



**RANCANG BANGUN APLIKASI PENCATATAN PESERTA KURSUS
PADA PUEDE EDUCATION**

KERJA PRAKTIK



Program Studi

S1 Sistem Informasi

**UNIVERSITAS
Dinamika**

Oleh:

DAVE LIANATA

22410100050

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2026

**RANCANG BANGUN APLIKASI PENCATATAN PESERTA KURSUS
PADA PUEDE EDUCATION**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana



Disusun Oleh:

Nama : Dave Lianata

NIM : 22410100050

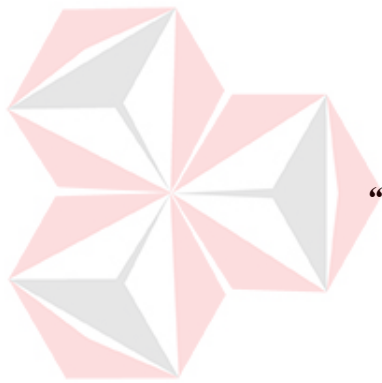
Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2026



Amsal 1 : 7a

“Takut akan Tuhan adalah permulaan pengetahuan”

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI PENCATATAN PESERTA KURSUS
PADA PUEDE EDUCATION**

Laporan Kerja Praktik

oleh:

Dave Lianata

NIM. 22410100050



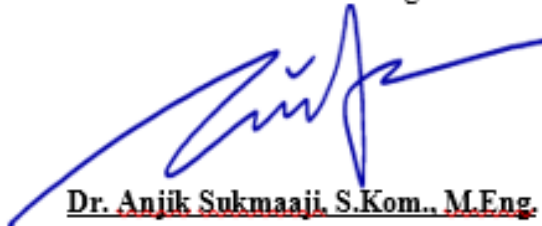
Telah diperiksa, diuji, dan disetujui

Surabaya, 10 Desember 2025

Disetujui

Dosen Pembimbing

Penyelia,



Dr. Anjik Sukmaaji, S.Kom., M.Eng.



Jimmy Gunawan

NIDN. 0731057301

Mengetahui,

Ketua Program Studi S1 Sistem Informasi



Digitally signed by
Endra Rahmawati
Date: 2026.01.15
09:27:57 +07'00'

Endra Rahmawati, M.Kom.

NIDN. 0712108701

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : Dave Lianata
NIM : 22410100050
Program Studi : SI Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Kerja Praktik
Judul Karya : RANCANG BANGUN APLIKASI PENCATATAN
PESERTA KURSUS PADA PUEDE EDUCATION

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 12 Januari 2026



Dave Lianata
NIM : 22410100050

ABSTRAK

Dalam era digital saat ini, efisiensi operasional menjadi faktor krusial bagi lembaga pendidikan, termasuk kursus seperti Puede Education yang masih mengandalkan sistem pencatatan manual berbasis Excel. Sistem manual ini rentan terhadap kesalahan input, kehilangan data, dan ketidakefisienan dalam penyusunan laporan, sehingga menghambat proses administrasi. Tujuan dari kerja praktik ini adalah merancang dan mengimplementasikan aplikasi pencatatan peserta kursus berbasis *web* untuk menggantikan sistem manual tersebut. Metode yang digunakan adalah *Waterfall*, yang meliputi tahapan analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Aplikasi ini dikembangkan menggunakan *PHP*, *MySQL*, *HTML*, *CSS*, dan *Javascript*, dengan fitur utama meliputi pengelolaan data peserta dan kelas serta pembuatan laporan dalam format *PDF*. Hasil pengujian *Blackbox* menunjukkan bahwa semua fitur berfungsi dengan baik, termasuk entri, edit, hapus data, dan pencetakan laporan. Kesimpulannya, aplikasi ini berhasil meningkatkan efisiensi operasional, mengurangi kesalahan pencatatan, dan memudahkan administrasi di Puede Education, sekaligus menjadi solusi modern untuk manajemen data peserta kursus.

Kata kunci: Aplikasi *web*, Manajemen data, *Waterfall*, *PHP*, *MySQL*, *HTML*, *CSS*, *Javascript*, *Blackbox testing*.

KATA PENGANTAR

Puji dan syukur saya panjatkan ke hadirat Tuhan Yesus Kristus atas kasih karunia dan penyertaan-Nya yang senantiasa memampukan saya dalam menyelesaikan laporan Kerja Praktek yang berjudul “Perancangan Aplikasi Pencatatan Peserta Kursus Pada Puede Education” Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan program Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Dinamika. Dalam proses penyusunan laporan ini, saya menyadari bahwa segala pencapaian tidak lepas dari dukungan berbagai pihak yang telah memberikan doa, semangat, motivasi, arahan, serta kritik dan saran yang sangat berarti.

Saya menyampaikan terima kasih yang sebesar-besarnya kepada:

1. kedua orang tua saya tercinta yang selalu memberikan dukungan moral dan spiritual, serta menjadi sumber kekuatan dalam menyelesaikan Kerja Praktik ini.
2. Ucapan terima kasih juga saya sampaikan kepada Ibu Endra Rahmawati, M.Kom., selaku Ketua Program Studi Sistem Informasi dan dosen pembimbing, yang telah memberikan bimbingan, saran, serta waktu dalam proses penyusunan laporan ini.
3. Saya juga berterima kasih kepada Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng. selaku dosen pembahas, yang telah memberikan masukan dan koreksi yang membangun demi kesempurnaan laporan ini.
4. Tidak lupa saya mengucapkan terima kasih kepada seluruh dosen dan staf Universitas Dinamika yang telah mendukung proses pembelajaran saya.

5. Serta kepada teman-teman dan semua pihak yang telah membantu secara langsung maupun tidak langsung dalam penyelesaian Kerja Praktik ini.

Saya menyadari bahwa laporan ini masih memiliki banyak kekurangan, oleh karena itu, segala bentuk kritik dan saran yang bersifat membangun sangat saya harapkan demi perbaikan di masa mendatang. Semoga laporan Kerja Praktik ini dapat memberikan manfaat dan menjadi bahan referensi yang berguna bagi pembaca. Segala pujian dan hormat hanya bagi Tuhan Yesus Kristus yang telah memberikan kekuatan dan hikmat sepanjang proses ini.

Surabaya, Januari 2026

Dave Lianata



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
BAB II GAMBARAN UMUM PERUSAHAAN	4
2.1. Profil Perusahaan.....	4
2.2. Identitas Perusahaan	6
2.3. Visi Perusahaan	6
2.4. Misi Perusahaan.....	6
2.5. Struktur Organisasi.....	7
BAB III LANDASAN TEORI.....	9
3.1. Pendaftaran Kursus Puede Education.....	9
3.2. Website	10
3.3. Database.....	11
3.4. PHP	12

3.5.	Metode <i>Waterfall</i>	12
3.6.	<i>Blackbox Testing</i>	13
BAB IV DESKRIPSI PEKERJAAN		14
4.1.	Metode <i>Waterfall</i>	14
4.2.	Analisis Kebutuhan	14
4.3.	Desain.....	15
4.4.	Use Case Sistem	15
4.5.	<i>Activity Diagram</i>	17
4.6.	Sequence Diagram.....	26
4.7.	User Interfaces.....	34
4.8.	Pengkodean/development	44
4.9.	Pengujian	44
BAB V PENUTUP.....		46
5.1.	Kesimpulan.....	46
5.2.	Saran.....	46
DAFTAR PUSTAKA.....		48
LAMPIRAN		49

DAFTAR TABEL

Halaman

Tabel 4. 1 Pengujian Black Box Testing Admin.....	44
---	----



UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Logo Perusahaan.....	4
Gambar 2. 2 Gedung Perusahaan	5
Gambar 2. 3 Struktur Organisasi Puede Education	8
Gambar 4. 1 Metode Waterfall.....	14
Gambar 4. 2 Use Case Sistem	16
Gambar 4. 3 Activity Entri Peserta	18
Gambar 4. 4 Activity Edit Peserta.....	19
Gambar 4. 5 Activity Hapus Peserta	20
Gambar 4. 6 Activity Entri Kelas.....	21
Gambar 4. 7 Activity Edit Kelas	22
Gambar 4. 8 Activity Hapus Kelas.....	24
Gambar 4. 9 Activity Cetak Laporan	25
Gambar 4. 10 Sequence Entri Peserta	26
Gambar 4. 11 Sequence Edit Peserta.....	28
Gambar 4. 12 Sequence Hapus Peserta	29
Gambar 4. 13 Sequence Entri Kelas.....	30
Gambar 4. 14 Sequence Edit Kelas	31
Gambar 4. 15 Sequence Hapus Kelas	32
Gambar 4. 16 Sequence Cetak Laporan	34
Gambar 4. 17 UI Dashboard.....	34
Gambar 4. 18 UI Siswa	35
Gambar 4. 19 UI Entri Siswa	36

Gambar 4. 20 UI Detail Siswa	37
Gambar 4. 21 UI Ubah Siswa.....	38
Gambar 4. 22 UI Kelas.....	39
Gambar 4. 23 UI Entri Kelas.....	40
Gambar 4. 24 UI Detail Kelas	41
Gambar 4. 25 UI Ubah Kelas	42
Gambar 4. 26 UI Laporan.....	42
Gambar 4. 27 UI Cetak Laporan	43



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam era digital saat ini, penggunaan teknologi dalam sektor bisnis menjadi sangat penting untuk meningkatkan efisiensi dan efektivitas operasional. Salah satu sektor yang sangat memerlukan sistem digital adalah industri pendidikan, termasuk tempat kursus. Pude Education, yang berlokasi di Surabaya, masih menggunakan sistem manual menggunakan excel dalam pencatatan peserta kursusnya, yang menyebabkan berbagai kendala seperti kesalahan pencatatan serta kesulitan dalam menyusun laporan pesertanya.

Sistem pencatatan manual yang masih diterapkan di Pude Education sering kali menimbulkan berbagai kendala, seperti kesalahan dalam pencatatan data peserta, kehilangan data peserta serta ketidakefisienan dalam penyusunan laporan, sehingga pemilik usaha kesulitan dalam melakukan analisis.

Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah sistem pencatatan berbasis *web* menggunakan *database MySQL* yang mampu mengelola data peserta secara terstruktur dan efisien. Sistem ini akan dirancang untuk mencatat data peserta kursus, mencatat data kelas serta pembuatan laporan.

Dengan adanya sistem ini, Pude Education diharapkan dapat meningkatkan kecepatan dan ketepatan dalam melayani peserta maupun calon peserta kursus serta mengelola laporan secara lebih efisien.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam Kerja Praktik ini adalah:

1. Bagaimana merancang sistem pencatatan yang dapat digunakan oleh admin untuk memproses pencatatan secara efisien?
2. Bagaimana sistem dapat mencatat data peserta dan kelas secara tepat dan cepat?
3. Bagaimana sistem dapat menghasilkan laporan yang akurat dan dapat dicetak untuk kebutuhan bisnis?

1.3. Batasan Masalah

Agar penelitian ini lebih terarah, batasan masalah yang diterapkan dalam Kerja Praktik ini adalah:

1. Sistem hanya digunakan untuk pencatatan data peserta kursus serta data kelas dan tidak mencakup aspek lain seperti penjadwalan.
2. Sistem akan dibuat berbasis *web* dengan *database MySQL* sebagai penyimpanan data.
3. Fitur utama dalam sistem meliputi:
 - Pengelolaan data siswa
 - Pengelolaan data kelas
 - Penyusunan laporan yang dapat dicetak dalam bentuk *PDF*

1.4. Tujuan

Tujuan dari perancangan sistem ini adalah:

1. Membangun sistem yang dapat membantu admin dan pemilik dalam memproses pencatatan dengan lebih cepat dan akurat.
2. Mempermudah admin dalam mengelola data peserta dan kelas.
3. Menyediakan fitur laporan yang dapat diakses dan dicetak untuk keperluan bisnis.

1.5. Manfaat

Adapun manfaat yang diharapkan dari perancangan sistem ini adalah:

1. Bagi Puede Education:

- Meningkatkan efisiensi operasional dengan sistem yang lebih cepat dan akurat.
- Mengurangi kesalahan pencatatan yang sering terjadi pada sistem manual.

2. Bagi Admin:

- Memudahkan proses pendataan serta mempercepat pelayanan kepada peserta maupun calon peserta kursus.

3. Bagi Pemilik:

- Menyediakan data yang akurat untuk analisis dan pengambilan keputusan.
- Membantu dalam menyusun laporan secara lebih sistematis dan profesional.

4. Bagi Penulis:

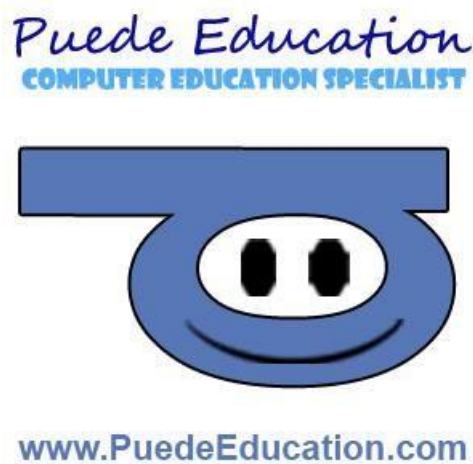
- Menambah wawasan dan pengalaman dalam perancangan serta implementasi sistem berbasis *MySQL*.
- Meningkatkan keterampilan pemrograman dan analisis sistem informasi.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1. Profil Perusahaan

Puede Education merupakan perusahaan yang bergerak di bidang pendidikan dan pelatihan teknologi informasi, yang berlokasi di Jalan Petemon IV / 1-B, Surabaya. Perusahaan ini didirikan dengan semangat untuk menciptakan sumber daya manusia yang unggul dan kompeten di era digital. Sejak awal berdirinya, Puede Education memfokuskan diri pada penyediaan layanan kursus pemrograman dan teknologi informasi yang menasar berbagai kalangan, mulai dari anak-anak, pelajar, mahasiswa, hingga profesional. Dengan pendekatan pengajaran yang sistematis dan aplikatif, serta didukung oleh instruktur yang berpengalaman, Puede Education telah tumbuh menjadi salah satu lembaga pendidikan nonformal yang terpercaya di bidangnya. Logo perusahaan Puede Education dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Logo Perusahaan

Produk utama dari Puede Education berupa kursus dan pelatihan yang mencakup berbagai materi teknologi informasi. Untuk bidang pemrograman dasar, Puede Education menyediakan pengajaran bahasa-bahasa pemrograman populer seperti Pascal, Java, C/C++, C#, *PHP*, R Language, Python, Go Language, VBA, dan *Javascript*. Materi ini dirancang agar dapat diakses oleh pemula sekalipun, dengan penyampaian yang terstruktur dan disesuaikan dengan level kemampuan peserta.

Kantor pusat Puede Education berada di Jalan Petemon IV / 1-B, Surabaya. Gedung kantor terdiri dari beberapa ruang kelas yang dilengkapi dengan fasilitas modern, termasuk komputer dengan spesifikasi terkini, pendingin ruangan, akses internet berkecepatan tinggi, serta ruang diskusi. Lingkungan belajar yang nyaman dan kondusif mendukung proses pelatihan agar berlangsung efektif dan menyenangkan. Foto gedung Puede Education dapat dilihat pada Gambar 2.2.



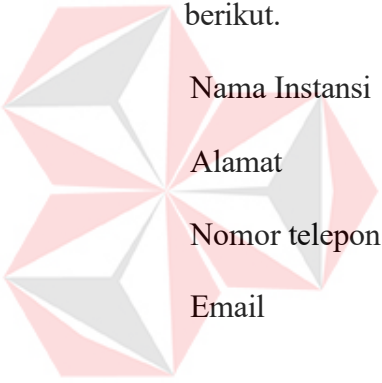
Gambar 2. 2 Gedung Perusahaan

Tenaga pengajar di Puede Education terdiri dari instruktur yang berpengalaman di bidangnya, baik secara akademik maupun praktis. Pengajaran

dilakukan secara sistematis, dengan pendekatan interaktif dan berbasis proyek. Peserta tidak hanya menerima materi secara teori, tetapi juga didorong untuk mengerjakan latihan dan tugas akhir yang dapat menjadi portofolio. Dengan metode pengajaran yang fokus pada pemahaman dan penerapan, Puede Education berkomitmen mencetak lulusan yang siap menghadapi tantangan dunia kerja dan industri digital yang terus berkembang.

2.2. Identitas Perusahaan

Identitas perusahaan yang diinformasikan meliputi nama perusahaan, alamat, nomor telepon, serta email. Adapun rincian identitasnya adalah sebagai berikut.



Nama Instansi	: Puede Education
Alamat	: Jalan Petemon IV / 1-B, Surabaya
Nomor telepon	: +62 838 5459 5888
Email	: cs@puedeeducation.com

2.3. Visi Perusahaan

Visi perusahaan Puede Education adalah sebagai Menjadi institusi pendidikan komputer yang terbaik di Indonesia.

2.4. Misi Perusahaan

Misi perusahaan Puede Education adalah membentuk pribadi yang peka dan peduli terhadap sesama, menghasilkan mitra belajar yang profesional sesuai dengan kompetensi, selalu mengikuti perkembangan, membuat dan mengembangkan teknologi.

2.5. Struktur Organisasi

Struktur organisasi Puede Education mengadopsi struktur hierarkis sederhana untuk memastikan efektivitas dalam pelaksanaan operasional dan pelayanan kepada peserta didik. Struktur ini terdiri dari empat peran utama: Owner, Manager, Coach/Lecturer dan Account Officer.

Deskripsi tugas dan fungsi tiap posisi:

a. Owner

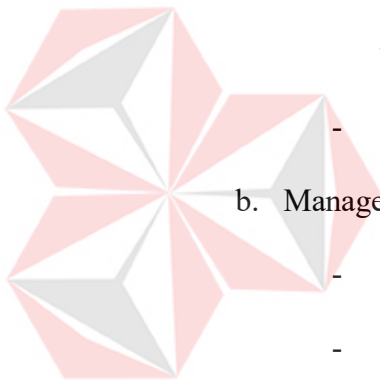
- Bertanggung jawab atas keseluruhan arah dan kebijakan strategis perusahaan.
- Mengambil keputusan utama terkait investasi, pengembangan usaha dan kemitraan strategis.
- Menunjuk dan mengevaluasi kinerja manager.

b. Manager

- Mengelola operasional harian Puede Education.
- Mengkoordinasikan tugas antara divisi coach/lecturer dan account officer.
- Menyusun program kerja dan strategi pemasaran bersama tim.
- Bertanggung jawab atas pencapaian target pendidikan dan bisnis.

c. Coach/Lecturer

- Memberikan pengajaran kepada peserta kursus sesuai bidang keahlian.
- Menyusun silabus dan materi pembelajaran.
- Mengevaluasi hasil belajar siswa secara berkala.



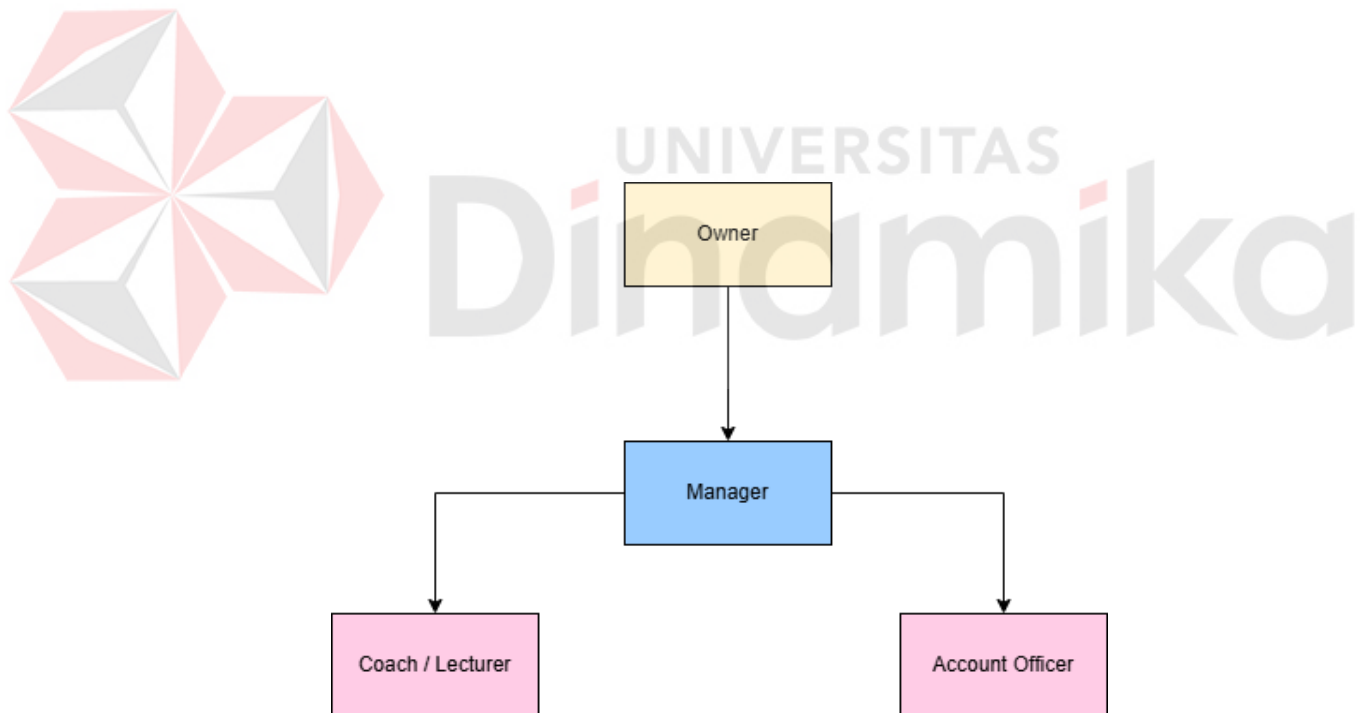
UNIVERSITAS
Dinamika

- Berinteraksi secara aktif untuk mendukung perkembangan kompetensi peserta.

d. Account Officer

- Bertanggung jawab dalam hal administrasi keuangan dan pendaftaran peserta.
- Melayani kebutuhan informasi bagi calon peserta kursus.
- Mengelola pembayaran, laporan keuangan dan data siswa.
- Menjalin hubungan dengan peserta untuk meningkatkan kepuasan layanan.

Berikut terlampir gambar struktur organisasinya:



Gambar 2. 3 Struktur Organisasi Puede Education

BAB III

LANDASAN TEORI

3.1. Pendaftaran Kursus Puede Education

Menurut Darmawan & Musril (2021), pendaftaran adalah proses pencatatan identitas pendaftar ke dalam media penyimpanan yang digunakan dalam proses administratif tertentu. Dalam konteks pendidikan, pendaftaran peserta kursus adalah tahapan awal yang sangat penting untuk memastikan bahwa data peserta terekam dengan baik dan dapat diakses saat dibutuhkan. Pendaftaran juga menjadi fondasi utama untuk proses administrasi selanjutnya, seperti pengelolaan kelas, pembayaran, dan penyusunan laporan kehadiran maupun perkembangan peserta.

Di Puede Education, proses pendaftaran saat ini dilakukan secara manual, di mana peserta datang langsung ke lokasi dan mengisi formulir pendaftaran secara tertulis. Setelah itu, mereka melakukan pembayaran secara langsung, kemudian staf administrasi akan mencatat data tersebut dalam file Excel. Meskipun sistem ini sederhana, kelemahan utamanya adalah potensi kesalahan input data, duplikasi, dan risiko kehilangan data akibat human error atau kerusakan file. Hal ini menyebabkan proses menjadi lambat dan tidak efisien, terutama saat volume peserta meningkat.

Dalam kerja praktik ini, pendaftaran peserta akan dialihkan ke sistem berbasis *web*. Hal ini memungkinkan proses pencatatan data menjadi lebih terstruktur dan otomatis, mengurangi ketergantungan pada pengolahan manual. Sistem akan mendukung admin untuk memasukkan data peserta secara digital dan tersimpan langsung dalam *database MySQL*, sehingga informasi dapat dengan

mudah diakses, dicari, dan diolah kembali kapan saja. Dengan sistem ini, alur pendaftaran akan menjadi lebih cepat, akurat, dan aman.

3.2. *Website*

Website didefinisikan sebagai kumpulan laman yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan protokol HTTP maupun HTTPS (Santhi, 2023). *Website* biasanya digunakan untuk menyampaikan informasi atau menyediakan layanan digital secara online. Dalam perkembangannya, *website* tidak hanya bersifat informatif, tetapi juga bersifat interaktif dan transaksional, memungkinkan pengguna untuk melakukan berbagai aktivitas seperti pendaftaran, pemesanan, hingga transaksi keuangan secara real-time.

Dalam konteks Kerja Praktik ini, *website* berperan sebagai antarmuka utama dari sistem pencatatan peserta kursus. *Website* memungkinkan admin atau operator untuk mengakses sistem di mana pun dan kapan pun selama terhubung dengan internet, tanpa perlu menggunakan dokumen fisik atau aplikasi khusus di komputer lokal. Dengan menggunakan teknologi *web*, sistem dapat dirancang lebih fleksibel dan user-friendly, serta mempermudah integrasi dengan fitur lain seperti pengelolaan data kelas, pelaporan, dan pencetakan dokumen.

Penggunaan *website* dalam kerja praktik ini memberikan banyak keuntungan, seperti kemudahan akses, efisiensi waktu, dan peningkatan keakuratan data. Selain itu, pengembangan berbasis *web* memungkinkan skalabilitas sistem, di mana fitur-fitur tambahan dapat ditambahkan di masa depan, seperti penjadwalan kelas atau integrasi sistem pembayaran online. *Website* juga dapat membantu

meningkatkan citra profesional institusi karena peserta akan merasa lebih nyaman dengan sistem yang modern dan mudah digunakan.

3.3. Database

Database merupakan kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diakses serta dikelola melalui perangkat lunak khusus yang disebut sistem manajemen basis data (DBMS). Menurut Aswiputri (2022) *database* memungkinkan penyimpanan dan pengelolaan informasi dalam jumlah besar secara efisien dan terstruktur. Hal ini penting dalam dunia bisnis dan pendidikan, karena menjamin integritas, konsistensi, dan keamanan data.

MySQL adalah salah satu sistem manajemen basis data relasional (RDBMS) yang paling banyak digunakan dalam pengembangan aplikasi *web*. Menurut Nusantara & Prasetyo (2025), *MySQL* bersifat open-source, ringan, dan mendukung pengelolaan data skala kecil hingga menengah dengan performa yang baik. *MySQL* juga mendukung penggunaan bahasa *SQL* (*Structured Query Language*) yang umum digunakan untuk manipulasi data, sehingga memudahkan integrasi dengan berbagai aplikasi *web* berbasis *PHP*, *Python*, atau *Node.js*.

Dalam kerja praktik ini, *MySQL* digunakan untuk menyimpan seluruh data peserta, data kelas, serta informasi lainnya yang berkaitan dengan kegiatan kursus di Puede Education. Penggunaan *database* memungkinkan admin untuk melakukan operasi seperti pencarian, pembaruan, dan penghapusan data dengan cepat dan akurat. Selain itu, *database* juga memfasilitasi proses backup dan keamanan data, memastikan bahwa informasi penting tidak mudah hilang atau rusak seperti pada sistem pencatatan manual menggunakan Excel.

3.4. *PHP*

PHP adalah bahasa pemrograman berbasis server-side yang digunakan untuk membuat aplikasi *web* dinamis. Menurut Noviana (2021) *PHP* banyak dipilih karena kemudahan sintaksnya, fleksibilitasnya, dan dokumentasi yang melimpah. Dengan *PHP*, pengembang dapat membuat halaman *web* yang mampu menampilkan konten yang berubah-ubah tergantung pada input pengguna, serta memungkinkan pemrosesan data secara real-time di server.

Dalam Kerja Praktik ini, *PHP* digunakan sebagai bahasa utama untuk membangun backend dari sistem pencatatan peserta. *PHP* akan mengatur logika program yang menghubungkan antara *website (frontend)* dan *database MySQL* (backend). Misalnya, ketika admin mengisi formulir pendaftaran peserta baru, *PHP* akan memproses data tersebut dan menyimpannya langsung ke *database*. *PHP* juga digunakan untuk fitur pencarian data, pembaruan informasi, serta pembuatan laporan dalam format digital.

Selain itu, *PHP* mendukung banyak pustaka dan framework tambahan yang mempercepat proses pengembangan sistem. *PHP* juga kompatibel dengan berbagai sistem operasi dan server, sehingga sistem dapat di-hosting dengan mudah di berbagai platform. Oleh karena itu, *PHP* sangat cocok untuk Kerja Praktik ini yang membutuhkan efisiensi waktu pengembangan dan kemudahan pemeliharaan jangka panjang.

3.5. Metode *Waterfall*

Menurut Wahid (2020) metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*” dimana hal ini menggambarkan

pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*contruction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan

3.6. *Blackbox Testing*

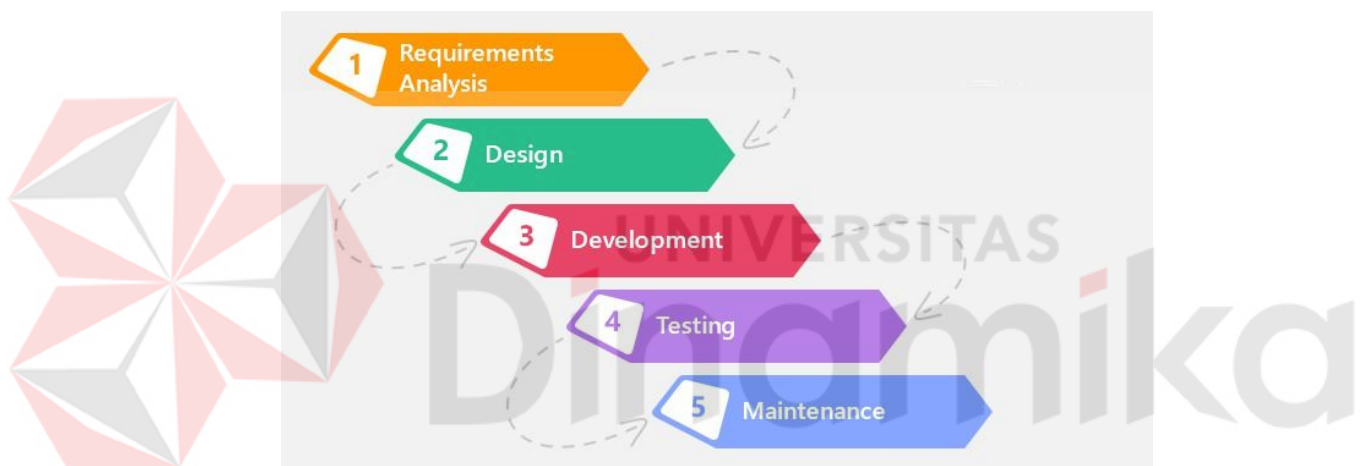
Menurut Uminingsih (2022) metode *Black Box Testing* merupakan pengujian untuk menunjukkan kesalahan pada system aplikasi seperti kesalahann pada fungsi system aplikasi, serta menu aplikasi yang hilang. Jadi *Black Box testing* merupakan metode uji fungsionalitas system aplikasi. Dalam melakukan pengujian menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti . Dikatakan pasti artinya bila salah ,maka di tolak oleh system informasi atau data input tersebut tidak dapat disimpan dalam data base, sedangkan bila data input benar maka dapat di terima / masuk di *database* system informasi.

BAB IV

DESKRIPSI PEKERJAAN

4.1. Metode *Waterfall*

Model penelitian untuk pengembangan aplikasi *website* ini menggunakan model pengembangan *Waterfall*, yang terdiri dari tahapan berurutan, seperti analisis kebutuhan, desain sistem, implementasi, pengujian dan pemeliharaan. Proses ini dapat dilihat pada Gambar 4.1 di bawah ini.



Gambar 4. 1 Metode Waterfall

4.2. Analisis Kebutuhan

Berdasarkan hasil diskusi dengan pihak Puede Education, kebutuhan sistem pencatatan peserta kursus dapat dirangkum dalam dua aspek utama: (1) Fitur manajemen data berbasis CRUD (Create, Read, Update, Delete) yang digunakan untuk mengelola data peserta kursus dan data kelas secara terstruktur dan efisien; (2) Fitur pelaporan, yang memungkinkan admin dan pemilik untuk menghasilkan

laporan data peserta dan kelas dalam format *PDF*, guna mendukung proses analisis dan pengambilan keputusan bisnis.

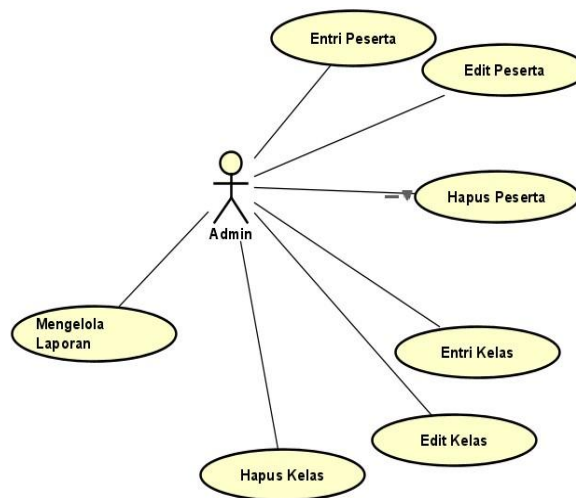
4.3. Desain

Dalam proses Desain ini, digunakan beberapa model pemodelan seperti Use Case Diagram, *Activity* Diagram, Sequence Diagram, dan Class Diagram yang disesuaikan dengan hasil analisis kebutuhan serta desain sistem. Adapun tahapan dalam merancang dan membangun sistem informasi berbasis objek meliputi:

1. Menyusun Use Case Diagram untuk menggambarkan interaksi antara aktor dan sistem.
2. Menyusun *Activity* Diagram yang merepresentasikan alur logika, proses bisnis, dan langkah-langkah dari setiap use case yang telah ditentukan.
3. Menyusun Sequence Diagram untuk menggambarkan urutan interaksi antar objek dalam sistem berdasarkan waktu, khususnya dalam menjalankan suatu skenario fungsional.
4. Menyusun Class Diagram untuk menjelaskan relasi antar objek yang digunakan dalam sistem ini.

4.4. Use Case Sistem

Gambar diagram dibawah ini memvisualisasikan beragam skenario use case yang terjadi dalam platform *web* yang sedang dibangun. Use Case dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Use Case Sistem

Pada gambar use case diagram di atas, terdapat satu aktor utama, yaitu Admin. Aktor ini merupakan pengguna internal yang memiliki hak akses penuh terhadap seluruh fitur dalam sistem pencatatan peserta kursus. Berikut adalah penjabaran hak akses dan fungsionalitasnya:

Admin adalah pengguna yang telah melakukan proses autentikasi untuk dapat mengakses sistem. Admin memiliki tanggung jawab penuh terhadap pengelolaan data peserta dan data kelas, serta pembuatan laporan. Adapun fungsionalitas yang dapat dilakukan oleh Admin meliputi:

- Entri Peserta: Admin dapat menambahkan data peserta kursus baru ke dalam sistem.
- Edit Peserta: Admin dapat mengubah atau memperbarui data peserta kursus yang sudah terdaftar.

- Hapus Peserta: Admin dapat menghapus data peserta dari sistem jika diperlukan (misalnya peserta batal mengikuti kursus).
- Entri Kelas: Admin dapat membuat atau menambahkan data kelas baru ke dalam sistem.
- Edit Kelas: Admin dapat mengubah detail informasi kelas seperti nama kelas, jadwal, atau kapasitas.
- Hapus Kelas: Admin dapat menghapus data kelas dari sistem.
- Mengelola Laporan: Admin dapat menghasilkan laporan data peserta dan data kelas, misalnya dalam bentuk *PDF*, untuk keperluan dokumentasi atau evaluasi.

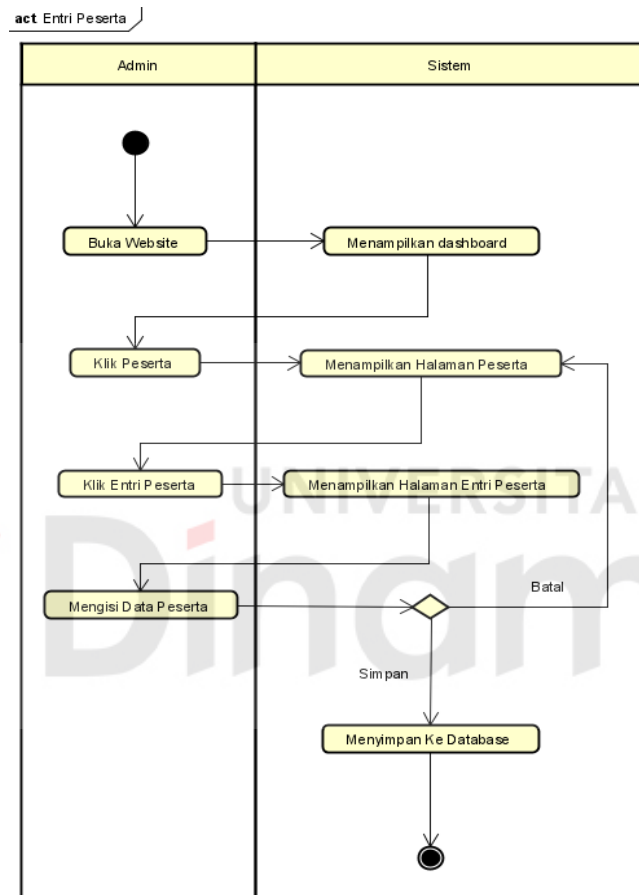
4.5. Activity Diagram

Bab ini menguraikan secara komprehensif alur aktivitas dan proses bisnis yang terjadi dalam sistem, mulai dari awal hingga akhir, dengan menggunakan *Activity* diagram untuk menggambarkan logika alur kerja secara terstruktur.

A. Entri Peserta

Activity Entri peserta dapat dilihat pada Gambar 4.3. *Activity* diagram di atas menggambarkan alur proses yang dilakukan admin dalam menambahkan data peserta ke dalam sistem. Proses dimulai ketika admin membuka *website*, dan sistem merespons dengan menampilkan *Dashboard*. Selanjutnya, admin mengklik menu "Peserta", lalu sistem menampilkan halaman peserta. Dari halaman tersebut, admin memilih untuk menambahkan peserta baru dengan mengklik "Entri Peserta", yang kemudian akan menampilkan halaman entri data peserta. Di halaman ini, admin mengisi data peserta yang diperlukan. Setelah data diisi, terdapat dua opsi: membatalkan proses atau menyimpan data. Jika admin memilih

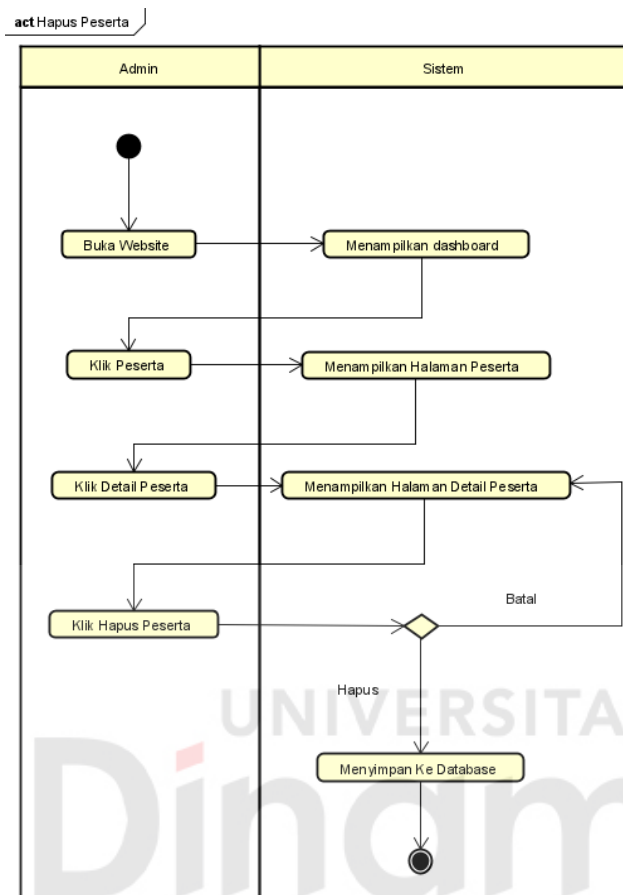
untuk menyimpan, sistem akan menyimpan data peserta yang telah diisi ke dalam *database*. Proses ini diakhiri setelah penyimpanan selesai. Diagram ini menjelaskan tahapan yang runtut dan jelas mengenai proses input data peserta oleh admin dalam sistem.



Gambar 4. 3 Activity Entri Peserta

B. Edit Peserta

Activity diagram ini menjelaskan proses yang dilakukan oleh admin untuk mengubah data peserta dalam sistem, *Activity* Edit Peserta dapat dilihat pada Gambar 4.4. Proses dimulai saat admin membuka *website*, dan sistem menampilkan halaman *Dashboard*. Setelah itu, admin mengklik menu "Peserta", lalu sistem menampilkan halaman peserta. Dari sana, admin memilih salah satu peserta dengan



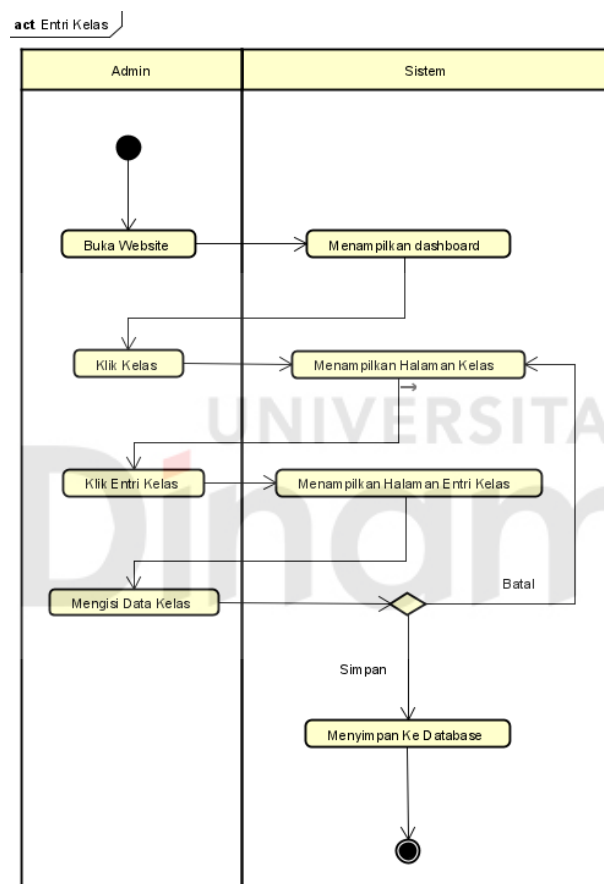
Gambar 4. 5Activity Hapus Peserta

Activity diagram di atas menggambarkan alur proses yang dilakukan admin untuk menghapus data peserta dari sistem. Proses dimulai ketika admin membuka *website* dan sistem merespons dengan menampilkan *Dashboard*. Setelah itu, admin mengklik menu "Peserta" dan sistem menampilkan halaman peserta. Admin kemudian memilih peserta tertentu dengan mengklik "Detail Peserta", sehingga sistem menampilkan informasi detail dari peserta yang dipilih. Dari halaman ini, admin memilih opsi "Hapus Peserta". Setelah itu, sistem memberikan pilihan apakah tindakan akan dibatalkan atau dilanjutkan. Jika admin memutuskan untuk

melanjutkan proses penghapusan, maka sistem akan menyimpan perubahan tersebut dengan menghapus data peserta dari *database*. Proses diakhiri setelah data berhasil dihapus. Diagram ini menunjukkan alur kerja yang sistematis dan jelas untuk proses penghapusan data peserta oleh admin.

D. Entri Kelas

Activity Diagram Entri Kelas dapat dilihat pada gambar 4.6.



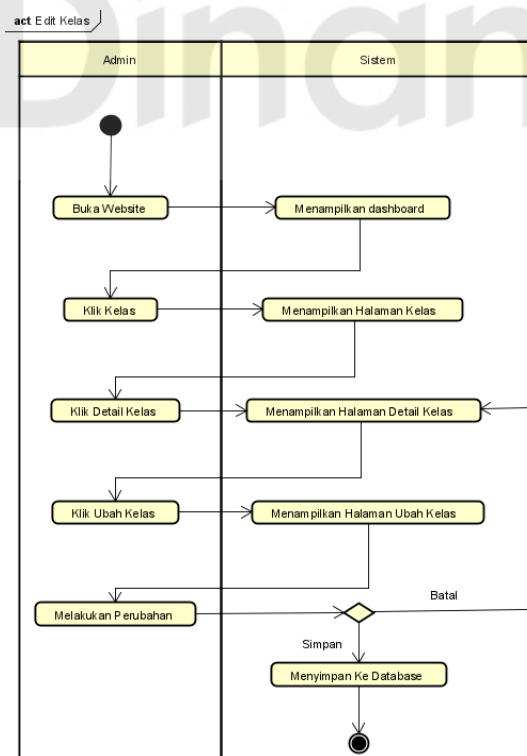
Gambar 4. 6 Activity Entri Kelas

Activity diagram di atas menggambarkan alur aktivitas yang dilakukan oleh seorang admin dalam mengelola data kelas pada sebuah sistem. Proses dimulai saat admin membuka *website*, dan sistem akan menampilkan *Dashboard*. Setelah itu, admin mengklik menu "Kelas", yang kemudian direspons oleh sistem dengan

menampilkan halaman kelas. Selanjutnya, admin memilih untuk melihat detail kelas tertentu, yang membuat sistem menampilkan halaman detail kelas. Jika admin ingin melakukan perubahan, maka ia akan mengklik tombol "Ubah Kelas", dan sistem menampilkan halaman ubah kelas. Admin kemudian melakukan perubahan pada data kelas. Setelah perubahan dilakukan, terdapat dua kemungkinan: admin membatalkan perubahan atau menyimpan perubahan. Jika admin memilih untuk menyimpan, maka sistem akan menyimpan data yang telah diubah ke dalam *database*. Aktivitas ini kemudian diakhiri, menandakan bahwa proses pengelolaan kelas oleh admin telah selesai. Diagram ini mencerminkan interaksi yang jelas antara pengguna (admin) dan sistem dalam konteks pengelolaan data kelas.

E. Edit Kelas

Activity Diagram Edit Kelas dapat dilihat pada Gambar 4.7.

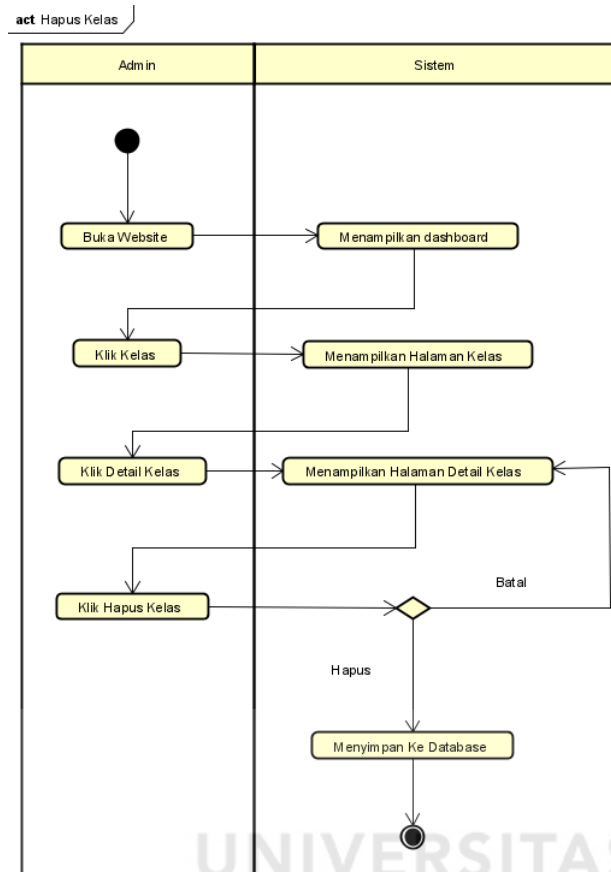


Gambar 4. 7 Activity Edit Kelas

Activity diagram di atas menggambarkan proses interaksi admin dengan sistem dalam mengelola informasi kelas. Proses dimulai ketika admin membuka *website*, yang direspons oleh sistem dengan menampilkan *Dashboard*. Selanjutnya, admin mengklik menu “Kelas”, dan sistem menampilkan halaman kelas. Setelah itu, admin memilih untuk melihat detail suatu kelas, lalu sistem menampilkan halaman detail kelas tersebut. Dari sana, admin dapat memilih untuk mengubah data kelas dengan mengklik tombol “Ubah Kelas”, yang menyebabkan sistem menampilkan halaman ubah kelas. Admin kemudian melakukan perubahan yang diinginkan. Setelah perubahan dilakukan, terdapat dua pilihan: membatalkan perubahan atau menyimpannya. Jika admin memilih untuk menyimpan, maka sistem akan menyimpan data perubahan ke dalam *database*. Proses berakhir setelah penyimpanan selesai. Diagram ini menunjukkan alur logis dan sistematis dari proses update data kelas oleh admin dalam sistem.

F. Hapus Kelas

Activity Diagram Hapus Kelas dapat dilihat pada Gambar 4.8.



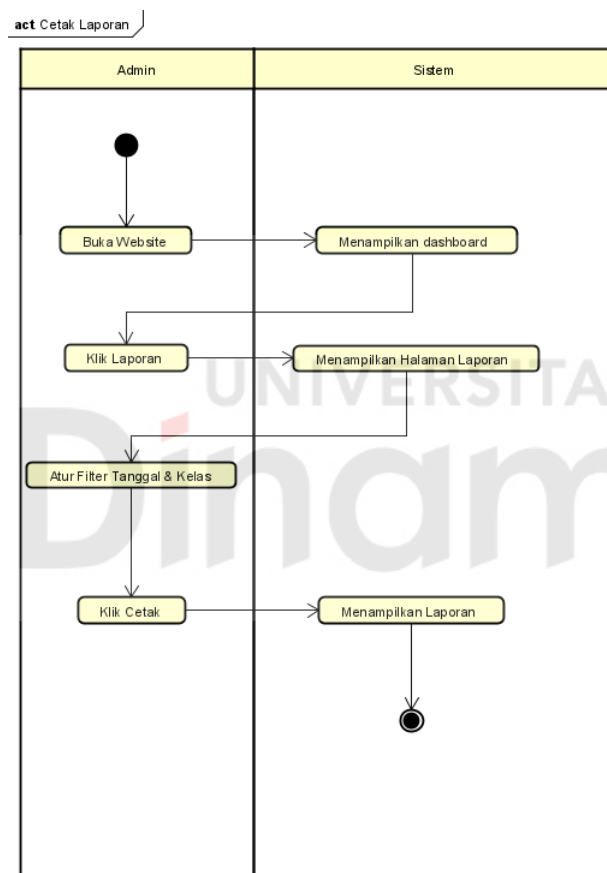
Gambar 4. 8 Activity Hapus Kelas

Activity diagram di atas menjelaskan alur aktivitas admin dalam proses menghapus data kelas pada sebuah sistem. Proses dimulai saat admin membuka *website*, yang kemudian direspons oleh sistem dengan menampilkan halaman *Dashboard*. Selanjutnya, admin mengklik menu "Kelas", dan sistem menampilkan halaman kelas. Admin kemudian memilih salah satu kelas dengan mengklik "Detail Kelas", sehingga sistem menampilkan informasi detail dari kelas yang dipilih. Dari halaman detail tersebut, admin dapat memilih untuk menghapus kelas dengan mengklik tombol "Hapus Kelas". Setelah itu, sistem akan memberikan pilihan apakah tindakan tersebut akan dibatalkan atau dilanjutkan. Jika admin memilih untuk melanjutkan proses penghapusan, maka sistem akan menyimpan perubahan

dengan menghapus data kelas dari *database*. Proses ini diakhiri setelah data berhasil dihapus. Diagram ini menunjukkan interaksi yang sederhana namun penting antara admin dan sistem dalam manajemen data kelas, khususnya untuk proses penghapusan.

G. Cetak Laporan

Activity Diagram Cetak Laporan dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Activity Cetak Laporan

Activity diagram di atas menggambarkan alur proses yang dilakukan oleh admin untuk menampilkan laporan berdasarkan filter tertentu dalam sistem. Proses dimulai ketika admin membuka *website*, dan sistem merespons dengan menampilkan *Dashboard*. Selanjutnya, admin mengklik menu "Laporan", lalu

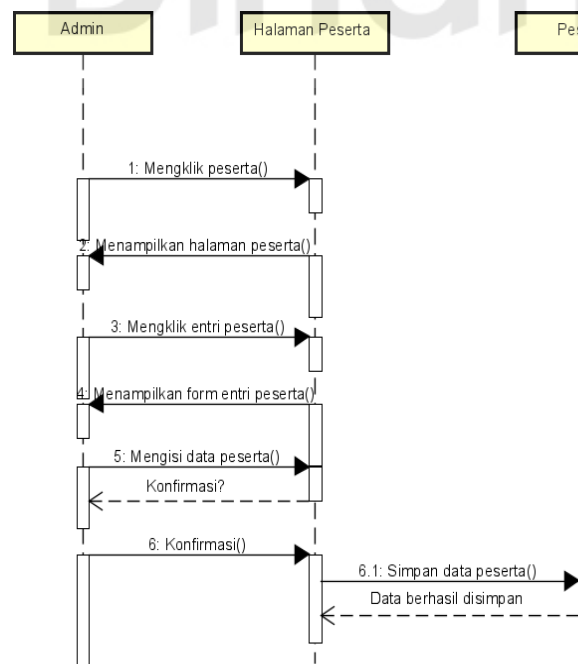
sistem menampilkan halaman laporan. Setelah itu, admin mengatur filter laporan berdasarkan tanggal dan kelas sesuai kebutuhan. Setelah filter diatur, admin mengklik tombol “Cetak”, dan sistem akan memproses permintaan tersebut dengan menampilkan laporan yang sesuai dengan filter yang telah ditentukan. Proses ini berakhir setelah laporan berhasil ditampilkan. Diagram ini menggambarkan interaksi yang sederhana dan terstruktur antara admin dan sistem dalam melakukan pencarian dan pencetakan laporan.

4.6. Sequence Diagram

Bab ini menguraikan secara komprehensif alur interaksi antara aktor (pengguna sistem) dengan komponen-komponen utama dalam platform melalui sequence diagram.

A. Sequence Diagram Entri Peserta

Sequence Entri Peserta dapat dilihat pada Gambar 4.10.



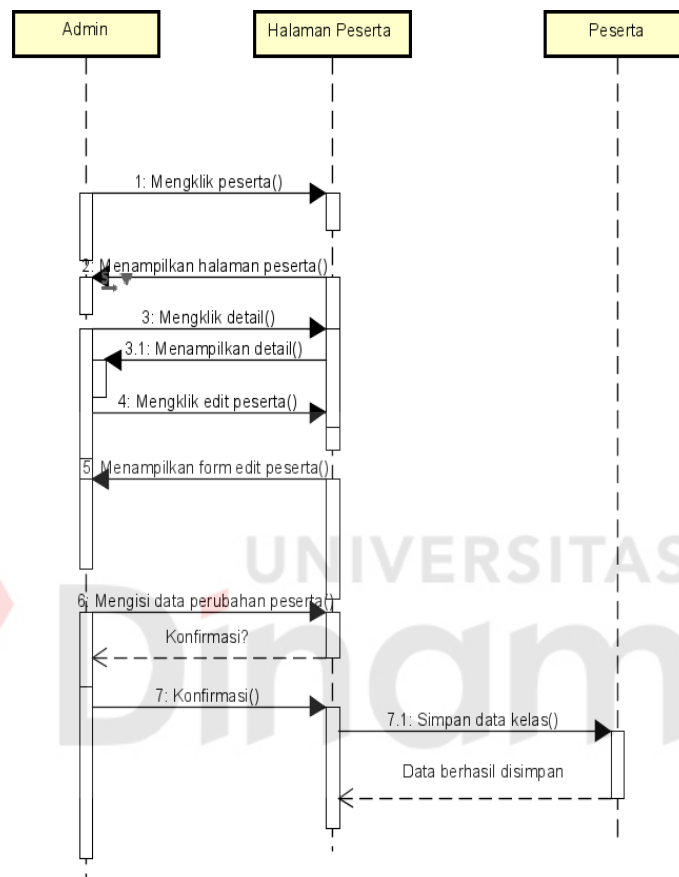
Gambar 4. 10 Sequence Entri Peserta

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam melakukan entri data peserta. Proses dimulai saat admin mengklik menu **peserta**, yang kemudian menyebabkan sistem menampilkan **halaman peserta**. Setelah itu, admin memilih opsi **entri peserta**, dan sistem menampilkan **form entri peserta**. Admin kemudian mengisi data peserta pada form yang tersedia dan mengirimkan permintaan konfirmasi. Sistem merespons dengan menanyakan apakah admin ingin melanjutkan, lalu admin memberikan **konfirmasi** untuk menyimpan data tersebut. Setelah dikonfirmasi, sistem melakukan aksi **menyimpan data peserta** ke dalam basis data. Akhirnya, sistem memberikan umpan balik kepada admin bahwa **data berhasil disimpan**. Diagram ini menunjukkan urutan proses yang sistematis dan interaktif dalam menambahkan data peserta ke sistem.

B. Sequence Diagram Edit Peserta

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam melakukan perubahan data peserta. Proses dimulai ketika admin mengklik menu peserta, yang memicu sistem untuk menampilkan halaman peserta. Setelah halaman peserta tampil, admin memilih untuk melihat detail data peserta dengan mengklik tombol detail, sehingga sistem menampilkan informasi detail peserta tersebut. Selanjutnya, admin melanjutkan dengan mengklik tombol edit peserta untuk memperbarui data, lalu sistem menampilkan form edit peserta. Admin kemudian mengisi data perubahan peserta yang diinginkan dan mengajukan permintaan konfirmasi. Sistem merespons dengan menanyakan apakah admin yakin untuk menyimpan perubahan tersebut, dan admin memberikan konfirmasi untuk melanjutkan proses. Setelah mendapatkan konfirmasi, sistem melakukan aksi

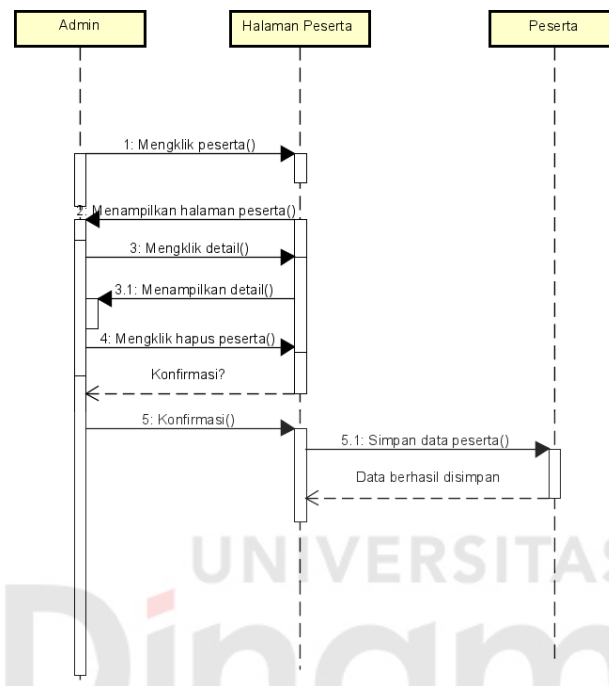
menyimpan data perubahan ke dalam basis data. Terakhir, sistem memberikan umpan balik bahwa data berhasil disimpan. Diagram ini memperlihatkan alur proses yang runtut dan interaktif antara admin dan sistem saat melakukan pembaruan data peserta. Sequence Edit Peserta dapat dilihat pada Gambar 4.11.



Gambar 4. 11 Sequence Edit Peserta

C. Sequence Diagram Hapus Peserta

Sequence Hapus Peserta dapat dilihat pada Gambar 4.12.



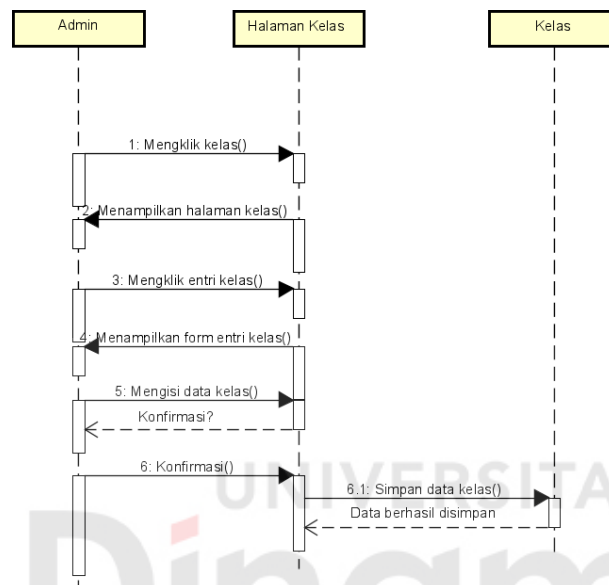
Gambar 4. 12 Sequence Hapus Peserta

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam menghapus data peserta. Proses diawali saat admin mengklik menu peserta, sehingga sistem menampilkan halaman peserta. Setelah itu, admin memilih untuk melihat detail peserta dengan mengklik tombol detail, dan sistem menampilkan detail informasi peserta. Selanjutnya, admin mengklik tombol hapus peserta untuk menghapus data yang diinginkan. Sistem kemudian menanyakan konfirmasi kepada admin apakah yakin ingin menghapus data tersebut. Admin memberikan konfirmasi, dan setelah menerima konfirmasi, sistem menjalankan proses penghapusan data peserta dari basis data. Terakhir, sistem memberikan

umpan balik kepada admin bahwa data peserta berhasil dihapus. Diagram ini menunjukkan alur proses yang terstruktur dan interaktif antara admin dan sistem untuk memastikan data peserta dapat dihapus dengan aman.

D. Sequence Diagram Entri Kelas

Sequence Entri Kelas dapat dilihat pada Gambar 4.13



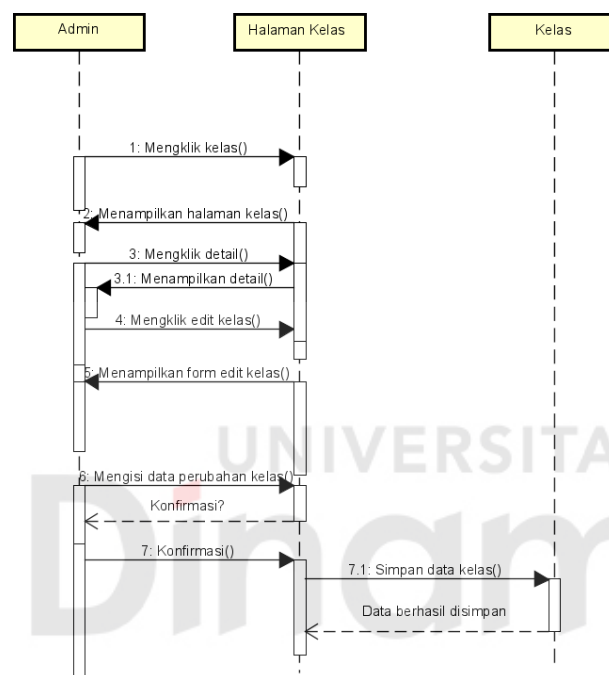
Gambar 4. 13 Sequence Entri Kelas

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam melakukan entri data kelas. Proses dimulai saat admin mengklik menu kelas, sehingga sistem menampilkan halaman kelas. Setelah itu, admin memilih opsi untuk melakukan entri kelas dengan mengklik tombol entri kelas, dan sistem merespons dengan menampilkan form entri kelas. Admin kemudian mengisi data kelas yang diperlukan pada form tersebut. Sebelum data disimpan, sistem akan meminta konfirmasi dari admin apakah ingin melanjutkan proses penyimpanan. Admin memberikan konfirmasi, dan selanjutnya sistem menyimpan data kelas ke dalam basis data. Terakhir, sistem memberikan umpan balik kepada admin bahwa

data berhasil disimpan. Diagram ini menunjukkan alur proses yang terstruktur dan interaktif antara admin dan sistem untuk memastikan data kelas dapat ditambahkan dengan benar ke dalam sistem.

E. Sequence Diagram Edit Kelas

Sequence Edit Kelas dapat dilihat pada Gambar 4.14.



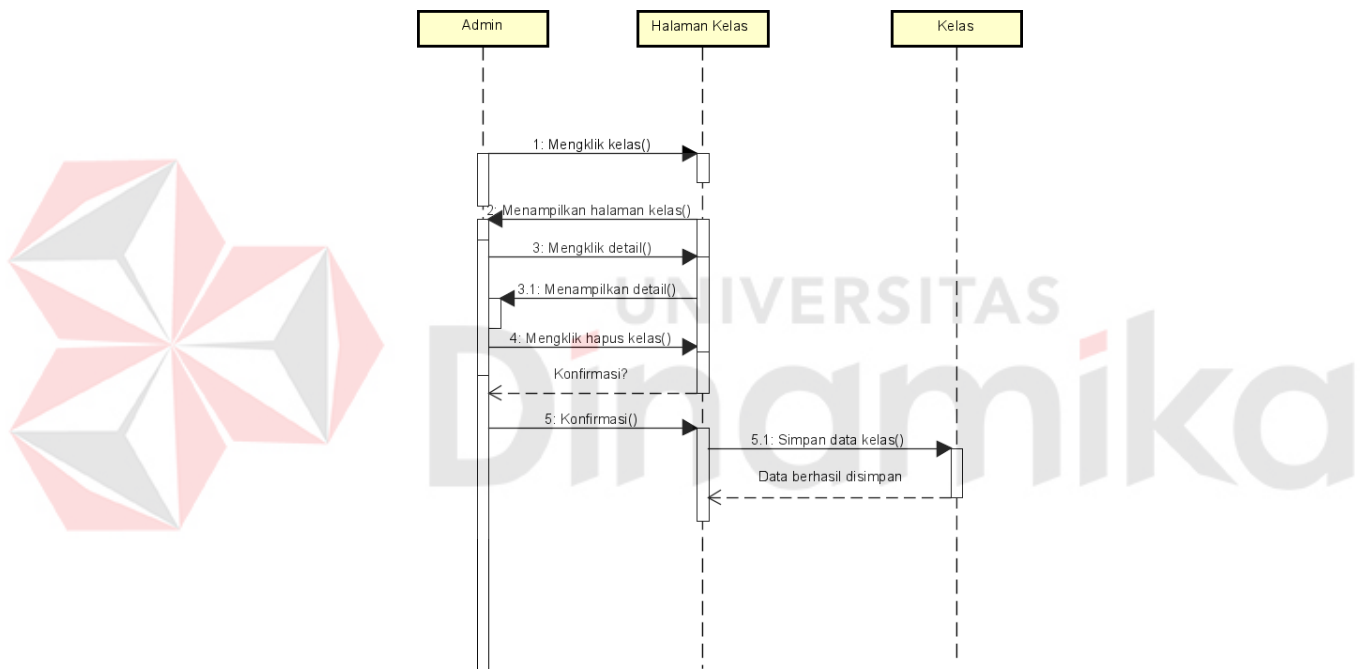
Gambar 4. 14 Sequence Edit Kelas

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam mengedit data kelas. Proses diawali ketika admin mengklik menu kelas, sehingga sistem menampilkan halaman kelas. Setelah itu, admin memilih untuk melihat detail kelas dengan mengklik tombol detail, dan sistem menampilkan informasi detail kelas tersebut. Selanjutnya, admin melanjutkan proses dengan mengklik tombol edit kelas untuk memperbarui data, kemudian sistem menampilkan form edit kelas. Admin mengisi data perubahan yang diperlukan pada form edit tersebut. Sebelum data disimpan, sistem meminta konfirmasi kepada

admin, dan admin memberikan konfirmasi untuk melanjutkan. Setelah konfirmasi diberikan, sistem menyimpan data perubahan ke dalam basis data. Terakhir, sistem memberikan umpan balik kepada admin bahwa data berhasil disimpan. Diagram ini memperlihatkan alur yang sistematis dan interaktif antara admin dan sistem untuk memastikan pembaruan data kelas dapat dilakukan dengan benar dan aman.

F. Sequence Diagram Hapus Kelas

Sequence Hapus Kelas dapat dilihat pada Gambar 4.15.



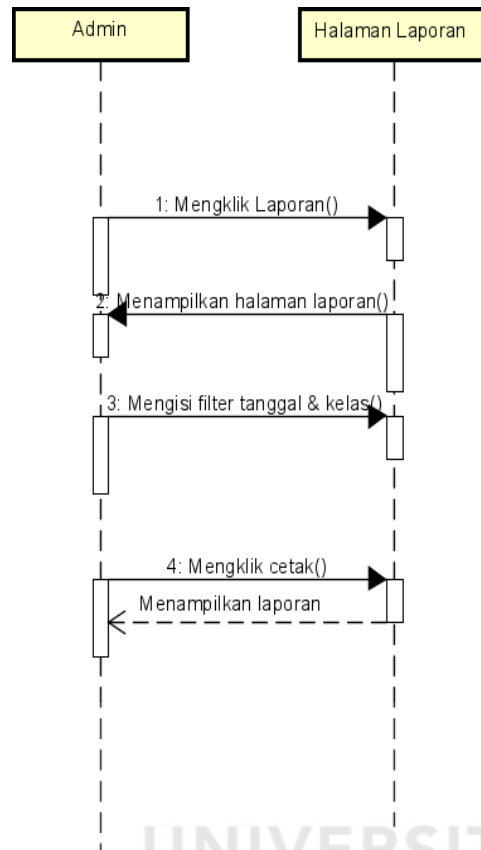
Gambar 4. 15 Sequence Hapus Kelas

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam menghapus data kelas. Proses diawali ketika admin mengklik menu kelas, sehingga sistem menampilkan halaman kelas. Setelah itu, admin memilih untuk melihat detail kelas dengan mengklik tombol detail, dan sistem

kemudian menampilkan informasi detail kelas tersebut. Selanjutnya, admin mengklik tombol hapus kelas untuk menghapus data yang diinginkan. Sistem merespons dengan meminta konfirmasi kepada admin, dan admin memberikan konfirmasi untuk melanjutkan proses penghapusan. Setelah konfirmasi diterima, sistem menjalankan aksi penghapusan data kelas dari basis data. Terakhir, sistem memberikan umpan balik kepada admin bahwa data berhasil disimpan (dalam konteks ini, berarti data perubahan akibat penghapusan berhasil diproses). Diagram ini memperlihatkan urutan langkah yang terstruktur dan interaktif untuk memastikan proses penghapusan data kelas berjalan dengan aman dan sesuai prosedur.

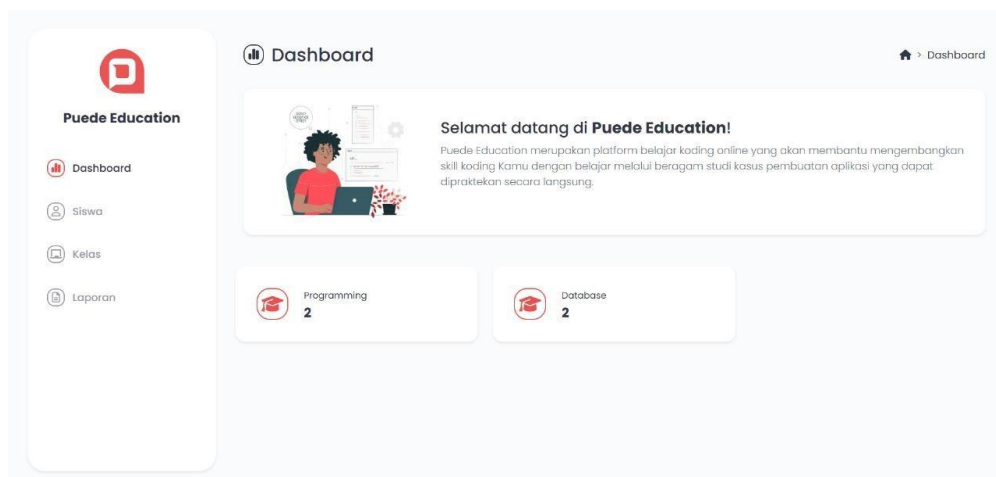
G. Sequence Digaram Cetak Laporan

Sequence diagram di atas menggambarkan proses interaksi antara admin dan sistem dalam mencetak laporan. Proses dimulai ketika admin mengklik menu laporan, sehingga sistem menampilkan halaman laporan. Setelah halaman laporan ditampilkan, admin kemudian mengisi filter berupa tanggal dan kelas untuk menentukan data laporan yang akan ditampilkan. Selanjutnya, admin mengklik tombol cetak untuk memproses laporan sesuai filter yang telah diisi. Sistem merespons dengan menampilkan laporan yang telah difilter berdasarkan input admin. Diagram ini memperlihatkan alur yang sederhana dan terstruktur untuk membantu admin menghasilkan laporan sesuai kebutuhan dengan cepat dan mudah. Sequence Cetak Laporan dapat dilihat pada Gambar 4.16.



Gambar 4. 16 Sequence Cetak Laporan

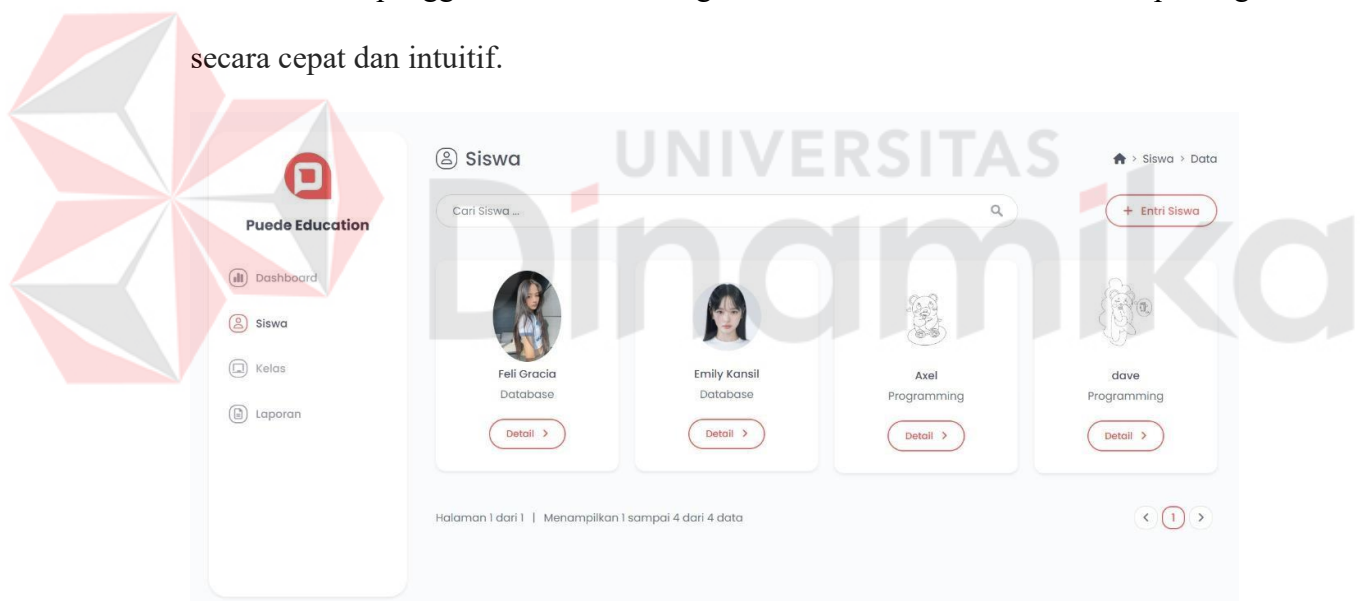
4.7. User Interfaces



Gambar 4. 17 UI Dashboard

Tampilan antarmuka pengguna (*UI*) adalah halaman *Dashboard* dari aplikasi bernama **Puede Education**, sebuah platform pembelajaran koding online,

Gambar *UI Dashboard* dapat dilihat pada Gambar 4.17. Di bagian kiri terdapat sidebar navigasi yang memuat logo aplikasi beserta menu utama, yaitu **Dashboard**, **Siswa**, **Kelas**, dan **Laporan**. Pada area utama *Dashboard*, terdapat ucapan selamat datang beserta penjelasan singkat mengenai fungsi platform, yaitu membantu mengembangkan skill koding pengguna melalui studi kasus pembuatan aplikasi yang dapat langsung dipraktikkan. Tepat di bawahnya, terdapat dua kartu informasi yang menampilkan jumlah kelas yang tersedia, yaitu **Programming** dan **Database**, masing-masing berjumlah 2 kelas. Secara visual, *UI* ini menggunakan desain modern dengan ikon yang simpel, warna yang bersih, serta tata letak yang memudahkan pengguna untuk menavigasi menu dan melihat informasi penting secara cepat dan intuitif.



Gambar 4. 18 UI Siswa

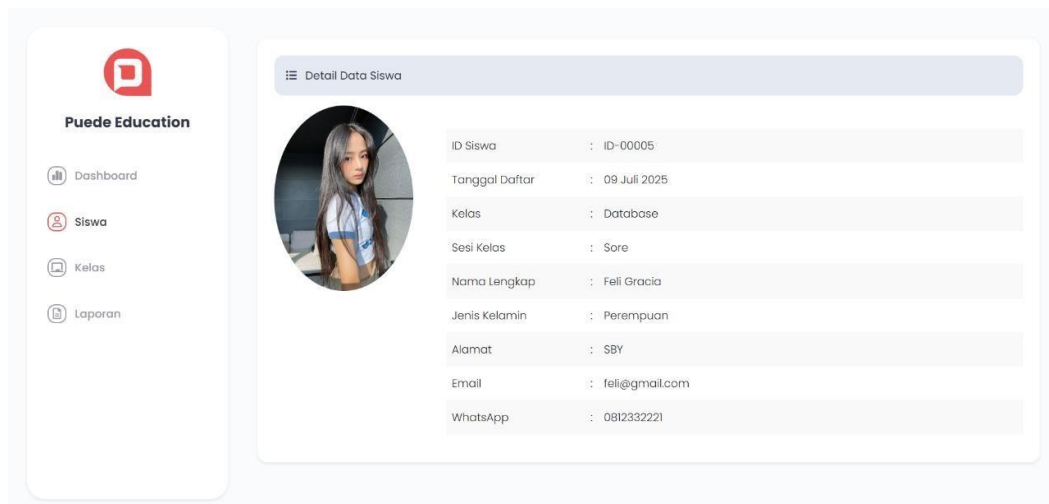
Tampilan antarmuka pengguna (*UI*) di atas adalah halaman **Siswa** dari aplikasi **Puede Education**, yang dirancang untuk menampilkan daftar data siswa yang terdaftar di platform. Pada bagian atas halaman, terdapat kolom pencarian yang memungkinkan admin mencari siswa berdasarkan nama, serta tombol **Entri Siswa** berwarna merah untuk menambah data siswa baru. Di

bawahnya, ditampilkan daftar siswa dalam bentuk kartu, masing-masing berisi nama siswa, nama kelas yang diikuti (misalnya *Database* atau *Programming*), dan tombol **Detail** untuk melihat informasi lebih lengkap. Terdapat juga informasi jumlah data yang sedang ditampilkan beserta navigasi halaman di bagian bawah. Desain ini modern, rapi, dan memudahkan admin dalam melihat, mencari, serta mengelola data siswa secara efisien. *UI* Siswa dapat dilihat pada Gambar 4.18

Gambar 4. 19 UI Entri Siswa

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman **Entri Data Siswa** pada aplikasi **Puede Education**, yang dirancang untuk memungkinkan admin menambahkan data siswa baru ke dalam sistem. Formulir entri data siswa ini terdiri atas beberapa kolom penting, seperti **ID Siswa**, **Tanggal Daftar**, **Kelas**, dan **Sesi Kelas** yang dapat dipilih melalui dropdown. Selain itu, admin juga harus mengisi **Nama Lengkap**, **Email**, **Jenis Kelamin** (dengan opsi radio button: Laki-laki atau Perempuan), **Nomor WhatsApp**, serta **Alamat**. Terdapat juga fitur untuk mengunggah **Foto Profil** siswa, dengan keterangan di bawahnya mengenai tipe file yang didukung (*.jpg atau *.png) dan batas ukuran file maksimal 1 MB. Desain *UI* ini bersih, terstruktur, dan memudahkan admin untuk memasukkan data secara

lengkap dan cepat, memastikan seluruh informasi penting tentang siswa dapat tercatat dengan baik di sistem. *UI* Entri Siswa dapat dilihat pada Gambar 4.19.

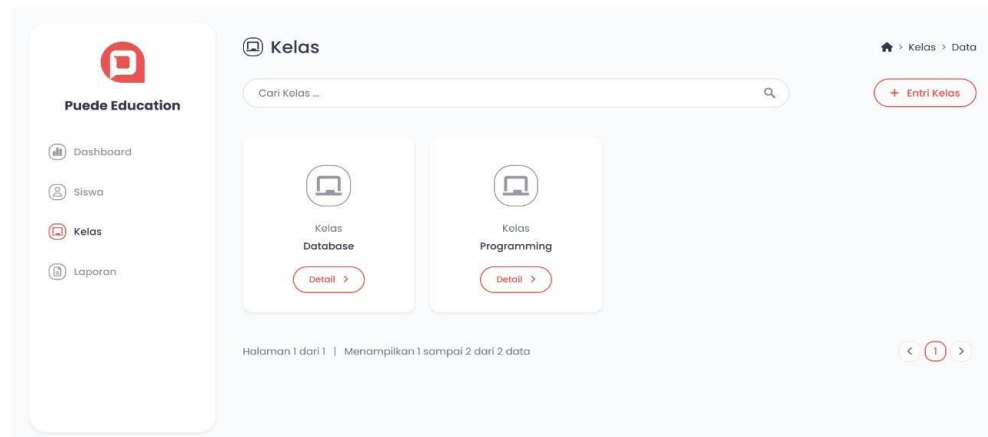


Gambar 4. 20 *UI* Detail Siswa

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman Detail Data Siswa pada aplikasi Puede Education, yang menampilkan informasi lengkap tentang seorang siswa yang sudah terdaftar dalam sistem. Halaman ini memuat beberapa menu navigasi seperti *Dashboard*, *Siswa*, *Kelas*, dan *Laporan*, yang memungkinkan admin untuk dengan mudah berpindah antara berbagai fitur aplikasi. Bagian utama halaman menampilkan detail data siswa dalam format tabel, mencakup ID Siswa , Tanggal Daftar , Kelas , Sesi Kelas, Nama Lengkap , Jenis Kelamin , Alamat , Email , dan Nomor WhatsApp . Desain *UI* ini sederhana namun informatif, dengan tata letak yang jelas dan terstruktur, memudahkan admin untuk melihat dan memverifikasi data siswa secara cepat. Informasi yang ditampilkan juga mencerminkan kelengkapan data yang diperlukan untuk manajemen siswa dalam sistem Puede Education. *UI* Detail Siswa dapat dilihat pada Gambar 4.20.

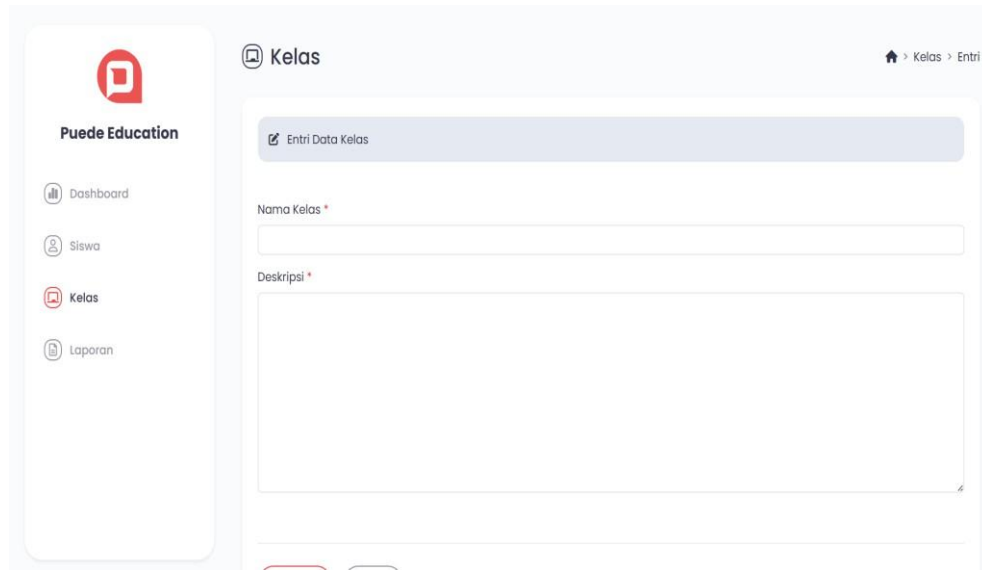
Gambar 4. 21 UI Ubah Siswa

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman Entri Data Siswa pada aplikasi Puede Education, yang dirancang untuk memungkinkan admin mengisi atau mengedit data siswa. Halaman ini terbagi menjadi beberapa bagian dengan form isian yang jelas dan terstruktur. Desain *UI* ini sederhana namun fungsional, memastikan admin dapat memasukkan data siswa dengan mudah dan lengkap. Penggunaan tanda bintang (*) pada label menandakan bahwa kolom tersebut wajib diisi, membantu menghindari kesalahan input. Meskipun ada beberapa typo (seperti "Tanggal" dan "Jenis Kelamin"), alur pengisian data tetap intuitif dan efisien. *UI* Ubah Siswa dapat dilihat pada Gambar 4.21.



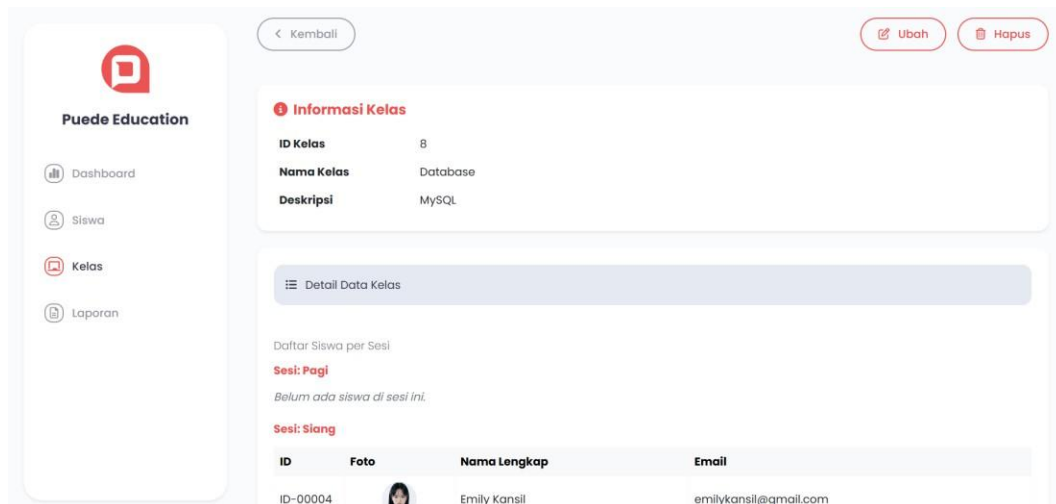
Gambar 4. 22 *UI* Kelas

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman Kelas pada aplikasi Puede Education, yang dirancang untuk mengelola data kelas yang tersedia dalam sistem. Bagian utama halaman menampilkan daftar kelas yang tersimpan dalam sistem, termasuk Kelas *Database* dan Kelas *Programming*, masing-masing dilengkapi dengan opsi *Detail* untuk melihat informasi lebih lanjut. Di bagian bawah, terdapat informasi paginasi yang menunjukkan "Halaman 1 dari 1" dengan total 2 data yang ditampilkan. Meskipun antarmukanya sederhana, fungsionalitas dasarnya tetap jelas, memungkinkan admin untuk melihat dan mengelola kelas dengan efisien. Desain ini memprioritaskan kemudahan navigasi dan kejelasan informasi. *UI* Kelas dapat dilihat pada Gambar 4.22.



Gambar 4. 23 *UI* Entri Kelas

Tampilan *UI* di atas adalah halaman Entri Data Kelas pada aplikasi Puede Education, yang dirancang untuk memungkinkan admin menambahkan data kelas baru ke dalam sistem. Bagian utama halaman berisi formulir pengisian data kelas dengan dua field wajib yaitu nama kelas dan deskripsi. Desain antarmuka ini sederhana dan minimalis, memfokuskan pada fungsionalitas utama yaitu menambahkan data kelas baru. *UI* Entri Kelas dapat dilihat pada Gambar 4.23.



Gambar 4. 24 UI Detail Kelas

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman Detail Kelas pada aplikasi Puede Education, yang dirancang untuk menampilkan informasi spesifik tentang suatu kelas beserta daftar siswa yang terdaftar di dalamnya. Halaman ini terbagi menjadi tiga bagian utama: navigasi menu, informasi kelas, dan detail data siswa. *UI* Detail Kelas dapat dilihat pada Gambar 4.24

Pada bagian navigasi, terdapat menu *Dashboard*, *Siswa*, *Kelas*, dan *Laporan*. Bagian Informasi Kelas menampilkan data lengkap kelas, memberikan gambaran umum tentang kelas tersebut. Bagian Detail Data Kelas menampilkan daftar siswa yang terdaftar dalam kelas ini. Desain antarmuka ini bersifat fungsional dan terstruktur, memungkinkan admin untuk dengan cepat mengakses informasi kelas dan siswa.

Gambar 4. 25 UI Ubah Kelas

Tampilan antarmuka pengguna (UI) di atas adalah halaman Ubah Data Kelas pada aplikasi Puede Education, yang dirancang untuk memungkinkan admin mengedit informasi kelas yang sudah ada dalam sistem. Bagian utama halaman berisi formulir pengeditan data kelas dengan dua field wajib yaitu nama kelas dan deskripsi. Desainnya sederhana dan fungsional, memungkinkan admin dengan cepat memodifikasi informasi kelas. Gambar UI Ubah Kelas dapat dilihat pada Gambar 4.25.

Gambar 4. 26 UI Laporan

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman Laporan pada aplikasi Puede Education, yang dirancang untuk memfilter dan mencetak data berdasarkan kriteria tertentu. Bagian utama halaman berisi Filter Tanggal Daftar, dengan dua field wajib yaitu tanggal awal dan tanggal akhir serta kelas. Terdapat tombol cetak untuk menghasilkan laporan berdasarkan filter yang dipilih. Desainnya minimalis dan terfokus pada tujuan utama, yaitu memfilter data sebelum dicetak. Gambar *UI* Laporan dapat dilihat pada Gambar 4.26.

LAPORAN DATA SISWA


Tanggal: 01 Juli 2025 s.d. 09 Juli 2025

Kelas: Database


Sesi: Pagi

No	Foto	ID Siswa	Tanggal Daftar	Nama Lengkap	Jenis Kelamin	Alamat	Email	WhatsApp
Tidak ada siswa di sesi ini.								

Sesi: Siang

No	Foto	ID Siswa	Tanggal Daftar	Nama Lengkap	Jenis Kelamin	Alamat	Email	WhatsApp
1		ID-00004	01-07-2025	Emily Kansil	Perempuan	SBY	emilykansil@gmail.com	082933

Sesi: Sore

No	Foto	ID Siswa	Tanggal Daftar	Nama Lengkap	Jenis Kelamin	Alamat	Email	WhatsApp
1		ID-00005	09-07-2025	Feli Gracia	Perempuan	SBY	feli@gmail.com	0812312221

Surabaya, 09 Juli 2025

Surabaya, 09 Juli 2025

Gambar 4. 27 UI Cetak Laporan

Tampilan antarmuka pengguna (*UI*) di atas adalah halaman Cetak Laporan Data Siswa pada aplikasi Puede Education, yang menampilkan daftar siswa berdasarkan periode tanggal dan kelas tertentu. Laporan ini terstruktur berdasarkan Sesi Kelas (Pagi, Siang, Sore), memudahkan admin dalam melihat distribusi siswa per sesi. Gambar *UI* Cetak Laporan dapat dilihat pada Gambar 4.27.

4.8. Pengkodean/development

Fase ini merupakan tahap pengkodean, di mana desain perangkat lunak mulai diimplementasikan melalui penulisan kode menggunakan bahasa pemrograman yang sesuai agar menghasilkan program yang efisien dan minim kesalahan. Dalam Kerja Praktik *website* pencatatan peserta kursus, kami menggunakan *PHP*, *HTML*, *CSS*, dan *Javascript*. *PHP* berperan sebagai penghubung antara server dan *database*. *HTML* digunakan untuk struktur halaman, *CSS* untuk desain responsif, dan *Javascript* untuk interaktivitas. Proses pengembangan dilakukan dengan Visual Studio Code untuk mempermudah pengelolaan file.

4.9. Pengujian

Tabel 4. 1 Pengujian Black Box Testing Admin

Halaman/Fitur	Hasil	Informasi
Tambah Peserta	OK	Dapat mengakses form tambah peserta dan menyimpan data peserta baru
Edit Peserta	OK	Dapat mengubah data peserta yang sudah terdaftar
Hapus Peserta	OK	Dapat menghapus data peserta tertentu setelah konfirmasi
Tambah Kelas	OK	Dapat menambahkan data kelas baru melalui form yang tersedia
Edit Kelas	OK	Dapat mengedit informasi kelas yang telah ditambahkan
Hapus Kelas	OK	Dapat menghapus data kelas tertentu setelah konfirmasi
Cetak Laporan	OK	Dapat melihat laporan data peserta dan kelas dalam bentuk tabel

Berdasarkan hasil pengujian *Blackbox testing* di atas, seluruh fitur utama pada *website* telah berjalan dengan baik dan lancar sesuai dengan fungsi yang diharapkan. Tidak ditemukan kendala fungsional dalam proses akses halaman, maupun pengelolaan data seperti peserta dan kelas. Sistem sudah siap digunakan oleh pengguna.



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil kerja praktik yang telah dilakukan, dapat disimpulkan bahwa aplikasi pencatatan peserta kursus berbasis *web* untuk Puede Education berhasil dibangun dengan memenuhi tujuan yang telah ditetapkan. Aplikasi ini dirancang untuk menggantikan sistem manual berbasis Excel yang sebelumnya rentan terhadap kesalahan pencatatan dan ketidakefisienan dalam pengelolaan data. Dengan menggunakan teknologi *PHP*, *MySQL*, *HTML*, *CSS*, dan *Javascript*, aplikasi ini menyediakan fitur-fitur utama seperti pengelolaan data peserta, pengelolaan data kelas, serta pembuatan laporan dalam format *PDF*.

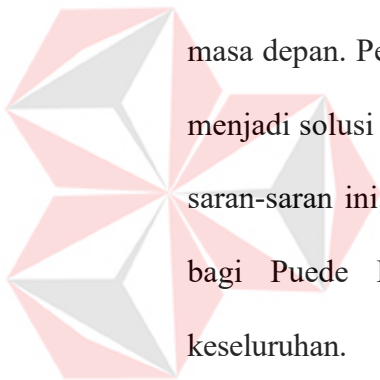
Pengujian *Blackbox testing* menunjukkan bahwa semua fitur berfungsi dengan baik, termasuk entri, edit, hapus data peserta dan kelas, serta pencetakan laporan. Aplikasi ini telah terbukti meningkatkan efisiensi operasional di Puede Education, mengurangi kesalahan input data, dan memudahkan admin dalam mengelola informasi peserta kursus. Selain itu, sistem ini juga memberikan manfaat bagi pemilik usaha dengan menyediakan data yang akurat untuk analisis dan pengambilan keputusan.

5.2. Saran

Meskipun aplikasi ini telah berhasil dibangun dan diuji, masih terdapat ruang untuk pengembangan lebih lanjut guna meningkatkan fungsionalitas dan kualitas sistem. Pertama, pengembangan fitur tambahan seperti modul pembayaran online dan sistem notifikasi via email atau WhatsApp akan memudahkan peserta

dalam melakukan transaksi serta memberikan pengingat tentang jadwal kursus atau pembayaran. Kedua, aspek keamanan sistem perlu ditingkatkan dengan menerapkan autentikasi dua faktor (2FA) untuk akun admin dan mengenkripsi data sensitif peserta guna melindungi privasi pengguna.

Selain itu, optimasi antarmuka pengguna juga menjadi hal penting untuk diperhatikan. Pengujian usability dapat dilakukan untuk memastikan antarmuka lebih intuitif dan mudah digunakan, sementara penambahan fitur pencarian dan filter yang lebih canggih akan memudahkan admin dalam mengelola data dalam jumlah besar. Terakhir, persiapan infrastruktur yang lebih robust diperlukan untuk menjamin skalabilitas sistem, terutama jika terjadi peningkatan jumlah pengguna di masa depan. Pengembangan versi mobile-friendly atau aplikasi mobile juga dapat menjadi solusi untuk memberikan akses yang lebih fleksibel. Dengan menerapkan saran-saran ini, aplikasi diharapkan dapat memberikan manfaat yang lebih besar bagi *Puede Education* serta meningkatkan pengalaman pengguna secara keseluruhan.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Aswiputri, M. (2022, 01 01). LITERATURE REVIEW DETERMINASI SISTEM INFORMASI MANAJEMEN: DATABASE, CCTV DAN BRAINWARE. *Jurnal Ekonomi Manajemen Sistem Informasi*, 3(3), 314. Diambil kembali dari <https://doi.org/10.31933/jemsi.v3i3>
- Darmawan, M. R., & Musril, H. A. (2021). Perancangan Sistem Pendaftaran Audiens Seminar Proposal di Institut Agama Islam Negeri (IAIN) Bukittinggi. *Jurnal Teknologi dan Informasi*, 11, 28. Dipetik 06 30, 2025, dari <https://ojs.unikom.ac.id/index.php/jati/article/view/3346>
- Noviana, R. (2021, 06 21). PEMBUATAN APLIKASI PENJUALAN BERBASIS WEB MONJA STORE MENGGUNAKAN PHP DAN MYSQL. *Jurnal Teknik dan Science*, 1, 114. Diambil kembali dari <https://journal.admi.or.id/index.php/JTS/article/view/128>
- Nusantara, B., & Prasetyo, S. Y. (2025, 3 1). Implementasi Framework Laravel 7.1 Pada Sistem Informasi Penjualan Convenience Store Emmi Shop. *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 10, 407. Diambil kembali dari <https://jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/view/5584>
- Santhi, T. (2023, 01 06). Pengembangan Sistem Informasi To Do List Berbasis Website Dalam Meninjau Kegiatan Mahasiswa Undiksha. *Jurnal Teknologi Ilmu Komputer*, 1, 30. Diambil kembali dari <https://ejurnal.bangunharapanbangsa.id/index.php/jtik/article/view/34>
- Uminingsih. (2022, 05 2). PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA. *Jurnal Ilmiah Teknik dan Ilmu Komputer*. Diambil kembali dari <https://doi.org/10.55123/storage.v1i2.270>
- Wahid, A. A. (2020, Oktober). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*. Diambil kembali dari https://www.researchgate.net/profile/Aceng-Wahid/publication/346397070_Analisis_Metode_Waterfall_Untuk_Pengembangan_Sistem_Informasi/links/5fbfa91092851c933f5d76b6/Analisis-Metode-Waterfall-Untuk-Pengembangan-Sistem-Informasi.pdf