih lama.
- Gangguan operasional akibat kesalahan saat modifikasi.
- Lambat beradaptasi terhadap perubahan kebutuhan bisnis.

##### 3. Solusi yang Disarankan

- Reverse engineering dokumentasi dengan tools seperti Doxygen atau PlantUML.
- Membuat dokumentasi internal dan knowledge base dari hasil analisis kode.
- Menetapkan kebijakan kontrak bahwa setiap proyek wajib disertai dokumentasi teknis.
- Melakukan knowledge transfer dari vendor ke tim internal sebelum proyek selesai.

Kesimpulan:

Dokumentasi merupakan elemen vital dalam menjaga keberlanjutan sistem. Tanpa dokumentasi, perusahaan menanggung technical debt tinggi yang berdampak pada biaya, waktu, dan risiko operasional.

Referensi :

1. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

2. Pressman, R. S. & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.

3. IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Kamis, 9 Oktober 2025, 18:52

Jawaban Anda sudah mengarah ke solusi praktis. Sangat baik untuk konteks dunia kerja nyata

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



### Re: Diskusi.1

oleh [MARTO 055035643](#) - Rabu, 8 Oktober 2025, 20:31

Assalamu'alaikum

mohon izin untuk menanggapi diskusi ini

mohon maaf dan mohon masukan kalau masih ada tanggapan yang kurang

1. Dampak negatif yang mungkin timbul akibat tidak adanya dokumen dari vendor adalah : kesulitan dalam pengembangan lanjutan pada akhirnya berdampak pada kerugian, seperti rugi waktu, rugi biaya dan rugi-rugi yang lain.
2. Dampak terhadap operasional perusahaan dalam jangka panjang adalah kesulitan perusahaan dalam pemeliharaan dan pengembangan program dan beresiko program tersebut berhenti beroperasi.  
Dampak terhadap biaya perusahaan dalam jangka panjang adalah mengeluarkan biaya tambahan untuk mengatasi dampak tersebut, biaya tambahan untuk vendor, dan biaya – biaya operasional yang lain yang berakibat fatal devisa perusahaan.
3. Solusi yang bisa dilakukan oleh perusahaan adalah dengan cara mengatasi masalah teknis secara bertahap sambil membangun kerangka kerja dan kebijakan agar kesalahan ini tidak terulang di proyek-proyek mendatang.

Terima kasih

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Kamis, 9 Oktober 2025, 18:53

Pemikiran Anda mengenai peran dokumentasi dalam efisiensi organisasi sangat tepat.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



### Re: Diskusi.1

oleh [053149469 TAUFIK HIDAYAT](#) - Rabu, 8 Oktober 2025, 21:28

Nama : Taufik Hidayat

Prodi : Sistem informasi

ijin Menjawab :

1. Dampak Negatif Tidak Adanya Dokumentasi Vendor
  - a. Kesulitan pemeliharaan program karena tim internal harus menghabiskan waktu ekstra untuk memahami kode yang tidak terdokumentasi.
  - b. Risiko kesalahan saat melakukan modifikasi yang dapat menimbulkan bug atau gangguan operasional.
  - c. Menurunnya efektivitas dan produktivitas tim pengembang internal karena proses pembelajaran yang panjang.
  - d. Ketergantungan tinggi pada vendor untuk melakukan perubahan, jika vendor masih tersedia, atau risiko kehilangan akses pengetahuan jika vendor sudah tidak bekerja sama.
  - e. Kesulitan transfer pengetahuan dan knowledge management dalam perusahaan.
2. Dampak pada Operasional dan Biaya Jangka Panjang
  - a. Biaya pengembangan dan perbaikan meningkat karena waktu debugging dan pemahaman kode menjadi lebih

lama.

- b. Potensi downtime dan gangguan operasional meningkat saat modifikasi dilakukan tanpa pemahaman yang cukup.
- c. Peluang kehilangan keunggulan kompetitif akibat lambatnya respons terhadap perubahan kebutuhan pasar.
- d. Risiko keamanan meningkat jika kode sulit di-review dengan baik karena minim dokumentasi.

### 3. Solusi yang Disarankan

- a. Memulai program dokumentasi ulang (reverse engineering) dengan tim internal atau pihak ketiga untuk membuat dokumentasi teknis yang lengkap dan terstruktur tentang program yang ada.
- b. Menggunakan alat bantu dokumentasi otomatis yang dapat mengekstrak struktur kode, diagram alur, dan dependensi dari source code.
- c. Menerapkan standar dokumentasi wajib pada setiap proyek perangkat lunak di perusahaan, termasuk pembaruan dokumentasi secara berkala.
- d. Membuat knowledge base internal yang mudah diakses oleh tim terkait untuk memudahkan transfer pengetahuan.
- e. Mengadopsi metode Agile dan DevOps yang menekankan kolaborasi dan dokumentasi berkelanjutan selama siklus hidup pengembangan perangkat lunak.
- f. Inovasi: Mengintegrasikan sistem AI atau alat bantuan cerdas yang dapat membantu memahami dan menjelaskan kode secara otomatis untuk mempercepat pemahaman kode oleh pengembang internal.

Terima kasih

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Kamis, 9 Oktober 2025, 18:55

Jawaban Anda ringkas namun bermakna. Terus tingkatkan kemampuan menjelaskan konsep secara efisien jangan lupa tambahkan referensinya

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [MOCH RISWAN LUTFIN ANFA 051141089](#) - Rabu, 8 Oktober 2025, 22:41

Nama : Moch Riswan lutfin Anfa

Nim : 051141089

Izin menjawab pertanyaan dari diskusi diatas, trimakasih

Kasus PT. ABC ini adalah contoh klasik dari kegagalan serah terima proyek, di mana aset teknis (kode program) diserahkan tanpa disertai aset pengetahuan (dokumentasi). Kegagalan ini menciptakan "Technical Debt" yang harus ditanggung oleh tim internal PT. ABC.

Berikut adalah analisis dan jawaban saya terhadap diskusi berdasarkan studi kasus tersebut:

#### 1. Dampak Negatif Akibat Tidak Adanya Dokumentasi dari Vendor

Dampak negatif utama dari kurangnya dokumentasi (terutama dokumentasi teknis, arsitektur, dan user manual) dapat dikelompokkan menjadi tiga aspek:

##### A. Hambatan Pemeliharaan dan Pengembangan (Maintenance and Development)

- Waktu Troubleshooting yang Lebih Lama: Tim internal harus menghabiskan waktu berjam-jam atau bahkan berhari-hari untuk menelusuri kode yang tidak familiar hanya untuk menemukan sumber bug atau error.
- Ketergantungan pada Individu (Single Point of Failure): Tim menjadi sangat bergantung pada satu atau dua orang yang mungkin sempat memahami alur kode (jika ada tim yang berinteraksi dengan vendor). Jika individu tersebut pindah, perusahaan kehilangan pengetahuan vital (knowledge loss).
- Kesulitan Modifikasi (Studi Kasus PT. ABC): Seperti yang terjadi pada PT. ABC, modifikasi fitur menjadi proses coba-

coba yang berisiko tinggi (trial and error). Tim tidak yakin bagian kode mana yang dapat diubah tanpa merusak fungsi lain.

- Menghambat Onboarding Karyawan Baru: Staf teknis baru akan membutuhkan waktu yang sangat lama untuk memahami sistem yang sudah ada, menunda kontribusi produktif mereka.

#### B. Risiko Keamanan dan Kualitas Kode

- Risiko Kerentanan Keamanan: Tanpa dokumentasi arsitektur, tim kesulitan mengidentifikasi titik-titik lemah (misalnya, di mana validasi input dilakukan) yang dapat dieksploitasi, meningkatkan risiko keamanan.
- Code Spaghettification: Modifikasi yang dilakukan tanpa pemahaman mendalam cenderung menghasilkan kode yang semakin rumit, tidak terstruktur, dan sulit dipertahankan (unmaintainable).

#### C. Masalah Kepatuhan (Compliance)

- Gagal Audit Teknis: Dalam industri yang diatur ketat (misalnya keuangan atau kesehatan), kegagalan menyajikan dokumentasi teknis dan alur data yang jelas dapat menyebabkan ketidakpatuhan terhadap regulasi eksternal.

### 2. Pengaruh Jangka Panjang terhadap Operasional dan Biaya Perusahaan

Kurangnya dokumentasi akan membebani PT. ABC dalam jangka panjang, memicu peningkatan biaya total kepemilikan (Total Cost of Ownership/TCO) program tersebut:

#### - Operasional (Efisiensi Waktu)

Pengaruh Jangka Panjang :

Penurunan Kecepatan Pasar (Time-to-Market): Setiap perubahan atau pengembangan fitur baru akan memakan waktu yang jauh lebih lama. PT. ABC akan lambat merespons perubahan kebutuhan pasar, yang mengurangi daya saing.

#### - Biaya (Cost)

Pengaruh Jangka Panjang :

Peningkatan Biaya Maintenance, perusahaan harus membayar gaji tim internal untuk menghabiskan waktu pada kegiatan "de-engineering" (menganalisis kode lama) daripada kegiatan yang menghasilkan nilai (pengembangan baru). Waktu yang terbuang ini berarti biaya operasional yang lebih tinggi.

#### - Biaya Risiko

Pengaruh Jangka Panjang :

Biaya Bug dan Downtime, risiko bug pasca-modifikasi menjadi sangat tinggi. Kegagalan sistem yang disebabkan oleh modifikasi yang tidak akurat dapat menyebabkan downtime operasional, yang berujung pada kerugian finansial langsung.

#### - Biaya Re-hire Vendor

Pengaruh Jangka Panjang :

PT. ABC mungkin terpaksa mengontrak vendor yang sama (dengan biaya tinggi) atau vendor baru hanya untuk memahami kode lama dan membuat dokumentasi dasar, yang seharusnya sudah termasuk dalam proyek awal.

### 3. Solusi yang Disarankan untuk Situasi PT. ABC

Mengingat program sudah berjalan dan modifikasi harus segera dilakukan, solusi yang disarankan harus fokus pada tindakan jangka pendek untuk mitigasi krisis dan tindakan jangka panjang untuk mencegah terulangnya masalah yang sama.

#### A. Solusi Jangka Pendek (Mitigasi Krisis)

- Reverse Engineering dan Sesi Knowledge Transfer Paksa:
  - Identifikasi Pakar Vendor: PT. ABC harus segera meminta vendor untuk mengirimkan pengembang kunci yang mengerjakan proyek tersebut untuk sesi knowledge transfer intensif.
  - Dokumentasi Cepat (Ad-hoc): Tim internal ditugaskan untuk melakukan reverse engineering (menganalisis kode untuk membuat dokumentasi) pada bagian-bagian yang paling kritis dan bagian yang akan dimodifikasi.
- Manfaatkan Tools Analisis Kode Otomatis:
  - Gunakan tools seperti Static Code Analysis untuk memetakan dependensi, struktur kelas, dan menghasilkan diagram sederhana (misalnya, diagram alur data atau diagram UML dasar) dari kode yang ada. Ini dapat memberikan

pemahaman awal yang lebih cepat daripada membaca kode mentah.

#### B. Solusi Jangka Panjang (Membangun Struktur Kontrak dan Tim)

- Klausul Dokumentasi yang Ketat dalam Kontrak:

- Dalam kontrak vendor di masa depan, dokumentasi teknis harus diidentifikasi sebagai deliverable utama yang sama pentingnya dengan source code. Kontrak harus merinci format, standar (misalnya, UML diagram, API documentation, deployment guide), dan bahasa yang digunakan.

- Pencairan Pembayaran Bertahap (Holdback): Tahan persentase pembayaran terakhir (misalnya 10-20% dari total proyek) hingga dokumentasi telah diserahkan, diverifikasi, dan disetujui oleh tim teknis internal.

- Menerapkan Standar Dokumentasi Internal:

- Tim internal harus segera menetapkan standar penulisan dan tata kelola dokumentasi (menggunakan wiki internal, Confluence, atau SharePoint).

- Prinsip Code as Documentation: Mendorong penggunaan komentar kode yang konsisten dan jelas (inline documentation) serta penamaan variabel yang deskriptif.

- Pendampingan Selama Proyek (Shadowing):

- Untuk proyek masa depan, tim internal PT. ABC harus menugaskan seorang pengembang atau manajer teknis untuk bekerja bersama vendor selama proses pengembangan (shadowing). Tujuannya adalah memastikan knowledge transfer terjadi secara berkelanjutan, bukan hanya pada akhir proyek.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Kamis, 9 Oktober 2025, 18:56

Jawaban yang menarik dan argumentatif. Anda sudah mampu menilai dampak dari sisi manajerial dan teknis. tambahkan referensinya

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



#### Re: Diskusi.1

oleh [ILHAN RAHARJA NALANKO 051475472](#) - Kamis, 9 Oktober 2025, 12:51

Izin Bapak Tutor memberikan pendapat saya

#### 1. Dampak negatif akibat tidak adanya dokumentasi dari vendor

dokumentasi merupakan bagian penting dari *software development life cycle* (SDLC) karena berfungsi sebagai panduan untuk pemeliharaan, pengembangan lanjutan, dan transfer pengetahuan.

Ketika dokumentasi tidak disediakan, dampak negatif yang muncul antara lain:

- **Kesulitan pemeliharaan sistem**, karena tim baru tidak memiliki acuan tentang struktur, logika, dan dependensi antar modul.
- **Risiko kesalahan tinggi** saat melakukan modifikasi, karena perubahan bisa memengaruhi fungsi lain yang tidak teridentifikasi.
- **Ketergantungan berlebihan pada vendor**, sehingga perusahaan tidak mandiri dalam mengelola sistemnya.
- **Hilangnya pengetahuan sistem (knowledge loss)** ketika pengembang asli sudah tidak tersedia.

#### 2. Pengaruh terhadap operasional dan biaya perusahaan dalam jangka panjang

Ketiadaan dokumentasi teknis dapat menghambat operasional karena waktu dan sumber daya harus dialokasikan untuk memahami ulang sistem sebelum melakukan perubahan. proses *reverse engineering* (membaca dan memahami ulang source code) membutuhkan waktu lama dan biaya besar.

Akibatnya:



- **Biaya pengembangan dan pemeliharaan meningkat**, karena waktu kerja tim internal menjadi lebih panjang.
- **Downtime sistem lebih sering terjadi**, yang berpotensi menurunkan produktivitas operasional.
- **Perusahaan kehilangan efisiensi**, karena tidak dapat beradaptasi cepat terhadap perubahan kebutuhan pasar.

### 3. Solusi yang disarankan

solusi yang dapat diterapkan antara lain:

- **Mewajibkan dokumentasi sebagai bagian kontrak proyek** (deliverable wajib) termasuk *system design*, *user manual*, dan *technical documentation*.
- **Melakukan reverse engineering terbatas**, yaitu membaca ulang kode dan membuat dokumentasi baru secara internal agar sistem bisa dipelihara.
- **Menerapkan standar dokumentasi dan repositori terpusat**, seperti menggunakan *version control system* (Git) untuk menyimpan kode dan catatan perubahan.
- **Meningkatkan manajemen pengetahuan internal**, dengan pelatihan teknis bagi tim perusahaan agar mampu memahami sistem tanpa tergantung vendor.

Sekian Dari pendapat saya jika ada kekurangan nya mohon di berikan tambahan nya BapakTutor Terimakasih.

Sumber refrensi :

Universitas Terbuka. (2021). *Modul MSIM4303 – Rekayasa Perangkat Lunak*. Tangerang Selatan: Penerbit Universitas Terbuka.

<https://stackoverflow.blog/2024/12/19/developers-hate-documentation-ai-generated-toil-work/>

<https://forum.technosstudio.com/d/1947-pentingnya-dokumentasi-dalam-mengelola-proyek-it-untuk-menghindari-kesalahan>

Tautan permanen Tampilkan induk Balas



#### Re: Diskusi.1

oleh [AJAY SUPRIADI 04001670](#) - Kamis, 9 Oktober 2025, 18:57

Jawaban Anda cukup kuat secara konsep. Teruslah berlatih menulis argumentasi akademik.

Tautan permanen Tampilkan induk Balas



#### Re: Diskusi.1

oleh [RACHMAT SAISHO 050216292](#) - Kamis, 9 Oktober 2025, 21:06

#### 1. Dampak Negatif Tanpa Dokumentasi:

- Tim internal sulit memahami struktur dan logika program.
- Risiko error tinggi saat modifikasi fitur.
- Waktu dan tenaga untuk analisis ulang meningkat.
- Ketergantungan pada vendor awal, sehingga fleksibilitas rendah.

#### 2. Dampak Operasional dan Biaya:

- Perubahan fitur jadi lambat → menghambat inovasi bisnis.
- Biaya pengembangan meningkat karena perlu waktu lebih lama.
- Potensi downtime sistem saat update.
- Meningkatkan risiko kehilangan pengetahuan teknis (knowledge loss).

## 3. Solusi yang Disarankan:

- Buat reverse engineering untuk memetakan struktur kode.
- Susun dokumentasi teknis baru (diagram arsitektur, alur logika, API, dan modul).
- Terapkan standar dokumentasi wajib untuk semua proyek selanjutnya.
- Gunakan sistem version control (misalnya Git) dan tools dokumentasi otomatis.
- Pastikan adanya transfer knowledge dari vendor ke tim internal sejak awal.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)
**Re: Diskusi.1**oleh [AJAY SUPRIADI 04001670](#) - Jumat, 10 Oktober 2025, 04:56

Terima kasih atas pemikirannya. Semoga di diskusi mendatang dapat menambahkan data pendukung atau referensi

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)
**Re: Diskusi.1**oleh [ULIKTA MUTAWAFINA 051517222](#) - Kamis, 9 Oktober 2025, 21:44**1. Dampak Negatif Akibat Tidak Adanya Dokumentasi dari Vendor**

Tidak adanya dokumentasi teknis yang memadai setelah proyek diserahkan akan menimbulkan berbagai dampak serius, baik secara teknis maupun manajerial. Berdasarkan Modul MSIM4303 – Rekayasa Perangkat Lunak, dokumentasi merupakan artefak penting yang menjadi bagian dari software development life cycle (SDLC) dan dibutuhkan untuk maintenance serta evolution perangkat lunak.

=> Beberapa dampak negatifnya adalah:

**1. Kesulitan dalam Pemeliharaan (Maintenance Difficulty)**

Tanpa dokumentasi, tim pengembang internal sulit memahami struktur kode, dependensi antar modul, serta alur logika sistem. Hal ini membuat proses bug fixing atau feature enhancement menjadi sangat lambat.

**2. Risiko Kesalahan Implementasi Tinggi**

Ketika pengembang mencoba memodifikasi sistem tanpa memahami rancangan awal, potensi munculnya error baru meningkat. Hal ini bisa menimbulkan gangguan pada sistem operasional yang sedang berjalan.

**3. Ketergantungan pada Vendor Awal**

Karena hanya vendor yang memahami struktur sistem, perusahaan menjadi tergantung pada vendor tersebut untuk setiap perubahan atau perbaikan. Ini menimbulkan risiko vendor lock-in dan menurunkan kemandirian tim internal.

**4. Kehilangan Pengetahuan Teknis (Knowledge Loss)**

Pengetahuan tentang sistem tidak terdokumentasi dan tidak dapat ditransfer ke anggota tim baru. Akibatnya, terjadi kesenjangan pengetahuan (knowledge gap) dalam organisasi.

**2. Dampak terhadap Operasional dan Biaya Perusahaan dalam Jangka Panjang**

Dampak terhadap operasional dan biaya perusahaan dalam jangka panjang antara lain :

**1. Biaya Pemeliharaan yang Tinggi**

Tanpa dokumentasi, setiap perubahan memerlukan waktu analisis yang lama. Waktu tambahan ini secara langsung meningkatkan biaya tenaga kerja dan waktu proyek.

## 2. Penurunan Efisiensi Operasional

Ketika sistem sulit dimodifikasi, perusahaan lambat beradaptasi terhadap kebutuhan pasar yang dinamis, sehingga berpotensi kehilangan keunggulan kompetitif.

## 3. Risiko Gangguan Operasional

Kesalahan modifikasi akibat minimnya dokumentasi bisa menyebabkan system downtime, yang menghambat aktivitas bisnis utama dan merugikan perusahaan secara finansial maupun reputasi.

## 4. Kesulitan dalam Audit dan Kepatuhan (Compliance Issue)

Beberapa industri seperti perbankan atau kesehatan mengharuskan adanya dokumentasi sistem untuk keperluan audit. Tanpa dokumentasi, perusahaan berisiko melanggar standar atau regulasi tertentu.

## 3. Solusi yang Disarankan

Untuk mengatasi dan mencegah permasalahan serupa di masa depan, solusi yang dapat diterapkan meliputi :

### a. Reengineering dan Reverse Engineering

Melakukan reverse engineering untuk memahami struktur sistem yang ada. Tim internal dapat menggunakan alat bantu seperti UML tools (Enterprise Architect, Visual Paradigm, dll) untuk merekonstruksi diagram arsitektur, alur data, dan dependensi antar modul.

### b. Pembuatan Dokumentasi Retrospektif

Setelah sistem dianalisis, tim internal perlu membuat dokumentasi teknis baru, meliputi:

- Software Architecture Document (SAD)
- Data Dictionary dan ERD
- UML Diagram (Use Case, Class, Sequence)
- Coding Standard dan Modul Description

Dokumentasi ini menjadi acuan untuk pengembangan selanjutnya.

### c. Penerapan Standar Proses Rekayasa Perangkat Lunak

Gunakan model proses seperti Waterfall, Spiral, atau Agile (Scrum) yang menekankan documentation as deliverable.

### d. Pengaturan Kontrak yang Lebih Ketat dengan Vendor

Dalam kontrak kerja sama berikutnya, PT. ABC perlu mencantumkan klausul bahwa vendor wajib menyerahkan dokumentasi lengkap sebagai bagian dari deliverables proyek. Termasuk juga hak atas source code dan design document.

### e. Pelatihan dan Penguatan Tim Internal

Meningkatkan kemampuan tim internal dalam membaca, memahami, dan membuat dokumentasi perangkat lunak. Ini dapat dilakukan melalui pelatihan software documentation dan configuration management.

sumber Referensi :

- Rosa Ariani Sukanto. (2022). Modul MSIM4303 – Rekayasa Perangkat Lunak. Jakarta: Universitas Terbuka.
- Sommerville, Ian. (2016). Software Engineering (10th Edition). Pearson Education Limited.
- Pressman, Roger S. (2015). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.

**Re: Diskusi.1**oleh [AJAY SUPRIADI 04001670](#) - Jumat, 10 Oktober 2025, 04:57

Terima kasih atas uraian yang mendalam. Anda sudah memahami pentingnya dokumentasi sebagai aset pengetahuan

Tautan permanen Tampilkan induk Balas

Hide sidebars

**Re: Diskusi.1**oleh [SUPRIYADI 050127551](#) - Jumat, 10 Oktober 2025, 08:10

Assalamualaikum...Ijin Menjawab diskusi 1

**1. Dampak negatif akibat tidak adanya dokumentasi dari vendor**

Tidak adanya dokumentasi teknis menyebabkan sejumlah masalah serius, antara lain:

- Kesulitan pemeliharaan sistem (maintenance): Tim internal kesulitan memahami struktur kode, alur logika, dan dependensi antar modul.
- Risiko kesalahan tinggi: Tanpa panduan, modifikasi kode bisa menyebabkan bug baru atau kerusakan pada fungsi lain.
- Ketergantungan terhadap vendor lama: PT. ABC mungkin harus kembali ke vendor yang sama untuk perbaikan karena hanya mereka yang memahami sistem tersebut.
- Lambatnya pengembangan fitur baru: Proses analisis dan implementasi perubahan menjadi sangat lambat karena perlu memahami ulang seluruh sistem.
- Menurunnya produktivitas tim IT internal: Waktu banyak tersita untuk reverse engineering (membaca dan menelusuri kode lama tanpa panduan).

**2. Dampak terhadap operasional dan biaya perusahaan dalam jangka panjang**

Dalam jangka panjang, tidak adanya dokumentasi dapat menimbulkan:

- Biaya operasional meningkat, karena perusahaan harus menghabiskan waktu dan sumber daya besar untuk memahami sistem.
- Ketergantungan vendor meningkat, yang dapat menghambat fleksibilitas dan meningkatkan biaya outsourcing.
- Gangguan pada layanan operasional, karena modifikasi atau perbaikan sistem memerlukan waktu lama.
- Risiko kehilangan pengetahuan organisasi (knowledge loss), terutama jika pegawai yang pernah memahami sistem sudah tidak bekerja lagi.

**3. Solusi yang disarankan**

Beberapa langkah solusi yang dapat diterapkan PT. ABC antara lain:

**1. Membangun dokumentasi retroaktif (reverse documentation):**

Tim internal dapat mulai membuat dokumentasi dari hasil pembacaan dan penelusuran kode sumber yang ada (code review, flow diagram, ERD).

**2. Menerapkan kebijakan kontrak yang lebih ketat:**

Setiap proyek pengembangan software harus mewajibkan vendor menyerahkan complete technical documentation (arsitektur sistem, desain database, API specification, user manual).

**3. Menerapkan knowledge transfer session di akhir proyek:**

Vendor wajib melakukan pelatihan dan penjelasan teknis kepada tim internal sebelum proyek dinyatakan selesai.

**4. Menggunakan alat bantu dokumentasi otomatis:**

Tools seperti Doxygen, Sphinx, atau JSDoc dapat membantu menghasilkan dokumentasi dari kode sumber secara otomatis.

**5. Menerapkan software documentation standard di internal perusahaan:**

Sehingga semua proyek selanjutnya mengikuti format dokumentasi yang baku, misalnya berdasarkan IEEE Std 1063-

2001 atau ISO/IEC/IEEE 26514:2008.

Sumber Referensi:

- IEEE Std 1063-2001: Software User Documentation.
- Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education.
- ISO/IEC/IEEE 26514:2008 – Systems and software engineering — Requirements for designers and developers of user documentation.
- Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th Edition). McGraw-Hill.

Tautan permanen Tampilkan induk Balas



### Re: Diskusi.1

oleh [054453816 ACH. RIYANTO](#) - Jumat, 10 Oktober 2025, 08:36

#### 1. Dampak Negatif Akibat Ketiadaan Dokumentasi

Ketiadaan dokumentasi teknis dari vendor dapat menimbulkan serangkaian dampak negatif yang merugikan PT. ABC, antara lain:

**Kesulitan Pemeliharaan dan Pengembangan**, kemampuan perangkat lunak untuk terus dipelihara seiring berkembangnya teknologi dan kebutuhan. Tanpa dokumentasi, tim internal PT. ABC tidak dapat memelihara atau mengembangkan program tersebut secara efektif, persis seperti yang terjadi saat mereka ingin memodifikasi fitur.

**Peningkatan Risiko Kesalahan**, saat mencoba memodifikasi kode tanpa memahami arsitektur dan alur logikanya secara menyeluruh dapat secara tidak sengaja merusak fungsi lain. Hal ini dapat mengurangi tingkat keandalan (dependability) sistem, yang bisa berakibat fatal bagi program yang mendukung operasional penting perusahaan.

**Ketergantungan Tinggi pada Vendor**, ketiadaan dokumentasi membuat PT. ABC secara tidak langsung "terkunci" pada vendor asli. Jika ada kebutuhan mendesak untuk perbaikan atau perubahan, mereka mungkin terpaksa harus kembali mengontrak vendor tersebut dengan biaya yang mungkin jauh lebih tinggi.

Menciptakan "**Tantangan Warisan**" Lebih Cepat: Materi menjelaskan adanya tantangan warisan (legacy challenge), di mana perangkat lunak menjadi sulit dipahami karena dikembangkan oleh orang yang berbeda. Dalam kasus ini, PT. ABC sudah menghadapi tantangan ini hanya dalam waktu satu tahun karena pengetahuan tentang sistem tidak pernah ditransfer melalui dokumentasi.

#### 2. Pengaruh Jangka Panjang pada Operasional dan Biaya

Peningkatan Biaya Kepemilikan (Total Cost of Ownership), biaya untuk memodifikasi atau memperbaiki program akan membengkak. Tim internal harus menghabiskan waktu berjam-jam (bahkan berhari-hari) hanya untuk memahami sebagian kecil dari kode sebelum bisa mulai bekerja. Waktu adalah uang, dan inefisiensi ini secara langsung menaikkan biaya pemeliharaan, bertentangan dengan prinsip efisiensi dalam rekayasa perangkat lunak.

Kelambatan Adaptasi Bisnis, PT. ABC akan kehilangan kelincihannya. Ketika kompetitor bisa meluncurkan fitur baru dengan cepat, PT. ABC akan tertinggal karena tim teknisnya masih berjuang "membedah" source code yang tidak terdokumentasi.

Risiko Operasional, terjadi kerusakan kritis pada program, waktu perbaikan akan menjadi sangat lama. Hal ini bisa menyebabkan terhentinya operasional perusahaan, kehilangan pendapatan, dan menurunnya kepercayaan pelanggan.

Potensi Biaya Pembangunan Ulang (Rewrite), program sudah terlalu rumit dan tidak mungkin lagi untuk dimodifikasi dengan aman, satu-satunya pilihan adalah dengan membangun ulang dari awal. Ini adalah skenario terburuk yang

akan memakan biaya sangat besar, setara dengan biaya proyek awal atau bahkan lebih.

### 3. Solusi yang Disarankan

#### Solusi Jangka Pendek (Reaktif)

Hubungi Kembali Vendor, langkah pertama adalah menghubungi vendor secara formal dan meminta kelengkapan dokumentasi teknis sesuai kontrak (jika tercantum). Seringkali dokumentasi ini sudah ada namun tidak disertakan saat serah terima.

Lakukan Rekayasa Mundur (Reverse Engineering), Jika vendor tidak kooperatif, tim internal PT. ABC harus mengalokasikan sumber daya untuk menganalisis kode dan membuat dokumentasi sendiri. Proses ini harus dimulai dari modul yang paling kritis atau yang akan segera dimodifikasi. Mereka harus memetakan arsitektur, alur data, dan logika bisnis utama.

Gunakan Alat Bantu Analisis Kode, manfaatkan tools statis maupun dinamis untuk menganalisis kode. Alat-alat ini dapat membantu memvisualisasikan struktur kode, dependensi antar modul, dan alur eksekusi untuk mempercepat pemahaman.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

◀ <https://engineering.fu...>

Lompat ke...



## Navigasi

### ▼ [Dasbor](#)

🏠 [Beranda situs](#)

> [Laman situs](#)

### ▼ [Kelasku](#)

> [STSI4203.108](#)

▼ [STSI4202.42](#)

> [Peserta](#)

📊 [Nilai](#)

> [Pendahuluan](#)

### ▼ [Sesi 1](#)

📅 [Kehadiran Sesi ke-1](#)

📖 [Materi Inisiasi](#)

📖 [Materi Pengayaan: Software Engineering A Practitio...](#)

📖 <https://engineering.futureuniversity.com/BOOKS%20F...>

🗨️ [Diskusi.1](#)

> [STSI4103.119](#)

> [MKKI4201.278](#)

> [STSI4201.161](#)

> [STSI4205.331](#)



- > [STSI4104.284](#)
- > [MKDI4202.1514](#)
- > [Kelas](#)

## Administrasi

- ✓ Forum administrasi
  - Berlangganan dinonaktifkan

Hide sidebars

Follow Us:      

UNIVERSITAS TERBUKA ©2025

Anda masuk sebagai [INDRAWAN LISANTO 053724113](#) (Keluar)  
[Dapatkan aplikasi seluler](#)