

BAB 4

HASIL DAN PEMBAHASAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini menghasilkan sebuah sistem informasi manajemen peserta PKL yang dikembangkan untuk mengatasi permasalahan administratif di Balai Tekkomdik DIY. Sistem dikembangkan menggunakan metode *Agile Scrum* agar dapat beradaptasi dengan kebutuhan pengguna secara iteratif. Proses pengumpulan kebutuhan dilakukan melalui wawancara dengan staff pendamping IT dan observasi langsung di Balai Tekkomdik DIY, yang kemudian dirumuskan dalam bentuk *user stories* dan *product backlog*. Hasil analisis menunjukkan bahwa peserta membutuhkan kemudahan dalam mengisi kehadiran dan *logbook* secara digital, admin memerlukan sistem verifikasi yang efisien, sementara pembimbing menginginkan fitur untuk *monitoring logbook* tanpa dokumen fisik.

Tahap pengembangan sistem dimulai dengan merancang arsitektur berbasis *web client-server* menggunakan *framework Flask* untuk *backend* dan MariaDB sebagai sistem manajemen basis data. Antarmuka dirancang dengan teknologi HTML, CSS, dan *Bootstrap* yang mendukung responsivitas serta kemudahan akses lintas perangkat. Struktur sistem disesuaikan untuk tiga jenis pengguna utama yaitu admin, peserta, dan pembimbing, dengan pembagian fitur dan hak akses yang jelas pada masing-masing *dashboard*. Proses implementasi dilanjutkan dengan penyusunan *use case* dan *activity diagram* guna menggambarkan interaksi pengguna serta alur kerja sistem secara menyeluruh. Diagram ini mencerminkan aktivitas nyata di lapangan seperti pendaftaran, pengisian kehadiran dan *logbook*, verifikasi, hingga penerbitan sertifikat. Validasi dilakukan bersama pengguna untuk memastikan bahwa alur tersebut sesuai kebutuhan operasional.

Pada tahap perancangan antarmuka, *wireframe* disusun untuk setiap halaman utama sistem. Desain tersebut kemudian diimplementasikan ke dalam

sistem dan diuji menggunakan metode *black box testing* untuk memastikan seluruh fitur berjalan sesuai fungsinya. Selain itu, dilakukan juga *User Acceptance Test* (UAT) yang melibatkan peserta, admin, dan pembimbing sebagai pengguna akhir. UAT bertujuan menilai tingkat fleksibilitas, adaptivitas, dan responsivitas sistem berdasarkan pengalaman langsung pengguna. Hasil pengujian menunjukkan bahwa sistem diterima dengan baik oleh seluruh peran pengguna. Sistem ini mampu mengurangi beban administratif, meningkatkan efisiensi pengelolaan data, dan mendukung pemantauan kegiatan secara digital dan *real-time*.

4.2 HASIL PENELITIAN

4.2.1 HASIL REQUIREMENT ANALYSIS

Berdasarkan hasil analisis kebutuhan melalui observasi di Balai Teknologi Komunikasi Pendidikan DIY dan wawancara dengan staf pendamping IT, ditemukan bahwa proses administrasi PKL yang berjalan selama ini masih dilakukan secara konvensional. Data peserta dicatat menggunakan formulir fisik dan *file spreadsheet* yang tersebar di perangkat lokal tanpa sistem pengelolaan terpusat. Prosedur tersebut menyebabkan data peserta sulit dilacak secara menyeluruh, memperlambat proses validasi pendaftaran, serta meningkatkan risiko duplikasi atau kehilangan informasi penting. Setiap peserta mengirimkan berkas pendaftaran melalui berbagai saluran komunikasi, seperti *email* atau pesan pribadi, yang menyulitkan admin untuk merekap secara cepat dan konsisten. Aktivitas harian peserta seperti kehadiran dan *logbook* juga masih dilaporkan secara konvensional, tanpa format standar dan tidak terdokumentasi secara berkelanjutan. Ketika jumlah peserta meningkat, admin dan pembimbing kesulitan dalam memantau progres harian dan menyusun laporan akhir program karena data tersebar, tidak terorganisir, dan tidak tersedia secara *real-time*. Tabel 4.1 menunjukkan *user stories* untuk pengembangan sistem informasi manajemen PKL di Balai Tekkomdik DIY.

Tabel 4.1 User Stories

User	Saya Ingin	Sehingga
Peserta	Mendaftar akun	Mengakses layanan PKL secara <i>online</i>
	Melakukan <i>login</i>	Mengakses akun
	Mengelola profil	Memperbarui profil
	Mengajukan pendaftaran	Mendaftar sebagai peserta PKL
	Mengisi kehadiran	Melaporkan kehadiran selama PKL
	Mengisi <i>logbook</i>	Melaporkan aktivitas selama PKL
	Mengunduh nilai dan sertifikat	Mendapatkan bukti resmi telah menyelesaikan PKL
Admin	Memverifikasi kehadiran peserta	Memastikan peserta hadir sesuai ketentuan
	Mengunggah sertifikat peserta	Memberikan nilai dan sertifikat peserta yang telah selesai
	Mengonfirmasi pendaftaran peserta	Memverifikasi peserta yang mendaftar
Pembimbing	Memverifikasi <i>logbook</i> peserta	Menilai dan memberikan umpan balik
	Mengunggah sertifikat peserta	Memberikan nilai dan sertifikat peserta yang telah selesai

Peserta membutuhkan sistem yang memungkinkan mereka mendaftarkan akun secara mandiri melalui *form daring* dan melakukan *login* menggunakan kredensial pribadi untuk mengakses layanan PKL sesuai status masing-masing. Setelah masuk ke sistem, peserta ingin dapat memperbarui informasi profil, termasuk data diri dan institusi asal,

serta mengajukan pendaftaran PKL dengan melengkapi dokumen yang dipersyaratkan. Setiap hari, peserta perlu mencatat waktu kehadiran melalui sistem tanpa harus menandatangani daftar konvensional, serta mengisi *logbook* harian secara langsung agar aktivitas mereka terdokumentasi dalam satu tempat. Seluruh data kehadiran dan *logbook* perlu dapat dilihat kembali oleh peserta melalui rekap yang terintegrasi, sehingga mereka bisa memantau keterpenuhan tugas selama program berlangsung. Setelah kegiatan selesai, peserta juga berharap bisa mengunduh sertifikat dalam format digital sebagai bukti resmi penyelesaian program PKL tanpa harus menunggu pengiriman fisik atau konfirmasi konvensional dari pihak admin.

Admin menghendaki sistem yang dapat memverifikasi pendaftaran peserta secara otomatis berdasarkan kelengkapan data dan dokumen yang diunggah melalui sistem. Fitur pengaturan waktu kehadiran seperti jam masuk, jam keluar, serta toleransi keterlambatan perlu tersedia agar pencatatan kehadiran berjalan sesuai kebijakan. Admin juga membutuhkan tampilan *dashboard* yang menyajikan data peserta aktif, status kehadiran, dan *logbook* secara ringkas, serta fitur untuk mengelola akun pembimbing dan menetapkan alokasi peserta bimbingan secara fleksibel. Selain itu, admin memerlukan kemampuan untuk mengunggah sertifikat peserta ke dalam sistem secara terpusat agar proses administrasi akhir lebih cepat dan terdokumentasi dengan baik.

Pembimbing memerlukan akses terhadap daftar peserta bimbingan lengkap beserta data institusi dan status PKL. Fitur peninjauan *logbook* secara berkala dibutuhkan agar pembimbing dapat memantau aktivitas peserta tanpa harus menunggu laporan konvensional. Pemantauan kehadiran juga menjadi aspek penting, terutama saat peserta menjalani program dalam waktu yang panjang. Setelah kegiatan selesai, pembimbing membutuhkan fitur untuk mengunggah sertifikat secara langsung ke akun peserta agar proses distribusi dokumen berjalan cepat dan tidak lagi dilakukan secara terpisah. Seluruh kebutuhan ini telah

dirumuskan ke dalam *user stories* sesuai peran masing-masing pengguna dan digunakan sebagai acuan utama dalam penyusunan *product backlog* untuk tahap pengembangan sistem informasi manajemen peserta PKL.

4.2.2 HASIL *PRODUCT BACKLOG*

Hasil identifikasi kebutuhan menunjukkan bahwa fitur-fitur yang berkaitan langsung dengan aktivitas peserta menjadi prioritas utama dalam pengembangan sistem. Tabel 4.2 adalah *product backlog* yang berisi daftar fitur yang akan dikembangkan dalam Sistem Informasi Manajemen PKL di Balai Tekkomdik DIY.

Tabel 4.2 Product Backlog

<i>User</i>	<i>Product Backlog Item</i>	Prioritas
Peserta	Mendaftar akun	Tinggi
	Melakukan <i>login</i>	Tinggi
	Mengelola profil	Menengah
	Mengajukan pendaftaran	Tinggi
	Mengisi kehadiran	Tinggi
	Mengisi <i>logbook</i>	Tinggi
	Mengunduh nilai dan sertifikat	Tinggi
Admin	Mengelola pengaturan kehadiran peserta	Menengah
	Memverifikasi kehadiran peserta	Menengah
	Mengunggah dan sertifikat	Tinggi
	Mengonfirmasi pendaftaran peserta	Tinggi
Pembimbing	Mengelola pengaturan kehadiran peserta	Menengah
	Memverifikasi <i>logbook</i> peserta	Tinggi
	Mengunggah sertifikat	Tinggi

Pendaftaran akun, *login*, pengajuan pendaftaran, pengisian kehadiran harian, pencatatan *logbook* digital, serta pengunduhan nilai dan sertifikat ditempatkan sebagai item dengan prioritas tinggi. Seluruh fitur tersebut

dianggap krusial karena berkaitan dengan keberlangsungan kegiatan peserta selama mengikuti program PKL. Kegiatan peserta berlangsung setiap hari dan membutuhkan pencatatan yang terdokumentasi secara sistematis. Sementara itu, fitur pengelolaan profil ditetapkan dalam prioritas menengah karena sifatnya hanya mendukung dan tidak bersifat mendesak terhadap kelancaran proses harian peserta.

Fitur-fitur pada peran admin difokuskan untuk memastikan bahwa seluruh proses administrasi berlangsung secara terkendali dan efisien. Konfirmasi pendaftaran peserta serta unggahan sertifikat merupakan dua fitur yang masuk dalam kategori prioritas tinggi karena berkaitan langsung dengan validitas status peserta di awal dan akhir program. Fitur lainnya seperti pengelolaan dan verifikasi kehadiran termasuk dalam prioritas menengah karena tidak harus diakses setiap saat, tetapi tetap penting untuk mendukung keberlanjutan pengawasan. Berdasarkan kondisi aktual yang diamati di Balai Tekkomdik DIY, admin memiliki beban administratif yang besar dalam mengelola peserta, sehingga fitur-fitur ini diharapkan mampu mengurangi ketergantungan terhadap sistem konvensional dan mempercepat proses kontrol data.

Pada peran pembimbing, fitur verifikasi *logbook* dan unggah sertifikat berada pada prioritas tinggi karena berhubungan langsung dengan pemantauan dan pengesahan aktivitas peserta bimbingan. Pembimbing perlu mendapatkan akses cepat terhadap catatan *logbook* peserta tanpa harus menerima laporan dalam bentuk fisik. Fitur pengelolaan pengaturan kehadiran ditempatkan sebagai prioritas menengah karena penggunaannya bersifat fleksibel dan tidak semua pembimbing terlibat dalam pengaturan waktu kehadiran. Seluruh fitur ini disesuaikan dengan tugas pembimbing di lapangan yang menekankan pada efisiensi pemantauan dan keterlibatan langsung terhadap progres peserta. Penyusunan *product backlog* ini menjadi cerminan kebutuhan riil pengguna yang ditemukan selama proses observasi dan wawancara dengan staff pendamping IT di Balai Tekkomdik DIY.

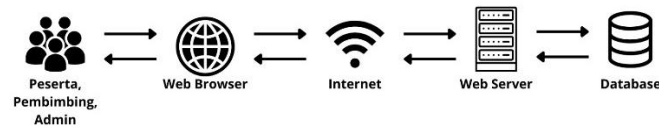
4.3 IMPLEMENTASI *DESIGN SYSTEM*

4.3.1 IMPLEMENTASI ARSITEKTUR SISTEM

Arsitektur sistem yang telah diimplementasikan mengikuti pola *client-server* berbasis *web*, dengan skema komunikasi yang melibatkan tiga elemen utama yaitu pengguna, *web server*, dan *database*. Seluruh pengguna sistem yang terdiri atas peserta, pembimbing, dan admin mengakses layanan menggunakan perangkat yang terhubung ke jaringan internet melalui *web browser*. Permintaan dari pengguna dikirimkan melalui *browser* menuju *web server*, kemudian diteruskan ke *database* untuk membaca, menyimpan, atau memperbarui data yang diperlukan. Proses ini berlangsung dua arah, sehingga sistem dapat merespons tindakan pengguna secara *real-time*. Arsitektur ini memungkinkan sistem bekerja secara terpusat dan terstruktur, dengan semua data tersimpan di *server* dan tidak bergantung pada perangkat pengguna.

Gambar 4.1 menampilkan visualisasi arsitektur sistem yang menggambarkan hubungan antar komponen utama, mulai dari antarmuka pengguna (*web browser*), server aplikasi, hingga *database*. Gambar ini menunjukkan bahwa pengguna tidak berinteraksi langsung dengan *database*, melainkan melalui serangkaian lapisan sistem. *Web browser* bertindak sebagai antarmuka visual yang digunakan untuk mengakses sistem, sedangkan *web server* menjalankan logika aplikasi dan mengelola komunikasi ke *database*. Implementasi ini memberikan keuntungan berupa kemudahan akses tanpa perlu instalasi perangkat lunak tambahan, serta pengendalian data yang lebih aman dan konsisten. Hasil pengujian menunjukkan bahwa sistem mampu menangani interaksi dari berbagai peran pengguna secara bersamaan tanpa konflik data. Skema ini menjadi solusi atas permasalahan sebelumnya, yaitu proses administrasi konvensional yang tersebar, tidak terdokumentasi secara terpusat, serta tidak memungkinkan akses lintas perangkat. Arsitektur ini

terbukti mendukung digitalisasi proses PKL di Balai Tekkomdik DIY secara efisien dan sesuai dengan kebutuhan pengguna.



Gambar 4.1 Arsitektur Sistem

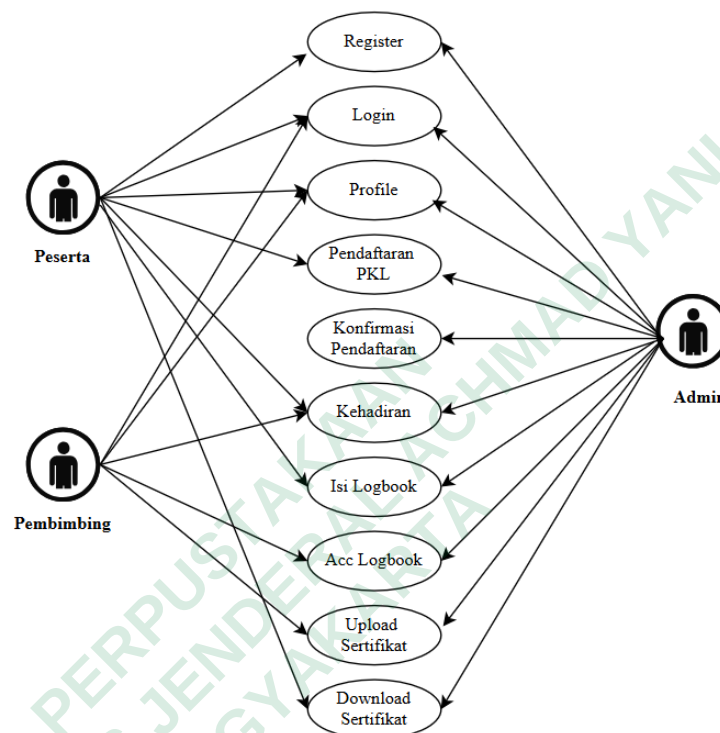
4.3.2 IMPLEMENTASI *USE CASE*

Implementasi sistem mencerminkan keberhasilan penerapan seluruh fungsionalitas utama yang telah diidentifikasi pada tahap analisis kebutuhan. Tiga pengguna utama, yakni peserta, admin, dan pembimbing, telah difasilitasi dengan akses serta fitur yang disesuaikan berdasarkan peran masing-masing. Akses pengguna dirancang berbasis peran untuk mendukung alur kegiatan PKL yang terstruktur, mencakup proses pendaftaran, pengelolaan data diri, pencatatan aktivitas harian, hingga pengunduhan sertifikat dalam format digital. Seluruh aktivitas dalam sistem tersimpan langsung ke dalam basis data, sehingga data dapat dikelola secara terpusat, konsisten, dan terdokumentasi secara otomatis.

Dari sisi pengelolaan, admin memiliki otoritas penuh untuk memverifikasi pendaftaran peserta, mengelola data kehadiran dan *logbook*, serta mengunggah sertifikat melalui sistem. Adapun pembimbing memiliki akses terhadap daftar peserta bimbingan, membuat pengaturan kehadiran, melakukan verifikasi catatan *logbook*, dan juga memiliki kewenangan untuk mengunggah sertifikat peserta sesuai tanggung jawabnya. Proses tersebut menggantikan metode konvensional yang sebelumnya mengandalkan dokumen cetak dan komunikasi tidak langsung, menjadi sistem digital yang lebih efisien dan terintegrasi.

Gambar 4.2 menampilkan diagram *use case* yang menggambarkan hubungan antara ketiga aktor pengguna utama dengan

berbagai fungsionalitas sistem yang mereka akses. Diagram ini menunjukkan bagaimana setiap peran memiliki interaksi khusus terhadap fitur yang relevan dengan tanggung jawabnya, sehingga mendukung kelancaran proses PKL secara terstruktur dan terkendali.



Gambar 4.2 Use Case

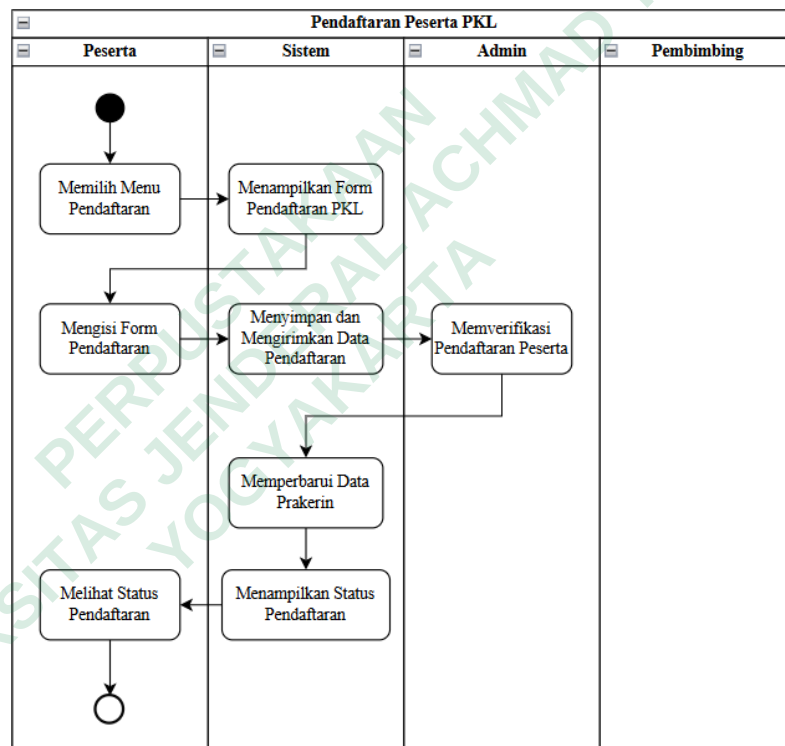
4.3.3 IMPLEMENTASI ACTIVITY DIAGRAM

1. Pendaftaran Peserta PKL

Tabel 4.3 menampilkan hasil implementasi *activity diagram* untuk proses pendaftaran peserta PKL yang menggambarkan alur kerja antar pengguna dan sistem secara terintegrasi. Proses dimulai ketika peserta memilih menu pendaftaran setelah berhasil masuk ke dalam sistem. Sistem kemudian menampilkan *form* isian pendaftaran PKL yang berisi data identitas, institusi asal, waktu pelaksanaan, dan dokumen pendukung. Setelah *form* diisi, sistem menyimpan dan mengirimkan data tersebut untuk diproses lebih lanjut. Admin menerima data pendaftaran yang masuk dan melakukan proses

verifikasi untuk memastikan kelengkapan serta kesesuaian dokumen yang dikirimkan. Jika pendaftaran dinyatakan *valid*, sistem akan memperbarui data prakerin dan mengubah status peserta menjadi “terdaftar.” Status pendaftaran ini ditampilkan langsung di halaman akun peserta sehingga mereka dapat memantau hasil pengajuan secara mandiri. Proses ini mampu menggantikan mekanisme konvensional yang sebelumnya dilakukan melalui pengiriman dokumen fisik dan konfirmasi pribadi kepada admin.

Tabel 4.3 Activity Diagram Pendaftaran Peserta PKL



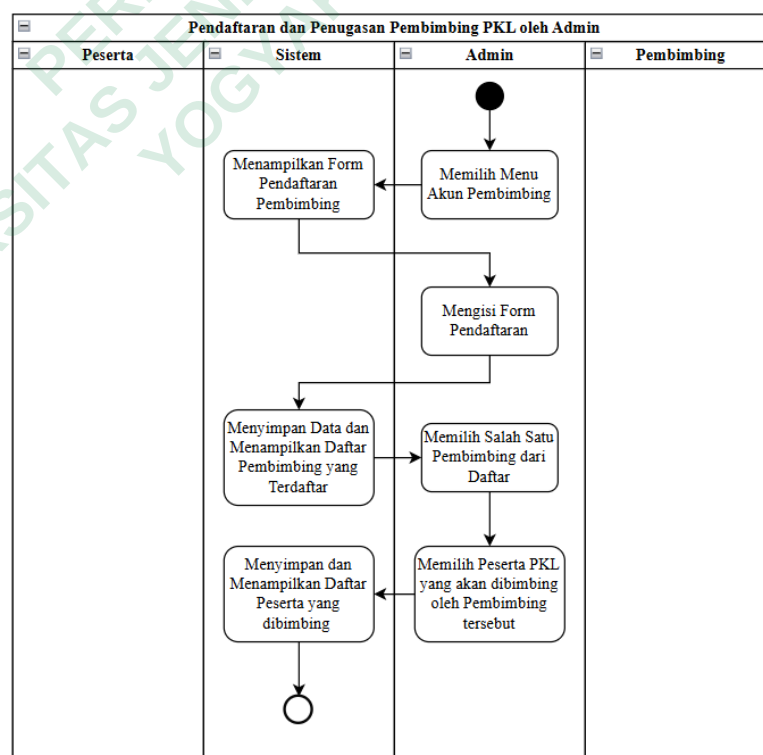
2. Pendaftaran dan Penugasan Pembimbing PKL oleh Admin

Tabel 4.4 menunjukkan hasil implementasi *activity diagram* untuk proses pendaftaran dan penugasan pembimbing PKL oleh admin. Aktivitas dimulai ketika admin memilih menu akun pembimbing melalui *dashboard* sistem. Setelah itu, sistem menampilkan *form* isian pendaftaran pembimbing yang harus dilengkapi oleh admin, mencakup identitas pembimbing, unit kerja,

dan informasi pendukung lainnya. Setelah *form* diisi, sistem menyimpan data tersebut dan secara otomatis memperbarui daftar pembimbing yang terdaftar dalam sistem. Daftar ini akan ditampilkan kembali saat admin melakukan proses penugasan peserta.

Setelah pembimbing terdaftar, admin memilih salah satu nama pembimbing dari daftar yang tersedia. Selanjutnya, admin akan memilih peserta PKL yang akan dibimbing oleh pembimbing tersebut. Setelah pemilihan dilakukan, sistem menyimpan data relasi antara pembimbing dan peserta, lalu menampilkan daftar peserta yang berada di bawah bimbingan masing-masing pembimbing. Implementasi alur pada tabel 4.4 ini menunjukkan bahwa proses yang sebelumnya dilakukan secara konvensional kini telah diintegrasikan secara digital, sehingga memungkinkan admin untuk menambahkan pembimbing baru dan mengelola relasi pembimbing-peserta secara cepat, akurat, dan terdokumentasi otomatis oleh sistem.

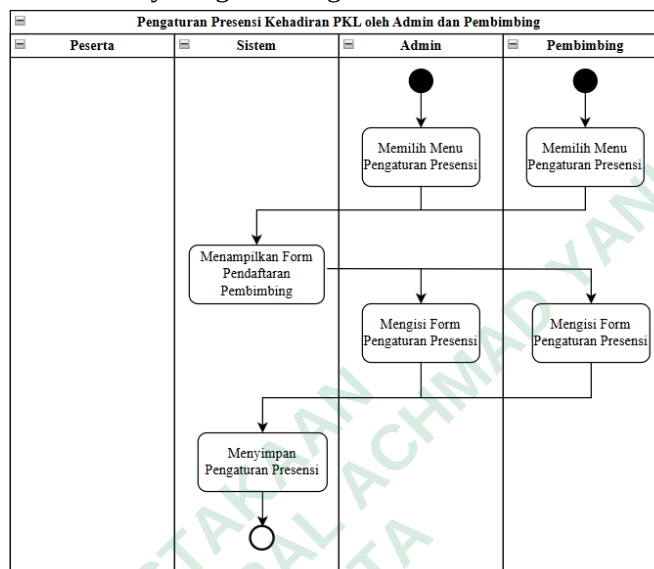
Tabel 4.4 Activity Diagram Pendaftaran dan Penugasan Pembimbing



3. Proses Pengaturan Kehadiran

Berikut tabel 4.5 merupakan *activity diagram* untuk proses pengaturan kehadiran harian peserta PKL:

Tabel 4.5 Activity Diagram Pengaturan Kehadiran Kehadiran PKL



Tabel 4.5 menunjukkan hasil implementasi *activity diagram* untuk proses pengaturan kehadiran harian peserta PKL yang dapat dilakukan oleh admin maupun pembimbing. Aktivitas dimulai ketika admin atau pembimbing memilih menu pengaturan kehadiran melalui *dashboard* masing-masing. Sistem kemudian menampilkan *form* pengaturan yang berisi kolom waktu masuk, waktu keluar, serta toleransi keterlambatan peserta selama mengikuti kegiatan PKL. *Form* ini bersifat fleksibel, sehingga masing-masing peran pengguna dapat menyesuaikan pengaturan sesuai kebijakan internal maupun kondisi teknis yang berlaku.

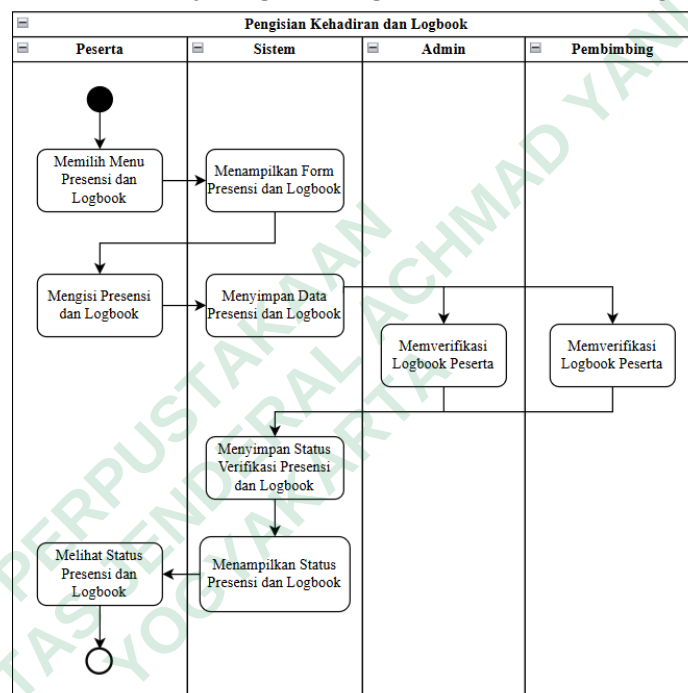
Setelah *form* diisi, sistem akan menyimpan konfigurasi yang telah ditentukan dan secara otomatis menerapkannya pada modul kehadiran peserta. Data pengaturan ini akan memengaruhi validasi kehadiran yang dilakukan sistem setiap kali peserta mengisi kehadiran harian. Implementasi proses yang digambarkan dalam tabel 4.5 ini memperlihatkan bahwa sistem mampu memfasilitasi fleksibilitas

pengaturan jadwal oleh dua peran sekaligus, yaitu admin sebagai pengelola utama dan pembimbing sebagai pemantau langsung peserta.

4. Proses Pengisian Kehadiran dan Logbook

Berikut tabel 4.6 merupakan *activity diagram* proses pengisian kehadiran dan *logbook*:

Tabel 4.6 Activity Diagram Pengisian Kehadiran dan Logbook



Tabel 4.6 menunjukkan hasil implementasi *activity diagram* pada proses pengisian kehadiran dan *logbook*. Aktivitas diawali ketika peserta memilih menu kehadiran dan *logbook* melalui antarmuka sistem. Sistem kemudian menampilkan *form* isian yang memuat kolom waktu kehadiran dan catatan aktivitas harian yang harus dilengkapi peserta. Setelah *form* terisi, sistem secara otomatis menyimpan data ke dalam *database* dan mencatat tanggal serta waktu *input*. Pada tahap ini, data yang masuk belum secara langsung ditandai sebagai *valid*, melainkan menunggu proses verifikasi dari admin atau pembimbing.

Setelah data tersimpan, admin dan/atau pembimbing memiliki akses untuk memverifikasi isi *logbook* sesuai peran masing-masing. Proses verifikasi dilakukan untuk memastikan bahwa catatan yang dikirimkan peserta relevan dan sesuai dengan aktivitas PKL yang dijalani. Setelah verifikasi dilakukan, sistem akan memperbarui status *logbook* dan kehadiran peserta, lalu menampilkan status tersebut kembali ke peserta melalui halaman *dashboard*. Melalui sistem ini, seluruh catatan *logbook* dan kehadiran tercatat otomatis dan dapat dipantau kembali oleh seluruh pihak yang terlibat.

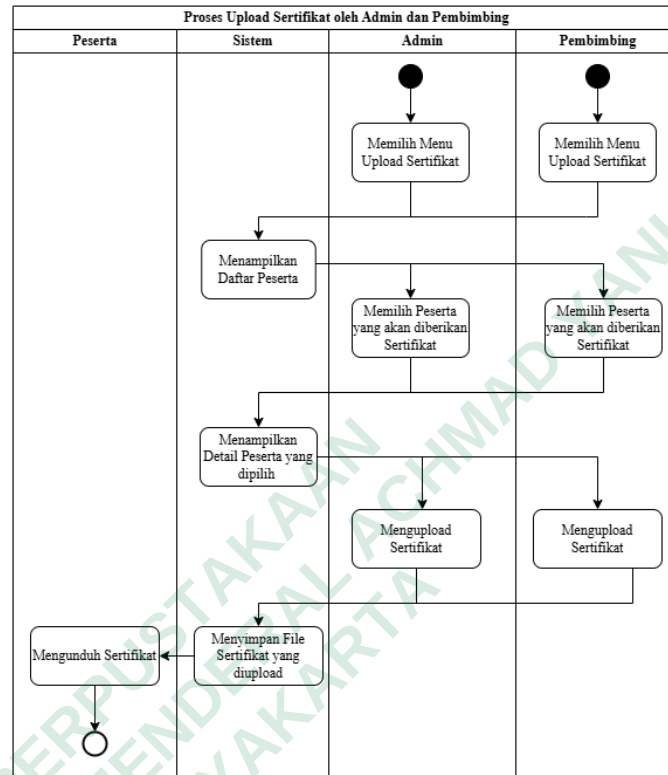
5. Proses unggah sertifikat oleh admin dan pembimbing

Tabel 4.7 menggambarkan hasil implementasi *activity diagram* pada proses unggah sertifikat oleh admin dan pembimbing. Proses dimulai ketika admin maupun pembimbing memilih menu unggah sertifikat melalui *dashboard* masing-masing. Setelah itu, sistem menampilkan daftar peserta yang tersedia berdasarkan data peserta aktif. Admin atau pembimbing kemudian memilih salah satu peserta yang akan diberikan sertifikat, dan sistem menampilkan detail peserta secara otomatis. Tahap ini memungkinkan pengguna untuk memastikan bahwa data peserta sudah sesuai sebelum proses unggah dilakukan.

Setelah data diverifikasi secara visual, admin atau pembimbing mengunggah *file* sertifikat ke sistem melalui *form* yang tersedia. Sistem kemudian menyimpan *file* tersebut ke dalam *database* terpusat dan mengaitkannya dengan akun peserta yang bersangkutan. Peserta yang telah diunggah sertifikatnya akan langsung memperoleh akses untuk mengunduh *file* tersebut dari halaman akun masing-masing. Hasil implementasi pada Tabel 4.7 ini membuktikan bahwa sistem telah berhasil mengotomatisasi distribusi sertifikat secara digital. Proses ini menghilangkan ketergantungan pada

pengiriman fisik dan mempercepat waktu akses peserta terhadap dokumen penting sebagai bukti penyelesaian program PKL.

Tabel 4.7 Activity Diagram Proses Upload Sertifikat



4.3.4 IMPLEMENTASI WIREFRAME

1. Pendaftaran peserta PKL

Gambar 4.3 menunjukkan tampilan halaman pendaftaran peserta PKL berdasarkan hasil implementasi *wireframe* yang dirancang sesuai kebutuhan pengguna di Balai Tekkomdik DIY. Visualisasi ini memperlihatkan dua komponen utama yang harus diisi oleh pengguna, yaitu formulir institusi dan formulir data peserta. Formulir institusi mencakup kolom *input* nama sekolah atau kampus pengirim, alamat, *email*, jenjang pendidikan, serta waktu pelaksanaan PKL. Sistem juga menyediakan fitur unggah dokumen berupa surat

permohonan dan proposal dalam format PDF sebagai syarat administrasi yang harus dipenuhi.

Pada bagian bawah, halaman ini menampilkan *form* data peserta yang berisi kolom nama lengkap, NIM/NISN, email, jurusan, serta tipe peserta. Fitur "Tambahkan Data" disediakan agar pengguna dapat mengajukan lebih dari satu peserta dalam satu kali pengisian form. Setelah semua data terisi, pengguna dapat mengirim formulir melalui tombol "Kirim" yang terletak di bagian akhir halaman. Desain antarmuka yang ditampilkan pada Gambar 4.3 dirancang agar sederhana, intuitif, dan responsif sehingga peserta dapat melakukan pendaftaran secara mandiri tanpa bantuan admin. Gambar ini menunjukkan bagaimana sistem memfasilitasi digitalisasi proses pendaftaran PKL sekaligus mendukung efisiensi waktu dan kemudahan akses bagi pengguna.

LOGO

Home Program Pendaftaran Data FAQ Kontak

Pendaftaran Prakerin

Pendaftaran Prakerin di Balai Tekkomdik DIY

Nama Sekolah/Perguruan Tinggi

Telp Sekolah/Perguruan Tinggi Email Sekolah/Perguruan Tinggi

Alamat Sekolah/Perguruan Tinggi

Jenis

Tanggal Mulai Prakerin/PKL Tanggal Selesai Prakerin/PKL

Upload Surat Permohonan

Format : PDF, DOC, atau DOCX

Upload Proposal

Format : PDF, DOC, atau DOCX

Data Peserta

Nama Peserta

NISNIM Jurusan

Email Peserta Telp Peserta

TAMBAH DATA KIRIM

FOOTER

Gambar 4.3 Wireframe Pendaftaran Peserta PKL

2. Pengisian kehadiran dan *logbook* harian

Berikut gambar 4.4 merupakan *wireframe* pengisian kehadiran dan *logbook* harian:

Gambar 4.4 Wireframe Pengisian Kehadiran dan Logbook

Gambar 4.4 menampilkan hasil implementasi halaman pengisian kehadiran dan *logbook* harian peserta PKL yang diakses melalui akun peserta. Tampilan ini memperlihatkan dua fitur utama dalam satu halaman, yaitu pencatatan kehadiran dan *form* pengisian *logbook*. Pada bagian atas halaman terdapat panel kehadiran yang memuat tombol “Absen Masuk” dan “Absen Keluar” untuk mencatat kehadiran peserta secara langsung. Halaman ini dikembangkan untuk mendukung pencatatan aktivitas peserta PKL secara digital, yang sebelumnya dilakukan secara konvensional menggunakan formulir cetak atau dokumen terpisah, sekaligus berfungsi sebagai alat kontrol kehadiran harian.

Pada bagian tengah halaman, sistem menyediakan *form logbook* harian dengan input berupa nama kegiatan, jam pelaksanaan, dan keterangan tambahan. Setelah *form* diisi, peserta dapat menyimpan data melalui tombol “Simpan” dan entri *logbook* akan tersimpan dalam sistem. Gambar ini juga memperlihatkan tabel riwayat *logbook* yang menampilkan seluruh aktivitas harian yang telah dicatat sebelumnya, dilengkapi dengan fitur pencarian, navigasi halaman, dan tombol unduh untuk rekap data. Tampilan pada Gambar

4.4 dirancang agar peserta dapat mencatat aktivitas dan kehadiran secara terintegrasi dalam satu halaman, sehingga proses dokumentasi harian menjadi lebih efisien, terdokumentasi otomatis, dan mudah diakses oleh admin maupun pembimbing untuk keperluan verifikasi.

3. Pengaturan kehadiran

Gambar 4.5 menampilkan hasil implementasi *wireframe* untuk fitur pengaturan kehadiran yang dapat diakses oleh admin maupun pembimbing melalui *dashboard* masing-masing. Tampilan ini disusun dalam bentuk *popup form* yang muncul secara otomatis ketika pengguna memilih menu pengaturan kehadiran, dengan tujuan untuk mempermudah proses penetapan atau pembaruan jam masuk dan keluar peserta PKL secara digital. Visualisasi ini memperlihatkan beberapa kolom input utama yang harus diisi pengguna, yaitu: Jam Masuk Mulai, Jam Masuk Selesai, Toleransi Masuk (Menit), Jam Keluar Mulai, Jam Keluar Selesai, dan Toleransi Keluar (Menit). Setiap kolom dilengkapi dengan ikon pemilih waktu (*time picker*) untuk memudahkan pengguna dalam memilih jam tertentu tanpa harus menginput secara manual.

Form pengaturan ini dirancang dengan struktur antarmuka yang sederhana dan fokus, sehingga pengguna dapat melakukan penyesuaian waktu presensi peserta secara fleksibel tanpa berpindah halaman. Fitur ini memungkinkan sistem menetapkan batas waktu presensi secara otomatis, serta mendeteksi keterlambatan masuk maupun kepulangan lebih awal sesuai kebijakan internal instansi. Pada bagian bawah *form*, terdapat dua tombol aksi, yakni tombol “Simpan” yang berfungsi untuk menyimpan seluruh konfigurasi kehadiran ke dalam sistem, serta tombol “Batal” yang memungkinkan pengguna membatalkan perubahan apabila terdapat kesalahan input atau ingin kembali ke pengaturan sebelumnya.

Desain antarmuka yang diimplementasikan pada Gambar 4.5 ini memberikan solusi konkret terhadap pengelolaan presensi peserta PKL secara digital, memperkuat penerapan disiplin waktu, serta meningkatkan efisiensi pengawasan oleh pihak admin maupun pembimbing.

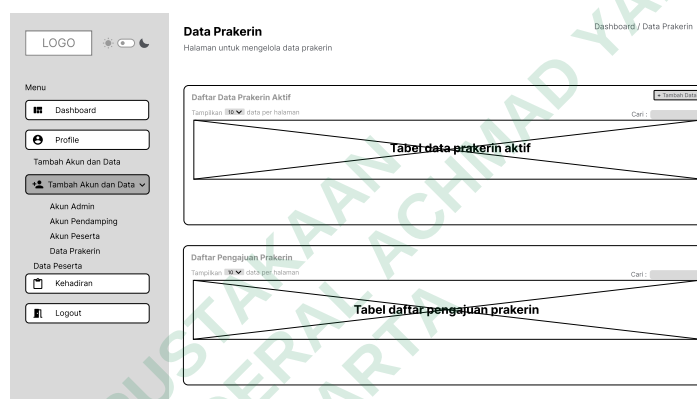
Gambar 4.5 Wireframe Pengaturan Kehadiran

4. Halaman data prakerin

Gambar 4.6 menunjukkan hasil implementasi *wireframe* halaman data prakerin yang dirancang khusus untuk digunakan oleh admin dalam proses pengelolaan data pendaftaran peserta PKL. Halaman ini dirancang agar admin dapat memantau status pendaftaran peserta secara menyeluruh dan responsif, baik untuk pengajuan baru maupun peserta yang telah aktif menjalani program PKL. Tampilan halaman ini dibagi menjadi dua bagian utama. Bagian pertama menampilkan tabel data prakerin aktif, yaitu daftar peserta yang sudah diverifikasi dan sedang menjalani PKL di Balai Tekkomdik DIY. Sementara itu, bagian kedua berisi tabel daftar pengajuan PKL, yaitu kumpulan data peserta yang telah mendaftar namun masih menunggu proses validasi dari admin.

Setiap tabel dilengkapi dengan fitur pencarian, kontrol jumlah data per halaman, serta opsi navigasi antar halaman untuk memudahkan admin dalam meninjau dan mengelola data secara efisien, terutama ketika jumlah pendaftar meningkat dalam satu periode tertentu. Tabel juga menampilkan informasi penting seperti

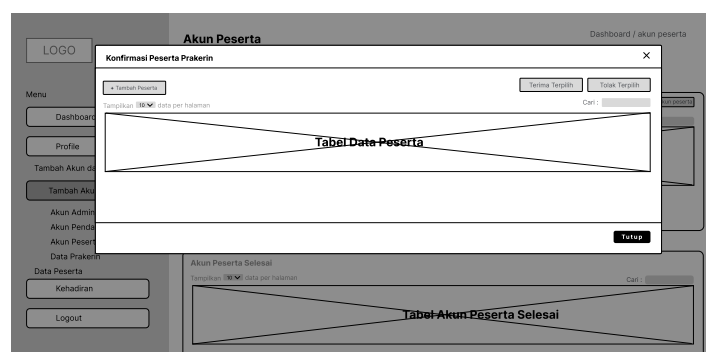
nama peserta, asal institusi, tanggal pengajuan, dan status verifikasi. Desain antarmuka ini menitikberatkan pada kemudahan navigasi dan keterbacaan informasi, sehingga admin dapat melakukan proses *monitoring*, verifikasi, dan pengambilan keputusan administratif secara cepat dan akurat. Implementasi *wireframe* pada Gambar 4.6 menjadi solusi konkret terhadap kebutuhan pengelolaan pendaftaran yang lebih sistematis, terdokumentasi dengan baik, dan terintegrasi dalam satu halaman tampilan yang efisien.



Gambar 4.6 Wireframe Data Prakerin

5. Proses konfirmasi pendaftaran peserta

Berikut gambar 4.7 merupakan *wireframe* konfirmasi pendaftaran peserta:



Gambar 4.7 Wireframe Konfirmasi Pendaftaran Peserta

Gambar 4.7 memperlihatkan hasil implementasi *wireframe* untuk proses konfirmasi peserta PKL oleh admin, yang dapat diakses melalui menu akun peserta pada sistem. Tampilan ini disusun dalam

bentuk *popup window* interaktif yang muncul secara otomatis ketika admin memilih menu konfirmasi pendaftaran.

Pada bagian utama *popup*, sistem menampilkan tabel daftar calon peserta yang telah mengajukan pendaftaran namun belum mendapatkan keputusan. Setiap baris dalam tabel mencantumkan informasi penting seperti nama peserta, asal institusi, tanggal pendaftaran, dan status pengajuan. Admin dapat melakukan seleksi terhadap satu atau beberapa peserta sekaligus dengan menggunakan fitur *checkbox* yang disediakan di samping setiap entri.

Setelah proses seleksi dilakukan, sistem menyediakan dua tombol aksi utama, yaitu “Terima Terpilih” untuk mengonfirmasi peserta yang lulus verifikasi, dan “Tidak Terpilih” untuk peserta yang tidak memenuhi kriteria. Setelah keputusan dikirim, sistem secara otomatis akan memperbarui status peserta dan memindahkan data yang bersangkutan ke status “selesai”, yang kemudian ditampilkan pada bagian bawah halaman utama akun peserta.

Desain tampilan ini dirancang untuk memberikan pengalaman verifikasi yang cepat, terpusat, dan efisien, tanpa memerlukan perpindahan ke halaman lain. Dengan demikian, admin dapat menyelesaikan proses validasi secara lebih praktis dan terdokumentasi secara otomatis di dalam sistem. Implementasi *wireframe* ini sekaligus mendukung transparansi serta pengurangan beban administratif manual dalam proses seleksi peserta PKL.

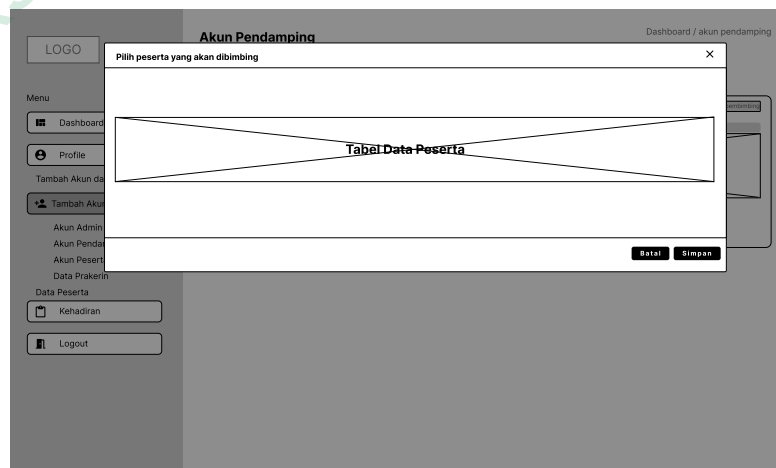
6. Distribusi peserta PKL ke pembimbing oleh admin

Gambar 4.8 menunjukkan hasil implementasi *wireframe* untuk fitur distribusi peserta PKL ke pembimbing, yang dirancang khusus bagi admin guna mempermudah proses alokasi bimbingan secara digital. Fitur ini menjadi solusi atas mekanisme konvensional yang sebelumnya dilakukan secara terpisah dan tidak terdokumentasi

dengan baik, sehingga sering menimbulkan kendala dalam proses pencatatan dan pemerataan distribusi peserta ke pembimbing.

Tampilan halaman ini memungkinkan admin untuk memilih nama pembimbing dari daftar *dropdown* yang tersedia, kemudian menetapkan peserta yang akan dibimbing oleh masing-masing pembimbing tersebut. Data peserta ditampilkan dalam bentuk tabel yang dilengkapi dengan fitur pencarian, navigasi antar halaman, serta *checkbox* untuk seleksi multipel, sehingga proses seleksi dapat dilakukan dengan cepat, akurat, dan terorganisir. Setelah pembagian peserta ditetapkan, sistem akan secara otomatis mencatat dan menyimpan relasi antara peserta dan pembimbing ke dalam *database*. Hasil alokasi ini akan ditampilkan kembali di akun pembimbing agar proses pemantauan, verifikasi *logbook*, dan pengunggahan sertifikat dapat dilakukan sesuai tanggung jawab masing-masing.

Implementasi *wireframe* ini mendukung efisiensi kerja admin, mempercepat proses distribusi peserta, serta memastikan bahwa setiap peserta teralokasi secara merata ke pembimbing yang tersedia dan terdokumentasi dengan baik dalam sistem. Dengan adanya fitur ini, pengelolaan bimbingan menjadi lebih terstruktur dan dapat dipantau secara menyeluruh selama kegiatan PKL berlangsung.



Gambar 4.8 Wireframe Distribusi Peserta PKL ke Pembimbing

4.3.5 IMPLEMENTASI DATABASE

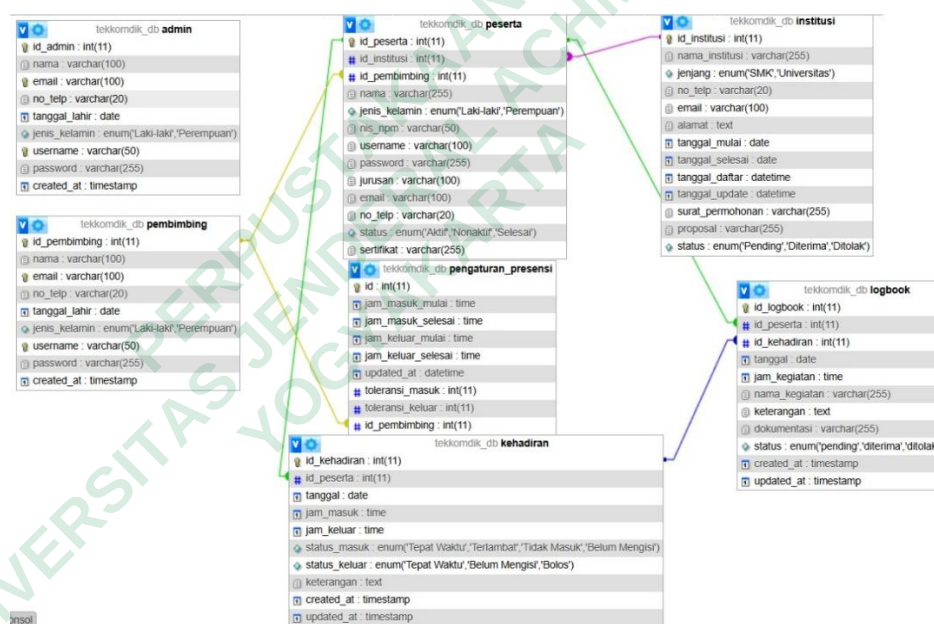
Gambar 4.9 menunjukkan hasil implementasi struktur *database* dari SIM peserta PKL yang dibangun menggunakan pendekatan basis data relasional. Visualisasi ini digambarkan dalam bentuk *Entity Relationship Diagram* (ERD) yang memperlihatkan hubungan antar entitas utama dalam sistem, seperti peserta, institusi, pembimbing, kehadiran, dan *logbook*. ERD ini berfungsi untuk memberikan gambaran menyeluruh tentang bagaimana data saling terhubung, dikelola, dan digunakan dalam sistem, serta menjadi dasar dalam pembangunan skema database yang terintegrasi dan efisien.

Rancangan ini terdiri atas delapan tabel utama, yaitu peserta, pembimbing, admin, institusi, kehadiran, *logbook*, pengaturan_kehadiran, dan kehadiran yang saling terhubung melalui relasi *primary key* dan *foreign key*. Tabel peserta menjadi pusat integrasi data dengan mengaitkan *id_pembimbing* dan *id_institusi*, sehingga setiap peserta terhubung langsung ke institusi asal dan pembimbing yang ditugaskan. Tabel institusi menyimpan informasi asal sekolah atau universitas peserta, termasuk jenjang, tanggal pelaksanaan, serta status verifikasi dokumen.

Data kehadiran harian disimpan dalam tabel kehadiran, yang mencatat tanggal, jam masuk, jam keluar, status kehadiran, serta keterangan khusus. Tabel ini terhubung langsung ke pengaturan_kehadiran, yang menyimpan konfigurasi jam masuk dan keluar, serta batas toleransi keterlambatan berdasarkan kebijakan masing-masing pembimbing. Aktivitas harian peserta selama PKL dicatat melalui tabel *logbook*, yang mencakup nama kegiatan, jam pelaksanaan, dokumentasi, serta status verifikasi. Setiap entri *logbook* terhubung ke peserta dan diverifikasi oleh pembimbing atau admin. Sementara itu, tabel admin dan pembimbing digunakan untuk menyimpan informasi akun pengguna sistem, termasuk nama, *email*, kata

sandi, dan data identitas lain yang diperlukan untuk proses autentikasi dan pengelolaan hak akses.

Struktur ini telah diuji menggunakan skenario penggunaan nyata seperti pendaftaran peserta, pengisian kehadiran, pencatatan *logbook*, serta distribusi peserta ke pembimbing. Seluruh relasi antar tabel berfungsi secara konsisten dan mendukung eksekusi *query* dengan cepat dan akurat. Hasil implementasi menunjukkan bahwa struktur *database* mampu mengakomodasi kebutuhan fungsional sistem secara menyeluruh dan siap digunakan sebagai basis data utama dalam pengelolaan PKL digital di Balai Tekkomdik DIY. Berikut gambar 4.9 adalah *Entity Relationship Database (ERD)* dari *database* yang telah dibuat:



Gambar 4.9 ERD Database SIM Peserta PKL

4.4 IMPLEMENTASI SPRINT PLANNING

Sprint planning dilaksanakan sebagai tahap awal penerapan metode *Scrum* dalam pengembangan SIM Peserta PKL. Kegiatan ini bertujuan menyusun rencana pengembangan berdasarkan *product backlog* yang telah dirumuskan pada tahap sebelumnya. Tim menyepakati bahwa fitur prioritas awal mencakup *login*, pendaftaran peserta, pengisian kehadiran harian, pencatatan *logbook*, dan *dashboard* pengguna karena fitur-fitur tersebut

langsung menyentuh proses administrasi PKL yang sebelumnya masih konvensional. Durasi *sprint* pertama ditetapkan satu bulan dengan estimasi total 90 jam kerja.

Dalam diskusi *sprint planning*, juga ditetapkan struktur tim *Scrum*. *Product Owner* berasal dari pihak Balai Tekkomdik DIY yang menetapkan prioritas fitur dan memvalidasi hasil. *Scrum Master* adalah staf pendamping IT yang memfasilitasi proses *Scrum* dan menghilangkan hambatan teknis. Avriil Putra Mahardika bertindak sebagai *developer frontend-backend*, sedangkan Sa'adatul Abadiyah berperan sebagai *tester*. Kesepakatan ini dituangkan dalam *Product Goal Canvas* dan rencana *sprint* pada tabel 4.8 berikut:

Tabel 4.8 Product Goal Canvas SIM Peserta PKL

Aspek	Uraian
<i>Vision</i>	Tersedianya sistem terintegrasi, efektif, dan transparan dalam pengelolaan peserta PKL di Balai Tekkomdik DIY.
<i>Product Goal</i>	• Digitalisasi pendaftaran dan pengelolaan peserta PKL • Otomatisasi kehadiran dan <i>logbook</i> • Rekapitulasi dan <i>monitoring</i> kehadiran serta <i>logbook</i> oleh pembimbing • Pengelolaan akun pembimbing dan peserta oleh admin
<i>Description</i>	• Sistem menggantikan formulir kertas serta <i>spreadsheet</i> lokal. • Pengguna utama yaitu peserta, admin, pembimbing. • Pengembangan bertahap menggunakan <i>Agile Scrum</i>.
<i>Valuable Outcome</i>	• Hemat waktu administrasi • Transparansi proses PKL • Data tersimpan aman • Evaluasi peserta lebih akurat
<i>Stakeholders</i>	Peserta PKL, Admin, Pembimbing, Subbag Umum dan Kepegawaian, serta Tim Pengembang

Aspek	Uraian
<i>Risk</i>	Adaptasi pengguna, gangguan jaringan, ketidaksesuaian alur awal.
<i>Scrum Team</i>	• <i>Product Owner</i>: Kepala Subbag PKL DIY • <i>Scrum Master</i>: Staf Pendamping IT • <i>Developer</i>: Avril Putra Mahardika • <i>Tester</i>: Sa'adatul Abadiyah

Rencana pengembangan sistem dibagi ke dalam tiga *sprint*, dengan pembagian fitur berdasarkan prioritas dan kompleksitasnya. Tabel 4.9 berikut menunjukkan daftar fitur yang akan dikembangkan pada *Sprint 1* dan estimasi durasinya:

Tabel 4.9 Fitur yang dikembangkan pada Sprint 1 dan Estimasi Durasinya

No	Fitur yang Dikembangkan	Estimasi Durasi (Jam)
1	<i>Login</i> dan autentikasi pengguna	12
2	Pendaftaran dan pendaftaran peserta PKL	14
3	Pengelolaan akun dan <i>dashboard</i> pengguna	10
4	Pengaturan Kehadiran	10
5	Pengisian kehadiran harian	16
6	Pencatatan <i>logbook</i>	18
7	Pengujian fungsional awal	10
Total Estimasi Durasi <i>Sprint 1</i>		90 Jam

Tabel 4.10 berikut menunjukkan daftar fitur yang akan dikembangkan pada *Sprint 2* dan estimasi durasinya:

Tabel 4.10 Fitur yang dikembangkan pada Sprint 2 dan Estimasi Durasinya

No	Fitur yang Dikembangkan	Estimasi Durasi (Jam)
1	Penyempurnaan tampilan dan navigasi <i>dashboard</i>	10
2	Rekap kehadiran oleh admin dan pembimbing	12
3	Rekap <i>logbook</i> oleh admin dan pembimbing	10
4	Pengaturan akses berbasis peran	12

No	Fitur yang Dikembangkan	Estimasi Durasi (Jam)
5	Revisi dan konfirmasi pendaftaran peserta	8
6	Uji regresi dan integrasi fitur baru	8
	Total Estimasi Durasi <i>Sprint 2</i>	60 Jam

Tabel 4.11 berikut menunjukkan daftar fitur yang akan dikembangkan pada *Sprint 3* dan estimasi durasinya:

Tabel 4.11 Fitur yang dikembangkan pada *Sprint 3* dan Estimasi Durasinya

No	Fitur yang Dikembangkan	Estimasi Durasi (Jam)
1	Unggah sertifikat peserta oleh admin dan pembimbing	12
2	Peserta mengunduh sertifikat	10
3	Pengelolaan akun pembimbing dan penugasan peserta	12
4	Revisi tampilan dan perbaikan minor	8
5	Pengujian sistem menyeluruh dan validasi akhir	10
	Total Estimasi Durasi <i>Sprint 3</i>	52 Jam

4.5 IMPLEMENTASI *DEVELOPMENT*

Pengembangan sistem informasi manajemen peserta PKL dilaksanakan secara bertahap melalui tiga *sprint* yang mengacu pada kerangka kerja *Agile Scrum*. Setiap *sprint* difokuskan pada pembangunan fitur prioritas, pengujian fungsional, dan penyesuaian hasil berdasarkan umpan balik pengguna. Hasil pengembangan disusun berdasarkan urutan *sprint* untuk menunjukkan progres sistem secara konkret dan terstruktur.

4.5.1 IMPLEMENTASI *SPRINT 1*

Pelaksanaan *sprint* pertama menghasilkan sejumlah fitur dasar yang berfungsi sebagai fondasi utama sistem. Fitur *login* telah dikembangkan untuk membedakan akses pengguna berdasarkan peran,

dilanjutkan dengan fitur pendaftaran peserta yang memungkinkan pengisian data institusi dan peserta secara terstruktur. Data yang diinput tersimpan otomatis dan siap diverifikasi oleh admin melalui halaman khusus. Pengaturan kehadiran juga berhasil diimplementasikan, memungkinkan admin maupun pembimbing menetapkan parameter waktu kehadiran yang akan menjadi acuan bagi peserta dalam mencatat kehadiran harian.

Fitur kehadiran dan *logbook* yang dikembangkan telah terintegrasi dalam satu halaman, memudahkan peserta mencatat aktivitas dan kehadiran tanpa perlu berpindah antarmuka. Setiap catatan *logbook* akan tersimpan dalam riwayat dan dapat diverifikasi langsung oleh admin atau pembimbing. Seluruh fitur diuji berdasarkan skenario penggunaan standar dan telah menunjukkan fungsionalitas sesuai kebutuhan awal. Adapun penjelasan lebih lanjut mengenai realisasi beberapa fitur penting yang dikembangkan pada *sprint* pertama dapat diuraikan sebagai berikut:

1. Realisasi Fitur Pendaftaran Peserta

Proses pendaftaran peserta dirancang melalui dua tahapan utama yang mencerminkan keterkaitan data antara institusi dan individu peserta. Tahapan pertama ditujukan untuk merekam informasi institusi asal ke dalam basis data melalui perintah `INSERT INTO` institusi. Data yang dicatat meliputi nama institusi, nomor telepon, alamat *email*, alamat fisik, jenjang pendidikan, tanggal mulai dan selesai kegiatan, serta dua dokumen pendukung, yakni surat permohonan dan proposal. Setelah proses penyimpanan berhasil, sistem menjalankan `mysql.connection.commit()` untuk mengunci data secara permanen, lalu memperoleh nilai `id_institusi` terakhir menggunakan `cur.lastrowid` sebagai identifikasi relasional yang akan digunakan pada tahap berikutnya.

Tahap kedua dilakukan dengan melakukan perulangan terhadap seluruh entri peserta yang telah diinputkan. Pada setiap iterasi, sistem mengeksekusi *query* INSERT INTO peserta guna menyimpan nama peserta, NIS/NIM, jurusan, *email*, dan nomor telepon ke dalam tabel peserta, sambil menyisipkan *id_institusi* sebagai *foreign key* yang menghubungkan peserta dengan institusi asalnya. Setiap eksekusi disertai pemanggilan `mysql.connection.commit()` agar perubahan langsung tercatat. Berikut kode program 4.1 adalah realisasi fitur pendaftaran peserta:

Kode Program 4.1 Pendaftaran Peserta

```
# Simpan data institusi ke DB
cur.execute("""
    INSERT INTO institusi (nama_institusi, no_telp,
email, alamat, jenjang, tanggal_mulai,
tanggal_selesai, surat_permohonan, proposal)
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    """, (nama_institusi, no_telp, email, alamat,
jenjang, tanggal_mulai, tanggal_selesai,
surat_permohonan_path, proposal_path))
mysql.connection.commit()
id_institusi = cur.lastrowid

# Simpan data peserta ke DB
for i in range(len(nama_peserta)):
    cur.execute("""
        INSERT INTO peserta (id_institusi, nama,
nis_npm, jurusan, email, no_telp)
        VALUES (%s, %s, %s, %s, %s, %s)
        """, (id_institusi, nama_peserta[i],
nis_nim[i], jurusan[i], email_peserta[i],
telp_peserta[i]))
    mysql.connection.commit()
```

Berikut gambar 4.10 adalah tampilan *dashboard* pendaftaran peserta bagian atas:

Pendaftaran Prakerin
Pendaftaran Prakerin di Balai Tekkomdik DIY

Nama Sekolah/Perguruan Tinggi Contoh: SMA 1 Jakarta	
Telp Sekolah/Perguruan Tinggi 0823 3456 7890	Email Sekolah/Perguruan Tinggi example@example.com
Alamat Sekolah/Perguruan Tinggi Alamat lengkap	
Jenjang Pilih Jenjang	
Tanggal Mulai Prakerin/PK hh/bb/tttt	Tanggal Selesai Prakerin/PK hh/bb/tttt
Upload Surat Pernyataan Pilih file Browse Format: PDF, DOC, atau DOCX	
Upload Proposal Pilih file Browse Format: PDF, DOC, atau DOCX	
Cekah Surat Pernyataan	Cekah Proposal

Gambar 4.10 Dashboard Pendaftaran Peserta Bagian Atas

Berikut gambar 4.11 adalah tampilan *dashboard* pendaftaran peserta bagian bawah:

Data Peserta

Nama Peserta Nama Peserta	
NIS/NIK 12345678	Jurusan Contoh: Teknik Informatika
Email Peserta example@example.com	Telp Peserta 0823 3456 7890
Tambah Orang	Kirim

Website ini merupakan sistem informasi mengenai layanan prakerin/pk, baik itu dari informasi layanan, pendaftaran, hingga data prakerin/pk yang dimiliki Balai Tekkomdik DIY tahun 2024.

Tautan
Balai Tekkomdik
JagabELAJAR
Diklat
MLS
DevKarin

Alamat
Jl. Kenari No.2, Semaki,
Kec. Umbulharjo, Kota Yogyakarta

Gambar 4.11 Dashboard Pendaftaran Peserta Bagian Bawah

2. Realisasi Fitur Pengaturan Kehadiran

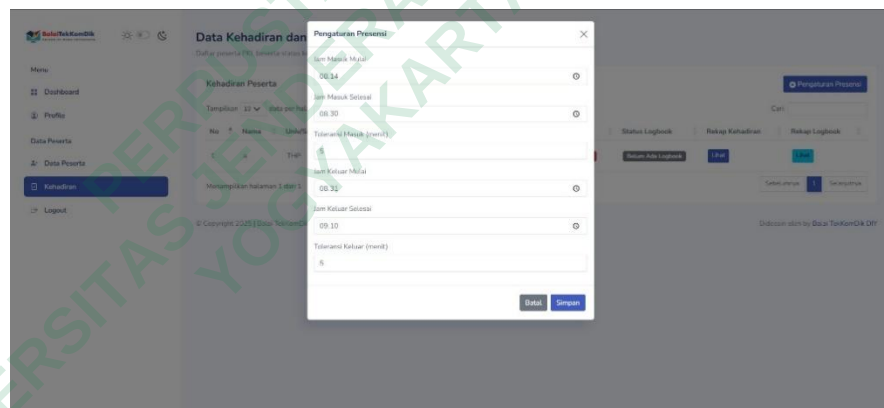
Fitur pengaturan kehadiran memungkinkan pembimbing menentukan rentang jam masuk dan keluar peserta beserta toleransi keterlambatan. Nilai-nilai waktu yang diinput dikonversi ke format durasi menggunakan fungsi `time_to_timedelta` agar penyimpanan data berlangsung seragam dan sesuai kebutuhan sistem. Data yang meliputi jam masuk mulai dan selesai, jam keluar mulai dan selesai, serta batas toleransi, kemudian disimpan ke dalam tabel `pengaturan_kehadiran` melalui perintah `INSERT`. ID pembimbing yang sedang `login` diambil dari sesi aktif dan disertakan sebagai referensi

agar pengaturan bersifat personal dan tidak berlaku umum. Berikut kode program 4.2 adalah realisasi pengaturan kehadiran:

Kode Program 4.2 Pengaturan Kehadiran

```
# Insert pengaturan baru
cur.execute("""
    INSERT INTO pengaturan_kehadiran
    (id_pembimbing, jam_masuk_mulai,
    jam_masuk_selesai, jam_keluar_mulai,
    jam_keluar_selesai, toleransi_masuk,
    toleransi_keluar)
    VALUES (%s, %s, %s, %s, %s, %s, %s)
""", (session['id_pembimbing'],
time_to_timedelta(jam_masuk_mulai),
time_to_timedelta(jam_masuk_selesai),
time_to_timedelta(jam_keluar_mulai),
time_to_timedelta(jam_keluar_selesai),
toleransi_masuk, toleransi_keluar))
```

Berikut gambar 4.12 adalah tampilan *dashboard* pengaturan kehadiran:



Gambar 4.12 Dashboard Pengaturan Kehadiran

3. Realisasi Fitur Pengisian Kehadiran Harian

Proses pengisian kehadiran harian dimulai dengan pengambilan data peserta aktif beserta pengaturan waktu kehadiran yang telah ditentukan oleh pembimbing. Perintah `SELECT` digunakan untuk mengambil informasi dari tabel peserta dan pengaturan_kehadiran secara bersamaan melalui relasi berdasarkan `id_pembimbing`. Data yang diperoleh mencakup rentang jam masuk dan keluar serta toleransi keterlambatan, yang akan

menjadi acuan sistem dalam menentukan status kehadiran peserta. Pengambilan data ini bersifat dinamis, karena hanya memproses peserta yang masih berstatus aktif dan memiliki pembimbing dengan pengaturan kehadiran yang *valid*.

Setelah data peserta berhasil diperoleh, sistem melanjutkan proses dengan memperbarui catatan kehadiran pada tabel kehadiran. Terdapat dua eksekusi UPDATE yang masing-masing berfungsi untuk mencatat waktu masuk dan waktu keluar peserta. Pada proses ini, nilai `jam_masuk` atau `jam_keluar` akan diisi berdasarkan waktu saat peserta melakukan *input*, sementara `status_masuk` atau `status_keluar` disesuaikan dengan hasil evaluasi sistem terhadap waktu *input* dibandingkan dengan batas toleransi yang berlaku. Kolom `updated_at` juga diperbarui secara otomatis menggunakan fungsi `NOW()` untuk merekam waktu aktual saat pembaruan dilakukan. Berikut kode program 4.3 adalah realisasi pengisian kehadiran harian:

Kode Program 4.3 Pengisian Kehadiran Harian

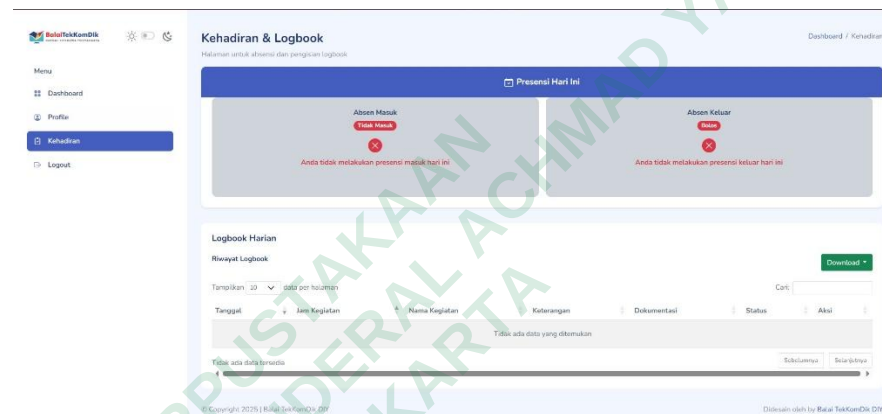
```
# Ambil data peserta dan pengaturan kehadiran
cursor = mysql.connection.cursor(DictCursor)
cursor.execute("""
    SELECT      p.*,          pp.jam_masuk_mulai,
    pp.jam_masuk_selesai, pp.toleransi_masuk,
                  pp.jam_keluar_mulai,
    pp.jam_keluar_selesai, pp.toleransi_keluar
    FROM peserta p
    JOIN pengaturan_kehadiran pp ON p.id_pembimbing
    = pp.id_pembimbing
    WHERE p.id_peserta = %s AND p.status = 'Aktif'
    """, (id_peserta,))
data_peserta = cursor.fetchone()

# Update kehadiran
cursor.execute("""
    UPDATE kehadiran
    SET jam_masuk = %s,
        status_masuk = %s,
        updated_at = NOW()
    WHERE id_kehadiran = %s
    """, (current_time,          status_masuk,
    kehadiran['id_kehadiran']))
```



```
# Update kehadiran
cursor.execute("""
    UPDATE kehadiran
    SET jam_keluar = %s,
    status_keluar = %s,
    updated_at = NOW()
    WHERE id_kehadiran = %s
    """, (current_time, status_keluar,
    kehadiran['id_kehadiran']))
```

Berikut gambar 4.13 adalah tampilan *dashboard* pengisian kehadiran harian:



Gambar 4.13 Dashboard Pengisian Kehadiran Harian

4. Realisasi Pencatatan *Logbook*

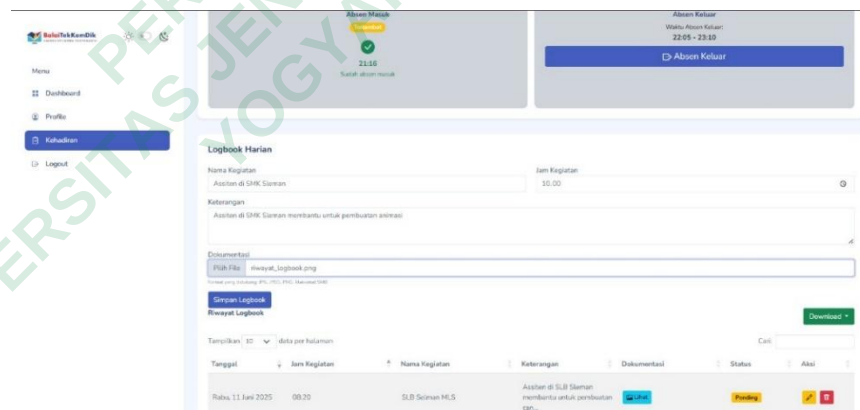
Fitur pencatatan *logbook* berfungsi untuk merekam aktivitas peserta berdasarkan kehadiran yang sudah tercatat pada hari yang sama. Sistem terlebih dahulu memverifikasi keberadaan data kehadiran melalui kondisi *if* kehadiran, dan jika *valid*, akan mengambil *id_kehadiran* untuk dihubungkan dengan entri *logbook*. Informasi yang disimpan mencakup ID peserta, ID kehadiran, tanggal, durasi kegiatan, nama aktivitas, keterangan, dan *file* dokumentasi, yang dimasukkan melalui perintah `INSERT INTO logbook`. Setelah data berhasil disimpan dan dikunci menggunakan `mysql.connection.commit()`, sistem mengirimkan respons sukses dalam format JSON. Sebaliknya, jika data kehadiran tidak ditemukan atau terjadi kesalahan selama proses, sistem akan mengembalikan

pesan kegagalan melalui mekanisme penanganan *error* yang telah disiapkan. Berikut kode program 4.4 adalah realisasi pencatatan *logbook*:

Kode Program 4.4 Pencatatan Logbook

```
if kehadiran:
    id_kehadiran = kehadiran['id_kehadiran']
    cur.execute("""
        INSERT INTO Logbook (id_peserta, id_kehadiran,
        tanggal, jam_kegiatan, nama_kegiatan, keterangan,
        dokumentasi)
        VALUES (%s, %s, %s, %s, %s, %s, %s)
        """, (id_peserta, id_kehadiran, tanggal,
        jam_kegiatan, nama_kegiatan, keterangan, filename))
    mysql.connection.commit()
    return jsonify({'success': True, 'message':
    'Logbook berhasil ditambahkan'})
else:
    return jsonify({'success': False, 'message':
    'Data kehadiran tidak ditemukan'})
except Exception as e:
    return jsonify({'success': False, 'message':
    str(e)})
```

Berikut gambar 4.14 adalah tampilan *dashboard* pencatatan *logbook*:



Gambar 4.14 Dashboard Pencatatan Logbook

4.5.2 IMPLEMENTASI *SPRINT 2*

Sprint kedua difokuskan pada penyempurnaan fitur utama yang telah dikembangkan pada *sprint* sebelumnya serta penambahan fungsi pendukung untuk memperkuat kelengkapan sistem. Fitur baru yang berhasil diimplementasikan meliputi *dashboard* peserta dan admin, rekap kehadiran dan *logbook* yang dapat diakses oleh pembimbing dan

admin, serta pengelolaan data *logbook* yang lebih terstruktur. Selain itu, sistem mengalami peningkatan dari sisi antarmuka, mencakup penyesuaian tampilan *form* pendaftaran dan perbaikan responsivitas agar lebih nyaman diakses melalui berbagai perangkat. Logika pembatasan hak akses berdasarkan peran juga mulai diterapkan, memastikan bahwa fitur hanya dapat diakses sesuai peran pengguna. Seluruh fitur tambahan ini berhasil terintegrasi tanpa mengganggu stabilitas sistem yang telah dibangun pada *sprint* sebelumnya, dan dinyatakan layak untuk digunakan pada tahap uji pengguna berikutnya. Adapun penjelasan lebih lanjut mengenai realisasi beberapa fitur penting yang dikembangkan pada *sprint* kedua dapat diuraikan sebagai berikut:

1. Realisasi Rekap Kehadiran Peserta oleh Admin dan Pembimbing

Fitur rekap kehadiran peserta berfungsi untuk menampilkan riwayat presensi secara lengkap berdasarkan data yang tersimpan di basis data. Sistem melakukan pengambilan data melalui perintah `SELECT` dengan menggabungkan tiga tabel, yaitu kehadiran, peserta, dan institusi, guna memperoleh informasi yang komprehensif mencakup identitas peserta dan institusi asalnya. Data difilter berdasarkan ID peserta dan diurutkan berdasarkan tanggal kehadiran dalam urutan paling baru ke paling lama menggunakan fungsi `ORDER BY k.tanggal DESC`. Setelah data berhasil diambil, sistem memformat nilai tanggal, jam masuk, dan jam keluar dengan fungsi `format_tanggal` dan `format_waktu`, sehingga informasi yang ditampilkan lebih konsisten dan mudah dipahami. Kolom keterangan juga disesuaikan agar menampilkan tanda strip apabila tidak diisi, sehingga hasil rekap menjadi lebih rapi, informatif, dan siap ditampilkan dalam antarmuka pengguna. Berikut kode program 4.5 adalah realisasi rekap kehadiran peserta oleh admin dan pembimbing:

Kode Program 4.5 Rekap Kehadiran Peserta

```
cur.execute("""
SELECT k.*, p.nama, i.nama_institusi
FROM kehadiran k
JOIN peserta p ON k.id_peserta = p.id_peserta
JOIN institusi i ON p.id_institusi =
i.id_institusi
WHERE k.id_peserta = %s
ORDER BY k.tanggal DESC
""", (id_peserta,))
kehadiran = cur.fetchall()
formatted_data = []
for k in kehadiran:
    formatted_data.append({
        'id_kehadiran': k['id_kehadiran'],
        'tanggal': format_tanggal(k['tanggal']),
        'jam_masuk': format_waktu(k['jam_masuk']),
        'jam_keluar': format_waktu(k['jam_keluar']),
        'status_masuk': k['status_masuk'],
        'status_keluar': k['status_keluar'],
        'keterangan': k['keterangan'] if
k['keterangan'] is not None else '-'
    })
```

Berikut gambar 4.15 adalah tampilan *dashboard* rekap kehadiran peserta oleh admin dan pembimbing:

Tanggal	Jam Masuk	Status Masuk	Jam Keluar	Status Keluar	Keterangan	Aksi
26-06-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
27-06-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
28-06-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
29-06-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
30-06-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
01-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
02-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
03-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
04-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
05-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
06-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
07-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
08-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
09-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
10-07-2025	-	Belum Masuk	-	Belum Keluar	-	✓ ✕
11-07-2025	-	Tidak Masuk	-	Belum Keluar	-	✓ ✕

Gambar 4.15 Dashboard Rekap Kehadiran Peserta

2. Realisasi Rekap Logbook oleh Admin dan Pembimbing

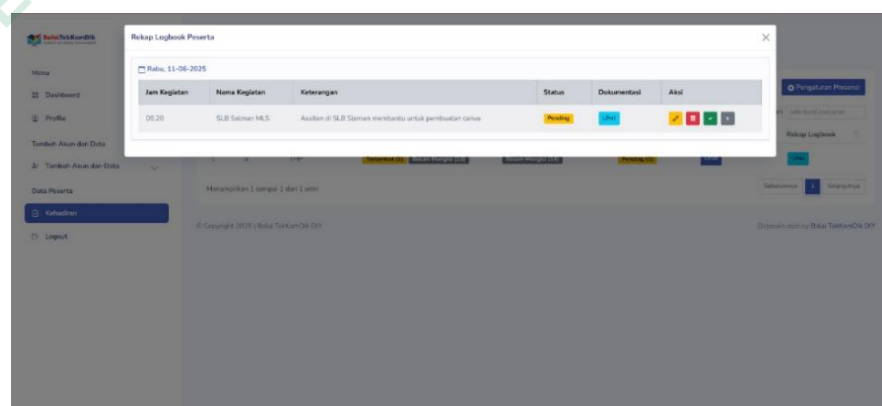
Fitur rekap *logbook* peserta digunakan untuk menampilkan rangkuman aktivitas harian yang telah dicatat selama masa pelaksanaan praktik. Sistem mengambil data *logbook* berdasarkan ID peserta menggunakan perintah `SELECT`, dengan bantuan fungsi

DATE_FORMAT dan TIME_FORMAT untuk menyesuaikan tampilan tanggal dan jam kegiatan agar lebih mudah dibaca oleh pengguna. Informasi yang disajikan mencakup tanggal, jam kegiatan, nama aktivitas, keterangan, status verifikasi, dokumentasi, serta ID *logbook* sebagai identifikasi unik setiap entri. Data diurutkan berdasarkan tanggal terbaru terlebih dahulu (*descending*), sehingga entri paling akhir akan muncul di bagian atas. Apabila terdapat beberapa catatan dalam satu hari, sistem akan menampilkan urutannya berdasarkan jam kegiatan dari paling awal ke paling akhir (*ascending*). Berikut kode program 4.6 adalah realisasi rekap *logbook* peserta:

Kode Program 4.6 Rekap Logbook Peserta

```
cur.execute("""
SELECT
    DATE_FORMAT(tanggal, '%d-%m-%Y') as tanggal,
    TIME_FORMAT(jam_kegiatan, '%H:%i') as
jam_kegiatan,
    nama_kegiatan,
    keterangan,
    status,
    dokumentasi,
    id_Logbook
FROM Logbook
WHERE id_peserta = %s
ORDER BY tanggal DESC, jam_kegiatan ASC
""", (id_peserta,))
```

Berikut gambar 4.16 adalah tampilan *dashboard* rekap *logbook*:



Gambar 4.16 Dashboard Rekap Logbook

3. Realisasi Revisi dan Konfirmasi Pendaftaran Peserta

a. Revisi Data Pendaftaran Peserta

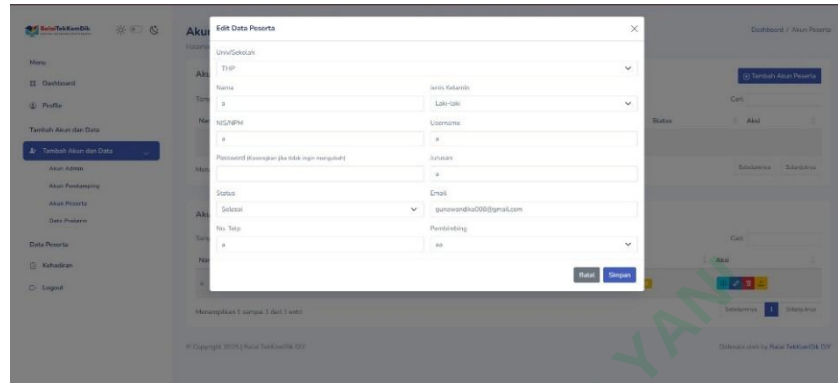
Fitur revisi data pendaftaran peserta memungkinkan admin memperbarui informasi peserta yang telah terdaftar dalam sistem. Proses pembaruan dilakukan melalui perintah UPDATE terhadap tabel peserta, dengan parameter yang mencakup nama, NIS/NPM, username, jurusan, status, *email*, dan nomor telepon. Jika pengguna turut mengubah kata sandi, sistem akan menambahkan kolom *password* dalam *query* dan mengenkripsi nilai yang dimasukkan menggunakan fungsi `generate_password_hash` dari pustaka `werkzeug.security`, guna menjaga keamanan data autentikasi. Selain itu, apabila terdapat pembaruan *file* sertifikat, data tersebut juga akan disisipkan dalam *query* sebagai parameter tambahan. Perintah WHERE ditambahkan di akhir *query* untuk memastikan bahwa pembaruan hanya dilakukan terhadap peserta dengan ID yang sesuai. Berikut kode program 4.7 adalah realisasi revisi data pendaftaran peserta:

Kode Program 4.7 Revisi Data Pendaftaran Peserta

```
# Update data peserta
update_query = ""
UPDATE peserta
SET nama = %s, nis_npm = %s, username = %s,
jurusan = %s, status = %s,
email = %s, no_telp = %s
""

params = [nama, nis_npm, username, jurusan,
status, email, no_telp]
if password:
    from werkzeug.security import
generate_password_hash
    update_query += ", password = %s"
    params.append(generate_password_hash(password))
if new_sertifikat:
    update_query += ", sertifikat = %s"
    params.append(new_sertifikat)
update_query += " WHERE id_peserta = %s"
params.append(id_peserta)
cur.execute(update_query, tuple(params))
```

Berikut gambar 4.17 adalah tampilan *dashboard* revisi data pendaftaran peserta:



Gambar 4.17 Dashboard Revisi Data Pendaftaran Peserta

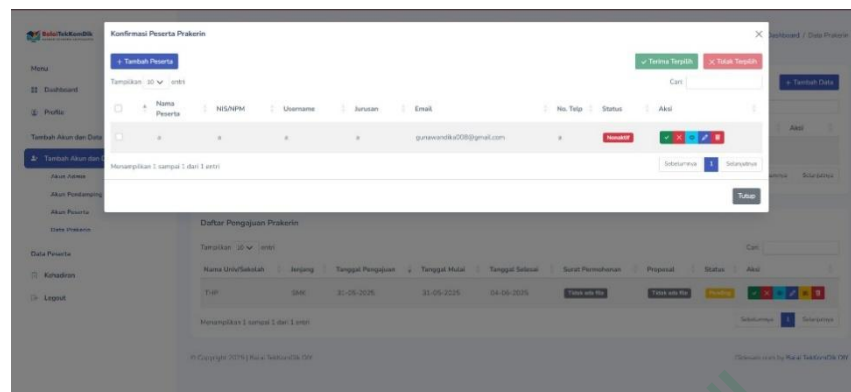
b. Konfirmasi Pendaftaran Peserta

Fitur konfirmasi pendaftaran peserta berfungsi untuk mengubah status pendaftaran calon peserta menjadi aktif, ditolak, atau bentuk status lainnya sesuai keputusan admin. Proses ini dilakukan melalui perintah UPDATE pada tabel peserta, yang memperbarui nilai kolom status berdasarkan ID peserta yang bersangkutan. Eksekusi perintah ini memungkinkan sistem mencatat hasil verifikasi admin secara langsung dalam basis data, sehingga status terbaru peserta dapat ditampilkan secara *real-time* pada antarmuka pengguna. Berikut kode program 4.8 adalah realisasi konfirmasi pendaftaran peserta:

Kode Program 4.8 Konfirmasi Pendaftaran Peserta

```
cur.execute("""
    UPDATE peserta
    SET status = %s
    WHERE id_peserta = %s
    """, (status, id_peserta))
```

Berikut gambar 4.18 adalah tampilan *dashboard* konfirmasi pendaftaran peserta:



Gambar 4.18 Dashboard Konfirmasi Pendaftaran Peserta

4.5.3 IMPLEMENTASI *SPRINT* 3

Sprint ketiga difokuskan pada penyempurnaan akhir sistem dan penambahan fitur pendukung proses evaluasi PKL. Pengembangan mencakup monitoring *logbook* oleh pembimbing, unggah sertifikat oleh admin, peserta mengunduh sertifikat, serta pengelolaan akun pembimbing dan peserta bimbingan. Hasil *sprint* menunjukkan bahwa seluruh target pengembangan telah tercapai dan sistem siap digunakan pada tahap implementasi. Adapun penjelasan lebih lanjut mengenai realisasi beberapa fitur penting yang dikembangkan pada *sprint* ketiga dapat diuraikan sebagai berikut:

1. Realisasi Unggah Sertifikat Peserta

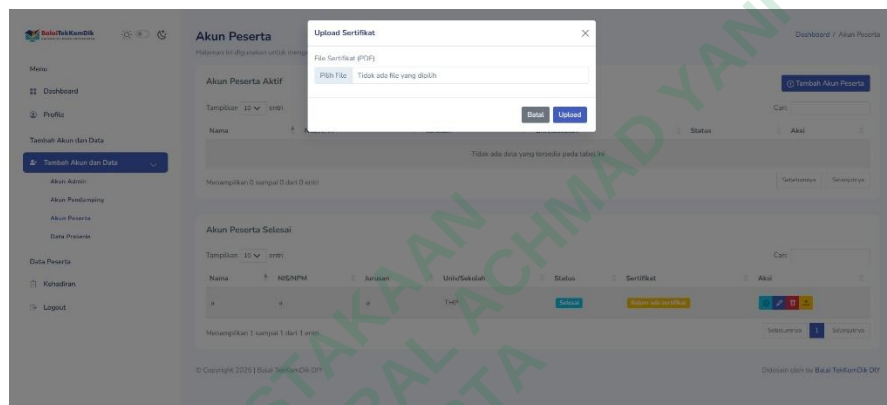
Fitur unggah sertifikat peserta berfungsi untuk menyimpan dokumen sertifikat ke dalam sistem dan mengaitkannya dengan peserta yang bersangkutan. Proses ini dilakukan melalui perintah `UPDATE` pada tabel peserta, yakni nilai kolom sertifikat diisi dengan jalur relatif *file* yang telah diunggah. Eksekusi *query* ini menggunakan parameter `id_peserta` sebagai acuan, sehingga *file* disimpan tepat pada entri peserta yang dituju. Perintah `mysql.connection.commit()` kemudian dijalankan untuk memastikan data perubahan tersimpan secara permanen dalam basis

data. Berikut kode program 4.9 adalah realisasi unggah sertifikat peserta:

Kode Program 4.9 Unggah Sertifikat Peserta

```
cur.execute("UPDATE peserta SET sertifikat = %s WHERE
id_peserta = %s", (relative_path, id_peserta))
mysql.connection.commit()
```

Berikut gambar 4.19 adalah tampilan *dashboard* unggah sertifikat peserta oleh admin:



Gambar 4.19 Dashboard Unggah Sertifikat Peserta

2. Realisasi Peserta Mengunduh Sertifikat

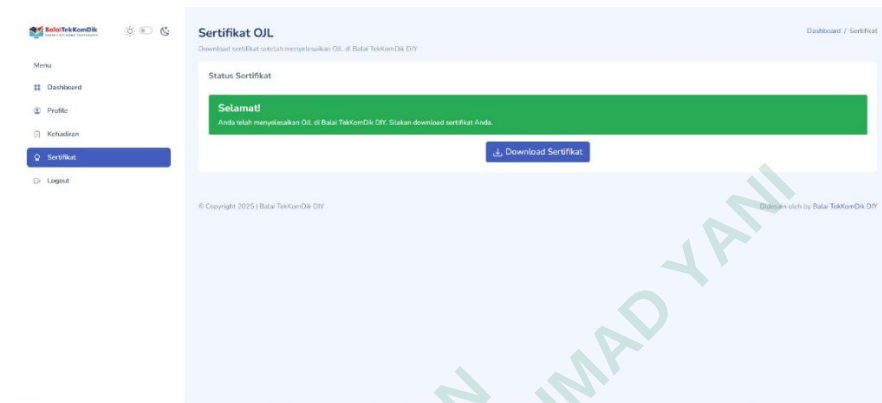
Fitur pengunduhan sertifikat memungkinkan peserta mengakses dan menyimpan dokumen sertifikat yang telah diunggah oleh admin atau pembimbing. Proses ini dijalankan menggunakan fungsi `send_file`, yang mengarahkan sistem untuk mengirim *file* sertifikat secara langsung kepada pengguna dalam bentuk lampiran (*attachment*). Parameter `sertifikat_path` menunjukkan lokasi *file* yang akan diunduh, sementara `download_name` digunakan untuk memberi nama *file* secara aman saat diunduh melalui *browser*. Format MIME ditetapkan sebagai `'application/pdf'` untuk memastikan kompatibilitas *file* dengan perangkat pengguna. Berikut kode program 4.10 adalah realisasi peserta mengunduh sertifikat:

Kode Program 4.10 Peserta Mengunduh Sertifikat

```
return send_file(
sertifikat_path,
as_attachment=True,
download_name=safe_filename,
```

```
mimetype='application/pdf'
)
```

Berikut gambar 4.20 adalah tampilan *dashboard* peserta mengunduh sertifikat:



Gambar 4.20 Dashboard Peserta Mengunduh Sertifikat

4.6 IMPLEMENTASI TESTING

Pengujian sistem dilakukan menggunakan metode *black box testing* guna mengevaluasi kesesuaian fungsional setiap fitur berdasarkan respons terhadap berbagai kombinasi *input* yang diberikan, tanpa menelusuri struktur internal dari kode program. Fokus pengujian diarahkan pada skenario penggunaan riil yang mencerminkan aktivitas pengguna sistem, seperti proses *login*, pendaftaran peserta, pengisian kehadiran, pencatatan *logbook*, verifikasi dan rekap data, serta pengunggahan sertifikat. Seluruh fitur diuji menggunakan kombinasi data yang sesuai maupun tidak sesuai ketentuan untuk memastikan sistem mampu memberikan keluaran yang tepat terhadap setiap kondisi. Hasil pengujian menunjukkan bahwa setiap fitur inti berjalan sesuai rancangan, menghasilkan *output* yang konsisten, dan mampu menangani kesalahan input dengan menampilkan pesan validasi yang informatif. Keberhasilan sistem dalam merespons berbagai skenario tersebut mencerminkan bahwa alur logika dan implementasi fitur telah diterapkan secara efektif selama proses pengembangan.

Secara keseluruhan, pengujian terhadap fitur-fitur utama menunjukkan bahwa seluruh komponen sistem telah berfungsi sebagaimana mestinya. Fitur

login berhasil memvalidasi kombinasi kredensial pengguna secara akurat, termasuk saat terjadi kesalahan pada kata sandi maupun ketika akun belum terdaftar. Proses pendaftaran dapat membedakan data baru dan menolak pendaftaran yang menggunakan *email* yang telah terdaftar sebelumnya. Fitur kehadiran dan *logbook* merespons entri *valid* dengan baik, serta menampilkan pesan kesalahan saat kolom wajib tidak diisi. Pengelolaan akun pembimbing, penugasan peserta, dan tampilan *dashboard* berdasarkan peran pengguna berjalan sesuai hak akses yang ditentukan. Selain itu, sistem mampu membatasi akses terhadap halaman yang tidak sesuai peran, serta mendukung proses *logout* dan reset kata sandi dengan lancar. Berdasarkan hasil tersebut, dapat disimpulkan bahwa sistem telah memenuhi seluruh kriteria fungsional dan siap digunakan dalam konteks operasional sesungguhnya.

4.7 IMPLEMENTASI USER ACCEPTANCE TEST (UAT)

User Acceptance Test (UAT) dilakukan oleh pengguna akhir sistem, yaitu peserta PKL, admin, dan pembimbing, sebagai perwakilan dari pihak Balai Tekkomdik DIY. UAT bertujuan untuk mengevaluasi tingkat kesesuaian sistem dengan kebutuhan nyata pengguna serta menilai keberterimaan sistem dari sisi kemudahan penggunaan, kelengkapan fitur, dan kejelasan alur. Indikator dalam UAT ini meliputi aspek:

a. Fleksibel:

- P1 : Sistem dapat diakses 24/7 dari mana saja tanpa perlu hadir ke kantor Balai Tekkomdik.
- P2 : Pendaftaran PKL sepenuhnya daring, tanpa tatap muka.
- P3 : Pengisian *logbook*, verifikasi, dan *monitoring* dapat dilakukan kapan pun.
- P4 : Sistem kompatibel dengan berbagai perangkat (HP, laptop, tablet).
- P5 : Koordinasi antar peran (peserta, admin, pembimbing) tidak memerlukan pertemuan langsung.

b. Adaptif:

- P6 : Alur sistem selaras dengan prosedur PKL di Balai Tekkomdik.

- P7 : Setiap fitur dikembangkan berdasarkan masukan pengguna.
- P8 : Setiap umpan balik pengguna diakomodasi oleh tim pengembang.
- P9 : Hak akses dan wewenang disesuaikan dengan struktur organisasi.
- P10 : Data dan laporan yang disajikan mendukung kebutuhan administrasi.

c. Responsif:

- P11 : Tim pengembang menindaklanjuti masukan pengguna dengan cepat.
- P12 : Pengguna dilibatkan dalam evaluasi dan uji coba sebelum implementasi.
- P13 : Sistem diperbaiki segera, setelah menerima umpan balik.

Berdasarkan hasil pengisian *User Acceptance Test* (UAT) oleh para responden, dapat disimpulkan tingkat penerimaan sistem yang direpresentasikan melalui rekap data pada tabel 4.12 berikut ini:

Tabel 4.12 Rekap Jumlah Hasil Pengujian UAT

Pertanyaan	Sangat Tidak Setuju	Tidak Setuju	Setuju	Sangat Setuju
P1	0	0	0	9
P2	0	0	1	8
P3	0	0	1	8
P4	0	0	1	8
P5	0	0	1	8
P6	0	0	1	8
P7	0	0	1	8
P8	0	0	1	8
P9	0	0	1	8

Pertanyaan	Sangat Tidak Setuju	Tidak Setuju	Setuju	Sangat Setuju
P10	0	0	1	8
P11	0	0	1	8
P12	0	0	1	8
P13	0	0	1	8

Tabel 4.12 menunjukkan rekap hasil UAT dari 13 pertanyaan yang diajukan menggunakan skala *Likert* 4 poin (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju). Hasilnya menunjukkan bahwa seluruh responden memberikan jawaban pada kategori Setuju atau Sangat Setuju, tanpa adanya tanggapan negatif. Sebanyak 8 dari 9 responden ($\pm 89\%$) menyatakan Sangat Setuju pada seluruh indikator, dan 1 responden ($\pm 11\%$) menyatakan Setuju pada sebagian besar indikator.

Hasil ini menunjukkan bahwa sistem telah memenuhi harapan pengguna dalam aspek fleksibilitas penggunaan, adaptasi terhadap kebutuhan instansi, dan responsivitas pengembang terhadap masukan. UAT juga memperkuat temuan sebelumnya bahwa sistem berfungsi secara teknis (melalui *black box testing*), sekaligus diterima secara fungsional oleh pengguna. Temuan ini mendukung kesimpulan bahwa metode *Agile Scrum* yang digunakan dalam pengembangan sistem telah berhasil mendorong keterlibatan pengguna serta menghasilkan produk yang relevan dan aplikatif di lingkungan Balai Tekkomdik DIY.

4.8 IMPLEMENTASI SPRINT REVIEW

Sprint review dilaksanakan di akhir setiap siklus pengembangan sebagai forum evaluasi terhadap fitur yang telah diimplementasikan dan untuk menerima umpan balik langsung dari *stakeholder*. Pada akhir *sprint* pertama, tim mempresentasikan fitur *login*, pendaftaran peserta, pengaturan kehadiran, pengisian kehadiran harian, dan pencatatan *logbook*. *Stakeholder* menyatakan

bahwa seluruh fitur berjalan sesuai kebutuhan awal, namun menyarankan adanya penambahan fitur validasi dan struktur pengelolaan akun yang lebih tertata. Saran ini direspons pada *sprint* kedua melalui implementasi *dashboard* peserta dan admin, rekap kehadiran dan *logbook*, serta pengaturan hak akses berdasarkan peran pengguna. Masukan pada tahap ini mendorong perbaikan pada tampilan antarmuka agar lebih intuitif dan penambahan peringatan sistem saat data belum lengkap. *Sprint* ketiga difokuskan pada integrasi fitur unggah dan unduh sertifikat serta pemilihan pembimbing yang berhasil diterapkan secara menyeluruh. Fitur-fitur tersebut diuji kembali bersama *stakeholder* untuk memastikan bahwa seluruh kebutuhan operasional telah terpenuhi secara end-to-end. Terakhir, *stakeholder* menyatakan bahwa sistem telah sesuai dengan kebutuhan lapangan dan layak digunakan sebagai *platform* resmi dalam pengelolaan administrasi peserta PKL di Balai Tekkomdik DIY.