

**ADOPSI CSD *MATRIX* UNTUK MENINGKATKAN
KUALITAS PROSES BISNIS TIM PENGEMBANG
PERANGKAT LUNAK (STUDI KASUS MAMIKOS)**



Disusun Oleh:

N a m a : Nabila Putri Ocktavia
NIM : 21523263

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING
ADOPSI CSD *MATRIX* UNTUK MENINGKATKAN
KUALITAS PROSES BISNIS TIM PENGEMBANG
PERANGKAT LUNAK (STUDI KASUS MAMIKOS)

TUGAS AKHIR



الجامعة الإسلامية
Yogyakarta, 22 Juli 2025

Pembimbing,

Pembimbing I,

(Beni Suranto, S.T., M.Soft.Eng.)

Pembimbing II,

(Chanifah Indah Ratnasari, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

ADOPSI CSD *MATRIX* UNTUK MENINGKATKAN KUALITAS PROSES BISNIS TIM PENGEMBANG PERANGKAT LUNAK (STUDI KASUS MAMIKOS)

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Juli 2025

Tim Penguji

Chanifah Indah Ratnasari, S.Kom., M.Kom.

Anggota 1

Elyza Gustri Wahyuni, S.T., M.Cs.

Anggota 2

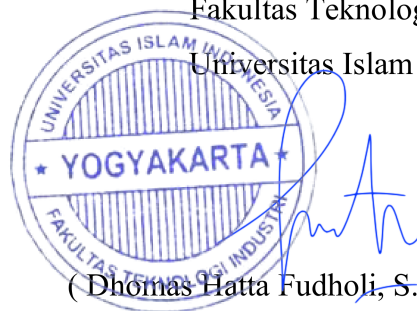
Sheila Nurul Huda, S.Kom., M.Cs.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Thomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nabila Putri Ocktavia

NIM : 21523263

Tugas akhir dengan judul:

ADOPSI CSD *MATRIX* UNTUK MENINGKATKAN KUALITAS PROSES BISNIS TIM PENGEMBANG PERANGKAT LUNAK (STUDI KASUS MAMIKOS)

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Juli 2025



(Nabila Putri Ocktavia)

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur dan kerendahan hati, skripsi ini saya persembahkan kepada: Mama saya dan Kakak tercinta saya yang telah mencurahkan kasih sayang, dukungan, dan doa yang tiada henti sehingga saya dapat menyelesaikan pendidikan ini. Setiap tetes keringat dan air mata Mama adalah motivasi terbesar bagi saya untuk terus berjuang. Saudara-saudara saya yang selalu memberikan semangat dan menjadi tempat berbagi cerita dalam setiap langkah perjalanan akademik ini. Para dosen dan pembimbing yang dengan sabar telah membagikan ilmu, bimbingan, dan arahan selama proses pembelajaran hingga penyelesaian skripsi ini. Sahabat-sahabat seperjuangan yang telah menemani, mendukung, dan bersama-sama melewati suka dan duka selama masa perkuliahan. Serta untuk diriku sendiri, yang telah bertahan dan berjuang hingga titik ini. Semoga pencapaian ini menjadi langkah awal untuk terus berkarya dan memberikan manfaat bagi masyarakat dan ilmu pengetahuan.

HALAMAN MOTO

“Setiap proses memiliki nilai, dan setiap masalah adalah peluang untuk berinovasi.”

“Kita tidak hanya membangun perangkat lunak, tetapi juga pengalaman dan perubahan.”

“Kemajuan sejati dalam teknologi lahir dari empati, kolaborasi, dan keberanian untuk berubah.”

“Logic will get you from A to B. Imagination will take you everywhere.”

— Albert Einstein

“Innovation distinguishes between a leader and a follower.”

“Design is not just what it looks like and feels like. Design is how it works.”

— Steve Jobs

“We cannot solve our problems with the same thinking we used when we created them.”

— Albert Einstein

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirabbil'alamini, dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang. Segala puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat, karunia, dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW beserta keluarga, sahabat, dan pengikutnya hingga akhir zaman.

Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan dan memperoleh gelar Sarjana Komputer (S.Kom). Dalam proses penyusunan Tugas Akhir ini, penulis merasakan berbagai macam tantangan, pembelajaran berharga, dan pengalaman yang tak terlupakan. Penulis menyadari bahwa keberhasilan dalam menyelesaikan Tugas Akhir ini tidak terlepas dari bantuan, bimbingan, doa, serta dukungan dari berbagai pihak.

Dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang tulus dan sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan kekuatan, kesehatan, kesabaran, dan kemudahan dalam menghadapi setiap tantangan selama proses penyelesaian Tugas Akhir ini.
2. Mama saya yang tidak pernah lelah memberikan kasih sayang, perhatian, motivasi, dan doa yang tiada henti. Terima kasih telah menjadi sumber kekuatan dan inspirasi bagi penulis untuk terus berjuang menggapai cita-cita.
3. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor yang telah memberikan kesempatan kepada penulis untuk menempuh pendidikan dan mengembangkan diri di kampus ini.
4. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU, ASEAN.Eng., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia yang senantiasa memberikan dukungan dan arahan selama proses studi penulis.
5. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika yang telah membimbing dan memfasilitasi kebutuhan akademik penulis selama masa perkuliahan.
6. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika yang telah memberikan arahan serta motivasi dalam proses penyusunan tugas akhir ini.

7. Bapak Beni Suranto, S.T., M.Soft.Eng. selaku dosen pembimbing yang dengan sabar telah meluangkan waktu untuk memberikan bimbingan, arahan, kritik, motivasi, dan saran yang sangat berharga bagi penulis dalam proses penyusunan Tugas Akhir ini.
8. Ibu Chanifah Indah Ratnasari, S.Kom., M.Kom. selaku dosen pembimbing kedua saya yang telah meluangkan waktunya untuk memberikan bimbingan, arahan, motivasi serta saran yang sangat membantu saya dalam proses penyusunan Tugas Akhir ini.
9. Seluruh dosen Program Studi Informatika yang telah memberikan ilmu pengetahuan, pengalaman, dan wawasan yang sangat bermanfaat bagi penulis selama menempuh pendidikan
10. Kakak/adik tersayang, Mas Eki, Mas Zulfa, Mas Rian, Adek Ayik dan sepupuku tersayang Mba Dida yang selalu memberikan semangat, keceriaan, dan motivasi kepada penulis untuk terus berjuang menyelesaikan Tugas Akhir ini.
11. Sahabat-sahabat terbaik: Jeihan, Adel, Fitri, Naela, Zila, Juan, Rista, Mahen, Arul, Zahwa, Asista, Mela yang selalu ada dalam suka dan duka, memberikan dukungan, bantuan, dan menjadi tempat berbagi cerita selama menjalani masa perkuliahan.
12. Teruntuk orang favoritku dan terspesial Mas Iqbal, terima kasih atas semuanya, kesabaran, perhatian, dan dukungan yang tiada henti. Terima kasih telah menjadi penyemangat, pendengar yang baik, dan selalu ada di saat suka maupun duka. Kehadiran dan dukunganmu memberikan kekuatan tersendiri bagi penulis untuk tetap bertahan dan berjuang menyelesaikan Tugas Akhir ini.
13. Teman-teman seperjuangan Program Studi Informatika Angkatan 2021 yang telah bersama-sama berjuang dan saling mendukung dalam menempuh pendidikan serta memberikan kenangan indah yang tak terlupakan.
14. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu baik secara langsung maupun tidak langsung dalam penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan dan masih terdapat banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang. Harapan penulis, semoga Tugas Akhir ini dapat memberikan manfaat dan kontribusi positif bagi perkembangan ilmu pengetahuan serta dapat menjadi inspirasi bagi pembaca.

Akhir kata, penulis berharap semoga Tugas Akhir ini dapat memberikan manfaat bagi penulis sendiri, almamater tercinta, dan bagi pembaca pada umumnya. Semoga Allah SWT membalas kebaikan semua pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini dan melimpahkan rahmat serta hidayah-Nya kepada kita semua.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 22 Juli 2025

A handwritten signature in black ink, consisting of a large, stylized initial 'N' followed by a series of loops and a long horizontal stroke.

(Nabila Putri Ocktavia)

SARI

Dalam proses pengembangan perangkat lunak, efektivitas kolaborasi tim, dokumentasi keputusan, serta pengelolaan risiko menjadi faktor krusial dalam menjaga kualitas hasil kerja. Tim pengembang perangkat lunak di Mamikos sebelumnya telah menerapkan metode *Scrum* berbasis *Agile*, namun masih ditemukan beberapa kelemahan, seperti kurangnya dokumentasi yang terstruktur dan tidak terdokumentasinya perubahan kebutuhan secara konsisten. Oleh karena itu, penelitian ini dilakukan untuk menganalisis dampak penerapan metode *CSD Matrix* terhadap peningkatan kualitas proses bisnis tim pengembang perangkat lunak.

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan metode studi kasus pada tim *Engineering* Mamikos. Teknik pengumpulan data meliputi observasi, wawancara, dokumentasi, dan kuesioner, yang mengacu pada indikator *Key Process Area (KPA)* dalam kerangka kerja *Capability Maturity Model Integration (CMMI)*. Responden dalam penelitian ini berjumlah 30 orang dari berbagai peran di tim engineer, mulai dari *Engineering manager*, *backend/frontend engineer*, QA, hingga *data engineer*. Evaluasi dilakukan sebelum dan sesudah penerapan *CSD Matrix* untuk melihat perubahan tingkat kematangan proses bisnis.

Hasil penelitian menunjukkan bahwa penerapan *CSD Matrix* memberikan dampak signifikan terhadap peningkatan struktur dokumentasi, efektivitas kolaborasi tim, serta kejelasan tanggung jawab dalam proses pengembangan proyek. Rata-rata capaian KPA meningkat dari 2,69 (54%) sebelum penerapan menjadi 3,71 (74%) setelah penerapan, yang berdasarkan standar CMMI menunjukkan peningkatan dari *Maturity Level 2 (Managed)* ke *Maturity Level 3 (Defined)*. Seluruh catatan proyek yang mencakup *Certainties*, *Suppositions*, dan *Doubts* kini terdokumentasi secara terpusat sehingga mudah diakses, diperbarui, dan digunakan kembali saat terjadi perubahan kebutuhan atau saat memulai proyek baru. Penerapan *CSD Matrix* terbukti menjadi langkah strategis dalam mendukung *Software Process Improvement* di lingkungan pengembangan perangkat lunak Mamikos.

Kata kunci: *CSD Matrix*, proses bisnis, dokumentasi, CMMI, *Software Process Improvement*.

GLOSARIUM

Istilah	Definisi
<i>Agile</i>	: Metodologi pengembangan perangkat lunak yang menekankan iterasi cepat, kolaborasi tim, dan kemampuan beradaptasi terhadap perubahan.
<i>Backend Engineer</i>	: Pengembang perangkat lunak yang fokus pada logika server, basis data, dan fungsionalitas sisi server.
<i>Brainstorming</i>	: Proses diskusi kelompok untuk menghasilkan ide-ide secara spontan dan kreatif.
<i>Certainties</i>	: Elemen dalam CSD <i>Matrix</i> yang mencakup informasi yang telah terverifikasi dan dianggap pasti.
<i>CMMI (Capability Maturity Model Integration)</i>	: Kerangka kerja untuk menilai dan meningkatkan kematangan proses dalam organisasi pengembangan produk dan layanan.
<i>CSD Matrix</i>	: Metode visual yang mengelompokkan pengetahuan tim ke dalam tiga kategori: <i>Certainties</i> , <i>Suppositions</i> , dan <i>Doubts</i> .
<i>DevOps</i>	: Praktik integrasi antara pengembangan perangkat lunak (Development) dan operasional sistem (Operations) untuk mempercepat pengiriman produk.
<i>Doubts</i>	: Bagian dari CSD <i>Matrix</i> yang berisi pertanyaan atau hal yang belum diketahui dan membutuhkan eksplorasi lebih lanjut.
<i>Frontend Engineer</i>	: Pengembang perangkat lunak yang fokus pada tampilan dan interaksi pengguna pada antarmuka aplikasi.
<i>KPA (Key Process Area)</i>	: Area proses utama dalam kerangka CMMI yang menjadi indikator penilaian kematangan proses organisasi.
<i>Managed (CMMI Level 2)</i>	: Tingkatan kedua dalam CMMI yang menunjukkan proses telah direncanakan, dilacak, dan dikendalikan.
<i>Mamikos</i>	: <i>Startup</i> teknologi di Indonesia yang menyediakan layanan pencarian dan pemesanan kos secara digital.
<i>QA (Quality Assurance)</i>	: Proses untuk memastikan bahwa perangkat lunak memenuhi standar kualitas dan bebas dari kesalahan.
<i>Software Process Improvement (SPI)</i>	: Upaya sistematis untuk meningkatkan efektivitas dan efisiensi proses pengembangan perangkat lunak.
<i>Suppositions</i>	: Hipotesis atau anggapan dalam CSD <i>Matrix</i> yang belum terbukti kebenarannya namun cukup masuk akal.
<i>User Research</i>	: Proses pengumpulan informasi tentang kebutuhan dan perilaku pengguna untuk mendukung pengembangan produk.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI.....	x
GLOSARIUM	xi
DAFTAR ISI	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR.....	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Metodologi	5
1.6.1 Metode Pengumpulan Data	5
1.6.2 Metode Perencanaan	6
1.6.3 Metode Pengukuran	6
1.6.4 Metode Pengujian	7
1.7 Sistematika Penulisan	7
BAB II LANDASAN TEORI	8
2.1 CSD Matrix	8
2.2 Software Process Improvement	11
2.3 Kualitas Proses Bisnis	13
2.4 ISO/IEC 15504 (SPICE – Software Process Improvement and Capability Determination)	15
2.5 ISO/IEC 33000 Series	16
2.6 Personal Software Process (PSP)	16

2.7	<i>Team Software Process (TSP)</i>	17
2.8	CMMI.....	18
2.9	Penelitian Terdahulu	20
2.9.1	Metode Pencarian Pustaka	20
BAB III METODOLOGI		29
3.1	Metode Penelitian	29
3.2	Subjek dan Objek Penelitian	29
3.3	Teknik Pengumpulan Data.....	31
3.4	Alur Penelitian	33
BAB IV HASIL DAN PEMBAHASAN.....		44
4.1	Analisis Identifikasi Masalah.....	44
4.2	Hasil Pengumpulan Data.....	44
4.2.1	Struktur Tim Mamikos.....	44
4.2.2	Lingkungan Kerja dan Alat yang Digunakan	46
4.3	Analisis Pengukuran <i>Pre-Test</i> Menggunakan CMMI Sebelum Penerapan CSD <i>Matrix</i>	47
4.4	Penerapan CSD <i>Matrix</i>	49
4.5	Analisis Pengukuran <i>Post-Test</i> Menggunakan CMMI Setelah Penerapan CSD <i>Matrix</i>	54
4.6	Perbandingan Tingkat Kematangan Proses Sebelum dan Sesudah Perhitungan Manual	55
4.7	Perancangan dan Pengembangan Sistem Android.....	58
4.7.1	<i>Interface</i> Dashboard	59
4.7.2	<i>Interface</i> Perhitungan Proses Bisnis	61
4.7.3	<i>Interface</i> Perbandingan Grafik.....	67
4.8	Analisis Perbandingan dan Pengujian Hasil Secara Manual dan Sistem.....	69
4.9	Pembahasan.....	70
4.10	Dampak Adopsi CSD <i>Matrix</i> Terhadap Proses Bisnis Tim	72
4.11	Tantangan dan Kendala Implementasi CSD <i>Matrix</i>	73
4.12	Keterbatasan Penelitian	73
BAB V KESIMPULAN DAN SARAN		75
5.1	Kesimpulan.....	75
5.2	Saran	75
DAFTAR PUSTAKA.....		77

LAMPIRAN	82
----------------	----

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	20
Tabel 2. 2 Perbandingan Metode dalam Evaluasi dan Peningkatan Proses Bisnis Tim Pengembang Perangkat Lunak.....	26
Tabel 3. 1 Pertanyaan Wawancara Berdasarkan Kualitas Proses Bisnis	36
Tabel 3. 2 Kuesioner Sebelum dan Sesudah Penerapan CSD Matrix.....	38
Tabel 3. 3 Pengukuran Pada CMMI	40
Tabel 4. 1 Hasil Perhitungan CMMI Sebelum Implementasi CSD <i>Matrix</i>	47
Tabel 4. 2 Perhitungan Setelah Implementasi CSD <i>Matrix</i>	54
Tabel 4. 3 Perbandingan Tingkat Kematangan Proses Bisnis Sebelum dan Sesudah Implementasi CSD <i>Matrix</i>	55
Tabel 4. 4 Codingan Grafik Perbandingan	57
Tabel 4. 5 Perbandingan Analisis Manual dan Analisis Aplikasi Pada Perhitungan CMMI ..	69
Tabel 4. 6 Perbandingan Kondisi Tim Pengembang Sebelum dan Sesudah Penerapan CSD <i>Matrix</i>	71

DAFTAR GAMBAR

Gambar 2. 1 Framework CSD <i>Matrix</i>	8
Gambar 3. 1 Alur Penelitian	33
Gambar 4. 1 Struktur Organisasi Mamikos	45
Gambar 4. 2 <i>Apple Music Affiliate Landing page</i> in Mamikos Web.....	49
Gambar 4. 3 CSD <i>Matrix</i> Fase Identifikasi	50
Gambar 4. 4 Fase <i>Brainstroming</i> CSD <i>Matrix</i>	51
Gambar 4. 5 Fase Validasi dan Pengambilan Keputusan	53
Gambar 4. 6 Grafik Perbandingan Persentase Berdasarkan KPA	57
Gambar 4. 7 Kode Untuk Dashboard.....	61
Gambar 4. 8 Tampilan Pada Dashboard	61
Gambar 4. 9 Kode Tampilan Kuesioner	63
Gambar 4. 10 Kode Untuk Pertanyaan Pengukuran Dengan CMMI.....	65
Gambar 4. 11 Tampilan Pertanyaan Sebelum Penerapan CSD <i>Matrix</i> Pada Apps	65
Gambar 4. 13 Kode Perhitungan KPA dan Menentukan <i>Maturity level</i>	66
Gambar 4. 14 Tampilan Hasil Perhitungan gan CMMI Sebelum dan Sesudah Penerapan CSD <i>Matrix</i>	67
Gambar 4. 15 Kode Perbandingan Grafik.....	68
Gambar 4. 16 Tampilan Perbandingan Grafik Sebelum dan Sesudah Pada Apss	69

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era transformasi digital yang sangat kompetitif, perangkat lunak bukan hanya menjadi produk, tetapi juga elemen inti dalam proses bisnis perusahaan digital. Terutama bagi *startup* teknologi, keberhasilan tidak hanya ditentukan oleh kualitas aplikasi yang dirilis, tetapi juga oleh bagaimana proses pengembangannya dijalankan. Kualitas proses bisnis tim pengembang perangkat lunak yang mencakup koordinasi, dokumentasi keputusan, respon terhadap perubahan, serta kemampuan berkolaborasi lintas divisi menjadi krusial dalam menentukan efisiensi dan daya saing perusahaan (Aguilar & Li, 2023).

Startup umumnya bergerak di dua sektor utama, yaitu penjualan produk dan penyediaan jasa. Pada kedua sektor ini, perangkat lunak berperan sebagai penghubung vital antara perusahaan dan pengguna. Namun, tim pengembang sering menghadapi tekanan tinggi akibat siklus pengembangan yang cepat dan tuntutan inovasi berkelanjutan. Tanpa proses bisnis internal yang solid, mereka akan kesulitan mengelola kompleksitas proyek, menurunkan kualitas produk akhir, dan memperlambat laju inovasi (Maulana, 2023).

Kualitas proses pengembangan mencakup praktik-praktik seperti alur kerja yang terdokumentasi dengan baik, proses pengambilan keputusan yang melibatkan seluruh anggota tim dan bersifat transparan, serta pemetaan risiko dan saran teknis secara terstruktur. Jika tidak dikelola dengan baik, akan timbul duplikasi tugas, miskomunikasi, keterlambatan rilis, hingga keputusan pengembangan yang keliru, yang pada akhirnya menurunkan performa aplikasi, pengalaman pengguna, dan reputasi bisnis (Albuquerque et al., 2021).

Mamikos adalah *startup* teknologi Indonesia yang bergerak di bidang layanan pencarian dan pemesanan kos secara digital. Berdiri sejak 11 November 2015, Mamikos telah menjalin kemitraan dengan lebih dari 120 ribu pemilik kos dan melayani 6–8 juta pencari kos tiap bulan di lebih dari 140 kota (Sukses, 2022). Dengan lebih dari 3 juta kamar terdaftar dan volume interaksi digital yang tinggi, tim pengembang perangkat lunak Mamikos menghadapi tantangan besar dalam menjaga stabilitas sistem, mempercepat siklus rilis, dan menyesuaikan fitur dengan kebutuhan pengguna yang terus berkembang (Mamikos, 2020).

Selain masalah teknis, kualitas proses bisnis tim pengembang juga diuji dari cara mereka menangani insiden dan perubahan kebutuhan. Kolaborasi lintas fungsi di Mamikos terbukti

sukses ketika mereka menggandeng *platform engagement* MoEngage, yang mampu meningkatkan pemesanan kamar hingga 20%, menaikkan CTR kampanye konversi sebesar 30%, dan mendorong 9% lebih banyak pengguna beralih dari tahap “Kamar Dilihat” ke “*Booking*”. Keberhasilan ini menegaskan pentingnya kerangka kerja prosedural yang formal untuk memastikan integrasi data, eksekusi kampanye, dan pengambilan keputusan dapat berjalan efektif (Antara, 2020).

Dari perspektif proses bisnis internal, data besar ini menuntut adanya alur pendokumentasian keputusan, pemetaan risiko, dan *feedback loop* yang terintegrasi antara unit produk, QA, dan DevOps. Tiap kali terjadi lonjakan trafik atau peluncuran fitur misalnya fitur virtual tour, filter fasilitas, atau Booking Langsung harus diikuti rapat koordinasi, klarifikasi technical debt, mitigasi bug, hingga evaluasi usulan perbaikan. Tanpa proses yang terstruktur, tim rentan melakukan tugas ganda, miskomunikasi, atau gagal menyelesaikan eskalasi secara tersistem (Mamikos, 2020).

Sebagai *startup* teknologi yang masih dalam tahap pertumbuhan, Mamikos menunjukkan karakteristik proses bisnis yang bersifat dinamis, kadang *ad-hoc*, dan belum sepenuhnya terdokumentasi secara sistematis terutama dalam hal pendefinisian kebutuhan, validasi asumsi, serta manajemen risiko. Karakter ini sangat cocok untuk diuji menggunakan *CSD Matrix*, karena *tools* tersebut memang dirancang untuk membantu tim merumuskan dan mengelola ketidakpastian sejak awal proyek. Selain itu, keberadaan Mamikos di Yogyakarta juga menyediakan kemudahan akses bagi peneliti dalam melakukan observasi langsung dan wawancara internal, sehingga data yang diuji bersifat kuat dan representatif (Lidyana, 2022).

Tantangan dalam proses bisnis tim pengembang di perusahaan seperti Mamikos bukan hanya berasal dari sisi teknis, tetapi juga dari sisi manajerial. Misalnya, pengambilan keputusan sering kali tidak terdokumentasi dengan baik, saran teknis dari anggota tim tidak tertangkap secara sistematis, dan tidak semua *concern* tim tersalurkan. Hal ini berpotensi menurunkan efisiensi dan menyebabkan inkonsistensi dalam pengembangan. Untuk itu, dibutuhkan sebuah pendekatan yang tidak hanya mendorong kolaborasi, tetapi juga mencatat proses pengambilan keputusan secara jelas (Castro & Peña, 2023).

Salah satu metode yang dapat diadopsi untuk menjawab tantangan tersebut adalah *CSD Matrix*. Metode ini merupakan alat bantu visual yang umumnya dikenal sebagai *CSD Matrix*, yang terdiri dari tiga elemen utama yaitu *Certainties* (hal-hal yang sudah diketahui dan terverifikasi), *Suppositions* (asumsi atau hipotesis yang masuk akal namun belum diuji), dan *Doubts* (keraguan atau pertanyaan terbuka yang memerlukan eksplorasi lebih lanjut) (Kaley,

2023). Kerangka ini digunakan untuk memetakan wawasan tim berdasarkan hasil riset pengguna guna mendukung pengambilan keputusan berbasis bukti. Hal ini secara langsung berdampak pada peningkatan kualitas proses bisnis dalam tim pengembang perangkat lunak.

Di sisi lain, Rukita sebagai salah satu proptech *startup* besar di Indonesia, telah mengembangkan ekosistem properti digital lengkap mulai dari layanan *coliving*, manajemen aset hingga *onboarding* dan digital monitoring yang tercermin dari keberadaan fitur RuManage dan integrasi teknologi dalam operasionalnya (Rukita, 2024). Struktur operasional yang sudah matang dan penggunaan sistem digital yang mapan membuat nilai tambah dari penerapan CSD *Matrix* di Rukita menjadi lebih rendah. Hal ini menguatkan keputusan untuk memilih Mamikos sebagai objek studi, karena keuntungan utama CSD *Matrix* justru terlihat jelas di lingkungan yang prosesnya masih fleksibel dan membutuhkan kerangka eksplorasi yang sistematis (Rukita, 2024).

Agar dampak penerapan CSD *Matrix* di Mamikos dapat dibuktikan secara objektif, diperlukan metode evaluasi yang mampu mengukur tingkat kematangan proses secara menyeluruh. Di sinilah *Capability Maturity Model Integration* (CMMI) dipilih, karena kerangka ini menyediakan lima level kematangan proses yang dapat menilai keberhasilan CSD *Matrix* dalam meningkatkan efektivitas, efisiensi, dan kontrol pengembangan perangkat lunak (Ruzhnikov & Prasetyo, 2024). Metode CSD *Matrix* paling tepat digunakan pada fase *brainstorming* untuk membantu tim menyampaikan kekhawatiran, merumuskan usulan, dan mengambil keputusan yang bertanggung jawab terutama dalam lingkungan *startup* dinamis seperti Mamikos yang membutuhkan keputusan cepat namun berdampak besar (Gavriel, 2023).

Dengan mempertimbangkan konteks tersebut, maka adopsi CSD *Matrix* di lingkungan tim pengembang Mamikos menjadi langkah yang relevan dan strategis untuk meningkatkan kualitas proses bisnis internal. Berdasarkan pemaparan yang telah disampaikan maka peneliti melakukan penelitian dengan judul “Adopsi CSD *Matrix* untuk Meningkatkan Kualitas Proses Bisnis Tim Pengembang Perangkat Lunak (Studi Kasus Mamikos).”

1.2 Rumusan Masalah

Berdasarkan latar belakang yang disusun di atas, penelitian ini memiliki rumusan masalah sebagai berikut:

- a. Tim pengembang Mamikos menghadapi tantangan dalam dokumentasi keputusan, koordinasi lintas divisi, dan manajemen risiko yang belum terstruktur, bagaimana adopsi CSD *Matrix* dapat meningkatkan kualitas proses bisnis?

- b. Bagaimana hasil evaluasi adopsi CSD *Matrix* menggunakan pengukuran CMMI dalam meningkatkan kualitas proses bisnis tim pengembang perangkat lunak (Studi Kasus Mamikos)?

1.3 Batasan Masalah

Untuk mengarahkan fokus penelitian secara lebih spesifik dan menghindari perluasan pembahasan yang terlalu umum, maka penelitian ini dibatasi pada hal-hal berikut:

- a. Penelitian ini hanya difokuskan pada tim pengembang perangkat lunak (*software development team*) internal di perusahaan Mamikos.
- b. Cakupan penelitian dibatasi pada penggunaan dan implementasi metode CSD *Matrix* dalam mendukung proses bisnis pengembangan perangkat lunak.
- c. Penerapan CSD *Matrix* (*Certainties, Suppositions, Doubts*) dalam kegiatan *brainstorming*, perencanaan, dan pengambilan keputusan pada proyek pengembangan perangkat lunak.
- d. Studi dilakukan pada proyek yang sedang berlangsung atau proyek pengembangan perangkat lunak yang telah selesai dalam 1 tahun terakhir.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk:

- a. Menganalisis cara adopsi metode CSD *Matrix* dalam proses kerja tim pengembang perangkat lunak di Mamikos.
- b. Menilai efektivitas CSD *Matrix* dalam membantu tim dalam menyusun keputusan, mengelola masukan, dan mengatasi ketidakpastian dalam pengembangan perangkat lunak.
- c. Mengidentifikasi dampak implementasi CSD *Matrix* terhadap kualitas proses bisnis dalam tim pengembang, khususnya terkait efisiensi kerja dan kolaborasi lintas fungsi.
- d. Untuk mengetahui hasil evaluasi pengukuran menggunakan CMMI dalam mengukur sebelum dan sesudah implementasi CSD *Matrix*.

Menemukan tantangan yang dihadapi dalam pengintegrasian metode ini, serta merumuskan solusi strategis untuk mendukung keberlanjutan implementasi CSD *Matrix*, sehingga dapat memastikan peningkatan kualitas proses bisnis tim pengembang perangkat lunak secara berkelanjutan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi baik secara teoritis maupun praktis dalam upaya meningkatkan kualitas proses pengembangan melalui adopsi metode CSD *Matrix*. Adapun manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

a. Manfaat Teoritis

Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan pengetahuan di bidang manajemen proses bisnis perangkat lunak, khususnya dalam konteks penerapan metode visual seperti CSD *Matrix*. Hasil penelitian ini juga diharapkan dapat memperkaya literatur tentang kolaborasi tim dan pengambilan keputusan berbasis struktur dalam proyek teknologi digital.

b. Manfaat Praktis

Bagi Mamikos, hasil penelitian ini diharapkan dapat memberikan wawasan dan rekomendasi praktis terkait penerapan CSD *Matrix* dalam aktivitas kerja tim pengembang perangkat lunak. Selain itu, temuan penelitian ini juga dapat menjadi referensi bagi *startup* teknologi lain dalam mengelola proses bisnis pengembangan perangkat lunak agar lebih efisien, transparan, dan terukur.

1.6 Metodologi

Bagian ini menjelaskan tahapan dan pendekatan yang digunakan peneliti dalam menjalankan proses penelitian, mulai dari pengumpulan data hingga evaluasi hasil. Penelitian ini dirancang menggunakan pendekatan *mixed methods* dengan desain eksplanatori sekuensial, yaitu menggabungkan pendekatan kualitatif dan kuantitatif secara berurutan dengan studi kasus pada tim pengembang perangkat lunak di perusahaan Mamikos.

1.6.1 Metode Pengumpulan Data

Teknik pengumpulan data dilakukan dengan berbagai pendekatan berikut:

1. Studi Pustaka

Data dikumpulkan melalui penelusuran literatur terkait topik penelitian, termasuk jurnal ilmiah, buku referensi, laporan penelitian, serta artikel akademik baik dalam bentuk cetak maupun elektronik. Studi pustaka ini digunakan untuk memperkuat dasar teori terkait CSD *Matrix*, SPI, dan model evaluasi CMMI.

2. Studi Lapangan

a. Observasi

Observasi dilakukan terhadap aktivitas tim pengembang perangkat lunak di Mamikos, khususnya pada proyek “*Apple Music Affiliate Landing page*.” Peneliti mencatat interaksi tim, penggunaan metode, serta alur kerja secara langsung untuk memahami implementasi *CSD Matrix* dalam konteks nyata.

b. Wawancara

Wawancara dilakukan kepada *Engineering Manager* dan beberapa anggota tim teknis lainnya. Wawancara bersifat semi-terstruktur dengan tujuan menggali informasi mengenai tantangan kolaborasi, alur pengambilan keputusan, dan persepsi terhadap penggunaan *CSD Matrix*.

c. Kuesioner

Kuesioner disebarikan kepada tim pengembang sebelum dan sesudah penerapan *CSD Matrix*. Instrumen kuesioner disusun berdasarkan indikator dari *Key Process Area* (KPA) dalam model CMMI untuk mengevaluasi kematangan proses bisnis secara kuantitatif.

d. Dokumentasi

Dokumentasi berupa arsip internal proyek seperti notulen rapat, hasil *brainstorming* *CSD Matrix*, struktur organisasi tim, dan laporan retrospektif digunakan untuk memperkuat data dari observasi dan wawancara.

1.6.2 Metode Perencanaan

Perencanaan dilakukan untuk mengintegrasikan metode *CSD Matrix* ke dalam alur kerja tim pengembang perangkat lunak. Tahapan ini mencakup identifikasi kebutuhan tim, penyusunan struktur *CSD Matrix* (*Certainties, Suppositions, Doubts*), dan pelaksanaan sesi *brainstorming*. Perencanaan juga memastikan bahwa metode ini digunakan sebagai alat bantu dalam merumuskan keputusan teknis dan meningkatkan dokumentasi serta kolaborasi dalam proyek.

1.6.3 Metode Pengukuran

Pengukuran dalam penelitian ini dilakukan untuk mengevaluasi efektivitas penerapan *CSD Matrix* terhadap kualitas proses tersebut. Teknik pengukuran menggunakan indikator dari *Capability Maturity Model Integration* (CMMI), khususnya *Key Process Area* (KPA) yang relevan. Data diperoleh dari hasil kuesioner *pre-test* dan *post-test*, kemudian dianalisis untuk mengetahui perubahan tingkat kematangan proses dari sebelum hingga setelah implementasi metode. Proses pengukuran dilakukan dengan dua pendekatan, yaitu perhitungan manual menggunakan tabel evaluasi CMMI, serta secara otomatis melalui sistem yang telah dirancang

pada *Android Studio*. Hasil dari kedua metode tersebut kemudian dibandingkan untuk memastikan konsistensi dan keakuratan hasil pengukuran.

1.6.4 Metode Pengujian

Pengujian dilakukan menggunakan sistem yang dirancang pada *Android Studio* dengan bahasa pemrograman *Java* untuk mengukur tingkat kematangan proses bisnis berdasarkan model CMMI. Sistem ini memungkinkan proses pengisian dan pengolahan kuesioner dilakukan secara otomatis. Hasil pengukuran dari sistem kemudian dibandingkan dengan perhitungan manual untuk memastikan akurasi dan keandalan evaluasi.

1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun dalam lima bab utama, yaitu:

BAB I PENDAHULUAN

Menjelaskan latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Membahas teori-teori yang relevan dengan penelitian, seperti *CSD Matrix*, *Software Process Improvement*, kualitas proses bisnis, dan kerangka CMMI. Juga memuat kajian penelitian terdahulu sebagai pembanding dan penguat argumen penelitian.

BAB III METODOLOGI

Menjelaskan pendekatan penelitian, subjek dan objek penelitian, teknik pengumpulan data (observasi, wawancara, kuesioner, dan dokumentasi), serta alur dan desain studi kasus yang diterapkan.

BAB IV HASIL DAN PEMBAHASAN

Menyajikan hasil temuan dari implementasi *CSD Matrix* di tim pengembang Mamikos, baik sebelum maupun sesudah penerapan, serta hasil evaluasi menggunakan indikator CMMI.

BAB V KESIMPULAN DAN SARAN

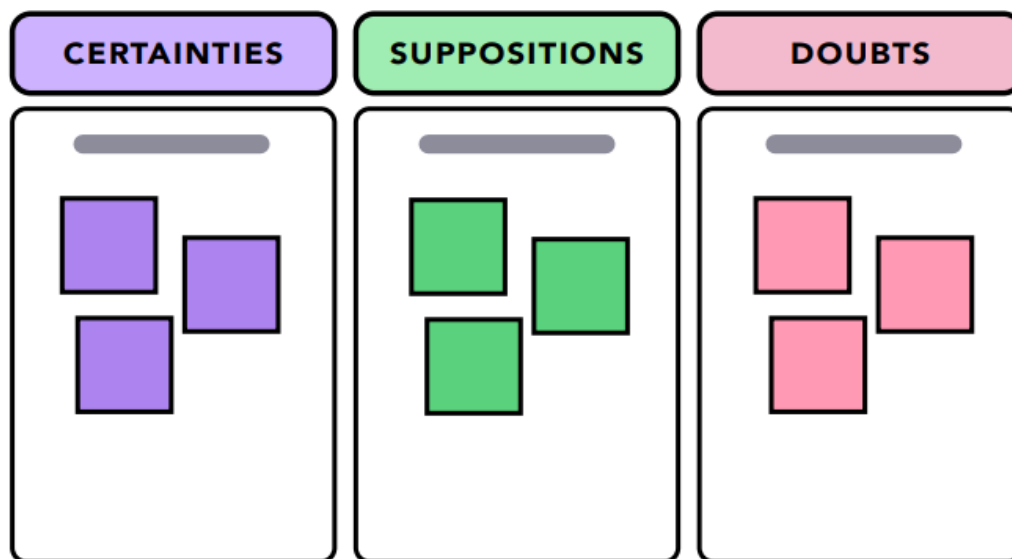
Menyimpulkan hasil penelitian, menjawab rumusan masalah, serta memberikan saran untuk implementasi *CSD Matrix* di masa depan dan pengembangan penelitian lebih lanjut.

BAB II

LANDASAN TEORI

2.1 CSD Matrix

CSD *Matrix* adalah alat bantu visual yang digunakan dalam proses pengambilan keputusan kolektif, perancangan sistem, dan pengembangan strategi, terutama dalam konteks desain sistem yang kompleks atau proyek multidisipliner. CSD merupakan singkatan dari *Certainties*, *Suppositions*, dan *Doubts*, yang merepresentasikan tiga kategori pemikiran tim terhadap suatu permasalahan. Alat ini dikembangkan untuk membantu tim dalam menyusun dan mengelola informasi berdasarkan tingkat kepastian dan ketidakpastian yang ada (George et al., 2022).



Gambar 2. 1 Framework CSD *Matrix*

CSD *Matrix* merupakan kerangka kerja visual yang digunakan untuk mengklasifikasikan informasi dan pengetahuan tim ke dalam tiga kategori utama, yaitu *Certainties*, *Suppositions*, dan *Doubts*. Menurut situs *Createely*, metode ini membantu tim dalam mengelompokkan apa yang sudah diketahui secara pasti (*Certainties*), hipotesis yang perlu divalidasi lebih lanjut (*Suppositions*), dan area yang masih belum diketahui atau menimbulkan pertanyaan (*Doubts*) (*Createely*, 2024). Struktur ini berguna dalam tahap awal proyek seperti perencanaan produk atau riset pengguna, karena memungkinkan tim untuk membuat keputusan berdasarkan informasi yang telah dipetakan secara sistematis.

Sementara itu, situs *Ctad.io* menekankan bahwa *CSD Matrix* membantu organisasi dalam merumuskan ide, mengelola risiko, dan menyusun agenda riset berdasarkan tingkat kepastian atas informasi yang dimiliki. Dengan memvisualisasikan tingkat kejelasan dan keyakinan terhadap berbagai informasi, *CSD Matrix* memungkinkan tim untuk memfokuskan perhatian pada bagian yang paling memerlukan eksplorasi lanjutan, sehingga proses pengembangan dapat berjalan lebih efisien dan terarah (Ctad, 2025).

Menurut Guimarães dan Aela Team (2023), *CSD Matrix* adalah sebuah kerangka kerja partisipatif yang digunakan untuk mengelompokkan informasi berdasarkan tingkat kepastian dalam tiga kategori utama, yaitu *Certainties*, *Suppositions*, dan *Doubts*. Metode ini membantu tim dalam tahap awal proyek (*Discovery*) untuk memetakan pemahaman mereka terhadap konteks, kebutuhan pengguna, dan aspek bisnis yang relevan, serta berfungsi sebagai alat hidup yang terus diperbarui seiring perkembangan proyek guna mengarahkan strategi validasi seperti survei pengguna, wawancara, atau uji coba (Guimaraes & Aela, 2023).

Berikut adalah beberapa keunggulan utama dari penggunaan *CSD Matrix* dalam konteks kerja tim, khususnya dalam pengembangan perangkat lunak:

1. Memfasilitasi Komunikasi Lintas Disiplin

CSD Matrix membantu menjembatani kesenjangan pemahaman antar anggota tim dari berbagai latar belakang dengan memberikan struktur diskusi yang jelas dan terarah.

2. Meningkatkan Keterlibatan Tim

Setiap anggota tim diberikan ruang untuk menyampaikan pandangan mereka berdasarkan tingkat kepastian informasi (*Certainties*, *Suppositions*, *Doubts*), sehingga proses *brainstorming* menjadi lebih inklusif.

3. Mengidentifikasi Risiko Sejak Dini

Dengan memetakan asumsi dan keraguan secara eksplisit, *CSD Matrix* memungkinkan tim untuk mengenali potensi risiko dan ketidakpastian sejak awal proyek.

4. Mendukung Transparansi Pengambilan Keputusan

Semua keputusan yang diambil dapat ditelusuri berdasarkan diskusi yang terdokumentasi dalam matriks, sehingga lebih transparan dan mudah dikaji ulang.

5. Meningkatkan Efektivitas Kolaborasi

Struktur sederhana namun fungsional dari *CSD Matrix* memungkinkan kolaborasi yang lebih terarah, efisien, dan produktif dalam lingkungan kerja yang kompleks.

6. Fleksibel dan Mudah Disesuaikan

CSD *Matrix* dapat diperbarui secara berkala sesuai perkembangan proyek, menjadikannya alat yang adaptif dalam mendukung dinamika tim.

Penggunaan CSD *Matrix* juga dapat menjadi metode dokumentasi yang sistematis terhadap evolusi pemahaman suatu tim terhadap proyek. Seiring berjalannya waktu, kategori dalam matriks dapat berubah misalnya, sebuah *Doubt* bisa berpindah menjadi *Suppositions* atau bahkan menjadi *Certainty* setelah dilakukan validasi atau eksperimen. Oleh karena itu, CSD *Matrix* juga bisa berfungsi sebagai alat reflektif dalam proses iteratif pengembangan. Adapun beberapa hal yang perlu diperhatikan dalam CSD *Matrix* menurut Kaley (2023) dalam website NN Group, yakni sebagai berikut (Kaley, 2023).

a. *Certainties* (Kepastian)

Kolom *Certainties* dalam CSD *Matrix* memuat informasi yang sudah diketahui dengan pasti dan telah tervalidasi melalui riset terpercaya atau data yang kuat. Informasi dalam kategori ini menjadi fondasi utama dalam proses pengambilan keputusan, karena bersifat objektif dan dapat dipertanggungjawabkan. Dengan mengetahui apa yang pasti, tim dapat merancang solusi atau melanjutkan pengembangan dengan kepercayaan diri yang tinggi. Namun demikian, informasi dalam kolom ini harus tetap dimonitor secara berkala karena situasi, preferensi pengguna, dan kebutuhan pasar dapat berubah, sehingga kepastian tersebut bisa saja tidak lagi relevan di masa depan.

b. *Suppositions* (Anggapan)

Kategori *Suppositions* mencakup hipotesis atau asumsi yang logis dan masuk akal, namun belum terbukti secara empiris. Anggapan ini biasanya didasarkan pada pengalaman sebelumnya, wawancara dengan pemangku kepentingan, studi literatur, atau intuisi awal yang muncul dari riset pengguna. Meskipun tidak sepenuhnya pasti, informasi dalam kategori ini cukup kuat untuk dijadikan dasar eksplorasi dan pengujian. Untuk mengonfirmasi validitasnya, metode seperti survei, pengujian A/B, atau *usability testing* dapat digunakan. Jika terbukti benar, informasi dari *Suppositions* dapat dipindahkan ke kolom *Certainties*.

c. *Doubts* (Keraguan)

Kolom *Doubts* berisi pertanyaan terbuka atau area yang belum jelas dan memerlukan eksplorasi lebih lanjut. Informasi dalam kategori ini mencerminkan ketidakpastian dalam proyek yang belum bisa dijawab dengan data yang tersedia. Keraguan ini sering muncul dari diskusi tim atau observasi awal yang belum memiliki landasan penelitian. Untuk mengatasi dan memahami *Doubts*, pendekatan kualitatif seperti wawancara mendalam,

metode etnografi, atau studi lapangan sangat penting dilakukan. Tujuannya adalah mengidentifikasi kebutuhan pengguna yang belum terpenuhi dan secara bertahap mengubah keraguan menjadi anggapan yang dapat diuji lebih lanjut.

2.2 *Software Process Improvement*

Menurut Ashrafi (2003), *Software Process Improvement* (SPI) adalah pendekatan metodologis yang digunakan oleh perusahaan perangkat lunak untuk meningkatkan kualitas produk dan manajemen proses mereka. SPI mencakup penerapan standar seperti ISO 9000-3 dan *Capability Maturity Model* (CMM), yang keduanya bertujuan untuk menilai kapabilitas organisasi dalam menghasilkan perangkat lunak berkualitas tinggi. Meskipun kedua metode ini berbeda dalam pendekatan CMM menitikberatkan pada peningkatan kematangan proses pengembangan perangkat lunak, sementara ISO lebih fokus pada konsistensi dokumentasi dan pencegahan ketidaksesuaian keduanya berkontribusi terhadap upaya perbaikan proses secara sistematis dan bertahap (Ashrafi, 2003).

Ashrafi (2003) juga menekankan bahwa SPI lebih memusatkan perhatian pada proses pengembangan daripada produk itu sendiri. Dalam hal ini, SPI dipandang dari sudut pandang pengembang, dengan asumsi bahwa perbaikan proses akan menghasilkan peningkatan kualitas produk akhir. Hal ini berbeda dengan *Software Quality Assurance* (SQA) yang lebih mencerminkan perspektif pengguna, karena mencakup elemen-elemen pendukung seperti dokumentasi sistem, pelatihan staf, hingga dukungan pasca-instalasi. Dengan demikian, SPI memberikan kerangka kerja generik yang dapat disesuaikan untuk memenuhi kebutuhan bisnis dan tujuan kualitas organisasi (Ashrafi, 2003).

Menurut Washizaki dan Olszewska (2024), *Software Process Improvement* (SPI) merupakan bagian integral dari upaya sistematis untuk meningkatkan kualitas proses perangkat lunak melalui pendekatan terstruktur seperti *Plan-Do-Check-Act* (PDCA). Paradigma ini memungkinkan organisasi untuk secara bertahap merencanakan perbaikan, menerapkannya, mengevaluasi hasilnya, dan melakukan penyesuaian berkelanjutan terhadap proses. Dalam konteks SPI, PDCA menjadi kerangka berpikir untuk memastikan bahwa setiap perbaikan yang dilakukan benar-benar memberikan dampak yang terukur dan konsisten terhadap peningkatan proses. Pengambilan keputusan berbasis bukti seperti dalam pemilihan model pengembangan atau teknik asesmen menjadi kunci sukses pelaksanaan SPI (Washizaki & Olszewska, 2024).

Berdasarkan hal itu Washizaki dan Olszewska (2024), menekankan bahwa SPI biasanya dijalankan melalui pendekatan berbasis kerangka kerja seperti *Capability Maturity Model*

Integration (CMMI) dan *ISO/IEC 33000 series* (SPICE), yang secara eksplisit dirancang untuk menilai dan meningkatkan kapabilitas proses perangkat lunak. Kerangka ini menggunakan *process reference model* dan *assessment model* sebagai alat ukur utama untuk mengevaluasi sejauh mana proses telah memenuhi tujuan tertentu dan mampu menghasilkan keluaran yang diharapkan. Dalam SPI, kerangka kerja ini bukan hanya digunakan untuk mengukur performa saat ini, tetapi juga menjadi panduan dalam merancang langkah-langkah perbaikan jangka panjang yang berorientasi pada peningkatan kualitas proses dan kematangan organisasi secara keseluruhan (Washizaki & Olszewska, 2024).

Menurut Albuquerque *et al.* (2021), banyak organisasi yang akhirnya menghentikan program *Software Process Improvement* (SPI) akibat berbagai hambatan internal yang signifikan. Dalam studi kasus yang dilakukan di delapan perusahaan perangkat lunak di Brasil, ditemukan bahwa faktor-faktor seperti manajemen proyek SPI yang kurang matang, keterbatasan sumber daya manusia yang kompeten, serta budaya organisasi yang tidak mendukung secara langsung mempengaruhi keberlanjutan program peningkatan proses tersebut. Ketika program SPI tidak dikelola dengan baik misalnya, tidak ada kepemimpinan kuat atau perencanaan yang jelas meskipun sebuah organisasi telah melewati tahap asesmen, mereka cenderung kehilangan momentum dan memilih untuk menghentikan inisiatif tersebut (Albuquerque *et al.*, 2021).

Peningkatan kualitas proses perangkat lunak (*Software Process Improvement*/SPI) merupakan elemen penting dalam pengembangan sistem informasi yang berkelanjutan. Metode untuk melakukan SPI telah berkembang seiring kemajuan teknologi dan kompleksitas kebutuhan pengguna. Tujuan utama dari SPI adalah untuk meningkatkan efisiensi proses, mengurangi kesalahan, meningkatkan kualitas produk, dan meningkatkan kepuasan pengguna melalui optimalisasi proses internal organisasi pengembang perangkat lunak.

Beberapa metode yang paling umum digunakan untuk SPI adalah model berbasis kapabilitas dan kematangan seperti *Capability Maturity Model Integration* (CMMI) dan *ISO/IEC 15504* (SPICE). CMMI, misalnya, menyediakan kerangka kerja terstruktur dalam lima level kematangan, yang mencakup mulai dari proses ad-hoc hingga proses yang dioptimalkan. Model ini memungkinkan organisasi untuk menilai posisi mereka saat ini, mengidentifikasi area lemah, dan merencanakan perbaikan jangka panjang. *ISO/IEC 15504* mengadopsi pendekatan serupa, tetapi lebih fleksibel untuk berbagai jenis organisasi dan lebih fokus pada pengukuran kapabilitas proses secara individual (Debou *et al.*, 1994).

Selain model standar tersebut, pendekatan kuantitatif juga mulai banyak digunakan. (Debou et al., 1994), mengusulkan metode *decision making* berbasis data untuk SPI, di mana evaluasi dilakukan melalui pendekatan *pre-post comparison*. Dalam pendekatan ini, performa proyek sebelum dan sesudah intervensi diperbandingkan untuk menilai efektivitas metode yang diterapkan. Parameter yang biasanya diukur meliputi waktu pengembangan, jumlah defect/error, tingkat kepuasan pengguna, serta efektivitas komunikasi dalam tim (Debou et al., 1994).

Penelitian terkini juga memperkenalkan metode inovatif seperti AZ-Model yang dikembangkan oleh (Akbar et al., 2018). Model ini mengintegrasikan pendekatan sistematis untuk perencanaan, pelaksanaan, pengawasan, dan dokumentasi proses pengembangan perangkat lunak. AZ-Model memiliki fokus kuat pada aspek komunikasi dan dokumentasi yang terstruktur, sehingga membantu dalam mengurangi miskomunikasi, meningkatkan keterlacakan, dan mendukung audit proses secara menyeluruh. Model ini sangat sesuai untuk organisasi yang ingin menggabungkan metode *Agile* dengan kerangka dokumentasi formal.

Metode lain yang juga mulai berkembang adalah pendekatan berbasis risiko (*risk-based process improvement*), di mana organisasi memprioritaskan perbaikan pada proses yang paling berdampak terhadap kualitas akhir perangkat lunak. Dengan menganalisis titik rawan kegagalan (seperti miskomunikasi antar tim, kesalahan dalam penerjemahan requirement, atau ketidaksesuaian implementasi dengan desain), tim pengembang dapat melakukan intervensi yang lebih tepat sasaran dan efisien (T. A. Kroeger et al., 2014).

2.3 Kualitas Proses Bisnis

Kualitas proses bisnis mengacu pada tingkat keefektifan dan efisiensi dari suatu proses bisnis dalam mencapai tujuan organisasi secara konsisten dan berkelanjutan. Proses bisnis yang berkualitas ditandai dengan kemampuan untuk menghasilkan *output* sesuai standar yang ditetapkan, tepat waktu, hemat biaya, fleksibel terhadap perubahan, serta mampu memberikan nilai tambah bagi pelanggan dan organisasi. Menurut (Stravinskiene & Serafinas, 2020), kualitas proses bisnis sangat ditentukan oleh penerapan prinsip-prinsip manajemen mutu secara menyeluruh, termasuk komunikasi yang efektif, kolaborasi antar tim, serta evaluasi dan perbaikan berkelanjutan.

Beberapa indikator utama dari kualitas proses bisnis antara lain:

- a. Ketepatan waktu penyelesaian tugas
- b. Kualitas produk atau layanan yang dihasilkan
- c. Efisiensi biaya operasional

- d. Tingkat kepuasan pengguna
- e. Fleksibilitas proses terhadap perubahan
- f. Kekuatan kolaborasi dan komunikasi antar unit kerja.

Komunikasi yang baik memungkinkan pemahaman yang jelas terhadap peran dan tanggung jawab, sedangkan kolaborasi mendorong terciptanya sinergi lintas fungsi. Dalam kerangka *Business Process Improvement* (BPI), indikator-indikator ini menjadi tolok ukur untuk mengidentifikasi area yang perlu ditingkatkan demi tercapainya keunggulan kompetitif dan adaptabilitas jangka panjang (Maulana, 2023).

Kualitas proses dalam pengembangan perangkat lunak tidak hanya ditentukan oleh *output* teknis, tetapi juga dipengaruhi oleh efektivitas komunikasi dalam tim pengembang. Komunikasi yang buruk kerap menjadi penyebab utama kegagalan proyek perangkat lunak, khususnya dalam tahap awal seperti *elicitation* dan *requirement analysis*. Masalah yang umum terjadi antara lain miskomunikasi antar anggota tim, ketidakjelasan informasi antara tim pengembang dan pengguna, serta kesulitan dalam menyampaikan atau memahami kebutuhan sistem (DeFranco & Laplante, 2017).

Untuk mengukur kualitas proses bisnis dalam proyek pengembangan perangkat lunak, dibutuhkan model yang mempertimbangkan berbagai dimensi seperti efisiensi waktu, kejelasan alur kerja, dan ketepatan pemenuhan kebutuhan pengguna. (T. A. Kroeger et al., 2014), mengusulkan model perspektif kualitas proses yang mencakup empat atribut utama: *suitability*, *usability*, *manageability*, dan *evolvability*. Keempat atribut ini dipengaruhi oleh properti proses dan lingkungan organisasi pengembang.

Dalam pendekatan lain (Kroeger & Davidson, 2009), mengusulkan *Perspective-based Model of Quality* untuk proses rekayasa perangkat lunak. Model ini menempatkan peran besar pada keterlibatan pemangku kepentingan dalam mengevaluasi kualitas proses melalui serangkaian metrik dan atribut yang relevan bagi masing-masing perspektif, seperti manajer proyek, pengembang, hingga pengguna akhir. Evaluasi dilakukan melalui pengumpulan data berbasis survei atau wawancara yang dikaitkan dengan tujuan proses tertentu. Pendekatan ini memperkuat metode *Goal-Question-Metric* (GQM) dalam menghubungkan tujuan strategis dengan indikator pengukuran spesifik yang dapat dinilai secara empiris.

Sementara itu, pendekatan modern diperkenalkan oleh (Bubevski, 2014), dalam penelitiannya mengenai *risk-based quality process management* menawarkan sebuah kerangka kerja untuk pengelolaan kualitas proses yang menggabungkan pengukuran risiko dengan penilaian kualitas. Model ini menilai kualitas proses melalui indikator seperti frekuensi

kesalahan yang disebabkan miskomunikasi, keterlambatan akibat ketidakjelasan tugas, dan efisiensi pengambilan keputusan dalam tim. Dengan menggunakan data historis dan pemetaan risiko yang bersifat prediktif, pendekatan ini memberikan nilai tambah karena memungkinkan organisasi untuk menilai tidak hanya apa yang terjadi, tetapi juga mengapa hal itu terjadi dan bagaimana mencegahnya di masa depan.

2.4 ISO/IEC 15504 (SPICE – *Software Process Improvement and Capability Determination*)

ISO/IEC 15504 juga dikenal sebagai SPICE (*Software Process Improvement and Capability Determination*) adalah standar internasional yang ditetapkan untuk menilai dan meningkatkan kapabilitas proses pengembangan perangkat lunak. Standar ini dikembangkan oleh ISO/IEC JTC1/SC7 dan berlandaskan pada model seperti CMM dan ISO/IEC 12207. Tujuannya adalah menyediakan kerangka sistematis untuk mengevaluasi seberapa baik suatu organisasi menjalankan proses perangkat lunak mereka serta memberikan panduan perbaikan berkelanjutan (Mayer & Sechser, 2022).

Struktur utama dari ISO/IEC 15504 terdiri atas dua dimensi yaitu *process dimension* dan *capability dimension*. Dimensi proses mencakup berbagai kategori proses, seperti proses teknis, manajerial, pendukung, dan organisasi, yang merujuk pada standar siklus hidup perangkat lunak ISO/IEC 12207. Sementara itu, dimensi kapabilitas menggambarkan sejauh mana proses tersebut dilakukan secara konsisten dan terukur, yang diklasifikasikan dalam enam level kapabilitas, mulai dari Level 0 (*Incomplete*) hingga Level 5 (*Optimizing*). Setiap atribut diberi skor menggunakan skala pencapaian yaitu *Not*, *Partially*, *Largely*, atau *Fully achieved* (Mayer & Sechser, 2022).

Keunggulan utama dari ISO/IEC 15504 adalah fleksibilitasnya yang memungkinkan organisasi dari berbagai ukuran dan tingkat kematangan untuk menerapkan standar ini sesuai kebutuhan masing-masing. Model ini tidak hanya berfungsi sebagai alat evaluasi, tetapi juga sebagai panduan dalam menyusun rencana peningkatan proses (*process improvement*). Dengan penerapan yang tepat, ISO/IEC 15504 dapat mendukung organisasi dalam mengidentifikasi kelemahan proses, merancang strategi perbaikan, dan membangun budaya mutu berkelanjutan dalam pengembangan sistem perangkat lunak (Methawachananont et al., 2020).

2.5 ISO/IEC 33000 Series

ISO/IEC 33000 Series adalah rangkaian standar internasional yang dirancang untuk menetapkan kerangka kerja dalam melakukan penilaian dan peningkatan proses perangkat lunak. Standar ini merupakan evolusi dari ISO/IEC 15504 (SPICE) dan bertujuan menyediakan struktur yang lebih modern, modular, dan fleksibel. ISO/IEC 33000 Series mencakup berbagai bagian, seperti model referensi proses, model penilaian, persyaratan untuk pelaksanaan asesmen, dan panduan interpretasi hasil penilaian. Standar ini digunakan oleh organisasi untuk mengevaluasi sejauh mana proses pengembangan perangkat lunak telah mencapai kapabilitas tertentu, serta untuk merencanakan langkah perbaikannya (Rodriguez et al., 2021).

Keunggulan utama ISO/IEC 33000 Series terletak pada pendekatan berbasis arsitektur yang memungkinkan pengguna untuk menyesuaikan model penilaian proses sesuai dengan kebutuhan organisasi. Ini mencakup konsep-konsep seperti *Process Reference Model* (PRM) dan *Process Assessment Model* (PAM) yang dapat diadaptasi dari standar lain seperti ISO/IEC 12207 dan ISO/IEC 15288. Selain itu, model ini mendukung berbagai metode asesmen, mulai dari internal *self-assessment* hingga asesmen formal oleh pihak ketiga. Hal ini menjadikan ISO/IEC 33000 dapat diterapkan baik di organisasi kecil maupun besar, lintas sektor industri (Rodriguez et al., 2021).

Penerapan ISO/IEC 33000 Series mendorong organisasi untuk fokus pada peningkatan proses secara berkelanjutan melalui siklus evaluasi, identifikasi kelemahan, dan penguatan kapabilitas proses. Standar ini juga sejalan dengan prinsip-prinsip manajemen mutu dan rekayasa perangkat lunak modern, yang menuntut efisiensi, konsistensi, dan transparansi dalam seluruh siklus hidup produk. Oleh karena itu, ISO/IEC 33000 Series tidak hanya berfungsi sebagai alat ukur kematangan proses, tetapi juga sebagai fondasi strategis dalam membangun daya saing dan keunggulan operasional organisasi di era digital (Verdugo Lara et al., 2021).

2.6 Personal Software Process (PSP)

Personal Software Process (PSP) adalah pendekatan rekayasa perangkat lunak yang berfokus pada peningkatan kinerja individu pengembang. Metode ini dikembangkan oleh Watts S. Humphrey dan bertujuan untuk menciptakan disiplin diri dalam proses pengembangan perangkat lunak melalui pengukuran, perencanaan, dan analisis mandiri. PSP mendorong pengembang untuk secara konsisten mencatat waktu kerja, ukuran produk, jumlah dan jenis kesalahan, serta memperkirakan durasi dan biaya proyek berdasarkan data historis pribadi.

Dengan demikian, PSP membantu individu mengenali pola kerja dan kelemahan yang dapat diperbaiki secara terukur (Contreras et al., 2021).

Struktur PSP terdiri atas beberapa level, mulai dari PSP0 hingga PSP3, yang masing-masing memperkenalkan praktik baru secara bertahap, seperti manajemen waktu, desain modular, inspeksi kode, hingga analisis cacat. Di setiap level, pengembang diajarkan untuk menggunakan skrip kerja, formulir pelaporan, dan log yang terstandarisasi untuk mencatat data proyek secara sistematis. Hal ini bertujuan untuk menciptakan transparansi dalam proses kerja dan memungkinkan perbaikan berkelanjutan berdasarkan data yang terukur. Melalui pendekatan ini, PSP tidak hanya menekankan pada hasil akhir, tetapi juga pada proses yang dilakukan untuk mencapainya (Contreras et al., 2021).

PSP memberikan manfaat yang signifikan bagi pengembang dalam meningkatkan akurasi estimasi, menurunkan jumlah kesalahan, dan meningkatkan efisiensi pengembangan. Dengan menerapkan metode ini secara konsisten, pengembang dapat membentuk kebiasaan kerja yang profesional dan bertanggung jawab atas kualitas hasil kerjanya. Selain itu, PSP juga memberikan fondasi penting bagi penerapan pendekatan tim (TSP) karena data individual yang dikumpulkan menjadi dasar koordinasi dan evaluasi kinerja tim secara keseluruhan. Oleh karena itu, PSP merupakan langkah awal yang penting dalam membangun praktik rekayasa perangkat lunak yang disiplin dan berbasis data (Soto & Rafael, 2020).

2.7 Team Software Process (TSP)

Team Software Process (TSP) adalah metode pengembangan perangkat lunak yang dirancang untuk meningkatkan produktivitas, kualitas, dan efektivitas kerja tim. TSP merupakan kelanjutan dari pendekatan *Personal Software Process* (PSP) dan dikembangkan oleh *Software Engineering Institute* (SEI). Tujuan utama TSP adalah membantu tim perangkat lunak dalam merencanakan, melacak, dan mengelola proyek dengan pendekatan terstruktur, berbasis data, dan kolaboratif. Dalam praktiknya, TSP menyediakan panduan dalam membentuk tim, mendefinisikan peran, mengidentifikasi risiko, serta menyusun rencana kerja dan komitmen tim secara kolektif (Haneef et al., 2024).

TSP memperkenalkan siklus pembangunan tim melalui *launch process*, yaitu fase awal proyek di mana seluruh anggota tim menyepakati tujuan, strategi, serta pembagian tugas. Selain itu, TSP menekankan pentingnya pelaporan berkala, tinjauan tim terhadap kemajuan proyek, dan penggunaan data PSP dari masing-masing anggota tim untuk mengukur performa keseluruhan. Dengan pendekatan ini, TSP mendorong budaya kerja yang transparan,

kolaboratif, dan berorientasi pada kualitas. Tim tidak hanya dinilai dari hasil akhir proyek, tetapi juga dari bagaimana mereka mengelola proses dan menghindari kesalahan sejak tahap perencanaan (Haneef et al., 2024).

Keberhasilan penerapan TSP sangat bergantung pada komitmen tim dan organisasi terhadap prinsip rekayasa perangkat lunak yang disiplin. Manfaat utama dari metode ini adalah meningkatnya akurasi estimasi waktu dan biaya, penurunan jumlah cacat perangkat lunak, serta terciptanya tanggung jawab bersama dalam keberhasilan proyek. Selain itu, TSP memperkuat komunikasi antar anggota tim dan antara tim dengan manajemen, sehingga pengambilan keputusan dapat dilakukan berdasarkan data dan analisis yang objektif. Dengan demikian, TSP menjadi model pengembangan perangkat lunak yang ideal untuk proyek berskala menengah hingga besar yang memerlukan koordinasi dan kendali mutu yang tinggi (Basuki & Angelina, 2023).

2.8 CMMI

Capability Maturity Model Integration (CMMI) adalah sebuah kerangka kerja yang dikembangkan untuk membantu organisasi dalam meningkatkan proses pengembangan produk dan layanan mereka. CMMI mencakup panduan umum dan model yang lintas-disiplin, mencakup seluruh siklus produk atau layanan mulai dari pengembangan, pengiriman hingga pemeliharaan (Gavriel, 2023). Model ini diciptakan sebagai penyempurnaan dari berbagai model sebelumnya seperti SW-CMM (*Software*), SECM (*System Engineering*), dan IPD-CMM (*Integrated Product Development*). Tujuan utama CMMI adalah untuk menyediakan satu model terpadu yang tidak hanya berfokus pada pengembangan produk, tetapi juga pada pemeliharaan dan pengelolaan proses yang mendukung peningkatan berkelanjutan (Ruzhnikov & Prasetyo, 2024).

Menurut (Gavriel, 2023), CMMI dibagi menjadi dua jenis representasi yaitu *staged* dan *continuous*. Representasi *staged* membagi organisasi ke dalam lima tingkat kematangan (*Maturity Levels*), dari *Initial* (ML1) hingga *Optimizing* (ML5), dengan setiap tingkatnya mencerminkan evolusi peningkatan proses. Sementara itu, representasi *continuous* memungkinkan penilaian dan pengembangan setiap area proses (*process area*) secara individual dengan enam tingkat kapabilitas (*capability levels*), dari *Incomplete* (CL0) hingga *Optimizing* (CL5). Model ini mengandalkan pencapaian spesifik dan praktik generik yang diterapkan untuk menilai seberapa matang suatu organisasi dalam menjalankan proses pengembangan perangkat lunak. Pilihan antara representasi *staged* atau *continuous* disesuaikan dengan tujuan dan kondisi organisasi yang mengimplementasikannya (Gavriel, 2023).

Model *staged* mengklasifikasikan kematangan organisasi ke dalam lima tingkatan (*Maturity Levels*) berdasarkan pada (Gavriel, 2023) dan rujukan pada jurnal (Khraiwesh, 2013):

1. Level 1 *Initial* (Tidak Terstruktur)

Proses bersifat ad-hoc, tidak terdokumentasi, dan cenderung reaktif. Keberhasilan proyek sangat bergantung pada kemampuan individu, bukan pada proses yang telah distandarkan. Tidak ada praktik atau kontrol proses yang konsisten di seluruh organisasi.

2. Level 2 *Managed* (Terkelola)

Proses telah direncanakan, dilaksanakan, dipantau, dan dikendalikan pada tingkat proyek. Organisasi mulai menerapkan praktik manajemen dasar seperti *Project Planning*, *Requirements Management*, *Project Monitoring and Control*, serta *Process and Product Quality Assurance*. Fokus utama pada pengelolaan proyek secara terstruktur untuk mencapai tujuan yang telah ditentukan.

3. Level 3 *Defined* (Terdefinisi)

Proses telah didokumentasikan, distandarisasi, dan diintegrasikan ke dalam proses organisasi. Semua proyek menggunakan proses yang disesuaikan dari standar organisasi. Area proses di level ini mencakup *Risk Management*, *Integrated Project Management*, *Organizational Process Definition*, *Validation*, *Verification*, hingga *Requirements Development*.

4. Level 4 *Quantitatively Managed* (Dikelola Secara Kuantitatif)

Proses diukur dan dikendalikan menggunakan data serta metrik kuantitatif. Organisasi memanfaatkan analisis statistik untuk memastikan stabilitas, prediktabilitas, dan konsistensi kinerja proses. Area proses yang menonjol di level ini adalah *Organizational Process Performance* dan *Quantitative Project Management*.

5. Level 5 *Optimizing* (Pengoptimalan Berkelanjutan)

Fokus pada peningkatan berkelanjutan proses melalui inovasi dan analisis akar penyebab masalah (*Causal Analysis and Resolution*). Organisasi mengadopsi teknologi baru, metode inovatif, serta *Organizational Innovation and Deployment* untuk terus meningkatkan kapabilitas dan efisiensi proses.

Pemilihan model *Capability Maturity Model Integration* (CMMI) dibandingkan model lainnya seperti ISO/IEC 15504, ISO/IEC 33000 Series, PSP, atau TSP didasarkan pada fleksibilitas, cakupan luas, dan kematangan struktur yang ditawarkan oleh CMMI dalam peningkatan proses organisasi. CMMI tidak hanya menyediakan kerangka kerja yang teruji

untuk pengembangan perangkat lunak, tetapi juga mencakup manajemen proyek, manajemen risiko, dan rekayasa sistem secara terpadu, sehingga lebih relevan untuk organisasi yang membutuhkan pendekatan holistik dan lintas-disiplin. Selain itu, CMMI memiliki dua pendekatan penilaian, yaitu staged dan continuous representation, yang memungkinkan organisasi menyesuaikan penerapannya sesuai dengan kondisi dan tujuan strategis masing-masing. Dukungan dokumentasi, sumber daya pelatihan, serta adopsi global yang luas juga menjadi alasan kuat mengapa CMMI lebih dipilih sebagai model acuan dalam penelitian atau implementasi peningkatan proses perangkat lunak.

2.9 Penelitian Terdahulu

2.9.1 Metode Pencarian Pustaka

Metode pencarian pustaka adalah langkah penelitian untuk mencari dan mengkaji sumber pustaka yang relevan. Ini mencakup identifikasi, seleksi, dan peninjauan literatur yang mendukung topik penelitian, memastikan kelengkapan data, dan memberikan dasar teoritis yang kuat. Penelitian kepustakaan atau *library research* adalah penelitian yang dilakukan dengan mengumpulkan data atau karya tulis ilmiah terkait objek penelitian dari bahan pustaka. Studi kepustakaan bertujuan untuk mengatasi masalah melalui penelaahan kritis dan mendalam terhadap bahan pustaka yang relevan.

Tujuan utamanya adalah menelusuri literatur tidak hanya untuk mendapatkan informasi penelitian sejenis, memperdalam kajian teoritis, atau mempertajam metodologi, tetapi juga memanfaatkan sumber perpustakaan untuk memperoleh data penelitian. Studi pustaka merupakan tahap penting dalam melakukan studi pendahuluan agar lebih memahami gejala baru yang berkembang di masyarakat. Data pustaka tetap relevan dan andal untuk menjawab penelitian. Dalam riset pustaka, penelusuran literatur tidak hanya menjadi langkah awal menyiapkan kerangka penelitian (*research design*) tetapi juga untuk memanfaatkan sumber-sumber perpustakaan sebagai data penelitian.

Tabel 2. 1 Penelitian Terdahulu

No	Sitasi	Judul	Rumusan Masalah	Tujuan	Metode	Kesimpulan
1.	(Omol et al., 2025)	<i>Digital Maturity Assessment Model (DMAM): assimilation of</i>	Bagaimana menilai kematangan digital UMKM secara efektif	Tujuan dari penelitian ini adalah untuk merancang dan mengembangkan	Penelitian ini menggunakan pendekatan	DMAM berhasil dikembangkan sebagai model yang relevan dan aplikatif untuk menilai

No	Sitasi	Judul	Rumusan Masalah	Tujuan	Metode	Kesimpulan
		<i>Design Science Research (DSR) and Capability Maturity Model Integration (CMMI)</i>	dengan mempertimbangkan keterbatasan sumber daya yang dimiliki?	<i>Digital Maturity Assessment Model (DMAM)</i> yang mampu menilai tingkat kematangan digital UMKM.	<i>Design Science Research (DSR).</i>	kematangan digital UMKM, dengan keunggulan dalam memahami ketergantungan sumber daya dan kebutuhan spesifik UMKM. Model ini tidak hanya mengisi kekosongan dalam literatur tetapi juga memberikan kontribusi nyata bagi strategi transformasi digital yang efektif pada sektor UMKM.
2.	(Wijaya et al., 2025)	<i>Application of RACI Framework for Optimization of Key Performance Indicators (KPI) in Manufacturing Companies</i>	Bagaimana merumuskan indikator kinerja utama (KPI) yang efektif dengan pendekatan RACI di PT XYZ Batam?	Mengembangkan KPI yang relevan dan terukur berdasarkan peran RACI untuk meningkatkan evaluasi kinerja di PT XYZ Batam.	Penelitian dilakukan melalui survei dan wawancara dengan stakeholder perusahaan.	Penerapan RACI dalam perumusan KPI meningkatkan akuntabilitas, kolaborasi antar departemen, dan efektivitas pemantauan kinerja perusahaan secara menyeluruh.
3.	(Aljedaibi & Alsulami, 2021)	<i>A CMMI-Based Method for Software Development Process Assessment: Applying CMMI Process in Saudi Arabia</i>	Mengapa tingkat adopsi praktik CMMI masih rendah di perusahaan pengembang perangkat lunak di Arab Saudi?	Menilai tingkat kesadaran dan penerapan praktik CMMI serta mengidentifikasi area proses yang perlu ditingkatkan untuk meningkatkan kualitas produk perangkat lunak.	Penelitian dilakukan melalui wawancara dan observasi.	Kesadaran dan pemahaman terhadap CMMI masih rendah, meskipun sebagian perusahaan menggunakan bentuk program CMMI. Penerapan penuh CMMI diperlukan untuk meningkatkan kualitas produk, terutama di area proses seperti PMC, MA, PPQA, dan

No	Sitasi	Judul	Rumusan Masalah	Tujuan	Metode	Kesimpulan
						CM.
4.	(Santos et al., 2023)	<i>A Comparative Analysis Of Agile Teamwork Quality Instruments In Agile Software Development: A Qualitative Approach</i>	Bagaimana hubungan antara instrumen pengukuran <i>Teamwork Quality</i> (TWQ) dengan faktor-faktor dalam literatur <i>Agile Software Development</i> (ASD)?	Memahami keterkaitan antara instrumen TWQ yang digunakan dalam konteks ASD dengan faktor-faktor yang ada dalam literatur ASD.	Model instrumen TWQ untuk ASD.	Terdapat korelasi antara frekuensi faktor dalam instrumen TWQ dan tema dari literatur ASD, terutama pada aspek <i>Team Orientation</i> dan <i>Coordination</i> . Studi lanjutan dengan instrumen lain diperlukan untuk memperkuat temuan ini.
5.	(Yassien, 2020)	<i>The Challenges of Capability Maturity Model Integration Application In The Dynamic environment</i>	Bagaimana mengatasi keterbatasan CMMI-DEV v1.3 dalam menghadapi dinamika lingkungan yang cepat berubah?	Mengembangkan model terintegrasi antara CMMI-DEV v1.3 dan <i>Dynamic Capabilities</i> (DC) untuk meningkatkan ketahanan dan nilai ekonomi organisasi perangkat lunak.	Model dikembangkan melalui tinjauan literatur.	Model integrasi CMMI-DEV v1.3 dan DC berhasil mengatasi kekurangan model sebelumnya dan dinilai relevan untuk menghadapi tantangan lingkungan bisnis yang dinamis.
6.	(Rajab & Alnoukari, 2025)	<i>DevOps Integration With Capability Model Maturity Integration: A Systematic Mapping Review</i>	Bagaimana integrasi CMMI dan DevOps dapat dilakukan untuk meningkatkan proses pengembangan perangkat lunak?	Mengkaji literatur yang ada terkait integrasi CMMI dan DevOps serta mengidentifikasi peluang, tantangan, dan manfaat dari integrasi tersebut.	Menggunakan <i>systematic mapping review</i> dengan penelusuran database dan seleksi studi berdasarkan kriteria inklusi dan eksklusi, menghasilkan 24 studi	Integrasi CMMI dan DevOps memiliki potensi meningkatkan kualitas proses pengembangan perangkat lunak, namun belum ada model integrasi yang utuh sehingga diperlukan penelitian lanjutan.

No	Sitasi	Judul	Rumusan Masalah	Tujuan	Metode	Kesimpulan
					relevan.	
7.	(Ajayi, 2021)	<i>A Comparative Analysis of Agile Methods and the Capacity Maturity Model Integration (CMMI)</i>	Apakah <i>Agile</i> dan CMMI dapat digunakan secara bersamaan untuk meningkatkan kualitas dalam pengembangan sistem?	Menganalisis komponen utama, pandangan terhadap kualitas, dan kontribusi metode <i>Agile</i> dan CMMI terhadap pengendalian kualitas proyek.	Studi analisis literatur dengan membandingkan pendekatan <i>Agile</i> dan CMMI.	<i>Agile</i> dan CMMI memiliki kontribusi yang berbeda namun saling melengkapi dalam pengendalian kualitas, dan dapat diterapkan bersama dalam satu proyek untuk mencapai hasil yang optimal.
8.	(Aguilar & Li, 2023)	<i>Responsibility Assignment Matrix in Responding to Critical Outages: A Case Study of IT Incident Management Activity Process Design</i>	Bagaimana cara memperbaiki proses manajemen insiden TI untuk meminimalkan waktu pemulihan (MTTR) akibat gangguan sistem kritis yang tidak terduga?	Merancang model bisnis untuk meningkatkan proses manajemen insiden TI dengan memperjelas peran dan tanggung jawab tiap pemangku kepentingan.	Model RACI dan penerapan <i>Responsibility Assignment Matrix (RAM)</i>	Model RACI dan RAM yang diintegrasikan dalam BPMN mampu memperjelas akuntabilitas tiap fase manajemen insiden, sehingga mendukung peningkatan efisiensi pemulihan insiden TI dan menjaga kepuasan serta retensi pelanggan.
9.	(Castro & Peña, 2023)	<i>Integration of The Software Engineering Theory And The Quality Function Deployment Methodology For Supporting The Measurement of the State of Progress in Software</i>	Bagaimana menyediakan pendekatan pengukuran kemajuan proyek perangkat lunak yang terstandarisasi, mudah diterapkan, dan dapat diandalkan oleh	Mengusulkan kerangka kerja pengukuran proyek perangkat lunak yang mengintegrasikan prinsip manajemen proyek dengan dasar rekayasa perangkat lunak menggunakan elemen SEMAT	Pengembangan pendekatan berbasis SEMAT dan QFD	Kerangka yang diusulkan mampu mendukung pengukuran status kemajuan proyek perangkat lunak secara standar dan andal, sekaligus menjembatani kesenjangan antara perspektif manajemen proyek dan praktik rekayasa perangkat lunak.

No	Sitasi	Judul	Rumusan Masalah	Tujuan	Metode	Kesimpulan
		<i>Projects</i>	profesional pengembang perangkat lunak?	dan metode QFD.		
10.	(Ojha, 2022)	<i>Role of Capability Maturity Model Integration (CMMI) in Software Process Improvement</i>	Mengapa perusahaan perangkat lunak masih enggan mengadopsi model peningkatan proses seperti CMMI?	Mengidentifikasi area proses utama dan menyusun strategi implementasi untuk meningkatkan kematangan proses pengembangan perangkat lunak melalui model CMMI.	Studi literatur dan analisis model-model berbasis CMMI yang digunakan dalam praktik pengembangan perangkat lunak.	Penerapan CMMI membantu manajer proyek dalam mengontrol kualitas perangkat lunak dan meningkatkan efisiensi proses, meskipun masih ada tantangan dalam adopsinya karena struktur yang kompleks.
11.	(Dingsøyr et al., 2024)	<i>Challenges in Understanding the Relationship between Teamwork Quality and Project Success in Large Scale Agile Projects</i>	Apakah kualitas kerja tim (teamwork quality) secara konsisten memengaruhi keberhasilan proyek dalam pengembangan perangkat lunak Agile skala besar sebagaimana yang ditemukan dalam proyek-proyek kecil?	Menyelidiki hubungan antara kualitas kerja tim dan keberhasilan proyek dalam konteks pengembangan Agile berskala besar, serta menguji kembali model teoretis sebelumnya yang dikembangkan untuk tim tunggal.	Survei terhadap 196 partisipan dari 34 tim dalam empat proyek besar, sebagai replikasi dari studi sebelumnya yang dilakukan pada satu tim.	Hasil penelitian menunjukkan bahwa hubungan antara kualitas kerja tim dan keberhasilan proyek tidak sesuai dengan model teoretis sebelumnya. Faktor-faktor seperti perbedaan persepsi terhadap kesuksesan, perubahan makna konsep teamwork, dan pengaruh kuat antar tim (interteam) dibandingkan dalam tim (intrateam) menjadi kemungkinan penyebab. Studi ini menekankan pentingnya penelitian lanjutan yang juga

No	Sitasi	Judul	Rumusan Masalah	Tujuan	Metode	Kesimpulan
						memperhatikan kualitas dan frekuensi interaksi antar tim dalam konteks pengembangan <i>Agile</i> skala besar.
12.	(Strode et al., 2022)	<i>A Teamwork Effectiveness Model for Agile Software Development</i>	Bagaimana menciptakan model efektivitas kerja tim yang sesuai dengan karakteristik tim <i>Agile</i> yang berkolaborasi secara intensif dan lintas fungsi?	Mengembangkan model efektivitas kerja tim khusus untuk tim <i>Agile</i> yang terkolokasi, berdasarkan bukti empiris dan literatur yang relevan.	Studi kasus dan Tinjauan Literatur	Model <i>Agile</i> Teamwork Effectiveness Model (ATEM) terdiri dari lima komponen utama shared leadership, team orientation, redundancy, adaptability, dan peer <i>feedback</i> yang didukung oleh mekanisme koordinasi berupa shared mental models, communication, dan mutual trust. Model ini relevan untuk berbagai pihak di komunitas <i>Agile</i> , dan berpotensi dikembangkan lebih lanjut untuk konteks tim kecil, multi-tim, terdistribusi, atau proyek yang bersifat kritis terhadap keselamatan.

Berdasarkan Tabel 2.1 merangkum berbagai studi terdahulu yang relevan dalam konteks pengembangan perangkat lunak, khususnya terkait peningkatan kualitas proses bisnis melalui pendekatan model evaluasi seperti CMMI dan praktik kerja tim dalam lingkungan *Agile*. Beberapa penelitian seperti yang dilakukan oleh Omol et al. (2025) dan Yassien (2020)

menunjukkan pentingnya kerangka evaluasi proses yang adaptif dalam lingkungan dinamis. Sementara itu, Rajab & Alnoukari (2025) serta Aljedaibi & Alsulami (2021) menyoroti integrasi CMMI dalam praktik DevOps dan rendahnya tingkat adopsi model ini akibat kompleksitas implementasi. Penelitian Ajayi (2021) secara khusus menyoroti bahwa pendekatan *Agile* dan CMMI dapat saling melengkapi, yang mendukung gagasan bahwa metode partisipatif seperti *CSD Matrix* dapat disandingkan dengan kerangka formal seperti CMMI untuk menciptakan proses pengembangan perangkat lunak yang lebih efisien dan terukur.

Penelitian-penelitian tersebut menjadi landasan penting bagi penelitian ini yang menggabungkan pendekatan partisipatif (*CSD Matrix*) dalam proses diskusi dan pengambilan keputusan tim dengan model evaluatif (CMMI) untuk menilai dampak dari pendekatan tersebut terhadap kualitas proses bisnis. Meskipun beberapa studi sebelumnya telah membahas efektivitas kerja tim dan pengaruh struktur akuntabilitas seperti RACI (Wijaya et al., 2025), belum ditemukan penelitian yang secara eksplisit mengintegrasikan *CSD Matrix* sebagai alat bantu visual dalam konteks pengukuran formal seperti CMMI. Oleh karena itu, penelitian ini memiliki kontribusi baru dengan menunjukkan bagaimana penerapan *CSD Matrix* dapat membantu tim pengembang dalam mendokumentasikan keputusan, meningkatkan kolaborasi, dan secara nyata memengaruhi tingkat kematangan proses bisnis menurut indikator CMMI, khususnya dalam studi kasus di perusahaan teknologi Mamikos.

Tabel 2. 2 Perbandingan Metode dalam Evaluasi dan Peningkatan Proses Bisnis Tim
Pengembang Perangkat Lunak

Metode	Fokus Utama	Kelebihan	Kekurangan
<i>CSD Matrix</i>	Identifikasi kepastian, asumsi, dan keraguan dalam tim	Sederhana, praktis, membangun persepsi bersama sejak awal proyek	Tidak menyusun struktur proses formal atau penugasan tanggung jawab terperinci
CMMI (<i>Aljedaibi & Alsulami, 2021; Ojha, 2022</i>)	Evaluasi kematangan proses pengembangan perangkat lunak	Terstruktur, digunakan secara global, indikator jelas dalam perbaikan proses	Kompleks, membutuhkan waktu dan komitmen tinggi dalam implementasi

Metode	Fokus Utama	Kelebihan	Kekurangan
RACI Matrix (Wijaya et al., 2025; Aguilar & Li, 2023)	Penugasan tanggung jawab dan akuntabilitas dalam proses organisasi	Jelas dalam distribusi peran, meningkatkan akuntabilitas antar stakeholder	Tidak menangkap komunikasi informal atau persepsi awal antar anggota tim
QFD–SEMAT Framework (Castro & Peña, 2023)	Pengukuran kemajuan proyek berbasis kebutuhan pengguna dan manajemen proyek	Memberi pelacakan progres proyek berbasis kebutuhan	Tidak menggali komunikasi tim atau asumsi awal dalam tahap perencanaan
DevOps–CMMI Integration (Rajab & Alnoukari, 2025)	Sinkronisasi antara praktik DevOps dan evaluasi kematangan proses	Meningkatkan kecepatan dan kualitas pengembangan perangkat lunak	Belum tersedia model integrasi utuh, perlu penelitian dan validasi lanjutan
Dynamic Capabilities CMMI (Yassien, 2020)	Ketahanan organisasi dalam lingkungan dinamis melalui integrasi model	Adaptif terhadap perubahan lingkungan bisnis	Belum terbukti praktis dalam tim kecil atau konteks <i>Agile</i> ringan
ATEM (Agile Teamwork Effectiveness Model) (Strode et al., 2022)	Efektivitas kerja tim dalam pengembangan <i>Agile</i>	Menekankan kepercayaan, <i>feedback</i> , koordinasi, dan kepemimpinan bersama	Tidak disertai alat implementatif praktis yang mudah digunakan oleh tim
TWQ Instruments (Santos et al., 2023)	Pengukuran kualitas kerja tim dalam konteks <i>Agile Software Development</i>	Menyediakan indikator kualitatif teamwork seperti koordinasi dan orientasi tim	Butuh validasi instrumen lebih lanjut untuk konteks dan budaya tim yang berbeda
DSM (Design Structure Matrix) (Zhang & Jia, 2011)	Visualisasi dependensi antar elemen proses pengembangan perangkat lunak	Mengurangi pekerjaan ulang, sangat informatif	Memerlukan data rinci dan waktu persiapan, kurang cocok untuk proyek kecil atau cepat

Metode	Fokus Utama	Kelebihan	Kekurangan
		untuk sistem kompleks	
ISM/MMM (Sabzian et al., 2019)	Pemodelan struktur komunikasi sistemik formal dan informal	Dokumentasi menyeluruh, cocok untuk proyek sistem rekayasa besar	Kompleks dan tidak mudah diterapkan dalam proyek <i>Agile</i> bersifat iteratif dan cepat

Metode *CSD Matrix*, yang menjadi fokus utama dalam penelitian ini, berfungsi untuk mengidentifikasi *Certainties* (kepastian), *Suppositions* (asumsi), dan *Doubts* (keraguan) dalam tim pengembang perangkat lunak. Untuk mengukur sejauh mana metode ini berkontribusi terhadap peningkatan kualitas proses pengembangan, digunakan kerangka evaluasi *Capability Maturity Model Integration* (CMMI) sebagai alat pengukurnya. *CSD Matrix* cocok diterapkan pada fase *brainstorming* karena mampu menciptakan ruang terbuka untuk menyampaikan kekhawatiran dan menyusun usulan yang dapat dipertanggungjawabkan, terutama dalam lingkungan dinamis seperti *startup* Mamikos. Kombinasi pendekatan partisipatif ini dengan evaluasi terstruktur seperti CMMI memungkinkan organisasi mendorong perbaikan berkelanjutan dan pengambilan keputusan yang lebih berkualitas dalam pengembangan perangkat lunak.

BAB III METODOLOGI

3.1 Metode Penelitian

Penelitian ini menggunakan pendekatan *mixed methods* dengan desain eksplanatori sekuensial, yaitu menggabungkan pendekatan kualitatif dan kuantitatif secara berurutan (Rukhmana et al., 2022). Pendekatan ini dipilih karena penelitian tidak hanya ingin memahami proses dan dinamika penggunaan CSD *Matrix* dalam tim pengembang perangkat lunak secara mendalam, tetapi juga ingin mengukur dampaknya secara terstruktur terhadap kualitas proses bisnis. Studi kualitatif dilakukan terlebih dahulu melalui observasi, wawancara, dan dokumentasi untuk menggali pengalaman serta persepsi anggota tim selama penerapan CSD *Matrix*.

Setelah itu, data kuantitatif dikumpulkan melalui kuesioner *pre-test* dan *post-test* yang disusun berdasarkan indikator *Capability Maturity Model Integration* (CMMI). Data ini digunakan untuk menilai perubahan tingkat kematangan proses bisnis tim, seperti peningkatan dari level *Managed* ke *Defined*. Gabungan dua pendekatan ini memberikan gambaran yang komprehensif, kualitatif menjelaskan bagaimana dan mengapa proses terjadi, sementara kuantitatif menunjukkan seberapa besar dampaknya secara terukur.

3.2 Subjek dan Objek Penelitian

1. Subjek Penelitian

Subjek penelitian adalah individu atau kelompok yang menjadi sumber informasi utama dan berperan langsung dalam memberikan data terhadap fenomena yang sedang diteliti (Rukhmana et al., 2022). Dalam penelitian kualitatif, subjek penelitian biasanya disebut sebagai informan, yaitu pihak yang memiliki pengalaman, pengetahuan, atau keterlibatan langsung dalam situasi atau proses yang menjadi fokus kajian. Subjek ini dipilih berdasarkan kriteria relevansi dan keterlibatannya terhadap permasalahan yang dikaji agar informasi yang diperoleh valid, kaya, dan kontekstual (Abdussamad, 2022).

Pada penelitian ini, subjek penelitian terdiri dari 30 anggota tim *Engineering* di Mamikos yang terlibat secara aktif dalam proyek pengembangan perangkat lunak, terdiri dari berbagai peran strategis, yaitu 1 *Engineering Manager* (EM) *Backend*, 1 EM *Frontend*, 1 EM *Mobile*, 1 EM *DevOps*, 1 QAE *Lead*, 6 *Backend Engineer*, 5 *Frontend*

Engineer, 2 *Android Engineer*, 2 *iOS Engineer*, 3 *DevOps Engineer*, 5 *QA Engineer*, 4 *Data Scientist*, dan 1 *Data Engineer*. Subjek-subjek ini dipilih karena memiliki peran langsung dalam pelaksanaan proses kerja dan penerapan metode *CSD Matrix* dalam aktivitas tim. Keragaman peran dalam tim memungkinkan diperolehnya pandangan yang menyeluruh dari berbagai aspek teknis, manajerial, hingga pengujian dan data.

Untuk memastikan bahwa informasi yang dikumpulkan relevan dan mendalam, subjek penelitian ditentukan berdasarkan kriteria berikut:

1. Merupakan anggota tim *Engineering* aktif di Mamikos yang terlibat dalam pengembangan perangkat lunak.
2. Berperan langsung dalam proses kerja teknis atau manajerial, seperti pengembangan *backend/frontend/mobile*, *DevOps*, *QA (Quality Assurance)*, hingga data *Engineering* dan *data science*.
3. Terlibat dalam proyek pengembangan perangkat lunak yang menjadi konteks studi (khususnya proyek “*Apple Music Affiliate Landing page in Mamikos Web*”).
4. Bersedia berpartisipasi secara aktif dalam proses pengumpulan data seperti wawancara, observasi, atau pengisian kuesioner.

Alasan pemilihan 30 subjek ini adalah untuk memperoleh gambaran yang komprehensif dan representatif terhadap proses bisnis dalam tim pengembang. Setiap peran memiliki tanggung jawab spesifik yang dapat memengaruhi dinamika komunikasi, kolaborasi, dan pengambilan keputusan. Dengan melibatkan berbagai peran mulai dari level manajerial hingga teknis, penelitian ini diharapkan mampu menangkap persepsi yang beragam terhadap efektivitas metode *CSD Matrix*, serta mengidentifikasi kendala atau potensi peningkatan proses yang selama ini mungkin tidak terungkap hanya dari satu perspektif saja.

2. Objek Penelitian

Objek dalam penelitian ini adalah penerapan metode *CSD Matrix* dalam proses bisnis tim pengembang perangkat lunak di Mamikos serta evaluasi tingkat kematangan prosesnya menggunakan kerangka *Capability Maturity Model Integration (CMMI)*. Fokus diarahkan pada bagaimana *CSD Matrix* digunakan dalam aktivitas *brainstorming* dan pengambilan keputusan tim, serta kontribusinya terhadap peningkatan kolaborasi, komunikasi, dan kualitas kerja. Objek ini dipilih karena relevan dengan kebutuhan industri *startup* yang bergerak cepat, di mana struktur diskusi yang

jas dan pengukuran proses yang sistematis sangat diperlukan untuk mendukung efisiensi dan keberlanjutan tim pengembang.

3.3 Teknik Pengumpulan Data

1. Observasi

Observasi adalah teknik pengumpulan data yang dilakukan dengan cara mengamati secara langsung objek atau fenomena yang menjadi fokus penelitian. Dalam penelitian kualitatif, observasi memungkinkan peneliti untuk memahami proses, perilaku, dan interaksi dalam konteks alami tanpa intervensi yang berlebihan. Observasi dapat bersifat partisipatif maupun non-partisipatif, tergantung tingkat keterlibatan peneliti dalam lingkungan yang diamati (Abdussamad, 2022). Teknik ini digunakan untuk memperoleh data nyata dari situasi aktual sehingga dapat memperkuat validitas hasil penelitian.

Pada penelitian ini, observasi dilakukan dengan menelusuri dan menganalisis webpage resmi Mamikos, terutama fitur-fitur yang berkaitan dengan proyek pengembangan “*Apple Music Affiliate Landing page in Mamikos Web.*” Peneliti berperan sebagai fasilitator yang memperkenalkan dan menjelaskan cara penggunaan CSD *Matrix* kepada tim, namun tidak terlibat langsung dalam pertemuan internal atau proses pengisian matriks yang bersifat privat. Informasi observasional diperoleh melalui pemantauan tidak langsung, seperti hasil isian CSD *Matrix* yang dibagikan tim, catatan perubahan isi matriks, serta dokumentasi progres aktivitas *brainstorming*. Observasi ini digunakan untuk menangkap pola umum dalam penggunaan CSD *Matrix* dan mendukung triangulasi data dari wawancara dan kuesioner.

2. Wawancara

Wawancara adalah metode pengumpulan data yang dilakukan melalui interaksi langsung antara peneliti dan informan untuk menggali informasi secara mendalam. Teknik ini memungkinkan peneliti mendapatkan data kualitatif yang lebih detail, termasuk opini, persepsi, pengalaman, dan alasan di balik tindakan yang dilakukan informan. Wawancara dapat bersifat terstruktur, semi-terstruktur, atau tidak terstruktur tergantung kebutuhan data dan fleksibilitas eksplorasi topik yang dibahas.

Dalam penelitian ini, wawancara dilakukan kepada *Senior Engineering Manager* Mamikos yaitu Giga Prakosa Hikmata, sebagai informan utama yang memiliki tanggung jawab strategis dalam pengelolaan tim pengembang. Pertanyaan wawancara

disusun berdasarkan indikator *Key Process Area* (KPA) untuk mengukur kinerja proses dalam pengembangan perangkat lunak. Fokus wawancara meliputi bagaimana proses kerja berlangsung sebelum dan sesudah penerapan *CSD Matrix*, hambatan yang dihadapi, serta pandangannya terhadap kolaborasi, dokumentasi, dan efektivitas tim dalam menjalankan proyek.

3. Kuesioner

Kuesioner adalah instrumen pengumpulan data yang berbentuk serangkaian pertanyaan tertulis yang diajukan kepada responden untuk mendapatkan informasi secara sistematis. Teknik ini digunakan untuk menjangkau lebih banyak partisipan dalam waktu relatif singkat, dan cocok untuk mengukur persepsi, penilaian, dan tanggapan individu terhadap suatu topik. Kuesioner dapat berupa pertanyaan terbuka atau tertutup, serta menggunakan skala tertentu seperti skala Likert untuk mempermudah analisis.

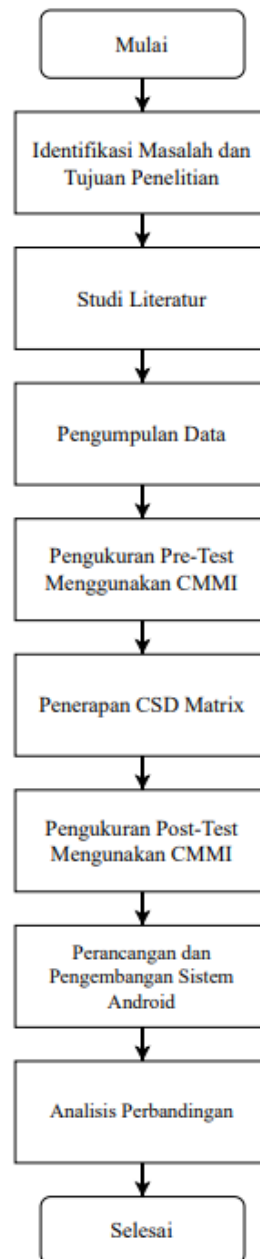
Dalam penelitian ini, kuesioner diberikan kepada 30 anggota tim *engineer* Mamikos dalam dua tahap, yaitu sebelum dan sesudah penerapan *CSD Matrix*. Tujuan kuesioner adalah untuk mengevaluasi dampak penggunaan *CSD Matrix* terhadap kualitas proses bisnis dalam tim. Instrumen kuesioner disusun berdasarkan indikator *Key Process Area* (KPA) dalam model CMMI. Perbandingan hasil *pre-test* dan *post-test* dari kuesioner ini akan memberikan gambaran tentang sejauh mana metode *CSD Matrix* berkontribusi terhadap peningkatan proses pengembangan perangkat lunak di Mamikos.

4. Dokumentasi

Dokumentasi merupakan teknik pengumpulan data dengan cara mengumpulkan dan menganalisis dokumen tertulis, gambar, catatan, atau arsip yang relevan dengan objek penelitian. Dalam penelitian ini, dokumentasi mencakup catatan aktivitas *brainstorming* menggunakan *CSD Matrix*, struktur tim, hasil retrospektif proyek, serta laporan internal yang dikelola oleh tim pengembang. Dokumentasi digunakan untuk memperkuat temuan dari wawancara dan observasi serta menjadi bukti tertulis atas proses kerja yang berlangsung selama proyek pengembangan berlangsung.

3.4 Alur Penelitian

Berikut merupakan alur penelitian yang akan dilakukan dalam penelitian Adopsi CSD *Matrix* Untuk Meningkatkan Kualitas Proses Bisnis Tim Pengembang Perangkat Lunak (Studi Kasus Mamikos).



Gambar 3. 1 Alur Penelitian

Berdasarkan diagram alur tersebut, berikut adalah penjelasan dari setiap tahapan penelitian yang dilakukan secara sistematis mulai dari identifikasi masalah hingga evaluasi hasil menggunakan model CMMI.

1. Identifikasi Masalah dan Tujuan Penelitian

Tahap pertama dalam penelitian ini adalah identifikasi masalah, yaitu proses untuk menemukan isu atau permasalahan nyata yang terjadi dalam konteks pengembangan perangkat lunak di Mamikos. Masalah yang diangkat berkaitan dengan belum optimalnya proses pengembangan, khususnya dalam hal kolaborasi, dokumentasi keputusan, serta penanganan ketidakpastian selama proses pengembangan. Melalui hasil pengamatan awal dan studi literatur, ditemukan bahwa tantangan seperti miskomunikasi, pengambilan keputusan yang tidak terdokumentasi, serta kurangnya struktur dalam proses *brainstorming* sering menjadi hambatan dalam pencapaian efisiensi kerja tim. Permasalahan ini menjadi semakin krusial di lingkungan *startup* yang bergerak cepat, sehingga diperlukan suatu pendekatan sistematis yang mampu meningkatkan transparansi dan efektivitas proses internal.

Setelah masalah dirumuskan secara jelas, tahap selanjutnya adalah merumuskan tujuan penelitian. Tujuan utama dari penelitian ini adalah untuk menganalisis bagaimana metode *CSD Matrix* dapat diadopsi untuk meningkatkan kualitas proses bisnis dalam tim pengembang perangkat lunak di Mamikos. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi dampak dari implementasi metode tersebut dengan menggunakan kerangka *Capability Maturity Model Integration* (CMMI) sebagai alat ukur tingkat kematangan proses. Dengan adanya tujuan yang terfokus, penelitian ini diarahkan untuk memberikan kontribusi praktis dan teoretis dalam pengembangan model kerja tim yang lebih efisien, kolaboratif, dan terdokumentasi dengan baik di industri teknologi digital.

2. Studi Literatur

Studi terhadap publikasi ilmiah bertujuan untuk mengumpulkan data sekunder yang relevan dengan fokus penelitian, yaitu peningkatan kualitas proses bisnis dalam pengembangan perangkat lunak, khususnya melalui perbaikan komunikasi dan kolaborasi dalam tim. Proses ini dilakukan dengan melakukan peninjauan literatur secara komprehensif dari berbagai sumber, seperti buku, jurnal ilmiah, artikel teknologi, dan publikasi akademik lainnya yang membahas teori, konsep, maupun hasil penelitian terdahulu terkait proses pengembangan perangkat lunak, kualitas kolaborasi tim, dan penggunaan metode *CSD Matrix*.

Topik utama yang menjadi fokus peninjauan mencakup kualitas proses bisnis pengembangan perangkat lunak, dinamika komunikasi dan koordinasi tim, serta

efektivitas metode *CSD Matrix* dalam membantu tim mengorganisir informasi berdasarkan kategori *Certainties*, *Suppositions*, dan *Doubts*. Peninjauan literatur ini penting dilakukan sebagai upaya membangun kerangka teoritis yang kokoh dan mendukung analisis data primer dengan landasan konseptual yang valid. Dengan memahami bagaimana metode visual seperti *CSD Matrix* telah diterapkan dalam konteks serupa, peneliti dapat merumuskan strategi analisis dan interpretasi data yang lebih tajam.

Untuk menjamin kualitas dan relevansi literatur yang dikaji, proses pencarian dilakukan melalui platform utama seperti Google Scholar, dengan rentang tahun publikasi 2017–2024. Kata kunci yang digunakan dalam pencarian meliputi: “*CSD Matrix*”, “*komunikasi tim dalam pengembangan perangkat lunak*”, “*kolaborasi tim Agile*”, dan “*kualitas proses bisnis software development*”. Setiap publikasi yang ditemukan dievaluasi berdasarkan kesesuaian tema dengan tujuan penelitian, kekuatan metodologinya, serta kontribusi konseptualnya terhadap pembahasan proses bisnis dalam tim pengembang perangkat lunak.

3. Pengumpulan Data

a. Observasi

Observasi dilakukan untuk mengidentifikasi langsung dinamika kerja tim pengembang perangkat lunak Mamikos, khususnya pada aktivitas *daily stand-up meeting* dan *brainstorming* proyek *Apple Music Affiliate Landing page* selama periode 1 Februari–15 Maret 2025. Peneliti menggunakan pendekatan peserta pasif, hanya mengamati tanpa ikut terlibat, namun tetap memberikan arahan teknis terkait penggunaan *CSD Matrix* jika dibutuhkan. *CSD Matrix* digunakan sebagai alat bantu untuk mengorganisir temuan ke dalam tiga kategori: *Certainties*, *Suppositions*, dan *Doubts*. Setiap temuan dicatat dan diperbarui secara berkala untuk melihat perkembangan pola komunikasi, pengambilan keputusan, serta potensi perbaikan proses kerja dalam tim.

b. Wawancara

Berikut adalah pengukuran kualitas proses bisnis menurut (Kroeger & Davidson, 2009), yang menjelaskan bahwa kualitas suatu proses dinilai dari empat atribut utama, yaitu *suitability* (tingkat kesesuaian proses terhadap kebutuhan dan konteks penerapannya), *usability* (kemudahan proses untuk dipahami dan dijalankan), *manageability* (kemudahan proses untuk direncanakan, dipantau, dan

dikendalikan), serta *evolvability* (kemudahan proses untuk dianalisis, dimodifikasi, dan ditingkatkan). Keempat atribut ini menjadi landasan dalam merancang pertanyaan wawancara yang bertujuan mengeksplorasi persepsi stakeholder terhadap kekuatan dan kelemahan proses yang sedang diterapkan dalam proyek pengembangan perangkat lunak. Wawancara dilakukan kepada Senior *Engineering Manager* Mamikos, Giga Prakosa Hikmata yaitu sebagai informan kunci yang memiliki pengaruh langsung terhadap struktur dan proses kerja tim pengembang.

Tabel 3. 1 Pertanyaan Wawancara Berdasarkan Kualitas Proses Bisnis

No.	Indikator Pengukuran	Pertanyaan Wawancara	Tujuan Pertanyaan
1	<i>Applicability</i>	Apakah proses bisnis yang Anda gunakan saat ini relevan dengan tugas-tugas yang Anda lakukan setiap hari?	Menggali sejauh mana proses sesuai kebutuhan pekerjaan.
2	<i>Effectiveness</i>	Menurut Anda, apakah proses ini mampu menghasilkan output yang sesuai dengan target organisasi?	Menilai kemampuan proses dalam menghasilkan hasil yang diharapkan.
3	<i>Efficiency</i>	Apakah proses ini dapat dijalankan dengan efisien, baik dari sisi waktu maupun sumber daya?	Mengukur efisiensi proses dari perspektif pelaksana.
4	<i>Compatibility</i>	Bagaimana tingkat kesesuaian proses ini ketika digunakan bersama sistem atau proses lainnya di organisasi Anda?	Menilai kemampuan integrasi proses dengan sistem/proses lain.
5	<i>Accessibility</i>	Apakah dokumentasi atau informasi proses ini mudah diakses oleh tim Anda?	Mengidentifikasi ketersediaan informasi proses bagi pengguna.
6	<i>Understandability</i>	Seberapa mudah memahami langkah-langkah dan tujuan dari proses bisnis yang ada?	Menilai kejelasan dan kemudahan pemahaman proses.
6	<i>Learnability</i>	Menurut Anda, apakah proses ini mudah dipelajari oleh anggota baru?	Menggali kemudahan proses dipelajari oleh personel baru.

No.	Indikator Pengukuran	Pertanyaan Wawancara	Tujuan Pertanyaan
7	<i>Adaptability</i>	Jika ada kondisi kerja yang berubah, apakah proses ini mudah untuk disesuaikan atau dimodifikasi?	Menilai fleksibilitas proses terhadap perubahan.
8	<i>Predictability</i>	Sejauh mana proses ini membantu Anda dalam merencanakan sumber daya, waktu, dan risiko?	Menilai dukungan proses terhadap perencanaan proyek.
9	<i>Monitorability</i>	Apakah mudah untuk memonitor atau mengevaluasi kemajuan dari proses ini?	Mengukur kemampuan monitoring dan tracking proses.
10	<i>Controllability</i>	Jika ada kendala, apakah Anda dapat dengan mudah melakukan perbaikan atau pengendalian pada proses tersebut?	Menggali kemudahan mengendalikan dan mengoreksi proses.
11	<i>Analyzability</i>	Apakah proses ini mudah untuk dianalisis guna mengetahui bagian mana yang perlu diperbaiki?	Menilai keterbukaan proses terhadap evaluasi dan analisis.
12	<i>Modifiability</i>	Jika ingin mengubah proses, apakah proses tersebut fleksibel untuk dimodifikasi?	Menggali kemudahan dalam melakukan perubahan pada proses.
13	<i>Deployability</i>	Bagaimana pengalaman Anda dalam mendistribusikan atau menerapkan proses baru ke dalam tim? Apakah mudah?	Menilai seberapa mudah proses baru diimplementasikan.
14	<i>Supportability</i>	Apakah ada dukungan (tools, pelatihan, dokumentasi) yang memadai untuk memastikan proses ini berjalan dengan baik?	Mengukur dukungan organisasi terhadap proses.
15	<i>Applicability</i>	Apakah proses bisnis yang Anda gunakan saat ini relevan dengan tugas-tugas yang Anda lakukan setiap hari?	Menggali sejauh mana proses sesuai kebutuhan pekerjaan.

c. Kuesioner

Sebagai bagian dari penelitian ini, peneliti menyebarkan kuesioner kepada tim pengembang perangkat lunak di Mamikos setelah implementasi *CSD Matrix* dalam proyek pengembangan perangkat lunak mereka, khususnya pada proyek “*Apple Music Affiliate Landing page in Mamikos Web.*” Kuesioner ini bertujuan untuk mengumpulkan umpan balik dari anggota tim mengenai pengalaman mereka dengan *CSD Matrix*, baik dalam hal pemahaman konsep, penerapan alat ini dalam pengambilan keputusan, serta dampaknya terhadap komunikasi, kolaborasi, dan efisiensi tim. Melalui kuesioner ini, peneliti ingin mengevaluasi sejauh mana *CSD Matrix* telah membantu tim dalam meningkatkan kualitas proses kerja, mengurangi miskomunikasi, dan mempercepat penyelesaian tugas atau deliverables proyek.

Kuesioner diberikan kepada 30 anggota tim *engineer* Mamikos dalam dua tahap, yaitu sebelum dan sesudah penerapan *CSD Matrix*. Tujuannya adalah untuk mengukur perubahan persepsi dan pengalaman anggota tim terhadap kualitas proses bisnis yang berlangsung akibat penerapan *CSD Matrix*. Instrumen kuesioner ini disusun berdasarkan indikator pengukuran CMMI dalam jurnal (Khraiweh, 2013), yang digunakan sebagai kerangka evaluatif dalam mengukur *Software Process Improvement* (SPI). Hasil dari kedua kuesioner, yaitu sebelum dan sesudah implementasi, akan dibandingkan untuk melihat dampak *CSD Matrix* terhadap peningkatan kualitas proses kerja tim pengembang.

Tabel 3. 2 Kuesioner Sebelum dan Sesudah Penerapan *CSD Matrix*

Proses Area	Kode	Pertanyaan	1	2	3	4	5
<i>Project Planning</i>	Q1	Apakah Anda memantau parameter perencanaan proyek sesuai dengan rencana proyek?					
	Q2	Apakah Anda memantau kemajuan proyek sesuai dengan jadwal?					
	Q3	Apakah Anda mengidentifikasi perbedaan signifikan terhadap perkiraan jadwal rencana proyek?					
	Q4	Apakah Anda secara berkala mengukur upaya aktual, biaya yang dikeluarkan, dan staf yang ditugaskan?					
	Q5	Apakah Anda membandingkan upaya aktual, biaya, penugasan staf, dan pelatihan dengan anggaran rencana proyek?					

Proses Area	Kode	Pertanyaan	1	2	3	4	5
<i>Project Risks</i>	Q6	Apakah Anda mengidentifikasi dan menganalisis risiko potensial terhadap proyek?					
	Q7	Apakah Anda mengidentifikasi tindakan pengurangan risiko dan memantau pelaksanaannya?					
<i>Data Management</i>	Q8	Apakah Anda memantau keterlibatan pemangku kepentingan dalam proyek?					
	Q9	Apakah Anda memantau komunikasi antara pemangku kepentingan?					
<i>Stakeholder Involvement</i>	Q10	Apakah Anda memantau data proyek untuk keakuratan dan kelengkapannya?					
	Q11	Apakah Anda mengidentifikasi dan memperbaiki ketidakkonsistenan dalam data proyek?					
<i>Corrective Action</i>	Q12	Apakah Anda memantau kepatuhan proses terhadap standar kualitas?					
	Q13	Apakah Anda memantau kepatuhan produk terhadap standar kualitas?					
	Q14	Apakah Anda memverifikasi bahwa tindakan perbaikan untuk masalah kualitas telah diimplementasikan?					

4. Pengukuran Pre-Test Menggunakan CMMI

Pengukuran pre-test dilakukan sebelum implementasi metode *CSD Matrix* untuk mengetahui tingkat awal kematangan proses bisnis dalam tim pengembang perangkat lunak. Instrumen pengukuran disusun berdasarkan indikator pada *Key Process Area* (KPA) dalam model *Capability Maturity Model Integration* (CMMI). Kuesioner dibagikan kepada anggota tim pengembang Mamikos guna memperoleh data persepsi terkait proses kerja, dokumentasi, manajemen risiko, dan efektivitas komunikasi sebelum intervensi dilakukan. Data yang diperoleh dari pre-test ini berfungsi sebagai tolok ukur awal yang akan dibandingkan dengan hasil post-test guna menilai dampak dari penerapan metode.

Dalam penelitian ini, model *Capability Maturity Model Integration* (CMMI) digunakan sebagai kerangka untuk mengevaluasi tingkat kematangan proses bisnis tim pengembang perangkat lunak sebelum dan sesudah implementasi *CSD Matrix*. CMMI merupakan salah satu standar dalam bidang *Software Process Improvement* yang digunakan secara luas untuk mengukur kapabilitas dan efektivitas proses organisasi dalam menghasilkan produk perangkat lunak berkualitas tinggi. CMMI dikembangkan

oleh *Software Engineering Institute* (SEI) dan menyajikan lima tingkat kematangan proses, yaitu:

- a. Level 1 *Initial*: Proses berjalan secara ad-hoc dan tidak terorganisir.
- b. Level 2 *Managed*: Proses telah direncanakan dan dijalankan sesuai prosedur dasar, namun belum terdokumentasi dengan baik.
- c. Level 3 *Defined*: Proses dikembangkan dan didokumentasikan secara standar organisasi.
- d. Level 4 *Quantitatively Managed*: Proses dimonitor dan dikendalikan menggunakan data kuantitatif.
- e. Level 5 *Optimizing*: Proses berfokus pada peningkatan berkelanjutan melalui inovasi dan perbaikan sistematis.

Penggunaan CMMI dalam konteks penelitian ini bertujuan untuk memberikan landasan konseptual dalam menilai perubahan kematangan proses bisnis tim sebelum dan sesudah penggunaan metode *CSD Matrix*. Penilaian dilakukan secara deskriptif berdasarkan indikator kunci dalam CMMI yang relevan dengan aspek pengelolaan proyek, manajemen tim, dokumentasi keputusan, serta proses peningkatan berkelanjutan.

Selain itu, kerangka CMMI membantu peneliti dalam menyusun indikator analisis yang dapat dikaitkan langsung dengan perubahan proses, seperti peningkatan konsistensi komunikasi tim, dokumentasi usulan dan keputusan, serta efektivitas proses retrospektif dan perencanaan *sprint*. Hal ini sejalan dengan prinsip evaluasi pre-post, di mana kondisi sebelum intervensi (*CSD Matrix*) dibandingkan dengan kondisi sesudahnya untuk mengukur dampaknya terhadap kualitas proses bisnis. Berikut adalah pengukuran yang digunakan dalam proses CMMI.

Tabel 3. 3 Pengukuran Pada CMMI

<i>Maturity Levels</i>	<i>Range Persentase</i>
<i>Initial</i>	< 40%
<i>Managed</i>	40–60%
<i>Defined</i>	>60–75%
<i>Quantitatively Managed</i>	>75–90%
<i>Optimizing</i>	>90%

Untuk memperoleh hasil kuantitatif, digunakan dua jenis perhitungan, yaitu nilai rata-rata dan persentase pencapaian terhadap skala maksimum. Adapun rumus yang digunakan adalah sebagai berikut:

1. Menghitung Nilai Rata-Rata

$$X = \frac{\sum X_i}{n}$$

Keterangan:

X = Nilai Rata-rata

$\sum X_i$ = Jumlah seluruh skor responden pada suatu KPA

n = Jumlah responden (dalam penelitian ini = 30)

2. Menghitung Persentase Capaian

$$Persentase = \left(\frac{X}{Skor Maksimum} \right) \times 100\%$$

Keterangan:

X = Nilai rata-rata hasil kuesioner

Skor maksimum = 5 (karena menggunakan skala Likert 1–5)

5. Penerapan CSD Matrix

Tahap implementasi CSD Matrix merupakan inti dari penelitian ini, di mana metode digunakan secara langsung oleh tim pengembang perangkat lunak di Mamikos dalam kegiatan *brainstorming* dan diskusi pengambilan keputusan. CSD Matrix (*Certainties, Suppositions, Doubts*) adalah metode visual dan partisipatif yang membantu tim memetakan ide atau rencana kerja berdasarkan tiga kategori utama: hal-hal yang pasti (*Certainties*), asumsi yang belum tervalidasi (*Suppositions*), dan hal-hal yang masih meragukan atau belum diketahui (*Doubts*). Tujuannya adalah untuk meningkatkan kesadaran bersama dalam tim, mendorong diskusi yang terbuka, serta meminimalkan risiko miskomunikasi dan asumsi yang tidak teruji dalam proses pengembangan.

Dalam konteks penelitian ini, CSD Matrix diimplementasikan pada salah satu proyek pengembangan fitur, yaitu “Apple Music Affiliate Landing page in Mamikos Web.” Tim yang terdiri dari berbagai peran seperti *backend*, *frontend*, QA, dan *mobile developer* berpartisipasi dalam sesi *brainstorming* menggunakan CSD Matrix untuk merumuskan strategi kerja yang lebih terstruktur. Hasil dari implementasi ini

didokumentasikan dan dianalisis untuk melihat bagaimana metode ini mempengaruhi pola kolaborasi, kejelasan tanggung jawab, serta efektivitas komunikasi tim. Selain itu, hasil diskusi yang tercermin dalam *CSD Matrix* dijadikan acuan dalam mengevaluasi kematangan proses menggunakan kerangka CMMI.

6. Pengukuran Post-Test Menggunakan CMMI

Setelah penerapan metode *CSD Matrix*, dilakukan pengukuran post-test menggunakan kuesioner yang sama berdasarkan indikator CMMI. Tujuan dari pengukuran ini adalah untuk mengevaluasi perubahan tingkat kematangan proses bisnis yang terjadi setelah metode diterapkan. Kuesioner disebarakan kembali kepada anggota tim pengembang pada akhir periode observasi. Hasil post-test ini dibandingkan dengan data pre-test untuk melihat adanya peningkatan atau perubahan dalam aspek-aspek seperti kolaborasi, pengambilan keputusan, dokumentasi, dan kontrol proses. Proses evaluasi ini dilakukan secara manual dan juga melalui sistem Android yang telah dirancang khusus untuk membantu perhitungan skor secara otomatis.

7. Perancangan dan Pengembangan Sistem Android

Perancangan dan pengembangan sistem Android dalam penelitian ini bertujuan untuk mengotomatisasi proses pengukuran tingkat kematangan proses bisnis berdasarkan model *Capability Maturity Model Integration* (CMMI). Sistem dikembangkan menggunakan Android Studio dengan bahasa pemrograman Java, dan dirancang agar pengguna dapat mengisi kuesioner pre-test dan post-test secara digital. Setiap input akan diproses secara otomatis untuk menghasilkan skor pada tiap *Key Process Area* (KPA). Hasil pengukuran dari sistem ini kemudian dibandingkan dengan perhitungan manual guna memastikan akurasi dan konsistensi data, sehingga sistem mampu mendukung evaluasi yang lebih efisien, objektif, dan terstandarisasi.

8. Analisis Perbandingan Manual dan Sistem

Metode analisis dalam penelitian ini dilakukan dengan membandingkan hasil pengukuran tingkat kematangan proses bisnis yang diperoleh melalui perhitungan manual dan sistem berbasis aplikasi Android. Perhitungan manual dilakukan dengan merekap skor dari kuesioner pre-test dan post-test menggunakan pedoman dari model *Capability Maturity Model Integration* (CMMI), sementara sistem memproses data secara otomatis berdasarkan algoritma yang telah dirancang. Hasil dari kedua metode kemudian dibandingkan untuk menilai konsistensi skor, efisiensi waktu, serta akurasi proses. Sebagai bagian dari analisis, sistem juga menampilkan grafik perbandingan

antara hasil manual dan hasil sistem untuk mempermudah interpretasi data secara visual. Langkah ini bertujuan untuk memvalidasi efektivitas sistem dalam memberikan evaluasi yang cepat, akurat, dan mudah dipahami.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Identifikasi Masalah

Identifikasi masalah dalam penelitian ini dilakukan berdasarkan observasi awal terhadap proses kerja tim pengembang perangkat lunak di Mamikos. Ditemukan bahwa dalam aktivitas pengambilan keputusan dan brainstorming, sering kali terjadi tumpang tindih informasi, asumsi yang tidak tervalidasi, serta minimnya dokumentasi terhadap keputusan tim. Kondisi ini menyebabkan proses bisnis berjalan kurang efisien, serta meningkatkan potensi kesalahan komunikasi dan pengulangan pekerjaan. Oleh karena itu, diperlukan suatu metode yang mampu memfasilitasi tim dalam mengelola informasi secara lebih terstruktur, seperti *CSD Matrix*, yang diharapkan dapat meningkatkan kolaborasi, kejelasan peran, dan kesadaran terhadap risiko dalam pengembangan perangkat lunak.

4.2 Hasil Pengumpulan Data

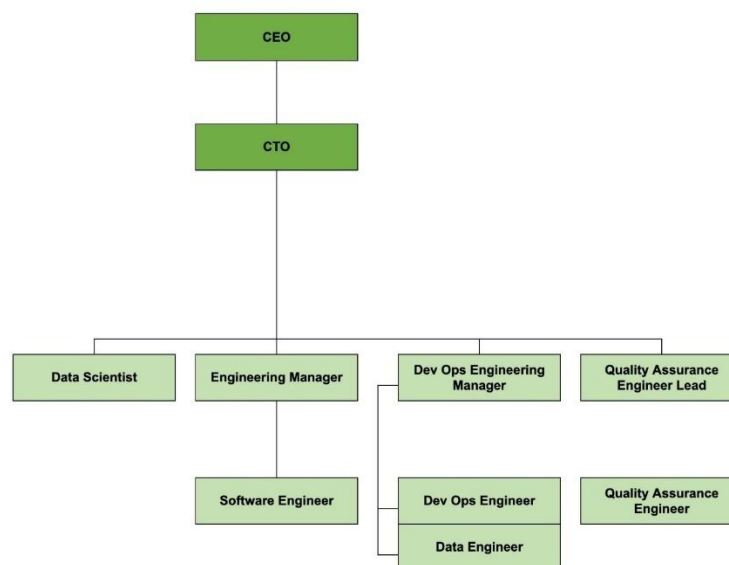
4.2.1 Struktur Tim Mamikos

Dalam upaya mendapatkan pemahaman mendalam mengenai penerapan metode *CSD Matrix* dalam proses pengembangan perangkat lunak, peneliti melakukan wawancara dengan Giga Prakosa Hikmata, selaku *Manager Software Engineering* di Mamikos. Wawancara dilakukan pada tanggal 10 November 2024, dan memberikan wawasan penting terkait bagaimana tim *Engineering* diorganisasi, bagaimana mereka menjalankan proses kerja, serta pendekatan manajerial yang digunakan dalam menangani proyek-proyek pengembangan fitur di lingkungan perusahaan. Wawancara ini juga membahas tantangan kolaboratif dan komunikasi yang sering muncul, serta upaya-upaya perbaikan proses yang telah dilakukan tim.

Dari wawancara tersebut diketahui bahwa Mamikos mengadopsi struktur tim berbasis *squad* sebagai pendekatan utama dalam membentuk organisasi pengembangan perangkat lunak. Pendekatan *squad-based* merupakan model struktur yang bersifat *Agile*, di mana tim-tim kecil dibentuk untuk bertanggung jawab terhadap area atau fitur spesifik dalam platform Mamikos. Model ini banyak diadopsi oleh perusahaan teknologi modern karena memberikan fleksibilitas, kecepatan adaptasi, serta ruang inovasi bagi tiap tim. Di Mamikos, tiap *squad* difokuskan pada produk tertentu, misalnya fitur pencarian kos, sistem pembayaran, layanan booking, atau manajemen pengguna.

Setiap *squad* terdiri dari berbagai peran teknis yang saling melengkapi, seperti *Backend Engineer*, *Frontend Engineer*, *Android/iOS Engineer*, *QA Engineer* dan lainnya. Komposisi tim disesuaikan dengan skala dan kompleksitas fitur yang dikerjakan. Model ini mendorong tiap tim untuk memiliki otonomi dalam merancang arsitektur, memilih teknologi yang digunakan, dan menentukan metode kerja yang paling sesuai dengan kebutuhan fitur yang mereka kembangkan. Namun, koordinasi tetap dilakukan secara lintas tim agar integrasi antar fitur tetap harmonis.

Saat ini, menurut Giga terdapat sekitar 10 *squad* inti yang aktif menangani berbagai bagian produk. Jumlah ini tidak bersifat tetap dan dapat berubah tergantung *roadmap* perusahaan dan kebutuhan bisnis. Beberapa *squad* bersifat jangka panjang dan stabil, terutama jika menangani fitur inti atau berkelanjutan. Namun, dalam kondisi tertentu, *squad* juga bisa bersifat sementara atau dibentuk untuk menangani proyek khusus. Fleksibilitas ini memungkinkan perusahaan untuk cepat merespons perubahan pasar atau kebutuhan pengguna.



Gambar 4. 1 Struktur Organisasi Mamikos

Selain *squad* utama, Mamikos juga memiliki tim-tim pendukung lintas *squad*. Tim-tim ini tidak tergabung langsung dalam *squad*, namun berperan penting dalam menjaga performa sistem, memastikan keamanan dan skalabilitas aplikasi, serta menyediakan dukungan data dan analisis. Mereka biasanya bekerja secara paralel dan berkolaborasi dengan lebih dari satu *squad* untuk memastikan semua aspek teknis yang bersifat fundamental dapat berjalan optimal.

Struktur organisasi semacam ini mendorong budaya kerja yang kolaboratif dan reflektif, di mana evaluasi proses internal menjadi bagian dari keseharian tim. Dalam konteks ini, metode *CSD Matrix* menjadi relevan karena mampu memperkuat kesadaran kolektif tim terhadap apa yang diketahui dengan pasti, apa yang masih merupakan dugaan, dan apa yang belum diketahui sama sekali. Dengan struktur tim yang *Agile* dan dinamis, penggunaan *CSD Matrix* dapat membantu menciptakan ruang diskusi terbuka dan mendorong perbaikan proses yang berbasis pada refleksi tim itu sendiri.

4.2.2 Lingkungan Kerja dan Alat yang Digunakan

Lingkungan kerja di Mamikos dibangun dengan mengedepankan fleksibilitas, kolaborasi, dan efisiensi. Sebagai perusahaan teknologi yang beroperasi di sektor properti digital, Mamikos menerapkan model kerja *hybrid*, di mana sebagian aktivitas dilakukan secara *daring (remote)* dan sebagian lainnya dilakukan secara langsung di kantor. Model ini dinilai sesuai dengan kebutuhan tim *Engineering* yang sangat bergantung pada komunikasi digital dan kolaborasi berbasis proyek. Budaya kerja di Mamikos menekankan pada keterbukaan antar tim, kecepatan adaptasi terhadap perubahan, dan otonomi dalam pengambilan keputusan teknis.

Untuk menunjang komunikasi sehari-hari, Mamikos menggunakan Slack sebagai platform utama. Slack memungkinkan seluruh tim baik antar *squad* maupun antar fungsi berkomunikasi secara cepat, efisien, dan terstruktur melalui kanal-kanal khusus. Selain digunakan untuk diskusi proyek, Slack juga terintegrasi dengan *tools* lain seperti GitHub dan Jira, sehingga notifikasi pengembangan, *merge request*, hingga pelacakan isu dapat dipantau langsung oleh tim tanpa berpindah aplikasi. Slack juga menjadi wadah informal yang memperkuat budaya komunikasi terbuka di lingkungan kerja *hybrid*.

Dalam hal perencanaan dan manajemen proyek, Mamikos mengandalkan Jira, platform manajemen proyek berbasis *Agile*. Melalui Jira, tim dapat menyusun *backlog* pekerjaan, membagi tugas berdasarkan prioritas *sprint*, serta memantau progres masing-masing anggota. Jira juga digunakan untuk dokumentasi historis keputusan teknis yang telah diambil, sehingga menjadi sumber rujukan penting dalam evaluasi kinerja dan perencanaan ulang. Setiap *squad* biasanya memiliki board tersendiri yang memungkinkan tim mengelola tugas mereka secara lebih mandiri dan terfokus.

Pengelolaan kode sumber dilakukan melalui GitHub, *platform version control* yang terintegrasi secara langsung dalam *workflow* tim. GitHub tidak hanya digunakan untuk menyimpan dan mengelola kode, tetapi juga untuk melakukan *code review*, diskusi teknis, dan

pengujian otomatis. Proses *pull request* menjadi mekanisme utama dalam mengontrol kualitas kode dan mendorong kolaborasi teknis. Setiap perubahan yang dilakukan harus melalui proses tinjauan tim, sehingga tidak hanya menjamin stabilitas sistem, tetapi juga mendorong transfer pengetahuan antar anggota.

Secara keseluruhan, lingkungan kerja di Mamikos dirancang untuk menciptakan ekosistem pengembangan perangkat lunak yang adaptif, *scalable*, dan berkelanjutan. Kombinasi alat-alat digital dan budaya kerja yang kolaboratif memungkinkan tim untuk menjalankan tugas dengan efisien dan menjaga kualitas produk. Dalam konteks ini, penerapan metode CSD *Matrix* berfungsi sebagai alat bantu reflektif untuk memperjelas asumsi, mengidentifikasi ketidakpastian, dan membangun pemahaman kolektif yang dapat meningkatkan efektivitas proses kerja dalam lingkungan yang kompleks dan dinamis.

4.3 Analisis Pengukuran *Pre-Test* Menggunakan CMMI Sebelum Penerapan CSD

Matrix

Sebelum penerapan metode CSD *Matrix*, dilakukan pengukuran awal terhadap kualitas proses bisnis tim pengembang Mamikos menggunakan kuesioner berbasis *Key Process Area* (KPA) dari CMMI. Kuesioner ini disebarikan kepada 30 anggota tim dari berbagai peran, seperti *Engineering Manager*, *QA Engineer*, dan *Data Scientist*, menggunakan skala Likert yang disesuaikan dengan referensi penelitian Oliveira. Tujuannya adalah untuk mengetahui tingkat kematangan proses dalam aspek perencanaan proyek, manajemen risiko, pengelolaan tim, dan integrasi proses. Hasil kuesioner memberikan gambaran awal mengenai area yang sudah terstandar maupun yang masih informal, serta menjadi *baseline* untuk menganalisis dampak penerapan CSD *Matrix* terhadap peningkatan kualitas proses bisnis. Hasil selengkapnya ditampilkan pada Tabel 4.1.

Tabel 4. 1 Hasil Perhitungan CMMI Sebelum Implementasi CSD *Matrix*

No	<i>Key Process Area</i>	Rata-Rata	Persentase
1	<i>ProjectPlanning</i>	2,74	55%
2	<i>Project Risks</i>	2,416666667	48%
3	<i>Data Management</i>	2,866666667	57%
4	<i>Stakeholder Involvement</i>	2,85	57%
5	<i>Corrective Action</i>	2,611111111	52%
Total		2,692098765	54%
<i>Maturity Levels: 2 Managed</i>			

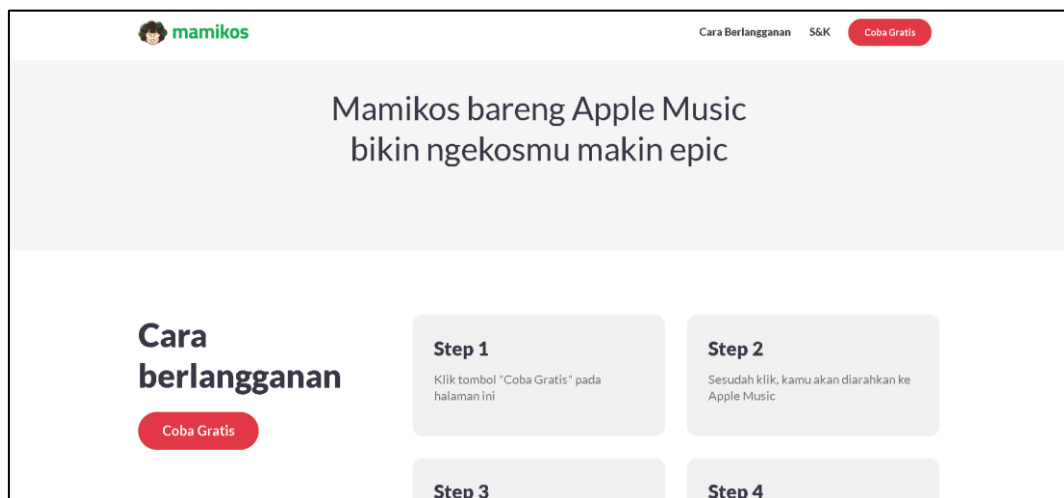
Berdasarkan Tabel 4.1, hasil perhitungan pra-implementasi menunjukkan bahwa rata-rata capaian keseluruhan *Key Process Area* (KPA) adalah 2,69 atau setara dengan 54%. Hal ini menandakan bahwa sebagian besar proses bisnis tim pengembang perangkat lunak Mamikos telah berada pada *Maturity Level 2 (Managed)*, di mana proses sudah mulai dikelola dan dipantau, namun dokumentasi, standarisasi, serta konsistensi pelaksanaan masih belum optimal. Tiga KPA dengan persentase terendah adalah *Project Risks* (48%), *Corrective Action* (52%), dan *Project Planning* (55%), yang menunjukkan adanya kelemahan pada pengelolaan risiko, tindak lanjut korektif terhadap permasalahan, serta perencanaan proyek yang masih perlu ditingkatkan.

Pada *Maturity Level 2*, organisasi telah mencapai seluruh tujuan generik dan spesifik pada area proses yang relevan. Artinya, kebutuhan (*requirements*) sudah dikelola, perencanaan proyek (*project planning*) telah dilakukan, pemantauan dan pengendalian proyek (*project monitoring and control*) berjalan, serta pengelolaan perjanjian dengan pemasok (*supplier agreement management*) diterapkan. Selain itu, pengukuran dan analisis (*measurement and analysis*), jaminan mutu proses dan produk (*process and product quality assurance*), serta manajemen konfigurasi (*configuration management*) sudah diimplementasikan. Namun pada level ini, pencapaian tersebut masih berada pada tahap pengelolaan awal sehingga efektivitasnya belum maksimal, khususnya pada area dengan nilai terendah. Kondisi ini menjadi dasar perlunya penerapan metode *CSD Matrix* untuk memperkuat proses dan meningkatkan tingkat kematangan pada evaluasi selanjutnya.

Berdasarkan hasil wawancara dengan Giga Prakosa Hikmata, Manager Software Engineering di Mamikos, diungkapkan bahwa pada lingkup awal pengembangan perangkat lunak tim telah menerapkan metode *Agile* untuk mengelola proyek. Meskipun demikian, dalam praktiknya masih sering terjadi *misscommunication* antar anggota tim, terutama terkait pembagian tugas dan pemahaman prioritas pekerjaan. Proses perencanaan proyek (*Project Planning*) telah dilakukan sejak awal, mencakup penetapan tujuan, ruang lingkup, dan estimasi waktu, namun belum sepenuhnya terdokumentasi secara formal ataupun disusun oleh peran khusus. Perencanaan masih bersifat informal dan kolektif, tanpa dokumen standar atau penanggung jawab yang mengatur jalannya proses secara terstruktur. Kondisi ini menunjukkan bahwa meskipun aktivitas perencanaan telah berjalan, pelaksanaannya masih belum mencapai tingkat dokumentasi dan kontrol penuh sesuai praktik terbaik CMMI.

4.4 Penerapan CSD Matrix

Analisis ini bertujuan mengevaluasi sejauh mana CSD Matrix membantu tim dalam mengidentifikasi kepastian, asumsi, dan keraguan selama proses pengembangan, serta kontribusinya terhadap pemahaman tim, kejelasan peran, dan komunikasi antaranggota. Penerapan CSD Matrix dilakukan pada tahap *Sprint* awal proyek *Apple Music Affiliate Landing page in Mamikos Web* melalui sesi *brainstorming* antara tim pengembang, *product manager*, dan *marketing*. Proses ini membantu mengelompokkan informasi berdasarkan *Certainties*, *Suppositions*, dan *Doubts*.



Gambar 4. 2 *Apple Music Affiliate Landing page in Mamikos Web*

1. Identifikasi Masalah/Topik Proyek

Proyek ini berfokus pada pengembangan *Apple Music Affiliate Landing page* di *platform web* Mamikos sebagai bagian dari inisiatif kerja sama afiliasi dengan Apple. Tujuan utama proyek ini adalah meningkatkan konversi pengguna Mamikos menjadi pelanggan *Apple Music*, khususnya dengan memanfaatkan potensi audiens pelajar melalui program diskon khusus. Untuk memastikan efektivitas strategi ini, dilakukan identifikasi awal terhadap elemen-elemen penting yang memengaruhi keberhasilan kampanye melalui pendekatan CSD Matrix.

[CSD Matrix for " Apple Music Affiliate Landing Page in Mamikos Web"]		
Certainties	Suppositions	Doubts
We know or understand that... Landing Page Purpose: The Apple Music Affiliate Landing Page is meant to drive new Apple Music subscriptions through Mamikos, targeting students and general users. Affiliate Model & Revenue: - Mamikos earns 51% commission on the first subscription month. - Commission rates: IDR 31,850/student and IDR 50,050/individual. - Revenue validation takes 2-10 days, with the first payment arriving within 45-60 days after reaching a \$30 threshold. All CTA clicks on the landing page will be tracked. - Events like "[User] Partnership Click" will be logged with parameters: partnership: "apple-music" and source: "twitter", "header_banner", "subscribe_step". - Tracking data is stored in internal trackers and Partnerize (affiliate tracking platform), with data refreshing every 30 minutes. Landing Page Components: - Header Navbar (sticky like Mamikos' main page). - Mobile Menu for better accessibility. - Header Banner Section for promotions. - Subscription Step Section guiding users through Apple Music sign-up. - CTA Buttons leading to Apple Music sign-up. - FAQ Section explaining Apple Music subscriptions, eligibility, and cancellation policies. Marketing Exposure Plan: - Homebanner on the Mamikos homepage. - Push Notifications (PN) sent on scheduled dates. - Newsletter Footers promoting the campaign. - Social Media (Instagram/TikTok) featuring redirections to the LP. Technical Constraints: - Apple's branding guidelines must be followed strictly. - CTA links should use Partnerize tracking parameters. - The LP structure should be similar to SomeByMe's previous LP, with necessary adjustments. Project Timeline: - LP development period: Feb 3 - Feb 11. - LP tracking implementation & testing: Feb 10 - Feb 14. - LP goes live: Feb 15, 2025. - Marketing exposure via PN, banners, and social media continues until Feb 28, 2025.	We suppose that... Student Users Will Drive Higher Conversion: Since the student plan offers a lower price (IDR 35,000), students are more likely to subscribe. Social Media Will Be a Strong Traffic Driver: Given Mamikos' active audience on Instagram and TikTok, these channels may bring significant traffic. Push Notifications Will Be Effective: Since Mamikos already engages users via PN, this could be a key conversion channel. LP Performance Depends on Apple's Approval Speed: If Apple delays approvals for branding assets, it may affect the LP launch timeline. Apple Music Free Trial Might Reduce Immediate Paid Conversions: Since users get a 1-month free trial, paid subscriptions might take longer to materialize. Affiliate Revenue Payouts Might Be Delayed: If users cancel before completing the first paid month, some commissions might be lost. Users Might Drop Off at Subscription Steps: If the subscription process on Apple's side is too complex, some users may abandon signups. Traffic Sources Need Optimization: Not all channels (PN, banners, or social media) will perform equally; testing and A/B analysis may be needed.	We have doubts about... Will Mamikos Users Engage with Apple Music? - Mamikos primarily serves rental seekers. Will they see value in Apple Music subscriptions? Will the Commission Model Be Profitable? - Is the 51% commission for the first month sustainable for Mamikos' marketing investment? What if Apple Changes Terms? - Apple might change its affiliate structure, reducing revenue potential. How Will We Handle Tracking Data Issues? - If internal tracker or Partnerize fails, how will data loss be mitigated? Will Branding Constraints Delay Execution? - Apple has strict branding rules. If revisions are needed, will it impact launch dates? Is the User Journey Smooth Enough? - Do users face friction when redirected to Apple Music's sign-up page? Will Users Cancel After the Free Trial? - If many users cancel before the first paid month, the actual revenue may be lower than expected.

Gambar 4. 3 CSD Matrix Fase Identifikasi

Berdasarkan pada Gambar 4.3, penerapan metode CSD *Matrix* dalam proyek *Apple Music Affiliate Landing page* membantu tim pengembang Mamikos dalam memahami dan mengelola elemen penting proyek secara sistematis. Berikut adalah poin-poin utama yang paling dirasakan manfaatnya oleh tim:

a. Kepastian (*Certainties*)

Tim menyadari bahwa *Landing page* menjadi titik konversi utama karena seluruh trafik kampanye diarahkan ke halaman ini. Selain itu, sistem pelacakan menggunakan internal tracker dan Partnerize telah siap digunakan, serta desain halaman telah disesuaikan dengan panduan resmi dari Apple. Jadwal peluncuran juga telah ditetapkan secara jelas, menandakan kesiapan operasional yang kuat.

b. Asumsi (*Suppositions*)

Tim mengasumsikan bahwa diskon khusus pelajar akan menjadi faktor utama dalam mendorong konversi, karena mayoritas pengguna Mamikos adalah mahasiswa. *Push notification* juga diasumsikan menjadi kanal paling efektif untuk komunikasi, sementara media sosial seperti Instagram dan TikTok diyakini mampu menarik trafik jika disertai konten yang relevan dengan target audiens.

c. Keraguan (*Doubts*)

Terdapat keraguan apakah pengguna Mamikos, yang utamanya mencari hunian, benar-benar tertarik dengan langganan musik seperti *Apple Music*. Kekhawatiran lain mencakup keberlanjutan skema komisi, risiko kegagalan sistem *tracking*, serta

potensi perubahan kebijakan dari Apple yang bisa memengaruhi peluncuran atau keuntungan kampanye.

2. *Brainstorming* dan Diskusi Tim

Tahap *Brainstorming* dan Diskusi Tim dilakukan untuk mengumpulkan berbagai perspektif dari anggota tim pengembang terkait peluang, asumsi, dan tantangan dalam implementasi proyek *Apple Music Affiliate Landing page*, yang kemudian dituangkan ke dalam struktur *CSD Matrix* sebagai dasar pengambilan keputusan berikutnya.

[CSD Matrix for " Apple Music Affiliate Landing Page in Mamikos Web"]		
Certainties	Suppositions	Doubts
We know or understand that... Landing Page is the Main Conversion Point - Since all traffic sources (PN, banners, social media, email) lead to the LP, its design, usability, and clarity are critical for success. - The CTA buttons must be prominent and well-placed to encourage conversions. Affiliate Model Provides Revenue, But Is Time-Delayed - Mamikos only earns revenue after the free trial converts into a paid subscription. - There's a 2-10 day validation period before commissions are confirmed, and payments are only made after reaching a \$30 threshold. - This means cash flow from this project will not be immediate. Tracking Clicks is Essential for Performance Measurement - Since tracking is integrated into internal trackers + Partnerize, the accuracy of this data will determine how success is measured. - Any misconfiguration in tracking could result in lost commission revenue. Marketing Exposure is Multi-Channel, So Performance Will Vary - The LP will be promoted via Homebanner, PN, Newsletter, and Social Media (IG/TikTok). - Some channels will outperform others, but we won't know until post-launch analysis. Apple's Branding Restrictions May Slow Execution - Apple's strict brand guidelines mean every creative asset requires approval. - If branding revisions are needed, this could impact the LP launch date.	We suppose that... Student Discounts Might Be the Strongest Selling Point - Since students get a lower price (IDR 35,000) and Apple Music is popular among younger demographics, this could be the best-performing plan. - If conversion data confirms this, future campaigns could focus primarily on student subscriptions. Push Notifications Could Have the Highest Engagement - Mamikos already sends PNs to its users, so leveraging this for Apple Music could drive the most clicks. - However, if open rates are low, we may need stronger messaging or incentives. Social Media Performance Might Depend on Content Style - If Mamikos' usual content style doesn't match Apple's sleek branding, engagement might be lower than expected. - A/B testing different messaging approaches (fun vs. professional tone) may help optimize performance. LP Friction at Subscription Step Might Impact Conversions - If users get confused when redirected from Mamikos to Apple Music's subscription page, drop-offs could increase. - This means the "How to Subscribe" section must be super clear to reduce confusion. Long-Term Revenue Depends on User Retention Beyond the Free Trial - If many users cancel before paying, Mamikos won't get commissions. - Retention strategies (e.g., sending reminders before the trial ends) might help keep more users subscribed.	We have doubts about... Mamikos Users Might Not Be Interested in Apple Music - Since Mamikos users are primarily rent-seekers, they may not be looking for music subscriptions. - If engagement is low, alternative incentives might be needed (e.g., bundling music subscriptions with rent-related services). What If Apple Changes the Affiliate Terms? - Apple could adjust commission rates at any time, reducing profitability. - Mamikos should have backup affiliate partnerships in case this happens. Tracking Errors Could Lead to Lost Revenue - If tracking breaks, Mamikos won't get credit for referrals. - Implementing redundant logging systems or backup trackers may help avoid data loss. Branding Approval Process Could Delay Launch - Apple has final say on LP design and branding. - If delays occur, having a pre-approved version of creatives may help speed up approval. Are Users Aware They Will Be Charged After the Free Trial? - Some users might not realize they'll be charged after 1 month, leading to dissatisfaction. - A clear FAQ and disclaimer will be needed to prevent misunderstandings.

Gambar 4. 4 Fase *Brainstroming* CSD Matrix

Berdasarkan pada Gambar 4.4, tahap *brainstorming* dilakukan oleh tim lintas fungsi untuk mengidentifikasi peluang, asumsi, dan tantangan proyek *Apple Music Affiliate Landing page*. Diskusi ini dituangkan ke dalam *CSD Matrix* yang menjadi dasar dalam pengambilan keputusan proyek ke depannya.

a. *Certainties*

Tim sepakat bahwa *landing page* merupakan titik utama konversi karena seluruh trafik dari berbagai kanal promosi diarahkan ke halaman tersebut. Selain itu, sistem tracking konversi telah dipastikan aktif menggunakan *Partnerize* dan *internal tracker*. Semua elemen teknis seperti desain halaman, FAQ, dan elemen *branding* juga telah sesuai dengan standar Apple, serta timeline proyek telah dirancang dan dikomunikasikan dengan jelas.

b. *Suppositions*

Tim mengasumsikan bahwa diskon pelajar akan menjadi faktor utama dalam menarik minat pengguna, mengingat mayoritas pengguna Mamikos adalah mahasiswa. *Push notification* dianggap sebagai kanal komunikasi paling efektif, dan

media sosial seperti Instagram serta TikTok diperkirakan dapat menghasilkan trafik tinggi bila kontennya dikemas dengan gaya yang sesuai. Semua asumsi ini masih memerlukan validasi lebih lanjut dalam implementasi.

c. *Doubts*

Beberapa keraguan mencuat dalam diskusi, seperti ketertarikan pengguna Mamikos terhadap layanan *Apple Music* yang tidak berhubungan langsung dengan pencarian kos. Selain itu, model komisi yang bergantung pada konversi berbayar setelah masa percobaan juga menjadi perhatian. Risiko lain termasuk potensi kegagalan *tracking*, perubahan kebijakan afiliasi dari Apple, hingga kerumitan proses langganan dari sisi pengguna.

d. Dampak CSD *Matrix* dalam Diskusi

Penggunaan CSD *Matrix* sangat membantu tim dalam menyusun *roadmap* proyek secara strategis. Dengan memisahkan kepastian, asumsi, dan keraguan, tim dapat mengalokasikan fokus dan sumber daya secara efisien dari eksekusi cepat hingga mitigasi risiko. Metode ini juga mendorong partisipasi aktif semua anggota tim, menghasilkan peta masalah dan peluang yang lebih realistis untuk menghadapi fase pengembangan berikutnya.

3. Validasi dan Pengambilan Keputusan

Tahap Validasi dan Pengambilan Keputusan merupakan lanjutan dari proses *brainstorming*, di mana tim pengembang di Mamikos melakukan evaluasi terhadap temuan dalam CSD *Matrix* dengan membandingkan *Suppositions* dan *Doubts* yang ada dengan data historis, hasil pengujian awal, serta masukan dari stakeholder terkait. Proses ini membantu memverifikasi asumsi yang paling mungkin berhasil, menetapkan prioritas tindak lanjut, serta menentukan langkah strategis berdasarkan kepastian dan risiko yang telah diidentifikasi sebelumnya.

Certainties	Suppositions	Doubts
We know or understand that... Optimize the Landing Page for Conversions - Ensure CTA buttons are clearly visible and placed strategically. - Use concise copy and step-by-step guidance to make the sign-up process clear. - Make sure the FAQ section clearly explains that the free trial auto-renews into a paid subscription. Double-Check Tracking Implementation - Test all tracking parameters (partnership: "apple-music", source: "mamikos", etc.). - Verify Partnerize integration to ensure Mamikos gets credited for referrals. - Conduct A/B testing on CTA placements to identify high-performing areas. Prepare for Affiliate Revenue Delays - Since revenue isn't immediate, Mamikos should not rely on instant cash flow from this campaign. - Monitor conversion data to calculate expected payouts and plan financials accordingly. Ensure Multi-Channel Marketing Execution - Push Notifications: Schedule and monitor click-through rates (CTRs). - Homebanner: Ensure branding consistency and track impressions. - Social Media: Run A/B tests on different messaging styles (fun vs. professional). - Newsletter Footers: Keep the message short and engaging. Prepare Contingency Plans for Branding Delays - Submit Apple branding approvals early to prevent timeline disruptions. - Have a backup version of creatives that comply with Apple's guidelines.	We suppose that... Leverage Student Discounts as the Primary Selling Point - Highlight the IDR 35,000/month plan more prominently. - Create marketing content that emphasizes how students can save money with this deal. Optimize Push Notifications for Maximum Impact - Experiment with different messaging formats (urgency-based vs. informative). - Monitor delivery and open rates to adjust timing and content. A/B Test Social Media Content to Find the Best Performing Style - Test Apple-style sleek branding vs. Mamikos' casual, fun branding. - Use Instagram Stories & TikTok interactive features (polls, Q&A, stickers) to engage users. Simplify the Subscription Process to Minimize Drop-offs - Include a short video or GIF showing how to subscribe. - Use numbered steps to guide users through Apple Music's sign-up process. Monitor Post-Free-Trial Cancellations - Track how many users cancel after the free trial. - Consider sending reminder emails or push notifications before the trial ends.	We have doubts about... Validate Whether Mamikos Users Are Interested in Apple Music - Conduct user surveys or analyze user behavior data from similar promotions. - If engagement is low, consider bundling Apple Music with other Mamikos services. Mitigate Risk of Apple Changing Affiliate Terms - Diversify affiliate partnerships to avoid reliance on Apple. - Negotiate with Apple for longer-term revenue-sharing commitments. Ensure Tracking Accuracy to Prevent Revenue Loss - Implement redundant tracking logs in case Partnerize fails. - Regularly audit tracking data to confirm all clicks are being recorded correctly. Prepare for Apple Branding Approval Delays - Maintain continuous communication with Apple for faster approvals. - If branding gets delayed, launch with a placeholder design while awaiting final approval. Reduce User Confusion About Subscription Charges - Make sure the "Auto-renewal after free trial" message is clearly stated. - Add a reminder feature that notifies users before the trial ends.

Gambar 4. 5 Fase Validasi dan Pengambilan Keputusan

Berdasarkan pada Gambar 4.5, tahap validasi dilakukan untuk menguji asumsi dan mengkonfirmasi kepastian yang telah diidentifikasi sebelumnya melalui data, pengujian awal, serta masukan dari stakeholder. Hasilnya menjadi dasar pengambilan keputusan strategis dalam peluncuran *Apple Music Affiliate Landing page* di Mamikos.

a. *Certainties*

Tim sepakat bahwa *landing page* merupakan titik utama konversi karena seluruh kanal promosi mengarah ke halaman tersebut. UI/UX dan CTA telah diuji secara internal dan terbukti efektif mendorong interaksi. Sistem *tracking* juga sudah dipastikan aktif menggunakan Partnerize dan parameter tertentu, serta seluruh elemen teknis seperti struktur halaman, FAQ, dan *branding* telah disesuaikan dengan standar Apple.

b. *Suppositions*

Tim mengasumsikan bahwa diskon pelajar sebesar IDR 35.000 menjadi daya tarik utama mengingat mayoritas pengguna Mamikos adalah mahasiswa. *Push notification* dianggap sebagai kanal yang paling efektif karena sudah familiar di kalangan pengguna, dan media sosial diasumsikan dapat mendatangkan trafik tinggi jika dikemas dengan gaya konten yang sesuai. Validasi asumsi ini akan dilakukan melalui *A/B testing* dan analisis kampanye.

c. *Doubts*

Beberapa keraguan muncul mengenai apakah pengguna Mamikos yang mencari kos benar-benar tertarik pada langganan musik seperti *Apple Music*. Model komisi juga menimbulkan risiko, karena pendapatan hanya diberikan setelah pengguna menyelesaikan masa trial dan melakukan pembayaran. Kekhawatiran lain mencakup potensi perubahan aturan afiliasi Apple dan kegagalan sistem tracking.

4.5 Analisis Pengukuran *Post-Test* Menggunakan CMMI Setelah Penerapan CSD

Matrix

Setelah penerapan metode CSD *Matrix* dalam pengembangan perangkat lunak di Mamikos, dilakukan evaluasi ulang terhadap kualitas proses bisnis tim, khususnya dalam dokumentasi keputusan, manajemen risiko, refleksi tim, dan kolaborasi lintas fungsi. Melalui identifikasi *Certainties*, *Suppositions*, dan *Doubts* dalam sesi *brainstorming*, tim menjadi lebih terbuka dalam menyusun ruang lingkup proyek dan langkah kerja. Berikut adalah hasil dari perhitungan yang telah dilakukan dapat dilihat pada Tabel 4.2 dibawah ini.

Tabel 4. 2 Perhitungan Setelah Implementasi CSD *Matrix*

No	Key Process Area	Rata-Rata	Persentase
1	<i>ProjectPlanning</i>	3,686666667	77%
2	<i>Project Risks</i>	3,65	73%
3	<i>Data Management</i>	3,733333333	73%
4	<i>Stakeholder Involvement</i>	3,8	75%
5	<i>Corrective Action</i>	3,655555556	73%
Total		3,71168	74%
<i>Maturity Levels: 3 Defined</i>			

Berdasarkan Tabel 4.2, hasil perhitungan CMMI pasca-implementasi menunjukkan rata-rata capaian keseluruhan *Key Process Area* (KPA) meningkat menjadi 3,71 atau setara dengan 74%. Peningkatan ini menandakan bahwa proses bisnis tim pengembang perangkat lunak Mamikos telah mencapai *Maturity Level 3 (Defined)*, di mana seluruh tujuan generik dan spesifik pada area proses level sebelumnya dan level 3 telah tercapai. Pada tahap ini, proses-proses sudah dipahami dengan baik, terdokumentasi, serta distandarisasi dalam bentuk prosedur dan metode yang berlaku di organisasi.

Perbedaan utama antara Level 2 dan Level 3 terletak pada cakupan standar dan deskripsi proses yang lebih luas, dengan prosedur yang dapat disesuaikan (*tailored*) untuk kebutuhan proyek atau unit organisasi. Hasil pada Tabel 4.2 menunjukkan peningkatan signifikan di seluruh KPA, terutama pada *Project Planning* (77%), *Stakeholder Involvement* (75%), dan *Risk Management* (73%) yang kini lebih terdokumentasi dan terkoordinasi. Mengacu pada Khraiweh (2013), Level 3 (*Defined*) mencakup proses seperti *Decision Analysis and Resolution*, *Risk Management*, dan *Integrated Project Management*, yang untuk konteks *startup* seperti Mamikos sudah tergolong ideal karena seluruh proses inti telah terdokumentasi, tersebar di seluruh tim, dan dapat diadaptasi sesuai kebutuhan proyek.

Hasil wawancara dengan Giga Prakosa Hikmata, *Manager Software Engineering* di Mamikos, menunjukkan bahwa proses perencanaan proyek (*Project Planning*) setelah penerapan pendekatan yang lebih terstruktur mengalami peningkatan signifikan. Setiap peran dalam tim kini memiliki keterlibatan yang lebih jelas dalam penyusunan rencana kerja, dengan dokumen perencanaan yang tidak hanya memuat tujuan dan ruang lingkup, tetapi juga mencakup hasil eksplorasi *Certainties*, *Suppositions*, dan *Doubts* sebagai dasar pertimbangan sejak tahap awal. Informasi tersebut membantu tim mengidentifikasi hal-hal yang sudah pasti, asumsi yang perlu divalidasi, dan keraguan yang harus diantisipasi, sehingga risiko dapat dikelola lebih proaktif. Pendekatan ini menciptakan dasar kerja yang lebih matang, kolaboratif, dan terdokumentasi dengan baik, memudahkan koordinasi lintas fungsi serta meningkatkan kesiapan tim dalam menghadapi tantangan selama pelaksanaan proyek.

4.6 Perbandingan Tingkat Kematangan Proses Sebelum dan Sesudah Perhitungan

Manual

Untuk melihat secara jelas perubahan tingkat kematangan proses bisnis, berikut disajikan perbandingan hasil penilaian sebelum dan sesudah penerapan *CSD Matrix* pada tim pengembang perangkat lunak Mamikos.

Tabel 4. 3 Perbandingan Tingkat Kematangan Proses Bisnis Sebelum dan Sesudah Implementasi *CSD Matrix*

No	Key Process Area	Sebelum (Rata-Rata)	Sesudah (Rata-Rata)	Sebelum (%)	Sesudah (%)
1	<i>Project Planning</i>	2,74	3,68667	55%	74%
2	<i>Project Risks</i>	2,41667	3,65	48%	73%
3	<i>Data Management</i>	2,86667	3,73333	57%	75%
4	<i>Stakeholder Involvement</i>	2,85	3,8	57%	76%
5	<i>Corrective Action</i>	2,61111	3,65556	52%	73%
Rata-rata Keseluruhan		2,69689	3,70511	54%	74%

Berdasarkan Tabel 4.3, rata-rata keseluruhan Key Process Area (KPA) mengalami peningkatan dari 2,69 (54%) sebelum penerapan menjadi 3,70 (74%) sesudah penerapan, yang mencerminkan transisi dari *Maturity Level 2 (Managed)* menuju *Level 3 (Defined)* dalam kerangka CMMI. Peningkatan ini menunjukkan bahwa proses yang sebelumnya hanya dikelola kini telah terdokumentasi, distandarisasi, dan diterapkan secara konsisten di seluruh tim. KPA

seperti *Project Planning* (55% menjadi 74%), *Stakeholder Involvement* (57% menjadi 76%), *Project Risk* (48% menjadi 73%), *Data Management* (57% menjadi 75%), dan *Corrective Action* (54% menjadi 74%) mengalami peningkatan signifikan, yang mencerminkan perbaikan pada aspek perencanaan, kolaborasi, dan pengelolaan risiko yang lebih terstruktur.

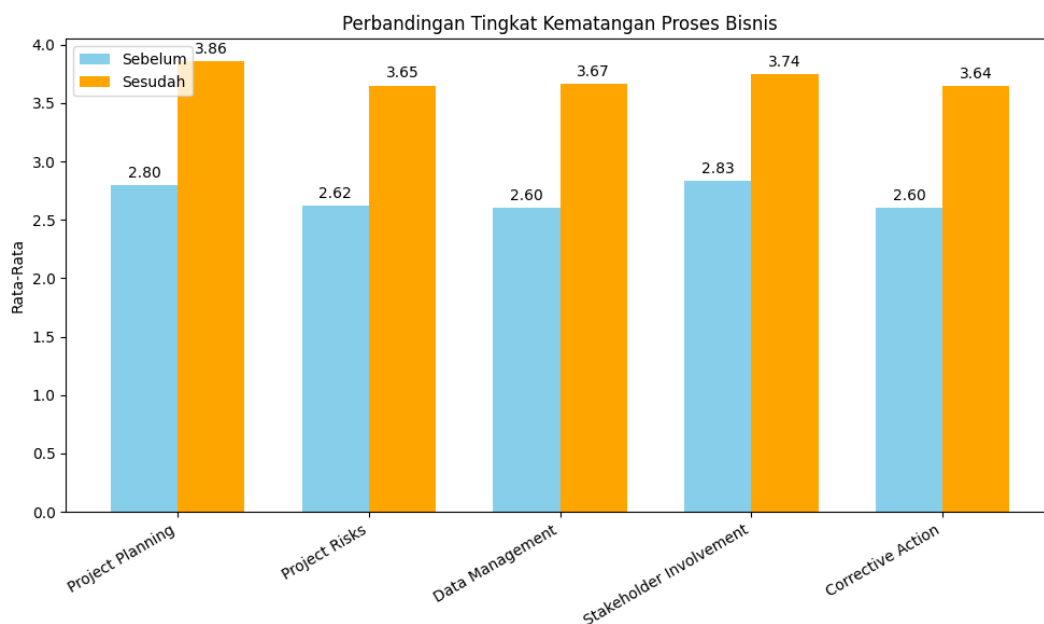
Selain itu, seluruh KPA menunjukkan kenaikan persentase antara 21% hingga 25%, menandakan adanya dampak positif yang merata pada berbagai aspek proses bisnis. Perubahan ini tidak hanya terlihat dari peningkatan skor, tetapi juga dari pola kerja tim yang menjadi lebih kolaboratif, terdokumentasi, dan terfokus pada pencapaian target proyek. Proses pelacakan progres, pengambilan keputusan teknis, serta dokumentasi risiko kini dilakukan secara lebih sistematis, sehingga mendukung pengendalian kualitas dan pengelolaan proyek secara berkelanjutan.

Berdasarkan Gambar 4.4, penerapan *CSD Matrix* pada tahap brainstorming proyek *Apple Music Affiliate Landing Page* memungkinkan tim lintas fungsi untuk memetakan *Certainties*, *Suppositions*, dan *Doubts* secara jelas. Pemisahan ini membuat tim mampu menentukan prioritas eksekusi berdasarkan kepastian yang sudah terverifikasi, menguji asumsi melalui eksperimen terukur, serta merumuskan mitigasi risiko untuk hal-hal yang masih diragukan. Pendekatan ini selaras dengan beberapa area proses pada Maturity Level 3 menurut Khraiweh (2013), khususnya *Decision Analysis and Resolution*, *Risk Management*, dan *Integrated Project Management*, yang menuntut adanya keterpaduan antara perencanaan, pelaksanaan, dan kontrol proses. Dalam konteks kuesioner CMMI, aspek ini berkontribusi langsung pada peningkatan skor pertanyaan Q1, Q2, dan Q3 terkait kesesuaian perencanaan dan pelaksanaan, karena setiap keputusan kini memiliki justifikasi berbasis data dan dokumentasi yang terstruktur.

Hasil implementasi *CSD Matrix* juga mencerminkan penguatan kualitas proses bisnis tim Mamikos berdasarkan empat atribut menurut Kroeger & Davidson (2009), yaitu *suitability*, *usability*, *manageability*, dan *evolvability*. *Suitability* terjaga melalui penyusunan rencana yang relevan dengan tujuan proyek dan kebutuhan pengguna, *usability* meningkat dengan tersedianya dokumentasi yang mudah dipahami oleh semua anggota tim, *manageability* diperkuat lewat pemantauan risiko dan pelacakan progres yang sistematis, sedangkan *evolvability* terlihat dari kemampuan tim untuk menyesuaikan strategi berdasarkan hasil validasi asumsi. Dengan demikian, *CSD Matrix* tidak hanya membantu mengklarifikasi arah proyek di tahap awal, tetapi juga berkontribusi signifikan terhadap peningkatan kualitas proses bisnis secara menyeluruh.

Berdasarkan wawancara Giga menyatakan bahwa penerapan *CSD Matrix* membantu tim dalam mendokumentasikan proses pengembangan dengan lebih terstruktur melalui pemisahan antara hal-hal yang sudah pasti (*Certainties*), asumsi yang perlu divalidasi (*Suppositions*), dan hal-hal yang masih menjadi pertanyaan (*Doubts*). Pendekatan ini memudahkan tim untuk mengidentifikasi prioritas, mengurangi potensi miskomunikasi, serta memastikan setiap langkah pengembangan didasarkan pada pemahaman yang jelas dan terarah.

Berikut adalah grafik batang yang menunjukkan perbandingan persentase capaian masing-masing *Key Process Area (KPA)* sebelum dan sesudah penerapan metode *CSD Matrix*.



Gambar 4. 6 Grafik Perbandingan Persentase Berdasarkan KPA

Berikut adalah *coding* untuk pembuatan grafik batang yang ditampilkan pada tabel di bawah ini, di mana *coding* tersebut dijalankan menggunakan platform *Google Colab*.

Tabel 4. 4 Codingan Grafik Perbandingan

```
import matplotlib.pyplot as plt
import numpy as np

# Data
key_process_area = [
    "Project Planning",
    "Project Risks",
    "Data Management",
    "Stakeholder Involvement",
    "Corrective Action"
```

```

]

# Nilai rata-rata sebelum & sesudah
sebelum = [2.80, 2.6222, 2.60, 2.8333, 2.6049]
sesudah = [3.856, 3.65, 3.665, 3.7431, 3.6443]

# Membuat posisi bar
x = np.arange(len(key_process_area))
width = 0.35 # lebar bar

# Plot
fig, ax = plt.subplots(figsize=(10, 6))
bars1 = ax.bar(x - width/2, sebelum, width, label='Sebelum',
color='skyblue')
bars2 = ax.bar(x + width/2, sesudah, width, label='Sesudah',
color='orange')

# Label dan judul
ax.set_ylabel('Rata-Rata')
ax.set_title('Perbandingan Tingkat Kematangan Proses Bisnis')
ax.set_xticks(x)
ax.set_xticklabels(key_process_area, rotation=30, ha='right')
ax.legend()

# Menampilkan nilai di atas bar
def autolabel(bars):
    for bar in bars:
        height = bar.get_height()
        ax.annotate(f'{height:.2f}',
                    xy=(bar.get_x() + bar.get_width() / 2,
height),
                    xytext=(0, 3),
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(bars1)
autolabel(bars2)

plt.tight_layout()
plt.show()

```

4.7 Perancangan dan Pengembangan Sistem Android

Penulisan kode program pada aplikasi ini menggunakan bahasa pemrograman Kotlin dengan pengembangan antarmuka dilakukan melalui Android Studio, yang terdiri dari beberapa halaman utama yaitu halaman dashboard utama, halaman perhitungan sebelum dan sesudah perlakuan, serta halaman perbandingan grafik hasil perhitungan untuk memvisualisasikan perbedaan secara lebih informatif dan interaktif.

4.7.1 Interface Dashboard

Pada tahap ini, dilakukan perancangan antarmuka (*interface*) halaman *Dashboard* sebagai tampilan utama yang dilihat pengguna setelah berhasil masuk ke dalam aplikasi. Langkah pertama yang dilakukan adalah membuat *file layout* XML pada Android Studio dengan struktur dasar menggunakan *ScrollView* agar konten dapat digulir apabila melebihi ukuran layar. Di dalamnya, digunakan *ConstraintLayout* sebagai *container* utama untuk menyusun komponen UI secara fleksibel dan responsif. Komponen pertama yang ditambahkan adalah bagian profil pengguna, yang dibuat dengan *LinearLayout horizontal* berisi gambar profil (*ImageView*) dan informasi pengguna. Setiap elemen disusun dengan mempertimbangkan tata letak responsif, padding, dan warna latar belakang yang sesuai untuk memberikan tampilan antarmuka yang bersih dan ramah pengguna. Adapun potongan kode XML untuk implementasi antarmuka *Dashboard* adalah sebagai berikut:

```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/scroll_root"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true"
    android:background="#F1F1F1"
    tools:context=".ActivityHomePage">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/dashboard_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <!-- Profile Greeting -->
        <LinearLayout
            android:id="@+id/profile_section"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
```

```

        android:padding="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <ImageView
            android:id="@+id/profile_image"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:src="@drawable/ic_profile"
            android:contentDescription="Profile Image"
            android:layout_marginEnd="12dp" />

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/user_name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Halo, Andika!"
                android:textStyle="bold"
                android:textSize="16sp"
                android:textColor="#000000"/>

            <TextView
                android:id="@+id/user_status"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Selamat datang kembali"
                android:textSize="14sp"
                android:textColor="#888888"/>

        </LinearLayout>
    </LinearLayout>

```

```

<!-- Dashboard Button -->
<Button
    android:id="@+id/btn_check_schedule"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Lihat Jadwal"

app:layout_constraintTop_toBottomOf="@id/profile_section"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginTop="24dp"
    android:layout_marginStart="16dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>

```

Gambar 4. 7 Kode Untuk Dashboard



Gambar 4. 8 Tampilan Pada Dashboard

4.7.2 Interface Perhitungan Proses Bisnis

Langkah 1: Membuat Tampilan Awal Untuk Kuesioner

Langkah pertama yang dilakukan adalah merancang dan membangun tampilan awal aplikasi yang akan digunakan sebagai wadah pengisian kuesioner. Tampilan ini biasanya memuat informasi pengantar, judul seperti “Penilaian Proses Bisnis” atau “*ProjectPlanning*”, serta struktur layout yang memungkinkan pengguna mengakses dan mengisi pertanyaan dengan mudah. Desain tampilan dibuat responsif dan sederhana menggunakan elemen-elemen UI seperti *ScrollView*, *LinearLayout*, dan *TextView* untuk memastikan kenyamanan pengguna saat menjawab.

```
package com.example.cmmiSoftwareProjectManagement
import ...
class SharedPrefHelper(context: Context) {
    private val prefs: SharedPreferences =
context.getSharedPreferences("KuisisionerPrefs",
Context.MODE_PRIVATE)

    companion object {
        const val KEY_ASES_SEBELUM = "asesmen_sebelum"
        const val KEY_ASES_SESUDAH = "asesmen_sesudah"
    }

    fun saveAspekSebelum(data: FloatArray) {
        val joined = data.joinToString(separator = ",")
        prefs.edit().putString(KEY_ASES_SEBELUM, joined).apply()
    }

    fun saveAspekSesudah(data: FloatArray) {
        val joined = data.joinToString(separator = ",")
        prefs.edit().putString(KEY_ASES_SESUDAH, joined).apply()
    }

    fun getAspekSebelum(): FloatArray? {
        val data = prefs.getString(KEY_ASES_SEBELUM, null) ?: return
null
        return data.split(",").map { it.toFloat() }.toFloatArray()
    }
}
```

```

fun getAspekSesudah(): FloatArray? {
    val data = prefs.getString(KEY_ASES_SESUDAH, null) ?: return
    null
    return data.split(",").map { it.toFloat() }.toFloatArray()
}
}

```

Gambar 4. 9 Kode Tampilan Kuesioner

Langkah 2: Membuat Kuesioner/Asmen Pertanyaan CMMI

Setelah tampilan awal siap, langkah selanjutnya adalah membuat daftar pertanyaan berdasarkan standar penilaian *Capability Maturity Model Integration* (CMMI). Pertanyaan ini disusun dalam bentuk kuesioner untuk mengukur sejauh mana praktik manajemen proyek dan proses bisnis telah dilakukan oleh organisasi. Masing-masing pertanyaan dirancang dengan opsi penilaian yang dapat dipilih pengguna, biasanya menggunakan komponen *RadioGroup* atau *CheckBox* untuk memudahkan *input data*.

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Project Planning"
    android:textStyle="bold"
    android:textSize="18sp"
    android:layout_marginTop="8dp" />

<TextView
    android:text="1. Apakah Anda memantau parameter perencanaan
    proyek sesuai dengan rencana proyek?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="2. Apakah Anda memantau kemajuan proyek sesuai
    dengan jadwal?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="3. Apakah Anda mengidentifikasi perbedaan
    signifikan terhadap perkiraan jadwal rencana proyek?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

```

<TextView
    android:text="4. Apakah Anda secara berkala mengukur upaya
    aktual, biaya yang dikeluarkan, dan staf yang ditugaskan?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="5. Apakah Anda membandingkan upaya aktual, biaya,
    penugasan staf, dan pelatihan dengan anggaran rencana proyek?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="6. Apakah Anda mengidentifikasi dan menganalisis
    risiko potensial terhadap proyek?"

<TextView
    android:text="7. Apakah Anda mengidentifikasi tindakan
    pengurangan risiko dan memantau pelaksanaannya?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="8. Apakah Anda memantau keterlibatan pemangku
    kepentingan dalam proyek?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="9. Apakah Anda memantau komunikasi antara pemangku
    kepentingan?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="10. Apakah Anda memantau data proyek untuk
    keakuratan dan kelengkapannya?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="11. Apakah Anda mengidentifikasi dan memperbaiki
    ketidakkonsistenan dalam data proyek?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

```

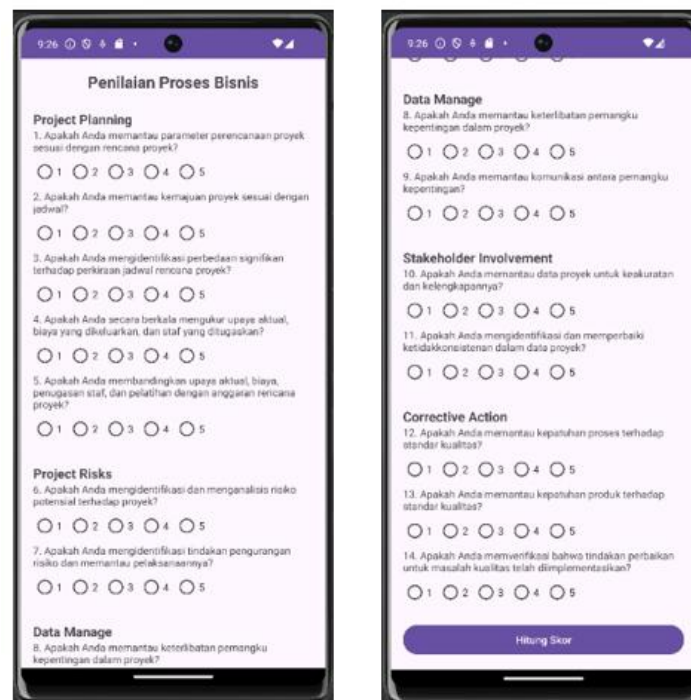
<TextView
    android:text="12. Apakah Anda memantau kepatuhan proses terhadap
standar kualitas?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="13. Apakah Anda memantau kepatuhan produk terhadap
standar kualitas?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="14. Apakah Anda memverifikasi bahwa tindakan
perbaikan untuk masalah kualitas telah diimplementasikan?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

Gambar 4. 10 Kode Untuk Pertanyaan Pengukuran Dengan CMMI



Gambar 4. 11 Tampilan Pertanyaan Sebelum Penerapan CSD Matrix Pada Apps

Langkah 3: Membuat Rumus Perhitungan CMMI

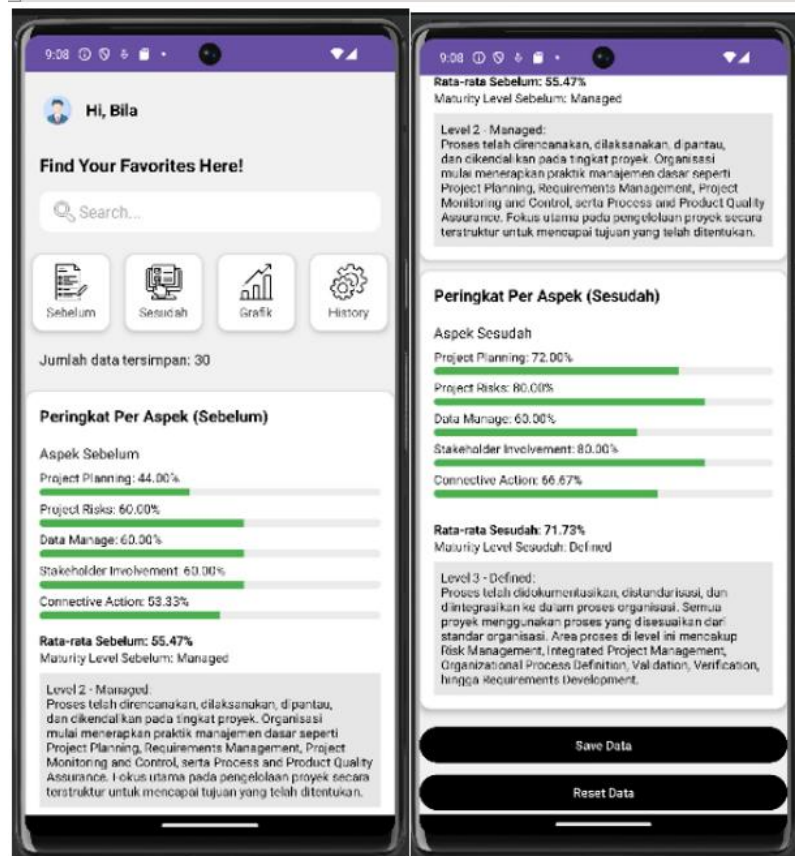
Langkah ketiga adalah menyusun logika atau algoritma perhitungan berdasarkan hasil jawaban kuesioner. Data dari jawaban-jawaban tersebut akan dikumpulkan dan diolah untuk

menghasilkan skor pada setiap aspek yang dinilai. Perhitungan ini biasanya dilakukan dengan cara mengkonversi jawaban menjadi nilai numerik, menghitung rata-rata, serta menerapkan rumus tertentu sesuai dengan pendekatan CMMI. Proses ini dilakukan di dalam kode program, memanfaatkan struktur data seperti array dan fungsi matematis. Langkah terakhir adalah melakukan perhitungan tingkat kapabilitas (*Capability level*) berdasarkan hasil penilaian sebelumnya. Tingkat kapabilitas ini akan menunjukkan sejauh mana kematangan proses yang dimiliki oleh organisasi, mulai dari level 1 (*Initial*) hingga level 5 (*Optimizing*). Hasil dari perhitungan ini akan ditampilkan kembali kepada pengguna dalam bentuk teks atau grafik sebagai laporan evaluasi. *Output* ini memberikan gambaran visual dan informatif terhadap performa proses yang telah dinilai.

```
// Hitung rata-rata per aspek (dikonversi ke persen)
val aspek1Rata = aspek1.average() / 5.0 * 100
val aspek2Rata = aspek2.average() / 5.0 * 100
val aspek3Rata = aspek3.average() / 5.0 * 100
val aspek4Rata = aspek4.average() / 5.0 * 100
val aspek5Rata = aspek5.average() / 5.0 * 100

// Hitung rata-rata keseluruhan
val rataKeseluruhan = listOf(
    aspek1Rata, aspek2Rata, aspek3Rata, aspek4Rata, aspek5Rata
).average()
// Tentukan Maturity Level
val level = when {
    rataKeseluruhan < 40 -> "Initial"
    rataKeseluruhan in 40.0..60.0 -> "Managed"
    rataKeseluruhan in 60.0..75.0 -> "Defined"
    rataKeseluruhan in 75.0..90.0 -> "Quantitatively"
    else -> "Optimizing"
}
tvOutput.text = output
```

Gambar 4. 12 Kode Perhitungan KPA dan Menentukan *Maturity level*



Gambar 4. 13 Tampilan Hasil Perhitungan CMMI Sebelum dan Sesudah Penerapan CSD Matrix

Berdasarkan sistem yang telah dibuat, diperoleh hasil perhitungan tingkat kapabilitas proses sebelum dan sesudah implementasi. Pada halaman “Sebelum”, rata-rata kapabilitas berada di angka 55,47% dengan level 2 (*Managed*), yang menunjukkan proses sudah direncanakan dan dijalankan namun masih dasar. Sementara itu, pada halaman “Sesudah”, terjadi peningkatan menjadi 71,73% dengan level 3 (*Defined*), di mana organisasi telah menggunakan data dan metrik untuk mengukur dan mengendalikan kualitas serta kinerja proses secara konsisten.

4.7.3 Interface Perbandingan Grafik

Interface Perbandingan Grafik dirancang untuk menampilkan visualisasi hasil penilaian sebelum dan sesudah asesmen CMMI dalam bentuk grafik batang dan *pie chart*. Tampilan ini memudahkan pengguna untuk membandingkan nilai dari setiap aspek seperti *Planning*, *Requirements*, hingga *Review* secara visual, sehingga perubahan atau peningkatan kinerja dapat

terlihat dengan jelas. Dengan antarmuka yang interaktif dan informatif, pengguna dapat memahami hasil asesmen secara lebih intuitif dan efisien.

```
package com.example.cmmiSoftwareProjectManagement
import ...
class GrafikPerbandinganActivity : AppCompatActivity() {

    private lateinit var barChart: BarChart
    private lateinit var pieChart: PieChart
    private lateinit var sharedPreferencesHelper: SharedPreferencesHelper

    private val aspekNames = listOf(
        "Planning", "Requirements", "Config", "Monitoring",
        "Risk", "Decision", "Analysis", "Training", "Review"
    )

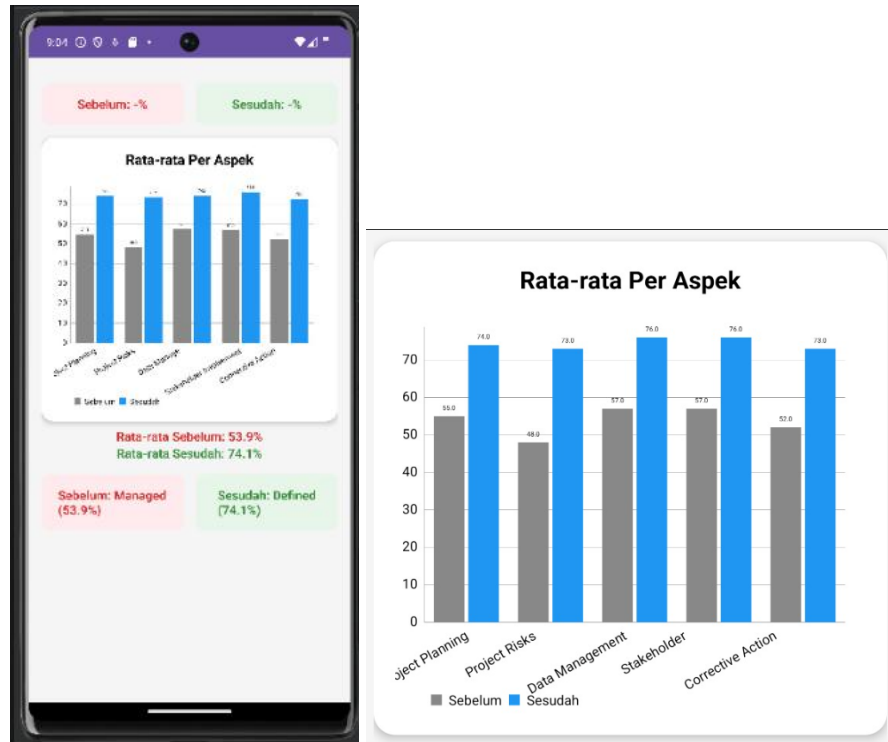
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_grafik_perbandingan)

        barChart = findViewById(R.id.barChart)
        pieChart = findViewById(R.id.pieChart)
        sharedPreferencesHelper = SharedPreferencesHelper(context = this)

        val dataSebelum = sharedPreferencesHelper.getAspekSebelum()
        val dataSesudah = sharedPreferencesHelper.getAspekSesudah()

        if (dataSebelum != null && dataSesudah != null &&
            dataSebelum.size == dataSesudah.size) {
            tampilkanGrafik(dataSebelum, dataSesudah)
            tampilkanPieChart(dataSebelum, dataSesudah)
        }
    }
}
```

Gambar 4. 14 Kode Perbandingan Grafik



Gambar 4. 15 Tampilan Perbandingan Grafik Sebelum dan Sesudah Pada Apss

4.8 Analisis Perbandingan dan Pengujian Hasil Secara Manual dan Sistem

Berdasarkan hasil pengujian terhadap sistem yang telah dikembangkan, dilakukan perbandingan antara perhitungan manual dan hasil yang ditampilkan oleh aplikasi untuk setiap aspek *Key Process Area* (KPA). Baik perhitungan manual maupun hasil yang ditampilkan aplikasi menunjukkan hasil yang sama, sehingga selisih rata-rata tercatat sebesar 20% peningkatan. Visualisasi perbandingan rata-rata skor pada aplikasi juga menunjukkan konsistensi. Selanjutnya dilakukan uji kecocokan antara hasil manual dan hasil dari sistem aplikasi untuk memastikan akurasi.

Tabel 4. 5 Perbandingan Analisis Manual dan Analisis Aplikasi Pada Perhitungan CMMI

KPA	Perhitungan Manual Sebelum	Perhitungan Manual Sesudah	Perhitungan Aplikasi Sebelum	Perhitungan Aplikasi Sesudah	Keterangan
<i>Project Planning</i>	55%	74%	55%	74%	Hasil manual dan aplikasi sama, menunjukkan konsistensi.

KPA	Perhitungan Manual Sebelum	Perhitungan Manual Sesudah	Perhitungan Aplikasi Sebelum	Perhitungan Aplikasi Sesudah	Keterangan
<i>Project Risks</i>	48%	73%	48%	73%	Hasil manual dan aplikasi sama, menunjukkan konsistensi.
<i>Data Management</i>	57%	75%	57%	76%	Hasil manual dan aplikasi hampir sama, selisih 1% sangat kecil.
<i>Stakeholder Involvement</i>	57%	76%	57%	76%	Hasil manual dan aplikasi sama, menunjukkan konsistensi.
<i>Corrective Action</i>	52%	73%	52%	73%	Hasil manual dan aplikasi sama, menunjukkan konsistensi.
Rata-rata	54%	74%	53,9%	74,1%	Rata-rata manual dan aplikasi hampir sama, perbedaan sangat kecil dan masih konsisten.

Berdasarkan hasil pengujian baik melalui perhitungan manual maupun hasil yang ditampilkan aplikasi, diketahui bahwa tidak terjadi perbedaan signifikan terhadap penentuan maturity level. Secara keseluruhan, baik manual maupun sistem menunjukkan hasil yang sama, yaitu *Maturity Levels 2 (Managed)* pada kondisi sebelum dan meningkat menjadi *Maturity Levels 3 (Defined)* pada kondisi sesudah. Hasil ini konsisten dan menunjukkan bahwa sistem mampu merepresentasikan tingkat kapabilitas proses organisasi dengan cukup akurat sesuai standar penilaian CMMI.

4.9 Pembahasan

Berdasarkan hasil analisis data yang diperoleh melalui kuesioner, wawancara, observasi, dan dokumentasi, dapat disimpulkan bahwa penerapan metode *CSD Matrix* memberikan dampak positif terhadap peningkatan kualitas proses bisnis di tim pengembang perangkat lunak Mamikos. Seluruh *Key Process Area (KPA)* dalam model *CMMI* menunjukkan adanya peningkatan skor secara signifikan, baik dari segi persentase maupun nilai rata-rata.

Penerapan metode *CSD Matrix* dalam tim pengembang perangkat lunak Mamikos menunjukkan dampak positif terhadap peningkatan kualitas proses bisnis, terutama dalam aspek dokumentasi, kolaborasi tim, dan pengambilan keputusan. Berikut adalah temuan utama yang didapatkan dalam penelitian ini dapat dilihat pada Tabel 4.6:

Tabel 4. 6 Perbandingan Kondisi Tim Pengembang Sebelum dan Sesudah Penerapan *CSD Matrix*

Aspek	Sebelum <i>CSD Matrix</i>	Sesudah <i>CSD Matrix</i>
Tingkat Kematangan Proses	<i>Maturity Levels 2 (Managed)</i> : proses belum terdokumentasi dan belum distandarisasi secara konsisten.	<i>Maturity Levels 3 (Defined)</i> : proses terdokumentasi, terukur, dan dikendalikan dengan baik.
Dokumentasi	Tidak konsisten, tersebar di banyak platform (<i>chat</i> , catatan <i>meeting</i> , dll).	Terintegrasi dalam satu dokumen <i>CSD Matrix</i> yang mudah diakses dan dilacak.
Pengambilan Keputusan	Sering tidak terdokumentasi, bergantung individu.	Kolektif dan berbasis data <i>CSD Matrix (Certainties, Suppositions, Doubts)</i> .
Refleksi & Evaluasi Proyek	Jarang dilakukan, tidak terdokumentasi dengan baik.	Menjadi rutinitas penting; digunakan untuk retrospektif, <i>onboarding</i> , dan perbaikan berkelanjutan.
Kolaborasi & Komunikasi	Kurang terstruktur, miskomunikasi sering terjadi.	Lebih terbuka, sistematis, dan berbasis konsensus bersama.
Risiko & Ketidakpastian	Sering terlewat atau tidak dikenali di awal.	Diidentifikasi lebih awal melalui <i>Matrix</i> ; menjadi bagian dari perencanaan.
Adaptasi Tim	Terbiasa dengan dokumen minimal dan cepat (<i>Scrum/Agile</i>).	Butuh waktu adaptasi awal, namun memberi manfaat jangka panjang dalam efektivitas dan keterpaduan kerja.

CSD Matrix terbukti efektif dalam meningkatkan kinerja proses tim pengembang Mamikos. Metode ini melengkapi pendekatan *Agile* dengan sistem dokumentasi yang

terstruktur dan reflektif, membantu tim mencapai *Maturity Levels: 3 Defined* dalam model CMMI. Meski awalnya butuh adaptasi, manfaat jangka panjangnya sangat signifikan dalam mendukung *Software Process Improvement (SPI)*.

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penerapan metode *CSD Matrix* memberikan dampak positif terhadap peningkatan proses pengembangan di Mamikos, khususnya dalam aspek kolaborasi, pengambilan keputusan, dan dokumentasi kerja tim. Melalui pendekatan kualitatif dan evaluasi menggunakan kerangka CMMI, ditemukan bahwa tim menunjukkan perbaikan dalam hal struktur kerja dan kesadaran terhadap risiko serta asumsi yang belum tervalidasi. Oleh karena itu, disarankan agar metode *CSD Matrix* diimplementasikan secara berkelanjutan dalam setiap fase pengembangan proyek, serta dikombinasikan dengan evaluasi rutin berbasis CMMI untuk memastikan proses bisnis berjalan lebih matang, adaptif, dan terukur.

4.10 Dampak Adopsi CSD Matrix Terhadap Proses Bisnis Tim

Adopsi *CSD Matrix* memberikan dampak yang signifikan terhadap peningkatan kualitas proses bisnis tim pengembang perangkat lunak di Mamikos. Salah satu dampak paling nyata adalah meningkatnya keteraturan dokumentasi proyek. Berikut adalah dampak utama dari adopsi *CSD Matrix* terhadap proses bisnis tim pengembang perangkat lunak di Mamikos yang memberikan kontribusi signifikan dalam meningkatkan efektivitas kerja tim:

a. **Peningkatan Kualitas Dokumentasi dan Transfer Pengetahuan**

Dengan struktur sistematis yang terdiri dari *Certainties*, *Suppositions*, dan *Doubts*, *CSD Matrix* membantu mendokumentasikan seluruh proses eksplorasi hingga pengambilan keputusan dalam satu dokumen utama. Hal ini memudahkan tim untuk meninjau keputusan sebelumnya dan mempercepat proses transfer pengetahuan, terutama saat ada pergantian anggota tim atau perubahan kebutuhan proyek.

b. **Koordinasi dan Kolaborasi Antar Fungsi Tim yang Lebih Efektif**

CSD Matrix menjadi acuan bersama yang memfasilitasi komunikasi antara berbagai fungsi dalam tim, seperti *frontend*, *backend*, QA, dan *product owner*. Dengan pemahaman yang sama atas asumsi, batasan, dan risiko sejak awal, proses diskusi menjadi lebih fokus, tanggung jawab lebih jelas, dan potensi miskomunikasi dapat diminimalkan secara signifikan.

c. **Evaluasi Proses yang Lebih Terstruktur dan Berbasis Data**

CSD *Matrix* menjadi acuan bersama yang memfasilitasi komunikasi antara berbagai fungsi dalam tim, seperti *frontend*, *backend*, QA, dan *product owner*. Dengan pemahaman yang sama atas asumsi, batasan, dan risiko sejak awal, proses diskusi menjadi lebih fokus, tanggung jawab lebih jelas, dan potensi miskomunikasi dapat diminimalkan secara signifikan.

4.11 Tantangan dan Kendala Implementasi CSD *Matrix*

Meskipun memberikan banyak dampak positif, implementasi CSD *Matrix* juga menghadapi sejumlah tantangan dan kendala, terutama pada tahap awal penerapan. Salah satu tantangan utama adalah waktu adaptasi. Beberapa kendala utama yang dihadapi sebagai berikut:

a. Adaptasi dan Pemahaman Awal

Tim membutuhkan waktu untuk memahami struktur dan fungsi dari CSD *Matrix*, terutama jika sebelumnya terbiasa dengan metode yang lebih informal seperti Scrum. Proses adaptasi ini bisa memperlambat alur kerja di tahap awal implementasi.

b. Inkonsistensi dan Tekanan Proyek

Penerapan CSD *Matrix* tidak selalu konsisten antar tim atau proyek. Tekanan deadline dan keterbatasan sumber daya membuat sebagian tim kembali ke kebiasaan lama yang cepat namun kurang terdokumentasi.

c. Keberlanjutan dan Dukungan Organisasi

Tanpa dukungan dari manajemen, pelatihan berkala, dan integrasi dengan *tools* seperti Jira, CSD *Matrix* berisiko menjadi metode sesaat. Diperlukan pendekatan manajemen perubahan agar metode ini menjadi bagian dari budaya kerja tim secara berkelanjutan.

4.12 Keterbatasan Penelitian

Hasil penelitian ini memberikan gambaran yang positif terhadap penerapan CSD *Matrix* dalam proses bisnis tim pengembang perangkat lunak, namun tetap memiliki beberapa keterbatasan yang perlu dicermati, antara lain:

1. Hasil peningkatan tingkat kematangan proses belum dapat dijadikan ukuran mutlak keberhasilan karena belum diuji dalam jangka panjang dan di berbagai jenis proyek yang berbeda.

2. Hasil evaluasi lebih menekankan pada persepsi dan pengalaman partisipan terhadap perubahan proses, sehingga belum dilengkapi dengan bukti kuantitatif berbasis metrik produktivitas atau kualitas kode secara teknis.
3. Peningkatan skor dalam *Key Process Area (KPA)* pada model *CMMI* menunjukkan tren positif, namun belum dapat menggambarkan secara menyeluruh efektivitas metode *CSD Matrix* apabila diterapkan dalam tim dengan budaya kerja atau tingkat kompleksitas yang berbeda.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uraian dari bab-bab sebelumnya, maka kesimpulan penelitian ini adalah:

1. Adopsi *CSD Matrix* membantu tim pengembang Mamikos mengatasi tantangan dokumentasi keputusan, koordinasi lintas divisi, dan manajemen risiko yang sebelumnya belum terstruktur. Melalui pemetaan *Certainties*, *Suppositions*, dan *Doubts*, tim dapat membedakan hal yang sudah pasti, asumsi yang perlu divalidasi, serta keraguan yang membutuhkan mitigasi, sehingga prioritas kerja menjadi jelas, miskomunikasi berkurang, dan dokumentasi keputusan lebih rapi serta mudah ditelusuri.
2. Hasil Evaluasi menggunakan CMMI menunjukkan peningkatan rata-rata seluruh KPA dari 2,69 (54%) pada pra-implementasi menjadi 3,71 (74%) pada pasca-implementasi. Hasil ini merefleksikan pergeseran tingkat kematangan proses dari *Maturity Level 2 (Managed)* ke *Maturity Level 3 (Defined)* artinya proses tidak hanya dikelola, tetapi juga terdefinisi, terdokumentasi, dan distandarisasi di tingkat organisasional dengan prosedur yang dapat ditailor untuk tiap proyek yang pada gilirannya meningkatkan kualitas proses bisnis tim secara konsisten.
3. Penggunaan *CSD Matrix* meningkatkan kualitas refleksi tim dalam sesi evaluasi dan *brainstorming*, sekaligus mendukung *onboarding* anggota baru. Seluruh hasil eksplorasi terdokumentasi di satu tempat sehingga mudah diakses, diperbarui, dan dimanfaatkan untuk monitoring serta pengambilan keputusan. Dibandingkan *Scrum* dalam kerangka *Agile*, *CSD Matrix* menawarkan kedalaman analisis dan struktur dokumentasi yang lebih kuat, membantu mengatasi keterbatasan dokumentasi sekaligus menjaga stabilitas proses pada proyek dengan kompleksitas tinggi atau ketidakpastian eksternal.

5.2 Saran

Berdasarkan hasil penelitian, maka beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dan implementasi *CSD Matrix* secara optimal antara lain:

1. Tim pengembang yang menggunakan *CSD Matrix* direkomendasikan untuk mengintegrasikan metode ini dengan alat pengambilan keputusan lainnya, seperti RACI

Matrix atau *PICK Chart*, guna memperkuat proses validasi dan eksekusi, terutama dalam proyek-proyek dengan banyak pemangku kepentingan.

2. Penelitian selanjutnya disarankan untuk melibatkan lebih banyak informan dari berbagai posisi dalam tim pengembang, termasuk *product owner*, *QA engineer*, dan *UI/UX designer*, agar dapat menggali perspektif yang lebih beragam dan memperoleh hasil yang lebih representatif.
3. Studi lanjutan dapat memperluas cakupan objek penelitian pada perusahaan teknologi lain atau sektor industri berbeda, untuk mengetahui sejauh mana efektivitas CSD *Matrix* dapat diadaptasi di luar konteks *startup* digital atau produk afiliasi.

DAFTAR PUSTAKA

- Abdussamad, Z. (2022). *Buku Metode Penelitian Kualitatif*.
- Aguilar, A. I., & Li, R. (2023). Responsibility Assignment *Matrix* in Responding to Critical Outages: A Case Study of IT Incident Management Activity Process Design. *Proceedings of the 2023 5th International Conference on Management Science and Industrial Engineering*, 1–8.
- Ajayi, W. (2021). A Comparative Analysis of Agile Methods and the Capacity Maturity Model Integration (CMMI). *Asian Journal of Computer Science and Technology*, 10(2), 25–33.
- Akbar, M. A., Sang, J., Khan, A. A., Fazal-E-Amin, Nasrullah, Shafiq, M., Hussain, S., Hu, H., Elahi, M., & Xiang, H. (2018). Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model. *IEEE Access*, 6, 4811–4823. <https://doi.org/10.1109/ACCESS.2017.2787981>
- Albuquerque, R., Santos, G., Malucelli, A., & Reinehr, S. (2021). Software Process Improvement Programs: What Are The Pitfalls That Lead to Abandonment? *Journal of Software Engineering Research and Development*, 9, 11–15.
- Aljedaibi, W., & Alsulami, A. A. (2021). A CMMI-Based Method for Software Development Process Assessment: Applying CMMI Process in Saudi Arabia. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, 12(4), 1–10. <https://doi.org/10.14456/ITJEMAST.2021.71>
- Antara. (2020, July 6). *Mamikos Gandeng MoEngage Tingkatkan Sewa Kamar Lebih Dari 20%*. Antara: Kantor Berita Indonesia.
- Ashrafi, N. (2003). The Impact of Software Process Improvement on Quality: In Theory And Practice. *Information & Management*, 40(7), 677–690. [https://doi.org/https://doi.org/10.1016/S0378-7206\(02\)00096-4](https://doi.org/https://doi.org/10.1016/S0378-7206(02)00096-4)
- Basuki, K. T., & Angelina, W. J. (2023). Simulation and Visualization of TSP Using Ant Colony Optimization. *Journal of Data Science*, 2023(22), 1–10.
- Bubevski, V. (2014). A Novel Approach to Software Quality Risk Management. *Software Testing*, 24. <https://doi.org/10.1002/stvr.1488>
- Castro, L. F., & Peña, E. E. (2023). Integration of The Software Engineering Theory And The Quality Function Deployment Methodology For Supporting The Measurement

- of the State of Progress in Software Projects. *Ingenierías USBMed*, 14(1), 48–56.
<https://doi.org/10.21500/20275846.6039>
- Contreras, J. Á. V., Arias-Masa, J., Hidalgo-Izquierdo, V., & Martín-Espada, R. (2021). Personal Software Process: A Study from an Academic Perspective. *World Conference on Qualitative Research*, 143–154.
- Creately. (2024, June 27). *CSD Matrix Explained: Amplify Certainties, Reduce Doubts in Projects*. Creately. https://creately.com/guides/csd-Matrix/?utm_source
- Ctad. (2025). *CSD Matrix*. Call to Action Digital. https://www.ctad.io/glossary/csd-Matrix?utm_source
- Debou, C., Lipták, J., & Schippers, H. (1994). Decision Making for Software Process Improvement: A Quantitative Approach. *Journal of Systems and Software*, 26(1), 43–52. [https://doi.org/https://doi.org/10.1016/0164-1212\(94\)90094-9](https://doi.org/https://doi.org/10.1016/0164-1212(94)90094-9)
- DeFranco, J. F., & Laplante, P. A. (2017). Review and Analysis of Software Development Team Communication Research. *IEEE Transactions on Professional Communication*, 60(2), 165–182. <https://doi.org/10.1109/TPC.2017.2656626>
- Dingsøyr, T., Schneider, P., Bergersen, G. R., & Lindsjörn, Y. (2024). Challenges in Understanding the Relationship between Teamwork Quality and Project Success in Large-Scale Agile Projects. *Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering*, 51–56. <https://doi.org/https://doi.org/10.48550/arXiv.2401.17725>
- Gavriel, S. (2023). *Design and Implementation of an Assessment Method based on the Capability Maturity Model Integration (CMMI)*. <http://essay.utwente.nl/97425/>
- George, M. de O., Andrine, N. C., Lamego, B. S., Monteiro, I. T., Gonçalves, E. J. T., & Basilio, A. F. (2022). Reporting The Application of User Experience Tools and Proxy Users in an Industrial Process Based on Double Diamond. *International Conference on Human-Computer Interaction*, 1, 57–74. https://doi.org/https://doi.org/10.1007/978-3-031-05897-4_5
- Guimaraes, F., & Aela. (2023, February 6). *CSD Matrix: A Tool To Boost Your Project*. *Aela*, 1–1. https://www.aela.io/en/blog/all/csd-Matrix-tool-boost-project?utm_source
- Haneef, H., Abbas, A., & Khan, S. U. (2024). A Process Improvement Model for Software Organizations during COVID-19 Pandemic. *Proceedings of 36th International Conference On*, 97, 20–29.

- Kaley, A. (2023). *CSD Matrix: framework and template for shared understanding*. Dover, USA: Nielsen Norman Group. <https://www.nngroup.com/articles/csd-Matrix/>
- Kroeger, T. A., Davidson, N. J., & Cook, S. C. (2014). Understanding The Characteristics of Quality For Software Engineering Processes: A Grounded Theory Investigation. *Information and Software Technology*, 56(2), 252–271. <https://doi.org/https://doi.org/10.1016/j.infsof.2013.10.003>
- Kroeger, T., & Davidson, N. (2009). A Perspective Based Model of Quality for Software Engineering processes. *2009 Australian Software Engineering Conference*, 152–161.
- Lidyana, V. (2022, June 29). Menggeliat Lagi, Permintaan Sewa Indekos Melonjak 125 Persen. *IDN Times*, 1. https://www.idntimes.com/business/economy/menggeliat-lagi-permintaan-sewa-indekos-melonjak-125-persen-00-3m8tp-4zwpjr?utm_source
- Mamikos. (2020, September). Pemilik Kost Wajib Tahu, Ini Alasan Orang Lebih Suka Cari Kost Via Online. *Mamikos*. https://mamikos.com/info/alasan-orang-lebih-suka-cari-kost-via-online/?utm_source
- Maulana, Y. M. (2023). Model of Business Process Improvement in Organizations Based on the Business Process Improvement Approach. *Journal of Advances in Information and Industrial Technology*, 5(2), 79–92. <https://doi.org/https://doi.org/10.52435/jaiit.v5i2.386>
- Mayer, R., & Sechser, B. (2022). Challenges with Multi-PAM SPICE Assessments. *Systems, Software and Services Process Improvement: 29th European Conference, EuroSPI 2022, Salzburg, Austria, August 31–September 2, 2022, Proceedings*, 271.
- Methawachananont, A., Buranarach, M., Amsuriya, P., Chaimongkhon, S., Krairaksa, K., & Supnithi, T. (2020). Software Process Capability Self-Assessment Support System Based On Task And Work Product Characteristics: A Case Study Of ISO/IEC 29110 Standard. *IEICE Transactions on Information and Systems*, 103(2), 339–347.
- Ojha, T. R. (2022). Role of Capability Maturity Model Integration (CMMI) in Software Process Improvement. *Journal of Advancement in Software Engineering and Testing*, 5(2).
- Omol, E. J., Mburu, L. W., & Abuonji, P. A. (2025). Digital Maturity Assessment Model (DMAM): assimilation of Design Science Research (DSR) and Capability Maturity Model Integration (CMMI). *Digital Transformation and Society*, 4(2), 128–152.

- Rajab, R., & Alnoukari, M. (2025). DevOps Integration With Capability Model Maturity Integration: A Systematic Mapping Review. *IEEE Access*, 13, 31829–31841. <https://doi.org/10.1109/ACCESS.2025.3542630>
- Rodriguez, M., Verdugo, J., Pino, F., Delgado, B., & Piattini, M. (2021). Software development process assessment with MMIS v. 2, an ISO/IEC 33000-Based Model. *IT Professional*, 23(6), 17–23.
- Rukhmana, T., Darwis, D., IP, S., Alatas, A. R., SE, M. M., Tarigan, W. J., Mufidah, Z. R., Muhamad Arifin, M. H. I., Cahyadi, N., & S ST, M. M. (2022). *Metode Penelitian Kualitatif*. CV Rey Media Grafika.
- Rukita. (2024, February). *The Rise of Proptech Companies in Indonesia*. PR Marketing. https://www.rukita.co/stories/proptech-company-in-indonesia-apt-higer?utm_source
- Ruzhnikov, A., & Prasetyo, A. (2024). Enhancing the Well Engineering Management System (WEMS) Through a Capability Maturity Model Integration (CMMI)-Based Approach. *Abu Dhabi International Petroleum Exhibition and Conference*, D011S002R003. <https://doi.org/https://doi.org/10.2118/221827-MS>
- Santos, R., Cunha, F., Rique, T., Perkusich, M., Almeida, H., Perkusich, A., & Costa, Í. (2023). A Comparative Analysis Of Agile Teamwork Quality Instruments In Agile Software Development: A Qualitative Approach. *JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS*, 18(2), 153–164. <https://doi.org/10.18293/DMSVIVA2023-217>
- Soto, B. P., & Rafael, G. R. (2020). Personal Software Process (PSP) Skills for the Software Industry in Latin America. *Industrial Data*, 23(1), 1.
- Stravinskiene, I., & Serafinas, D. (2020). The Link Between Business Process Management and Quality Management. *Journal of Risk and Financial Management*, 13(10), 225. <https://doi.org/https://doi.org/10.3390/jrfm13100225>
- Strode, D., Dingsøyr, T., & Lindsjorn, Y. (2022). A Teamwork Effectiveness Model for Agile Software Development. *Empirical Software Engineering*, 27(2), 56. <https://doi.org/10.1007/s10664-021-10115-0>
- Sukses, O. (2022, September 15). Mengenal Maria Regina Anggit, Jadi Juragan Indekos Berkat Inovasi di Mamikos. *Kumparan.Id*, 1. https://kumparan.com/profil-orang-sukses/mengenal-maria-regina-anggit-jadi-juragan-indekos-berkat-inovasi-di-mamikos-1yrASr6ebB1?utm_source

- Verdugo Lara, J., Pino, F., Delgado, B., Rodríguez Monje, M., & Piattini Velthuis, M. G. (2021). *Software Development Process Assessment With MMIS v. 2, an ISO/IEC 33000-Based Model*.
- Washizaki, H., & Olszewska, J. I. (2024). *Guide to the Software Engineering Body of Knowledge v4. 0: Vol. 4.0*. IEEE Computer Society. <https://research-portal.uws.ac.uk/en/publications/guide-to-the-software-engineering-body-of-knowledge-v40>
- Wijaya, I. S., Bora, M. A., Permatasari, R. D., & Zulkarnaian, M. A. F. (2025). Application of RACI Framework for Optimization of Key Performance Indicators (KPI) in Manufacturing Companies. *Dinasti International Journal of Education Management and Social Science*, 6(4). <https://doi.org/https://doi.org/10.38035/dijemss.v6i4>
- Yassien, E. (2020). The Challenges of Capability Maturity Model Integration Application In The Dynamic environment. *International Journal of Information Systems and Change Management*, 12(1), 17–34. <https://doi.org/https://doi.org/10.1504/IJISCM.2020.112045>

LAMPIRAN

Lampiran Lembar Hasil Wawancara

Stakeholder Interview : Giga Prakosa Hikmata

Role : *Manager Software Engineering Mamikos*

Wawancara Penerapan CSD Matrix

No	Key Process Area	Pertanyaan Wawancara	Jawaban
1	Project Planning	Apakah perencanaan proyek dilakukan di awal proyek (tujuan, scope, estimasi waktu)?	Ya.
2		Adakah role/person khusus yang menyusun perencanaan proyek?	Tidak ada secara khusus.
3		Apakah perencanaan tersebut disusun dalam dokumen khusus?	Tidak ada secara khusus.
4	Project Risks	Bagaimana tim mencatat dan memantau kebutuhan proyek?	Menggunakan dokumen seperti google docs / slides lalu diterjemahkan menjadi Jira task.
5		Apakah ada mekanisme/tools untuk mengelola perubahan requirement secara terkontrol?	Belum ada, hanya secara komunikasi verbal saja (<i>via chat atau meeting</i>).
6		Apakah kebutuhan proyek terdokumentasi dalam bentuk khusus?	Ya, dalam bentuk google docs / slides.
7	Data Management	Apakah versi dokumen atau <i>output</i> teknis disimpan dan bisa ditelusuri kembali?	Ya.
8		Adakah role/person khusus yang bertanggung jawab dalam pengelolaan versi dokumen?	Tidak ada secara khusus.
9	Stakeholder Involvement	Bagaimana tim memantau progres proyek secara berkala?	Melalui <i>daily standup meeting & weekly sync</i> .
10		Adakah role/person yang secara rutin melakukan pelacakan progres proyek?	Oleh PM (<i>stakeholder</i>) & EM (<i>tech leaders</i>).
11		Apakah ada sistem atau template khusus untuk pelacakan progres?	Ada, menggunakan Jira.
12	Corrective Action	Apakah risiko proyek pernah dibahas secara eksplisit dalam tim?	Ya, namun tidak terlalu mendetail.
13		Jika ya, di tahap apa dan oleh siapa risiko tersebut biasanya dibahas?	Di tahap awal ketika membahas requirement <i>Project</i> dan estimasi pengerjaan.
14		Apakah risiko-risiko yang muncul terdokumentasi dengan rapi?	Tidak secara khusus.

No	Key Process Area	Pertanyaan Wawancara	Jawaban
15	Decision Documentation	Bagaimana cara tim mengambil keputusan penting selama proyek berlangsung?	Melalui diskusi via <i>meeting</i> ataupun <i>chat</i> .
16		Apakah keputusan penting tersebut biasanya dicatat atau direkap?	Tidak secara khusus.

Lampiran Lembar Kuesioner

Pertanyaan

Jawaban

Setelan

1. Apakah Anda memantau parameter perencanaan proyek sesuai dengan rencana proyek?

1

2

3

4

5

☐

☐

☐

☐

☐

2. Apakah Anda memantau kemajuan proyek sesuai dengan jadwal?

1

2

3

4

5

☐

☐

☐

☐

☐

3. Apakah Anda mengidentifikasi perbedaan signifikan terhadap perkiraan jadwal rencana proyek?

1

2

3

4

5

☐

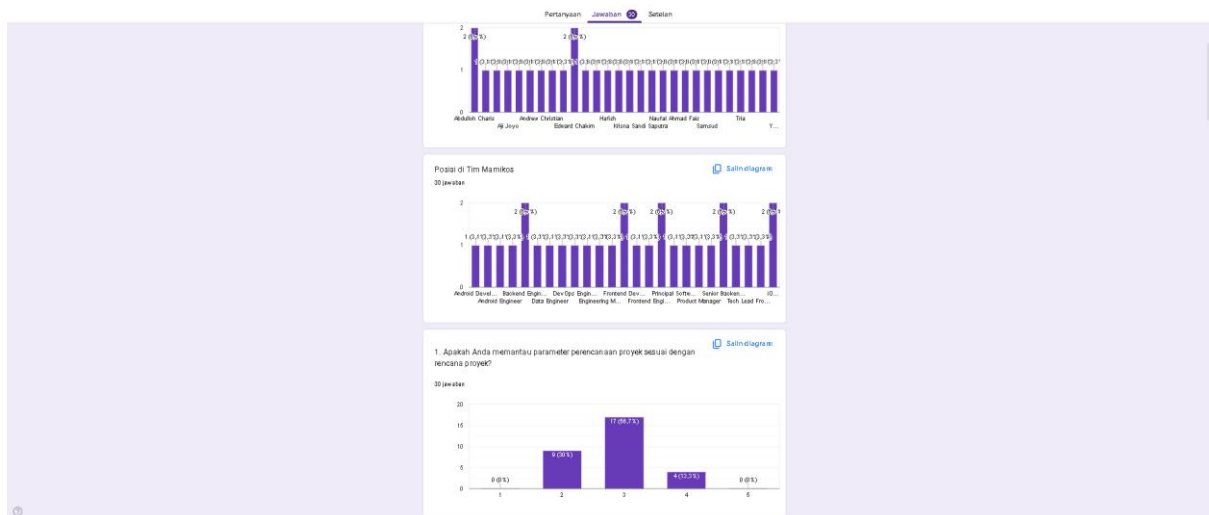
☐

☐

☐

☐

Lampiran Diagram Kuesioner



Lampiran Data

7/9/2025 16:35:05	Giga Polosa Hôma	Product Manager	4	5	3	3	3	3
7/9/2025 12:14:53	Tia	Product Manager	3	2	2	2	2	2
7/9/2025 23:40:07	Allan	Frontend Engineer Manager/DM	3	3	3	3	3	2
7/9/2025 23:57:26	Edw and Chaelen	Core Frontend Backend	3	3	2	3	3	3
7/9/2025 00:35:34	Naji	Backend Engineer	3	3	3	3	3	2
7/9/2025 05:39:28	Halah	Android Eng	3	2	3	3	2	3
7/9/2025 02:20:43	Al-Joon	iOS Engineer	3	3	4	3	3	3
7/9/2025 03:40:43	Siamed	iOS Engineer	5	2	3	3	2	1
7/9/2025 04:40:09	Kirwa Sandi	Frontend Developer	3	2	2	3	2	1
7/9/2025 05:07:06	Appenji	DevOps Engineer	3	4	2	3	2	1
7/9/2025 09:46:47	Indra S Trone	Quality Assurance Engineer Lead	5	3	3	2	3	3
7/9/2025 18:47:11	Abdullah Chavis	Senior Backend Engineer /Spec	2	3	2	2	2	3
7/9/2025 18:43:25	Yusef Nourine Naji	Backend Engineer	3	3	3	3	3	3
7/9/2025 18:51:10	Angga Bhuw Sanj	Backend Engineer	3	3	3	2	3	3
7/9/2025 19:13:32	Sandi-Budhyarto	Backend Developer	2	2	2	3	2	3
7/9/2025 19:54:35	Fajar Norman	Frontend Engineer	3	3	2	3	2	3
7/9/2025 19:56:22	Edw and Chaelen	Frontend /Backend Engineer	2	2	2	3	2	3
7/9/2025 19:58:22	Andrew Christian	Test Lead /Frontend	3	2	2	2	2	2
7/9/2025 19:59:55	Christiana Budisaputra	iOS Engineer	3	2	3	2	4	3
7/9/2025 20:00:19	Paula Paramita	iOS Engineer	3	3	3	3	3	1
7/9/2025 20:02:40	Andika Dita Parama	Android Developer	3	3	3	3	3	3
7/9/2025 20:05:45	Nuzul Al-Muhamad Fau	Android Engineer	3	3	3	3	3	2
7/9/2025 20:06:45	Michael Patrick	Engineering Manager (Mobile)	3	3	3	3	3	3
7/9/2025 20:08:08	Vahid Mansour	Data Scientist	3	3	3	3	3	3
7/9/2025 20:09:17	Ayung Dwi Cahyo	Data Engineer	3	2	3	3	3	3
7/9/2025 20:18:47	Tirta Nugro	Principal Sd /Frontend	3	3	3	3	3	3
7/9/2025 20:20:00	Ummata Aditya	Senior Frontend Engineer	2	3	3	3	3	3
7/9/2025 20:15:12	Kirwa Sandi Saputra	Frontend Developer	3	3	3	3	3	3
7/9/2025 20:16:11	Sholikhah Huda	Principal Software Engineer /Block	2	2	2	3	2	3
7/9/2025 20:17:06	Abdullah Chavis	Principal Backend Engineer /The user	2	2	2	2	2	3

Posisi/tingkat	Nama	Posisi di Tim Managemen	1. Peran dan tanggung jawab manajerial	2. Peran yang bertanggung jawab	3. Dokumen perencanaan proyek	4. Indikator keberhasilan proyek	5. Indikator keberhasilan proyek	6. Indikator keberhasilan proyek	7. Indikator keberhasilan proyek
7/9/2025 18:35:05	Giga Prakosa Hikmat	Engineering Manager	4	4	4	4	5	4	4
7/9/2025 23:24:19	Tia	Product Manager	4	4	4	4	5	4	4
7/9/2025 23:40:07	Allan	Frontend Engineer Manager EM-Frc	4	5	5	5	4	4	4
7/9/2025 23:57:26	Edward Chakim	Core Frontend Backend	4	4	4	4	5	5	5
7/10/2025 0:08:35	Naki	backend engineer	4	5	5	5	5	5	5
7/10/2025 0:18:39	Hafidh	Android Eng	4	4	4	4	5	4	5
7/10/2025 0:28:43	Al-Joyo	iOS Engineer	5	5	5	5	5	4	5
7/10/2025 0:38:43	Samsud	iOS Engineer	5	5	5	5	4	5	5
7/10/2025 0:48:09	Kisna Sandi	Frontend Developer	4	5	5	5	5	5	5
7/10/2025 0:57:05	Ageng P	DevOps Engineer	4	4	4	4	4	5	5
7/10/2025 1:06:47	Indra E Touwe	Quality Assurance Engineer Lead	5	5	5	5	5	5	5
7/13/2025 19:47:11	Abdullah Charis	Senior Backend Engineer / Squad L	3	3	4	4	4	3	3
7/13/2025 19:49:25	Yusep Noverto Nugro	Backend Engineer	3	3	4	4	5	5	5
7/13/2025 19:51:10	Angga Bayu Setiati	Backend Engineer	4	4	4	4	3	4	4
7/13/2025 19:53:12	Sandy Budiyanto	Backend Developer	3	3	5	5	4	4	4
7/13/2025 19:54:35	Fajar Norman	Frontend Engineer	4	4	4	4	3	5	5
7/13/2025 19:56:22	Edward Chakim	Frontend / Backend Engineer	4	4	4	4	4	4	4
7/13/2025 19:58:12	Andrew Christian	Tech Lead Frontend	4	4	4	4	3	4	4
7/13/2025 19:59:55	Christianto Budi Saputra	iOS Engineer	3	3	4	4	4	4	4
7/13/2025 20:01:18	Putra Adi Pratama	iOS Engineer	3	4	4	4	5	3	3
7/13/2025 20:02:43	Andika Eka Pratama	Android Developer	3	4	4	4	3	5	5
7/13/2025 20:05:11	Nasrul Ahmad Fauz	Android Engineer	4	4	4	4	4	4	4
7/13/2025 20:08:45	Michael Patrick	Engineering Manager (Mobile)	5	4	5	5	4	4	4
7/13/2025 20:08:08	Valerie Pramesti	Data Scientist	3	3	4	4	3	5	5
7/13/2025 20:09:17	Agung Dwi Cahyo	Data Engineer	3	3	3	3	4	3	3
7/13/2025 20:10:47	Ti Hargianto	Principal Web Frontend	4	3	4	4	5	4	4
7/13/2025 20:12:00	Ummu Almi	Senior Frontend Engineer	5	4	5	5	4	4	4
7/13/2025 20:13:12	Kisna Sandi Saputra	Frontend Developer	4	4	4	4	5	5	5
7/13/2025 20:15:11	M. Sholikhul Huda	Principal Software Engineer (Backend)	4	4	4	4	3	5	5
7/13/2025 20:17:04	Abdullah Charis	Senior Backend Engineer / Squad L	4	4	3	3	5	5	5

Lampiran Perhitungan

Perhitungan Sebelum Penerapan CSD Matrix

No	Key Process Area	Rata-Rata	Persentase
1	ProjectPlanning	2,8	56%
2	Project Risks	2,622222222	52%
3	Data Management	2,6	52%
4	Stakeholder Involvement	2,833333333	57%
5	Corrective Action	2,604938272	52%
Total		2,692098765	54%
Maturity Levels: 2 Managed			

Perhitungan Setelah Penerapan CSD Matrix

No	Key Process Area	Rata-Rata	Persentase
1	ProjectPlanning	3,856	77%
2	Project Risks	3,65	73%
3	Data Management	3,665	73%
4	Stakeholder Involvement	3,7431	75%
5	Corrective Action	3,6443	73%
Total		3,71168	74%
Maturity Levels : 3 Defined			

B	C	D	E	F	G	H	I
Profil 1. Proyek, didefinisikan sebagai: 2. Peran yang bertanggung jawab manajerial; 3. Dokumen perencanaan proyek disusun sesuai 4. Keberhasilan proyek telah dicapai dan 5. Perubahan keberhasilan proyek, diidentifikasi dan 6. Dokumentasi keberhasilan proyek tersusun dan 7. Versi dokumen dan							
Aji Iyora	iOS E	3	4	3	3	3	3
Samsud	iOS E	4	5	3	3	2	3
Kisna Sandi	Front	3	4	2	3	2	2
Ageng P	DevD	4	4	2	3	2	2
Indra S Touwe	Quali	4	5	3	3	2	3
Abdullah Charis	Senio	2	2	3	3	2	2
Yusep Noverto Nugro	Back	2	3	3	3	4	2
Angga Bayu Setiati	Back	3	3	3	2	4	2
Sandy Budiyanto	Back	3	3	2	3	3	3
Fajar Norman	Front	3	3	3	2	3	2
Edward Chakim	Front	2	2	3	3	2	2
Andrew Christian	Tech	3	2	3	3	2	3
Christianto Budi Saputra	iOS E	3	2	3	2	4	3
Putra Adi Pratama	iOS E	3	3	3	3	2	1
Andika Eka Pratama	Andri	3	3	3	3	3	3
Nasrul Ahmad Fauz	Andri	2	2	2	2	3	3
Michael Patrick	Engin	3	3	3	2	3	3
Valerie Pramesti	Data I	3	3	3	2	3	3
Agung Dwi Cahyo	Data I	3	3	3	3	3	3
Ti Hargianto	Princi	2	2	3	3	2	2
Ummu Almi	Senio	2	3	3	3	3	2
Kisna Sandi Saputra	Front	3	3	3	3	3	3
M. Sholikhul Huda	Princi	2	2	2	3	3	3
Abdullah Charis	Senio	2	2	2	2	3	3
No	Key Process Area	Rata-Rata		Persentase		Maturity Levels	
1	Project Planning	2,8		56%		Initial	< 40%
2	Project Risks	2,622222222		52%		Managed	40-60%
3	Data Management	2,6		52%		Defined	>60-75%
4	Stakeholder Involvement	2,833333333		57%		Quantitatively Managed	>75-90%
5	Corrective Action	2,604938272		52%		Optimizing	>90%
Total		2,692098765		54%			
		Maturity Levels: 2. Managed					

Lampiran Dokumentasi Wawancara dan Penerapan CSD Matrix

Nabila Putri Ocktavia (Anda, presenter) | Audio presentasi | Hentikan presentasi

CSD Matrix

Certainties	Suppositions	Doubts
We know or understand that...	We suppose that...	We have doubts about...
		

CSD matrices organize information about users and the organization into 3 columns: Certainties, Suppositions, and Doubts. Use the framework to track acquired knowledge, new hypotheses, and lingering uncertainties.

Free Downloads

- Excel Template - NN/g - CSD Matrix (XLSX)
- Miro Template - NN/g - CSD Matrix (RTB)

Related Courses

- UX Roadmaps**
Set a strategic, practical plan to align, prioritize, and communicate UX work.
MANAGEMENT
- Design Thinking Essentials**
Unearth user pain points to drive breakthrough design concepts.
INTERACTION
- Lean UX and Agile**
Strategies to stay user-centered in fast-paced Agile environments.

17.26 | vqf-brkw-rig

meet.google.com/vqf-brkw-rig



Giga Prakosa Hikmata



Nabila Putri Ocktavia

17.24 | vqf-brkw-rig

Giga

Arian

Tri

Ed

Zach

White Room

Map view

5

CSD Matrix_Implementasi di MAMIKOS

File Edit Tampilan Susunan Format Data Alat Bantuan

Menu 100% Hanya lihat

A1C1 Nielsen Norman Group CSD Matrix Template

	A	B	C	D	E	F	G	H	I	J	K
1	Nielsen Norman Group CSD Matrix Template © See template directions at nngroup.com/articles/csd-matrix										
2	Date : 01 Feb 2025 - 28 Feb 2025										
3	[CSD Matrix for " Apple Music Affiliate Landing Page in Mamikos Web"]										
4	Certainties					Suppositions					
5	We know or understand that...					We suppose that...					We have doubt...
6	Landing Page Purpose: The Apple Music Affiliate Landing Page is meant to drive new Apple Music subscriptions through Mamikos, targeting students and general users.					Student Users Will Drive Higher Conversion: Since the student plan offers a lower price (IDR 35,000), students are more likely to subscribe.					Will Mamikos Users - Mamikos priority
7	Affiliate Model & Revenue: - Mamikos earns 91% commission on the first subscription month. - Commission rates: IDR 31,850/student and IDR 50,050/individual. - Revenue validation takes 2-10 days, with the first payment arriving within 45-60 days after reaching a \$30 threshold.					Social Media Will Be a Strong Traffic Driver: Given Mamikos' active audience on Instagram and TikTok, these channels may bring significant traffic.					Will the Commission - Is the 91% commi
8	All CTA clicks on the landing page will be tracked. - Events like "[User] Partnership Click" will be logged with parameters: partnership: "apple-music" and source: "navbar", "header_banner", "subscribe_step". - Tracking data is stored in internal trackers and Partnerize (affiliate tracking platform), with data refreshing every 30 minutes.					Push Notifications Will Be Effective: Since Mamikos already engages users via PN, this could be a key conversion channel.					What If Apple Chan - Apple might chan
9	Landing Page Components: - Header Navbar (sticky like Mamikos' main page). - Mobile Menu for better accessibility. - Header Banner Section for promotions. - Subscription Step Section guiding users through Apple Music sign-up. - CTA Buttons leading to Apple Music sign-up. EAD Section outlines Apple Music subscription, eligibility, and promotional entries					LP Performance Depends on Apple's Approval Speed: If Apple delays approvals for branding assets, it may affect the LP launch timeline.					How Will We Handl - If internal tracker t

Iterasi 1 Iterasi 2 Iterasi 3

Giga please chat on Slack

5