



## Attachment 1 Observation Letter



KEMENTERIAN PENDIDIKAN TINGGI,  
SAINS, DAN TEKNOLOGI  
UNIVERSITAS PENDIDIKAN GANESHA  
FAKULTAS TEKNIK DAN KEJURUAN  
Jalan Udayana Nomor 11 Singaraja Bali  
Laman: <http://ftk.undiksha.ac.id>

Nomor : 622/UN48.11.1/KM/2025

Singaraja, 7 Maret 2025

Perihal : Surat Permohonan Pengambilan Data

Yth. Kepala SMAN 1 Blahbatuh  
di tempat

Dengan hormat, sehubungan dengan proses penyelesaian Tugas Akhir/Skripsi, maka melalui surat ini kami mohon Bapak/Ibu berkenan memberikan data yang terkait dengan data yang dibutuhkan. Adapun mahasiswa yang akan melakukan pengambilan data seperti tersebut di bawah ini:

Nama : Anak Agung Gede Pradnya Prameswara  
NIM : 2215051040  
Program Studi : Pendidikan Teknik Informatika  
Jurusan : Teknik Informatika  
Data yang dibutuhkan : Wawancara ini menggali metode evaluasi, penyusunan soal, dan pembuatan Katalog diagnostik. Guru juga diminta menjelaskan tantangan menghadapi Siswa yang sering lupa materi serta strategi yang diterapkan untuk meningkatkan Retensi Pembelajaran  
Judul Skripsi : Pengembangan Sistem Diagnostik Cerdas untuk Evaluasi Pembelajaran Siswa di SMAN 1 Blahbatuh

Demikian kami sampaikan, atas perhatian dan kerjasamanya, diucapkan terima kasih.

a.n Dekan  
Wakil Dekan Bidang Akademik,  
  
Made Windu Antara Kesiman  
NIP 198211112008121001

## Attachment 2 Disposition Sheet

( P ) PENTING		( B ) BIASA		( R ) RAHASIA	
Tanggal Diterima	7/3-25	Tgl. Surat Masuk	: 9/3/25		
No. Agenda Surat Masuk	/ 1. 19. 5.1 / SMA / 2025	No	: -		
		Tgl. Penyelesaian	:		
Asal Surat	Undiksha				
Perihal	Formulir permohonan surat pengantar				
Instruksi / Informasi			Diteruskan Kepada		
Tolong ditindaklanjuti.			Pk. A.A. Gd. Agung		



Attachment 4 Guidelines & Results of Interviews with ICT Subject Teachers (1)

Soal 1: Kenapa web sekolah masih halaman yang lama-lama?

Jawaban Guru:

**Web sekolah tidak update karena pihak yang bertanggung jawab, khususnya tim jurnalis osis, kurang proaktif. Padahal banyak kegiatan sekolah seperti ulang tahun dan Saraswati, tapi beritanya tidak diunggah. Guru sudah mengingatkan, namun mereka harus terus diingatkan. Guru berharap mereka lebih aktif mendokumentasikan kegiatan, membuat berita, atau artikel singkat agar web tidak stagnan. Sekolah sudah menyediakan "rumah" (web), tinggal bagaimana siswa mengisinya dengan foto, berita, artikel, atau video. Siswa juga sudah diajarkan cara mengunggahnya.**

Soal 2: Bagaimana cara Bapak mengatasi siswa yang sering lupa materi yang sudah diajarkan, misalnya di bidang IT, sudah diajarkan HTML tapi lupa saat mau belajar CSS?

Jawaban Guru:

**Cara mengatasinya adalah dengan mengulang materi. Guru harus sabar mengulang karena ini adalah kenyataan di lapangan. Siswa, terutama di mata pelajaran Informatika, mungkin bisa saat diajarkan, tapi besoknya bisa lupa. Ini karena kemungkinan siswa hanya praktik di sekolah, padahal materi Informatika harus sering dipraktikkan di luar sekolah atau di rumah. Jika siswa punya inisiatif untuk mencoba dan mengembangkan, itu akan lebih bagus. Ada siswa yang bahkan kesulitan untuk hal paling dasar seperti membuat folder dan menyimpan file. Jika dasar saja belum bisa, akan sulit mengajarkan pemrograman seperti HTML.**

Soal 3: Bagaimana pandangan Bapak tentang pemanfaatan teknologi AI sebagai pembantu dalam proses asesmen untuk masalah siswa yang sering lupa materi?

Jawaban Guru:

**Guru jarang memanfaatkan AI untuk mengevaluasi siswa secara langsung. Evaluasi siswa biasanya berdasarkan real-nya per individu. Namun, ide**

untuk mengembangkan AI sebagai asisten guru yang bisa mempersonalisasi materi dan evaluasi (seperti untuk materi HTML atau CSS) adalah ide yang bagus dan efektif.



## Attachment 5 Guidelines &amp; Results of Interviews with ICT Subject Teachers (2)

Soal 1: Bagaimana Ibu mengidentifikasi kesulitan siswa yang mengalami kesulitan pemahaman materi yang pernah Ibu ajarkan sebelumnya? Apakah berdasarkan benar atau salahnya saja?

Jawaban Guru:

**Kesulitan utama dalam mengidentifikasi kesulitan siswa terletak pada niat siswa itu sendiri untuk memperbaiki kemampuan mereka. Meskipun guru sudah berulang kali melakukan evaluasi dan remedi, jika siswa tidak memiliki niat untuk memperbaiki diri, maka akan sulit. Guru mengidentifikasi kesulitan siswa berdasarkan hasil tes atau soal yang dibuat sesuai dengan tujuan pembelajaran. Mereka melihat tujuan pembelajaran mana yang belum tercapai oleh siswa, atau indikator mana yang belum dikuasai siswa. Intinya, kemauan atau niat siswa untuk memperbaiki diri (belajar lagi di rumah, bertanya, dll.) adalah kunci.**

Soal 2: Bagaimana Ibu menyesuaikan metode pembelajaran di kelas dengan mengetahui siswa sering lupa materi yang Ibu ajarkan sebelumnya?

Jawaban Guru:

**Metode pembelajaran harus bervariasi atau menggunakan metode diferensiasi. Guru harus menyesuaikan pendekatan yang sesuai dengan cara belajar dan metode belajar siswa yang berbeda-beda.**

Soal 3: Bagaimana pandangan Ibu mengenai pemanfaatan teknologi, khususnya AI, dalam membantu proses asesmen diagnostik dan evaluasi?

Jawaban Guru:

**Pemanfaatan AI dalam asesmen diagnostik dan evaluasi sangat membantu karena mempercepat proses, seperti dalam menganalisis data atau membuat soal otomatis. Namun, guru lebih suka membuat soal sendiri agar sesuai dengan karakteristik siswa, karena soal yang dihasilkan AI cenderung general atau umum. Mengenai rekomendasi pembelajaran interaktif dari AI, guru berpandangan bahwa jika ada AI yang bisa interaktif untuk proses pembelajaran, itu sangat bagus. AI bisa mencari**

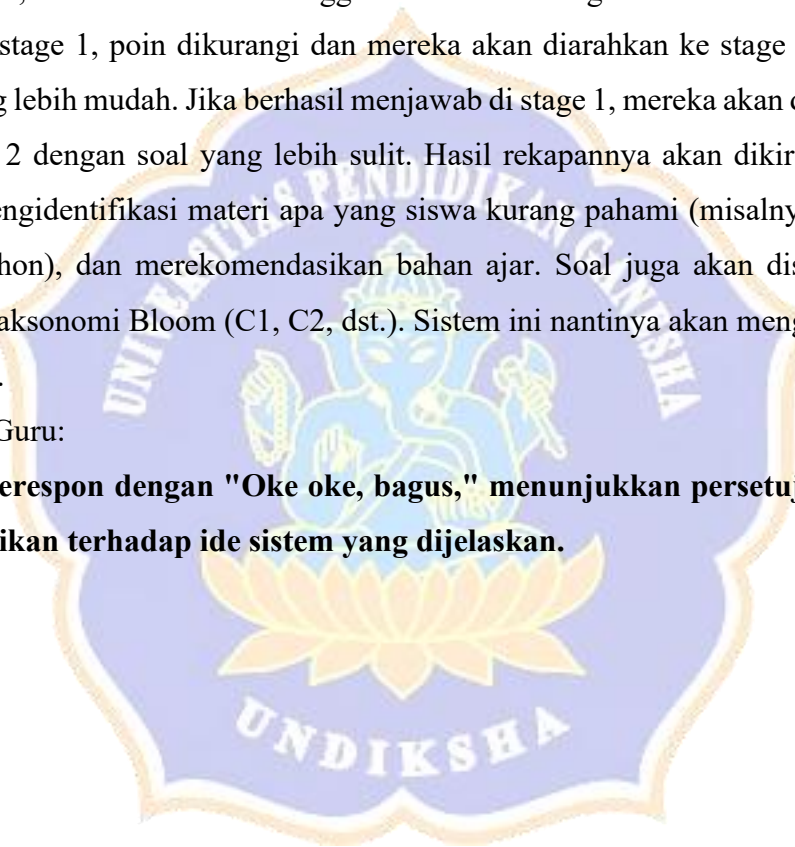
**jawaban dari big data, yang akan sangat membantu dalam proses pembelajaran. Namun, sejauh ini guru belum terlalu familiar dengan AI interaktif untuk pembelajaran.**

Soal 4: Penjelasan tentang Sistem AI Diagnostik yang Dikembangkan

Sistem AI diagnostik yang dikembangkan bertujuan untuk membantu guru. Guru akan mengunggah file berisi bahan ajar atau capaian ajar, lalu AI akan meng-generate soal otomatis yang akan ditampilkan kepada siswa. Siswa akan diminta menjawab, dan sistem akan menggunakan sistem stage. Jika siswa menjawab salah di stage 1, poin dikurangi dan mereka akan diarahkan ke stage 2 dengan soal yang lebih mudah. Jika berhasil menjawab di stage 1, mereka akan diarahkan ke stage 2 dengan soal yang lebih sulit. Hasil rekapannya akan dikirimkan ke guru, mengidentifikasi materi apa yang siswa kurang pahami (misalnya HTML atau Python), dan merekomendasikan bahan ajar. Soal juga akan disesuaikan dengan taksonomi Bloom (C1, C2, dst.). Sistem ini nantinya akan menggunakan database.

Respon Guru:

**Guru merespon dengan "Oke oke, bagus," menunjukkan persetujuan dan ketertarikan terhadap ide sistem yang dijelaskan.**



## Attachment 6 Teacher Needs Analysis Questionnaire

⋮

Bagaimana cara Anda menyusun soal evaluasi untuk siswa? \*

Menggunakan soal dari buku teks atau sumber lain

Menyusun sendiri berdasarkan kurikulum dan kebutuhan siswa

Menggunakan sistem digital atau software tertentu

Lainnya...

Seberapa sering Anda membuat soal yang berbeda untuk menyesuaikan dengan tingkat pemahaman siswa? \*

Selalu

Sering

Kadang-kadang

Tidak pernah

Apakah Anda mengalami kesulitan dalam menyusun soal yang sesuai dengan tingkat pemahaman siswa? \*

Ya

Tidak

Jika ya, jelaskan kendala yang sering Anda hadapi

Teks jawaban panjang

Bagaimana cara Anda mengidentifikasi siswa yang mengalami kesulitan memahami materi? \*

Dari nilai ujian atau tugas mereka

Dari observasi selama pembelajaran

Dari umpan balik siswa secara langsung

Lainnya...

Seberapa sering siswa Anda melupakan materi yang telah diajarkan sebelumnya? \*

- Sangat sering
- Sering
- Kadang-kadang
- Jarang

Menurut Anda, apa alasan utama siswa sering melupakan materi?

(Boleh memilih lebih dari satu)

- Kurangnya pemahaman konseptual
- Hanya menghafal tanpa memahami
- Kurangnya latihan dan pengulangan
- Materi tidak relevan dengan kehidupan sehari-hari
- Kurangnya minat terhadap pelajaran

Bagaimana cara Anda mengetahui materi apa saja yang paling sulit dipahami siswa?

- Dari nilai ujian atau tugas mereka
- Dari observasi selama pembelajaran
- Dari jawaban siswa dalam diskusi kelas
- Lainnya...

Saat siswa mengalami kesulitan memahami materi, bagaimana Anda biasanya mengatasinya?

- Memberikan pengulangan materi dengan cara yang berbeda
- Memberikan soal latihan tambahan
- Menganalisis kesalahan siswa dan memberikan penjelasan tambahan

Apa strategi yang Anda gunakan untuk membantu siswa mengingat kembali materi yang telah diajarkan? \*

- Mengulang materi dalam beberapa pertemuan berikutnya
- Memberikan latihan soal tambahan
- Menggunakan metode pembelajaran yang lebih interaktif (game, diskusi, dll.)
- Lainnya...

Apakah Anda merasa asesmen yang digunakan saat ini sudah cukup membantu dalam mengidentifikasi kesulitan siswa? \*

- Ya
- Tidak

Jika tidak, apa kendala yang Anda rasakan?

Teks jawaban panjang

...

Apakah sekolah Anda telah menggunakan sistem evaluasi berbasis digital atau AI? \*

- Ya, sudah digunakan secara rutin
- Ya, tetapi masih terbatas
- Tidak, tetapi ada rencana untuk menggunakannya
- Tidak pernah

Apakah Anda tertarik menggunakan sistem evaluasi berbasis AI yang dapat membuat soal otomatis dan memberikan umpan balik kepada siswa? \*

- Sangat tertarik
- Tertarik
- Ragu-ragu
- Tidak tertarik

...

Menurut Anda, bagaimana AI dapat membantu dalam evaluasi pembelajaran? \*

(Boleh memilih lebih dari satu)

- Membantu menyusun soal yang sesuai dengan tingkat pemahaman siswa
- Memberikan umpan balik otomatis kepada siswa dan guru
- Menghemat waktu dalam penyusunan evaluasi
- Menyesuaikan tingkat kesulitan soal secara otomatis
- Lainnya, sebutkan:

Jika ada sistem AI yang dapat membantu dalam evaluasi pembelajaran, fitur apa yang menurut Anda paling dibutuhkan? \*

- Pembuatan soal otomatis berdasarkan tingkat pemahaman siswa
- Umpan balik real-time untuk guru dan siswa
- Analisis kesulitan siswa dalam memahami materi
- Rekomendasi strategi pembelajaran tambahan
- Lainnya...

⋮

Jika terdapat sistem berbasis AI yang dapat memberikan laporan otomatis mengenai kesulitan siswa, bagaimana format laporan yang menurut Anda paling membantu? \*

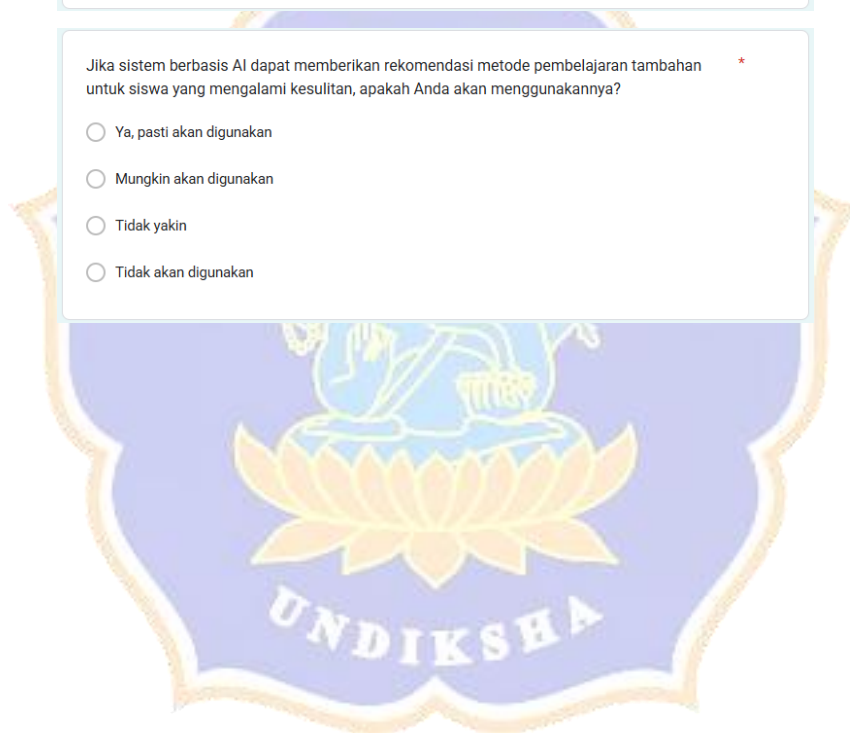
- Grafik tingkat pemahaman siswa
- Rekomendasi metode pembelajaran tambahan
- Kategori pemahaman siswa (misalnya: "sudah paham", "masih perlu latihan", "belum paham sama sekali")
- Lainnya...

Seberapa penting menurut Anda fitur umpan balik otomatis dari sistem evaluasi untuk membantu guru dalam memahami kesulitan siswa? \*

- Sangat penting
- Penting
- Cukup penting
- Tidak penting

Jika sistem berbasis AI dapat memberikan rekomendasi metode pembelajaran tambahan untuk siswa yang mengalami kesulitan, apakah Anda akan menggunakannya? \*

- Ya, pasti akan digunakan
- Mungkin akan digunakan
- Tidak yakin
- Tidak akan digunakan



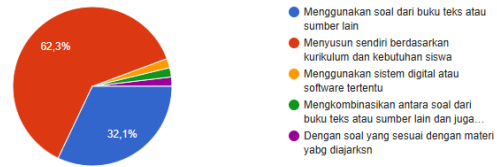
## Attachment 7 Teacher Needs Analysis Questionnaire Results

## Penyusunan dan Evaluasi Soal Pembelajaran

Bagaimana cara Anda menyusun soal evaluasi untuk siswa?

[Salin diagram](#)

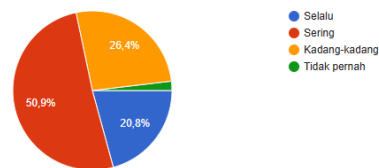
53 jawaban



Seberapa sering Anda membuat soal yang berbeda untuk menyesuaikan dengan tingkat pemahaman siswa?

[Salin diagram](#)

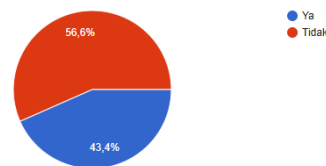
53 jawaban



Apakah Anda mengalami kesulitan dalam menyusun soal yang sesuai dengan tingkat pemahaman siswa?

[Salin diagram](#)

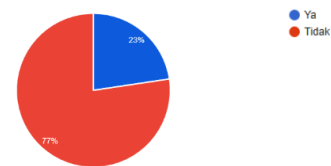
53 jawaban



Apakah Anda mengalami kesulitan dalam menyusun soal yang sesuai dengan tingkat pemahaman siswa?

[Salin diagram](#)

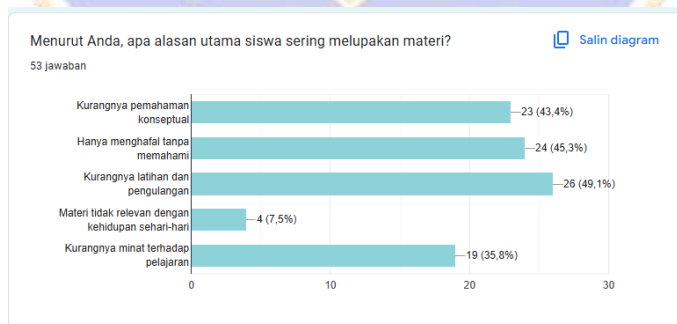
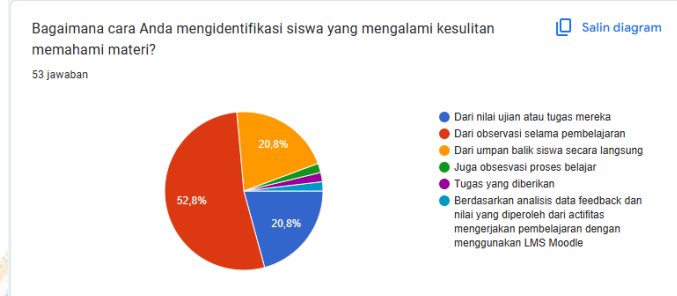
53 jawaban



Jika ya, jelaskan kendala yang sering Anda hadapi

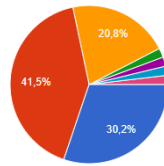
29 jawaban

- Tingkat pemahaman yg berbeda
- 0
- Aman
- Menentukan tingkat kesulitan soal
- Menurut guru soal yg dibuat dirasa sudah sesuai kemampuan siswa dan semua telah diajarkan, namun nilai kurang memuaskan
- Membuat soal yang bisa dipahami dengan banyaknya karakteristik siswa dalam memahami materi
- Disetiap kelas kemampuan siswa tidak sama
- Menyesuaikan tingkat pemahaman siswa
- Jumlah siswa yang banyak dan tingkat pemahannya yang heterogen



Bagaimana cara Anda mengetahui materi apa saja yang paling sulit dipahami siswa? [Salin diagram](#)

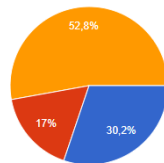
53 jawaban



- Dari nilai ujian atau tugas mereka
- Dari observasi selama pembelajaran
- Dari jawaban siswa dalam diskusi kelas
- Ini banyak sisi yg perlu di perhatikan
- lambahan buat di atas, kenapa siswa melupakan materi, karena tugas yang banyak dari guru2 lainnya
- membaca dan menulis Bali
- Hasil analisis data di LMS Moodle

Saat siswa mengalami kesulitan memahami materi, bagaimana Anda biasanya mengatasinya? [Salin diagram](#)

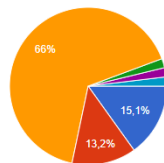
53 jawaban



- Memberikan pengulangan materi dengan cara yang berbeda
- Memberikan soal latihan Tambahan
- Menganalisis kesalahan siswa dan memberikan penjelasan lambahan

Apa strategi yang Anda gunakan untuk membantu siswa mengingat kembali materi yang telah diajarkan? [Salin diagram](#)

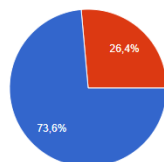
53 jawaban



- Mengulang materi dalam beberapa pertemuan berikutnya
- Memberikan latihan soal tambahan
- Menggunakan metode pembelajaran yang lebih interaktif (game, diskusi, dll.)
- Mengulang beberapa poin penting dan dikaitkan dengan materi berikutnya
- Dengsn mengetahui kelemahan siswa
- Memberikan pertanyaan yang sifatnya menggali maupun mengarahkan

Apakah Anda merasa asesmen yang digunakan saat ini sudah cukup membantu dalam mengidentifikasi kesulitan siswa? [Salin diagram](#)

53 jawaban



- Ya
- Tidak

Jika tidak, apa kendala yang Anda rasakan?

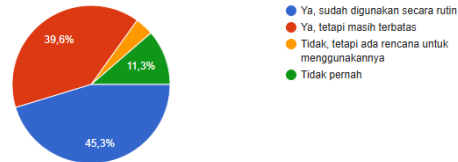
25 jawaban

- 
- 0
- Tentunya Trus berbenah
- Terbatasnya waktu
- Dipatok dengan sistem nilai siswa tidak boleh turun pada raport
- Mengkur secara komprehensif penguasaan siswa terhadap materi
- Belum mencerminkan hasil yang diharapkan
- Ya
- Tidak bisa melihat perbedaan kesulitan siswa

### Penggunaan Teknologi dalam Evaluasi Pembelajaran

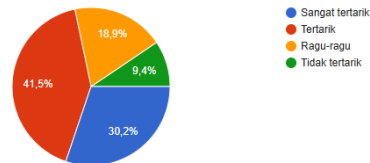
Apakah sekolah Anda telah menggunakan sistem evaluasi berbasis digital atau AI? [Salin diagram](#)

53 jawaban



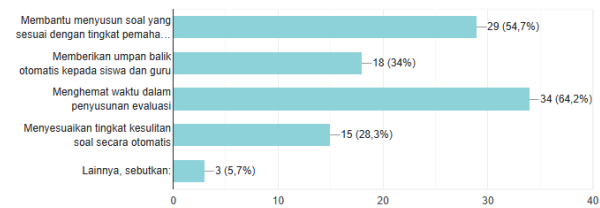
Apakah Anda tertarik menggunakan sistem evaluasi berbasis AI yang dapat membuat soal otomatis dan memberikan umpan balik kepada siswa? [Salin diagram](#)

53 jawaban



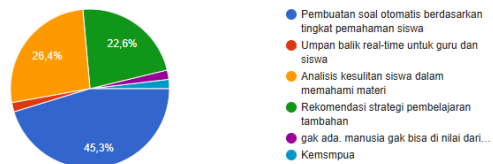
Menurut Anda, bagaimana AI dapat membantu dalam evaluasi pembelajaran? [Salin diagram](#)

53 jawaban



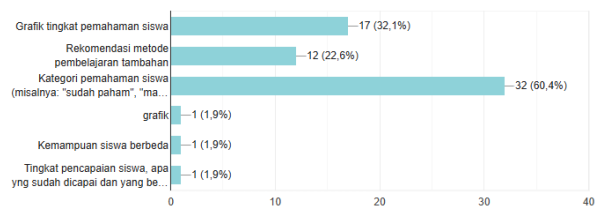
Jika ada sistem AI yang dapat membantu dalam evaluasi pembelajaran, fitur apa yang menurut Anda paling dibutuhkan? [Salin diagram](#)

53 jawaban



Jika terdapat sistem berbasis AI yang dapat memberikan laporan otomatis mengenai kesulitan siswa, bagaimana format laporan yang menurut Anda paling membantu? [Salin diagram](#)

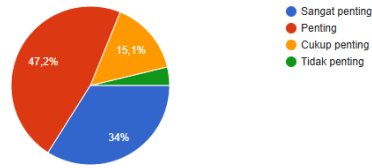
53 jawaban



Seberapa penting menurut Anda fitur umpan balik otomatis dari sistem evaluasi untuk membantu guru dalam memahami kesulitan siswa?

[Salin diagram](#)

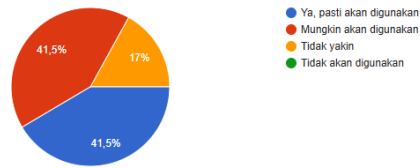
53 jawaban



Jika sistem berbasis AI dapat memberikan rekomendasi metode pembelajaran tambahan untuk siswa yang mengalami kesulitan, apakah Anda akan menggunakannya?

[Salin diagram](#)

53 jawaban



## Attachment 8 Student Needs Analysis Questionnaire

Dalam mata pelajaran informatika di sekolah, materi yang menurut kamu paling sulit dipahami apa? \*

Teks jawaban panjang

Seberapa sulit kamu memahami materi tersebut? \*

- Sangat Mudah
- Mudah
- Cukup Sulit
- Sulit
- Sangat Sulit

Apakah kamu merasa kesulitan ketika mengerjakan soal atau proyek yang berkaitan dengan materi tersebut? \*

- Ya, sangat sulit.
- Cukup sulit.
- Kadang-kadang sulit.
- Tidak sulit.

Apa yang membuat kamu kesulitan memahami materi tersebut? (Boleh pilih lebih dari satu) \*

- Penjelasan guru kurang jelas.
- Materi terlalu rumit.
- Terlalu banyak teori yang harus dipahami.
- Kurangnya latihan soal atau contoh yang diberikan.
- Terlalu cepat pindah ke materi baru.
- Kurangnya media pembelajaran seperti video atau praktik langsung.
- Lainnya...

Apakah kamu sering lupa materi yang sudah diajarkan ketika mengerjakan tugas atau ujian? \*

- Sangat Sering
- Cukup Sering
- Kadang-kadang
- Jarang
- Tidak Pernah

Apakah kamu kesulitan menggabungkan berbagai materi yang berbeda untuk mengerjakan tugas atau proyek? \*

- Sangat Sulit
- Cukup Sulit
- Kadang-kadang Sulit
- Tidak Sulit

...

Tuliskan pendapat atau saran kamu agar pelajaran Informatika lebih mudah dipahami. \*

Teks jawaban panjang

## Attachment 9 Results of Student Needs Analysis

Dalam mata pelajaran informatika di sekolah, materi yang menurut kamu paling sulit dipahami apa?  
120 jawaban

flowchart

Excel

coding

pemrograman

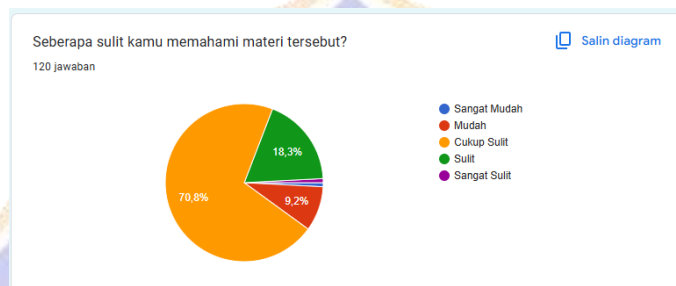
C++

Flowchart

Algoritma

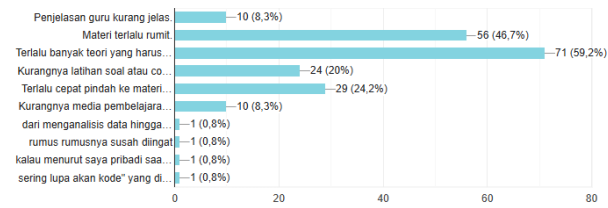
Materi pemrograman bahasa C++

c++



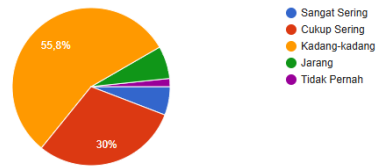
Apa yang membuat kamu kesulitan memahami materi tersebut? (Boleh pilih lebih dari satu) [Salin diagram](#)

120 jawaban



Apakah kamu sering lupa materi yang sudah diajarkan ketika mengerjakan tugas atau ujian? [Salin diagram](#)

120 jawaban



Tuliskan pendapat atau saran kamu agar pelajaran Informatika lebih mudah dipahami.

120 jawaban

Kasih materi yang agak gampang dikit

cara menjelaskan lebih detail dan pelan-pelan agar siswa dapat paham

saran saya agar lebih diperjelas mengenai materi" yang diberikan sehingga mudah untuk mengerjakan tugas yang diberikan

Menurut saya agar pelajaran informatika lebih mudah untuk dipahami ialah guru harus menerapkan metode pelajaran kinetik dan ketika menampilkan materi lewat proyektor usahakan menampilkan gambar yang cukup menarik, dan yang paling penting untuk pemberian soal atau tugas terkait materi yang sudah dijelaskan

dengan cara menambah media yang singkat dan dapat di pahami

semoga dalam membuat coding lebih seru dan menarik

tidak ada

UNDIKSHA



## Attachment 11 Black Box Test Instruments

Nama Lengkap :  
 Institusi :  
 Bidang Keahlian :  
 Jabatan :

Untuk setiap modul, jawab pertanyaan "Relevan" jika hasil yang diharapkan sesuai dengan keluaran sistem saat diuji. Berikan "Tidak Relevan" jika tidak sesuai, jika pertanyaan tidak berlaku. Cantumkan catatan atau detail hasil observasi pada kolom "Catatan/Hasil Observasi".

No	Modul	Pertanyaan Uji	Catatan/ Hasil Observasi	Relevan (✓)	Tidak Relevan (x)
1	Autentikasi Pengguna	Apakah sistem berhasil memberikan akses dan mengarahkan pengguna ke dashboard yang sesuai (Guru/Siswa) ketika nama pengguna dan kata sandi valid dimasukkan?			
2	Autentikasi Pengguna	Apakah sistem menolak akses dan menampilkan pesan galat yang jelas ketika nama pengguna atau kata sandi yang tidak valid dimasukkan?			
3	Registrasi Pengguna	Apakah akun baru berhasil dibuat dan pengguna diarahkan kembali ke halaman masuk setelah mengisi data pendaftaran yang valid?			
4	Registrasi Pengguna	Apakah sistem menolak proses pendaftaran jika email sudah terdaftar?			
5	Unggah Modul Ajar	Apakah sistem berhasil menyimpan berkas, mengekstrak teks, dan menginisiasi proses AI setelah guru mengunggah berkas valid (PDF/DOCX)?			

No	Modul	Pertanyaan Uji	Catatan/ Hasil Observasi	Relevan (✓)	Tidak Relevan (x)
6	Unggah Modul Ajar	Apakah sistem menolak berkas dan menampilkan pesan galat jika format tidak didukung?			
7	Generasi Soal Diagnostik (AI)	Apakah AI berhasil menghasilkan soal pilihan ganda sesuai distribusi level dan kesulitan (MST) yang disimpan rapi ke dalam database?			
8	Pembuatan Koleksi Soal	Apakah koleksi soal baru berhasil dibuat dan muncul di daftar koleksi guru?			
9	Penambahan Soal ke Koleksi	Apakah soal dari bank soal berhasil ditambahkan ke koleksi tanpa duplikasi?			
10	Validasi Soal	Apakah status soal berubah menjadi "Tervalidasi" di database saat guru menyetujuinya?			
11	Revisi Soal	Apakah sistem menyimpan versi baru soal ketika guru melakukan penyuntingan?			
12	Penambahan Siswa ke Koleksi	Apakah siswa berhasil ditambahkan ke koleksi tes tertentu?			
13	Penghapusan Siswa dari Koleksi	Apakah akses siswa berhasil dicabut dari koleksi tes?			
14	Penghapusan Koleksi Soal	Apakah koleksi berhasil dihapus beserta relasinya tanpa menghapus data			

No	Modul	Pertanyaan Uji	Catatan/ Hasil Observasi	Relevan (✓)	Tidak Relevan (x)
		master soal?			
15	Pemilihan Asesmen (Siswa)	Apakah dasbor siswa menampilkan daftar tes yang ditugaskan dan tombol "Kerjakan"?			
16	Pengerjaan Tes (MST Adaptif)	Apakah sistem menyajikan soal satu per satu dan menyesuaikan tingkat kesulitan (routing) berdasarkan jawaban siswa (Benar/Salah)?			
17	Hasil & Umpan Balik AI (Siswa)	Apakah sistem menampilkan skor akhir dan paragraf umpan balik personal yang di-generate oleh AI setelah tes selesai?			
18	Ekspor Hasil (Siswa)	Apakah berkas PDF laporan hasil pribadi berhasil diunduh siswa?			
19	Analisis Hasil (Guru)	Apakah sistem menampilkan grafik statistik distribusi level dan akurasi siswa pada halaman detail koleksi?			
20	Ekspor Laporan (Guru)	Apakah berkas Excel berisi rekapitulasi nilai seluruh siswa berhasil diunduh oleh guru?			

Mengetahui,

## Attachment 12 White Box Test Instruments

Nama Lengkap :  
 Institusi :  
 Bidang Keahlian :  
 Jabatan :

Untuk setiap modul, jawab pertanyaan "Relevan" jika hasil yang diharapkan sesuai dengan keluaran sistem saat diuji. Berikan "Tidak Relevan" jika tidak sesuai, jika pertanyaan tidak berlaku. Cantumkan catatan atau detail hasil observasi pada kolom "Catatan/Hasil Observasi".

No	Modul	Pertanyaan Uji	Catatan/Hasil Observasi	Relevan (✓)	Tidak Relevan (✓)
1	Autentikasi	Apakah proses login berhasil untuk pengguna valid dan gagal untuk kredensial salah secara konsisten?			
2	Registrasi	Apakah sistem mencegah data tidak valid/duplikat dan menyimpan data registrasi dengan benar?			
3	Unggah Modul Ajar	Apakah berkas dengan format valid dapat diunggah dan format tidak valid ditolak dengan penanganan galat tepat?			
4	Evaluasi Jawaban	Apakah jawaban siswa dievaluasi dengan benar, termasuk untuk jawaban mirip atau salah ketik ringan?			
5	Interaksi Basis Data	Apakah data hasil jawaban siswa konsisten dan tetap terjaga saat akses bersamaan (konkurensi)?			
6	Ekstraksi Komponen Modul Ajar	Apakah sistem dapat mengekstrak komponen pembelajaran penting dari dokumen dengan akurat?			
7	Generasi Soal AI	Apakah sistem mampu menghasilkan soal sesuai tingkat kesulitan yang ditentukan?			
8	Relevansi Konten	Apakah soal yang dihasilkan relevan dengan topik materi yang diinput?			
9	Kesesuaian Tujuan	Apakah soal mengukur langsung tujuan pembelajaran yang			

No	Modul	Pertanyaan Uji	Catatan/Hasil Observasi	Relevan (✓)	Tidak Relevan (✓)
		ditetapkan?			
10	Level Kognitif	Apakah soal sesuai dengan level kognitif (Bloom) yang ditargetkan?			
11	Kejelasan Bahasa	Apakah bahasa soal dan opsi jawaban jelas serta tidak ambigu?			
12	Kualitas Pengecoh	Apakah opsi jawaban salah (pengecoh) masuk akal dan efektif untuk menguji pemahaman siswa?			
13	Akurasi Konten	Apakah soal dan kunci jawaban faktual serta tidak menyesatkan?			
14	Bebas Bias	Apakah soal bebas dari bias (gender, suku, budaya) dan relevan untuk siswa?			

Mengetahui,



## Attachment 13 Content Expert Test Instrument

Nama Lengkap :  
 Instansi/ Sekolah :  
 Mata Pelajaran yang Diampu :  
 Jabatan :

No	Pernyataan	Relevan <input checked="" type="checkbox"/>	Tidak Relevan <input type="checkbox"/>
1	Materi dalam soal sesuai dengan indikator capaian pembelajaran pada modul ajar.		
2	Butir soal secara jelas mengarah pada pencapaian tujuan pembelajaran yang ditargetkan.		
3	Isi materi dalam soal dan kunci jawaban yang disediakan sudah akurat secara akademik.		
4	Konteks/topik yang digunakan dalam soal relevan dengan materi ajar Informatika.		
5	Tingkat kesulitan dan level kognitif soal sudah sesuai untuk level siswa (Kelas X).		
6	Opsi jawaban (pengecoh) yang disediakan masuk akal dan efektif untuk menguji pemahaman.		
7	Bahasa yang digunakan dalam soal jelas, tidak ambigu, dan mudah dipahami oleh siswa.		
8	Stimulus yang digunakan (jika ada, seperti teks atau gambar) relevan dan efektif untuk mendukung pertanyaan.		
9	Soal bebas dari bias (misalnya gender, suku, budaya) dan tidak mengarah pada jawaban tertentu.		
10	Butir soal memiliki potensi untuk mendeteksi miskonsepsi atau kesulitan belajar spesifik siswa.		

Mengetahui,

## Attachment 14 Student Questionnaire Data Results

11/14/2025 13:05:49	oktalestari251@gmail.com	ni Kadek Oktalestari	x.i	7	6	7	7	7	7	6	7
11/14/2025 13:05:54	dava2619@gmail.com	Anak Agung Gede Devadatta Aparajita	X.C	7	7	7	7	7	7	7	7
11/14/2025 13:05:54	agusabisena@gmail.com	I KOMA NG AGUS ABISENA WEDANTA	X.C	7	7	7	7	7	7	7	7
11/14/2025 13:06:09	rkode264@gmail.com	Dewa Putra Dharmayasa	X.C	7	7	7	7	7	7	7	7
11/14/2025 13:06:47	dedyindrprasetya@gmail.com	Ketut Dedy Indra prasetya	X.c	7	7	7	7	7	7	7	5
11/14/2025 13:07:57	givanoputra5244@gmail.com	I Kadek Givano Gunawan Putra	X.J	7	7	7	7	7	7	7	7
11/14/2025 13:09:12	ayu393727@gmail.com	Ni Komang Ayuk Trisna Dewi	X.C	4	3	5	4	6	6	6	7
11/14/2025 13:09:57	ayu393727@gmail.com	Ni Komang Ayuk Trisna Dewi	X.C	7	7	7	7	7	7	7	7
11/14/2025 13:10:19	krisnawiguna495@gmail.com	I Gusti Ngurah Adryan Krisna Wiguna	X.J	7	7	7	7	7	7	7	7
11/14/2025 13:10:26	rayndraananta@gmail.com	Komang rayndra mahananta putra	X.i	4	6	6	7	6	6	7	6

11/14/2025 13:11:44	k55027551@gmail.com	I Kadek Dwi Putra Anugrah	X.I	7	6	5	6	7	6	7	7
11/14/2025 13:13:10	sukarmayasagede@gmail.com	I Putu Gede Sukarama Yasa	X.i	7	7	7	7	7	7	6	5
11/14/2025 13:13:41	divyahariwati@gmail.com	Ni Wayan Divya Hariwati	X.J	7	7	7	6	7	7	5	5
11/14/2025 13:13:43	iwayannovawiguna120@gmail.com	I Wayan Nova Wiguna	17	7	5	7	6	7	7	6	7
11/14/2025 13:13:56	kwindu66@gmail.com	I Kadek Windu Yana Putra	X.i	6	7	6	7	7	7	7	7
11/14/2025 13:17:55	bintang88maharani@gmail.com	Komang Bintang Ayu Maharani	X.J	6	6	6	5	6	6	6	6
11/14/2025 13:19:21	excellentpasaribu017@gmail.com	Excellent Pasaribu	X.j	6	6	6	6	6	6	6	6
11/14/2025 13:22:25	kadekalita78@gmail.com	Ni Kadek Alita Dwi Ani	X.i	7	7	7	5	7	7	7	7
11/14/2025 13:22:45	kybhasmaputra@gmail.com	I Wayan Bhasma Putra Kardana	X.I	6	7	7	7	6	7	7	7
11/14/2025 13:27:05	wsatya767@gmail.com	wayan satya dharma putra	10C	7	7	7	7	6	5	7	7
11/14/2025 13:37:41	aalmiraa29@gmail.com	Putu Almira Priscillia Ananda Wijaya	X.i	7	7	3	3	7	7	7	7
11/14/2025 13:38:00	dewayogalesmana@gmail.com	Ni Kadek Febri Anggreni	X.J	4	5	3	3	3	4	5	4

11/14/2025 13:41:47	cahayaputri266245@gmail.com	Pande komang cahaya putri	X.i	6	5	3	5	5	6	6	7
11/14/2025 13:43:03	marcelpradifta@gmail.com	Gede Willy Marcelino Pradifta	X.I	6	7	6	7	6	5	5	5
11/14/2025 13:45:21	ngurahyoga280@gmail.com	I Gusti Ngurah Yoga Pratista	X.i	5	6	4	5	5	6	5	5
11/14/2025 13:45:54	komangcahyapuspadewi@gmail.com	Ni Komang Cahya Puspade wi	X.J	7	6	7	7	7	7	7	7
11/14/2025 13:46:50	dewayogalesmana@gmail.com	Ni Kadek Febri Anggreni	X.J	7	7	7	7	7	7	7	7
11/14/2025 13:55:21	josesitumorang25@gmail.com	Jose Rizal amri Situmorang	xi	7	7	6	6	7	7	7	7
11/14/2025 13:55:23	ayusaras2112@gmail.com	Gusti Ayu Sarasmin arti	X.i	7	6	7	6	6	7	6	7
11/14/2025 13:55:42	liaantari62@gmail.com	Kadek Agus Satriya Jaya	X.i	7	7	7	7	7	7	1	1
11/14/2025 13:56:03	komangwidya23@gmail.com	Ni Komang Widya Trisna Sari	X.I	5	6	6	7	6	7	6	7
11/14/2025 13:56:14	sofianaluna2@gmail.com	Ni Ketut Luna Sofiana Putri	X.I	7	6	7	6	7	7	7	7
11/14/2025 13:56:25	dewadimadimas@gmail.com	Dewa Gede Dimas Prastya	X.i	6	6	6	7	7	7	7	7
11/14/2025 13:57:	dewaayuari944@gmail.com	Dewa Ayu Ari Dwijaya	X.i	7	7	5	7	4	6	7	7



56		bagaskar a										
11/14/ 2025 16:25: 57	adityapermana5566 @gmail.com	i Kadek Aditya Permana Putra	X .C	7	7	7	7	7	7	7	7	7
11/14/ 2025 16:26: 31	dayuagung529@gma il.com	Ida Ayu Agung Dinda Cahyani	X C	6	7	6	7	6	5	6	6	
11/14/ 2025 16:31: 25	wanowa58@gmail.co m	Komang Nicki Anugrah	x. C	6	6	6	7	6	7	6	6	
11/14/ 2025 17:15: 10	naylachan44@gmail. com	Desak putu carma cinta belva	X .C	6	6	5	4	5	5	5	4	
11/14/ 2025 17:40: 44	ekaseptiari263@gma il.com	Ni Putu Eka Septiari Santi	X .C	7	7	7	7	7	7	7	7	
11/14/ 2025 18:38: 05	ogekis668@gmail.co m	I Gusti Ayu Istri Mahasw ari Ningrat	X C	7	7	7	7	7	7	7	7	
11/14/ 2025 18:40: 56	pratiwierlinn1809@g mail.com	Luh Ade Erlinda Pratiwi	X .C	5	6	5	6	5	6	6	6	
11/14/ 2025 18:41: 53	pratiwierlinn1809@g mail.com	luh ade erlinda pratiwi	X .C	7	7	7	7	7	7	4	2	
11/14/ 2025 18:45: 37	putuhannasekarprati wi@gmail.com	Putu Hanna Sekar Pratiwi	X C	7	7	7	7	7	7	7	6	
11/14/ 2025 18:47: 12	desakcintya123@gm ail.com	Desak Made Cintya Dewi	X C	7	7	7	7	7	6	7	6	
11/14/ 2025 18:55: 20	cpratiwiwijaya@gma il.com	Cahyani Pratiwi Wijaya	X C	7	7	7	7	7	6	5	7	
11/14/ 2025 18:55: 22	putuarua10@gmail.c om	I PUTU ARYA EKA PUTRA	X .C	4	4	5	6	4	4	3	4	

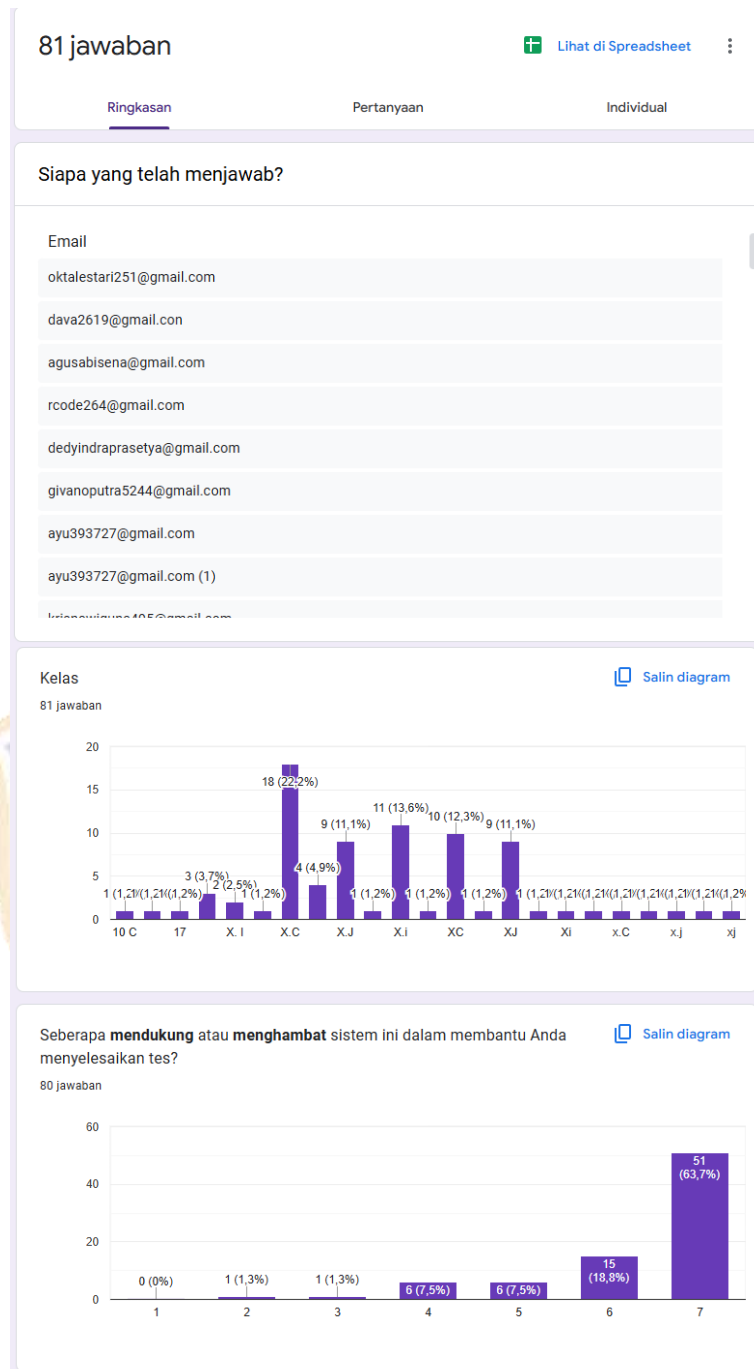


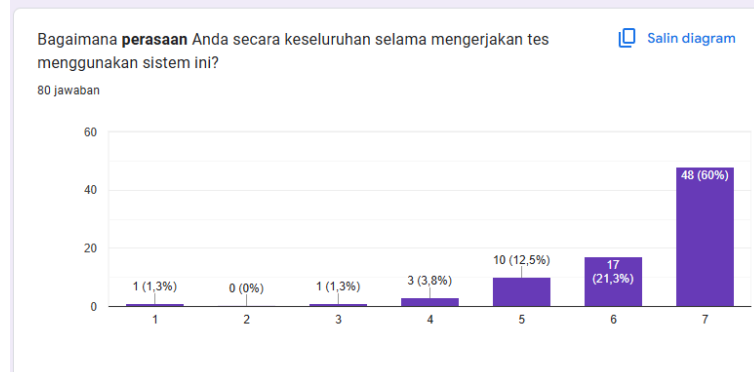
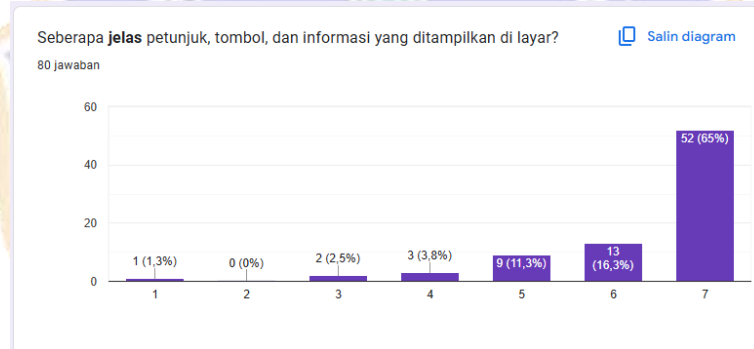
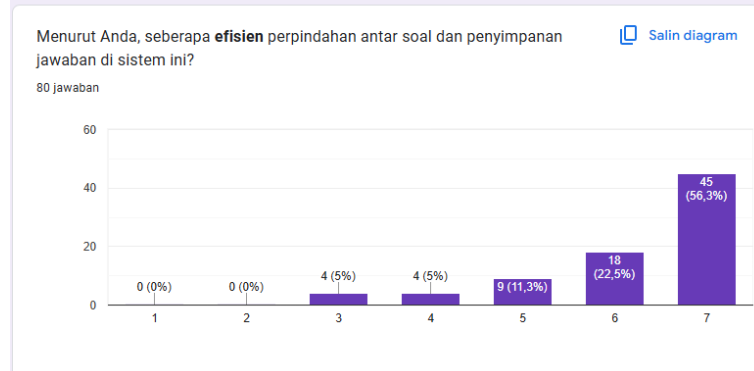
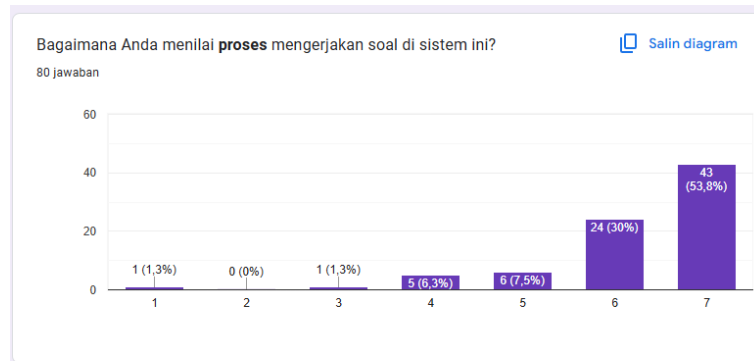
11/14/2025 21:04:28	kadededek93@gmail.com	I Kadek Dedek Setiawan	X _J	6	6	5	6	6	7	7	7
11/14/2025 21:14:37	kanyaputri7788@gmail.com	Putu kanya putri susila	X J	7	7	6	5	5	7	5	4
11/14/2025 21:28:09	iketutesakrisnatasugara@gmail.com	iketut esa krisnata sugara	X J	5	5	5	5	5	6	4	5
11/14/2025 21:58:50	komangasri381@gmail.com	Ni Komang Asri Septiani	X .C	7	7	7	7	7	7	7	7
11/14/2025 22:29:53	sutaidabagusnarendra@gmail.com	IDA BAGUS MADE NARENDRA NAROTAMA SUTAKENITEN	X I	5	5	4	5	5	5	5	5
11/14/2025 23:25:01	septisepta240903@gmail.com	putu ayurda pratistya	x. j	7	7	7	7	7	7	7	7
11/15/2025 4:30:30	iputukrishnawijayaputra@gmail.com	I Putu Krisna Wijaya Putra	X .J	6	7	6	6	6	6	6	6
11/15/2025 7:24:55	gungdengurah75@gmail.com	Anak Agung Gede Dwika Bagaskara	X .J	6	6	6	7	7	7	7	7
11/15/2025 10:08:50	anyakusuma@gmail.com	Ni Nyoman Pradnya Kusuma Dewi	X .J	7	6	7	6	7	6	5	7
11/15/2025 10:14:43	gkrisna979@gmail.com	I gede ananda krisna djelantik	xj	3	4	4	7	6	6	5	5
11/16/2025 10:58:	dindakesya75@gmail.com	Ni Putu Dibda Kesya	X .J	7	6	7	7	6	6	7	7

30		Putri										
----	--	-------	--	--	--	--	--	--	--	--	--	--



Attachment 15 Short UEQ Student Results

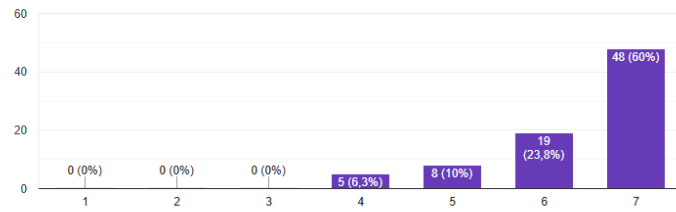




Bagaimana Anda menilai **tampilan visual** (desain, warna, huruf) dari sistem ini?

[Salin diagram](#)

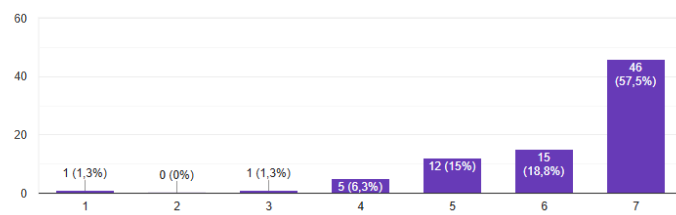
80 jawaban



Seberapa **kreatif** ide atau fitur yang ada di dalam sistem tes ini?

[Salin diagram](#)

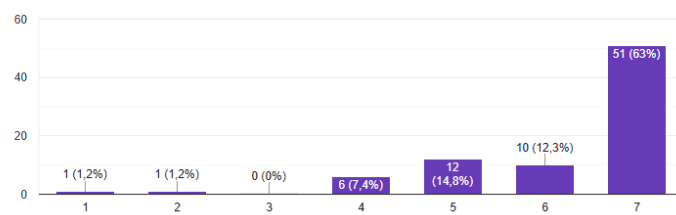
80 jawaban



Secara tampilan, apakah sistem ini terasa **modern** atau **umum** (biasa saja)?

[Salin diagram](#)

81 jawaban



## Attachment 16 Prompting Code (Prompt Engineering)

```
def prompt_pembuatan_soal(module_components):

    def format_section(title, key, default="[Tidak tersedia]"):
        content = module_components.get(key, "")
        if content:
            if isinstance(content, list):
                joined = "\n".join(
                    [str(item).strip() for item in content if str(item).strip()]
                )
                if len(joined) > 10:
                    return f"{title}:\n{joined}"
            elif isinstance(content, dict):
                joined = "\n".join([f"{k}: {v}" for k, v in content.items()])
                if len(joined.strip()) > 10:
                    return f"{title}:\n{joined.strip()}"
            else:
                text = str(content).strip()
                if len(text) > 10:
                    return f"{title}:\n{text}"
        return f"{title}:\n{default}"

    mata_pelajaran = module_components.get("mata_pelajaran", "Informatika")

    prompt = f"""Anda adalah GURU {mata_pelajaran} (SMA) yang ahli dalam
    membuat soal Asesmen Diagnostik.
    Tugas: Menyusun 55 Soal Pilihan Ganda untuk MEMETAKAN TINGKAT
    KEMAMPUAN SISWA.
```

**TUJUAN ASESMEN DIAGNOSTIK:**

Asesmen ini bertujuan untuk: (1) mengetahui pemahaman konsep dasar siswa, (2) mendeteksi miskonsepsi umum siswa pemula, dan (3) mengukur kesiapan siswa untuk materi lanjutan.

**GAYA SOAL (WAJIB): FAKTUAL–KONSEPTUAL**

- Ubah soal reflektif/opini menjadi soal yang memiliki jawaban benar yang objektif.
- DILARANG membuat pertanyaan berbentuk perbandingan nilai seperti: “Mana yang lebih penting...”, “Mana yang lebih krusial...”, “Mana yang lebih fundamental...”.
- Hindari juga pertanyaan opini/refleksi seperti: “Menurut kamu...”, “Bagaimana pendapatmu...”, atau “Mengapa ... penting...” jika jawabannya bergantung pada pendapat.
- Gunakan bentuk tanya yang objektif, misalnya:
  - “Apa yang dimaksud dengan ...”
  - “Apa fungsi ...” / “Apa tujuan ...”
  - “Manakah contoh dari ...” / “Manakah yang termasuk ...”
  - “Urutan langkah yang benar adalah ...”

- “Pernyataan manakah yang SALAH/BENAR tentang ...”

#### BATASAN SCOPE SOAL (WAJIB DIPATUHI):

Semua soal WAJIB bersumber dari materi yang EKSPLISIT ada pada MODUL AJAR di bawah, terutama bagian TUJUAN PEMBELAJARAN, TOPIK PEMBELAJARAN KONTEKSTUAL, LANGKAH-LANGKAH PEMBELAJARAN, dan CAPAIAN PEMBELAJARAN. Anda hanya boleh memakai istilah, konsep, tokoh, isu, aktivitas, konteks, dan contoh yang muncul di modul.

JANGAN memperluas ke topik baru yang tidak dibahas. Jika modul tidak membahas suatu domain/aktivitas (misalnya menulis kode, menghitung rumus tertentu, membahas teori lanjutan, atau istilah teknis spesifik), maka domain tersebut OUT OF SCOPE dan DILARANG menjadi bahan soal.

Kesulitan ditentukan dari DEPTH (kedalaman penalaran pada materi yang sama), bukan BREADTH (menambah topik baru). Soal HARD boleh lebih menantang, tetapi tetap pada scope modul.

Validasi internal sebelum menulis JSON: untuk tiap soal, tanyakan “Apakah konsep ini ada di modul?” Jika ragu, JANGAN gunakan.

---



---

#### ANALISIS MODUL AJAR:

---



---

```
{format_section("1. MATA PELAJARAN", "mata_pelajaran")}  
  
{format_section("2. IDENTIFIKASI KESIAPAN PESERTA DIDIK",  
"identifikasi_kesiapan")}  
  
{format_section("3. KARAKTERISTIK MATERI PELAJARAN",  
"karakteristik_materi")}  
  
{format_section("4. DIMENSI PROFIL LULUSAN", "dimensi_profil_lulusan")}  
  
{format_section("5. CAPAIAN PEMBELAJARAN", "capaian_pembelajaran")}  
  
{format_section("6. LINTAS DISIPLIN ILMU", "lintas_disiplin")}  
  
{format_section("7. TUJUAN PEMBELAJARAN", "tujuan_pembelajaran")}  
  
{format_section("8. TOPIK PEMBELAJARAN KONTEKSTUAL",  
"topik_pembelajaran")}  
  
{format_section("9. LANGKAH-LANGKAH PEMBELAJARAN",  
"langkah_pembelajaran")}
```

{format\_section("10. ASESMEN PEMBELAJARAN",  
"asesmen\_pembelajaran")}

---

#### PANDUAN TAKSONOMI TEKNOLOGI (5 LEVEL MST):

[!] CATATAN: Semua level HARUS menggunakan MATERI dari MODUL AJAR di atas!

##### 1. LEVEL 1 - AWARENESS (Kesadaran)

- Fokus: Following the development (mengikuti perkembangan), Care for existence (peduli keberadaan), dan Willingness to observe the characteristics (kesediaan mengamati karakteristik).
- Aspek: Mengingat fakta/istilah dasar DARI MODUL dengan fokus pada kesadaran terhadap perkembangan teknologi, kepedulian akan keberadaan teknologi, dan kemauan untuk mengamati karakteristiknya.
- Contoh: "Apa yang dimaksud dengan [konsep di modul]?" atau "Manakah contoh dari [konsep di modul]?" atau "Apa karakteristik utama [konsep/teknologi di modul]?"

##### 2. LEVEL 2 - LITERACY (Literasi)

- Fokus: Comprehensive understanding (pemahaman komprehensif), Depth in learning (kedalaman dalam belajar), dan Knowledge of the functions and benefits (pengetahuan tentang fungsi dan manfaat).
- Aspek: Memahami konsep dan hubungan YANG DIAJARKAN di modul dengan fokus pada pemahaman menyeluruh, kedalaman pembelajaran, serta pengetahuan tentang fungsi dan manfaat teknologi.
- Contoh: "Apa fungsi [konsep/fitur di modul]?" atau "Apa tujuan [aktivitas/proses di modul]?" atau "Manakah pernyataan yang BENAR tentang [konsep di modul]?"

##### 3. LEVEL 3 - CAPABILITY (Kapabilitas)

- Fokus: Technical ability (kemampuan teknis) dan Accessibility capabilities (kemampuan aksesibilitas).
- Aspek: Menerapkan prosedur/rumus YANG DIAJARKAN di modul dengan fokus pada kemampuan teknis dalam menggunakan teknologi dan aksesibilitas fitur/teknologi tersebut.
- Contoh: "Bagaimana langkah menggunakan/mengakses [alat/fitur/layanan yang disebut modul] untuk mencapai [tujuan aktivitas di modul]?" atau "Urutan tindakan yang benar untuk [aktivitas di modul] adalah..."

##### 4. LEVEL 4 - CREATIVITY (Kreativitas)

- Fokus: Discovery ability (kemampuan menemukan) dan Usefulness of discovery (kegunaan penemuan).
- Aspek: Analisis konseptual sederhana dalam SCOPE modul (bukan debugging program kompleks) dengan fokus pada kemampuan menemukan konsep/langkah yang keliru dan memilih perbaikan yang tepat.
- Contoh: "Manakah konsep yang keliru pada pernyataan berikut tentang [konsep

di modul]?" atau "Jika urutan langkah [aktivitas di modul] salah, langkah mana yang harus diperbaiki?" atau "Manakah langkah yang TIDAK sesuai prosedur [di modul]?"

#### 5. LEVEL 5 - CRITICISM (Evaluasi)

- Fokus: Critical in operation (kritis dalam operasi) dan Critical in choosing technology (kritis dalam memilih teknologi).

- Aspek: Mengevaluasi pilihan/ Pernyataan secara objektif berdasarkan KRITERIA yang disebutkan di modul (tujuan, syarat, karakteristik, manfaat, batasan), tanpa pertanyaan reflektif/perbandingan nilai.

- Contoh: "Berdasarkan kriteria [yang disebut di modul], manakah pilihan yang PALING sesuai?" atau "Apa tujuan utama dari [pendekatan/metode di modul]?" atau "Manakah alasan yang SESUAI untuk memilih [opsi di modul] dalam kondisi sederhana yang dinyatakan di soal?"

#### KALIBRASI KESULITAN (TETAP DALAM SCOPE):

Kesulitan soal mengukur KEDALAMAN PENALARAN pada materi yang SAMA — BUKAN menambah topik baru.

Tiga label yang diizinkan: Easy | Medium | Hard

- EASY (Mudah):

- Mengingat/mengenali: definisi, fakta, istilah, atau karakteristik dasar yang TERSURAT di modul.

- Cukup satu langkah penalaran langsung dari teks modul.

- Contoh tipe soal: "Apa yang dimaksud dengan X?", "Manakah contoh dari Y?", "Apa tujuan Z?"

- Distraktor: salah definisi yang umum terjadi pada pemula.

- MEDIUM (Sedang):

- Memahami/menerapkan: hubungan antar konsep, fungsi, sebab-akibat, atau prosedur yang DIAJARKAN di modul.

- Memerlukan 1–2 langkah penalaran berdasarkan konsep di modul.

- Contoh tipe soal: "Apa fungsi A dalam konteks B?", "Manakah pernyataan yang BENAR tentang X?", "Urutan langkah yang benar adalah ..."

- Distraktor: salah fungsi atau salah urutan langkah yang masuk akal.

- HARD (Sulit):

- Menganalisis/mengevaluasi: identifikasi miskonsepsi, memilih koreksi/solusi terbaik, atau mengevaluasi pernyataan berdasarkan KRITERIA yang ada di modul.

- Memerlukan penalaran  $\geq 2$  langkah; siswa harus membedakan konsep yang mirip atau mengevaluasi pilihan.

- Contoh tipe soal: "Manakah pernyataan yang KELIRU tentang X?", "Jika langkah Y salah, perbaikan yang tepat adalah...", "Berdasarkan kriteria Z, manakah pilihan yang paling sesuai?"

- Distraktor: miskonsepsi yang sering terjadi atau pilihan yang terlihat benar namun tidak sesuai kriteria modul.

CATATAN: Kesulitan ditentukan dari DEPTH (kedalaman penalaran pada materi

yang sama), BUKAN BREADTH (menambah topik baru di luar modul).

#### ATURAN REDAKSI / DIKSI (WAJIB):

- Hindari kata "Pertimbangkan" (termasuk variasi seperti "pertimbangkanlah").
- Gunakan "Perhatikan" sebagai pengganti.
- Gunakan frasa instruksi yang tegas, objektif, singkat, dan mudah dipahami siswa.
- Jika butuh pembuka instruksi, gunakan pola kalimat di bawah (pilih yang paling sesuai).

#### POLA INSTRUKSI YANG DISARANKAN (CONTOH):

- "Perhatikan pernyataan berikut. Manakah yang BENAR/SALAH tentang ...?"
- "Perhatikan konteks berikut. Apa fungsi/tujuan ...?"
- "Perhatikan langkah-langkah berikut. Urutan yang benar adalah ..."
- "Perhatikan kasus singkat berikut. Manakah tindakan yang sesuai prosedur ...?"
- "Perhatikan definisi berikut. Istilah yang dimaksud adalah ..."

#### POLA YANG DIHINDARI:

- "Pertimbangkan ..." (ganti menjadi "Perhatikan ...")

#### ATURAN FORMAT SOAL (WAJIB):

Gunakan bahasa natural yang mudah dipahami siswa SMA. Soal harus langsung ke inti, tidak ambigu, dan tidak bertele-tele. Anda boleh menambahkan konteks singkat agar siswa paham situasinya, tetapi tetap batasi maksimal 3 kalimat dan maksimal 250 karakter untuk teks soal.

#### ATURAN FORMAT TEKS (WAJIB, ANTI-MARKDOWN):

- DILARANG menggunakan tanda backtick `...` untuk menandai istilah/rumus/potongan kode.
- DILARANG menggunakan Markdown/code fence seperti ``` atau format "inline code".
- Jika perlu menandai rumus/variabel/istilah, gunakan tanda kurung ( ... ) atau petik "...".

Hindari istilah yang terlalu abstrak/terlalu akademis jika modul tidak menuntutnya. Prioritaskan kata-kata yang umum dipakai siswa. Jika Anda harus memakai istilah penting (misalnya istilah hukum/ekonomi/teknologi yang ada di modul), sertakan konteks singkat atau padanan kata sederhana agar maknanya jelas.

Batasi konteks “debugging” dan kasus dunia nyata. Gunakan konteks sederhana dan netral (misal definisi, contoh/non-contoh, situasi kelas umum, atau urutan langkah prosedur di modul). Jangan membuat soal yang mengharuskan siswa membayangkan sistem nyata/kompleks.

#### JANGAN merujuk dokumen/modul secara eksplisit di teks soal.

- DILARANG memakai frasa seperti: “dalam modul...”, “menurut modul...”, “berdasarkan modul...”, “sesuai modul...”, “berdasarkan deskripsi modul...”, “pada modul...”, atau menyebut “dokumen/PDF/DOCX”.
- Prinsipnya: uji pemahaman konsep. Ambil istilah/konteks dari modul, tapi

tuliskan seolah-olah itu bagian dari materi pelajaran, bukan rujukan ke dokumen.

Setiap soal WAJIB memiliki tepat 1 jawaban benar. Tiga opsi lainnya adalah distractor yang masuk akal namun salah. Pastikan nilai pada field "jawaban\_benar" sama persis dengan salah satu string pada "options".

#### ATURAN MISKONSEPSI (WAJIB):

- Setiap butir harus memetakan tepat 1 konsep inti dan tepat 1 miskonsepsi utama.
- Distraktor harus merepresentasikan miskonsepsi umum (contoh: salah definisi, salah fungsi, salah urutan langkah, salah generalisasi), bukan jawaban acak.
- Soal tidak boleh bergantung pada pengalaman praktik; cukup dari pemahaman konsep yang tertulis di modul.

Jika modul memuat prosedur/urutan langkah, Anda boleh membuat soal berbasis prosedur dengan menyajikan langkah singkat menggunakan pemisah yang jelas (misal “1) ... ; 2) ... ; 3) ...”). Jika modul TIDAK memuat prosedur teknis/kode/rumus, maka DILARANG membuat soal berbasis kode/rumus/perhitungan.

Hindari soal yang membutuhkan gambar/diagram untuk dipahami; ubah menjadi deskripsi teks.

Berikan "explanation" singkat (20–150 karakter) yang menjelaskan alasan jawaban benar.

#### DISTRIBUSI SOAL (TOTAL 55):

- L1 (Awareness) : Medium (5)
- L2 (Literacy) : Easy (5), Hard (5)
- L3 (Capability) : Medium (5), Hard (5)
- L4 (Creativity) : Easy (5), Medium (5), Hard (5)
- L5 (Criticism) : Easy (5), Medium (5), Hard (5)

#### PENTING — DIFFICULTY WAJIB SESUAI DISTRIBUSI DI ATAS:

- L1 HANYA boleh pakai: Medium
- L2 HANYA boleh pakai: Easy atau Hard (TIDAK ada Medium di L2)
- L3 HANYA boleh pakai: Medium atau Hard (TIDAK ada Easy di L3)
- L4 & L5: boleh Easy, Medium, atau Hard

Jika soal di luar distribusi ini, soal TIDAK VALID.

#### FORMAT OUTPUT (HARUS VALID JSON ARRAY!):

```
[
  {
    "level": 1,
    "difficulty": "Medium",
    "question_type": "multiple_choice",
    "soal": "Apa yang dimaksud dengan [KONSEP UTAMA DARI MODUL]?",
    "options": ["[Jawaban benar sesuai modul]", "[Distractor 1]", "[Distractor 2]",
    "[Distractor 3]"],
```

```

    "jawaban_benar": "[Jawaban benar sesuai modul]",
    "explanation": "[Alasan singkat sesuai modul]"
  }},
  ... (54 soal lagi)
]

```

#### CATATAN TEMPLATE:

Contoh JSON di atas adalah TEMPLATE. Anda WAJIB mengganti semua teks dalam tanda [ ... ] menjadi konten nyata yang diambil dari modul.

#### PENTING:

- Output HARUS berupa JSON Array dengan 55 soal
- JANGAN tambahkan teks apapun di luar JSON
- JANGAN gunakan markdown code block (`` `json`)
- JANGAN sertakan field numerik seperti "p" / "p\_value" / "probability" — field ini sudah dihapus dari sistem.
- Field "difficulty" WAJIB berisi TEPAT salah satu dari: "Easy" | "Medium" | "Hard" (huruf kapital di awal, tanpa spasi tambahan).
- Field "level" WAJIB berisi angka integer 1–5 sesuai distribusi (bukan string "L1" dst).
- Difficulty WAJIB sesuai distribusi per level: L1→Medium; L2→Easy/Hard; L3→Medium/Hard; L4/L5→Easy/Medium/Hard.
- Explanation: 20-150 karakter (singkat & padat)
- SETIAP soal WAJIB punya 1 jawaban benar (tidak boleh 0 atau >1)
- Soal maksimal 250 karakter (cukup untuk konteks + pertanyaan)
- PRIORITAS: Kejelasan soal > Singkat (siswa harus paham maksud soal)

#### PENGAMAN SCOPE (WAJIB):

Jika modul membahas konteks sosial/ekonomi/hukum/sejarah/karier, gunakan konteks tersebut secara sederhana dan netral (tanpa studi kasus kompleks). Jangan menyisipkan domain lain seperti latihan sintaks pemrograman, hasil operasi kode, atau debugging kode, kecuali modul memang mengajarkannya.

#### POLA SOAL (template lintas konteks modul):

- Level 1: istilah, definisi, tokoh, fakta dasar, atau karakteristik yang disebut modul.
- Level 2: fungsi/manfaat, hubungan konsep, sebab–akibat dalam konteks modul.
- Level 3: penerapan prosedur/aktivitas yang diajarkan (cara melakukan, cara mengakses, cara menggunakan) sesuai modul.
- Level 4: identifikasi konsep/langkah yang keliru, memilih koreksi/perbaikan langkah, atau memilih pernyataan yang tepat dalam skenario sederhana.
- Level 5: evaluasi pilihan/pernyataan berdasarkan kriteria di modul (tujuan, syarat, batasan, manfaat) secara objektif.

```

"""

```

```

    return prompt

```

## Attachment 17 Adaptive Assessment Logic (Grading &amp; MST Routing)

```

# MST Routing Logic - Simple 5 Stage System with User's Taxonomy
# Implements 4 Scenarios: High Ability (L1->L2H->L3H->L4H->L5H),
Medium-High (L1->L2H->L3H->L4M->L5M->STOP),
# Medium-Low (L1->L2H->L3M->L4E->STOP), Low Ability (L1->L2E-
->STOP)
MST_ROUTING = {
  # Stage 1: L1 Awareness (Medium only) - Starting point for all students
  1: {
    "technology_level": 1,
    "difficulty": "Medium",
    "pass": {"stage": 2, "technology_level": 2, "difficulty": "Hard"}, # L1M
Pass -> L2H (Stage 2)
    "fail": {"stage": 2, "technology_level": 2, "difficulty": "Easy"} # L1M Fail
-> L2E (Stage 2)
  },
  # Stage 2: L2 Literacy (Easy or Hard based on Stage 1 result)
  2: {
    "Hard": { # L2H from L1M Pass
      "pass": {"stage": 3, "technology_level": 3, "difficulty": "Hard"}, # L2H
Pass -> L3H (Stage 3)
      "fail": {"stage": 3, "technology_level": 3, "difficulty": "Medium"} # L2H
Fail -> L3M (Stage 3)
    },
    "Easy": { # L2E from L1M Fail
      "pass": {"stage": 3, "technology_level": 3, "difficulty": "Medium"}, #
L2E Pass -> L3M (Stage 3)
      "fail": {"stage": "STOP", "diagnosis": "< L2"} # L2E Fail -
> STOP (Diagnosis: < L2)
    }
  },
  # Stage 3: L3 Capability (Medium or Hard based on Stage 2 result)
  3: {
    "Hard": { # L3H from L2H Pass
      "pass": {"stage": 4, "technology_level": 4, "difficulty": "Hard"}, # L3H
Pass -> L4H (Stage 4)
      "fail": {"stage": 4, "technology_level": 4, "difficulty": "Medium"} # L3H
Fail -> L4M (Stage 4)
    },
    "Medium": { # L3M from L2H Fail or L2E Pass
      "pass": {"stage": 4, "technology_level": 4, "difficulty": "Medium"}, #
L3M Pass -> L4M (Stage 4)
      "fail": {"stage": 4, "technology_level": 4, "difficulty": "Easy"} # L3M
Fail -> L4E (Stage 4)
    }
  },
  # Stage 4: L4 Creativity (Easy, Medium, or Hard based on Stage 3 result)

```

```

4: {
  "Hard": { # L4H from L3H Pass (High Ability Path)
    "pass": {"stage": 5, "technology_level": 5, "difficulty": "Hard"}, # L4H
    Pass -> L5H (Stage 5)
    "fail": {"stage": 5, "technology_level": 5, "difficulty": "Medium"} # L4H
    Fail -> L5M (Stage 5)
  },
  "Medium": { # L4M from L3H Fail or L3M Pass (Medium-High Path)
    "pass": {"stage": 5, "technology_level": 5, "difficulty": "Medium"}, #
    L4M Pass -> L5M (Stage 5)
    "fail": {"stage": 5, "technology_level": 5, "difficulty": "Easy"} # L4M
    Fail -> L5E (Stage 5)
  },
  "Easy": { # L4E from L3M Fail (Lower Path)
    "pass": {"stage": 5, "technology_level": 5, "difficulty": "Easy"}, # L4E
    Pass -> L5E (Stage 5)
    "fail": {"stage": "STOP", "diagnosis": "L3"} # L4E Fail ->
    STOP (Diagnosis: L3)
  }
},
# Stage 5: L5 Criticism (Easy, Medium, or Hard based on Stage 4 result) - Final
stage
5: {
  "Hard": { # L5H from L4H Pass (Highest ability path)
    "pass": {"stage": "STOP", "diagnosis": "L5"}, # L5H Pass
    -> STOP (Diagnosis: L5)
    "fail": {"stage": "STOP", "diagnosis": "L4"} # L5H Fail -
    > STOP (Diagnosis: L4)
  },
  "Medium": { # L5M from L4H Fail or L4M Pass (Medium-high path)
    "pass": {"stage": "STOP", "diagnosis": "L5"}, # L5M Pass
    -> STOP (Diagnosis: L5)
    "fail": {"stage": "STOP", "diagnosis": "L4"} # L5M Fail -
    > STOP (Diagnosis: L4)
  },
  "Easy": { # L5E from L4M Fail or L4E Pass (Lower ability path)
    "pass": {"stage": "STOP", "diagnosis": "L5"}, # L5E Pass -
    > STOP (Diagnosis: L5)
    "fail": {"stage": "STOP", "diagnosis": "L4"} # L5E Fail ->
    STOP (Diagnosis: L4)
  }
}
}
}

```

```

def determine_next_stage(state):
    """

```

Implementasi alur MST adaptif 5 Stage sesuai taxonomy user - 1 SOAL per stage

Returns: (next\_stage, next\_level, should\_stop, diagnosis)

4 Skenario Alur MST:

1. High Ability: L1M(PASS)->L2H(PASS)->L3H(PASS)->L4H(PASS)->L5H(PASS/FAIL)->STOP
2. Medium-High: L1M(PASS)->L2H(PASS)->L3H(FAIL)->L4M(PASS)->L5M(PASS/FAIL)->STOP
3. Medium-Low: L1M(PASS)->L2H(FAIL)->L3M(PASS)->L4E(PASS/FAIL)->STOP
4. Low Ability: L1M(FAIL)->L2E(FAIL)->STOP

```
stage = state.current_stage
```

```
level = state.current_level
```

```
correct_answers = state.questions_correct
```

```
# 1 soal per stage, jadi passed = (correct_answers >= 1)
```

```
passed = (correct_answers >= 1)
```

```
print(f"[MST] Stage {stage} L {stage} {level[0]}: {correct_answers} correct = {'PASS' if passed else 'FAIL'}")
```

```
# Get routing info from MST_ROUTING
```

```
if stage in MST_ROUTING:
```

```
    stage_config = MST_ROUTING[stage]
```

```
    # Stage 1 is special - no difficulty branching
```

```
    if stage == 1:
```

```
        route = stage_config["pass"] if passed else stage_config["fail"]
```

```
    else:
```

```
        # Stages 2-5 have difficulty-based routing
```

```
        if level in stage_config:
```

```
            route = stage_config[level]["pass"] if passed else
```

```
            stage_config[level]["fail"]
```

```
        else:
```

```
            print(f"[MST ERROR] Stage {stage} level {level} not found in routing config")
```

```
            return (None, None, True, "Configuration Error")
```

```
    # Process route result
```

```
    if route["stage"] == "STOP":
```

```
        return (None, None, True, route["diagnosis"])
```

```
    else:
```

```
        return (route["stage"], route["difficulty"], False, None)
```

```
# Fallback - should not happen with proper config
```

```
print(f"[MST ERROR] Stage {stage} not found in MST_ROUTING")
```

```
return (None, None, True, f"L {stage}")
```

```

# Endpoint untuk submit jawaban (Bagian Penilaian)
@app.route('/submit_answer', methods=['POST'])
def submit_answer():
    """MST Adaptive Answer Processing"""
    data = request.get_json(silent=True) or {}
    user_id = data.get("user_id", "default")
    raw_answer = data.get("answer", "")
    answer = raw_answer.strip() if isinstance(raw_answer, str) else str(raw_answer)
    question_id = data.get("question_id")
    collection_id = data.get("collection_id")

    print(f"[MST] ===== SUBMIT ANSWER REQUEST =====")
    print(f"[MST] user_id: {user_id} (type: {type(user_id)})")
    print(f"[MST] jawaban: {answer[:80]}...")
    print(f"[MST] question_id: {question_id} (type: {type(question_id)})")
    print(f"[MST] collection_id: {collection_id} (type: {type(collection_id)})")
    print(f"[MST] =====")

    try:
        # Validasi dan konversi ID user
        try:
            siswa_id = int(user_id)
        except (TypeError, ValueError) as e:
            return jsonify({
                "status": "error",
                "message": f"User ID tidak valid: '{user_id}'. Pastikan Anda sudah
login."
            }), 400

        # Validasi siswa exists
        siswa = Siswa.query.get(siswa_id)
        if not siswa:
            return jsonify({
                "status": "error",
                "message": f"Siswa dengan ID {siswa_id} tidak ditemukan. Silakan
login ulang."
            }), 400

        print(f"[MST] OK Siswa validated: {siswa.nama} (ID: {siswa.id})")

        # Validasi input
        if question_id is None or collection_id is None:
            return jsonify({
                "status": "error",
                "message": "Data tidak lengkap: question_id atau collection_id hilang"
            }), 400

        # Konversi ID dengan safety check

```

```

try:
    question_id_int = int(question_id)
    collection_id_int = int(collection_id)
except (TypeError, ValueError):
    return jsonify({
        "status": "error",
        "message": "ID soal atau koleksi tidak valid"
    }), 400

# Get MST state
mst_state = get_mst_state(siswa_id, collection_id_int)
print(f'[MST] MST State SEBELUM proses: Stage
{mst_state.current_stage}, Level {mst_state.current_level}, Selesai?
{mst_state.test_completed}')

if mst_state.test_completed:
    return jsonify({
        "status": "game over",
        "message": f'Tes sudah selesai! Diagnosis: {mst_state.final_diagnosis}'
    }), 200

# Get question
question = Question.query.get(question_id_int)
if not question:
    return jsonify({"status": "error", "message": "Soal tidak ditemukan"}), 404

# Check if answer is correct - DENGAN NULL SAFETY
correct_answer = (question.jawaban_benar or "").strip()
user_answer_norm = answer.strip().lower()
correct_answer_norm = correct_answer.lower()
is_correct = (user_answer_norm == correct_answer_norm) if correct_answer
else False

print(f'[MST] Jawaban: '{answer}' vs '{correct_answer}' => {'BENAR' if
is_correct else 'SALAH'}')

# --- PERBAIKAN LOGIKA: Simpan SEMUA jawaban (benar dan salah)
untuk riwayat ---
try:
    level_name = TECHNOLOGY_LEVELS.get(mst_state.current_stage,
str(mst_state.current_stage))
    user_answer_record = SiswaAnswer(
        siswa_id=siswa_id,
        question_id=question_id_int,
        collection_id=collection_id_int,
        siswa_answer=answer,
        selected_answer=answer,
        is_correct=is_correct, # Simpan status benar/salah

```

```

        stage=mst_state.current_stage,
        level=level_name,
        difficulty=question.difficulty if hasattr(question, 'difficulty') and
question.difficulty else 'Medium'
    )
    db.session.add(user_answer_record)
    # Jangan commit dulu, tunggu update state MST
    print(f"[MST] Answer record DIBUAT (is_correct={is_correct}) (belum
commit)")
    except Exception as e_ans:
        print(f"[MST] Error membuat record jawaban: {str(e_ans)}")
        db.session.rollback()
        return jsonify({"status": "error", "message": f"Error menyimpan jawaban:
{str(e_ans)}"}), 500
    # --- PERBAIKAN SELESAI ---

    # Update MST state counters (setelah jawaban dibuat, sebelum menentukan
stage berikutnya)
    # Note: counter ini HANYA untuk stage saat ini
    mst_state.questions_answered = (mst_state.questions_answered or 0) + 1
    if is_correct:
        mst_state.questions_correct = (mst_state.questions_correct or 0) + 1

    # Check if stage module is completed - MST ADAPTIF: 1 SOAL PER
STAGE
    stage_question_limit = 1 # Sesuai skenario: 1 soal per stage

    if (mst_state.questions_answered or 0) >= stage_question_limit:
        # Evaluate stage completion
        next_stage, next_level_difficulty, should_stop, diagnosis =
determine_next_stage(mst_state)
        print(f"[MST] Hasil determine_next_stage: next_stage={next_stage},
next_level_difficulty={next_level_difficulty}, should_stop={should_stop},
diagnosis={diagnosis}")

    # --- PERBAIKAN UTAMA DIMULAI: Update state jika tes selesai ---
    if should_stop:
        # Test selesai
        mst_state.test_completed = True
        mst_state.final_diagnosis = diagnosis
        print(f"[MST] TES SELESAI! Diagnosis: {diagnosis} akan disimpan.")
        try:
            db.session.commit() # Commit semua perubahan (jawaban + state
MST yang sudah final)
            print("[MST] Commit Berhasil - Status tes selesai disimpan")
        except Exception as e_commit_stop:
            print(f"[MST] Error commit saat tes selesai: {str(e_commit_stop)}")
            db.session.rollback()

```

```

        return jsonify({"status": "error", "message": f"Error finalisasi tes:
{str(e_commit_stop)}"}), 500

    return jsonify({
        "status": "game over", # Frontend akan redirect ke halaman hasil
        "message": f"Tes selesai! Diagnosis akhir Anda adalah: {diagnosis}",
        "explanation": question.explanation,
        "is_correct": is_correct,
        "correct_answer": correct_answer
    }), 200
# --- PERBAIKAN UTAMA SELESAI ---
else:
    # Lanjut ke stage/level berikutnya
    print(f"[MST] Lanjut ke Stage {next_stage} Difficulty
{next_level_difficulty}")
    mst_state.current_stage = next_stage
    mst_state.current_level = next_level_difficulty # Simpan difficulty
    untuk stage berikutnya
    mst_state.questions_answered = 0 # Reset counter untuk stage baru
    mst_state.questions_correct = 0 # Reset counter untuk stage baru

    try:
        # === SOLUSI TAMBAHAN: UPDATE SiswaResult SEKARANG
        ===
        student_result = SiswaResult.query.filter_by(
            siswa_id=siswa_id,
            collection_id=collection_id_int
        ).first()

        if student_result:
            # Update level siswa saat ini ke stage yang baru
            student_result.current_level = next_stage
            print(f"[MST] SiswaResult diupdate: current_level={next_stage}")
            # === SOLUSI TAMBAHAN SELESAI ===

            db.session.commit() # Commit semua perubahan (jawaban + state
MST + state SiswaResult)
            print("[MST] Commit Berhasil - Lanjut ke stage berikutnya")
        except Exception as e_commit_next:
            print(f"[MST] Error commit saat lanjut stage: {str(e_commit_next)}")
            db.session.rollback()
            return jsonify({"status": "error", "message": f"Error lanjut stage:
{str(e_commit_next)}"}), 500

    return jsonify({
        "status": "continue",
        "message": f"{'Benar!' if is_correct else 'Salah.'} Lanjut ke Stage
{next_stage} - {next_level_difficulty}",

```

```

        "explanation": question.explanation,
        "next_stage": next_stage,
        "next_level": next_level_difficulty, # Kirim difficulty berikutnya
        "is_correct": is_correct,
        "correct_answer": correct_answer
    }), 200
else:
    # Kasus ini seharusnya tidak terjadi jika stage_question_limit = 1
    # Jika limit > 1, commit jawaban saja
    try:
        db.session.commit()
        print("[MST] Commit Berhasil - Masih dalam stage yang sama (limit >
1)")
    except Exception as e_commit_same:
        print(f"[MST] Error commit saat masih di stage sama:
{str(e_commit_same)}")
        db.session.rollback()
        return jsonify({"status": "error", "message": f"Error simpan jawaban:
{str(e_commit_same)}"}), 500

    progress = f"{mst_state.questions_answered}/{stage_question_limit}"
    return jsonify({
        "status": "continue",
        "message": f"{'Benar!' if is_correct else 'Salah.'} Progress Stage
{mst_state.current_stage}: {progress}",
        "explanation": question.explanation,
        "stage_progress": progress,
        "is_correct": is_correct,
        "correct_answer": correct_answer
    }), 200

except ValueError as e:
    print(f"[MST] ValueError di submit_answer: {str(e)}")
    # Tidak perlu rollback karena belum ada perubahan DB signifikan
    return jsonify({
        "status": "error",
        "message": f"Format data tidak valid: {str(e)}"
    }), 400

except Exception as e:
    print(f"[MST] Exception di submit_answer: {str(e)}")
    print(f"[MST] Error type: {type(e).__name__}")
    import traceback
    print(f"[MST] Traceback: {traceback.format_exc()}")
    db.session.rollback() # Rollback jika ada error tak terduga
    return jsonify({
        "status": "error",
        "message": f"Kesalahan server: {str(e)}"
    }), 500

```



## Attachment 18 Black Box Test Results

## UJI BLACK BOX

Nama Lengkap : Anak Agung Gede Pradnya Prameswara  
 Instansi/ Sekolah : Universitas Pendidikan Ganesha

**PETUNJUK:**

Untuk setiap modul, jawab pertanyaan "Relevan" jika hasil yang diharapkan sesuai dengan keluaran sistem saat diuji. Berikan "Tidak Relevan" jika tidak sesuai, jika pertanyaan tidak berlaku. Cantumkan catatan atau detail hasil observasi pada kolom "Catatan/Hasil Observasi".

No	Modul	Pertanyaan Uji	Catatan/ Hasil Observasi	Relevan (✓)	Tidak Relevan (x)
1	Autentikasi Pengguna	Apakah sistem berhasil memberikan akses dan mengarahkan pengguna ke dashboard yang sesuai (Guru/Siswa) ketika nama pengguna dan kata sandi valid dimasukkan?	Sistem berhasil memverifikasi <i>hash</i> password. Pengguna dengan <code>user_type='guru'</code> langsung diarahkan ke rute <code>/guru</code> (Dashboard Guru), dan <code>user_type='siswa'</code> diarahkan ke <code>/DashboardSiswa</code> .	✓	
2	Autentikasi Pengguna	Apakah sistem menolak akses dan menampilkan pesan galat yang jelas ketika nama pengguna atau kata sandi yang tidak valid dimasukkan?	Login gagal. Sistem tetap di halaman login dan menampilkan <i>flash message</i> "Email atau password salah" sesuai logika di fungsi <code>login()</code> .	✓	
3	Registrasi Pengguna	Apakah akun baru berhasil dibuat dan pengguna diarahkan kembali ke halaman masuk setelah mengisi data pendaftaran yang valid?	Data tersimpan di tabel <code>guru</code> atau <code>siswa</code> . Pengguna diarahkan ke halaman login dengan notifikasi sukses "Registrasi berhasil! Silakan login."	✓	
4	Registrasi Pengguna	Apakah sistem menolak proses pendaftaran jika email sudah terdaftar?	Sistem mendeteksi duplikasi email di database. Pendaftaran ditolak dan muncul pesan peringatan "Email sudah	✓	

			terdaftar".		
5	Unggah Modul Ajar	Apakah sistem berhasil menyimpan berkas, mengekstrak teks, dan menginisiasi proses AI setelah guru mengunggah berkas valid (PDF/DOCX)?	File PDF/DOCX terbaca oleh PyPDF2/python-docx. Teks terekstrak dan dikirim ke API Gemini. Respons JSON diterima dan disimpan ke tabel questions.	✓	
6	Unggah Modul Ajar	Apakah sistem menolak berkas dan menampilkan pesan galat jika format tidak didukung?	Upload file selain .pdf atau .docx ditolak oleh validator ekstensi file di app.py dengan pesan error yang sesuai.	✓	
7	Generasi Soal Diagnostik (AI)	Apakah AI berhasil menghasilkan soal pilihan ganda sesuai distribusi level dan kesulitan (MST) yang disimpan rapi ke dalam database?	Soal tersimpan di database dengan distribusi lengkap (L1-L5) sesuai <i>prompt</i> . Format JSON valid, opsi jawaban dan kunci jawaban tersimpan dengan benar.	✓	
8	Pembuatan Koleksi Soal	Apakah koleksi soal baru berhasil dibuat dan muncul di daftar koleksi guru?	Data baru masuk ke tabel <code>question_collections</code> . Nama dan deskripsi koleksi tampil dengan benar di halaman <code>collections.html</code> .	✓	
9	Penambahan Soal ke Koleksi	Apakah soal dari bank soal berhasil ditambahkan ke koleksi tanpa duplikasi?	Soal terpilih berhasil dihubungkan ke koleksi melalui tabel relasi <code>collection_questions</code> . Tidak ada duplikasi ID soal dalam satu koleksi.	✓	
10	Validasi Soal	Apakah status soal berubah menjadi "Tervalidasi" di database saat guru menyetujuinya?	Tombol "Validasi" di dashboard mengubah status kolom <code>is_validated</code> menjadi 1 (True) di tabel <code>questions</code> .	✓	
11	Revisi Soal	Apakah sistem menyimpan versi baru	Saat guru mengedit soal, data lama	✓	

		soal ketika guru melakukan penyuntingan?	tersimpan di tabel question_versions dan data utama di tabel questions diperbarui (versi naik +1).		
12	Penambahan Siswa ke Koleksi	Apakah siswa berhasil ditambahkan ke koleksi tes tertentu?	Siswa atau kelas yang dipilih berhasil terhubung ke koleksi melalui tabel collection_students. Siswa dapat melihat tes tersebut di dashboard mereka.	✓	
13	Penghapusan Siswa dari Koleksi	Apakah akses siswa berhasil dicabut dari koleksi tes?	Relasi di tabel collection_students terhapus. Tes hilang dari daftar "Tes Tersedia" di dashboard siswa yang bersangkutan.	✓	
14	Penghapusan Koleksi Soal	Apakah koleksi berhasil dihapus beserta relasinya tanpa menghapus data master soal?	Koleksi terhapus dari question_collections, namun data soal asli di tabel questions tetap aman (hanya relasinya yang putus).	✓	
15	Pemilihan Asesmen (Siswa)	Apakah dasbor siswa menampilkan daftar tes yang ditugaskan dan tombol "Kerjakan"?	Daftar tes yang ditugaskan muncul di Dashboard_siswa.html. Tombol "Kerjakan" mengarahkan siswa ke rute pengerjaan tes.	✓	
16	Pengerjaan Tes (MST Adaptif)	Apakah sistem menyajikan soal satu per satu dan menyesuaikan tingkat kesulitan (routing) berdasarkan jawaban siswa (Benar/Salah)?	Soal tampil per halaman. Logika determine_next_stage berfungsi: jawaban benar mengarah ke soal level lebih tinggi/sulit, jawaban salah ke level lebih rendah/mudah.	✓	
17	Hasil & Umpan Balik AI (Siswa)	Apakah sistem menampilkan skor akhir dan paragraf umpan balik personal yang di-	Halaman result.html menampilkan skor akurasi dan paragraf narasi umpan balik	✓	

		generate oleh AI setelah tes selesai?	yang dihasilkan Gemini berdasarkan analisis pola jawaban siswa.		
18	Ekspor Hasil (Siswa)	Apakah berkas PDF laporan hasil pribadi berhasil diunduh siswa?	Tombol ekspor memicu fungsi WeasyPrint. File PDF berhasil terunduh berisi nama siswa, skor, dan teks umpan balik AI.	✓	
19	Analisis Hasil (Guru)	Apakah sistem menampilkan grafik statistik distribusi level dan akurasi siswa pada halaman detail koleksi?	Halaman collection-analytics.html berhasil me-render grafik batang distribusi level dan tabel ringkasan nilai siswa secara <i>real-time</i> .	✓	
20	Ekspor Laporan (Guru)	Apakah berkas Excel berisi rekapitulasi nilai seluruh siswa berhasil diunduh oleh guru?	File .xlsx berhasil di-generate menggunakan pandas. Berisi data lengkap seluruh siswa (Nama, Kelas, Skor Benar/Salah, Level Akhir).	✓	

Mengetahui, 14 Oktober 2025



Anak Agung Gede Pradnya Prameswara

## Attachment 19 Results of the Phase 1 PROMPT AI White Box Expert Trial

## UJI WHITE BOX

Nama Lengkap : Ir. Gede Surya Mahendra, S.Pd.,M.Kom.

Instansi/ Sekolah : Progam Studi Sistem Informasi, Undiksha

Mata Pelajaran yang Diampu : Pemrograman Web

Jabatan : Dosen

**PETUNJUK:**

Untuk setiap fungsi, jawab pertanyaan "[X]" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "[X]" jika tidak tercapai. Cantumkan detail observasi, cakupan kode, atau log hasil uji pada "Catatan".

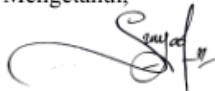
No	Modul	Fungsi	Kutipan Kode	Pertanyaan Uji	Relevan [X]	Tidak Relevan [X]
1	Autentikasi	login()	user = Guru.query...	Semua jalur eksekusi login diuji (guru, siswa, salah password).	[X]	
2	Registrasi	register()	if password != confirm_password:	Percabangan validasi diuji (password mismatch, email/username duplikat, kelas siswa kosong).	[X]	
3	Unggah Modul Ajar	upload_file ()	if file_ext not in ...	Validasi format file (PDF/DOC/DOCX), role guru, dan error handling diuji.	[X]	
4	Evaluasi Jawaban	is_answer_correct()	return user_answer.lower().strip() == expected_answer.lower().strip()	Perbandingan jawaban sensitif pada trimming, lowercasing, variasi minor.	[X]	
5	Interaksi DB	SiswaResult, SiswaAnswer	user_result.correct+= 1...	Update correct, incorrect, current_level konsisten.	[X]	
6	Generate Soal Multi-Stage	upload_file () (prompt)	prompt = f"...INSTRUKSI..."	Prompt menghasilkan 35 soal (7 level × 5), opsi max 15 kata, JSON valid, tanpa eksplisit "Modul Ajar".	[X]	
7	Parsing Soal AI	upload_file ()	json.loads(cleaned_text)	JSON dari AI dibersihkan & fallback parsing berjalan bila gagal.	[X]	
8	Generate Rekomendasi Belajar	generate_ai_recommendations()	prompt = f"...Anda adalah GURU AHLI..."	Rekomendasi 5–6 poin, gaya guru, bullet (•), sesuai mastery level.	[X]	

9	Ekstraksi Tujuan Pembelajaran	<code>extract_learning_objects()</code>	<code>if hasattr(question,'module_reference')...</code>	Fungsi mengambil tujuan dari modul → fallback explanation → fallback keyword soal.	<input checked="" type="checkbox"/>	
10	Ekstraksi Kata Kunci	<code>extract_keywords_from_text()</code>	<code>word_freq[word] = ...</code>	Kata kunci diambil berdasar frekuensi dengan stopwords filtering.	<input checked="" type="checkbox"/>	
11	Ekstraksi Tujuan dari Penjelasan	<code>extract_objective_from_text()</code>	<code>'if re.search(r'\b(tujuan kompetensi capaian)\b', sentence)'</code>	Kalimat dengan kata tujuan, kompetensi, capaian ditangkap.	<input checked="" type="checkbox"/>	
12	Identifikasi Domain	<code>identify_learning_domains()</code>	<code>if any(word in text for word in [...])</code>	Domain kognitif/praktis/reflektif diidentifikasi lewat kata kunci instruksi soal.	<input checked="" type="checkbox"/>	
13	Konteks untuk Rekomendasi	<code>get_questions_context_for_recommendations()</code>	<code>context_parts.append(...)</code>	Fungsi gabungkan referensi modul, konsep kunci, isi modul, distribusi level.	<input checked="" type="checkbox"/>	
14	Analisis Performa Siswa	<code>analyze_student_performance()</code>	<code>if (perf['correct']/perf['total'])&lt;0.6:</code>	Performa per level & modul dihitung benar; area lemah (<60%) terdeteksi.	<input checked="" type="checkbox"/>	
15	Parsing Rekomendasi AI	<code>parse_ai_recommendations_for_teacher()</code>	<code>clean_line = re.sub(...)</code>	Hasil AI dibersihkan dari format teknis & dipaksa jadi kalimat guru-style.	<input checked="" type="checkbox"/>	
16	Rekomendasi Default Guru	<code>get_teacher_style_recommendations()</code>	<code>if mastery_level=="rendah": ...</code>	Fallback rekomendasi dibuat sesuai mastery level + motivasi.	<input checked="" type="checkbox"/>	
17	Rekomendasi Kontekstual	<code>get_contextual_default_recommendations()</code>	<code>recommendations={"PENGUATAN KONSEP":...}</code>	Rekomendasi dibagi 3 domain (Konsep, Praktik, Evaluasi), pakai keyword tujuan pembelajaran.	<input checked="" type="checkbox"/>	

Catatan: Sedikit di luar pengujian white box, dapat dipelajari alasan penggunaan angka 60% dalam pengujian 14, alasannya dari mana, apakah ada teori atau kalkulasi yang mendukung.

Singaraja, 22 September 2025

Mengetahui,



Ir. Gede Surya Mahendra, S.Pd.,M.Kom.

## Attachment 20 Results of the Phase 2 PROMPT AI White Box Expert Trial

## UJI WHITE BOX

Nama Lengkap : Ir. Gede Surya Mahendra, S.Pd.,M.Kom.  
 Instansi/ Sekolah : Progam Studi Sistem Informasi, Undiksha  
 Mata Pelajaran yang Diampu : Pemrograman Web  
 Jabatan : Dosen

## PETUNJUK:

Untuk setiap fungsi, jawab pertanyaan "☑" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "☒" jika tidak tercapai. Cantumkan detail observasi, cakupan kode, atau log hasil uji pada "Catatan".

No	Modul	Fungsi	Kutipan Kode	Pertanyaan Uji	Relevan ☑	Tidak Relevan ☒
1	Autentikasi	login()	user = Guru.query...	Semua jalur eksekusi login diuji (guru, siswa, salah password).	✓	
2	Registrasi	register()	if password != confirm_password:	Percabangan validasi diuji (password mismatch, email/username duplikat, kelas siswa kosong).	✓	
3	Unggah Modul Ajar	upload_file()	if file_ext not in ...	Validasi format file (PDF/DOC/DOCX), role guru, dan error handling diuji.	✓	
4	Evaluasi Jawaban	is_answer_correct()	return user_answer.lower().strip() == expected_answer.lower().strip()	Perbandingan jawaban sensitif pada trimming, lowercasing, variasi minor.	✓	
5	Interaksi DB	SiswaResult, SiswaAnswer	user_result.correct++	Update correct, incorrect, current_level konsisten.	✓	
6	Generate Soal Multi-Stage	upload_file(prompt)	prompt = f"...INSTRUKSI..."	Prompt menghasilkan 35 soal (7 level x 5), opsi max 15 kata, JSON valid, tanpa eksplisit "Modul Ajar".	✓	
7	Parsing Soal AI	upload_file()	json.loads(cleaned_text)	JSON dari AI dibersihkan & fallback parsing berjalan bila gagal.	✓	
8	Generate Rekomendasi Belajar	generate_ai_recommendations()	prompt = f"...Anda adalah GURU AHLI..."	Rekomendasi 5-6 poin, gaya guru, bullet (-), sesuai mastery level.	✓	



9	Ekstraksi Tujuan Pembelajaran	extract_learning_objectives()	if hasattr(question,'module_reference')...	Fungsi mengambil tujuan dari modul → fallback explanation → fallback keyword soal.	✓	
10	Ekstraksi Kata Kunci	extract_keywords_from_text()	word_freq[word] = ...	Kata kunci diambil berdasar frekuensi dengan stopwords filtering.	✓	
11	Ekstraksi Tujuan dari Penjelasan	extract_objective_from_text()	'if re.search(r'^b(tujuan kompetensi capaian)\b', sentence)	Kalimat dengan kata tujuan, kompetensi, capaian ditangkap.	✓	
12	Identifikasi Domain	identify_learning_domains()	if any(word in text for word in [...])	Domain kognitif/praktis/reflektif diidentifikasi lewat kata kunci instruksi soal.	✓	
13	Konteks untuk Rekomendasi	get_questions_context_for_recommendations()	context_parts.append(...)	Fungsi gabungkan referensi modul, konsep kunci, isi modul, distribusi level.	✓	
14	Parsing Rekomendasi AI	parse_ai_recommendations_for_teacher()	clean_line = re.sub(...)	Hasil AI dibersihkan dari format teknis & dipaksa jadi kalimat guru-style.	✓	
15	Rekomendasi Default Guru	get_teacher_style_recommendations()	if mastery_level=="rendah": ...	Fallback rekomendasi dibuat sesuai mastery level + motivasi.	✓	
16	Rekomendasi Kontekstual	get_contextual_default_recommendations()	recommendations={"PENGUATAN KONSEP":...}	Rekomendasi dibagi 3 domain (Konsep, Praktik, Evaluasi), pakai keyword tujuan pembelajaran.	✓	

Catatan:

*Sudah saya dengan harapan uji. Modul yang diujikan  
 klah sesuai dengan petanyaan uji yang diuraikan.*

Singaraja, 16 Oktober 2025

Mengetahui,



Ir. Gede Surya Mahendra, S.Pd.,M.Kom.

## Attachment 21 White Box Results of System Expert Test

## UJI WHITE BOX

Nama Lengkap : Dr. Bagus Gede Krishna Yudistira, M.Kom.  
 Instansi/ Sekolah : Progam Studi Pendidikan Teknik Informatika, Undiksha  
 Mata Pelajaran yang Diampu : Keamanan Cyber; Jaringan Komputer; Komputasi Cloud  
 Jabatan : Dosen

## PETUNJUK:

Untuk setiap fungsi, jawab pertanyaan "☑" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "☒" jika tidak tercapai. Cantumkan detail observasi, cakupan kode, atau log hasil uji pada "Catatan".

No	Modul	Fungsi	Kutipan Kode	Pertanyaan Uji	Relevan ☑	Tidak Relevan ☒
1	Autentikasi	login()	user = Guru.query...	Semua jalur eksekusi login diuji (guru, siswa, salah password).	✓	
2	Registrasi	register()	if password != confirm_password:	Percabangan validasi diuji (password mismatch, email/username duplikat, kelas siswa kosong).	✓	
3	Unggah Modul Ajar	upload_file()	if file_ext not in ...	Validasi format file (PDF/DOC/DOCX), role guru, dan error handling diuji.	✓	
4	Evaluasi Jawaban	is_answer_correct()	return user_answer.lower().strip() == expected_answer.lower().strip()	Perbandingan jawaban sensitif pada trimming, lowercasing, variasi minor.	✓	
5	Interaksi DB	SiswaResult, SiswaAnswer	user_result.correct+=1...	Update correct, incorrect, current_level konsisten.	✓	
6	Generate Soal Multi-Stage	upload_file(prompt)	prompt = f"...INSTRUKSI..."	Prompt menghasilkan 35 soal (7 level x 5), opsi max 15 kata, JSON valid, tanpa eksplisit "Modul Ajar".	✓	
7	Parsing Soal AI	upload_file()	json.loads(cleaned_text)	JSON dari AI dibersihkan & fallback parsing berjalan bila gagal.	✓	
8	Generate Rekomendasi Belajar	generate_ai_recommendations()	prompt = f"...Anda adalah GURU AHLI..."	Rekomendasi 5-6 poin, gaya guru, bullet (•), sesuai mastery level.	✓	

UNDIKSHA



## Attachment 22 Expert Test Results Contents 1 Stage 1

## UJI AHLI ISI

Nama Lengkap : Dr. Nyoman Sugihartini, S.Pd., M.Pd.  
 Instansi/ Sekolah : Universitas Pendidikan Ganesha  
 Mata Pelajaran yang Diampu : Micro teaching  
 Jabatan : Dosen

## PETUNJUK:

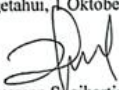
Untuk setiap fungsi, jawab pertanyaan "☑" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "☒" jika tidak tercapai. Cantumkan detail observasi, fitur, atau saran pada "Catatan".

No	Pernyataan	Relevan	Tidak Relevan
		☑	☒
1	Materi dalam soal sesuai dengan indikator capaian pembelajaran pada modul ajar.	✓	
2	Butir soal secara jelas mengarah pada pencapaian tujuan pembelajaran yang ditargetkan.	✓	
3	Isi materi dalam soal dan kunci jawaban yang disediakan sudah akurat secara akademik.	✓	
4	Konteks/topik yang digunakan dalam soal relevan dengan materi ajar Informatika.	✓	
5	Tingkat kesulitan soal sudah sesuai untuk level siswa (Kelas X).	✓	
6	Opsi jawaban (pengecoh) yang disediakan masuk akal dan efektif untuk menguji pemahaman.	✓	
7	Bahasa yang digunakan dalam soal jelas, tidak ambigu, dan mudah dipahami oleh siswa.	✓	
8	Stimulus yang digunakan (jika ada, seperti teks atau gambar) relevan dan efektif untuk mendukung pertanyaan.	✓	tidak ada gambar
9	Soal bebas dari bias (misalnya gender, suku, budaya dan tidak mengarah pada jawaban tertentu.)	✓	
10	Butir soal memiliki potensi untuk mendeteksi miskonsepsi atau kesulitan belajar spesifik siswa.	✓	

## Catatan:

Soal yang ditampilkan sudah baik dan sesuai, hanya saja belum ada unsur SW + H pada awal pertanyaan. Akan tetapi secara umum soal sudah dapat dipahami maknanya.

Mengetahui, 1 Oktober 2025

  
 Dr. Nyoman Sugihartini, S.Pd., M.Pd.



## Attachment 24 Expert Test Results Content 2

**UJI AHLI ISI**

Nama Lengkap : Dr. Gede Saindra Santyadiputra, S.T., M.Cs.  
 Instansi/ Sekolah : Universitas Pendidikan Ganesha  
 Mata Pelajaran yang Diampu : Komunikasi Data dan Jaringan Komputer; Sistem Informasi;  
 Dasar Sistem Komputer; Organisasi dan Arsitektur Komputer  
 Jabatan : Dosen

**PETUNJUK:**

Untuk setiap fungsi, jawab pertanyaan "" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "" jika tidak tercapai. Cantumkan detail observasi, fitur, atau saran pada "Catatan".

No	Pernyataan	Relevan	Tidak Relevan
		<input checked="" type="checkbox"/>	<input type="checkbox"/>
1	Materi dalam soal sesuai dengan indikator capaian pembelajaran pada modul ajar.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Butir soal secara jelas mengarah pada pencapaian tujuan pembelajaran yang ditargetkan.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Isi materi dalam soal dan kunci jawaban yang disediakan sudah akurat secara akademik.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Konteks/topik yang digunakan dalam soal relevan dengan materi ajar Informatika.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Tingkat kesulitan soal sudah sesuai untuk level siswa (Kelas X).	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Opsi jawaban (pengecoh) yang disediakan masuk akal dan efektif untuk menguji pemahaman.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Bahasa yang digunakan dalam soal jelas, tidak ambigu, dan mudah dipahami oleh siswa.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Stimulus yang digunakan (jika ada, seperti teks atau gambar) relevan dan efektif untuk mendukung pertanyaan.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Soal bebas dari bias (misalnya gender, suku, budaya dan tidak mengarah pada jawaban tertentu.)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Butir soal memiliki potensi untuk mendeteksi miskonsepsi atau kesulitan belajar spesifik siswa.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Catatan:

1. Nilai  $p$  (proporsi jawaban benar) yang diprediksi untuk Level 5 ke atas (0.2, 0.17, 0.15) menunjukkan tingkat kesulitan yang sangat tinggi. Meskipun bisa digunakan untuk membedakan siswa dengan kemampuan di atas rata-rata, perlu dipastikan apakah tingkat kesulitan ini sesuai dengan kurikulum dan kemampuan mayoritas siswa kelas X yang menjadi target. Jika tujuannya adalah tes diagnostik umum, soal-soal ini mungkin terlalu sulit. Namun, jika ini untuk seleksi atau evaluasi siswa peminatan khusus, tingkat kesulitan ini dapat diterima.
2. Pada Level 1, soal nomor 4 menggunakan istilah "Keputusan" untuk merujuk pada struktur kontrol. Istilah ini sebenarnya sudah benar, namun dalam praktik pengajaran di Indonesia, istilah "Percabangan" lebih umum digunakan. Perlu dipertimbangkan agar tidak membingungkan siswa yang terbiasa dengan terminologi lain.
3. Setelah validasi ahli, sangat disarankan agar instrumen ini diuji coba secara terbatas (pilot test) kepada sekelompok siswa yang setara dengan target penelitian yang akan memberikan data empiris mengenai tingkat kesulitan ( $p$ ) dan daya pembeda ( $d$ ) setiap butir soal, serta reliabilitas instrumen secara keseluruhan.

Mengetahui, 6 Oktober 2025



Dr. Gede Saindra Santyadiputra, S.T., M.Cs.



## Attachment 25 Results of Teacher Content Expert Test 1

## UJI AHLI ISI

Nama Lengkap : Komang Athi Wijaya, S.Kom, M.Pd.  
 Instansi/ Sekolah : SMA Kegeni 1 Blahbatuh  
 Mata Pelajaran yang Diampu : Informatika  
 Jabatan : Guru Mapel

## PETUNJUK:

Untuk setiap fungsi, jawab pertanyaan "☑" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "☒" jika tidak tercapai. Cantumkan detail observasi, fitur, atau saran pada "Catatan".

No	Pernyataan	Relevan	Tidak Relevan
		☑	☒
1	Materi dalam soal sesuai dengan indikator capaian pembelajaran pada modul ajar.	✓	
2	Butir soal secara jelas mengarah pada pencapaian tujuan pembelajaran yang ditargetkan.	✓	
3	Isi materi dalam soal dan kunci jawaban yang disediakan sudah akurat secara akademik.	✓	
4	Konteks/topik yang digunakan dalam soal relevan dengan materi ajar Informatika.	✓	
5	Tingkat kesulitan soal sudah sesuai untuk level siswa (Kelas X).	✓	
6	Opsi jawaban (pengecoh) yang disediakan masuk akal dan efektif untuk menguji pemahaman.	✓	
7	Bahasa yang digunakan dalam soal jelas, tidak ambigu, dan mudah dipahami oleh siswa.	✓	
8	Stimulus yang digunakan (jika ada, seperti teks atau gambar) relevan dan efektif untuk mendukung pertanyaan.	✓	
9	Soal bebas dari bias (misalnya gender, suku, budaya dan tidak mengarah pada jawaban tertentu.)	✓	
10	Butir soal memiliki potensi untuk mendeteksi miskonsepsi atau kesulitan belajar spesifik siswa.	✓	

Catatan:

.....

.....

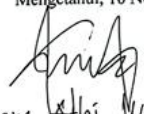
.....

.....

.....

.....

Mengetahui, 10 November 2025

  
 Komang Athi Wijaya, S.Kom, M.Pd.

## Attachment 26 Results of Teacher Content Expert Test 2

## UJI AHLI ISI

Nama Lengkap : Ni Kadet Dwyanti, S.Pd  
 Instansi/ Sekolah : SMAN 1 Blahbatuh  
 Mata Pelajaran yang Diampu : Informatika  
 Jabatan : Guru Ahli Pertama

## PETUNJUK:

Untuk setiap fungsi, jawab pertanyaan "☑" jika tujuan pengujian tercapai berdasarkan teknik pengujian yang dijelaskan. Berikan "☒" jika tidak tercapai. Cantumkan detail observasi, fitur, atau saran pada "Catatan".

No	Pernyataan	Relevan ☑	Tidak Relevan ☒
1	Materi dalam soal sesuai dengan indikator capaian pembelajaran pada modul ajar.	✓	
2	Butir soal secara jelas mengarah pada pencapaian tujuan pembelajaran yang ditargetkan.	✓	
3	Isi materi dalam soal dan kunci jawaban yang disediakan sudah akurat secara akademik.	✓	
4	Konteks/topik yang digunakan dalam soal relevan dengan materi ajar Informatika.	✓	
5	Tingkat kesulitan soal sudah sesuai untuk level siswa (Kelas X).	✓	
6	Opsi jawaban (pengecoh) yang disediakan masuk akal dan efektif untuk menguji pemahaman.	✓	
7	Bahasa yang digunakan dalam soal jelas, tidak ambigu, dan mudah dipahami oleh siswa.	✓	
8	Stimulus yang digunakan (jika ada, seperti teks atau gambar) relevan dan efektif untuk mendukung pertanyaan.	✓	
9	Soal bebas dari bias (misalnya gender, suku, budaya dan tidak mengarah pada jawaban tertentu.)	✓	
10	Butir soal memiliki potensi untuk mendeteksi miskonsepsi atau kesulitan belajar spesifik siswa.	✓	

## Catatan:

.....

.....

.....

.....

.....

.....

Mengetahui, 10 November 2025



(Ni Kadet Dwyanti, S.Pd.....)



Items							
1	2	3	4	5	6	7	8
2	1	3	1	1	5	6	6
4	4	5	5	5	5	4	5
7	7	7	7	7	7	7	7
6	7	6	7	6	6	7	7
7	7	7	7	7	7	7	7
7	7	7	7	7	7	7	7
6	7	6	7	6	5	6	6
6	6	6	7	6	7	6	6
6	6	5	4	5	5	5	4
7	7	7	7	7	7	7	7
7	7	7	7	7	7	7	7
5	6	5	6	5	6	6	6
7	7	7	7	7	7	4	2
7	7	7	7	7	7	7	6
7	7	7	7	7	6	7	6
7	7	7	7	7	6	5	7
4	4	5	6	4	4	3	4
7	6	7	7	7	7	7	7
7	7	7	7	7	7	7	7
6	6	6	6	5	4	5	4
7	7	7	7	7	7	7	7
7	7	7	7	7	7	7	7
7	6	6	7	7	7	7	6
7	6	7	7	7	6	6	7
5	4	7	7	7	7	7	7
5	6	6	7	6	7	7	5
4	4	4	5	5	4	4	4
7	7	7	7	7	6	6	5
7	7	7	7	7	7	7	7
6	6	5	6	6	7	7	7
7	7	6	5	5	7	5	4
5	5	5	5	5	6	4	5
7	7	7	7	7	7	7	7
5	5	4	5	5	5	5	5
7	7	7	7	7	7	7	7
6	7	6	6	6	6	6	6
6	6	6	7	7	7	7	7
7	6	7	6	7	6	5	7
3	4	4	7	6	6	5	5
7	6	7	7	6	6	7	7



-2	-3	-1	-3	-3	1	2	2
0	0	1	1	1	1	0	1
3	3	3	3	3	3	3	3
2	3	2	3	2	2	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
2	3	2	3	2	1	2	2
2	2	2	3	2	3	2	2
2	2	1	0	1	1	1	0
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
1	2	1	2	1	2	2	2
3	3	3	3	3	3	0	-2
3	3	3	3	3	3	3	2
3	3	3	3	3	2	3	2
3	3	3	3	3	2	1	3
0	0	1	2	0	0	-1	0
3	2	3	3	3	3	3	3
3	3	3	3	3	3	3	3
2	2	2	2	1	0	1	0
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	2	2	3	3	3	3	2
3	2	3	3	3	2	2	3
1	0	3	3	3	3	3	3
1	2	2	3	2	3	3	1
0	0	0	1	1	0	0	0
3	3	3	3	3	2	2	1
3	3	3	3	3	3	3	3
2	2	1	2	2	3	3	3
3	3	2	1	1	3	1	0
1	1	1	1	1	2	0	1
3	3	3	3	3	3	3	3
1	1	0	1	1	1	1	1
3	3	3	3	3	3	3	3
2	3	2	2	2	2	2	2
2	2	2	3	3	3	3	3
3	2	3	2	3	2	1	3
-1	0	0	3	2	2	1	1
3	2	3	3	2	2	3	3





**MODUL AJAR DEEP LEARNING**  
**MATA PELAJARAN : INFORMATIKA**  
**BAB 2 : ALGORITMA DAN PEMROGRAMAN**

**A. IDENTITAS MODUL**

<b>Nama Sekolah</b>	: SMA Negeri 1 Blahbatuh
<b>Nama Penyusun</b>	: Komang Adhi Wijaya,S.Kom.,M.Pd
<b>Mata Pelajaran</b>	: Informatika
<b>Kelas / Fase /Semester</b>	: X/ E / Genap
<b>Alokasi Waktu</b>	: 12 Jam Pelajaran (JP)
<b>Tahun Pelajaran</b>	: 2025 / 2026

**B. IDENTIFIKASI KESIAPAN PESERTA DIDIK**

- **Pengetahuan Awal:** Peserta didik diharapkan memiliki pemahaman dasar tentang berpikir komputasional (sebagaimana yang sudah dibahas pada bab sebelumnya). Ini meliputi kemampuan dekomposisi masalah, pengenalan pola, abstraksi, dan merumuskan algoritma sederhana dalam bahasa alami. Mereka mungkin sudah pernah berinteraksi dengan aplikasi atau game yang didasarkan pada logika pemrograman (misalnya, game puzzle logika, aplikasi visual programming seperti Scratch di tingkat SMP). Beberapa mungkin sudah memiliki pengalaman bermain dengan blok coding atau editor teks sederhana.
- **Minat:** Minat terhadap pemrograman sangat bervariasi. Pendekatan yang mengaitkan pemrograman dengan pembuatan aplikasi yang mereka gunakan sehari-hari (game sederhana, kalkulator, chatbot), atau pemecahan masalah yang relevan dengan hobi mereka, akan meningkatkan minat. Penggunaan platform pemrograman visual yang *user-friendly* seperti Scratch atau Blockly (untuk transisi ke teks) dapat membuat pengalaman awal lebih menyenangkan.
- **Latar Belakang:** Peserta didik berasal dari latar belakang yang beragam dalam hal akses dan paparan terhadap teknologi dan pemrograman. Beberapa mungkin sudah memiliki komputer pribadi dan koneksi internet yang stabil, sementara yang lain mungkin terbatas. Beberapa mungkin sudah pernah mencoba *coding* secara otodidak, sementara yang lain sama sekali belum. Perlu pendekatan berdiferensiasi untuk menjembatani kesenjangan ini.
- **Kebutuhan Belajar:**
  - **Visual:** Membutuhkan diagram alur (flowchart), contoh kode dengan *highlighting* sintaks, video tutorial langkah demi langkah, simulasi

eksekusi kode, atau visualisasi data struktur.

- **Auditori:** Membutuhkan penjelasan lisan yang jelas tentang konsep pemrograman, diskusi kelompok untuk memecahkan *bug*, atau mendengarkan penjelasan dari guru/teman.
- **Kinestetik:** Membutuhkan praktik langsung menulis kode, debugging, mencoba berbagai input, atau bahkan aktivitas *unplugged* untuk mensimulasikan kontrol aliran program.
- Beberapa peserta didik mungkin memerlukan bimbingan ekstra dalam memahami sintaks atau logika pemrograman yang kompleks, sementara yang lain mungkin siap untuk tantangan proyek yang lebih besar dan mandiri.

### C. KARAKTERISTIK MATERI PELAJARAN

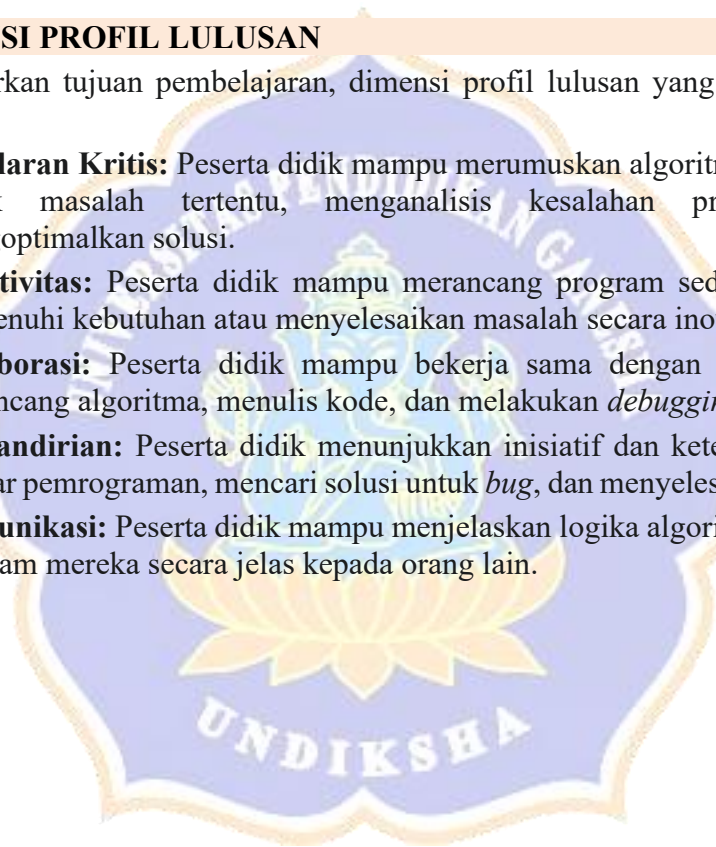
- **Jenis Pengetahuan yang Akan Dicapai:** Pengetahuan konseptual (definisi algoritma, pemrograman, bahasa pemrograman, variabel, tipe data, operator, struktur kontrol/kondisional, perulangan, fungsi), pengetahuan prosedural (menulis algoritma dalam pseudocode/flowchart, menulis kode program sederhana, melakukan debugging, menguji program), dan pengetahuan metakognitif (merencanakan solusi sebelum coding, merefleksikan efisiensi kode, memilih struktur kontrol yang tepat untuk masalah tertentu).
- **Relevansi dengan Kehidupan Nyata Peserta Didik:** Materi ini sangat relevan dan fundamental di era digital. Pemrograman adalah bahasa yang menggerakkan sebagian besar teknologi yang mereka gunakan (ponsel, aplikasi, game, website). Belajar pemrograman mengembangkan kemampuan problem-solving, logika, kreativitas, dan ketelitian yang bermanfaat di berbagai bidang.
- **Tingkat Kesulitan:** Tingkat kesulitan materi ini bersifat moderat hingga tinggi. Memahami konsep dasar variabel dan urutan instruksi relatif mudah. Namun, memahami struktur kontrol yang kompleks (if-else bersarang, perulangan bertingkat), menemukan dan memperbaiki *bug*, serta menulis program yang efisien untuk masalah yang lebih besar memerlukan ketelitian, penalaran logis yang kuat, dan banyak latihan.
- **Struktur Materi:** Materi diawali dengan pengantar dari algoritma ke pemrograman, pengenalan bahasa pemrograman sederhana (disarankan Python atau Blockly/Scratch sebagai jembatan), konsep dasar (variabel, tipe data, operator), struktur kontrol (kondisional dan perulangan), hingga pengenalan fungsi dan *debugging* sederhana. Setiap konsep dibangun secara bertahap.
- **Integrasi Nilai dan Karakter:**
  - **Penalaran Kritis:** Mengembangkan kemampuan menganalisis masalah, merancang solusi logis (algoritma), dan menemukan serta memperbaiki *bug* (kesalahan) dalam program.
  - **Kreativitas:** Mendorong peserta didik untuk merancang program yang inovatif, menemukan berbagai cara untuk memecahkan masalah, dan mengimplementasikan ide-ide baru.

- **Kolaborasi:** Melatih kemampuan bekerja sama dalam kelompok untuk merancang algoritma, menulis kode, dan melakukan *debugging*.
- **Kemandirian:** Mendorong ketekunan dan kesabaran dalam menghadapi *bug* dan tantangan pemrograman, serta mencari solusi secara mandiri.
- **Ketelitian:** Menekankan pentingnya sintaks yang benar dan logika yang presisi dalam pemrograman.
- **Inovatif:** Memicu minat untuk menciptakan sesuatu yang baru atau meningkatkan yang sudah ada melalui kode.
- **Komunikasi:** Melatih kemampuan menjelaskan logika program dan berkolaborasi dalam tim.

## D DIMENSI PROFIL LULUSAN

Berdasarkan tujuan pembelajaran, dimensi profil lulusan yang akan dicapai adalah:

- **Penalaran Kritis:** Peserta didik mampu merumuskan algoritma yang tepat untuk masalah tertentu, menganalisis kesalahan program, dan mengoptimalkan solusi.
- **Kreativitas:** Peserta didik mampu merancang program sederhana untuk memenuhi kebutuhan atau menyelesaikan masalah secara inovatif.
- **Kolaborasi:** Peserta didik mampu bekerja sama dengan teman dalam merancang algoritma, menulis kode, dan melakukan *debugging* bersama.
- **Kemandirian:** Peserta didik menunjukkan inisiatif dan ketekunan dalam belajar pemrograman, mencari solusi untuk *bug*, dan menyelesaikan proyek.
- **Komunikasi:** Peserta didik mampu menjelaskan logika algoritma dan kode program mereka secara jelas kepada orang lain.



## DESAIN PEMBELAJARAN

### A. CAPAIAN PEMBELAJARAN (CP) NOMOR : 32 TAHUN 2024

Pada akhir Fase E, peserta didik mampu menerapkan proses berpikir efektif dan efisien untuk menyelesaikan persoalan secara algoritmik sebagai solusi atas rancangan instruksi dan data yang dapat dijalankan secara efektif dan efisien oleh sistem komputasi, menerapkan berpikir kritis dalam menyikapi beragam data yang tersedia di internet untuk menjadi informasi yang bermanfaat, mempunyai wawasan tentang profesi informatika, serta memahami hak dan kewajiban sebagai warga digital dan aspek hukumnya. Capaian Pembelajaran setiap elemen adalah sebagai berikut.

Elemen	Capaian Pembelajaran
Berpikir Komputasional	<p>Peserta didik mampu memahami validitas sumber data; memahami konsep struktur data dan algoritma standar; menerapkan proses komputasi yang dilakukan manusia secara mandiri atau berkelompok untuk mendapatkan data yang bersih, benar, dan terpercaya; menerapkan struktur data dan algoritma standar untuk menghasilkan berbagai solusi dalam menyelesaikan persoalan yang mengandung himpunan data berstruktur kompleks dengan volume tidak kecil; serta menuliskan solusi rancangan program sederhana dalam format <i>pseudocode</i> yang dekat dengan bahasa komputer.</p> <p>Peserta didik mampu memahami model dan menyimulasikan dinamika Input-Proses-Output dalam sebuah komputer <i>Von Neumann</i>, serta memahami peran sistem operasi.</p>
Literasi Digital	<p>Peserta didik mampu memahami penggunaan mesin pencari dengan variabel yang lebih banyak; mengetahui ekosistem periksa fakta untuk memilah fakta dan bukan; menggunakan cara membaca lateral untuk mengevaluasi berbagai informasi digital; memahami pemanfaatan lebih beragam perkakas teknologi digital untuk membuat laporan, presentasi, serta analisis dan interpretasi data; memahami konsep dan penerapan serta konfigurasi keamanan dasar untuk konektivitas jaringan data lokal dan internet baik kabel maupun nirkabel; serta memahami pemanfaatan media digital untuk produksi dan diseminasi konten, partisipasi dan kolaborasi.</p> <p>Peserta didik mampu menghargai hak atas kekayaan intelektual, mengenal profesi bidang Informatika, memahami penerapan digitalisasi budaya Indonesia, menyaring konten negatif di dunia digital, menerapkan pengelolaan kata sandi dengan manajer kata sandi, dan menerapkan autentikasi dua langkah secara sederhana, serta menerapkan konfigurasi privasi dan keamanan pada akun platform digital.</p>

## B. LINTAS DISIPLIN ILMU

- **Matematika:** Logika, aljabar, diskrit, fungsi, bilangan. Banyak masalah pemrograman yang memerlukan pemahaman matematis.
- **Bahasa Inggris:** Banyak istilah dan sintaks dalam pemrograman berasal dari Bahasa Inggris.
- **Fisika/Kimia/Biologi:** Pemodelan simulasi, analisis data, visualisasi hasil eksperimen.
- **Desain Grafis/Seni:** Membuat aplikasi dengan antarmuka pengguna yang menarik, game, atau visualisasi.
- **Ekonomi/Bisnis:** Pembuatan aplikasi keuangan sederhana, analisis data penjualan, otomatisasi tugas.

## C. TUJUAN PEMBELAJARAN

### Pertemuan 1: Pengantar Algoritma dan Pemrograman (Struktur Dasar)

Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:

- Menjelaskan perbedaan antara algoritma dan program komputer dengan tepat.
- Menyusun algoritma sederhana dalam bentuk pseudocode atau flowchart untuk masalah sehari-hari.
- Mengenal lingkungan pengembangan (IDE/editor online) untuk bahasa pemrograman yang digunakan (misal: Python).
- Menulis program "Hello World!" dan program sederhana untuk mencetak teks atau melakukan operasi aritmatika dasar.

### Pertemuan 2: Variabel, Tipe Data, dan Operator

Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:

- Mendefinisikan variabel dan menjelaskan pentingnya dalam pemrograman.
- Mengidentifikasi berbagai tipe data (integer, float, string, boolean) dalam bahasa pemrograman yang digunakan.
- Menggunakan operator aritmatika, perbandingan, dan logika dalam ekspresi program.
- Menulis program sederhana yang melibatkan input dari pengguna, pemrosesan variabel, dan menampilkan output.

### Pertemuan 3: Struktur Kontrol: Kondisional (If-Else)

Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:

- Menjelaskan konsep percabangan (kondisional) dalam algoritma dan program.
- Menggunakan pernyataan if, elif (atau else if), dan else untuk membuat program yang mengambil keputusan berdasarkan kondisi.
- Menyelesaikan masalah yang memerlukan pemilihan tindakan berdasarkan kriteria tertentu (misal: menentukan kelulusan, diskon harga).

### Pertemuan 4: Struktur Kontrol: Perulangan (Looping)

Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:

- Menjelaskan konsep perulangan dalam algoritma dan program.
- Menggunakan pernyataan `for` dan `while` untuk melakukan iterasi dalam program.
- Menyelesaikan masalah yang memerlukan pengulangan tindakan sejumlah kali atau sampai kondisi tertentu terpenuhi (misal: menghitung jumlah deret, mencetak pola).

Pertemuan 5: Fungsi dan Modularitas Program

Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:

- Mendefinisikan fungsi sebagai blok kode yang dapat digunakan kembali.
- Membuat dan memanggil fungsi sederhana dalam program untuk memecah masalah menjadi bagian yang lebih kecil.
- Menjelaskan manfaat modularitas dalam pemrograman (keterbacaan, reusable, debugging).
- Melakukan *debugging* dasar untuk menemukan dan memperbaiki kesalahan dalam kode program sederhana.

Pertemuan 6: Proyek Pemrograman Sederhana dan Debugging

Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:

- Menerapkan konsep algoritma dan pemrograman yang telah dipelajari untuk membuat proyek program sederhana (misal: kalkulator, game tebak angka, aplikasi *to-do list* sederhana).
- Menganalisis dan memperbaiki *bug* dalam program yang lebih kompleks secara mandiri atau kolaboratif.
- Mempresentasikan hasil proyek program dan menjelaskan logika di baliknya.

#### D. TOPIK PEMBELAJARAN KONTEKSTUAL

- **Pencetakan Teks/Informasi:** Membuat program yang mencetak informasi pribadi, ucapan selamat, atau lirik lagu sederhana.
- **Kalkulator Sederhana:** Program untuk penjumlahan, pengurangan, perkalian, pembagian.
- **Konversi Satuan:** Program konversi suhu (Celcius ke Fahrenheit), panjang (meter ke kilometer).
- **Penentuan Kriteria:** Program untuk menentukan apakah seseorang lulus/tidak, diskon harga, status cuaca (hujan/tidak).
- **Perulangan/Pola:** Program untuk mencetak deret angka, pola bintang, atau menghitung akumulasi (misalnya, total belanja).
- **Game Sederhana:** Game tebak angka, permainan "batu-gunting-kertas" sederhana.
- **Aplikasi Mini:** Aplikasi untuk mengelola daftar belanja, daftar tugas sederhana, atau catatan digital.

#### E. KERANGKA PEMBELAJARAN

## 1. PRAKTIK PEDAGOGIK:

- **Model Pembelajaran:** *Discovery Learning* (untuk menemukan cara kerja sintaks/konsep), *Problem-Based Learning* (PBL) untuk menyelesaikan masalah dengan kode, dan *Project-Based Learning* (PjBL) untuk membuat aplikasi sederhana.
- **Strategi Pembelajaran:**
  - **Mindful Learning:** Fokus pada setiap baris kode, latihan *tracing* manual eksekusi kode, sesi *debugging* yang sistematis, refleksi terhadap *bug* sebagai peluang belajar.
  - **Meaningful Learning:** Mengaitkan setiap konsep pemrograman dengan contoh aplikasi nyata (misal: mengapa kita butuh perulangan di kalkulator?), studi kasus pembuatan program sederhana, dan mendorong peserta didik untuk membuat program yang relevan dengan minat mereka.
  - **Joyful Learning:** Tantangan *coding* yang kompetitif (misal: siapa tercepat menyelesaikan *puzzle coding*), permainan *debugging* interaktif, membuat game sederhana, penggunaan platform visual yang menyenangkan, sesi *pair programming*.
- **Metode Pembelajaran:** Demonstrasi langsung (live coding), latihan terstruktur, *pair programming*, *code review*, diskusi kelompok, *debugging* bersama, proyek pemrograman.

## 2. KEMITRAAN PEMBELAJARAN:

- **Lingkungan Sekolah:** Guru TIK/komputer sebagai fasilitator, pengelola lab komputer. Anggota klub pemrograman atau ekstrakurikuler IT sebagai mentor sebaya.
- **Lingkungan Luar Sekolah:** Narasumber (programmer, *software developer*, desainer game) untuk berbagi pengalaman dan inspirasi (melalui daring atau kunjungan singkat). Komunitas *online coding* sebagai sumber belajar dan dukungan.
- **Masyarakat:** Mengidentifikasi masalah sehari-hari di lingkungan sekitar yang berpotensi diselesaikan dengan program sederhana.

## 3. LINGKUNGAN BELAJAR:

- **Ruang Fisik:** Laboratorium komputer dengan perangkat yang memadai dan koneksi internet stabil. Jika tidak ada, laptop/tablet pribadi peserta didik yang dapat diakses. Meja yang cukup luas untuk kerja kelompok. Proyektor untuk demonstrasi guru.
- **Ruang Virtual:** Platform Google Classroom untuk berbagi materi (slide, link IDE online, video tutorial), penugasan *coding*, dan pengumpulan proyek. IDE online (misal: Replit, Google Colab untuk Python) atau platform blok visual (Scratch, Blockly) sebagai lingkungan coding. Forum diskusi daring untuk tanya jawab dan *peer-support*.
- **Budaya Belajar:** Lingkungan yang mendukung eksperimen dan mencoba hal baru. Mendorong peserta didik untuk tidak takut membuat kesalahan (*bug*), melainkan melihatnya sebagai bagian alami dari proses belajar. Menekankan kolaborasi dan berbagi pengetahuan.

#### 4. PEMANFAATAN DIGITAL:

- **Perpustakaan Digital:** Mengakses tutorial pemrograman online, dokumentasi bahasa pemrograman, e-book tentang algoritma dan pemrograman.
- **Forum Diskusi Daring:** Google Classroom atau platform lain untuk memposting *bug* yang ditemui, meminta bantuan, atau berbagi ide kode.
- **Penilaian Daring:** Kuis sintaks atau konsep dasar melalui Kahoot/Quizizz. Sistem *auto-grader* (jika tersedia) untuk tugas *coding* sederhana. Pengumpulan file program melalui Google Classroom.
- **Alat Interaktif:**
  - **IDE Online/Platform Coding:** Replit, Google Colab (Python), Scratch, Blockly (untuk pemula) sebagai lingkungan untuk menulis dan menjalankan kode.
  - **Situs Tutorial Interaktif:** Codecademy, freeCodeCamp, W3Schools untuk belajar sintaks dan konsep dasar secara interaktif.
  - **YouTube:** Menonton video tutorial pemrograman dan *live coding*.

#### F. LANGKAH-LANGKAH PEMBELAJARAN BERDIFERENSIASI

##### PERTEMUAN 1: PENGANTAR ALGORITMA DAN PEMROGRAMAN (STRUKTUR DASAR)

##### KEGIATAN PENDAHULUAN (15 MENIT)

- **Mindful Learning:** Guru memulai dengan pertanyaan: "Apa yang membuat sebuah aplikasi atau game bisa bekerja?" "Bagaimana komputer bisa mengerti perintah kita?" Guru meminta peserta didik untuk fokus pada ide "memberikan instruksi" kepada mesin.
- **Joyful Learning:** Guru menampilkan video singkat tentang sebuah robot yang melakukan tugas sederhana (misalnya, robot penyapu lantai) dan bertanya: "Menurut kalian, bagaimana cara robot ini tahu apa yang harus dilakukan?" Ini memicu rasa ingin tahu.
- **Meaningful Learning:** Guru mengaitkan ide instruksi ini dengan konsep "algoritma" dan "pemrograman" sebagai cara kita "berbicara" dengan komputer. Guru menyampaikan tujuan pembelajaran hari ini dan memperkenalkan bahasa pemrograman yang akan digunakan (misalnya Python sebagai bahasa teks pertama yang ramah pemula, atau Scratch/Blockly sebagai jembatan).

##### KEGIATAN INTI (60 MENIT)

- **Memahami (20 menit):**
  - Guru menjelaskan perbedaan antara algoritma (langkah-langkah umum) dan program (implementasi spesifik dalam bahasa komputer). Guru menampilkan contoh pseudocode dan flowchart sederhana untuk masalah "membuat kopi".
  - Guru mendemonstrasikan lingkungan pemrograman yang akan digunakan (IDE online) dan menunjukkan cara menulis serta

- menjalankan program "Hello World!".
- **Mengaplikasi (25 menit):**
    - Peserta didik secara individu atau berpasangan diminta untuk menulis program "Hello World!" pertama mereka di IDE online. (Diferensiasi proses: peserta didik yang lebih cepat dapat diminta untuk memodifikasi pesan atau menambahkan lebih banyak baris cetak).
    - Guru memberikan masalah sederhana (misalnya, "Cetak namamu 5 kali", "Cetak hasil penjumlahan  $2 + 3$ "). Peserta didik berlatih menulis program untuk masalah ini.
  - **Merefleksi (15 menit):**
    - Guru meminta peserta didik menuliskan di kolom komentar Google Classroom: "Apa perasaanmu saat berhasil menjalankan program pertamamu?" dan "Apa tantangan terbesar saat menulis program 'Hello World!'?"
    - Beberapa peserta didik diminta untuk membagikan pengalaman mereka.

#### KEGIATAN PENUTUP (15 MENIT)

- **Umpan Balik Konstruktif:** Guru memberikan apresiasi atas keberanian peserta didik dalam menulis program pertama. Guru mengulas kembali konsep dasar dan potensi kesalahan umum (misalnya, *typo*).
- **Menyimpulkan Pembelajaran:** Peserta didik secara kolaboratif menyimpulkan bahwa pemrograman adalah cara kita memberikan instruksi kepada komputer, dimulai dari algoritma yang jelas.
- **Perencanaan Pembelajaran Selanjutnya:** Guru memberikan pengantar untuk pertemuan berikutnya (variabel, tipe data, operator) dan memberikan tugas eksplorasi (mencari tahu berbagai jenis informasi yang bisa disimpan komputer).

### PERTEMUAN 2: VARIABEL, TIPE DATA, DAN OPERATOR

#### KEGIATAN PENDAHULUAN (15 MENIT)

- **Mindful Learning:** Guru bertanya: "Bagaimana komputer bisa 'mengingat' nama kita atau angka yang kita masukkan?" Guru meminta peserta didik untuk fokus pada ide "menyimpan informasi".
- **Joyful Learning:** Guru dapat memainkan permainan "Simpan Informasi Rahasia" di mana setiap peserta didik memiliki "kotak" (variabel) untuk menyimpan informasi tertentu, dan mereka harus bertukar "kotak" sesuai aturan.
- **Meaningful Learning:** Guru mengaitkan aktivitas ini dengan konsep "variabel" dan "tipe data" dalam pemrograman, serta bagaimana komputer memanipulasi informasi tersebut dengan "operator". Guru menyampaikan tujuan pembelajaran.

#### KEGIATAN INTI (60 MENIT)

- **Memahami (20 menit):**
  - Guru menjelaskan konsep variabel (sebagai wadah data) dan aturan penamaan variabel.
  - Guru memperkenalkan berbagai tipe data (integer, float, string, boolean) dengan contoh penggunaannya.
  - Guru menjelaskan berbagai jenis operator (aritmatika, perbandingan, logika) dan bagaimana mereka digunakan dalam ekspresi.
- **Mengaplikasi (25 menit):**
  - Peserta didik dalam kelompok kecil (diferensiasi proses: beberapa kelompok mendapatkan soal konversi tipe data yang lebih sederhana, yang lain dengan ekspresi logika yang lebih kompleks) mengerjakan latihan *coding* yang melibatkan:
    - Deklarasi dan inisialisasi variabel.
    - Melakukan operasi aritmatika dan mencetak hasilnya.
    - Mengambil input dari pengguna dan menyimpannya dalam variabel.
    - Membuat ekspresi perbandingan atau logika sederhana.
  - Guru berkeliling memberikan bantuan dan tips.
- **Merefleksi (15 menit):**
  - Guru meminta peserta didik menuliskan di jurnal: "Apa yang paling membingungkan tentang variabel dan tipe data?" dan "Bagaimana saya bisa menggunakan operator untuk membuat kalkulator sederhana?"

#### **KEGIATAN PENUTUP (15 MENIT)**

- **Umpan Balik Konstruktif:** Guru memberikan umpan balik terhadap penggunaan variabel, tipe data, dan operator. Mengoreksi miskonsepsi umum.
- **Menyimpulkan Pembelajaran:** Guru dan peserta didik bersama-sama menyimpulkan pentingnya variabel untuk menyimpan data dan operator untuk memanipulasi data tersebut.
- **Perencanaan Pembelajaran Selanjutnya:** Guru memberikan pengantar untuk pertemuan berikutnya (struktur kondisional) dan tugas latihan menulis program yang mengambil input dan melakukan perhitungan sederhana.

### **PERTEMUAN 3: STRUKTUR KONTROL: KONDISIONAL (IF-ELSE)**

#### **KEGIATAN PENDAHULUAN (15 MENIT)**

- **Mindful Learning:** Guru bertanya: "Bagaimana kita membuat keputusan dalam hidup? (misalnya, jika hujan, maka bawa payung). Bagaimana komputer bisa membuat keputusan serupa?" Guru meminta peserta didik untuk fokus pada ide "membuat pilihan".
- **Joyful Learning:** Guru dapat memulai dengan permainan "Siapa yang Berhak Lewat?" di mana peserta didik harus mengikuti aturan if-else (misalnya, jika siswa mengenakan baju biru, dia boleh lewat; jika tidak,

dia harus duduk).

- **Meaningful Learning:** Guru mengaitkan aktivitas ini dengan konsep "kondisional" atau "percabangan" dalam pemrograman. Guru menyampaikan tujuan pembelajaran.

#### KEGIATAN INTI (60 MENIT)

- **Memahami (20 menit):**
  - Guru menjelaskan konsep struktur kondisional (if, elif/else if, else) dengan contoh diagram alir dan kode.
  - Guru mendemonstrasikan bagaimana program dapat menjalankan blok kode yang berbeda berdasarkan suatu kondisi.
- **Mengaplikasi (25 menit):**
  - Peserta didik dalam kelompok (diferensiasi konten: beberapa kelompok mengerjakan soal penentuan diskon sederhana, yang lain dengan kondisi bersarang/lebih kompleks) mengerjakan latihan *coding* yang melibatkan:
    - Program penentu kelulusan berdasarkan nilai.
    - Program penentu diskon berdasarkan total belanja.
    - Program penentu jenis bilangan (positif, negatif, nol).
  - Mereka didorong untuk menguji berbagai kasus input untuk memastikan program berjalan benar.
- **Merefleksi (15 menit):**
  - Guru meminta peserta didik merefleksikan: "Apa yang paling menantang dari menulis kondisi dalam program?" dan "Berikan satu contoh masalah sehari-hari yang akan lebih mudah dipecahkan dengan pernyataan if-else."

#### KEGIATAN PENUTUP (15 MENIT)

- **Umpan Balik Konstruktif:** Guru memberikan umpan balik terhadap logika kondisional peserta didik. Mengoreksi kesalahan umum dalam penulisan kondisi atau indentasi (jika menggunakan Python).
- **Menyimpulkan Pembelajaran:** Guru dan peserta didik bersama-sama menyimpulkan bahwa kondisional memungkinkan program untuk membuat keputusan.
- **Perencanaan Pembelajaran Selanjutnya:** Guru memberikan pengantar untuk pertemuan berikutnya (struktur perulangan) dan tugas latihan membuat program yang menerapkan kondisional untuk masalah sederhana (misalnya, detektor angka ganjil/genap).

### PERTEMUAN 4: STRUKTUR KONTROL: PERULANGAN (LOOPING)

#### KEGIATAN PENDAHULUAN (15 MENIT)

- **Mindful Learning:** Guru bertanya: "Bayangkan jika kita harus mencetak

'Hello!' 1000 kali. Apakah kita akan menuliskannya 1000 kali? Bagaimana cara yang lebih efisien?" Guru meminta peserta didik untuk fokus pada ide "pengulangan efisien".

- **Joyful Learning:** Guru dapat memulai dengan demonstrasi fisik sederhana yang berulang (misalnya, melompat 5 kali, tepuk tangan 3 kali) dan bertanya: "Bagaimana saya bisa memberitahu komputer untuk melakukan ini?"
- **Meaningful Learning:** Guru mengaitkan aktivitas ini dengan konsep "perulangan" atau "looping" dalam pemrograman. Guru menyampaikan tujuan pembelajaran.

### KEGIATAN INTI (60 MENIT)

- **Memahami (20 menit):**
  - Guru menjelaskan konsep perulangan (for loop dan while loop) dengan contoh diagram alir dan kode. Guru menjelaskan kapan menggunakan for (jumlah iterasi diketahui) dan while (jumlah iterasi tidak diketahui/berdasarkan kondisi).
  - Guru mendemonstrasikan bagaimana perulangan dapat mengotomatisasi tugas yang berulang.
- **Mengaplikasi (25 menit):**
  - Peserta didik dalam kelompok (diferensiasi konten: beberapa kelompok mengerjakan soal deret sederhana, yang lain dengan pola perulangan bertingkat atau penggunaan break/continue) mengerjakan latihan *coding* yang melibatkan:
    - Mencetak angka dari 1 sampai 10.
    - Menghitung jumlah deret.
    - Mencetak pola bintang atau angka.
    - Program tebak angka sederhana (menggunakan while loop).
  - Mereka didorong untuk memvisualisasikan bagaimana *loop* bekerja.
- **Merefleksi (15 menit):**
  - Guru meminta peserta didik merefleksikan: "Kapan saya harus menggunakan for loop dan kapan while loop?" dan "Apa manfaat terbesar dari perulangan dalam pemrograman?"

### KEGIATAN PENUTUP (15 MENIT)

- **Umpan Balik Konstruktif:** Guru memberikan umpan balik terhadap penggunaan perulangan. Mengoreksi kesalahan umum seperti *infinite loop* atau kondisi terminasi yang salah.
- **Menyimpulkan Pembelajaran:** Guru dan peserta didik bersama-sama menyimpulkan bahwa perulangan adalah alat yang ampuh untuk mengotomatisasi tugas berulang.
- **Perencanaan Pembelajaran Selanjutnya:** Guru memberikan pengantar untuk pertemuan berikutnya (fungsi dan modularitas) dan tugas latihan membuat program yang menggunakan perulangan.

## PERTEMUAN 5: FUNGSI DAN MODULARITAS PROGRAM

### KEGIATAN PENDAHULUAN (15 MENIT)

- **Mindful Learning:** Guru bertanya: "Ketika kita memasak, kita sering menggunakan resep yang sama untuk bagian tertentu (misalnya, membuat saus). Bagaimana kita bisa menerapkan ide ini dalam pemrograman agar kode kita rapi dan mudah digunakan kembali?" Guru meminta peserta didik untuk fokus pada ide "mengorganisir kode".
- **Joyful Learning:** Guru dapat mempresentasikan sebuah program yang sangat panjang dan sulit dibaca, lalu menampilkan versi yang sama tetapi dengan fungsi yang terstruktur, dan bertanya: "Mana yang lebih mudah dipahami?"
- **Meaningful Learning:** Guru mengaitkan ide "reusable code" ini dengan konsep "fungsi" dan "modularitas" dalam pemrograman. Guru menyampaikan tujuan pembelajaran.

### KEGIATAN INTI (60 MENIT)

- **Memahami (20 menit):**
  - Guru menjelaskan konsep fungsi (mendefinisikan dan memanggil). Guru menjelaskan manfaat modularitas (kode lebih rapi, mudah dibaca, mudah di-*debug*, *reusable*).
  - Guru mendemonstrasikan cara membuat fungsi sederhana (misalnya, fungsi untuk menjumlahkan dua angka, fungsi untuk mencetak pesan sapaan).
  - Guru memperkenalkan teknik *debugging* dasar (misal: menggunakan print statement, membaca pesan error).
- **Mengaplikasi (25 menit):**
  - Peserta didik dalam kelompok (diferensiasi proses: beberapa kelompok membuat fungsi sederhana tanpa parameter, yang lain dengan parameter dan nilai kembalian, atau bahkan fungsi bersarang) mengerjakan latihan *coding* yang melibatkan:
    - Membuat program yang menggunakan fungsi untuk setiap operasi (input, proses, output).
    - Membagi program sebelumnya menjadi beberapa fungsi.
    - Melakukan *debugging* pada kode yang sengaja disisipi *bug* sederhana oleh guru.
  - Mereka didorong untuk berdiskusi dan membantu satu sama lain dalam *debugging*.
- **Merefleksi (15 menit):**
  - Guru meminta peserta didik merefleksikan: "Bagaimana fungsi membuat program saya lebih mudah dipahami?" dan "Apa tips terbaik saya untuk menemukan *bug*?"

### KEGIATAN PENUTUP (15 MENIT)

- **Umpan Balik Konstruktif:** Guru memberikan umpan balik terhadap

penggunaan fungsi dan kemampuan *debugging*.

- **Menyimpulkan Pembelajaran:** Guru dan peserta didik bersama-sama menyimpulkan bahwa fungsi dan modularitas sangat penting untuk membuat program yang baik dan mudah dikelola.
- **Perencanaan Pembelajaran Selanjutnya:** Guru memberitahu bahwa pertemuan terakhir adalah proyek akhir dan persiapan presentasi.

## PERTEMUAN 6: PROYEK PEMROGRAMAN SEDERHANA DAN DEBUGGING

### KEGIATAN PENDAHULUAN (15 MENIT)

- **Mindful Learning:** Guru bertanya: "Setelah belajar semua konsep ini, aplikasi apa yang paling ingin kalian buat?" Guru meminta peserta didik untuk fokus pada ide "menciptakan sesuatu".
- **Joyful Learning:** Guru menampilkan beberapa contoh proyek sederhana yang dapat dibuat oleh pemula (misalnya, kalkulator BMI, game tebak angka dengan *feedback*, program penghitung rata-rata nilai).
- **Meaningful Learning:** Guru menekankan bahwa tujuan akhir dari belajar pemrograman adalah untuk menciptakan solusi. Guru menyampaikan instruksi proyek dan rubrik penilaian.

### KEGIATAN INTI (60 MENIT)

- **Memahami (20 menit):**
  - Guru menjelaskan secara rinci tugas proyek: setiap kelompok (atau individu, tergantung kebutuhan) memilih satu masalah sederhana dan merancang programnya menggunakan konsep yang telah dipelajari (variabel, kondisional, perulangan, fungsi).
  - Mereka harus membuat perencanaan (algoritma/flowchart) dan kemudian menulis kode programnya.
  - Guru memberikan daftar masalah yang dapat dipilih atau siswa dapat mengusulkan masalah sendiri.
- **Mengaplikasi (25 menit):**
  - Peserta didik bekerja secara kolaboratif (jika berkelompok) atau mandiri untuk mengembangkan proyek mereka. Guru berperan sebagai fasilitator dan *debugger* ahli, memberikan bimbingan sesuai kebutuhan (diferensiasi dukungan).
  - Mereka harus menguji program mereka secara menyeluruh dan melakukan *debugging* jika ada kesalahan.
- **Merefleksi (15 menit):**
  - Setiap kelompok/individu mempresentasikan proyek mereka, mendemonstrasikan programnya, dan menjelaskan logika di baliknya.
  - Guru memfasilitasi diskusi refleksi akhir: "Apa tantangan terbesar saat membuat proyek ini dan bagaimana kalian mengatasinya?" "Apa yang paling kalian banggakan dari proyek ini?"

### KEGIATAN PENUTUP (15 MENIT)

- **Umpan Balik Konstruktif:** Guru memberikan umpan balik individu dan kelompok terhadap proyek. Guru menyoroti kekuatan program (misalnya, logika yang efisien, UI yang baik, kode yang bersih) dan area yang masih bisa ditingkatkan.
- **Menyimpulkan Pembelajaran:** Guru dan peserta didik bersama-sama menyimpulkan bahwa pemrograman adalah keterampilan yang kuat untuk memecahkan masalah dan menciptakan inovasi.
- **Perencanaan Pembelajaran Selanjutnya:** Guru mendorong peserta didik untuk terus berlatih dan menjelajahi dunia pemrograman lebih jauh (misalnya, mengikuti *bootcamp* online, bergabung dengan komunitas *coding*).

### G. ASESMEN PEMBELAJARAN

Asesmen akan dilakukan secara komprehensif untuk mengukur pencapaian kompetensi peserta didik.

#### ASSESSMENT AS LEARNING (AS):

- **Self-Assessment:** Jurnal refleksi setelah setiap sesi *coding* tentang pemahaman konsep, *bug* yang ditemui dan cara mengatasinya, serta strategi belajar pemrograman yang efektif.
- **Peer Assessment:** Peserta didik saling melakukan *code review* terhadap program teman, memberikan masukan konstruktif, dan menilai partisipasi dalam kelompok.
- **Diskusi Kode:** Observasi partisipasi aktif dalam diskusi tentang logika kode, *debugging*, dan berbagi solusi.

#### ASSESSMENT FOR LEARNING (AFL):

- **Kuis Singkat/Tanya Jawab Lisan:** Menguji pemahaman sintaks dasar, definisi variabel, fungsi operator, atau identifikasi tipe data (menggunakan Kahoot/Quizizz).
- **Latihan Coding Harian/Mingguan:** Memberikan tugas *coding* singkat sebagai latihan untuk mengidentifikasi pemahaman dan kesulitan siswa secara formatif, serta memberikan umpan balik langsung.
- **Umpan Balik Guru:** Memberikan umpan balik langsung saat *live coding*, saat *debugging* bersama, atau saat mengoreksi program.
- **Observasi:** Guru mengamati kemampuan peserta didik dalam merancang algoritma, menulis kode, mengidentifikasi *bug*, dan menerapkan konsep pemrograman.

#### ASSESSMENT OF LEARNING (AOL):

- **Tes Tertulis:** Soal-soal yang mencakup:
  - Menulis pseudocode atau flowchart untuk masalah tertentu.
  - Menganalisis potongan kode dan memprediksi outputnya.
  - Mengidentifikasi kesalahan (bug) dalam kode program.
  - Menjelaskan konsep pemrograman (variabel, tipe data, kondisional, perulangan, fungsi).
- **Penilaian Kinerja/Proyek Pemrograman:**
  - **Proyek Akhir:** Penilaian terhadap program yang dihasilkan (berfungsi dengan benar, memenuhi spesifikasi, kode bersih/terbaca, efisien) dan presentasinya (kemampuan menjelaskan logika, proses *debugging*, dan interpretasi hasil). Rubrik penilaian proyek mencakup aspek ketepatan, fungsionalitas, keterbacaan kode, dan kreativitas (sesuai dengan kriteria di buku ajar pada halaman 128).
  - **Tantangan *Debugging*:** Peserta didik diberikan program dengan *bug* dan diminta untuk menemukan serta memperbaikinya dalam waktu tertentu. Penilaian didasarkan pada ketepatan dan efisiensi perbaikan.
- **Portofolio Digital:** Kumpulan semua program yang telah dibuat, laporan proyek, dan jurnal refleksi yang menunjukkan perkembangan kompetensi peserta didik.

Mengetahui  
Kepala SMA Negeri 1 Blahbatuh,

Gianyar, 21 Juli 2025  
Guru Mata Pelajaran,

I Ketut Sulatra, S.Pd., M.Pd.  
S.Kom., M.Pd.

NIP. 19700203 199702 1 004

Komang Adhi Wijaya,

NIP. 19820814 200903 1 006



Attachment 31 Documentation





