

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><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	-	Rutin cek ulang sebelum berpindah topik. Fokus menuntaskan satu kartu sampai 'Done' selesai membuat yang lain. Tuliskan 'next step' singkat di setiap perpindahan kolom. Rapiskan urutan kerja agar tidak sering ganti konten. Prioritaskan checklist inti terlebih dahulu.	Konsistensi membuat minimal satu item selesai per hari. Tingkatkan 'next step' menjadi lebih singkat di setiap perpindahan kolom. Rapiskan urutan kerja agar tidak sering ganti konten. Prioritaskan checklist inti terlebih dahulu.	Gooj job, tinggal dipertahankan. Upaya step-by-step untuk memperbaiki konsistennya. Kamu sudah bergerak ke arah yang tepat. Progres teruskan. Ada usaha yang bertujuan dan patut diapresiasi. Terus juga usaha ini agar stabil.
MHS_2	MALAS	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 08:00	<p>Menjelaskan perbedaan paradigma pemrograman procedural dan OOP.</p> <p><input type="checkbox"/> Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme.</p> <p><input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.</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 seleksialah checklist paling mudah terlebih dahulu. Tulis catatan singkat agar langkah berikutnya jelas.	Agresiensi kemajuan selesai apa pun untuk menjaga konsistensi. Kamu bisa mengejar secara berpasang-pasang. Jika macet, pilih satu tugas yang paling sederhana untuk memulai.	Kebutuhan memulai adalah langkah penting. Ada usaha yang terhitung. Meskipun belum berhasil, tetapi masih ada potensi. Terima kasih sudah memiliki proaktifnya.
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><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</p>	-	Hindari pindah kolom sebelum item inti selesai. Tulis catatan singkat agar langkah berikutnya jelas. Mulailah dengan satu kartu prioritas dan seleksialah checklist paling mudah terlebih dahulu. Setelah satu kartu selesai, baru lanjut ke topik berikutnya.	Tetapkan target kecil harian agar ada rasa selesai setiap hari. Apresiasi selesai apa pun untuk menjaga ritme konsisten dan rapida sprint sesaat. Kamu bisa mengejar secara berpasang-pasang. Jika macet, pilih satu tugas yang paling sederhana untuk memulai.	Setiap langkah leci akan berdampak bila konsisten. Ada usaha yang terhitung. Meskipun masih terbatas. Perhatikan hal baik ini.
MHS_4	MALAS	Planning	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-06 17:00	<p><input type="checkbox"/> Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</p> <p><input type="checkbox"/> Membuat kelas turunan dan meng-overriding metode dari superklass.</p> <p><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</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 seleksialah checklist paling mudah terlebih dahulu.	Agresiensi kemajuan selesai apa pun untuk menjaga ritme konsisten dan rapida sprint sesaat. Kamu bisa mengejar secara berpasang-pasang. Jika macet, pilih satu tugas yang paling sederhana untuk memulai.	Setiap langkah leci akan berdampak bila konsisten. Ada usaha yang terhitung. Meskipun masih terbatas. Perhatikan hal baik ini.
MHS_5	MALAS	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-06 23:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <p><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	-	Hindari pindah kolom sebelum item inti selesai. Tulis catatan singkat agar langkah berikutnya jelas.	Tetapkan target kecil harian agar ada rasa selesai setiap hari. Apresiasi selesai apa pun untuk menjaga ritme konsisten dan rapida sprint sesaat. Kamu bisa mengejar secara berpasang-pasang. Jika macet, pilih satu tugas yang paling sederhana untuk memulai.	Setiap langkah leci akan berdampak bila konsisten. Ada usaha yang terhitung. Meskipun masih terbatas. Perhatikan hal baik ini.
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><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	Planning --> Monitoring (2025-06-07 15:45)	Fokus menuntaskan satu kartu sampai 'Done' sebelum membuat yang lain. Rutin cek ulang sebelum berpindah topik. Tuliskan 'next step' singkat di setiap perpindahan kolom. Prioritaskan checklist inti terlebih dahulu.	Bangun momentum dengan kemenganan keadaan. Gunakan ritme konsisten dan seleksialah checklist paling mudah terlebih dahulu. Tulis catatan singkat agar langkah berikutnya jelas.	Terus juga usaha ini agar stabil. Good job, jalil dipertahankan. Upaya step-by-step untuk memperbaiki konsistennya. Kamu sudah bergerak ke arah yang tepat. Progres teruskan. Ada usaha yang bertujuan dan patut diapresiasi.
MHS_7	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><input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.</p>	Planning --> Monitoring (2025-06-07 14:48)	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).	Pertahankan hal baik ini. Setiap langkah leci akan berdampak bila konsisten.	
MHS_7	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><input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.</p>	Planning --> Monitoring (2025-06-07 14:19)	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).	Terima kasih sudah memulai prosesnya. Ada usaha yang terhitung. Meskipun masih terbatas. Perhatikan hal baik ini.	
MHS_7	SEDIKIT MALAS	Planning	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-05 08:00	<p><input type="checkbox"/> Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</p> <p><input type="checkbox"/> Membuat kelas turunan dan meng-overriding metode dari superklass.</p> <p><input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode.</p>	-	Hindari pindah kolom sebelum item inti selesai. Setelah satu kartu selesai, baru lanjut ke topik berikutnya. Mulailah dengan satu kartu prioritas dan seleksialah checklist paling mudah terlebih dahulu.	jika macet, pilih satu tugas yang paling sederhana untuk memulai.	Pertahankan hal baik ini. Setiap langkah leci akan berdampak bila konsisten. Ada usaha yang terhitung. Meskipun masih terbatas. Perhatikan hal baik ini.
MHS_7	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><input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode.</p>	-	Mendesain class diagram berdasarkan kebutuhan dari studi kasus.	Terus juga usaha ini agar stabil. Good job, jalil dipertahankan. Upaya step-by-step untuk memperbaiki konsistennya. Kamu sudah bergerak ke arah yang tepat. Progres teruskan. Ada usaha yang bertujuan dan patut diapresiasi.	

	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-06 04:00	<ul style="list-style-type: none"> <input type="checkbox"/> 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 16:11)			
	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 08:00	<ul style="list-style-type: none"> Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Menerapkan rancangan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	Planning → Monitoring (2025-06-07 14:26)			
MHS_8	SEDIKIT MALAS	Planning	Konsep Dasar dan Pilar OOP	2025-06-06 06:00	<ul style="list-style-type: none"> <input type="checkbox"/> Menjelaskan perbedaan paradigma pemrograman procedural 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. 		<p>Jaga ritme konsisten dalam agenda sprint sekarang. Kamu bisa mengejar berikutnya secara bertahap; tidak perlu segera sempurna di awal.</p> <p>Tulis catatan singkat agar langkah berikutnya jelas. Mulalah dengan satu kartu prioritas dan seleksikan checklist yang mudah terlebih dahulu. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus. Hindari pindah kolom sebelum item itu selesai.</p>	<p>Pertahankan ritme yang setiap langkah setidaknya sempurna. Aprestasi konsisten memanglah hal yang penting. Ada usaha yang benar dan meskipun masih terbatas, Terima kasih sudah memiliki prosesnya.</p>	
MHS_8	SEDIKIT MALAS	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-05 06:00	<ul style="list-style-type: none"> <input type="checkbox"/> Mendesain class diagram berdasarkan kebutuhan dari studi kasus. <input type="checkbox"/> Menentukan relasi antar objek: asosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode. 	Planning → Monitoring (2025-06-07 15:01)			
MHS_8	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-05 06:00	<ul style="list-style-type: none"> Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Menerapkan rancangan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	Planning → Monitoring (2025-06-07 15:40)			
MHS_9	SEDIKIT MALAS	Planning	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 12:00	<ul style="list-style-type: none"> 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. 		<p>Hindari pindah kolom sebelum item itu selesai.</p> <p>Mulalah dengan satu kartu prioritas dan seleksikan checklist yang mudah terlebih dahulu. Tulis catatan singkat agar langkah berikutnya jelas. Susun waktu kerja pendek (15–25 menit) untuk menjaga fokus.</p>	<p>Aprestasi konsisten selama puncak selesai. Jaga ritme konsisten dari pada sprint semangat. Jaga pertahankan ritme konsisten dari pada sprint sekarang. Jika macet, pilih satu tugas yang paling segera selesaikan segera.</p> <p>Terima kasih sudah memiliki prosesnya. Pertahankan niat baik ini. Terima kasih juga yang terhitah meskipun masih terbatas. Setelah langkah berikutnya berlaku berdampak bila konsisten.</p>	
MHS_9	SEDIKIT MALAS	Planning	Perancangan Kelas dan Interaksi Objek	2025-06-05 19:00	<ul style="list-style-type: none"> 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. 				
MHS_9	SEDIKIT MALAS	Monitoring	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-06 15:00	<ul style="list-style-type: none"> <input type="checkbox"/> 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. 	Planning → Monitoring (2025-06-07 08:59)			
MHS_10	SEDIKIT MALAS	Planning	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 04:00	<ul style="list-style-type: none"> 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. 		<p>Sertakan satu kartu selesai, baru lanjut ke topik berikutnya.</p> <p>Mulalah dengan satu kartu prioritas dan seleksikan checklist paling mudah terlebih dahulu. Hindari pindah kolom sebelum kolom sebelumnya.</p>	<p>Jaga ritme konsisten dan rasa selesaikan. Kamu bisa mengejar secara bertahap; tidak perlu segera sempurna di awal.</p> <p>Terima kasih sudah memiliki prosesnya. Ada usaha yang benar dan meskipun masih terbatas. Setelah langkah berikutnya berlaku berdampak bila konsisten.</p>	
MHS_10	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-10 10:00	<ul style="list-style-type: none"> <input type="checkbox"/> 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 10:10)			
MHS_10	SEDIKIT MALAS	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-06 06:00	<ul style="list-style-type: none"> Menjelaskan perbedaan paradigma pemrograman procedural 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 09:37)			
MHS_11	SEDIKIT MALAS	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 19:00	<ul style="list-style-type: none"> 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. 		<p>Fokus mempersiapkan satu kartu sampai "Done" sebelum membuka yang lain.</p> <p>Prioritaskan checklist inti terlebih dahulu.</p> <p>Rapikan urutan kerja agar tidak sering ganti konten.</p>	<p>Targetkan minimal satu kartu selesai per hari untuk memenuhi rincie sedikit demi sedikit.</p> <p>Bangun momentum dengan kerangka keadaan.</p> <p>Tuliskan "next step" singkat di setiap perpindahan kolom. Rutin cek ulang sebelum berpindah topik.</p>	<p>Terima kasih juga lewat in agar stabil. Kamu sudah bergerak ke arah yang tepat. Upaya step-by-step membutuhkan komitmen. Good job, tinggal diperpanjang. Terima kasih banyak. Terima kasih dan patut dipresasi.</p>
MHS_11	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 03:00	<ul style="list-style-type: none"> Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Menerapkan rancangan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	Planning → Monitoring (2025-06-07 13:18)			
MHS_11	SEDIKIT MALAS	Monitoring	Perancangan Kelas dan Interaksi Objek	2025-06-05 15:00	<ul style="list-style-type: none"> Menjelaskan diagram berdasarkan kebutuhan dari studi kasus. <input type="checkbox"/> Menentukan relasi antar objek: asosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode. 	Planning → Monitoring (2025-06-07 11:25)			
MHS_12	SEDIKIT MALAS	Planning	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 10:00	<ul style="list-style-type: none"> 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. 		<p>Fokus menuntaskan satu kartu sampai "Done" sebelum membuka yang lain.</p> <p>Rapikan urutan kerja agar tidak sering ganti konten.</p> <p>Prioritaskan checklist inti terlebih dahulu.</p> <p>Rutin cek ulang sebelum berpindah topik.</p>	<p>Kamu si jalur yang benar—teruskan. Kamu sudah bergerak ke arah yang tepat.</p> <p>Good job, tinggal diperpanjang. Progres mulai terlihat dan patut dipresasi.</p> <p>Upaya step-by-step membutuhkan komitmen. Kamu sudah bergerak ke arah yang tepat.</p>	
MHS_12	SEDIKIT MALAS	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 03:00	<ul style="list-style-type: none"> <input type="checkbox"/> 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 12:41)			
MHS_12	SEDIKIT MALAS	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-05 11:00	<ul style="list-style-type: none"> Menjelaskan perbedaan paradigma pemrograman procedural 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 13:02)			
MHS_12	SEDIKIT MALAS	Monitoring	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-06 18:00	<ul style="list-style-type: none"> Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. Menerapkan rancangan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	Planning → Monitoring (2025-06-07 15:20)			

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

CUKUP	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 14:12	<p>Menjelaskan perbedaan paradigma pemrograman procedural dan OOP.</p> <ul style="list-style-type: none"> <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. 	<p>Planning --> Monitoring (2025-06-07 10:04)</p> <p>Monitoring --> Controlling (2025-06-07 10:20)</p>			
CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:10	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	<p>Planning --> Monitoring (2025-06-07 08:35)</p> <p>Monitoring --> Controlling (2025-06-07 09:04)</p>			
MHS_19				<p>Menjelaskan perbedaan paradigma pemrograman procedural dan OOP.</p> <ul style="list-style-type: none"> <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. 		<p>Tambahkan ringkasan singkat saat close untuk dokumentasi. Pertimbangkan alur yang masih terbentuk. Alurnya sudah rapi; fokus pada finalisasi checklist penting. Selesaikan satu topik tuntas sebelum pindah. Kurangi perunduan kecil agar tidak menumpuk di akhir.</p>	<p>Teruskan ketekunan bahwa yang sudah ada. Tetapkan target harian yang realistik. Jaga tempo agar tidak jatuh. Dengan lanjutkan setelahnya. Sedikit dorongan lagi untuk tembus "Done" kondisi.</p>	<p>Ritme kerja yang terjaga patut diapresiasi. Keramu stabil dan konsisten. Dokumentasi dan perpindahan kolom sudah cukup baik. Banyakkan performa ini. Kualitas progresnya terhitah.</p>
CUKUP	Planning	Konsep Dasar dan Pilar OOP	2025-06-06 03:00	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Melakukan refactoring kode agar sesuai dengan prinsip SOLID. <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP. 	<p>Planning --> Monitoring (2025-06-07 11:55)</p>			
CUKUP	Controlling	Perancangan Kelas dan Interaksi Objek	2025-06-07 15:40	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Menentukan relasi antar objek: assosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode. 	<p>Planning --> Monitoring (2025-06-07 12:56)</p> <p>Monitoring --> Controlling (2025-06-07 14:00)</p>			
CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 14:11	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	<p>Planning --> Monitoring (2025-06-07 11:30)</p> <p>Monitoring --> Controlling (2025-06-07 12:29)</p>			
MHS_20				<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Mengorganisasi struktur folder proyek-OOP berdasarkan lapisan fungsional. <input type="checkbox"/> Mengimplementasikan interaksi antar objek dengan menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 		<p>Rapikan urutan kerja agar tidak sering ganti konteks. Rutin cari update seluruh perpindahan kolom. Fokus menuntaskan satu kartu saja sampai 'Done' sebelum membuka yang lain. Prioritaskan checklist inti terlebih dahulu.</p>	<p>Bangun momentum dengan menyelesaikan kecil. Konsistensi akan membuat progresmu terus bergerak ke arah yang benar. Targetkan jaga usaha ini agar stabil. Good job, tinggal berusaha terus. Tingkatkan ritme sedikit demi sedikit.</p>	<p>Upaya step-by-step untuk menjunjukkan konsistenni. Kamu sudah bergerak ke arah yang benar. Tingkatkan jaga usaha ini agar stabil. Good job, tinggal berusaha terus. Tingkatkan progres mulai terhitah dan patut diapresiasi.</p>
CUKUP	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 14:00	<p>Melakukan refactoring kode agar sesuai dengan prinsip SOLID.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP. 	<p>Planning --> Monitoring (2025-06-07 13:31)</p>			
CUKUP	Monitoring	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 15:54	<p>Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected).</p> <ul style="list-style-type: none"> <input type="checkbox"/> Membuat kelas turunan dan meng-overriding metode dari superclass. <input type="checkbox"/> Menggunakan polimorfisme melalui overloading dan overriding metode. 	<p>Planning --> Monitoring (2025-06-07 13:25)</p> <p>Monitoring --> Controlling (2025-06-07 14:07)</p>			
CUKUP	Controlling	Perancangan Kelas dan Interaksi Objek	2025-06-07 14:10	<p>Mendesain class diagram berdasarkan kebutuhan dari studi kasus.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Menentukan relasi antar objek: assosiasi, agregasi, komposisi. <input type="checkbox"/> Mengimplementasikan rancangan interaksi objek dalam bentuk kode. 	<p>Planning --> Monitoring (2025-06-07 12:25)</p> <p>Monitoring --> Controlling (2025-06-07 14:02)</p>			
CUKUP	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 15:31	<p>Menjelaskan perbedaan paradigma pemrograman procedural dan OOP.</p> <ul style="list-style-type: none"> <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. 	<p>Planning --> Monitoring (2025-06-07 07:29)</p>			
MHS_21				<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>Tambahkan catatan singkat agar mutlak dilacak. Jaga kualitas sambil menjaga kecepatan. Cegah carry-over dengan menutup kartu lama terlebih dahulu. Ekskulus sudah kuat; segera finalkan item tersisa agar cepat 'Done'.</p>	<p>Kamu pereda di jalur yang benar. Sedikit lagu untuk merapikan semua kartu aktiv. Jangan lupa setelah selesai harian sebagai pendorong. Energi positifmu terlahir – lanjutkan.</p>	<p>Kualitas pekerjaan tampak konsisten. Produktivitasmu terus meningkat. Performa solid dan efisien. Kerja bagus – teruskan standar ini.</p>
CUKUP	Controlling	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 14:33	<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 08:10)</p> <p>Monitoring --> Controlling (2025-06-07 09:56)</p>			
CUKUP	Controlling	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 15:01	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	<p>Planning --> Monitoring (2025-06-07 08:42)</p> <p>Monitoring --> Controlling (2025-06-07 10:14)</p>			
MHS_22				<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>Pastikan dokumentasi ringkas tersedia saat close. Bila memungkinkan, bantu review atau handoff.</p> <p>Jaga agar tidak ada pekerjaan yang mengganggu. Pertimbangkan disiplin penutupan yang konsisten. Ekskulus sangat rapat fokus pada finishing dan validasi akhir.</p>	<p>Momentum positifmu sangat terasa. Tetapkan target ambisius agar harian sebagai pendorong. Teruskan ritme dan kualitas ini.</p>	<p>Dokumentasi dan penutupan kartu sangat terasa. Keterbukaan dan konsistennya. Produktivitasmu terus meningkat nyata. Performa solid dan efisien. Dan acceptansnya mengesankan.</p>
CUKUP	Monitoring	Prinsip SOLID dan Adaptive Code	2025-06-07 14:25	<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 08:54)</p>			
MHS_23				<p>Menjelaskan perbedaan paradigma pemrograman procedural dan OOP.</p> <ul style="list-style-type: none"> <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. 		<p>Planning --> Monitoring (2025-06-07 09:26)</p> <p>Monitoring --> Controlling (2025-06-07 09:56)</p> <p>Controlling --> Reflection (2025-06-07 11:04)</p>	<p>Sedikit lagi untuk merapikan semua kartu aktiv. Jangan lupa setelah selesai harian sebagai pendorong. Peningkatan konsistensi dan kecepatan dan kualitas ini.</p>	<p>Transisi antar kolom rapi dan teratur. Produktivitasmu terus meningkat nyata. Kerja bagus – teruskan standar ini. Performa solid dan efisien.</p>
CUKUP	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 14:36	<p>Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID.</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 09:19)</p> <p>Monitoring --> Controlling (2025-06-07 12:01)</p> <p>Controlling --> Reflection (2025-06-07 12:44)</p>			
MHS_23				<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 09:01)</p>	<p>Jaga kualitas sambil menjaga kecepatan. Ekskulus sudah kuat; segera finalkan item tersisa agar cepat 'Done'.</p>	<p>Dokumentasi dan penutupan kartu sangat terasa. Keterbukaan dan konsistennya. Produktivitasmu terus meningkat nyata. Kerja bagus – teruskan standar ini. Performa solid dan efisien.</p>
CUKUP	Controlling	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 14:35	<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 08:50)</p> <p>Monitoring --> Controlling (2025-06-07 09:13)</p>			
CUKUP	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 14:18	<p>Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular. 	<p>Planning --> Monitoring (2025-06-07 08:59)</p> <p>Monitoring --> Controlling (2025-06-07 09:47)</p> <p>Controlling --> Reflection (2025-06-07 11:44)</p>			

						Lakukan pengecekan detail kecil sebelum harian penutupan. Jaga kualitas sambil menjaga kecepatan. Cegah carry-over dengan menutup kartu lama terlebih dahulu. Eksusi sudah kuat; segera finalkan item tersisa agar cepat. Tambahkan catatan singkat agar mudah dilacak.	Jadikan target selesai harian sebagai prioritas. Sedikit lagi untuk meraih semua kartu tertiak. Kamu berada di jalur yang tepat. Pertahankan konsistensi dan pace saat ini.	Kualitas pelaporan tanggap. Produktivitas ini memberi dampak nyata. Transisi antar kolom rapi dan teratur. Kerja bagus.	
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)	Eksusi sudah kuat; segera finalkan item tersisa agar cepat "Done". Lakukan pengecekan detail kecil sebelum penutupan. Cegah 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 meraih semua kartu tertiak. Kamu juga berhasil—lanjutkan. Jadikan target selesai harian sebagai prioritas.	
	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. □ Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	-	Tambahkan ringkasan singkat saat close untuk dokumentasi. Selesaikan satu topik tuntas sebelum pindah. Perhatikan alur yang sudah terbentuk. Kurangi penundaan kecil agar tidak menumpuk di akhir. Alurnya sudah rapi; fokus pada finalisasi checklist penting.	Tetapkan target selesai harian yang terjaga patut. Teruskan kebiasaan baik yang sudah ada di tengah jalur—lanjutkan. Dokumentasi dan perbaikan selalu diakhiri ketika selesai.	
	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. □ Mengenali 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. □ 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 menumpuk di akhir. Tambahkan ringkasan singkat saat close untuk dokumentasi. Selesaikan satu topik tuntas sebelum pindah. Alurnya sudah rapi; fokus pada finalisasi checklist penting. Perhatikan alur yang sudah terbentuk.	Jaga tempo agar tidak terlambat. Tetapkan target selesai harian yang terjaga patut. Teruskan kebiasaan baik yang sudah ada. Dokumentasi dan perbaikan selalu diakhiri ketika selesai.	
	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. □ 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. □ 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. □ Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Tambahkan ringkasan singkat saat close untuk dokumentasi. Selesaikan satu topik tuntas sebelum pindah. Alurnya sudah rapi; fokus pada finalisasi checklist penting. Perhatikan alur yang sudah terbentuk.	Sedikit dorongan lagi untuk tembus "Done" konsisten. Kurangi penundaan kecil agar tidak menumpuk di akhir. Selesaikan satu topik tuntas sebelum pindah. Resmi selesai. Sudah ada di tengah jalur—lanjutkan. Jaga tempo agar tidak turun.	Pertahankan performa ini. Dokumentasi dan perbaikan selalu diakhiri ketika selesai.
	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. □ 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)			

	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)			
MHS_29	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 projek mini berbasis OOP.	-	Alurnya sudah rapi; fokus pada finalisasi checklist penting. Pertahanan alur yang sudah terlalu banyak. Kurang penundukan kecil agar tidak memenuhi di akhir. Tambahkan ringkasan singkat saat close untuk dokumentasi.	Tetapkan target harian yang relevan. Terlakukan kebiasaan baik yang sudah ada. Jaga tempo agar tidak terlambat. Sedikit dorongan lagi untuk tembus "Done". Konsisten.	
MHS_30	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. <input type="checkbox"/> Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 11:00)			
MHS_30	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.	Planning → Monitoring (2025-06-07 11:09) Monitoring → Controlling (2025-06-07 11:58)			
MHS_30	RAJIN	Controlling	Konsep Dasar dan Pilar OOP	2025-06-07 16:00	<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 11:02) Monitoring → Controlling (2025-06-07 12:35)			
MHS_30	RAJIN	Controlling	Enkapsulasi, 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)			
MHS_31	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 07:00	Menjelaskan perbedaan paradigma pemrograman procedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. <input type="checkbox"/> Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Cegah carry-over dengan menutup kartu lama terlebih dahulu. Jaga kualitas sambil menjaga kecepatan. Ekoksu sudah kuat; segera finalkan item terisya agar cepat "Done". Lakukan pengecekan detail kecil sebelum penutupan.	Energi positifmu terlahir— lanjutkan. Jadikan target seluruh harian sebagai prioritas. Performa solid dan efisien. Kerja bagus— teruskan standar ini. Produktivitas dan inovasi dampaikan nyata. Kualitas pekerjaan tetap lancar dan konsisten.	
MHS_31	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)			
MHS_31	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 11:40) Monitoring → Controlling (2025-06-07 11:58) Controlling → Reflection (2025-06-07 12:51)			
MHS_31	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 rangange 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 projek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:02) Monitoring → Controlling (2025-06-07 12:12) Controlling → Reflection (2025-06-07 12:31)			
MHS_32	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 09:00	Menjelaskan perbedaan paradigma pemrograman procedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Jaga kelembaban antara kecepatan dan ketelitian. Ekoksu sangat kuat; segera finalkan item terisya agar cepat "Done". Lakukan pengecekan detail kecil sebelum penutupan.	Dokumentasi dan penutupan yang sangat terinci. Kinerja sangat unggul dan efisien. Hasil dan kecepatannya mengesankan. Momentum positifnya sangat atas kerja habis.	
MHS_32	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 projek mini berbasis OOP.	Planning → Monitoring (2025-06-07 12:00)			
MHS_32	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 rangange 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)			
MHS_32	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_33	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 rangange interaksi objek dalam bentuk kode.	-	Cegah carry-over dengan menutup kartu lama terlebih dahulu. Ekoksu sudah kuat; segera finalkan item terisya agar cepat "Done". Lakukan pengecekan detail kecil sebelum penutupan.	Energi positifmu terlahir— lanjutkan. Jadikan target seluruh harian sebagai prioritas. Produktivitas u memberi dampak nyata. Performa solid dan efisien. Kualitas pekerjaan tetap lancar dan konsisten.	
MHS_33	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 projek mini berbasis OOP.	Planning → Monitoring (2025-06-07 10:13)			
MHS_33	RAJIN	Controlling	Enkapsulasi, 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)			
MHS_33	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 procedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	-	Lakukan pengecekan detail kecil sebelum penutupan. Tambahkan catatan singkat agar mudah dilacak.	Sedikit lagi untuk merubah sertai kartu aktif. Ekoksu sudah kuat; segera finalkan item terisya agar cepat "Done". Jaga kualitas sambil menjaga kecepatan.	Produktivitas u memberi dampak nyata. Performa kolom rapi dan teratur. Kerja bagus— teruskan standar ini. Performa solid dan efisien.
MHS_33	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)			
MHS_33	RAJIN	Controlling	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 12:00	Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected). Menentukan relasi antar objek: asosiasi, agregasi, komposisi. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning → Monitoring (2025-06-07 09:40) Monitoring → Controlling (2025-06-07 10:21)			
MHS_33	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 projek 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 mengantang. Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pertahankan disiplin penutupan yang konsisten.	Tetapkan target ambisius namun realistik. Momentum positifmu tinggi dan jadi acuan bagi yang lain. Jaga kelembangan akurasi kecepatan dan ketelitian.	Terima kasih bahwa resepsi berjalan Kinerja sangat unggul dan konsisten. Standarmu jelas terlihat pada alur kerja. Dokumentasi dan penutupan kartu sangat tertib.
MHS_34	RAJIN	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 07:00	Menjelaskan perbedaan paradigma pemrograman procedural 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 fungisional. Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class. Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.	Planning → Monitoring (2025-06-07 09:53)		
	RAJIN	Controlling	Enkapsulasi, Inheritance, dan Polimorfisme	2025-06-07 08:00	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 projek mini berbasis OOP.	Planning → Monitoring (2025-06-07 11:21) Monitoring → Controlling (2025-06-07 11:37) Controlling → Reflection (2025-06-07 13:13)	Cegah carry-over dengan memulai kartu lama terlebih dahulu. Eksekusi sudah kuat; segera finalkan item tersebut. Cepat "Done". Jaga kualitas sambil menjaga kecepatan. Lakukan pengcekan detail kecil sebelum penutupan.	
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.	-	Pertahankan konsistensi dan pace saat ini. Jadikan target selanjutnya sebagai pendorong. Jaga kualitas sambil menjaga kecepatan. Lakukan pengcekan detail kecil sebelum penutupan.	Kualitas pekerjaan tampak konsisten. Kerja bagus – tetapi belum standar. Transisi antara kolom rapi dan teratur. Hasil termasuk solid dan efisien. Produktivitas u memberi dampak nyata.
	RAJIN	Monitoring	Konsep Dasar dan Pilar OOP	2025-06-07 07:00	Menjelaskan perbedaan paradigma pemrograman procedural 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 projek 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). 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)	Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pertahankan detail dan penutupan yang konsisten. Pastikan dokumentasi ringkas tersebut saat close. Bila memungkinkan, bantu review atau handoff.	
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)	Standarmu sangat baik dan konsisten. Hasil dan kecepatannya sangat baik. Dokumentasi dan penutupan kartu sangat tertib. Terima kasih atas kerja hebatnya.	
	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 projek 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	Menjelaskan perbedaan paradigma pemrograman procedural 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 10:36) 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 fungisional. 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)		
MHS_37	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 projek 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. Pertahankan detail dan penutupan yang konsisten. Pastikan dokumentasi ringkas tersebut saat close. Teruskan ritme dan kualitas ini. Jaga agar tidak ada pekerjaan mengantang. Pertahankan disiplin penutupan yang konsisten.	
	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 10:00	Menjelaskan perbedaan paradigma pemrograman procedural 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)	Momentum positifmu sangat baik dan konsisten. Dokumentasi dan penutupan kartu sangat tertib. Hasil dan kecepatannya sangat baik. Komunikasi dan koordinasi tetap terjalin dengan baik.	
	SANGAT RAJIN	Reflection	Kolaborasi Objek dan Pengorganisasian Kode	2025-06-07 08:00	Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungisional. 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)	Jaga kelembangan antara kecepatan dan ketelitian. Tinggi dan jadi acuan bagi yang lain. Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pastikan dokumentasi ringkas tersebut saat close. Teruskan ritme dan kualitas ini. Tetapkan target ambisius namun realistik.	
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 mengantang. Eksekusi sangat rapi; fokus pada finishing dan validasi akhir. Pastikan dokumentasi ringkas tersebut saat close. Teruskan ritme dan kualitas ini. Tetapkan target ambisius namun realistik.	
	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 13:00	Menjelaskan perbedaan paradigma pemrograman procedural 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 fungisional. 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 projek 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 09:46) Monitoring → Controlling (2025-06-07 11:10) Controlling → Reflection (2025-06-07 12:54)	Hasil dan kecepatannya sangat mengejarkan. Terima kasih atas kerja kerasmu. Komunikasi dan koordinasi tetap terjalin dengan baik.	

							Pertahanan disiplin penutupan yang konsisten, dan jadi keselarasan bagi yang ada pekerjaan mengantang. Eksekusi fokus pada finishing dan validasi akhir. Pastikan dokumentasi ringkas tersebut saat close. Bila memungkinkan, bantu review atau handoff.	Standarmu tinggi dan keseimbangan antara kebutuhan dan kualitas. Hasil dan kecepatannya sangat terasa. Teruskan rime dan kualitas ini. Tetapkan target ambisius namun realistik.	Dokumentasi dan penutupan kartu sangat terit. Terima kasih atas kerja kerasnya. Hasil dan kecepatannya mengesangkan. Kinerja sangat unggul dan konsisten.
MHS_39	SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 09:00	Menjelaskan perbedaan paradigma pemrograman procedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pemisalan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning --> Monitoring (2025-06-07 10:22) Monitoring --> Controlling (2025-06-07 12:07) Controlling --> Reflection (2025-06-07 12:57)			
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 10:36) Monitoring --> Controlling (2025-06-07 12:00) Controlling --> Reflection (2025-06-07 13:11)				
SANGAT RAJIN	Reflection	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 12:04) Monitoring --> Controlling (2025-06-07 13:21) Controlling --> Reflection (2025-06-07 13:51)				
SANGAT RAJIN	Reflection	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. Menggunakan polimorfisme melalui overloading dan overriding metode.	Planning --> Monitoring (2025-06-07 11:24) Monitoring --> Controlling (2025-06-07 11:44) Controlling --> Reflection (2025-06-07 13:42)				
SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 09:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: assosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning --> Monitoring (2025-06-07 10:43) Monitoring --> Controlling (2025-06-07 12:09) Controlling --> Reflection (2025-06-07 12:27)				
MHS_40	SANGAT 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 06:52) Monitoring --> Controlling (2025-06-07 07:53) Controlling --> Reflection (2025-06-07 08:18)	Persehankan disiplin penutupan yang konsisten. Bila memungkinkan, namun realistik. Tetapkan target ambisius namun realistik. Teruskan rime dan kualitas ini. Standermu tinggi dan jadi keselarasan antara kebutuhan dan kualitas ini. Jaga agar tidak ada pekerjaan mengantang.	Hasil dan kecepatannya mengesangkan. Terima kasih atas kerja kerasnya. Dokumentasi dan penutupan kartu sangat terit. Terima kasih atas kerja kerasnya. Hasil dan kecepatannya sangat unggul dan konsisten.	
SANGAT RAJIN	Reflection	Prinsip SOLID dan Adaptive Code	2025-06-07 13:00	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 07:23) Monitoring --> Controlling (2025-06-07 08:04) Controlling --> Reflection (2025-06-07 09:28)				
SANGAT RAJIN	Reflection	Perancangan Kelas dan Interaksi Objek	2025-06-07 13:00	Mendesain class diagram berdasarkan kebutuhan dari studi kasus. Menentukan relasi antar objek: assosiasi, agregasi, komposisi. Mengimplementasikan rancangan interaksi objek dalam bentuk kode.	Planning --> Monitoring (2025-06-07 06:10) Monitoring --> Controlling (2025-06-07 07:39) Controlling --> Reflection (2025-06-07 07:59)				
SANGAT RAJIN	Reflection	Konsep Dasar dan Pilar OOP	2025-06-07 12:00	Menjelaskan perbedaan paradigma pemrograman procedural dan OOP. Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pemisalan, Polimorfisme. Menerapkan konsep dasar objek dan kelas dalam program sederhana.	Planning --> Monitoring (2025-06-07 07:32) Monitoring --> Controlling (2025-06-07 08:56) Controlling --> Reflection (2025-06-07 10:18)				
SANGAT 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 06:53) Monitoring --> Controlling (2025-06-07 07:36) Controlling --> Reflection (2025-06-07 08:12)				