

Pengembangan Aplikasi Mobile Komunitas Hiking Terintegrasi Strava Dengan Fitur Manajemen Event Dan Story

I Putu Krisna Putra Yasa^{1a)}, I Made Arya Budhi Saputra^{1b)}, Edwar^{2c)}

¹⁾Sistem Komputer, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia

²⁾Manajemen Informatika, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia

e-mail: ^{a)}210010129@stikom-bali.ac.id, ^{b)}aryabudhi@stikom-bali.ac.id, ^{c)}edwar.ridwan@stikom-bali.ac.id

Abstrak

Hiking merupakan aktivitas luar ruang yang umumnya dilakukan secara berkelompok dalam suatu komunitas. Namun, pengelolaan komunitas hiking masih menghadapi kendala dalam perencanaan event, pengelolaan rute, serta dokumentasi aktivitas yang belum terintegrasi dengan data pelacak olahraga. Di sisi lain, aplikasi Strava banyak digunakan untuk merekam aktivitas olahraga, tetapi belum secara khusus mendukung kebutuhan manajemen komunitas hiking. Penelitian ini bertujuan untuk mengembangkan aplikasi mobile komunitas hiking yang terintegrasi dengan Strava guna mendukung pengelolaan grup, pembuatan dan partisipasi event, pemanfaatan rute pendakian, serta dokumentasi kegiatan melalui fitur story. Metode pengembangan yang digunakan adalah Agile Development dengan kerangka kerja Scrum agar sistem dapat dikembangkan secara iteratif. Aplikasi dibangun menggunakan React Native sebagai antarmuka mobile, ElysiaJS sebagai backend, serta integrasi Strava API melalui mekanisme OAuth 2.0. Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan fungsi berjalan sesuai kebutuhan. Hasil penelitian menunjukkan bahwa aplikasi mampu mendukung pengelolaan komunitas hiking secara lebih terstruktur dan terintegrasi.

Kata kunci: Komunitas Hiking, Strava API, OAuth 2.0, React Native, Manajemen Event.

1. Pendahuluan

Perkembangan teknologi informasi saat ini telah mengubah paradigma masyarakat dalam menjalani gaya hidup sehat. Integrasi teknologi Global Positioning System (GPS) ke dalam perangkat seluler telah menjadi standar modern dalam mendukung aktivitas olahraga, memungkinkan pengguna untuk melacak, menganalisis, dan membagikan data aktivitas fisik mereka secara akurat [1]. Salah satu platform yang mendominasi ekosistem ini adalah Strava. Popularitas Strava tidak terlepas dari kualitas pengalaman pengguna (*User Experience*) yang ditawarkannya, di mana aplikasi ini mampu menyajikan antarmuka yang intuitif untuk aktivitas lari dan bersepeda, sehingga memudahkan pengguna dalam memantau kinerja olahraga mereka [2].

Peran aplikasi pelacak aktivitas seperti Strava memiliki kontribusi yang signifikan dalam mendukung ekosistem olahraga modern. Melalui fitur pencatatan jarak, waktu, dan performa aktivitas, Strava membantu pengguna dalam memantau serta mengevaluasi capaian latihan harian secara objektif [3]. Selain mendukung evaluasi performa, penggunaan aplikasi ini juga memberikan dampak positif pada aspek fisiologis dan psikologis pengguna, terutama ketika digunakan secara rutin [4]. Fitur sosial dan kompetitif yang disediakan Strava, seperti berbagi aktivitas dan perbandingan capaian, terbukti mampu meningkatkan motivasi pengguna untuk berolahraga secara konsisten. Pada konteks komunitas lari, fitur tersebut berkontribusi terhadap peningkatan motivasi dan kesehatan anggota komunitas secara keseluruhan [5].

Meskipun teknologi pelacak aktivitas terus berkembang, penerapannya secara khusus untuk mendukung manajemen komunitas pendaki gunung masih memiliki keterbatasan. Pengembangan aplikasi pendakian di Indonesia umumnya berfokus pada penyediaan informasi jalur dan kebutuhan logistik, sebagaimana ditunjukkan pada penelitian perancangan aplikasi pendakian berbasis Android dengan metode prototype [6]. Namun, aspek sosial dan pengelolaan komunitas belum menjadi perhatian utama. Padahal, interaksi dan koordinasi antaranggota merupakan elemen penting dalam aktivitas pendakian berkelompok. Studi mengenai desain aplikasi *mobile* untuk komunitas pendaki menunjukkan bahwa keterbatasan fitur sosial dapat menghambat terbentuknya kolaborasi dan keterlibatan pengguna secara optimal [7].

Kondisi tersebut menjadi latar belakang penelitian ini, di mana perencanaan rute dan dokumentasi kegiatan komunitas *hiking* masih menghadapi keterbatasan akibat terpisahnya data aktivitas Strava dengan sistem manajemen komunitas. Oleh karena itu, penelitian ini mengusulkan pengembangan aplikasi *mobile* komunitas *hiking* yang terintegrasi dengan Strava untuk mendukung pengelolaan event dan interaksi sosial secara lebih terstruktur.

2. Metode Penelitian

Penelitian ini menerapkan metode pengembangan perangkat lunak Agile Development dengan kerangka kerja Scrum. Metode ini dipilih karena karakteristiknya yang iteratif dan inkremental, yang sangat relevan untuk proyek dengan kebutuhan dinamis seperti integrasi API pihak ketiga. Penerapan Agile Scrum terbukti efektif dalam meningkatkan fleksibilitas pengembangan sistem informasi, memungkinkan tim untuk mengakomodasi perubahan kebutuhan pengguna di tengah siklus pengembangan tanpa mengganggu stabilitas proyek secara keseluruhan [8].



Gambar 1. Metode Agile

2.1 Analisis Sistem

Analisis sistem dilakukan untuk mengidentifikasi kebutuhan pengembangan aplikasi komunitas *hiking* terintegrasi Strava melalui studi literatur terkait aplikasi olahraga dan komunitas digital. Hasil kajian menunjukkan bahwa fitur komunitas dan *event* dapat meningkatkan partisipasi pengguna, sementara integrasi Strava mendukung pemanfaatan data aktivitas yang sudah tersedia sehingga mengurangi input manual [2], [5]. Berdasarkan analisis tersebut, sistem dirancang untuk mendukung autentikasi pengguna, sinkronisasi data Strava, pengelolaan grup, *event hiking*, dan penyajian *story* kegiatan.

2.2 Desain Sistem

Desain sistem disusun berdasarkan hasil analisis kebutuhan untuk memberikan gambaran pembentukan aplikasi secara menyeluruh. Tahap ini mencakup pemodelan alur proses, aliran data, dan struktur basis data menggunakan UML Activity Diagram, Data Flow Diagram (DFD), dan Entity Relationship Diagram (ERD) sebagai acuan pengembangan dan pembahasan implementasi pada tahap berikutnya.

2.3 Implementasi Sistem

Tahap implementasi merupakan proses penerjemahan desain ke dalam kode program (*coding*) yang dilakukan dalam beberapa iterasi:

- Iterasi 1 (Fondasi): Membangun struktur *backend* ElysiaJS, konfigurasi *database* PostgreSQL, dan sistem autentikasi OAuth 2.0.
- Iterasi 2 (Fitur Inti): Mengembangkan antarmuka React Native untuk fitur *Unified Route List* dan sinkronisasi data *polyline*.
- Iterasi 3 (Fitur Pelengkap): Mengimplementasikan logika *Auto-Story* dan manajemen anggota komunitas.

2.4 Pengujian Sistem

Pengujian kualitas perangkat lunak dilakukan menggunakan metode *Black Box Testing*. Fokus pengujian adalah pada validasi fungsionalitas *input-output* aplikasi untuk memastikan kesesuaian dengan kebutuhan pengguna. Teknik pengujian ini efektif untuk mendeteksi kesalahan antarmuka dan fungsi *database* tanpa perlu membedah struktur kode internal [9].

3. Hasil dan Pembahasan

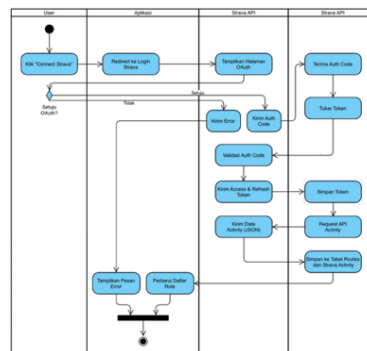
Implementasi aplikasi komunitas *hiking* terintegrasi Strava dilakukan berdasarkan perancangan pada tahap sebelumnya. Sistem direalisasikan dalam bentuk aplikasi *mobile*, *backend server*, serta integrasi layanan pihak ketiga melalui API. Proses implementasi dilakukan secara bertahap untuk memastikan setiap fitur utama dapat berjalan dengan baik dan memenuhi kebutuhan pengguna.

3.1 Desain Sistem

Implementasi aplikasi komunitas *hiking* terintegrasi Strava dilakukan berdasarkan perancangan pada tahap sebelumnya. Sistem direalisasikan dalam bentuk aplikasi *mobile*, *backend server*, serta integrasi layanan pihak ketiga melalui API. Proses implementasi dilakukan secara bertahap untuk memastikan setiap fitur utama dapat berjalan dengan baik dan memenuhi kebutuhan pengguna.

3.1.1 Activity Diagram

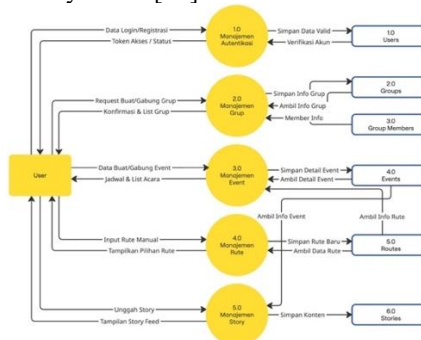
Alur pada Activity Diagram dimulai ketika pengguna menginisiasi proses integrasi Strava melalui aplikasi mobile. Aplikasi kemudian mengarahkan pengguna ke halaman otorisasi Strava untuk memberikan izin akses menggunakan mekanisme OAuth. Setelah pengguna menyetujui permintaan akses, Strava mengirimkan *authorization code* ke *backend* sistem. *Backend* selanjutnya menukarkan kode tersebut menjadi *access token* dan *refresh token* yang valid. Token ini digunakan oleh *backend* untuk mengambil data aktivitas atau rute dari Strava API. Data yang diperoleh kemudian diproses dan disimpan agar dapat digunakan oleh aplikasi mobile. Alur ini menggambarkan koordinasi aktivitas antar sistem secara berurutan dan terkontrol [10].



Gambar 2. Activity Diagram Integrasi Sistem antara Aplikasi *Mobile*, *Backend*, dan Strava API

3.1.2 Data Flow Diagram

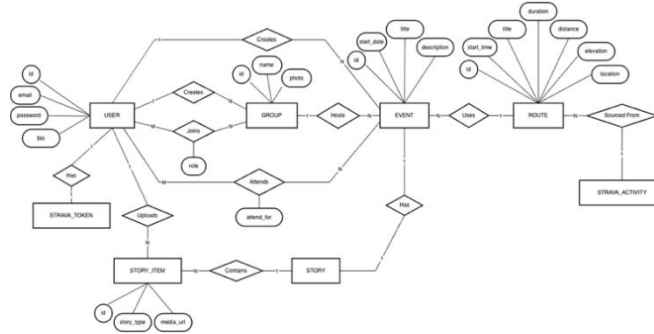
Alur data pada DFD dimulai dari pengguna yang melakukan interaksi melalui aplikasi mobile, seperti mengelola grup, membuat *event*, memilih rute, atau mengunggah *story*. Data dari aplikasi mobile dikirim ke *backend* untuk diproses sesuai fungsi masing-masing. *Backend* bertugas memvalidasi, mengolah, serta menyimpan data ke dalam basis data. Selain itu, *backend* juga berperan sebagai penghubung dengan layanan pihak ketiga, seperti Strava API, untuk mengambil data aktivitas eksternal. Seluruh data yang telah diproses kemudian dikembalikan ke aplikasi mobile dalam bentuk informasi yang dapat ditampilkan kepada pengguna. DFD membantu memperjelas alur pergerakan data antar entitas, proses, dan penyimpanan data secara menyeluruh [11].



Gambar 3. Data Flow Diagram (DFD) Level 1 Sistem Aplikasi Komunitas *Hiking*

3.1.3 Entity Relationship Diagram

Alur hubungan data pada ERD menunjukkan keterkaitan antar entitas utama dalam sistem. Entitas pengguna berperan sebagai pusat relasi yang terhubung dengan grup komunitas, *event*, rute, dan *story*. Setiap grup dapat memiliki banyak *event*, dan setiap *event* dapat dikaitkan dengan satu atau lebih rute. *Story* berelasi dengan pengguna dan *event* sebagai bentuk dokumentasi aktivitas. Relasi-relasi ini menggambarkan bagaimana data saling terhubung dan digunakan dalam sistem. Dengan alur relasi yang jelas, ERD membantu memastikan struktur basis data mampu mendukung kebutuhan fitur aplikasi secara konsisten dan terintegrasi [12].



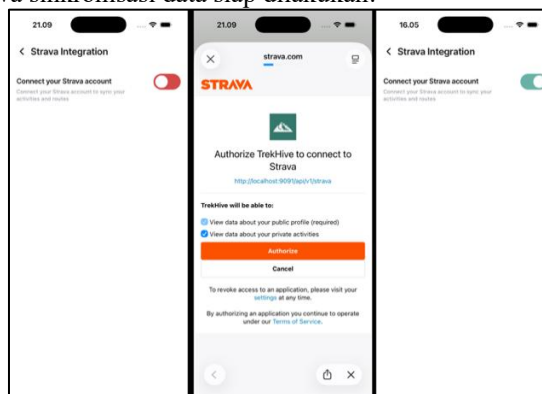
Gambar 4. Entity Relationship Diagram

3.2 Implementasi Sistem

Tahap implementasi sistem merupakan fase realisasi rancangan perangkat lunak yang telah didefinisikan pada tahap desain sebelumnya. Pada tahap ini, spesifikasi kebutuhan dan model logika (UML, DFD, ERD) ditransformasikan menjadi kode program yang dapat dieksekusi. Implementasi mencakup penyiapan lingkungan pengembangan, pembangunan struktur basis data, penulisan logika *backend* menggunakan *framework* ElysiaJS, serta penyusunan antarmuka *frontend* berbasis React Native.

a. Implementasi Fitur Integrasi Strava

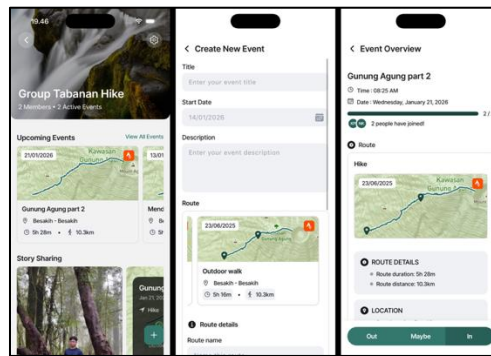
Fitur integrasi Strava tersedia pada halaman Strava Integration di menu pengaturan akun. Pengguna dapat menghubungkan akun Strava dengan menekan tombol "Connect your Strava account", yang akan mengarahkan ke halaman otorisasi Strava menggunakan mekanisme OAuth 2.0. Setelah otorisasi berhasil, sistem menyimpan token akses dan menampilkan status Connected sebagai indikator bahwa sinkronisasi data siap dilakukan.



Gambar 5. Tampilan Implementasi Fitur Integrasi Strava pada Aplikasi *Mobile*

b. Implementasi Fitur Manajemen Rute dan *event*

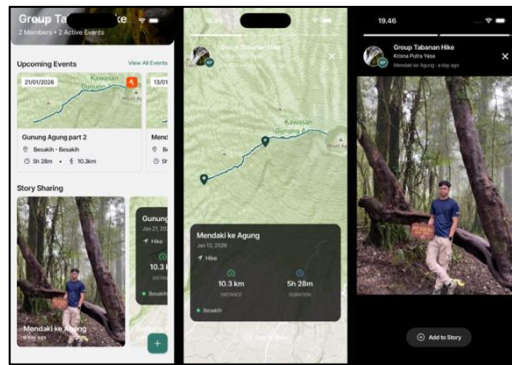
Tampilan Group Overview menampilkan informasi grup, jumlah anggota, serta daftar *event* yang akan datang. Setiap *event* terhubung dengan rute tertentu yang divisualisasikan dalam bentuk peta ringkas. Pada halaman *Create New Event*, pengguna dapat membuat *event* dengan menentukan judul, tanggal, deskripsi, serta memilih rute yang tersedia, baik rute manual maupun hasil sinkronisasi dari Strava. Halaman *Event Overview* menampilkan detail *event*, informasi rute, lokasi, serta opsi partisipasi (*In*, *Maybe*, atau *Out*) bagi anggota grup, sehingga memudahkan koordinasi kegiatan pendakian.



Gambar 6. Tampilan Implementasi Fitur Integrasi Strava pada Aplikasi *Mobile*

c. Implementasi Fitur *Auto-Story* dan *Upload Story*

Fitur *story* diintegrasikan pada halaman grup untuk mendukung dokumentasi aktivitas pendakian. Pengguna dapat mengunggah *story* berupa foto atau video yang berkaitan dengan *event* atau aktivitas yang diikuti. *Story* yang diunggah akan ditampilkan pada *story feed* grup, sehingga anggota lain dapat melihat dan mengikuti aktivitas yang sedang atau telah berlangsung. Fitur ini berfungsi sebagai sarana berbagi pengalaman dan meningkatkan interaksi antaranggota komunitas.



Gambar 7. Tampilan Implementasi Fitur *Auto-Story* dan *Upload Story* pada Grup

3.1 Pengujian Sistem

Tabel 1. *Black box Testing*

No	Fitur yang Diuji	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
1	Integrasi Strava	Pengguna menekan tombol <i>Connect Strava</i>	Persetujuan OAuth Strava	Status akun berubah menjadi <i>Connected</i>	Berhasil
2	Sinkronisasi Aktivitas Strava	Sistem mengambil data aktivitas	Akun Strava valid	Data aktivitas tersimpan dan ditampilkan	Berhasil
3	Manajemen Grup	Pengguna membuat grup komunitas	Nama dan deskripsi grup	Grup berhasil dibuat	Berhasil
4	Manajemen Rute	Pengguna memilih rute pendakian	Rute manual / Strava	Rute tersimpan dan dapat digunakan	Berhasil
5	Pembuatan <i>Event</i>	Pengguna membuat <i>event hiking</i>	Rute, tanggal, dan waktu	<i>Event</i> berhasil dibuat dan tampil di grup	Berhasil
6	Partisipasi <i>Event</i>	Anggota menekan tombol <i>In</i>	Aksi partisipasi	Status kehadiran tercatat	Berhasil
7	<i>Upload Story</i>	Pengguna mengunggah <i>story</i>	Foto / video	<i>Story</i> tampil pada feed grup	Berhasil

8	<i>Auto-Story</i>	Sistem menghasilkan Data <i>event</i> selesai <i>Story</i> otomatis tampil di	Berhasil
		<i>story</i> otomatis	grup

Berdasarkan hasil pengujian *Black Box* yang disajikan pada Tabel 1, seluruh fungsi utama pada aplikasi komunitas *hiking* terintegrasi Strava dapat berjalan sesuai dengan kebutuhan sistem. Setiap skenario pengujian menunjukkan bahwa input yang diberikan pengguna menghasilkan keluaran yang sesuai dengan spesifikasi yang telah ditentukan. Fitur autentikasi pengguna, integrasi Strava, manajemen grup, pengelolaan *event* dan rute, serta fitur *story* berhasil diuji tanpa ditemukan kesalahan fungsional yang signifikan. Hasil pengujian ini menunjukkan bahwa sistem mampu memproses setiap permintaan pengguna dengan benar dan menampilkan respons yang sesuai.

4. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem, dapat disimpulkan bahwa aplikasi komunitas *hiking* terintegrasi Strava berhasil dikembangkan sesuai dengan tujuan penelitian. Sistem mampu mengintegrasikan aplikasi *mobile*, *backend server*, dan layanan Strava melalui mekanisme OAuth untuk mendukung sinkronisasi data aktivitas pengguna. Fitur utama seperti manajemen grup, rute, *event*, serta *story* telah diimplementasikan dan diuji menggunakan metode *Black Box Testing* dengan hasil yang menunjukkan seluruh fungsi berjalan sesuai dengan kebutuhan pengguna. Dengan demikian, aplikasi ini dapat digunakan sebagai sarana pendukung aktivitas komunitas *hiking* yang terstruktur, interaktif, dan terintegrasi dengan platform pihak ketiga.

Daftar Pustaka

- [1] M. Aiman et al., "Integrasi GPS Dalam Aplikasi Olahraga Sebagai Cara Modern Untuk Gaya Hidup Aktif dan Sehat," 2024, doi: 10.6732/jayabama.v2i2.3514.
 - [2] H. Fauzan Azhar, I. Aknuranda, and A. Pinandito, "Analisis Pengalaman Pengguna Aplikasi 'Strava: Run, Bike, Ride,'" 2025. [Online]. Available: <http://j-ptiik.ub.ac.id>
 - [3] A. Nur, M. Jidan, J. Cesare Zidane, and K. K. Imar Aulia, "Analisis Peranan Aplikasi Strava Terhadap Kegiatan Olahraga Lari Pagi," 2024, doi: 10.8734/Kohesi.v1i2.365.
 - [4] E. T. Simanungkalit et al., "Olahraga Strava Terhadap Kebugaran Fisik Seseorang," vol. 6, no. 1, 2025.
 - [5] B. Aryatama, B. Shodiq, M. I. Hufadz, A. Ramadhani, B. S. Tarigan, and A. A. Nabillah, "Influence of The Strava Application on The Health And Motivation of The Running Community in Metro Cities," *Jurnal Performa Olahraga*, vol. 9, no. 2, pp. 66–71, Dec. 2024, doi: 10.24036/jpo673019.
 - [6] A. A. Fauzi and A. Adisuputra, "Perancangan Aplikasi Informasi Pendakian Gunung di Indonesia Berbasis Android Dengan Menggunakan Metode Prototype," *INTI Nusa Mandiri*, vol. 19, no. 1, pp. 126–137, Aug. 2024, doi: 10.33480/inti.v19i1.5601.
 - [7] N. Huda, M. Razif, M. Razeef, and A. Razak, "Social Mobile Applications Design: Case Study of Social Hikers Community," 2020.
 - [8] N. Etrariadi and E. A'inunisyah, "Pengembangan Website Manajemen Proyek Menggunakan Metode Agile Scrum (Studi Kasus Diskopindag Kota Malang)," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, pp. 55–66, Jan. 2023, doi: 10.25077/TEKNOSI.v9i1.2023.55-66.
 - [9] A. Jailani and M. A. Yaqin, "Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis," *Journal Automation Computer Information System*, vol. 4, no. 2, pp. 60–66, Jul. 2024, doi: 10.47134/jacis.v4i2.78.
 - [10] C. Ayu Binangkit, A. Voutama, and N. Heryana, "Pemanfaatan UML (Unified Modeling Language) Dalam Perencanaan Sistem Pengelolaan Sewa Alat Musik Berbasis Website," 2023.
 - [11] A. Fatah and R. Meina Widiastuti, "SEMINAR NASIONAL CORISINDO Penggunaan Data Flow Diagram (DFD) untuk Perancangan Sistem Informasi Kebutuhan Material di PT. XYZ," 2024.
 - [12] M. Rahman Akbar, A. Zurfadly, and M. Apriani, "Perancangan Database Elite Hotel Tembilihan Menggunakan ERD (Entity Relationship Diagram)," *Jurnal Sistem Informasi (TEKNOFILE)*, vol. 3, pp. 105–117, 2025.
-