

LAPORAN PROJECT AKHIR
“Sahabatbumi: Sistem Aksi Pelestarian Lingkungan”

Mata Kuliah:

Pemrograman WEB



Link Video Presentasi:

<https://youtu.be/2QiLUImKR9I>

Link Sumber Kode:

mmaaull/SahabatBumi

Disusun Oleh:

Kelas: 2024C

Kelompok 4

Aftita Choirunnisa	24091397083
Octavia Habeahan	24091397094
Faiz Maulana	24091397103

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2025

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABLE	vi
BAB I PEDAHLUAN.....	1
1.1 Latar Belakang	1
1.2 Dasar Permasalahan	2
1.3 Tujuan Aplikasi	2
1.4 Ruang Lingkup.....	2
1.5 Manfaat Aplikasi.....	3
BAB II DESKRIPSI UMUM.....	4
2.1 Deskripsi Produk.....	4
2.2 Daftar Fitur Aplikasi	4
2.3 Tipe dan Daftar Pengguna.....	5
2.4 Use Case Diagram.....	6
BAB III ANALISA DAN DESAIN.....	7
3.1 Analisa Sistem	7
3.2 Desain Sistem.....	12
3.3 Alur Aplikasi.....	15
BAB IV IMPLEMENTASI.....	28
4.1 Lingkungan Pengembangan.....	28
4.2 Struktur Folder Proyek.....	30
4.3 Struktur Code Script dan Program.....	31
4.4 Code Snippet.....	32
BAB V HASIL APLIKASI.....	42
5.1 Tampilan Aplikasi	42
BAB VI PENUTUP	49

6.1	Kesimpulan	49
6.2	Kelebihan dan Kekurangan Aplikasi	49
	DAFTAR PUSTAKA.....	51
	LAMPIRAN.....	52

DAFTAR GAMBAR

Gambar 2. 1 Use Case Diagram SahabatBumi	6
Gambar 3. 1 Diagram ERD SahabatBumi	10
Gambar 3. 2 Database Schema SahabatBumi.....	12
Gambar 3. 3 Class Diagram SahabatBumi	14
Gambar 3. 4 Activity Diagram Registrasi.....	15
Gambar 3. 5 Activity Diagram Login	15
Gambar 3. 6 Activity Diagram Lihat Profil	16
Gambar 3. 7 Activity Diagram Lihat Daftar Event.....	16
Gambar 3. 8 Activity Diagram Cari Event.....	17
Gambar 3. 9 Activity Diagram Prndaftaran Event.....	18
Gambar 3. 10 Activity Diagram Unggah Aksi User	19
Gambar 3. 11 Activity Diagram Edit Aksi User	19
Gambar 3. 12 Activity Diagram Hapus Aksi User.....	20
Gambar 3. 13 Activity Diagram Lihat Riwayat Aksi User	20
Gambar 3. 14 Activity Diagram Lihat Daftar User Admin.....	21
Gambar 3. 15 Activity Diagram Buat Event Admin	22
Gambar 3. 16 Activity Diagram Edit Event Admin	23
Gambar 3. 17 Activity Diagram Hapus Event Admin	24
Gambar 3. 18 Activity Diagram Manajemen Pendaftar Event	24
Gambar 3. 19 Activity Diagram Lihat Riwayat Aksi User	25
Gambar 3. 20 Activity Diagram Approved Aksi.....	25
Gambar 3. 21 Activity Diagram Rejected Aksi	26
Gambar 3. 22 Activity Diagram Logout	27
Gambar 4. 1 Code Database Connection	33
Gambar 4. 2 Code Role-Based Acces Control.....	33
Gambar 4. 3 Code CRUD Create.....	34
Gambar 4. 4 Code CRUD Read.....	35
Gambar 4. 5 Code CRUD Update.....	36
Gambar 4. 6 Code CRUD Delete.....	37
Gambar 4. 7 Code Error Handling.....	38
Gambar 4. 8 Code Input Validation & Sanitization	39
Gambar 4. 9 Code JavaScript AJAX Call dengan FormData	40
Gambar 5. 1 Menu Beranda	42

Gambar 5. 2 Halaman Event.....	42
Gambar 5. 3 Halaman About	43
Gambar 5. 4 Halaman Contact.....	43
Gambar 5. 5 Halaman Login.....	44
Gambar 5. 6 Halaman Registrasi	44
Gambar 5. 7 Halaman Profil	45
Gambar 5. 8 Halaman Daftar Riwayat.....	45
Gambar 5. 9 Tampilan Footer	46
Gambar 5. 10 Dashboard Admin	46
Gambar 5. 11 Manajemen User	47
Gambar 5. 12 Manajemen Event	47
Gambar 5. 13 Manajemen Pendaftar Event	48
Gambar 5. 14 Daftar Laporan User.....	48

DAFTAR TABLE

Table 3. 1 Kebutuhan Fungsional SahabatBumi.....	8
Table 3. 2 Kebutuhan Non-Fungsional SahabatBumi.....	10
Table 4. 1 Development Machine	28
Table 4. 2 User Device.....	29
Table 4. 3 Software Development Machine.....	29
Table 4. 4 Frontend Technologies	29
Table 4. 5 Backend Technologies	30
Table 4. 6 Database Tools	30

BAB I PEDAHLUAN

1.1 Latar Belakang

Permasalahan lingkungan seperti penumpukan sampah di berbagai wilayah, pencemaran udara dan air, serta rendahnya kesadaran masyarakat terhadap pelestarian alam masih menjadi isu krusial di Indonesia. Berdasarkan data Kementerian Lingkungan Hidup dan Kehutanan (KLHK) melalui SIPSN tahun 2024, timbulan sampah nasional mencapai 37,3 juta ton per tahun atau sekitar 93.700 ton per hari, dengan hanya 32,2% sampah yang terkelola secara efektif. Pemanfaatan teknologi informasi sebagai media edukasi dan penggerak aksi lingkungan belum optimal meskipun memiliki potensi besar untuk meningkatkan partisipasi masyarakat.

Oleh karena itu, website Sahabat Bumi dikembangkan sebagai platform berbasis web yang menyediakan informasi kegiatan lingkungan terkini dan mengajak masyarakat berpartisipasi aktif dalam aksi nyata. Platform ini memfasilitasi individu, komunitas, dan organisasi untuk terlibat secara kolektif, mengatasi rendahnya partisipasi aktif yang masih menjadi tantangan utama, di mana survei menunjukkan hanya sebagian kecil masyarakat yang konsisten terlibat dalam pengelolaan lingkungan. Sistem dokumentasi terintegrasi juga memungkinkan pengguna mengunggah bukti kegiatan secara transparan, mengisi kekosongan sistem pelaporan yang saat ini belum memadai.

Namun, keterbatasan sistem partisipasi yang ada masih menghambat upaya berkelanjutan. Banyak komunitas lingkungan kesulitan mencari informasi kegiatan terdekat dan mendokumentasikan kontribusi nyata mereka, sehingga kurangnya transparansi menjadi penghalang utama. Selain itu, manajemen kegiatan yang masih manual menyebabkan koordinasi relawan tidak efisien, sebagaimana sering dialami organisasi dalam mengelola pendaftaran dan pengawasan acara lingkungan.

Ketiga permasalahan tersebut, rendahnya partisipasi, kurangnya transparansi dokumentasi, dan manajemen event yang tidak terstruktur semakin mendesak karena absennya sistem apresiasi terstruktur seperti reward dan leaderboard. Tanpa pengakuan atas kontribusi mereka, motivasi masyarakat untuk berpartisipasi dalam aksi lingkungan cenderung menurun dalam jangka panjang. Hal ini memperkuat urgensi pengembangan platform terintegrasi yang mampu mengatasi semua kendala tersebut sekaligus.

Oleh karena itu, diperlukan aplikasi web dinamis seperti Sahabat Bumi yang mengintegrasikan pengelolaan event, dokumentasi berbasis foto, serta mekanisme penghargaan untuk mendorong kontribusi berkelanjutan. Platform ini tidak hanya

memfasilitasi partisipasi aktif, tetapi juga memberikan transparansi dan efisiensi manajemen yang selama ini hilang. Dengan demikian, Sahabat Bumi diharapkan menjadi solusi komprehensif bagi tantangan pelestarian lingkungan di Indonesia.

1.2 Dasar Permasalahan

1. Rendahnya Partisipasi Aktif Masyarakat: Kurangnya wadah yang terorganisir untuk memfasilitasi individu, komunitas, dan organisasi dalam melakukan aksi pelestarian lingkungan secara kolektif.
2. Kurangnya Transparansi Dokumentasi: Belum adanya sistem yang mumpuni untuk melaporkan, mendokumentasikan, dan memvalidasi bukti nyata dari aksi lingkungan yang telah dilakukan oleh relawan.
3. Manajemen Event yang Tidak Terstruktur: Organisasi atau komunitas lingkungan seringkali menghadapi kendala dalam mengelola pendaftaran peserta dan koordinasi relawan secara efisien.

1.3 Tujuan Aplikasi

Website Sahabat Bumi bertujuan untuk mengatasi secara langsung ketiga dasar permasalahan yang telah diidentifikasi, yaitu rendahnya partisipasi masyarakat, kurangnya transparansi dokumentasi, dan manajemen event yang tidak terstruktur. Melalui pendekatan sistem terintegrasi, proyek ini tidak hanya menyediakan solusi teknis tetapi juga menciptakan ekosistem digital yang berkelanjutan untuk pelestarian lingkungan. Berikut adalah tujuan utama yang ingin dicapai :

1. Membuat platform kolaborasi aktif antara individu, komunitas, dan organisasi untuk pelestarian lingkungan yang selama ini terpisah-pisah.
2. Memfasilitasi proses publikasi event oleh penyelenggara hingga pendaftaran relawan yang mudah dan cepat, mengantikan sistem manual yang ribet.
3. Menyediakan fitur validasi dan penyimpanan bukti aksi lingkungan (foto/laporan) dalam database yang akurat dan bisa diakses semua pihak.
4. Mendorong motivasi berkelanjutan melalui sistem peringkat, reward, dan leaderboard digital untuk menjaga semangat partisipasi masyarakat jangka panjang.

1.4 Ruang Lingkup

1. Aplikasi "Sahabat Bumi" hanya mencakup modul pendaftaran pengguna (visitor, volunteer, admin), autentikasi (registrasi & login), manajemen profil, pendaftaran dan

pengelolaan aksi/event, unggah laporan dan bukti kegiatan, serta dashboard sederhana untuk admin dan volunteer.

2. Fungsi pelaporan mencakup input laporan aksi, upload file (foto/dokumen), dan tampilan ringkasan laporan; tidak termasuk pemrosesan pembayaran atau integrasi pihak ketiga (mis. payment gateway).
3. Cakupan data meliputi informasi pengguna, data event/aksi, data laporan, dan file upload yang disimpan pada server lokal/DB MySQL.
4. Antarmuka yang disediakan adalah web responsif untuk desktop dan perangkat mobile dasar; tidak termasuk aplikasi native mobile.
5. Pengujian yang dicakup: fungsional dasar (registrasi, login, unggah, pendaftaran aksi), validasi input, dan pengujian akses peran (role-based access). Tidak termasuk uji beban skala besar atau audit keamanan mendalam.

1.5 Manfaat Aplikasi

1. Meningkatkan Kolaborasi Komunitas: menyediakan ruang terpusat yang menghubungkan individu, komunitas, dan organisasi sehingga inisiatif pelestarian lingkungan dapat dikordinasikan lebih mudah dan berkelanjutan.
2. Mempercepat dan Mempermudah Manajemen Event: menggantikan proses manual dengan alur pendaftaran event dan relawan yang cepat, terstruktur, dan bisa dipantau sehingga tingkat partisipasi meningkat dan administrasi menjadi efisien.
3. Menjamin Transparansi dan Dokumentasi Bukti Aksi: menyimpan foto, laporan, dan bukti lain secara terstruktur di database sehingga verifikasi, pelacakan, dan publikasi hasil aksi menjadi transparan dan dapat diakses oleh pemangku kepentingan.
4. Meningkatkan Motivasi Partisipasi Jangka Panjang: menerapkan mekanisme peringkat, reward, dan leaderboard digital untuk memberi penghargaan, meningkatkan keterlibatan, dan menjaga kontinuitas partisipasi masyarakat.
5. Mendukung Pengambilan Keputusan dan Pelaporan: menyediakan data terstruktur mengenai partisipasi, frekuensi event, dan hasil aksi yang berguna untuk evaluasi, pelaporan, dan perencanaan program lingkungan selanjutnya.
6. Memperluas Jejak Sosial dan Edukasi: memudahkan publikasi kegiatan dan hasilnya sehingga meningkatkan kesadaran publik serta memberi bahan edukasi bagi komunitas lain untuk mengadopsi praktik pelestarian serupa.

BAB II DESKRIPSI UMUM

2.1 Deskripsi Produk

Aplikasi "Sahabat Bumi" adalah platform web terintegrasi yang dirancang untuk menghubungkan individu, komunitas, dan organisasi dalam inisiatif pelestarian lingkungan dengan cara yang lebih terstruktur, transparan, dan partisipatif. Sistem ini memungkinkan penyelenggara membuat dan mempublikasikan event aksi lingkungan, sementara volunteer mendaftar secara online, mengikuti alur kerja yang jelas, dan mengunggah bukti kegiatan (foto/laporan) untuk diverifikasi oleh admin; semua metadata dan berkas disimpan di basis data MySQL dan direktori uploads sehingga dapat dilacak dan diarsipkan. Dengan antarmuka yang responsif dan fungsi dashboard, pengguna dapat melihat riwayat partisipasi, status laporan, serta peringkat atau reward untuk mendorong keterlibatan berkelanjutan; admin mendapatkan alat manajemen lengkap untuk mengelola event, memverifikasi laporan, dan mengekspor data statistik. Teknologi inti meliputi PHP untuk backend, MySQL untuk penyimpanan, serta HTML/CSS/JavaScript untuk frontend dengan fokus pada alur fungsional: pembuatan event → pendaftaran volunteer → pelaksanaan aksi → unggah bukti → verifikasi dan pelaporan.

2.2 Daftar Fitur Aplikasi

2.2.1 Fitur User/Pengguna

1. Registrasi dan Login: Pendaftaran akun sebagai volunteer atau anggota, lengkap dengan verifikasi dasar serta autentikasi berdasarkan peran pengguna.
2. Profil Pengguna: Tampilan dan pengeditan data profil, upload foto profil, serta riwayat partisipasi dalam berbagai kegiatan.
3. Pendaftaran Aksi/Event: Melihat daftar event tersedia, mendaftar sebagai volunteer, dan mendapat notifikasi konfirmasi pendaftaran.
4. Unggah Laporan & Bukti: Pengiriman laporan aksi beserta upload kegiatan, dan bisa edit, serta menghapus juga.

2.2.2 Fitur Admin

1. Manajemen Pengguna: Admin bisa monitoring siapa saja yang daftar aplikasi.
2. Manajemen Event/Aksi: Pembuatan, pengeditan, atau penghapusan event, termasuk pengaturan kuota, waktu, lokasi, dan deskripsi lengkap.
3. Validasi Laporan: Pemeriksaan laporan yang diunggah, verifikasi keabsahan, pemberian status, serta tambahan komentar.

4. Dashboard Statistik: Ringkasan partisipasi pengguna, jumlah event, laporan berdasarkan periode, dan fitur ekspor data.
5. Manajemen Pendaftaran Event : Admin bisa monitoring siapa saja pengguna/user yang daftar event.

2.3 Tipe dan Daftar Pengguna

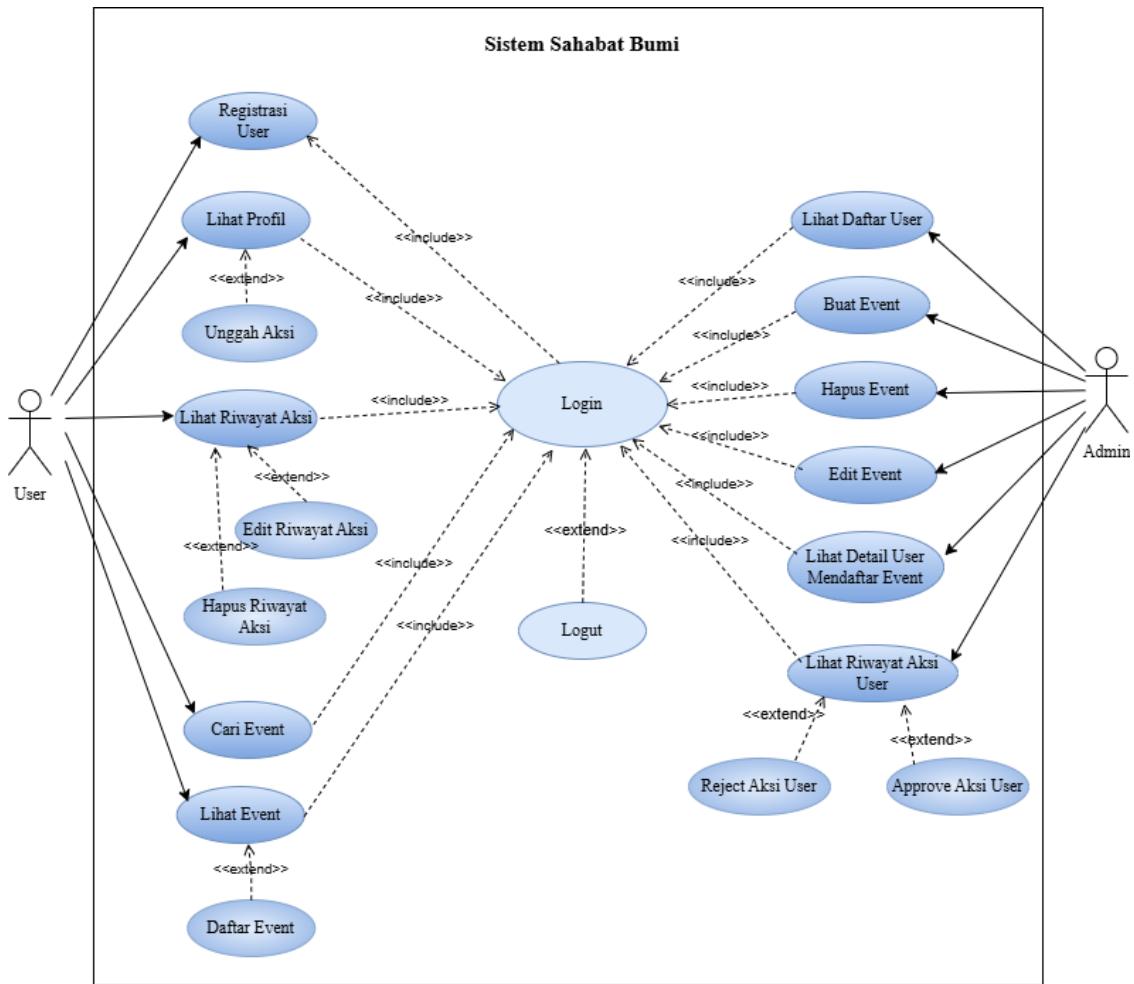
2.3.1 Admin

1. Peran utama admin adalah mengelola keseluruhan sistem termasuk konten event, data pengguna, dan laporan. Admin juga bertanggung jawab memverifikasi bukti aksi, serta mengakses statistik dan laporan keseluruhan sistem.
2. Hak akses meliputi operasi CRUD (Create, Read, Update, Delete) untuk event dan pengguna, akses penuh ke dashboard statistik, kemampuan menonaktifkan akun, serta fitur ekspor data.

2.3.2 User

1. Peran anggota atau volunteer adalah mengikuti kegiatan sosial melalui pendaftaran event, pelaksanaan aksi, pengunggahan laporan beserta bukti, dan memantau peringkat pribadi.
2. Hak akses terbatas pada registrasi/login, pengeditan profil pribadi, pendaftaran event, pengunggahan file laporan, serta akses dashboard personal.

2.4 Use Case Diagram



Gambar 2. 1 Use Case Diagram SahabatBumi

Use Case Diagram Sistem Sahabat Bumi menunjukkan interaksi antara User dan Admin dalam pengelolaan aksi dan event lingkungan. User dapat melakukan registrasi, login, mengelola profil, mengunggah dan mengelola riwayat aksi, mencari serta mendaftar event. Admin berperan dalam mengelola data sistem, meliputi pengelolaan event, melihat data dan riwayat aksi user, serta menyetujui atau menolak aksi yang diunggah. Seluruh fitur utama dalam sistem mengharuskan pengguna untuk login terlebih dahulu sebagai bentuk pengamanan akses.

BAB III ANALISA DAN DESAIN

3.1 Analisa Sistem

3.1.1 Analisa Kebutuhan Fungsional

Functional Requirements (FR) atau Kebutuhan Fungsional adalah deskripsi tentang apa yang harus dilakukan oleh sistem. Kebutuhan fungsional ini berkaitan dengan fungsi dan fitur utama yang harus tersedia dalam perangkat lunak untuk mendukung penggunaan dan tujuan sistem. FR menjelaskan interaksi antara pengguna dan sistem, serta proses yang sistem harus lakukan. Ini termasuk:

- a. Proses bisnis yang harus dilakukan oleh sistem.
- b. Operasi yang harus dilakukan oleh sistem terhadap data.
- c. Interaksi pengguna dengan sistem (misalnya, pengguna dapat mendaftar, login, atau mengelola event).
- d. Keluaran sistem (misalnya, laporan, feedback kepada pengguna, notifikasi).

No	Nama Fitur	Aktor	Deskripsi
FR-01	Autentikasi & Otorisasi	Admin, User	Sistem menyediakan registrasi akun volunteer dengan input nama, email, password, dan nomor kontak. Semua peran dapat melakukan login dengan session management serta penerapan role-based access control untuk membedakan hak akses admin dan volunteer.
FR-02	Manajemen Profil	Admin, User	User dapat melihat profil, dan admin bisa monitoring profil pengguna yang sudah daftar
FR-03	Manajemen Event/Aksi	Admin, User	Admin dapat membuat, mengedit, dan menghapus event, termasuk pengaturan kuota, tanggal, lokasi, dan deskripsi. User dapat melihat daftar event yang tersedia.
FR-04	Pencarian & Filter Event	User	Sistem menyediakan fitur pencarian dan filter event berdasarkan tanggal, lokasi, dan kategori aksi lingkungan.
FR-05	Pendaftaran Event	User	User dapat mendaftar ke event yang tersedia, dengan sistem mencatat status pendaftaran sesuai kuota dan kebijakan event.

FR-06	Konfirmasi & Status Pendaftaran	User, Admin	Sistem menampilkan status pendaftaran (terdaftar, tidak terdaftar, atau ditolak) serta memberikan notifikasi atau konfirmasi pendaftaran kepada volunteer.
FR-07	Pengunggahan Laporan & Bukti	User	User dapat mengunggah laporan aksi yang berisi judul, deskripsi, tanggal kegiatan, jumlah peserta, serta file foto atau dokumen pendukung. Sistem melakukan validasi format dan ukuran file.
FR-08	Validasi & Verifikasi Laporan	Admin	Admin dapat melihat laporan yang masuk, memverifikasi atau menolak laporan, serta memberikan komentar atau catatan verifikasi.
FR-09	Dashboard User	User	Sistem menyediakan dashboard personal yang menampilkan status pendaftaran event, perolehan poin, dan riwayat partisipasi.
FR-10	Dashboard Admin & Statistik	Admin	Admin dapat melihat statistik event, jumlah volunteer, dan jumlah laporan.
FR-11	Manajemen File	Admin	Sistem mampu mengelola file laporan dengan fitur penghapusan atau pengarsipan file bermasalah, pembatasan ukuran file, dan penanganan nama file duplikat.

Table 3. 1 Kebutuhan Fungsional SahabatBumi

3.1.2 Analisa Kebutuhan Non-Fungsional

Non-Functional Requirements (NFR) atau Kebutuhan Non-Fungsional adalah deskripsi tentang bagaimana sistem akan beroperasi. NFR tidak fokus pada apa yang sistem lakukan (fungsi spesifik), melainkan lebih pada kualitas dan karakteristik sistem yang harus dipenuhi agar sistem dapat bekerja secara efektif dan efisien.

NFR lebih berkaitan dengan aspek-aspek kinerja, keamanan, ketersediaan, usabilitas, dan faktor lain yang mempengaruhi kualitas sistem secara keseluruhan. NFR memastikan bahwa sistem dapat memenuhi standar performa dan kebutuhan pengguna dalam jangka panjang.

ID	Aspek Kebutuhan	Deskripsi
NFR-01	Keamanan (Security)	Sistem harus menerapkan penyimpanan password yang aman menggunakan hashing dan salt. Selain itu, sistem wajib melakukan validasi input untuk mencegah serangan SQL Injection dan Cross-Site Scripting (XSS), menerapkan kontrol akses berbasis peran (role-based access control), serta melakukan validasi file upload guna mencegah unggahan malware.
NFR-02	Performa (Performance)	Sistem harus mampu menampilkan halaman utama dengan waktu respon kurang dari 2 detik pada beban penggunaan normal. Untuk meningkatkan efisiensi, sistem menerapkan pagination pada daftar event dan laporan.
NFR-03	Ketersediaan & Reliabilitas (Availability & Reliability)	Sistem harus tersedia dan dapat diakses selama jam operasional. Database harus dilakukan backup secara berkala untuk mencegah kehilangan data akibat kegagalan sistem.
NFR-04	Skalabilitas (Scalability)	Sistem dirancang secara sederhana namun fleksibel sehingga memungkinkan scale-up, seperti pemisahan antara penyimpanan file dan database apabila jumlah pengguna dan data meningkat.
NFR-05	Portabilitas & Kompatibilitas (Portability & Compatibility)	Sistem harus dapat berjalan pada browser modern dan memiliki tampilan responsif untuk perangkat desktop maupun mobile. Aplikasi berbasis PHP dan MySQL

		sehingga mudah dideploy pada lingkungan LAMP atau XAMPP.
NFR-06	Usability	Sistem harus memiliki antarmuka yang intuitif dan mudah dipahami, form input yang terstruktur, serta validasi data dan pesan kesalahan yang jelas bagi pengguna.
NFR-07	Maintainability	Sistem harus dikembangkan dengan kode yang modular, memiliki dokumentasi dasar, serta menerapkan naming convention yang konsisten untuk memudahkan pemeliharaan dan pengembangan lanjutan.

Table 3. 2 Kebutuhan Non-Fungsional SahabatBumi

3.1.3 Analisa Data (ERD)



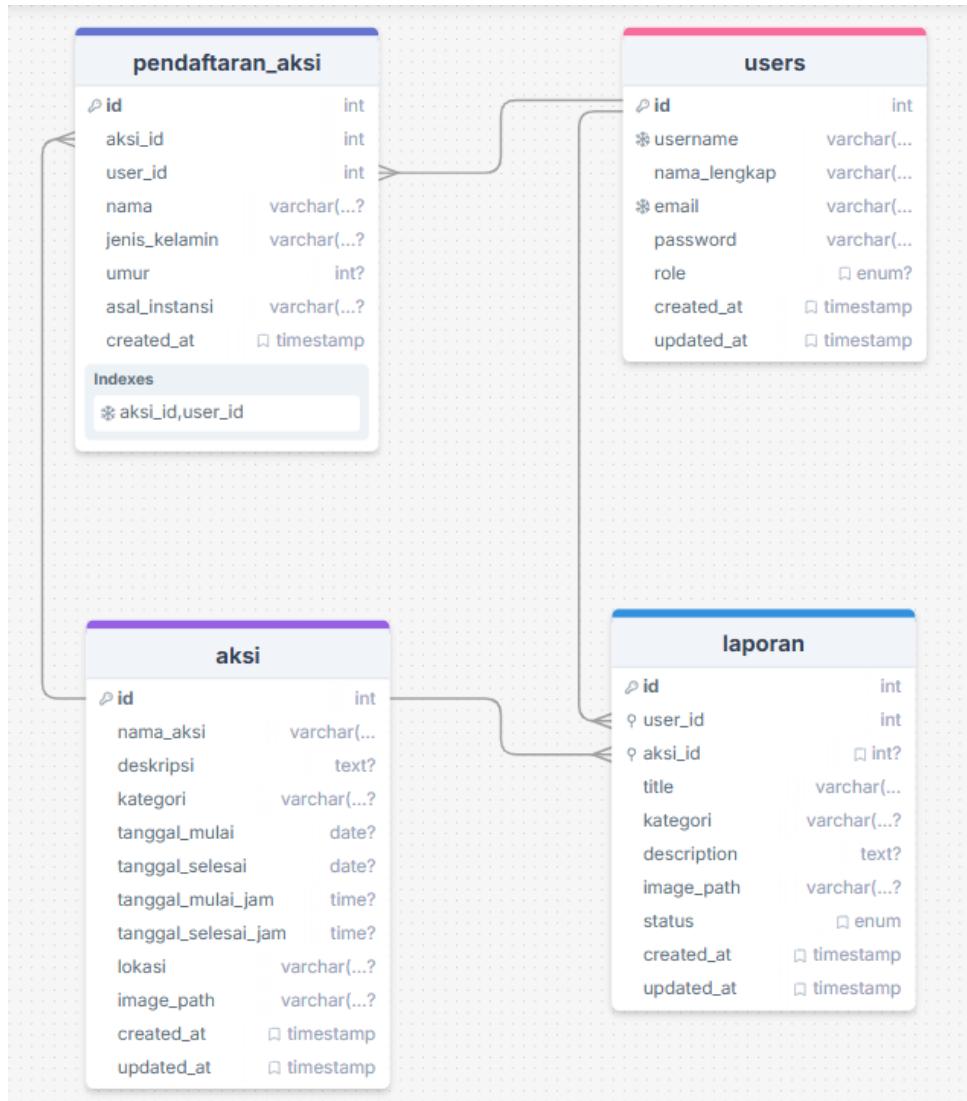
Gambar 3. 1 Diagram ERD SahabatBumi

Relasi Antar Entitas :

1. Users - Pendaftaran Aksi : One-to-Many (1:N)
 - Satu aksi dapat diikuti oleh banyak user.
 - Setiap pendaftaran hanya terkait ke satu aksi.
2. Aksi - Pendaftaran Aksi : One-to-Many (1:N)
 - Satu aksi dapat diikuti oleh banyak user.
 - Setiap pendaftaran hanya terkait ke satu aksi.
3. Users - Laporan : One-to-Many (1:N)
 - Satu user dapat membuat banyak laporan.
 - Setiap laporan hanya dimiliki oleh satu user.
4. Aksi – Laporan : One-to-Many (opsional)
 - Satu aksi dapat memiliki banyak laporan.
 - Laporan boleh tidak terkait dengan aksi tertentu.

3.2 Desain Sistem

3.2.1 Database Schema



Gambar 3. 2 Database Schema SahabatBumi

1. Table User

Tabel users menyimpan seluruh data pengguna sistem, baik admin maupun user. Fungsi utama adalah Mengelola autentikasi, otorisasi, dan identitas pengguna dalam sistem.

2. Table Aksi

Tabel aksi digunakan untuk menyimpan data event atau aksi lingkungan yang dibuat oleh admin. Fungsi utama adalah Menyimpan informasi kegiatan yang dapat diikuti oleh pengguna.

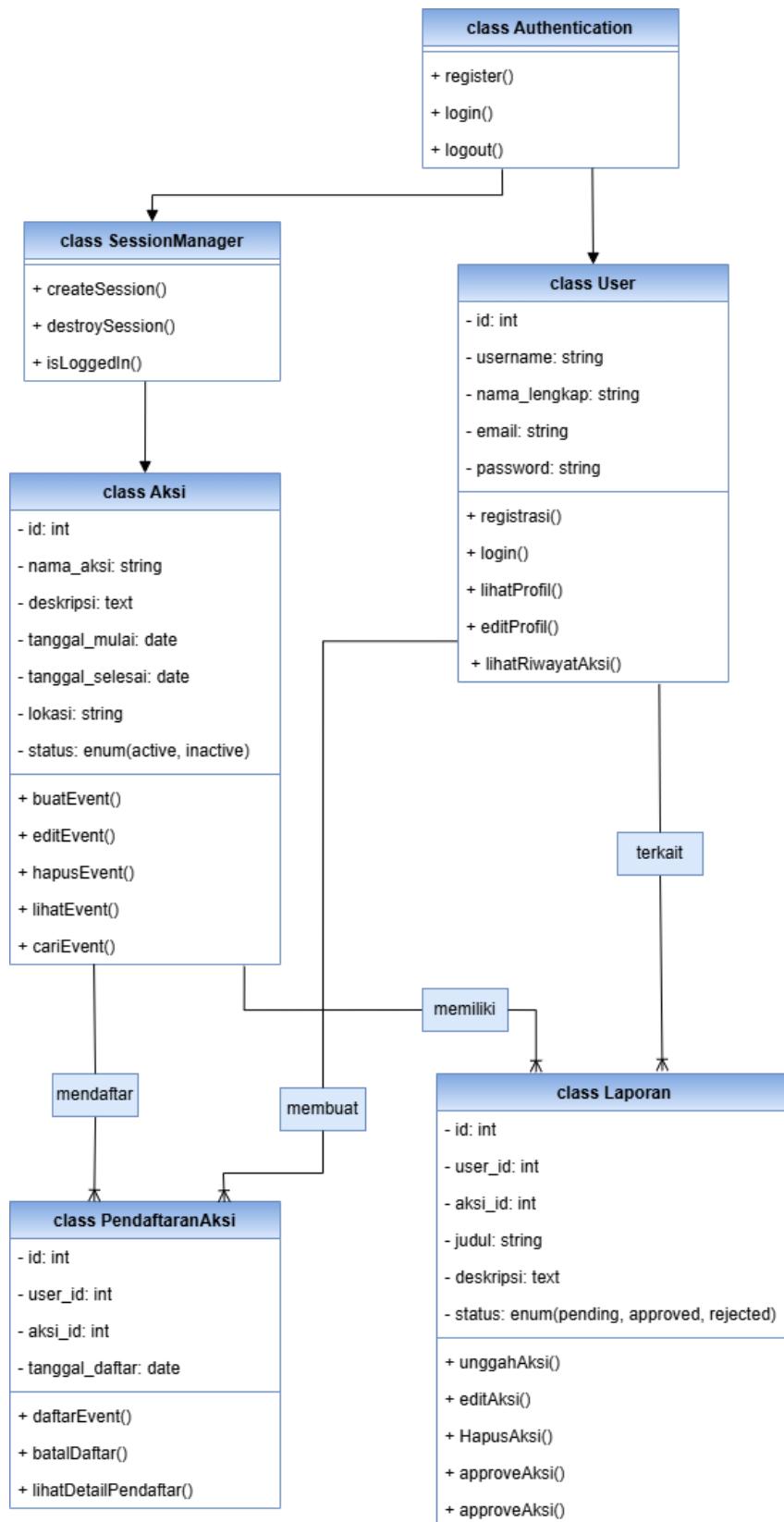
3. Table Pendaftaran Aksi

Tabel ini berfungsi sebagai tabel relasi (junction table) antara users dan aksi. Fungsi utama adalah Mencatat partisipasi user dalam suatu aksi. Ada Constraint penting yaitu Kombinasi (aksi_id, user_id) bersifat unik, sehingga satu user tidak bisa mendaftar aksi yang sama lebih dari satu kali.

4. Table Laporan

Tabel laporan digunakan untuk menyimpan laporan aktivitas yang dibuat oleh user. Fungsi utama adalah Mendokumentasikan aktivitas pengguna, baik yang terkait dengan aksi tertentu maupun aktivitas mandiri.

3.2.2 Class Diagram

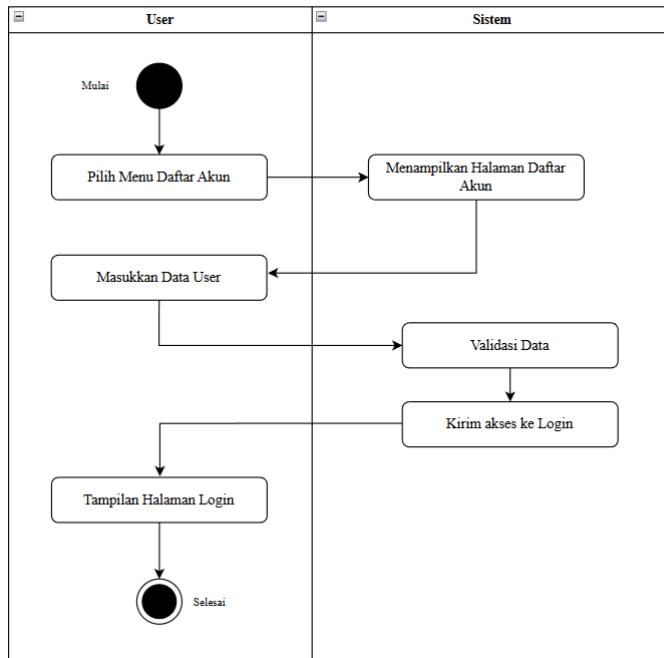


Gambar 3. 3 Class Diagram SahabatBumi

3.3 Alur Aplikasi

3.3.1 Activity Diagram

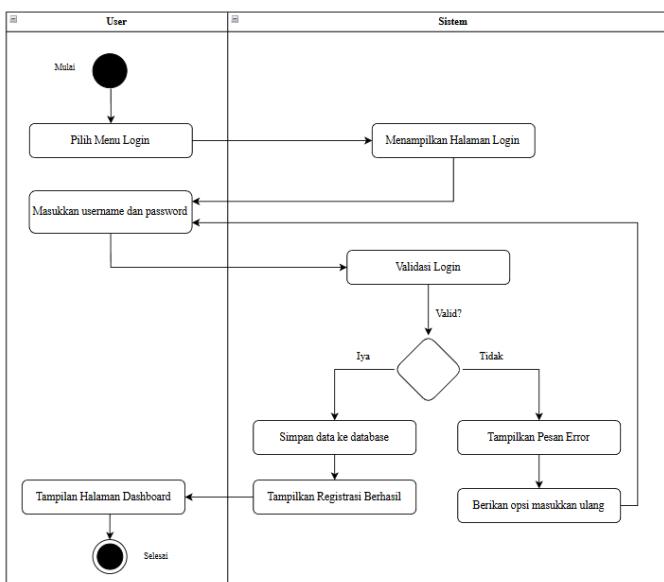
1. Registrasi



Gambar 3. 4 Activity Diagram Registrasi

Activity diagram ini menggambarkan alur proses pendaftaran akun oleh user. User memilih menu daftar akun, mengisi data diri, kemudian sistem melakukan validasi data. Jika valid, sistem mengarahkan user ke halaman login dan proses selesai.

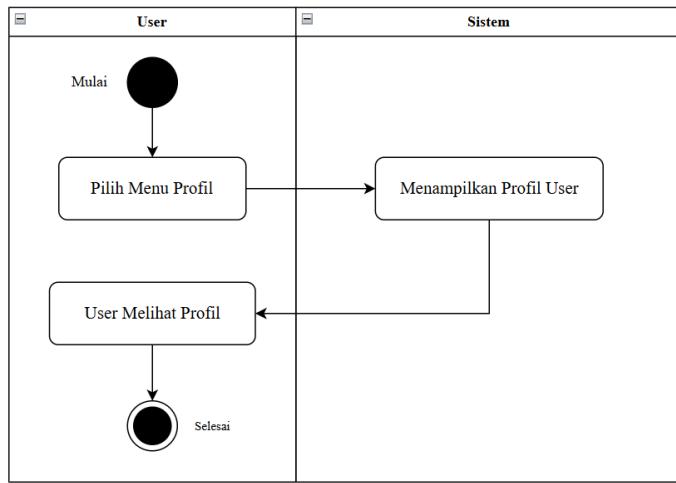
2. Login



Gambar 3. 5 Activity Diagram Login

Activity diagram ini menjelaskan proses login user ke dalam sistem. User memasukkan username dan password, lalu sistem memvalidasi data login. Jika data valid, user diarahkan ke halaman dashboard. Jika tidak valid, sistem menampilkan pesan error dan memberikan opsi untuk menginput ulang data login.

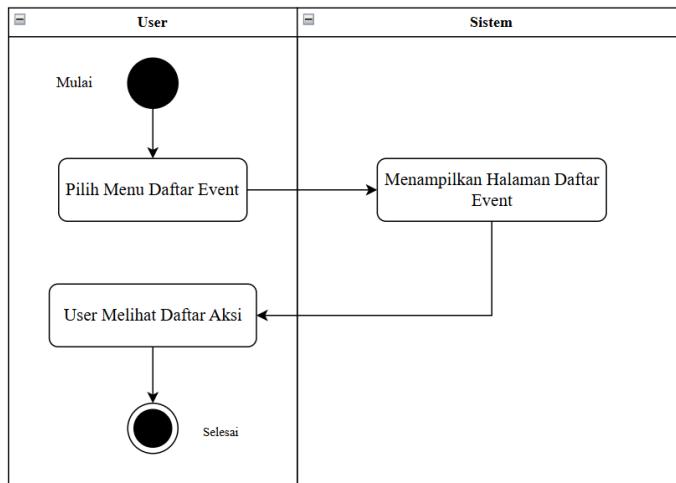
3. Lihat Profil



Gambar 3. 6 Activity Diagram Lihat Profil

Activity diagram ini menunjukkan alur user dalam melihat profil. User memilih menu profil, kemudian sistem menampilkan informasi profil user, dan proses berakhir setelah user melihat data tersebut.

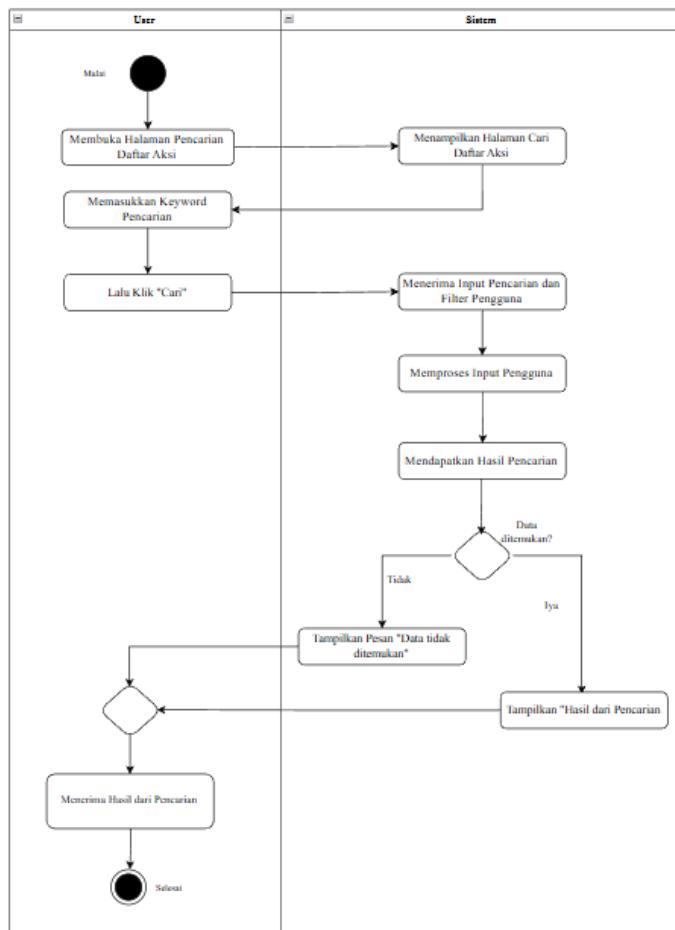
4. Lihat Daftar Event



Gambar 3. 7 Activity Diagram Lihat Daftar Event

Activity diagram ini menunjukkan alur user dalam melihat daftar event. User memilih menu daftar event, kemudian sistem menampilkan informasi daftar event, dan proses berakhir setelah user melihat data tersebut.

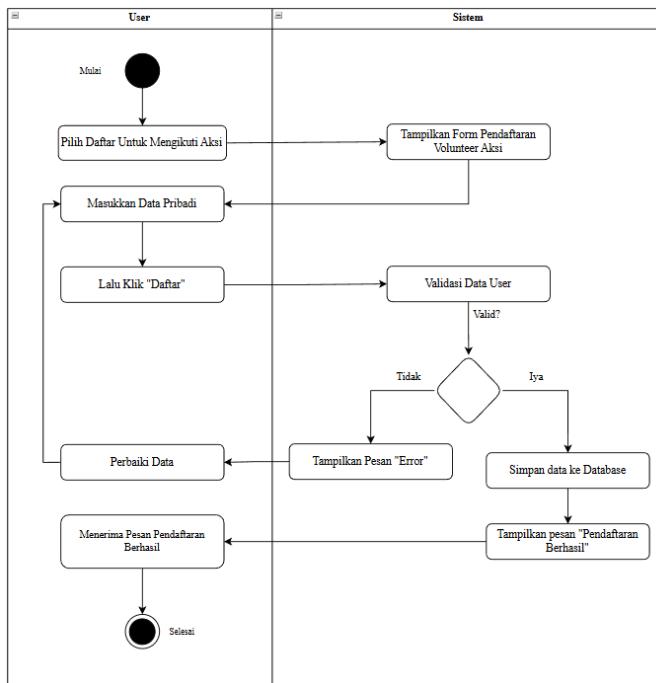
5. Cari Event



Gambar 3. 8 Activity Diagram Cari Event

Activity diagram ini menggambarkan proses user dalam melakukan pencarian daftar aksi. User membuka halaman pencarian, memasukkan keyword, dan menekan tombol cari. Sistem kemudian memproses input pencarian dan menampilkan hasil jika data ditemukan. Apabila data tidak ditemukan, sistem menampilkan pesan bahwa data tidak tersedia, lalu proses pencarian selesai.

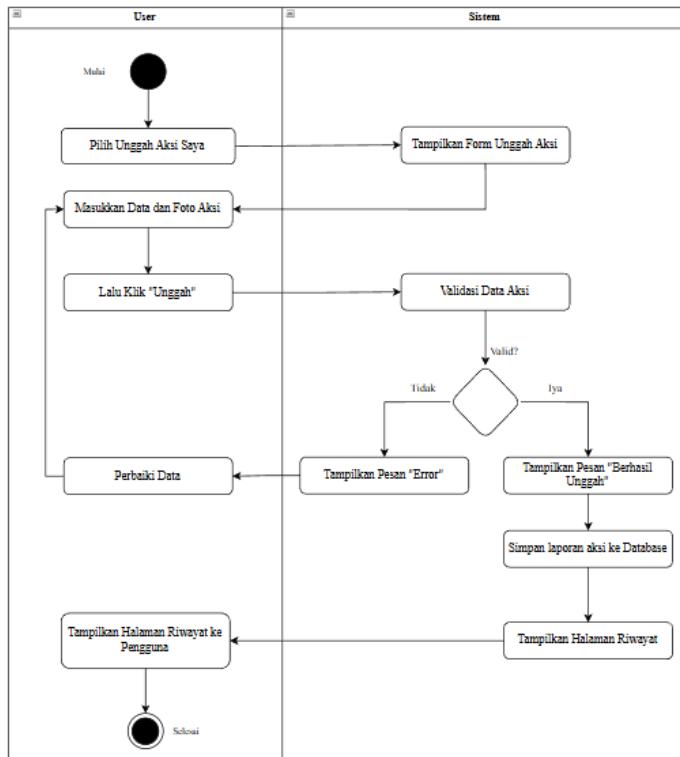
6. Pendaftaran Event



Gambar 3. 9 Activity Diagram Pendaftaran Event

Activity diagram ini menunjukkan proses pendaftaran event oleh user, dimulai dari memilih event, mengisi data pendaftaran, hingga sistem memproses data. Jika data valid, sistem menampilkan pesan pendaftaran berhasil, sedangkan jika tidak valid user diminta mengisi ulang data.

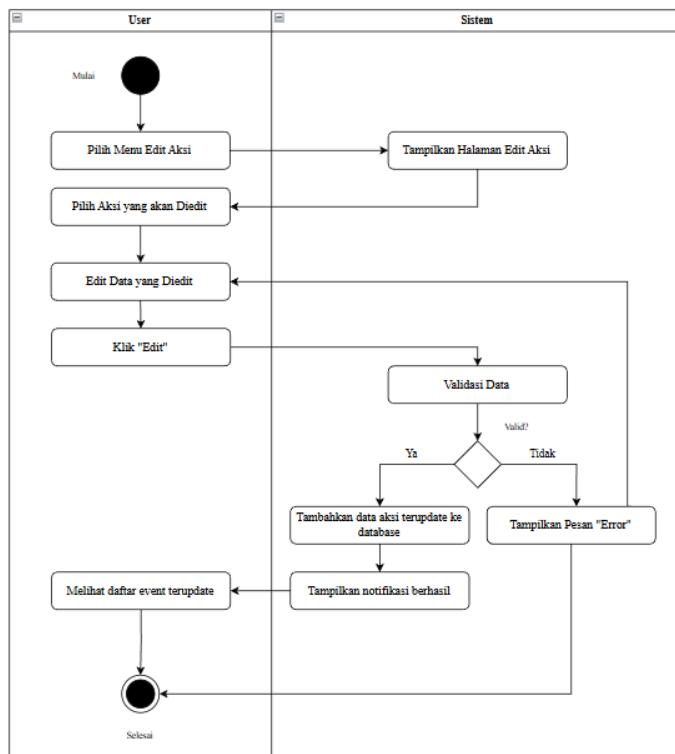
7. Unggah Aksi User



Gambar 3. 10 Activity Diagram Unggah Aksi User

Activity diagram ini menggambarkan proses unggah aksi oleh user, mulai dari pengisian data dan foto, hingga sistem memvalidasi dan menampilkan hasil unggahan.

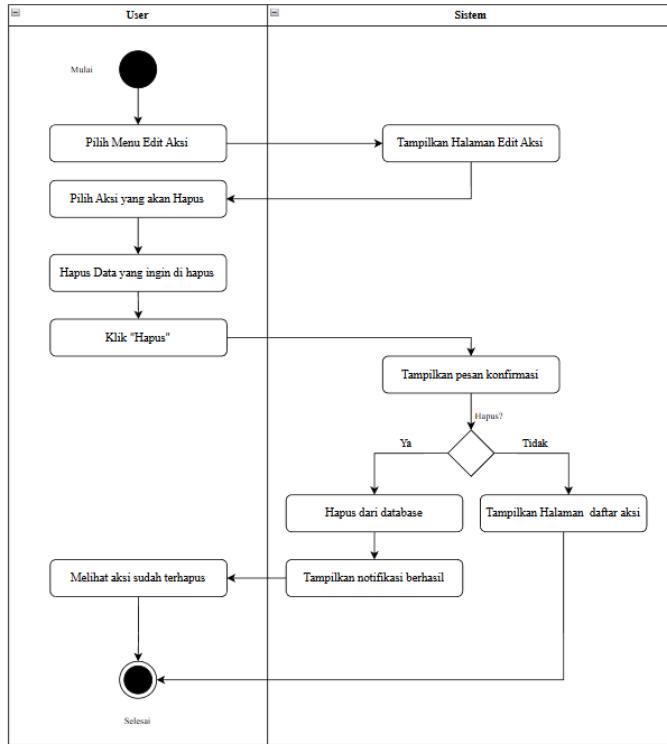
8. Edit Aksi User



Gambar 3. 11 Activity Diagram Edit Aksi User

Activity diagram menunjukkan proses edit aksi, dimulai dari memilih aksi, mengedit data aksi, hingga sistem memproses perubahan. Jika data valid, sistem menampilkan pesan edit aksi berhasil, lalu jika tidak valid user diminta mengisi ulang data.

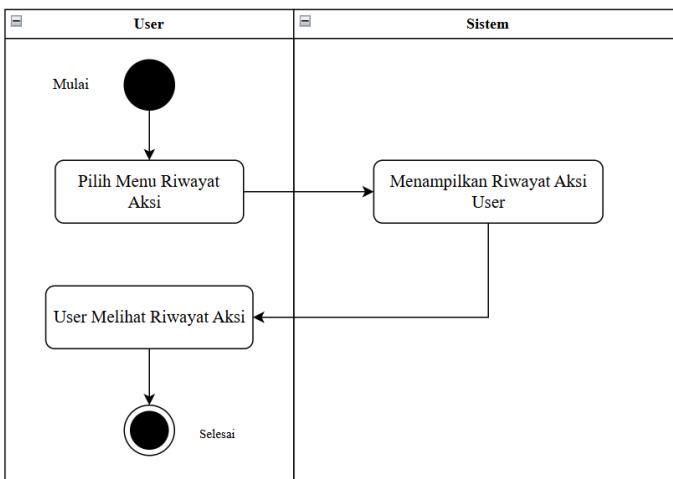
9. Hapus Aksi User



Gambar 3. 12 Activity Diagram Hapus Aksi User

Activity diagram ini menunjukkan proses hapus aksi oleh user, dimulai dari memilih aksi yang akan dihapus dan melakukan konfirmasi, hingga sistem memproses penghapusan.

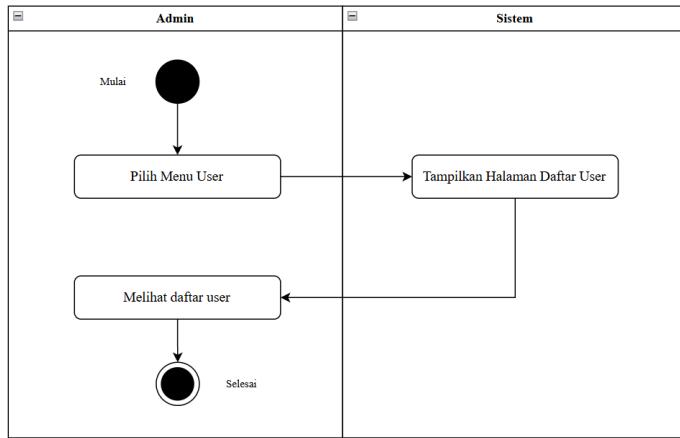
10. Lihat Riwayat Aksi User



Gambar 3. 13 Activity Diagram Lihat Riwayat Aksi User

Activity diagram ini menunjukkan alur user dalam melihat Riwayat aksinya. User memilih menu Riwayat aksi, kemudian sistem menampilkan informasi Riwayat aksi user, dan proses berakhir setelah user melihat data tersebut.

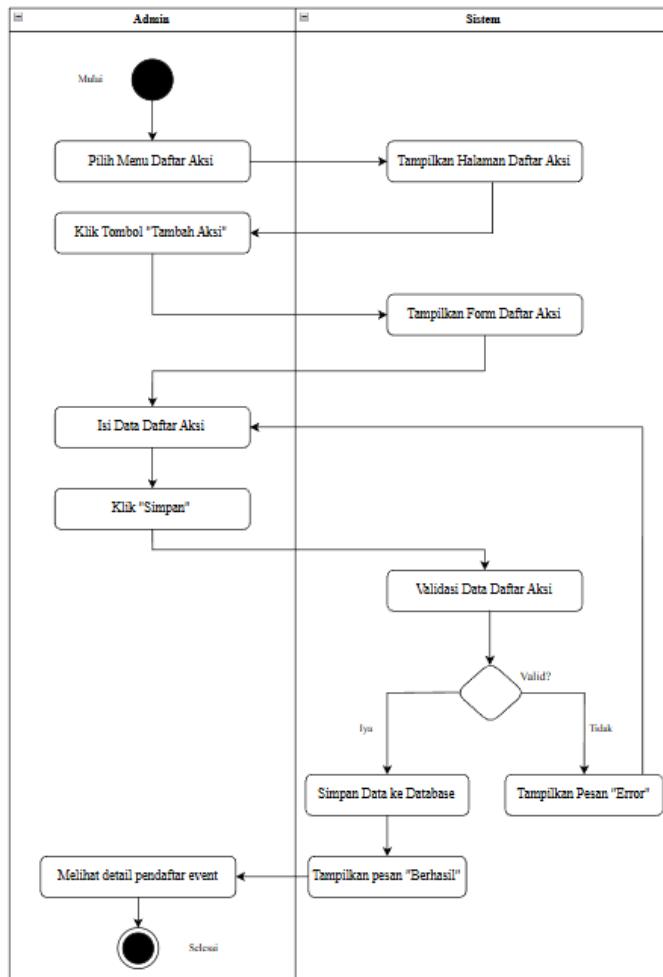
11. Lihat Daftar User Oleh Admin



Gambar 3. 14 Activity Diagram Lihat Daftar User Admin

Activity diagram ini menunjukkan alur Admin untuk melihat daftar user yang login sahabar bumi. Admin memilih menu users, kemudian sistem menampilkan informasi daftar user yang login, dan proses berakhir setelah admin melihat data tersebut.

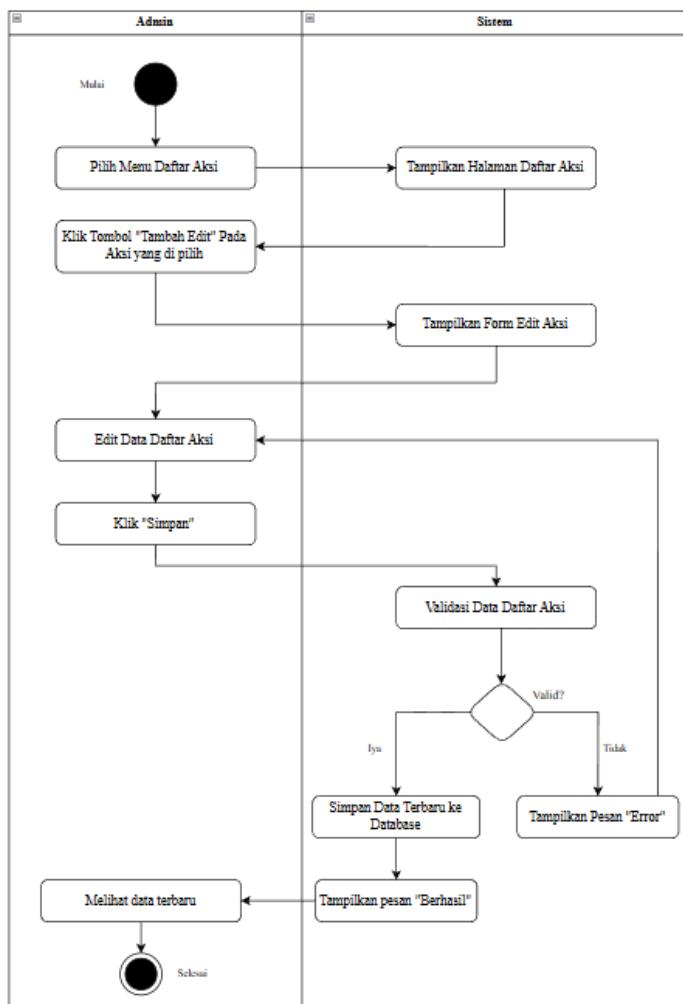
12. Buat Event oleh Admin



Gambar 3. 15 Activity Diagram Buat Event Admin

Activity diagram ini menunjukkan proses admin dalam menambahkan daftar aksi. Admin memilih menu daftar aksi, mengisi form pendaftaran, lalu menyimpan data. Sistem memvalidasi data, menyimpan ke database jika valid, dan menampilkan pesan berhasil, atau menampilkan pesan error jika data tidak valid.

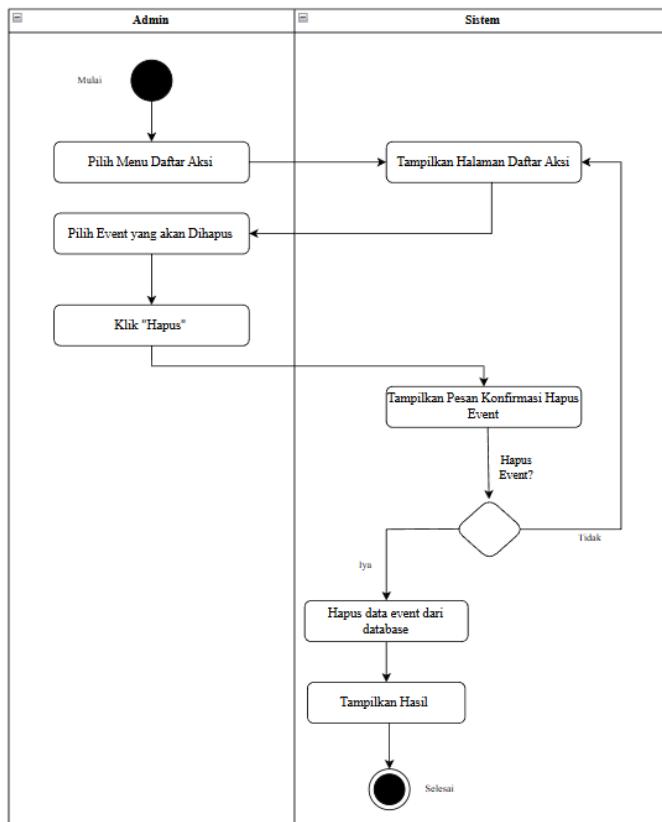
13. Edit Event oleh Admin



Gambar 3. 16 Activity Diagram Edit Event Admin

Activity diagram ini menggambarkan proses admin mengedit data aksi, mulai dari memilih aksi, mengubah data melalui form edit, hingga menyimpan perubahan yang kemudian divalidasi dan diperbarui oleh sistem.

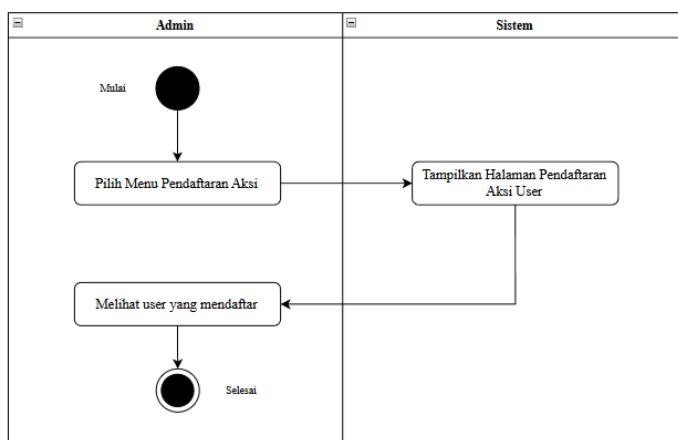
14. Hapus Event oleh Admin



Gambar 3. 17 Activity Diagram Hapus Event Admin

Activity diagram ini menunjukkan proses admin menghapus data aksi, dimulai dari memilih aksi, melakukan konfirmasi penghapusan, hingga sistem menghapus data dari database dan menampilkan hasil.

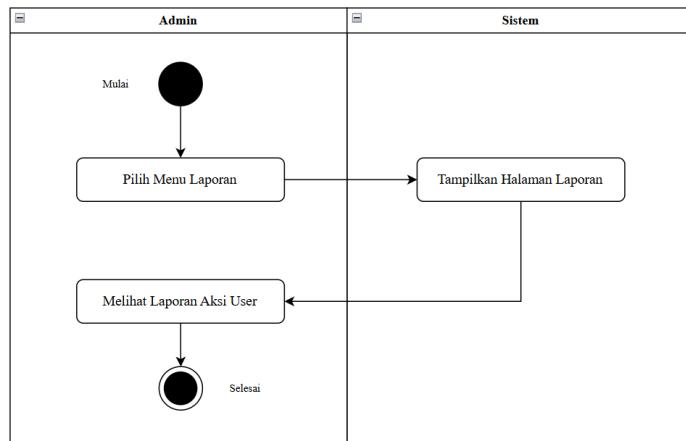
15. Lihat Detail User Mendaftar Event



Gambar 3. 18 Activity Diagram Manajemen Pendaftar Event

Activity diagram ini menunjukkan alur Admin dalam melihat Riwayat siapa saja yang mendaftar pada aksi tertentu. Admin memilih menu Pendaftaran aksi, kemudian sistem menampilkan informasi Riwayat pendafatran aksi user, dan proses berakhir setelah admin melihat data tersebut.

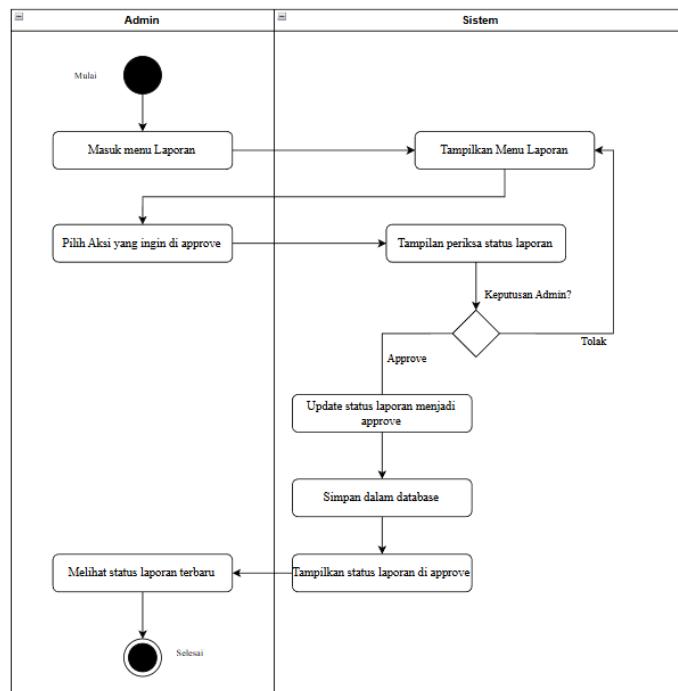
16. Lihat Riwayat Aksi User



Gambar 3. 19 Activity Diagram Lihat Riwayat Aksi User

Activity diagram ini menunjukkan alur Admin dalam melihat Riwayat aksi User. Admin memilih menu Laporan, kemudian sistem menampilkan informasi Riwayat aksi user, dan proses berakhir setelah user melihat data tersebut.

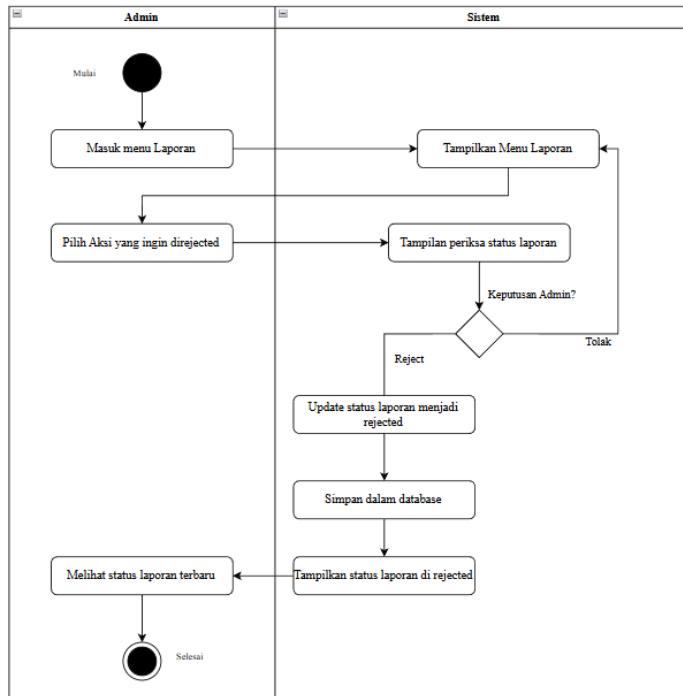
17. Approved Aksi



Gambar 3. 20 Activity Diagram Approved Aksi

Activity diagram ini menggambarkan proses admin melakukan persetujuan laporan, mulai dari memilih laporan, meninjau status, hingga memperbarui dan menyimpan status laporan menjadi disetujui oleh sistem.

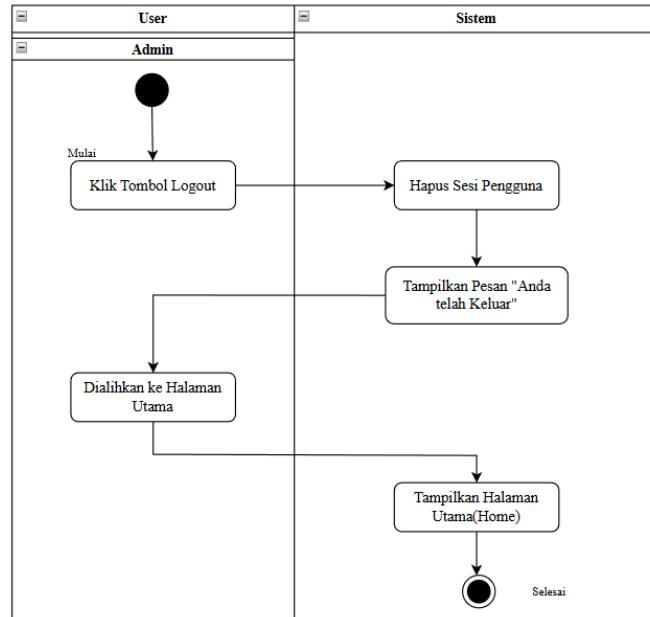
18. Rejected Aksi



Gambar 3. 21 Activity Diagram Rejected Aksi

Activity diagram ini menggambarkan proses admin melakukan penolakan laporan aksi, mulai dari memilih laporan, meninjau status, hingga memperbarui dan menyimpan status laporan tidak disetujui yang ditampilkan oleh sistem.

19. Logout



Gambar 3. 22 Activity Diagram Logout

Activity diagram logout menggambarkan proses admin keluar dari sistem, dimulai dari memilih menu logout hingga sistem mengakhiri sesi dan mengarahkan kembali ke halaman utama home.

BAB IV IMPLEMENTASI

4.1 Lingkungan Pengembangan

4.1.1 Hardware

1. Server/Development Machine (Development)

Spesifikasi komputer untuk tim developer yang mengembangkan aplikasi Sahabat Bumi.

Komponen	Spesifikasi Minimum	Spesifikasi Rekomendasi
Processor	Intel Core i3 / AMD Ryzen 3	Intel Core i5/i7 / AMD Ryzen 5/7
RAM	8 GB DDR4	16 GB DDR4 atau lebih
Storage	500 GB HDD	256 GB SSD (untuk OS + XAMPP + Project)
Monitor	1 x 1920 x 1080 (24")	2 x 1920 x 1080 (24") atau 1 x 4K (27")
Keyboar & Mouse	Standard	Mechanical / Ergonomic
Internet	10 Mbps	50+ Mbps Fiber Optic
OS	Windows 10/11, macOS, Linux	Windows 10/11 atau Ubuntu 20.04+

Table 4. 1 Development Machine

2. Client/User Device

Spesifikasi perangkat pengguna akhir (volunteer, visitor) yang mengakses aplikasi.

Device Type	Spesifikasi	Browser Support
Desktop/Laptop	Windows 10/11, macOS, Linux	Chrome, Firefox, Safari, Edge (latest 2 versions)
Tablet	iPad, Android tablet 7"+	Chrome, Safari, Firefox mobile
SmartPhone	OS 12+, Android 8+	Chrome Mobile, Safari Mobile, Firefox Mobile
Internet Speed	Minimal 2 Mbps	Rekomendasi 10+ Mbps untuk optimal experience
Ram Device	Minimal 2 Gb	Rekomendasi 4+ GB

Screen Size (mobile)	320px width minimum	1920px width (desktop)
-------------------------	---------------------	------------------------

Table 4. 2 User Device

4.1.2 Software

1. Software untuk Development Machine

Software	Versi	Fungsi
XAMPP	7.4+ / 8.0+	Apache Web Server + PHP + MySQL Bundle
PHP	7.4+ / 8.0+	Backend Server-side Language
MySQL / MariaDB	7.4+ / 8.0+	Database Relational Management System
Apache	2.4+	Web Server HTTP
Visual Studio Code	Latest	Code Editor & IDE

Table 4. 3 Software Development Machine

2. Frontend Technologies (Actual)

Technology	Versi	Fungsi
HTML5	Standard	Markup & Semantic Structure
CSS3	Standard	Styling & Responsive Layout
JavaScript	ES6+	DOM Manipulation & Interactivity
Font Awesome Icons	6.4.0	Icon Library (UI Elements)
Google Fonts (Poppins)	Latest	Typography
Responsive Design	Mobile-First	Viewport Meta Tags

Table 4. 4 Frontend Technologies

3. Backend Technologies

Technology	Versi	Fungsi
PHP Native	7.4+ / 8.0+	PHP GD + File I/O
MySQLi (Object-Oriented)	Built-in PHP	Database Connection & Queries
Prepared Statements	MySQLi feature	SQL Injection Prevention
Sessions (\$_SESSION)	Built-in PHP	User Authentication & State Management

Superglobals	Built-in PHP	Request handling (\$_GET, \$_POST, \$_FILES)
File Upload Handling	PHP GD + File I/O	Image/Document Upload & Validation

Table 4. 5 Backend Technologies

4. Database & Related Tools

Tool	Versi	Fungsi
MySQL Server	5.7+ / 8.0+	Relational Database (RDBMS)
phpMyAdmin	5.0+	Web-based MySQL GUI

Table 4. 6 Database Tools

4.2 Struktur Folder Proyek

Struktur folder dirancang memisahkan tampilan publik (halaman user), area admin, asset statis (CSS/JS/IMG), dan berkas server (upload + konfigurasi). Ini memudahkan pengelolaan hak akses, maintenance, dan pengembangan fitur.

1. Root (root project) :

- config.php : File konfigurasi utama mengatur koneksi database, pembuatan database/tabel bila belum ada, helper kompatibilitas, dan pembuatan folder uploads.
- login.php, sign-up.php, logout.php, navbar.php, footer.php : Halaman akses dan komponen tampilan bersama.

2. Folder Admin :

- Berisi halaman untuk administrasi seperti dashboard.php, users.php, daftar-aksi.php, pendaftaran-vol.php, laporanaksi.php dan subfolder uploads untuk gambar/event yang diunggah melalui panel admin.
- Kode admin mengimplementasikan CRUD event (aksi), manajemen laporan, dan manajemen pengguna.

3. Folder User :

- Halaman publik yang diakses user terdaftar: homepage.php, daftar-event.php, daftar-laporan.php, profil.php, unggah.php, form-pendaftaran-aksi.php dan subfolder uploads untuk file yang dikaitkan user.

4. Folder Css ;

- Semua stylesheet (*.css) di css untuk tiap halaman/komponen.

5. Folder Js :

- Semua skrip frontend (*.js) di js, mis. script_sign-up.js, script_daftar-event.js.

6. Folder Img :

- img menyimpan aset grafis (logo, ikon, placeholder).

7. Folder Uploads :

- Menyimpan file yang diunggah (gambar event, lampiran laporan). config.php memastikan folder dibuat dan diatur permission-nya bila memungkinkan.

4.3 Struktur Code Script dan Program

1. Konfigurasi & Inisialisasi (config.php):

- Membuka/menginisialisasi session, membuat koneksi mysqli, meng-set charset ke utf8mb4.
- Membuat database (sahabat_bumi) jika belum ada dan membuat tabel utama (users, aksi, pendaftaran_aksi, laporan) menggunakan SQL CREATE TABLE IF NOT EXISTS.
- Menyediakan helper stmt_get_assoc untuk kompatibilitas ketika get_result() tidak tersedia.
- Menjamin kolom tambahan (compatibility) dengan ensure_column_exists() sehingga fitur baru tidak merusak instalasi lama.
- Menyiapkan folder uploads dan memastikan dapat ditulis.

2. Autentikasi & Autorisasi

- sign-up.php: Validasi input (username, email, password), cek uniknya username/email, hashing password dengan password_hash(), menyimpan user baru ke tabel users (role default user).
- login.php: Mengambil user berdasarkan username, verifikasi password dengan password_verify(), menyimpan \$_SESSION['user_id'], \$_SESSION['user_role'] dan redirect berdasarkan role (admin → dashboard.php, user → profil.php).
- Session-based redirects mencegah akses halaman tertentu bila belum login.

3. Fitur CRUD Admin

- daftar-aksi.php menggabungkan tampilan daftar dan endpoint AJAX untuk operasi CRUD :
 - Endpoint action=create (POST): validasi fields, proses upload gambar, simpan record ke tabel aksi.
 - Endpoint action=read (GET): mengembalikan JSON semua aksi.

- Endpoint action=update dan action=delete: mengupdate data termasuk mengganti/men-delete file gambar secara aman.
- Penggunaan prepared statements (\$conn->prepare) untuk mencegah SQL Injection dan penanganan error yang mengembalikan JSON pada panggilan AJAX.

4. Form Handler User

- a. form-pendaftaran-aksi.php :
 - Memverifikasi bahwa aksi_id valid dan user login.
 - Cek duplikasi pendaftaran (UNIQUE KEY unique_aksi_user di DB), insert ke tabel pendaftaran_aksi.
 - Menyajikan feedback (\$success_msg, \$error_msg) pada halaman setelah submit.

5. Uploads & File Handling

- a. Validasi ekstensi dan ukuran file (contoh: hanya jpg,jpeg,png,gif dan max 5MB di daftar-aksi.php).
- b. Penamaan file unik: event_<timestamp>_<unqid>.<ext> untuk mencegah bentrok nama dan caching issues.
- c. Hapus file lama saat update/delete agar storage tidak penuh.

6. Frontend & Interaksi

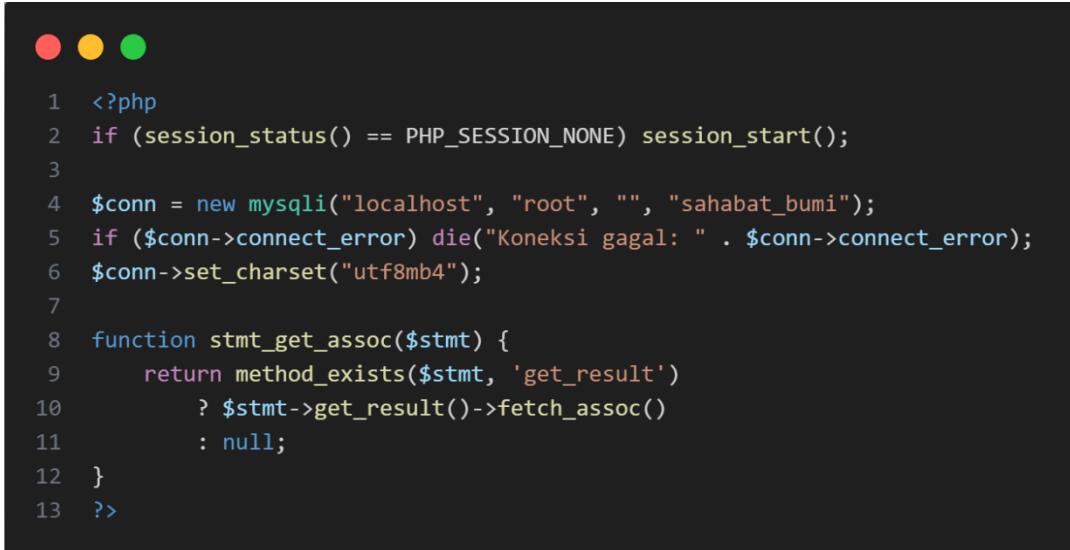
- a. Per halaman ada CSS khusus di css untuk menjaga konsistensi tampilan.
- b. JS di js meng-handle validasi klien, UX (animasi fokus input, alert, dan placeholder integrasi OAuth).
- c. Admin menggunakan Bootstrap + modul JS untuk modal/form dan mengonsumsi endpoint JSON AJAX.

7. Keamanan & Best Practices Implemented

- a. Prepared statements untuk query yang melibatkan input user.
- b. Hashing password standar PHP (password_hash/password_verify).
- c. Pembatasan upload (tipe file & ukuran) dan sanitasi output (htmlspecialchars) saat menampilkan data.
- d. Penanganan error yang membungkus output AJAX agar tidak bocor stack trace ke client.

4.4 Code Snippet

1. Database Connection (config.php)



```
1 <?php
2 if (session_status() == PHP_SESSION_NONE) session_start();
3
4 $conn = new mysqli("localhost", "root", "", "sahabat_bumi");
5 if ($conn->connect_error) die("Koneksi gagal: " . $conn->connect_error);
6 $conn->set_charset("utf8mb4");
7
8 function stmt_get_assoc($stmt) {
9     return method_exists($stmt, 'get_result')
10        ? $stmt->get_result()->fetch_assoc()
11        : null;
12 }
13 ?>
```

Gambar 4. 1 Code Database Connection

Penjelasan :

- session_start(): Mengaktifkan PHP session untuk tracking user login. Check status terlebih dahulu untuk mencegah error "session already started".
- new mysqli(): Membuat koneksi ke database menggunakan MySQLi object-oriented (lebih aman & modern).
- set_charset("utf8mb4"): Set charset ke UTF8MB4 untuk support karakter spesial dan emoji dari semua user.
- Error checking: Jika koneksi gagal, tampilkan pesan dan stop script execution.
- stmt_get_assoc(): Helper function untuk kompatibilitas; beberapa server PHP tidak support get_result(), fungsi ini fallback ke method yang compatible. Digunakan untuk fetch single row dari prepared statement.

2. Role-Based Access Control (Auth Check)



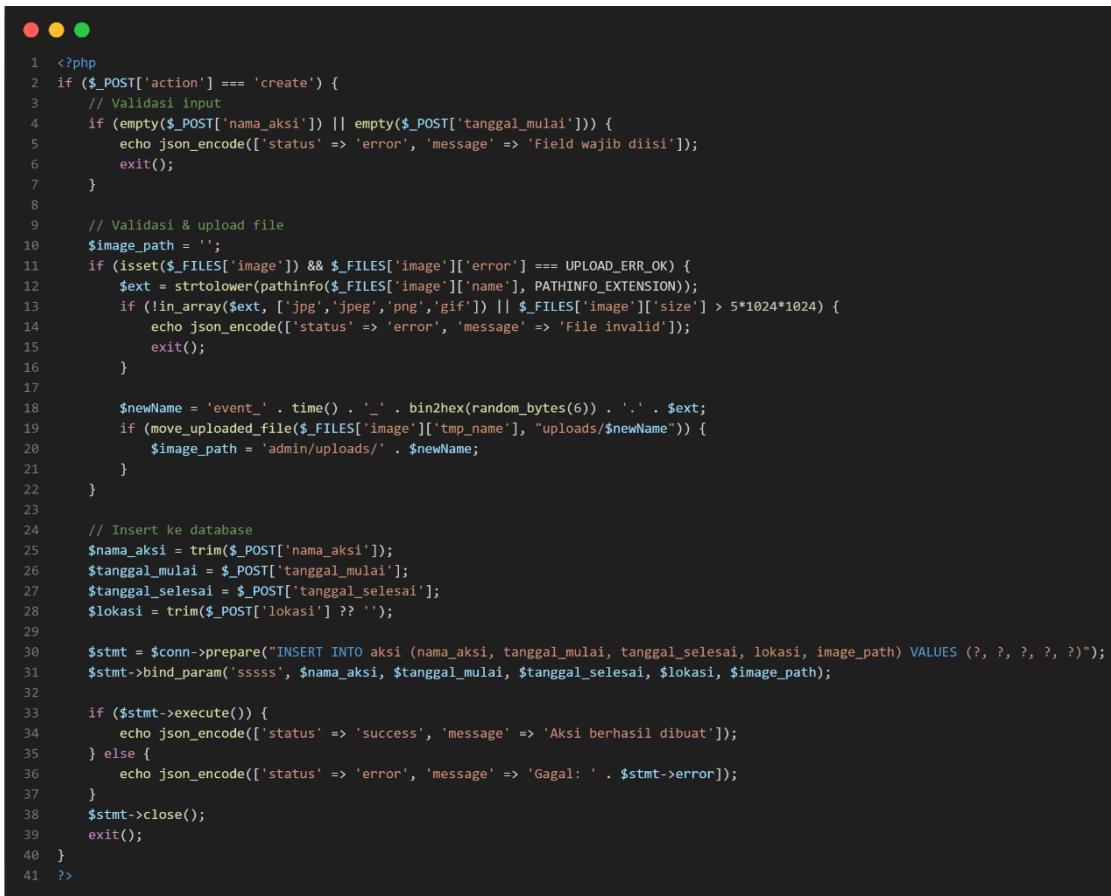
```
1 <?php
2 include '../user/config.php';
3 if (!isset($_SESSION['user']) || $_SESSION['user']['role'] !== 'admin') {
4     header('Location: ../user/login.php');
5     exit("Akses ditolak.");
6 }
7 ?>
```

Gambar 4. 2 Code Role-Based Acces Control

Penjelasan :

- Session check: Verifikasi user sudah login (session user ada di `$_SESSION`). Jika belum, redirect ke login.
- Role verification: Setelah login, cek apakah user role adalah 'admin'. Jika user adalah 'user' biasa, tidak boleh akses halaman admin.
- Header redirect: `header('Location: ...')` mengarahkan user ke login page. `exit()` memastikan script stop di sini (tidak melanjutkan eksekusi).
- Security: Dengan check ini, hanya admin yang bisa akses file `daftar-aksi.php`, `laporanaksi.php`, dll. Jika user mengakses langsung URL admin, akan di-redirect ke login.

3. CRUD Create (Insert Event dengan Upload Gambar)



```

1 <?php
2 if ($_POST['action'] === 'create') {
3     // Validasi input
4     if (empty($_POST['nama_aksi']) || empty($_POST['tanggal_mulai'])) {
5         echo json_encode(['status' => 'error', 'message' => 'Field wajib diisi']);
6         exit();
7     }
8
9     // Validasi & upload file
10    $image_path = '';
11    if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
12        $ext = strtolower(pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION));
13        if (!in_array($ext, ['jpg', 'jpeg', 'png', 'gif']) || $_FILES['image']['size'] > 5*1024*1024) {
14            echo json_encode(['status' => 'error', 'message' => 'File invalid']);
15            exit();
16        }
17
18        $newName = 'event_' . time() . '_' . bin2hex(random_bytes(6)) . '.' . $ext;
19        if (move_uploaded_file($_FILES['image']['tmp_name'], "uploads/$newName")) {
20            $image_path = 'admin/uploads/' . $newName;
21        }
22    }
23
24    // Insert ke database
25    $nama_aksi = trim($_POST['nama_aksi']);
26    $tanggal_mulai = $_POST['tanggal_mulai'];
27    $tanggal_selesai = $_POST['tanggal_selesai'];
28    $lokasi = trim($_POST['lokasi'] ?? '');
29
30    $stmt = $conn->prepare("INSERT INTO aksi (nama_aksi, tanggal_mulai, tanggal_selesai, lokasi, image_path) VALUES (?, ?, ?, ?, ?)");
31    $stmt->bind_param('sssss', $nama_aksi, $tanggal_mulai, $tanggal_selesai, $lokasi, $image_path);
32
33    if ($stmt->execute()) {
34        echo json_encode(['status' => 'success', 'message' => 'Aksi berhasil dibuat']);
35    } else {
36        echo json_encode(['status' => 'error', 'message' => 'Gagal: ' . $stmt->error]);
37    }
38    $stmt->close();
39    exit();
40 }
41 ?>

```

Gambar 4. 3 Code CRUD Create

Penjelasan :

- Input validation: Cek apakah field required (`nama_aksi`, `tanggal_mulai`) sudah diisi. Jika kosong, return error JSON.
- File upload validation:
 - `$_FILES['image']['error'] === UPLOAD_ERR_OK` — cek bahwa file berhasil diupload (tidak ada error).

- pathinfo(..., PATHINFO_EXTENSION) — ekstrak extension file (jpg, png, dll).
- in_array(\$ext, [...]) — whitelist extensions untuk security (prevent upload .php/.exe/dll).
- \$_FILES['image']['size'] > 5*1024*1024 — cek ukuran file (limit 5MB).
- Generate unique filename: 'event_'. time() . '_'. bin2hex(random_bytes(6)) . '' . \$ext membuat nama unik (timestamp + random 6 bytes) untuk prevent filename collision dan directory traversal attacks.
- move_uploaded_file(): Pindahkan file dari temporary folder ke folder destination (uploads/).
- Prepared statement: \$stmt->prepare() + bind_param() + execute() mencegah SQL injection. Setiap ? adalah placeholder untuk value yang aman.
- JSON response: Return JSON dengan status dan message untuk AJAX handler di JavaScript mengetahui success/error.

4. CRUD Read (Fetch Event dengan JSON Response)

```

● ● ●
1 <?php
2 if ($_GET['action'] === 'read') {
3     $result = $conn->query("SELECT id, nama_aksi, kategori, tanggal_mulai, tanggal_selesai, lokasi FROM aksi ORDER BY created_at DESC");
4
5     $data = [];
6     while ($row = $result->fetch_assoc()) {
7         $data[] = $row;
8     }
9
10    header('Content-Type: application/json');
11    echo json_encode(['status' => 'success', 'total' => count($data), 'data' => $data]);
12    exit();
13 }
14 ?>

```

Gambar 4. 4 Code CRUD Read

Penjelasan :

- SELECT query: Ambil data event dari tabel aksi, urutkan by created_at DESC (terbaru di atas).
- fetch_assoc(): Loop setiap row dari hasil query. Setiap row menjadi associative array (key = nama kolom).
- Array push: Akumulasi setiap row ke dalam array \$data.
- JSON response structure: Return consistent JSON dengan:
 - status — 'success' atau 'error'
 - total — jumlah records
 - data — array events untuk ditampilkan ke datatable

- Header Content-Type: Set response header ke JSON sehingga browser/AJAX tahu menerima JSON, bukan HTML.
- Used by: AJAX loadTable() function di JavaScript untuk populate tabel.

5. CRUD Update (Edit Event dengan Replace Gambar)



```

1 <?php
2 if ($_POST['action'] === 'update') {
3     $id = intval($_POST['id']);
4     $nama_aksi = trim($_POST['nama_aksi']);
5     $tanggal_mulai = $_POST['tanggal_mulai'];
6
7     // Fetch current image
8     $stmt = $conn->prepare("SELECT image_path FROM aksi WHERE id = ?");
9     $stmt->bind_param('i', $id);
10    $stmt->execute();
11    $row = $stmt->get_result();
12    $image_path = $row['image_path'] ?? '';
13    $stmt->close();
14
15    // Handle new image
16    if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
17        if (!empty($image_path)) @unlink("../" . $image_path); // Delete old
18
19        $ext = strtolower(pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION));
20        $newName = 'event_' . time() . '_' . bin2hex(random_bytes(6)) . '.' . $ext;
21        move_uploaded_file($_FILES['image']['tmp_name'], "uploads/$newName");
22        $image_path = 'admin/uploads/' . $newName;
23    }
24
25    // Update database
26    $stmt = $conn->prepare("UPDATE aksi SET nama_aksi = ?, tanggal_mulai = ?, image_path = ? WHERE id = ?");
27    $stmt->bind_param('sssi', $nama_aksi, $tanggal_mulai, $image_path, $id);
28    echo json_encode($stmt->execute() ? ['status' => 'success'] : ['status' => 'error']);
29    exit();
30 }
31 ?>

```

Gambar 4. 5 Code CRUD Update

Penjelasan :

- intval(\$id): Convert ID dari string ke integer untuk SQL safety.
- Fetch current image: Query SELECT untuk ambil image_path lama. Perlu tahu path ini untuk delete file lama jika user upload gambar baru.
- stmt_get_assoc(): Helper function yang di-define di config.php; fetch single row dari prepared statement.
- @unlink(): Delete file lama dari disk sebelum upload yang baru. @ suppress warning jika file tidak ada. Path harus .[htdocs](http://_vscodecontentref_/3) . \$image_path karena file berada di parent directory.
- Upload new file: Generate nama unik & pindahkan file baru ke folder uploads/.
- UPDATE statement: Update kolom nama_aksi, tanggal_mulai, image_path di tabel aksi WHERE id sesuai. Menggunakan prepared statement untuk SQL injection prevention.

- Ternary operator: \$stmt->execute() ? ['status' => 'success'] : ['status' => 'error'] — jika execute sukses return success JSON, jika gagal return error JSON.

6. CRUD Delete (Hapus Event & File)



```

1  <?php
2  if ($_POST['action'] === 'delete') {
3      $id = intval($_POST['id']);
4
5      // Fetch & delete file
6      $stmt = $conn->prepare("SELECT image_path FROM aksi WHERE id = ?");
7      $stmt->bind_param('i', $id);
8      $stmt->execute();
9      $row = $stmt->get_result();
10     $stmt->close();
11
12     if ($row && !empty($row['image_path'])) {
13         @unlink("../" . $row['image_path']);
14     }
15
16     // Delete from DB
17     $stmt = $conn->prepare("DELETE FROM aksi WHERE id = ?");
18     $stmt->bind_param('i', $id);
19     echo json_encode($stmt->execute() ? ['status' => 'success'] : ['status' => 'error']);
20     exit();
21 }
22 ?>

```

Gambar 4. 6 Code CRUD Delete

Penjelasan :

- Fetch image path: Query SELECT untuk ambil path file yang akan didelete. Jangan langsung delete record dari DB sebelum tahu lokasi file.
- Delete file dari disk: Jika row ada dan image_path tidak kosong, delete file dari folder uploads/. @unlink() prevent warning jika file sudah tidak ada.
- Delete from database: DELETE statement dengan WHERE id = ?. Karena database schema FK pendaftaran_aksi.aksi_id punya ON DELETE CASCADE, saat event dihapus, semua pendaftaran event otomatis terhapus juga (data consistency).
- Two-phase delete: 1) Delete file from disk, 2) Delete record from DB. Jika hanya delete DB, file orphan akan tetap menempati disk space.

7. Error Handling dengan Error Handler & Exception Handler (AJAX Safe)



```
1 <?php
2 if (isset($_REQUEST['action'])) {
3     ini_set('display_errors', 0);
4     error_reporting(E_ALL);
5     header('Content-Type: application/json; charset=utf-8');
6
7     set_error_handler(function($errno, $errstr, $errfile, $errline) {
8         http_response_code(500);
9         error_log("[$errfile:$errline] $errstr");
10        echo json_encode(['status' => 'error', 'message' => 'Server error']);
11        exit();
12    });
13 }
14 ?>
```

Gambar 4. 7 Code Error Handling

Penjelasan :

- Conditional activation: Hanya aktifkan error handler jika ada \$_REQUEST['action'] (AJAX request). Kalau tidak ada action, biarkan error display normal untuk development.
- ini_set('display_errors', 0): Disable error display ke output (prevent mencampurkan error detail dengan JSON response).
- error_reporting(E_ALL): Tangkap semua jenis error (E_WARNING, E_NOTICE, dll) untuk logging.
- header('Content-Type: application/json'): Set response header agar client tahu ini JSON, bukan HTML.
- set_error_handler(): Custom function untuk intercept PHP errors. Saat error terjadi:
 - http_response_code(500) — set HTTP status 500 (Internal Server Error)
 - error_log() — log error ke server log file (biasanya /var/log/apache2/error.log)
 - echo json_encode() — return JSON error response ke client (aman, tidak expose detail)
- exit() — stop script setelah handle error.

8. Input Validation & Sanitization (Form & File)

```
1  <?php
2  function validate_input($nama_aksi, $tanggal_mulai, $tanggal_selesai) {
3      $errors = [];
4      if (strlen($nama_aksi) < 3) $errors[] = 'Nama minimal 3 karakter';
5      if (strtotime($tanggal_selesai) < strtotime($tanggal_mulai)) {
6          $errors[] = 'Tanggal selesai tidak boleh lebih awal';
7      }
8      return $errors;
9  }
10
11 function validate_file($file) {
12     $allowed = ['jpg', 'jpeg', 'png', 'gif'];
13     $ext = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
14
15     if (!in_array($ext, $allowed)) return 'Format tidak didukung';
16     if ($file['size'] > 5*1024*1024) return 'Ukuran max 5MB';
17     if (!getimagesize($file['tmp_name'])) return 'File bukan gambar';
18
19     return null;
20 }
21
22 $errors = validate_input($nama_aksi, $tanggal_mulai, $tanggal_selesai);
23 if (isset($_FILES['image'])) {
24     $file_error = validate_file($_FILES['image']);
25     if ($file_error) $errors[] = $file_error;
26 }
27 if ($errors) {
28     echo json_encode(['status' => 'error', 'errors' => $errors]);
29     exit();
30 }
31 ?>
```

Gambar 4. 8 Code Input Validation & Sanitization

Penjelasan :

- validate_input(): Function untuk validate field form:
 - strlen(\$nama_aksi)<3 — cek panjang nama minimal 3 karakter (prevent data garbage).
 - strtotime() convert string date ke timestamp, compare tanggal selesai >= tanggal mulai (business logic validation).
 - Return array errors untuk caller bisa tahu apa yang salah.
- validate_file(): Function untuk validate file upload:

- `in_array($ext, $allowed)` — whitelist extension (security: prevent .php/.exe upload).
- `$file['size'] > 5*1024*1024` — cek ukuran file (prevent bandwidth abuse).
- `getimagesize($file['tmp_name'])` — verify file adalah benar-benar gambar (prevent fake JPG yang contain PHP code).
- Return format: Return string error message jika ada masalah, null jika valid.
- Accumulate errors: Kumpulkan semua input & file errors ke array `$errors`. Jika ada error, return semua sekaligus ke client.

9. JavaScript AJAX Call dengan FormData (untuk Upload File)

```

● ● ●

1  function saveAksi() {
2      const formData = new FormData(document.getElementById('aksiForm'));
3      formData.append('action', document.getElementById('editId').value ? 'update' : 'create');
4      if (document.getElementById('editId').value) {
5          formData.append('id', document.getElementById('editId').value);
6      }
7
8      fetch('daftar-aksi.php', {method: 'POST', body: formData})
9          .then(r => r.json())
10         .then(d => {
11             if (d.status === 'success') {
12                 alert(d.message);
13                 loadTable();
14             } else {
15                 alert('Error: ' + d.message);
16             }
17         });
18     }
19
20 // Load table
21 function loadTable() {
22     fetch('daftar-aksi.php?action=read')
23         .then(r => r.json())
24         .then(d => {
25             let html = '';
26             d.data.forEach(item => {
27                 html += `<tr><td>${item.nama_aksi}</td><td>${item.lokasi}</td>
28                     <td><button onclick="editAksi(${item.id})>Edit</button>
29                     <button onclick="deleteAksi(${item.id})>Delete</button></td></tr>`;
30             });
31             document.querySelector('table tbody').innerHTML = html;
32         });
33     }
34
35 // Delete
36 function deleteAksi(id) {
37     if (confirm('Yakin?')) {
38         fetch('daftar-aksi.php', {
39             method: 'POST',
40             body: new URLSearchParams({action: 'delete', id: id})
41         }).then(r => r.json()).then(d => {
42             alert(d.message);
43             loadTable();
44         });
45     }
46 }

```

Gambar 4. 9 Code JavaScript AJAX Call dengan FormData

Penjelasan :

- saveAksi(): Function untuk submit form (create/update):
 - new FormData() — automatic serialize semua form fields + file input (multipart/form-data).
 - formData.append('action', ...) — ternary operator: jika editId ada (edit mode), action = 'update'; jika kosong (create mode), action = 'create'.
 - fetch(..., {method: 'POST', body: formData}) — send AJAX POST request ke daftar-aksi.php dengan form data.
 - .then(r => r.json()) — parse response sebagai JSON.
 - .then(d => {...}) — handle respons JSON: jika success, show alert & reload table; jika error, show error message.
- loadTable(): Function untuk fetch & display semua event ke datatable:
 - fetch('daftar-aksi.php?action=read') — GET request ke backend untuk ambil data.
 - d.data.forEach() — loop setiap event, construct HTML row (<tr>...</tr>).
 - template literal `<tr>...</tr>` — construct dynamic HTML.
 - document.querySelector('table tbody').innerHTML = html — replace table body dengan HTML baru (re-render datatable).
- deleteAksi(): Function untuk delete:
 - confirm('Yakin?') — ask confirmation sebelum delete (UX best practice).
 - new URLSearchParams({action: 'delete', id: id}) — create URL-encoded form data (action=delete&id=123).
 - Setelah delete sukses, reload table & show success message.

BAB V HASIL APLIKASI

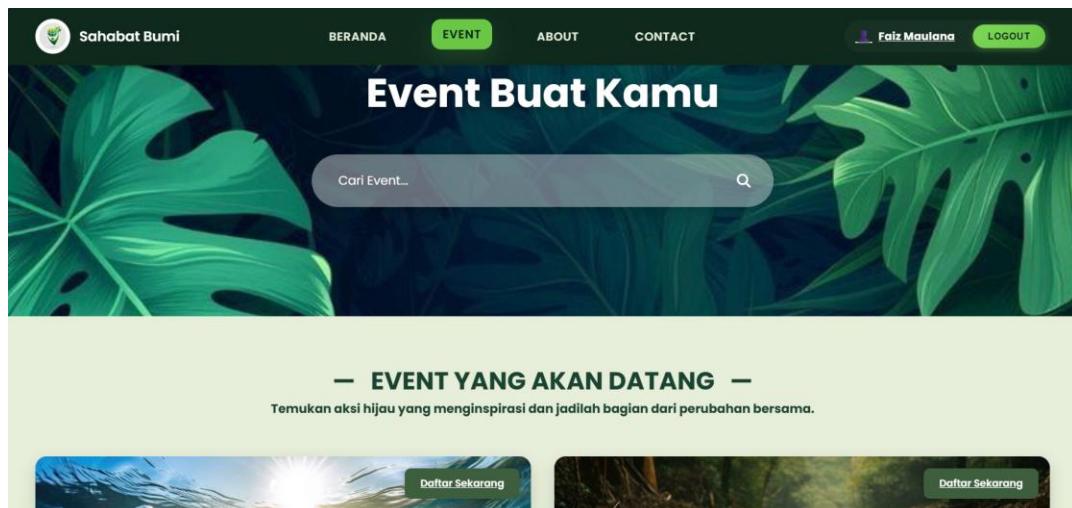
5.1 Tampilan Aplikasi

1. Tampilan User
 - a. Menu Beranda



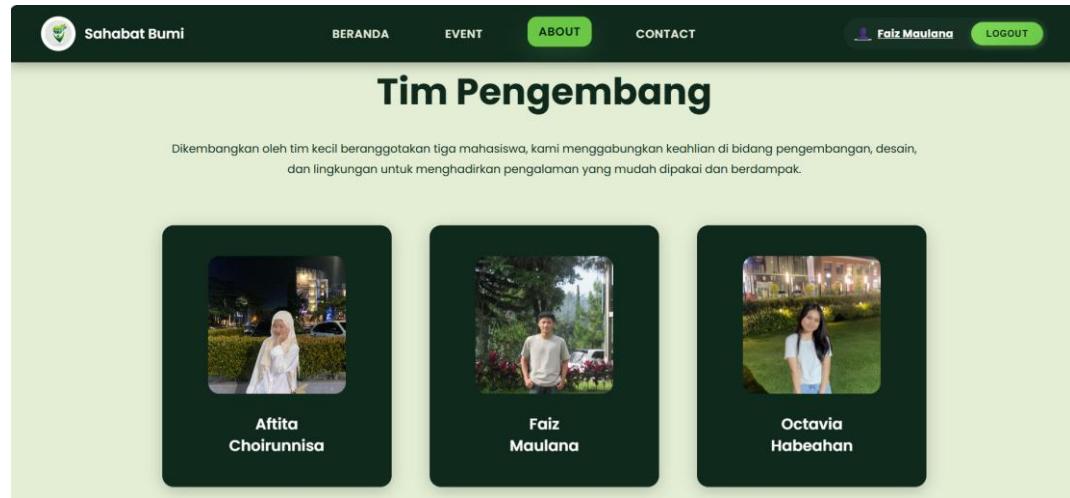
Gambar 5. 1 Menu Beranda

- b. Halaman Event



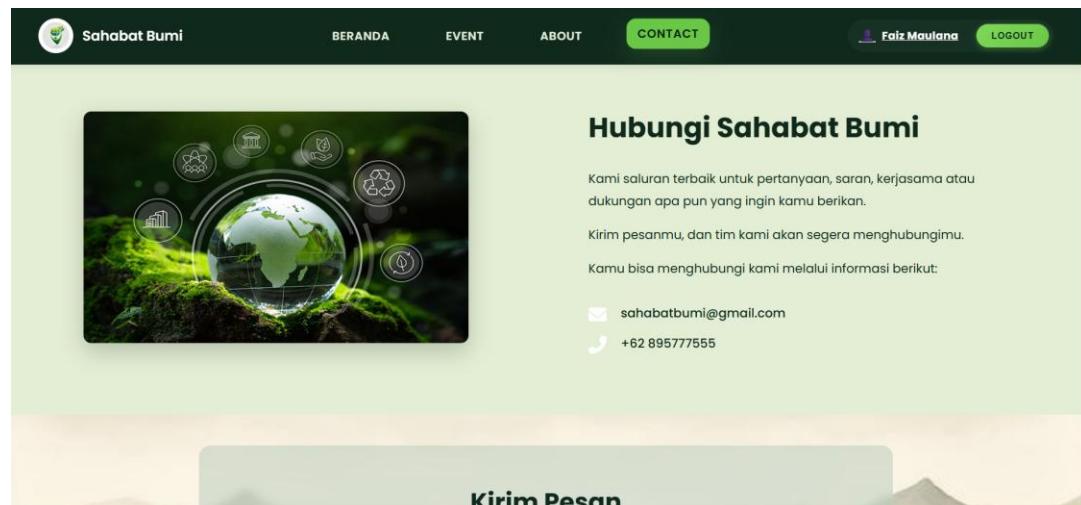
Gambar 5. 2 Halaman Event

c. Halaman About



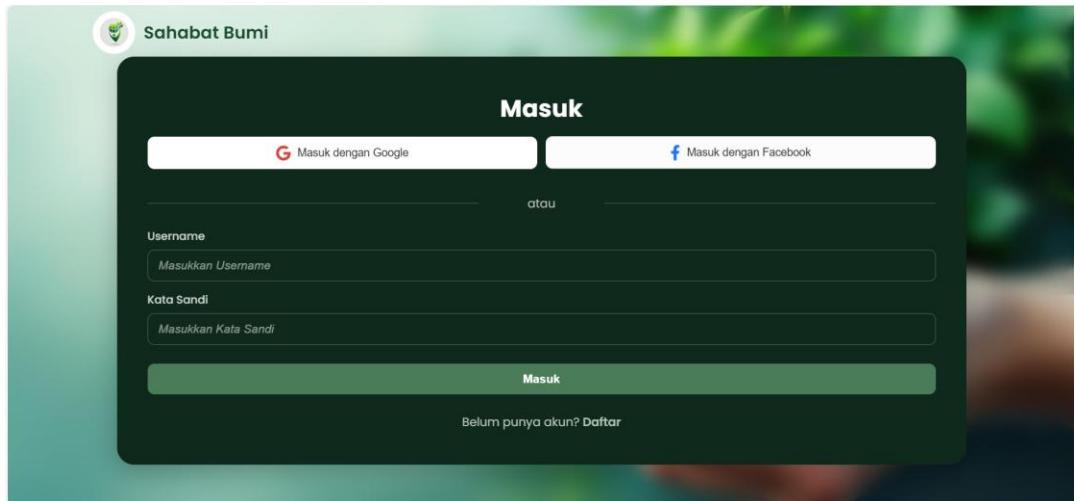
Gambar 5. 3 Halaman About

d. Halaman Contact



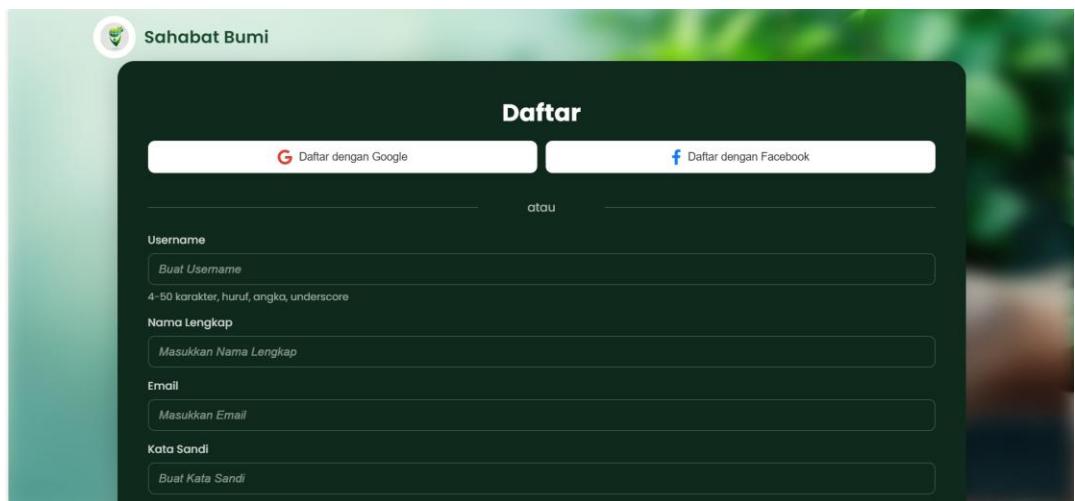
Gambar 5. 4 Halaman Contact

e. Halaman Login



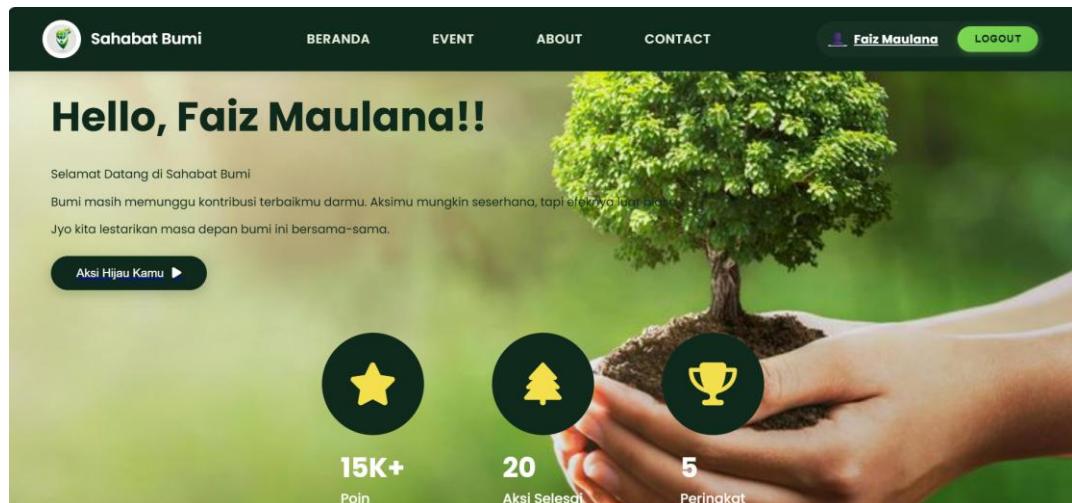
Gambar 5. 5 Halaman Login

f. Halaman Register



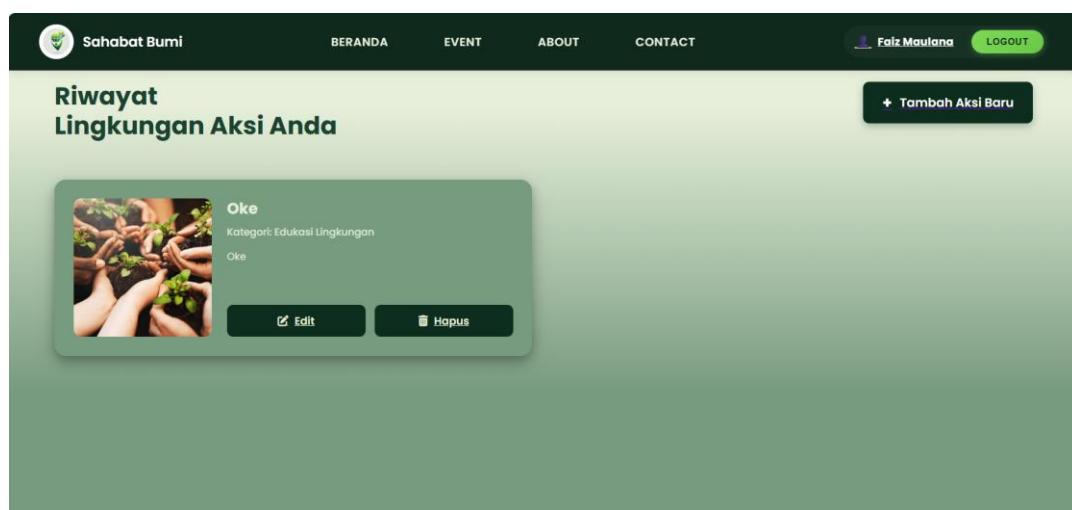
Gambar 5. 6 Halaman Registrasi

g. Halaman Profil



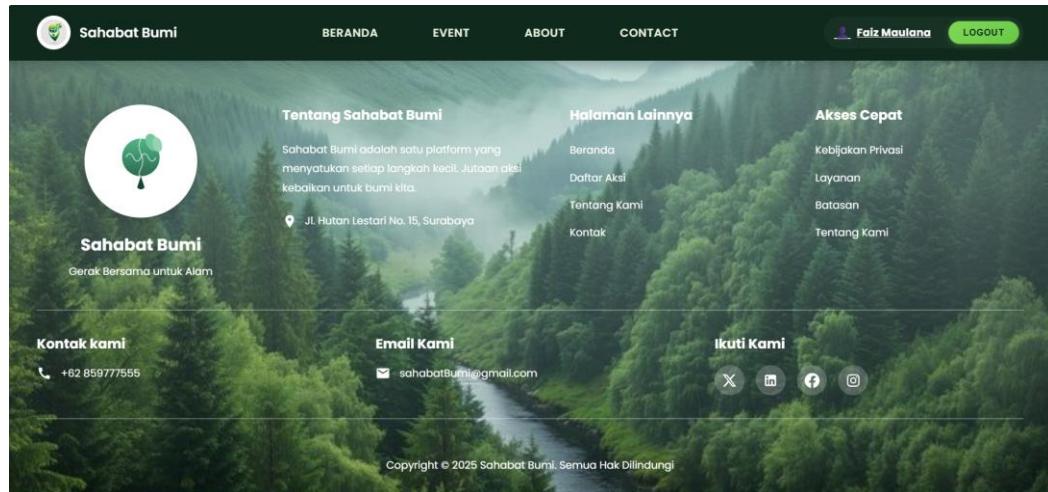
Gambar 5. 7 Halaman Profil

h. Halaman Daftar Riwayat/Laporan



Gambar 5. 8 Halaman Daftar Riwayat

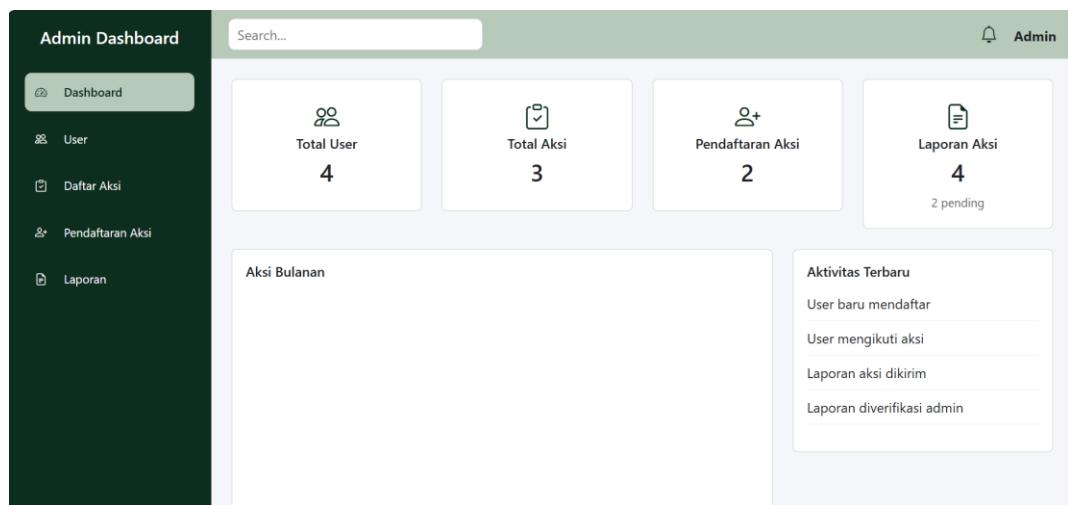
i. Halaman Footer



Gambar 5. 9 Tampilan Footer

2. Tampilan Admin

a. Halaman Dashboard



Gambar 5. 10 Dashboard Admin

b. Halaman Manajemen Users

#	Nama Lengkap	Username	Email	Tanggal Daftar	Aksi
1	Faiz Maulana	faiz10	maul@gmail.com	24-12-2025 13:07	<button>Lihat</button>
2	Messi	messi00	messi@gmail.com	24-12-2025 11:43	<button>Lihat</button>
3	Octavia Habeahan	octa22	octa@gmail.com	23-12-2025 15:49	<button>Lihat</button>
4	Aftita Choirunnisa	aftitafaz	aftita@gmail.com	23-12-2025 10:43	<button>Lihat</button>

Gambar 5. 11 Manajemen User

c. Halaman Daftar Event

#	Gambar	Nama Aksi	Kategori	Tanggal	Lokasi	Aksi
1		Gerakan Pilah Sampah: Dari Rumah untuk Bumi	Aksi Lingkungan – Edukasi & Pengelolaan Sampah	25-12-2025 s/d 25-12-2025	Ketintang, Surabaya	<button>Edit</button> <button>Delete</button>
2		Aksi Bersih Pantai: Laut Sehat Tanpa Sampah	Aksi Lingkungan – Kebersihan & Konservasi Laut	26-12-2025 s/d 26-12-2025	Pantai Kenjeran, Surabaya	<button>Edit</button> <button>Delete</button>
3		Bersih-Bersih Hutan Kota: Tanpa Sampah, Lebih Hijau	Aksi Lingkungan – Kebersihan & Konservasi	24-12-2025 s/d 24-12-2025	Surabaya	<button>Edit</button> <button>Delete</button>

Gambar 5. 12 Manajemen Event

d. Halaman Pendaftaran Event

Data Pendaftaran Aksi

Menampilkan daftar relawan yang telah mendaftar pada setiap aksi

#	Nama Aksi	Nama Relawan	Username	Jenis Kelamin	Umur	Asal Instansi	Tanggal Daftar
1	Bersih-Bersih Hutan Kota: Tanpa Sampah, Lebih Hijau	Faiz Maulana	faiz10	Laki-laki	20	Universitas Negeri Surabaya	24-12-2025 13:09
2	Aksi Bersih Pantai: Laut Sehat Tanpa Sampah	Octavia Habeahan	octa22	Perempuan	20	Universitas Negeri Surabaya	23-12-2025 20:34

Gambar 5. 13 Manajemen Pendaftar Event

e. Halaman Daftar Laporan User

Filter Status

Semua Status

#	User	Judul Laporan	Kategori	Deskripsi	Gambar	Status	Tanggal Upload	Aksi
1	Octavia Habeahan @octa22	Kerja Bakti	Penanaman Pohon	oke...		Rejected	23-12-2025 23:54	@ Lihat Rejected
2	Octavia Habeahan @octa22	Kerja Bakti	Daur Ulang	Oke...		Pending	23-12-2025 15:50	@ Lihat Approved Rejected
3	Aftita Choirunnisa @aftitafaz	Oke	Daur Ulang	Iya...		Approved	23-12-2025 14:51	@ Lihat Approved
4	Aftita Choirunnisa @aftitafaz	Iya	Daur Ulang	b...		Pending	23-12-2025 12:59	@ Lihat Pending

Gambar 5. 14 Daftar Laporan User

BAB VI PENUTUP

6.1 Kesimpulan

Aplikasi "Sahabat Bumi" telah berhasil diwujudkan sebagai platform web fungsional yang menghubungkan penyelenggara acara lingkungan dan relawan dengan alur yang terstruktur—mulai dari pembuatan dan publikasi aksi oleh admin, pendaftaran relawan, pelaksanaan aksi, hingga unggah dan verifikasi laporan berisi bukti kegiatan. Secara teknis, sistem dibangun dengan PHP native dan MySQL (MySQLi), frontend HTML/CSS/JavaScript vanilla, serta arsitektur folder yang rapi (pemisahan user, admin, css, js, dan uploads), sehingga mudah di-deploy pada lingkungan lokal seperti XAMPP dan juga mudah dimodifikasi oleh tim pengembang.

Implementasi prepared statements, validasi input dasar, pembatasan ekstensi/ukuran file, serta penggunaan session untuk otentikasi memberikan pondasi keamanan yang memadai untuk penggunaan skala kecil hingga menengah, sementara mekanisme AJAX pada daftar-aksi.php meningkatkan pengalaman pengguna dengan operasi CRUD tanpa reload penuh. Meski demikian ada beberapa batasan yang masih perlu diperbaiki untuk kesiapan produksi skala lebih besar, antara lain penerapan hashing password modern (password_hash()), proteksi CSRF pada form, pemeriksaan malware pada file upload, penggunaan HTTPS, dan strategi penyimpanan file yang lebih tahan banting (mis. object storage). Secara fungsional dan pendidikan, aplikasi ini sudah memenuhi tujuan proyek—mempermudah koordinasi aksi lingkungan, mendokumentasikan bukti kegiatan, dan memberi dasar bagi pengembangan fitur lanjutan seperti gamifikasi dan notifikasi—with arsitektur yang cukup sederhana untuk pembelajaran namun juga memadai sebagai prototipe yang dapat ditingkatkan.

6.2 Kelebihan dan Kekurangan Aplikasi

6.2.1 Kelebihan

1. Sederhana dan ringan: Dibangun dengan PHP native sehingga mudah dipahami dan dikembangkan oleh tim akademik.
2. Fitur fungsional lengkap untuk level dasar: Registrasi/login, CRUD event, pendaftaran, upload laporan, verifikasi admin, dan penyimpanan file.
3. Keamanan dasar diterapkan: Penggunaan prepared statements, validasi file (ekstensi & ukuran), dan sanitasi tampilan (htmlspecialchars) di beberapa titik.

4. Arsitektur modular: File dan folder terpisah (user, admin, css, js, uploads) memudahkan pemeliharaan.
5. UX interaktif sederhana: AJAX CRUD pada daftar-aksi.php + script_daftar-aksi.js membuat interaksi lebih cepat tanpa reload penuh.

6.2.2 Kekurangan

1. Keamanan yang belum lengkap: Belum ada CSRF token pada form, password hashing/penanganan password belum terdokumentasi konsisten (contoh sampel MD5 di schema lama harus diganti ke password_hash()), dan belum ada proteksi rate-limiting.
2. Manajemen file dasar: Upload belum punya scanning malware/virus, kontrol versi/retensi file, atau storage terpisah (mis. object storage).
3. Belum ada enkripsi transport (HTTPS) default: Pada deployment publik harus pasang SSL/TLS.
4. Skalabilitas terbatas: Dirancang untuk deployment sederhana (XAMPP / single server); tidak siap skala otomatis atau beban tinggi tanpa refactor.
5. Testing & observability minim: Belum ada suite automated tests, logging terpusat, atau monitoring performa.
6. Fitur tambahan non-implementasi: Gamifikasi (points/leaderboard) dan notifikasi email masih dasar/opsional dan butuh penguatan.

DAFTAR PUSTAKA

Kementerian Lingkungan Hidup dan Kehutanan. (2024). Sistem Informasi Pengelolaan Sampah Nasional (SIPSN). Diakses dari <https://sipsn.kemenlh.go.id>

Ratnasari, S. (2022). "Partisipasi Komunitas Masyarakat Lokal pada Program Lingkungan Hidup di Indonesia". Sosioinforma, 8(1). Diakses dari <https://ejournal.poltekkesos.ac.id>

Environment Indonesia. (n.d.). "12 Tantangan Umum dalam Penerapan Sistem Manajemen Lingkungan ISO 14001:2015". Diakses dari <https://environment-indonesia.com>

Demmanggasa, Y. (2024). "Pengaruh Partisipasi Masyarakat dalam Pengelolaan Lingkungan". Jurnal Cahaya Mandalika, 5(3). Diakses dari <https://ojs.cahayamandalika.com>

PTALI. (2025). "Realita Volume Sampah Harian di Tempat Pembuangan Akhir (TPA) di Indonesia". Diakses dari <https://ptali.org>

Jasa ISO. (n.d.). "Kendala dalam Manajemen Lingkungan ISO 14001". Diakses dari <https://jasaiso.id>

LAMPIRAN

Lampiran 1 Daftar Anggota dan Jobdesk

NIM	Nama	Jobdesk
24091397083	Aftita Choirunnisa	<ul style="list-style-type: none">- Design UI Menu Admin- Design UI Halaman Event- Design UI Halaman About- Design UI Profil User- Design UI Footer- Membuat Use Case Diagram- Membuat Class Diagram- Membuat Activity Diagram User- Dokumentasi Laporan
24091397094	Octavia Habeahan	<ul style="list-style-type: none">- Design UI Halaman Login dan SignUp- Design UI Contact- Design UI Halaman Beranda- Membuat Activity Diagram Admin- Editing Video- Dokumentasi Laporan
24091397103	Faiz Maulana	<ul style="list-style-type: none">- Perancangan & Implementasi Database- Membuat Diagram ERD dan Database Schema- JavaScript Client-Server Integration- Koneksi & Helper Database- Autentikasi, Otorisasi & Session- CRUD & File Upload (Event & Laporan)- Validasi, Keamanan & Error Handling- Dokumentasi Laporan