

Nama	Kategori	Kolom	Sub-judul	Dibuat (UTC)	Checklist lengkap	Riwayat pergerakan	Feedback	Motivation	Appreciation
MHS_1	MALAS	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 01:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <p>☐ Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	-	<p>Rutin cek ulang sebelum berpindah topik. Fokus menuntaskan satu kartu sampai 'Done' sebelum membuka yang lain. Tuliskan 'next step' singkat di setiap perpindahan kolom. Rapikan urutan kerja agar tidak sering ganti konteks. Prioritaskan checklist inti terlebih dahulu.</p>	<p>Konsistensi akan membuat progresmu terasa. Targetkan minimal satu item selesai per hari. Tingkatkan ritme sedikit demi sedikit. Kamu di jalur yang benar—teruskan. Bangun momentum dengan kemenangan kecil.</p>	<p>Good job, tinggal dipertahankan. Uswa step-by-step menunjukkan komitmen. Kamu sudah bergerak ke arah yang tepat. Progres mulai terlihat dan patut diapresiasi. Terus jaga usaha ini agar stabil.</p>
MHS_2	MALAS	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 08:00	<p>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</p> <p>☐ Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</p> <p>☐ Menerapkan konsep dasar objek dan kelas dalam program sederhana.</p>	-	<p>Hindari pindah kolom sebelum item inti selesai. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Setelah satu kartu selesai, baru lanjut ke topik berikutnya. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Tulis catatan singkat agar langkah berikutnya jelas.</p>	<p>Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Kamu bisa mengerjakan secara bertahap; tidak perlu sempurna di awal. Jika macet, pilih satu tugas yang paling sederhana untuk memulai.</p>	<p>Keberanian memulai adalah langkah penting. Pertahankan niat baik ini. Ada usaha yang meskipun masih terbatas. Terima kasih sudah memulai prosesnya.</p>
MHS_3	MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 10:00	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <p>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</p> <p>☐ Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</p>	-	<p>Hindari pindah kolom sebelum item inti selesai. Tulis catatan singkat agar langkah berikutnya jelas. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Setelah satu kartu selesai, baru lanjut ke topik berikutnya. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Setelah satu kartu selesai, baru lanjut ke topik berikutnya.</p>	<p>Tetapkan target kecil harian agar ada rasa selesai setiap hari. Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Jaga ritme konsisten daripada sprint sesaat. Kamu bisa mengerjakan secara bertahap; tidak perlu sempurna di awal. Jika macet, pilih satu tugas yang paling sederhana untuk memulai.</p>	<p>Setiap langkah kecil akan berdampak bila konsisten. Ada usaha yang terlihat meskipun masih terbatas. Pertahankan niat baik ini. Ada usaha yang sudah memulai prosesnya. Keberanian memulai adalah langkah penting.</p>
	MALAS	Planning	Konsep Dasar dan Pilar OOP	2025-06-04 21:00	<p>☐ Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</p> <p>☐ Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</p> <p>☐ Menerapkan konsep dasar objek dan kelas dalam program sederhana.</p>	-			
MHS_4	MALAS	Planning	Encapsulasi, inheritance, dan Polimorfisme	2025-06-06 17:00	<p>☐ Menerapkan encapsulasi menggunakan modifier akses (private, public, protected).</p> <p>☐ Membuat kelas turunan dan meng-overriding metode dari superclass.</p> <p>☐ Menggunakan polimorfisme melalui overloading dan overriding metode.</p>	-	<p>Hindari pindah kolom sebelum item inti selesai. Tulis catatan singkat agar langkah berikutnya jelas. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Hindari pindah kolom sebelum item inti selesai. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Tulis catatan singkat agar langkah berikutnya jelas.</p>	<p>Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Jaga ritme konsisten daripada sprint sesaat. Kamu bisa mengerjakan secara bertahap; tidak perlu sempurna di awal. Tetapkan target kecil harian agar ada rasa selesai setiap hari.</p>	<p>Setiap langkah kecil akan berdampak bila konsisten. Keberanian memulai adalah langkah penting. Terima kasih sudah memulai prosesnya. Ada usaha yang terlihat meskipun masih terbatas. Pertahankan niat baik ini.</p>
	MALAS	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 21:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <p>☐ Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	-			
MHS_5	MALAS	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-06 23:00	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <p>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</p> <p>☐ Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	-	<p>Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Hindari pindah kolom sebelum item inti selesai. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Tulis catatan singkat agar langkah berikutnya jelas.</p>	<p>Tetapkan target kecil harian agar ada rasa selesai setiap hari. Jika macet, pilih satu tugas yang paling sederhana untuk memulai. Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Jaga ritme konsisten daripada sprint sesaat. Kamu bisa mengerjakan secara bertahap; tidak perlu sempurna di awal.</p>	<p>Terima kasih sudah memulai prosesnya. Ada usaha yang terlihat meskipun masih terbatas. Pertahankan niat baik ini. Setiap langkah kecil akan berdampak bila konsisten. Keberanian memulai adalah langkah penting.</p>
	MALAS	Planning	Encapsulasi, inheritance, dan Polimorfisme	2025-06-06 10:00	<p>☐ Menerapkan encapsulasi menggunakan modifier akses (private, public, protected).</p> <p>☐ Membuat kelas turunan dan meng-overriding metode dari superclass.</p> <p>☐ Menggunakan polimorfisme melalui overloading dan overriding metode.</p>	-			
MHS_6	SEDIKIT MALAS	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-06 08:00	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <p>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</p> <p>☐ Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	Planning → Monitoring (2025-06-07 15:45)	<p>Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Rutin cek ulang sebelum berpindah topik. Tuliskan 'next step' singkat di setiap perpindahan kolom. Prioritaskan checklist inti terlebih dahulu.</p>	<p>Fokus menuntaskan satu kartu dengan kemenangan kecil. Tangkatkan ritme sedikit demi sedikit. Konsistensi akan membuat progresmu terasa. Kamu di jalur yang benar—teruskan.</p>	<p>Terus jaga usaha ini agar stabil. Good job, tinggal dipertahankan. Upaya step-by-step menunjukkan komitmen. Kamu sudah bergerak ke arah yang tepat. Progres mulai terlihat dan patut diapresiasi.</p>
	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 08:00	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <p>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</p> <p>☐ Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</p>	Planning → Monitoring (2025-06-07 14:48)			
	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-05 15:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <p>☐ Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	Planning → Monitoring (2025-06-07 14:19)			
MHS_7	SEDIKIT MALAS	Planning	Encapsulasi, inheritance, dan Polimorfisme	2025-06-05 08:00	<p>Menerapkan encapsulasi menggunakan modifier akses (private, public, protected).</p> <p>☐ Membuat kelas turunan dan meng-overriding metode dari superclass.</p> <p>☐ Menggunakan polimorfisme melalui overloading dan overriding metode.</p>	-	<p>Hindari pindah kolom sebelum item inti selesai. Setelah satu kartu selesai, baru lanjut ke topik berikutnya. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus.</p>	<p>Jika macet, pilih satu tugas yang paling sederhana untuk memulai. Kamu bisa mengerjakan secara bertahap; tidak perlu sempurna di awal. Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Jaga ritme konsisten daripada sprint sesaat. Tetapkan target kecil harian agar ada rasa selesai setiap hari.</p>	<p>Pertahankan niat baik ini. Setiap langkah kecil akan berdampak bila konsisten. Terima kasih sudah memulai prosesnya. Keberanian memulai adalah langkah penting. Ada usaha yang terlihat meskipun masih terbatas.</p>
	SEDIKIT MALAS	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-07 02:00	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <p>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</p> <p>☐ Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	-			

	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-06 04:00	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div><input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 16:11)					
	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 08:00	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 14:26)					
MHS_8							Jaga ritme konsisten daripada sprint sesaat. Kamu bisa mengejar secara bertahap; tidak perlu sempurna di awal.	Pertahankan niat baik ini. Setiap langkah kecil akan berdampak bila konsisten.			
	SEDIKIT MALAS	Planning	Konsep Dasar dan Pilar OOP	2025-06-06 06:00	<div><input type="checkbox"/> Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	-	Tulis catatan singkat agar langkah berikutnya jelas. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Hindari pindah kolom sebelum item ini selesai.	Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Jika macet, pilih satu tugas yang paling sederhana untuk memulai. Tetapkan target kecil harian agar ada rasa selesai setiap hari.			
	SEDIKIT MALAS	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-05 06:00	<div><input type="checkbox"/> Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</div> <div><input type="checkbox"/> Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</div> <div><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</div>	Planning → Monitoring (2025-06-07 15:01)					
	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 06:00	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 15:40)					
MHS_9							Hindari pindah kolom sebelum item ini selesai. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Tulis catatan singkat agar langkah berikutnya jelas. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus.	Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Jaga ritme konsisten daripada sprint sesaat. Jika macet, pilih satu tugas yang paling sederhana untuk memulai. Kamu bisa mengejar secara bertahap; tidak perlu sempurna di awal.			
	SEDIKIT MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 12:00	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div><input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	-		Terima kasih sudah memulai prosesnya. Pertahankan niat baik ini. Ada usaha yang terlihat meskipun masih terbatas. Setiap langkah kecil akan berdampak bila konsisten.			
	SEDIKIT MALAS	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-05 19:00	<div>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</div> <div>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</div> <div><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</div>	-					
	SEDIKIT MALAS	Monitoring	Encapsulasi, Inheritance, dan Polimorfisme	2025-06-06 15:00	<div><input type="checkbox"/> Menerapkan encapsulasi menggunakan modifier akses (private, public, protected).</div> <div><input type="checkbox"/> Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	Planning → Monitoring (2025-06-07 08:59)					
MHS_10							Setelah satu kartu selesai, baru lanjut ke topik berikutnya. Mulailah dengan satu kartu prioritas dan selesaikan checklist paling mudah terlebih dahulu. Hindari pindah kolom sebelum item ini selesai. Tulis catatan singkat agar langkah berikutnya jelas.	Jaga ritme konsisten daripada sprint sesaat. Kamu bisa mengejar secara bertahap; tidak perlu sempurna di awal. Jika macet, pilih satu tugas yang paling sederhana untuk memulai. Apresiasi kemajuan sekecil apa pun untuk menjaga semangat. Tetapkan target kecil harian agar ada rasa selesai setiap hari.			
	SEDIKIT MALAS	Planning	Encapsulasi, Inheritance, dan Polimorfisme	2025-06-07 04:00	<div>Menerapkan encapsulasi menggunakan modifier akses (private, public, protected).</div> <div><input type="checkbox"/> Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	-		Terima kasih sudah memulai prosesnya. Ada usaha yang terlihat meskipun masih terbatas. Keberanian memulai adalah langkah penting. Pertahankan niat baik ini.			
	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-05 10:00	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div><input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 10:10)					
	SEDIKIT MALAS	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-06 06:00	<div>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	Planning → Monitoring (2025-06-07 09:37)					
MHS_11							Fokus menuntaskan satu kartu sampai 'Done' sebelum membuka yang lain. Prioritaskan checklist ini terlebih dahulu. Rapiakan urutan kerja agar tidak sering ganti konteks. Tuliskan 'next step' singkat di setiap perpindahan kolom. Rutin cek ulang sebelum berpindah topik.	Targetkan minimal satu item selesai per hari. Tingkatkan ritme sedikit demi sedikit. Bangun momentum dengan kemenangan kecil. Konsistensi akan membuat progresmu terasa.			
	SEDIKIT MALAS	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 19:00	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	-		Terus jaga usaha ini agar stabil. Kamu sudah bergerak ke arah yang tepat. Upaya step-by-step menunjukkan komitmen. Good job, tinggal dipertahankan. Progres mulai terlihat dan patut diapresiasi.			
	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 03:00	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 13:18)					
	SEDIKIT MALAS	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-05 15:00	<div>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</div> <div><input type="checkbox"/> Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</div> <div><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</div>	Planning → Monitoring (2025-06-07 11:25)					
MHS_12							Menerapkan encapsulasi menggunakan modifier akses (private, public, protected). <div>Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	-	Fokus menuntaskan satu kartu sampai 'Done' sebelum membuka yang lain. Rapiakan urutan kerja agar tidak sering ganti konteks. Prioritaskan ritme sedikit demi sedikit. Targetkan minimal satu item selesai per topik.	Kamu di jalur yang benar—teruskan. Konsistensi akan membuat progresmu terasa. Bangun momentum dengan kemenangan kecil. Tingkatkan ritme sedikit demi sedikit. Targetkan minimal satu item selesai per hari.	Good job, tinggal dipertahankan. Progres mulai terlihat dan patut diapresiasi. Upaya step-by-step menunjukkan komitmen. Kamu sudah bergerak ke arah yang tepat.
	SEDIKIT MALAS	Planning	Encapsulasi, Inheritance, dan Polimorfisme	2025-06-07 10:00	<div>Menerapkan encapsulasi menggunakan modifier akses (private, public, protected).</div> <div>Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	-					
	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 03:00	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div><input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 12:41)					
	SEDIKIT MALAS	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-05 11:00	<div>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	Planning → Monitoring (2025-06-07 13:02)					
	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 18:00	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 15:20)					

MHS_13	SEDIKT MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 18:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-	Kurangi penundaan kecil agar tidak menumpuk di akhir. Pertahankan alur yang sudah terbentuk. Tambahkan ringkasan singkat saat close untuk dokumentasi. Selesaikan satu topik tuntas sebelum pindah.	Teruskan kebiasaan baik yang sudah ada. Kamu sudah ada di tengah jalan— lanjutkan momentum ini. Jika tempo agar tidak turun. Sedikit dorongan lagi untuk tembus "Done" konsisten. Tetapkan target harian yang realistis.	Pertahankan performa ini. Ritme kerja yang terjaga patut diapresiasi. Kerjamu stabil dan konsisten. Dokumentasi dan perpindahan kolom sudah cukup jelas.
	SEDIKT MALAS	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-07 12:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	-			
	SEDIKT MALAS	Monitoring	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 13:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 14:40)			
	SEDIKT MALAS	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-06 12:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 14:45)			
MHS_14	SEDIKT MALAS	Planning	Prinsip SOLID dan Adaptive Code	2025-06-07 00:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. <input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID. <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	-	Fokus menuntaskan satu kartu sampai "Done" sebelum membuka yang lain. Prioritaskan checklist ini terlebih dahulu. Tuliskan "next step" singkat di setiap perpindahan kolom. Rutin cek ulang sebelum berpindah topik. Rapihan urutan kerja agar tidak sering ganti konteks.	Konsistensi akan membuat progresmu terasa. Tingkatkan ritme sedikit demi sedikit. Targetkan minimal satu item selesai per hari. Kamu di jalur yang benar— teruskan.	Kamu sudah bergerak ke arah yang tepat. Good job, tinggal dipertahankan. Terus jaga usaha ini agar stabil. Upaya step-by-step menunjukkan komitmen.
	SEDIKT MALAS	Monitoring	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-06 23:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 17:34)			
	SEDIKT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 18:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 16:58)			
	SEDIKT MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 07:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-	Tambahkan ringkasan singkat saat ada di tengah dokumentasi. Kurangi penundaan kecil agar tidak menumpuk di akhir. Pertahankan alur yang sudah terbentuk. Selesaikan satu topik tuntas sebelum pindah.	Kamu sudah ada di tengah jalan— lanjutkan momentum ini. Jika tempo agar tidak turun. Teruskan kebiasaan baik yang sudah ada. Tetapkan target harian yang realistis.	Kerjamu stabil dan konsisten. Pertahankan performa ini. Ritme kerja yang terjaga patut diapresiasi. Dokumentasi dan perpindahan kolom sudah cukup jelas. Kualitas progresnya terlihat.
MHS_15	SEDIKT MALAS	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 16:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	-			
	SEDIKT MALAS	Planning	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-05 05:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.				
	SEDIKT MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 07:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-			
	SEDIKT MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 07:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-			
MHS_16	CUKUP	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-07 06:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	-	Prioritaskan checklist ini terlebih dahulu. Rutin cek ulang sebelum berpindah topik. Rapihan urutan kerja agar tidak sering ganti konteks. Fokus menuntaskan satu kartu sampai "Done" sebelum membuka yang lain.	Tingkatkan ritme sedikit demi sedikit. Targetkan minimal satu item selesai per hari. Konsistensi akan membuat progresmu terasa. Bangun momentum dengan kemenangan kecil. Kamu di jalur yang benar— teruskan.	Kamu sudah bergerak ke arah yang tepat. Progres mulai terlihat dan patut diapresiasi. Good job, tinggal dipertahankan. Terus jaga usaha ini agar stabil. Upaya step-by-step menunjukkan komitmen.
	CUKUP	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 09:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-			
	CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 14:42	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. <input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID. <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 08:48)			
	CUKUP	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 14:43	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. <input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 10:16)			
	CUKUP	Controlling	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 14:40	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 09:55) Monitoring → Controlling (2025-06-07 10:38)			
MHS_17	CUKUP	Planning	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 08:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). <input type="checkbox"/> Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	-	Rapihan urutan kerja agar tidak sering ganti konteks. Prioritaskan checklist ini terlebih dahulu. Fokus menuntaskan satu kartu sampai "Done" sebelum membuka yang lain. Tuliskan "next step" singkat di setiap perpindahan kolom. Rutin cek ulang sebelum berpindah topik.	Targetkan minimal satu item selesai per hari. Konsistensi akan membuat progresmu terasa. Bangun momentum dengan kemenangan kecil. Kamu di jalur yang benar— teruskan.	Terus jaga usaha ini agar stabil. Upaya step-by-step menunjukkan komitmen. Progres mulai terlihat dan patut diapresiasi. Kamu sudah bergerak ke arah yang tepat.
	CUKUP	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 14:43	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. <input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 10:14)			
	CUKUP	Controlling	Prinsip SOLID dan Adaptive Code	2025-06-07 15:40	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. <input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID. <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 10:12) Monitoring → Controlling (2025-06-07 11:13)			
	CUKUP	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 14:19	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 09:31) Monitoring → Controlling (2025-06-07 09:58)			
	CUKUP	Planning	Prinsip SOLID dan Adaptive Code	2025-06-06 01:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. <input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID. <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	-	Prioritaskan checklist ini terlebih dahulu. Fokus menuntaskan satu kartu sampai "Done" sebelum membuka yang lain. Tuliskan "next step" singkat di setiap perpindahan kolom. Rutin cek ulang sebelum berpindah topik.	Konsistensi akan membuat progresmu terasa. Targetkan minimal satu item selesai per hari. Kamu di jalur yang benar— teruskan. Bangun momentum dengan kemenangan kecil.	Upaya step-by-step menunjukkan komitmen. Progres mulai terlihat dan patut diapresiasi. Kamu sudah bergerak ke arah yang tepat. Terus jaga usaha ini agar stabil.
MHS_18	CUKUP	Planning	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-06 04:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	-			
	CUKUP	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-07 14:35	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 09:06)			

MHS_19	CUKUP	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 14:12	<div>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	Planning → Monitoring (2025-06-07 10:04) Monitoring → Controlling (2025-06-07 10:20)			
	CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:10	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div><input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 08:35) Monitoring → Controlling (2025-06-07 09:04)			
	CUKUP	Planning	Konsep Dasar dan Pilar OOP	2025-06-06 03:00	<div>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	-	<div>Tambahkan ringkasan singkat saat close untuk dokumentasi. Pertahankan alur yang sudah terbentuk.</div> <div>Alumnya sudah rapi, fokus pada finalisasi checklist</div> <div>Selesaikan satu topik tuntas sebelum pindah. Kurangi penundaan kecil agar tidak menumpuk di akhir.</div>	<div>Teruskan kebiasaan baik yang sudah ada. Tetapkan target harian yang realistis. Jaga tempo agar tidak turun.</div> <div>Kamu sudah ada di tengah perjalanan momentum ini. Sedikit dorongan lagi untuk tembus "Done" konsisten.</div>	<div>Ritme kerja yang terjaga patut diapresiasi.</div> <div>Kerjamu stabil dan konsisten. Dokumentasi dan perpindahan kolom sudah cukup jelas. Pertahankan performa ini. Kualitas progresnya terlihat.</div>
	CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 15:51	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div><input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 11:55)			
	CUKUP	Controlling	Perancangan Kelas dan Interaksi Objek	2025-06-07 15:40	<div>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</div> <div><input type="checkbox"/> Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</div> <div><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</div>	Planning → Monitoring (2025-06-07 12:56) Monitoring → Controlling (2025-06-07 14:00)			
CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 14:11	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div><input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 11:30) Monitoring → Controlling (2025-06-07 12:29)				
MHS_20	CUKUP	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 14:00	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div><input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	-	<div>Rapikan urutan kerja agar tidak sering ganti konteks. Rutin cek ulang sebelum berpindah topik. Tulislah 'next step' singkat di setiap perpindahan kolom. Fokus memuntaskan satu kartu sampai "Done" sebelum membuka yang lain. Prioritaskan checklist ini terlebih dahulu.</div>	<div>Bangun momentum dengan kemenangan kecil. Konsistensi akan membuat progresmu terasa. Targetkan minimal satu item selesai per hari. Kamu di jalur yang benar—teruskan.</div> <div>Tingkatkan ritme sedikit demi sedikit.</div>	<div>Upaya step-by-step menunjukkan komitmen. Kamu sudah bergerak ke arah yang tepat. Terus jaga usaha ini agar stabil. Good job, tinggal dipertahankan. Progres mulai terlihat dan patut diapresiasi.</div>
	CUKUP	Planning	Prinsip SOLID dan Adaptive Code	2025-06-06 20:00	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	-			
	CUKUP	Monitoring	Encapsulasi, Inheritance, dan Polimorfisme	2025-06-07 15:54	<div>Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</div> <div>Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	Planning → Monitoring (2025-06-07 13:31)			
	CUKUP	Controlling	Perancangan Kelas dan Interaksi Objek	2025-06-07 14:10	<div>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</div> <div>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</div> <div><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</div>	Planning → Monitoring (2025-06-07 13:25) Monitoring → Controlling (2025-06-07 14:07)			
	CUKUP	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 15:31	<div>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	Planning → Monitoring (2025-06-07 12:25) Monitoring → Controlling (2025-06-07 14:02)			
MHS_21	CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 15:17	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div>Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 07:29)	<div>Tambahkan catatan singkat agar mudah dilacak. Jaga kualitas sambil menjaga kecepatan.</div> <div>Cegah carry-over dengan menutup kartu lama terlebih dahulu.</div> <div>Eksekusi sudah kuat; segera finalkan item tersisa agar cepat "Done".</div>	<div>Kamu berada di jalur yang tepat. Sedikit lagi untuk mencapai semua kartu aktif. Jadikan target selesai harian sebagai pendorong.</div> <div>Energi positifmu terlihat—lanjutkan.</div>	<div>Kualitas pekerjaan tampak konsisten. Produktivitasmu memberi dampak nyata. Performa solid dan efisien. Kerja bagus—teruskan standar ini.</div>
	CUKUP	Controlling	Encapsulasi, Inheritance, dan Polimorfisme	2025-06-07 14:33	<div>Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</div> <div>Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div>Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	Planning → Monitoring (2025-06-07 08:10) Monitoring → Controlling (2025-06-07 09:56)			
	CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:01	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div>Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 08:42) Monitoring → Controlling (2025-06-07 10:14)			
MHS_22	CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 14:25	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div>Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 08:54)	<div>Pastikan dokumentasi ringkas tersedia saat close. Bila memungkinkan, bantu review atau handoff.</div> <div>Jaga agar tidak ada pekerjaan yang menggantung.</div> <div>Pertahankan disiplin penutupan yang konsisten. Eksekusi sangat rapi, fokus pada finishing dan validasi akhir.</div>	<div>Momentum positifmu sangat terasa. Tetapkan target namun realistis. Jaga keseimbangan antara kecepatan dan ketelitian.</div> <div>Teruskan ritme dan kualitas ini.</div>	<div>Dokumentasi dan penutupan kartu sangat terbantu.</div> <div>Kontribusi kamu jelas terlihat pada alur kerja. Kinerja sangat unggul dan konsisten. Terima kasih atas kerja hebatnya. Hasil dan kecepatannya mengesankan.</div>
	CUKUP	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 14:36	<div>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</div> <div>Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme.</div> <div>Menerapkan konsep dasar objek dan kelas dalam program sederhana.</div>	Planning → Monitoring (2025-06-07 09:26) Monitoring → Controlling (2025-06-07 09:56) Controlling → Reflection (2025-06-07 11:04)			
	CUKUP	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:44	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div>Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 10:19) Monitoring → Controlling (2025-06-07 12:01) Controlling → Reflection (2025-06-07 12:44)			
MHS_23	CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 15:13	<div>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</div> <div>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</div> <div>Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</div>	Planning → Monitoring (2025-06-07 09:01)	<div>Jaga kualitas sambil menjaga kecepatan.</div> <div>Eksekusi sudah kuat; segera finalkan item tersisa agar cepat "Done".</div> <div>Lakukan pengecekan detail kecil sebelum penutupan.</div> <div>Cegah carry-over dengan menutup kartu lama terlebih dahulu.</div>	<div>Sedikit lagi untuk mencapai semua kartu aktif. Jadikan target selesai harian sebagai pendorong.</div> <div>Pertahankan konsistensi dan pake saat ini.</div> <div>Kamu berada di jalur yang tepat.</div>	<div>Transisi antar kolom rapi dan teratur.</div> <div>Produktivitasmu memberi dampak nyata. Kerja bagus—teruskan standar ini. Performa solid dan efisien.</div>
	CUKUP	Controlling	Encapsulasi, Inheritance, dan Polimorfisme	2025-06-07 14:35	<div>Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</div> <div>Membuat kelas turunan dan meng-overriding metode dari superclass.</div> <div>Menggunakan polimorfisme melalui overloading dan overriding metode.</div>	Planning → Monitoring (2025-06-07 08:50) Monitoring → Controlling (2025-06-07 09:13)			
	CUKUP	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 14:18	<div>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</div> <div>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</div> <div>Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</div>	Planning → Monitoring (2025-06-07 08:59) Monitoring → Controlling (2025-06-07 09:47) Controlling → Reflection (2025-06-07 11:44)			

							Lakukan pengecekan detail kecil sebelum penutupan. Jaga kualitas sambil menjaga kecepatan. Cekah carry-over dengan menutup kartu lama terlebih dahulu. Eksekusi sudah kuat; segera finalkan item terisa agar cepat 'Done'. Tambahkan catatan singkat agar mudah dilacak.	Jadikan target selesai harian sebagai pendorong. Sedikit lagi untuk merapikan semua kartu aktif. Energi positifmu terlihat—lanjutan. Kamu berada di jalur yang tepat. Pertahankan konsistensi dan pace saat ini.	Kualitas pekerjaan tampak konsisten. Performa solid dan efisien. Produktivitasn u memberi dampak nyata. Transisi antar kolom rapi dan teratur. Kerja bagus—teruskan standar ini.
MHS_24	CUKUP	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-05 11:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	-			
	CUKUP	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 14:50	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 08:21)			
	CUKUP	Controlling	Prinsip SOLID dan Adaptive Code	2025-06-07 14:56	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 07:34) Monitoring → Controlling (2025-06-07 08:03)			
	CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 14:56	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 08:52) Monitoring → Controlling (2025-06-07 09:22)			
MHS_25	CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 14:38	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 09:24)	Eksekusi sudah kuat; segera finalkan item terisa agar cepat 'Done'. Lakukan pengecekan detail kecil sebelum penutupan. Cekah carry-over dengan menutup kartu lama terlebih dahulu. Jaga kualitas sambil menjaga kecepatan. Tambahkan catatan singkat agar mudah dilacak.	Kamu berada di jalur yang tepat. Sedikit lagi untuk merapikan semua kartu aktif. Energi positifmu terlihat—lanjutan. Jadikan target selesai harian sebagai pendorong.	Transisi antar kolom rapi dan teratur. Kerja bagus—teruskan standar ini. Performa solid dan efisien. Produktivitasn u memberi dampak nyata.
	CUKUP	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-07 15:19	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 08:52)			
	CUKUP	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 15:15	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 07:31) Monitoring → Controlling (2025-06-07 08:57) Controlling → Reflection (2025-06-07 09:48)			
MHS_26	RAJIN	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 09:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-	Tambahkan ringkasan singkat saat close untuk dokumentasi. Selesaikan satu topik tuntas sebelum pindah. Pertahankan alur yang sudah terbentuk. Kurangi penundaan kecil agar tidak memumpuk di akhir. Alurnya sudah rapi; fokus pada finalisasi checklist penting.	Tetapkan target harian yang realistis. Teruskan kebiasaan baik yang sudah ada. Kamu sudah ada di tengah jalan—lanjutkan momentum ini. Jaga tempo agar tidak turun. Sedikit dorongan lagi untuk tembus "Done" konsisten.	Ritme kerja yang terjaga patut diapresiasi. Kualitas progresnya terlihat. Dokumentasi dan perpindahan kolom sudah cukup jelas. Pertahankan performa ini.
	RAJIN	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 10:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 05:39)			
	RAJIN	Controlling	Perancangan Kelas dan Interaksi Objek	2025-06-07 15:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 06:45) Monitoring → Controlling (2025-06-07 08:41)			
	RAJIN	Controlling	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 14:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 06:14) Monitoring → Controlling (2025-06-07 06:54)			
	RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 16:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 06:06) Monitoring → Controlling (2025-06-07 07:28) Controlling → Reflection (2025-06-07 07:48)			
MHS_27	RAJIN	Planning	Prinsip SOLID dan Adaptive Code	2025-06-07 09:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	-	Kurangi penundaan kecil agar tidak memumpuk di akhir. Tambahkan ringkasan singkat saat close untuk dokumentasi. Selesaikan satu topik tuntas sebelum pindah. Alurnya sudah rapi; fokus pada finalisasi checklist penting. Pertahankan alur yang sudah terbentuk.	Jaga tempo agar tidak turun. Tetapkan target harian yang realistis. Sedikit dorongan lagi untuk tembus "Done" konsisten. Teruskan kebiasaan baik yang sudah ada.	Kerjamu stabil dan konsisten. Pertahankan performa ini. Kualitas progresnya terlihat. Dokumentasi dan perpindahan kolom sudah cukup jelas.
	RAJIN	Monitoring	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 13:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 07:20)			
	RAJIN	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 07:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 07:18)			
	RAJIN	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 07:33) Monitoring → Controlling (2025-06-07 08:51)			
	RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 13:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 06:03) Monitoring → Controlling (2025-06-07 06:29) Controlling → Reflection (2025-06-07 08:03)			
MHS_28	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 09:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Tambahkan ringkasan singkat saat close untuk dokumentasi. Kurangi penundaan kecil agar tidak memumpuk di akhir. Selesaikan satu topik tuntas sebelum pindah. Alurnya sudah rapi; fokus pada finalisasi checklist penting. Pertahankan alur yang sudah terbentuk.	Sedikit dorongan lagi untuk tembus "Done" konsisten. Teruskan kebiasaan baik yang sudah ada. Tetapkan target harian yang realistis. Kamu sudah ada di tengah jalan—lanjutkan momentum ini. Jaga tempo agar tidak turun.	Pertahankan performa ini. Dokumentasi dan perpindahan kolom sudah cukup jelas. Kerjamu stabil dan konsisten. Kualitas progresnya terlihat.
	RAJIN	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-07 08:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 12:47)			
	RAJIN	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 12:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:21)			
	RAJIN	Controlling	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 15:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 13:19) Monitoring → Controlling (2025-06-07 14:23)			

MHS_29	RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 12:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 12:22) Monitoring → Controlling (2025-06-07 13:41) Controlling → Reflection (2025-06-07 15:27)	Alurmu sudah rapi; fokus pada finalisasi checklist penting. Pertahankan alur yang sudah terbentuk. Kurangi penundaan kecil agar tidak menumpuk di akhir. Tambahkan ringkasan singkat saat close untuk dokumentasi.	Tetapkan target harian yang realistis. Teruskan kebiasaan baik yang sudah ada. Jika tempo kerja tidak turun. Sedikit dorongan lagi untuk tembus "Done" konsisten.	Kerjamu stabil dan konsisten. Kualitas progresnya terlihat. Ritme kerja yang terjaga patut diapresiasi. Pertahankan performa ini.
	RAJIN	Planning	Prinsip SOLID dan Adaptive Code	2025-06-07 16:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	-			
	RAJIN	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 16:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 11:00)			
	RAJIN	Controlling	Perancangan Kelas dan Interaksi Objek	2025-06-07 13:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode. Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.	Planning → Monitoring (2025-06-07 11:09) Monitoring → Controlling (2025-06-07 11:58)			
	RAJIN	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 16:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 11:02) Monitoring → Controlling (2025-06-07 12:35)			
MHS_30	RAJIN	Controlling	Encapsulasi, inheritance, dan Polimorfisme	2025-06-07 12:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 10:36) Monitoring → Controlling (2025-06-07 11:52)			
	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 07:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Cegah carry-over dengan menutup kartu lama terlebih dahulu. Jaga kualitas sambil menjaga kecepatan. Eksekusi sudah kuat; segera finalkan item tersisa agar cepat "Done". Lakukan pengecekan detail kecil sebelum penutupan.	Energi positifmu terlihat—lanjutan. Jadikan target sebagai semua kartu aktif. Pertahankan konsistensi dan pace saat ini.	Performa solid dan efisien. Kerja bagus—teruskan standar ini. Produktivitasmu memberi dampak nyata. Kualitas pekerjaan tampak konsisten.
	RAJIN	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 11:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 11:34)			
	RAJIN	Reflection	Encapsulasi, inheritance, dan Polimorfisme	2025-06-07 09:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 11:40) Monitoring → Controlling (2025-06-07 11:58) Controlling → Reflection (2025-06-07 12:51)			
	RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 13:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 12:45) Monitoring → Controlling (2025-06-07 14:05) Controlling → Reflection (2025-06-07 14:44)			
MHS_31	RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 12:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:02) Monitoring → Controlling (2025-06-07 12:12) Controlling → Reflection (2025-06-07 12:31)			
	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 09:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Bila memungkinkan, bantu review atau handoff. Jaga agar tidak ada pekerjaan menggantung. Pastikan kualitas ritme dan dokumentasi ringkas tersedia saat close.	Jaga keseimbangan antara kecepatan dan ketelitian. Tetapkan target ambisius namun realistis. Standarmu tinggi dan jadi acuan bagi yang lain. Teruskan kualitas ritme dan dokumentasi ringkas tersedia saat close.	Dokumentasi dan penutupan kartu sangat terib. Kontribusi kamu jelas terlihat pada alur kerja. Kinerja sangat unggul dan konsisten. Hasil dan kecepatannya mengesankan. Terima kasih atas kerja hebatnya.
	RAJIN	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 07:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:00)			
	RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 07:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 11:35) Monitoring → Controlling (2025-06-07 11:53) Controlling → Reflection (2025-06-07 12:07)			
	RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 10:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 11:28) Monitoring → Controlling (2025-06-07 11:47) Controlling → Reflection (2025-06-07 13:02)			
MHS_32	RAJIN	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-07 08:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	-	Cegah carry-over dengan menutup kartu lama terlebih dahulu. Eksekusi sudah kuat; segera finalkan item tersisa agar cepat "Done". Lakukan pengecekan detail kecil sebelum penutupan. Tambahkan catatan singkat agar mudah dilacak.	Energi positifmu terlihat—lanjutan. Pertahankan konsistensi dan pace saat ini. Sedikit lagi untuk merapikan semua kartu aktif. Kamu berada di jalur yang tepat.	Produktivitasmu memberi dampak nyata. Performa solid dan efisien. Kualitas pekerjaan tampak konsisten. Transisi antar kolom rapi dan teratur. Kerja bagus—teruskan standar ini.
	RAJIN	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 11:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 10:13)			
	RAJIN	Controlling	Encapsulasi, inheritance, dan Polimorfisme	2025-06-07 12:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 10:36) Monitoring → Controlling (2025-06-07 11:13)			
	RAJIN	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 11:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 10:02) Monitoring → Controlling (2025-06-07 11:44)			
MHS_33	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 10:00	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Encapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Lakukan pengecekan detail kecil sebelum penutupan. Cegah carry-over dengan menutup kartu lama terlebih dahulu. Tambahkan catatan singkat agar mudah dilacak. Eksekusi sudah kuat; segera finalkan item tersisa agar cepat "Done". Jaga kualitas sambil menjaga kecepatan.	Sedikit lagi untuk merapikan semua kartu aktif. Pertahankan konsistensi dan pace saat ini. Energi positifmu terlihat—lanjutan. Jadikan target sebagai semua hari kerja sebagai pendorong.	Produktivitasmu memberi dampak nyata. Transisi antar kolom rapi dan teratur. Kerja bagus—teruskan standar ini. Performa solid dan efisien.
	RAJIN	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 09:46)			
	RAJIN	Controlling	Encapsulasi, inheritance, dan Polimorfisme	2025-06-07 12:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 09:40) Monitoring → Controlling (2025-06-07 10:21)			
	RAJIN	Controlling	Prinsip SOLID dan Adaptive Code	2025-06-07 09:00	Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 09:28) Monitoring → Controlling (2025-06-07 11:21)			

									Bila memungkinkan bantu review atau handoff. Jaga agar tidak ada pekerjaan menggantung. Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pertahankan disiplin penutupan yang konsisten.	Tetapakan target ambulus namun realistis. Momentum positifmu sangat terasa. Standarmu tinggi dan jadi acuan bagi yang lain. Jaga keseimbangan antara kecepatan dan ketelitian.	Terima kasih atas kerja bembayanya. Kinerja sangat unggul dan konsisten. Kontribusiimu jelas terlihat pada alur kerja. Dokumentasi dan penutupan kartu sangat terbib.
MHS_34	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 07:00	-	Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.					
	RAJIN	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 09:00		Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 09:55)				
	RAJIN	Controlling	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 08:00		Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 09:53) Monitoring → Controlling (2025-06-07 11:27)				
	RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 09:00		Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 10:27) Monitoring → Controlling (2025-06-07 11:31) Controlling → Reflection (2025-06-07 12:33)				
	RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 15:00		Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 11:21) Monitoring → Controlling (2025-06-07 11:31) Controlling → Reflection (2025-06-07 13:13)				
MHS_35	RAJIN	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-07 09:00	-	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.		Cegah carry-over dengan menutup kartu lama terlebih dahulu. Eksekusi sudah kuat; segera fixalkan item tersisa agar cepat "Done". Jaga kualitas sambil menjaga kecepatan. Lakukan pengecekan detail kecil sebelum penutupan.	Pertahankan konsistensi dan pace saat ini. Jadikan target sebagai pendorong. Kamu berada di jalur yang tepat. Energi positifmu terlihat—lanjutan.	Kualitas pekerjaan tampak konsisten. Kerja bagus—teruskan standar ini. Transisi antar kolom rapi dan teratur. Performa solid dan efisien. Produktivitasmu memberi dampak nyata.	
	RAJIN	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 07:00		Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 13:15)				
	RAJIN	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 07:00		Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:09)				
	RAJIN	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 11:00		Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 13:50) Monitoring → Controlling (2025-06-07 14:43) Controlling → Reflection (2025-06-07 16:35)				
MHS_36	SANGAT RAJIN	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 10:00		Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 10:25) Monitoring → Controlling (2025-06-07 11:06) Controlling → Reflection (2025-06-07 11:40)	Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pertahankan disiplin penutupan yang konsisten. Pastikan dokumentasi ringkas tersedia saat close. Bila memungkinkan, bantu review atau handoff.	Standarmu tinggi dan jadi acuan bagi yang lain. Tetapkan target ambulus namun realistis. Momentum positifmu sangat terasa. Teruskan ritme dan kualitas ini.	Kinerja sangat unggul dan konsisten. Hasil dan kecepatannya mengesankan. Dokumentasi dan penutupan kartu sangat rapi dan efisien. Terima kasih atas kerja bembayanya.	
	SANGAT RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 14:00		Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 11:27) Monitoring → Controlling (2025-06-07 12:34) Controlling → Reflection (2025-06-07 13:01)				
	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 05:00		Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 10:36) Monitoring → Controlling (2025-06-07 11:21) Controlling → Reflection (2025-06-07 12:30)				
	SANGAT RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 07:00		Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 10:42) Monitoring → Controlling (2025-06-07 11:48) Controlling → Reflection (2025-06-07 13:41)				
	SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 14:00		Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 11:56) Monitoring → Controlling (2025-06-07 13:16) Controlling → Reflection (2025-06-07 13:38)				
	SANGAT RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 16:00		Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:56) Monitoring → Controlling (2025-06-07 14:15) Controlling → Reflection (2025-06-07 16:12)	Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pastikan dokumentasi ringkas tersedia saat close. Bila memungkinkan, bantu review atau handoff. Jaga agar tidak ada pekerjaan menggantung. Pertahankan disiplin penutupan yang konsisten.	Momentum positifmu sangat terasa. Standarmu tinggi dan jadi acuan bagi yang lain. Teruskan ritme dan kualitas ini. Jaga keseimbangan antara kecepatan dan ketelitian. Tetapakan target namun realistis.	Dokumentasi dan penutupan kartu sangat rapi dan efisien. Terima kasih atas kerja bembayanya. Kinerja sangat unggul dan konsisten.	
MHS_37	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 10:00		Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 11:51) Monitoring → Controlling (2025-06-07 13:37) Controlling → Reflection (2025-06-07 15:35)				
	SANGAT RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 08:00		Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 12:09) Monitoring → Controlling (2025-06-07 13:11) Controlling → Reflection (2025-06-07 14:23)				
	SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 12:00		Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 12:05) Monitoring → Controlling (2025-06-07 13:06) Controlling → Reflection (2025-06-07 13:23)				
	SANGAT RAJIN	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 15:00		Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 12:29) Monitoring → Controlling (2025-06-07 12:39) Controlling → Reflection (2025-06-07 13:38)				
MHS_38	SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 15:00		Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning → Monitoring (2025-06-07 11:11) Monitoring → Controlling (2025-06-07 13:01) Controlling → Reflection (2025-06-07 14:09)	Pertahankan disiplin penutupan yang konsisten. Jaga agar tidak ada pekerjaan menggantung. Pastikan dokumentasi ringkas tersedia saat close.	Jaga keseimbangan antara kecepatan dan ketelitian. Standarmu tinggi dan jadi acuan bagi yang lain. Teruskan ritme dan kualitas ini. Tetapakan target namun realistis.	Hasil dan kecepatannya mengesankan. Terima kasih atas kerja bembayanya. Kontribusiimu jelas terlihat pada alur kerja. Kinerja sangat unggul dan konsisten.	
	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 13:00		Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning → Monitoring (2025-06-07 10:41) Monitoring → Controlling (2025-06-07 12:20) Controlling → Reflection (2025-06-07 14:03)				
	SANGAT RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 12:00		Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 10:48) Monitoring → Controlling (2025-06-07 12:22) Controlling → Reflection (2025-06-07 14:03)				
	SANGAT RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 09:00		Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. Melakukan refactoring kode agar sesuai dengan prinsip SOLID. Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.	Planning → Monitoring (2025-06-07 09:47) Monitoring → Controlling (2025-06-07 10:21) Controlling → Reflection (2025-06-07 11:08)				
	SANGAT RAJIN	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 09:00		Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Membuat kelas turunan dan meng-overriding metode dari superclass. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 11:10) Controlling → Reflection (2025-06-07 12:54)				

MHS_39	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 09:00	<p>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</p> <p>Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme.</p> <p>Menerapkan konsep dasar objek dan kelas dalam program sederhana.</p>	<p>Planning → Monitoring (2025-06-07 10:22)</p> <p>Monitoring → Controlling (2025-06-07 12:07)</p> <p>Controlling → Reflection (2025-06-07 12:57)</p>	<p>Pertahankan disiplin penutupan yang konsisten. Jaga agar tidak ada pekerjaan menggantung. Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pastikan dokumentasi ringkas tersedia saat close. Bila memungkinkan bantu review atau handoff</p>	<p>Standarmu tinggi dan jadi acuan bagi yang lain. Jaga keseimbangan antara kecepatan dan ketelitian. Momentum positifmu sangat terasa. Teruskan ritme dan kualitas ini. Tetapkan target ambisius namun realistis.</p>	<p>Dokumentasi dan penutupan kartu sangat terb. Terima kasih atas kerja hebatnya. Hasil dan kecepatannya mengesankan. Kinerja sangat unggul dan konsisten.</p>
	SANGAT RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 14:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <p>Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	<p>Planning → Monitoring (2025-06-07 10:36)</p> <p>Monitoring → Controlling (2025-06-07 12:50)</p> <p>Controlling → Reflection (2025-06-07 13:11)</p>			
	SANGAT RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:00	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <p>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</p> <p>Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</p>	<p>Planning → Monitoring (2025-06-07 12:04)</p> <p>Monitoring → Controlling (2025-06-07 13:21)</p> <p>Controlling → Reflection (2025-06-07 13:51)</p>			
	SANGAT RAJIN	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 13:00	<p>Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</p> <p>Membuat kelas turunan dan meng-overriding metode dari superclass.</p> <p>Menggunakan polimorfisme melalui overloading dan overriding metode.</p>	<p>Planning → Monitoring (2025-06-07 11:24)</p> <p>Monitoring → Controlling (2025-06-07 11:44)</p> <p>Controlling → Reflection (2025-06-07 13:42)</p>			
	SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 09:00	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <p>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</p> <p>Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	<p>Planning → Monitoring (2025-06-07 10:43)</p> <p>Monitoring → Controlling (2025-06-07 12:09)</p> <p>Controlling → Reflection (2025-06-07 12:27)</p>			
MHS_40	SANGAT RAJIN	Reflection	Enkapsulasi, inheritance, dan Polimorfisme	2025-06-07 11:00	<p>Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</p> <p>Membuat kelas turunan dan meng-overriding metode dari superclass.</p> <p>Menggunakan polimorfisme melalui overloading dan overriding metode.</p>	<p>Planning → Monitoring (2025-06-07 06:52)</p> <p>Monitoring → Controlling (2025-06-07 07:53)</p> <p>Controlling → Reflection (2025-06-07 08:18)</p>	<p>Pertahankan disiplin penutupan yang konsisten. Bila memungkinkan bantu review atau handoff. Pastikan dokumentasi ringkas tersedia saat close. Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Jaga agar tidak ada pekerjaan menggantung.</p>	<p>Tetapkan target ambisius namun realistis. Teruskan ritme dan kualitas ini. Standarmu tinggi dan jadi acuan bagi yang lain. Jaga keseimbangan antara kecepatan dan ketelitian.</p>	<p>Hasil dan kecepatannya mengesankan. Terima kasih atas kerja hebatnya. Dokumentasi dan penutupan kartu sangat terb. Kinerja sangat unggul dan konsisten.</p>
	SANGAT RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 13:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <p>Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	<p>Planning → Monitoring (2025-06-07 07:23)</p> <p>Monitoring → Controlling (2025-06-07 08:04)</p> <p>Controlling → Reflection (2025-06-07 09:28)</p>			
	SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 13:00	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <p>Menentukan relasi antar objek: asosiasi, agregasi, komposisi.</p> <p>Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	<p>Planning → Monitoring (2025-06-07 06:10)</p> <p>Monitoring → Controlling (2025-06-07 07:39)</p> <p>Controlling → Reflection (2025-06-07 07:59)</p>			
	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 12:00	<p>Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP.</p> <p>Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme.</p> <p>Menerapkan konsep dasar objek dan kelas dalam program sederhana.</p>	<p>Planning → Monitoring (2025-06-07 07:32)</p> <p>Monitoring → Controlling (2025-06-07 08:56)</p> <p>Controlling → Reflection (2025-06-07 10:18)</p>			
	SANGAT RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 10:00	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <p>Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class.</p> <p>Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</p>	<p>Planning → Monitoring (2025-06-07 06:53)</p> <p>Monitoring → Controlling (2025-06-07 07:36)</p> <p>Controlling → Reflection (2025-06-07 08:12)</p>			