

**Rancang Bangun Sistem Informasi *Lost and Found*
Menggunakan Metode *Feature Driven Development* (FDD)
Berbasis *Location Based Service* (LBS)**

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh
Gelar Sarjana Program Strata 1 (S1)
dalam Ilmu Teknologi Informasi



Diajukan oleh:

Daffa Raihan Dimyati

NIM : 2108096087

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG**

2025

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Daffa Raihan Dimiyati

NIM : 2108096087

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

**Rancang Bangun Sistem Informasi *Lost and Found*
Menggunakan Metode *Feature Driven Development*
(FDD) Berbasis *Location Based Service* (LBS)**

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 4 Juli 2025

Pembuat Pernyataan



Daffa Raihan Dimiyati

NIM: 2108096087

NOTA PEMBIMBING

Semarang, /06/2025

Yth. Ketua Program Studi Teknologi Informasi Fakultas
Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan, bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan :

Judul : Rancang Bangun Sistem Informasi Lost And Found Menggunakan Metode Feature Driven Development (FDD) Berbasis Location Based Service (LBS)

Penulis : Daffa Raihan Dimiyati

NIM : 2108096087

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. wr. wb.

Pembimbing I,



Hery Mustofa, M.Kom
NIP. 198703172019031007

NOTA PEMBIMBING

Semarang, /06/2025

Yth. Ketua Program Studi Teknologi Informasi Fakultas
Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan, bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan :

Judul : Rancang Bangun Sistem Informasi Lost And Found Menggunakan Metode Feature Driven Development (FDD) Berbasis Location Based Service (LBS)

Penulis : Daffa Raihan Dimiyati

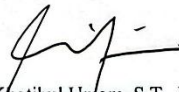
NIM : 2108096087

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. wr. wb.

Pembimbing II,



Dr. Khotibul Ulam, S.T., M.Kom
NIP. 197908272011011007



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof Dr. Hamka kampus III Ngaliyan
Semarang Telp. 7601295 Fax.7615387

LEMBAR PENGESAHAN

Naskah skripsi berikut ini:

Judul : RANCANG BANGUN SISTEM INFORMASI
LOST AND FOUND MENGGUNAKAN METODE
FEATURE DRIVEN DEVELOPMENT (FDD)
BERBASIS LOCATION BASED SERVICE (LBS)

Penulis : Daffa Raihan Dimiyati

NIM : 2108096087

Jurusan : Teknologi Informasi

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji Fakultas
Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah
satu syarat memperoleh gelar sarjana dalam Ilmu Teknologi Informasi.

Semarang, 2 Juli 2025

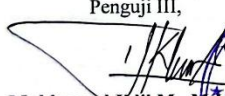
DEWAN PENGUJI

Penguji I,


Dr. WENTY DWILYUNIARTI, S.Pd.,
M.Kom

NIP. 197706222006042005

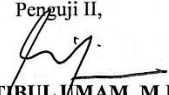
Penguji III,


Mokhammad Ikil M., M.Kom.
NIP. 1988080720190310100

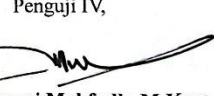
Pembimbing I,


Hery Mustofa, M.Kom.
NIP. 198703172019031007

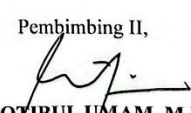
Penguji II,


Dr. KHOTIBUL UMAM, M.Kom
NIP. 197908272011011007

Penguji IV,


Adzhar Arwani Mahfudh, M.Kom.
NIP. 199107032019031006

Pembimbing II,


Dr. KHOTIBUL UMAM, M.Kom
NIP. 197908272011011007



LEMBAR PERSEMBAHAN

Dengan rasa syukur yang mendalam, dengan telah diselesaikannya Skripsi ini penulis mempersembahkan kepada:

1. Keluarga besar Penulis yang senantiasa mendukung dan selalu mendoakan penulis.
2. Segenap civitas akademik UIN Walisongo Semarang, staff, pengajar, karyawan, dan seluruh mahasiswa semoga selalu dalam keadaan sehat dan tetap semangat dalam beraktivitas mengisi hari-harinya di kampus tercinta UIN Walisongo semarang.
3. Teman-teman Penulis khususnya mahasiswa Teknologi Informasi angkatan 2021, yang selalu memberikan semangat dalam suka maupun duka selama pembuatan skripsi sehingga skripsi dapat terselesaikan dengan baik.

Semoga karya ini dapat memberikan manfaat bagi semua pihak serta menjadi langkah awal menuju pengabdian yang lebih luas.

MOTTO

“Work hard and even harder! Do not carry out the evil just because it is small, and do not hesitate to do the good because it is small.”

(Liu Bei – Dynasty Warrior)

RANCANG BANGUN SISTEM INFORMASI *LOST AND FOUND* MENGGUNAKAN METODE *FEATURE DRIVEN DEVELOPMENT (FDD)* BERBASIS *LOCATION BASED SERVICE (LBS)*

Oleh:

Daffa Raihan Dimiyati
NIM. 2108096087

ABSTRAK

Sistem informasi *Lost & Found* dikembangkan untuk memudahkan mahasiswa dan civitas akademik UIN Walisongo dalam melaporkan serta mencari barang hilang secara lebih akurat, efisien, dan terpusat. Sistem ini hadir sebagai solusi atas proses pelaporan manual yang sebelumnya digunakan, yang dirasa kurang efektif dalam menjangkau seluruh area universitas. Dengan platform digital ini, pengguna dapat mengakses layanan pelaporan dan pencarian barang kapan saja tanpa perlu interaksi langsung. Metode pengembangan yang digunakan adalah *Feature Driven Development (FDD)*, yang terdiri dari lima tahap: *develop an overall model*, *build a feature list*, *plan by feature*, *design by feature*, dan *build by feature*. Untuk mendukung pengelolaan waktu dan progres pengembangan, digunakan *Gantt Chart* sebagai alat bantu penjadwalan. Hasil perbandingan antara jadwal rencana dan realisasi menunjukkan bahwa setiap tahapan pengembangan dapat dikontrol dengan baik dan proyek selesai lebih cepat dari jadwal, yang membuktikan efektivitas *FDD* dalam mengelola proyek secara terstruktur. Sistem ini telah melalui pengujian *black-box* sebanyak 25 skenario, terdiri dari 9 skenario untuk admin dan 16 untuk user, dengan tingkat keberhasilan mencapai 100%, yang berarti semua fungsi berjalan sesuai kebutuhan. Selain itu, pengujian *User Acceptance Test (UAT)* terhadap 45 responden menunjukkan tingkat kepuasan sebesar 91,4%, yang menandakan bahwa sistem ini telah diterima dengan baik dan layak digunakan secara fungsional oleh pengguna.

Kata kunci : Sistem Informasi, *Lost & Found*, pelaporan, pencarian dan barang

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, sehingga Laporan Tugas Akhir Skripsi yang berjudul “Rancang Bangun Sistem Informasi Lost And Found Menggunakan Metode Feature Driven Development (FDD) Berbasis Location Based Service (LBS)”, dapat diselesaikan dengan baik dan tepat waktu.

Penyusunan laporan tugas akhir ini tidak terlepas dari bantuan beberapa pihak, oleh karena itu penulis hendak mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Nizar, M.Ag. selaku Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Prof. Dr. H. Musahadi, M.Ag., selaku Dekan Fakultas Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
3. Bapak Dr. Khotibul Umam, S.T., M.Kom selaku Ketua Program Studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
4. Bapak Hery Mustofa, S.T., M.Kom dan Bapak Dr. Khotibul Umam, S.T., M.Kom selaku Dosen Pembimbing Skripsi.
5. Staff, karyawan dan dosen di lingkungan Universitas Islam Negeri Walisongo Semarang yang telah memberikan ilmu pengetahuan yang tidak ternilai selama menempuh pendidikan.

6. Orang tua penulis, Bapak Mochammad Mashar dan Ibu Tri Yulianingsih, serta Kakak Miftah Faris Dimiyati, dan Maftuh Dimiyati yang selalu mendukung penulis
7. Kepada diri sendiri yang selalu percaya dalam keputusan dan usaha yang dilakukan, melakukan yang terbaik di segala hal yang dilakukan tanpa terkecuali
8. Rekan Seperjuangan Skripsi, Shauqi Nazmi Fauzan, Habiburohman, Ardan Fajar Setiady, Nur Cahyo Adi Saputra, Desmi Salsabila, Nimas Lian Kusuma, Muhammad Satria Akmal, Eka Juni Rahayu, Mutiara Ilma Daniati.
9. Rekan yang dikira membantu melepas penat selama mengerjakan skripsi, Ralief Langga Rivansyah, Shidiq Fadhillah Wahyu, Cahya Nugroho, Aswilliandini, Nicky Bryant, Adam Mareno Barri, Raihan Ashil Zaki, Gilang.
10. Teman-teman S1 Teknologi Informasi yang telah berjuang dan menimba ilmu bersama penulis, Dias Bayu Rizqianto, Yudha Naufal Rizky, Fahmi Tri Rahmadani, Muhammad Sultan Khabibi, Alif Haidar, Syahrul dan teman-teman lain.
11. Streamer yang menemani dan memberi hiburan kepada penulis dikala jenuh dalam mengerjakan, Luthfi Halimawan.
12. Semua pihak yang telah membantu hingga terselesaikan pembuatan skripsi yang tidak dapat disebutkan satu persatu.

Sebagai penutup, semoga Allah membalas segala kebaikan dan ketulusan yang telah diberikan dalam membantu penyusunan skripsi ini dengan limpahan nikmat dan karunia-Nya. Harapannya, skripsi ini dapat menjadi referensi yang bermanfaat serta memberikan kontribusi bagi penelitian-penelitian selanjutnya.

Semarang, 22 Juni 2025

Penulis

DAFTAR ISI

TUGAS AKHIR	i
LEMBAR PERSEMBAHAN	iii
MOTTO	v
ABSTRAK	vii
KATA PENGANTAR	ix
DAFTAR ISI	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvii
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	8
C. Tujuan Penelitian.....	8
D. Batasan Penelitian.....	8
E. Manfaat Penelitian	9
BAB II LANDASAN PUSTAKA	12
A. Kajian Pustaka.....	12
1. <i>Lost and Found</i>	12
2. Sistem Informasi	13
3. <i>Website</i>	15
4. <i>Hyper Text Markup Language (HTML)</i>	16
5. <i>Cascading Style Sheets (CSS)</i>	17
6. <i>Hypertext Preprocessor (PHP)</i>	18

7. <i>Visual Studio Code (VSCode)</i>	19
8. <i>Unified Model Language (UML)</i>	20
9. <i>Framework Laravel</i>	31
10. <i>Location Based Service (LBS)</i>	33
11. <i>Feature Driven Development (FDD)</i>	35
12. <i>Black Box Testing</i>	41
B. Kajian Penelitian yang Relevan.....	43
BAB III METODE PENELITIAN	49
A. <i>Develop an overall model</i> (Pengembangan model secara keseluruhan)	49
B. <i>Build a feature list</i> (Membangun daftar list fitur).....	53
C. <i>Plan by feature</i> (Perencanaan berdasarkan fitur).....	70
D. <i>Design by Feature</i> (Desain Fitur).....	75
E. <i>Build by Feature</i> (Membangun fitur).....	85
BAB IV HASIL DAN PEMBAHASAN	96
A. Implementasi Perangkat Lunak.....	96
B. Implementasi Perangkat Keras	97
C. Implementasi Basis Data	97
1. Tabel Users	98
2. Tabel barang	99
3. Tabel FAQ	100
4. Tabel Notifikasi	100
D. Implementasi Sistem	100
1. Halaman <i>Dashboard</i>	101

2.	Halaman Login	102
3.	Halaman Register	103
4.	Halaman About	104
5.	Halaman FAQ	104
6.	Halaman Lost & Found	105
7.	Halaman Form Lost & Found	108
8.	Halaman Detail Lost & Found	111
9.	Halaman Profile	112
10.	Halaman List Barang - Admin	114
11.	Halaman List Pengguna - Admin	115
12.	Halaman List FAQ - Admin	117
E.	Hasil Implementasi Perencanaan Waktu	119
F.	Hasil Pengujian Sistem	120
1.	Hasil Pengujian Blackbox	120
2.	Hasil Pengujian <i>User Acceptance Test</i> (UAT)	126
BAB V KESIMPULAN		132
A.	Kesimpulan	132
B.	Saran	133
DAFTAR PUSTAKA		134
LAMPIRAN		138

DAFTAR TABEL

Tabel 2.1 Notasi Use Case Diagram.....	22
Tabel 2.2 Notasi Activity Diagram	24
Tabel 2.3 Notasi Sequence Diagram	26
Tabel 2.4 Notasi Class Diagram	29
Tabel 2.5 Perbandingan Metode <i>Agile Software Development</i>	40
Tabel 2.6 Kajian penelitian terdahulu.....	43
Tabel 3.1 Pertanyaan Wawancara	51
Tabel 3.2 Daftar kebutuhan pengguna	52
Tabel 3.3 Skenario kasus pada sistem register user	55
Tabel 3.4 Skenario utama register user.....	55
Tabel 3.5 Skenario kasus pada sistem login user	56
Tabel 3.6 Skenario Utama sistem login user	56
Tabel 3.7 Skenario kasus pada sistem edit profile user	57
Tabel 3.8 Skenario utama edit profile user	58
Tabel 3.9 Skenario kasus pada sistem posting user.....	58
Tabel 3.10 Skenario utama posting user	59
Tabel 3.11 Skenario kasus pada sistem edit user	60
Tabel 3.12 Skenario utama edit user	60
Tabel 3.13 Skenario kasus pada sistem delete user	61
Tabel 3.14 Skenario utama delete user	62
Tabel 3.15 Skenario kasus pada sistem login admin	62
Tabel 3.16 Skenario utama login admin	63
Tabel 3. 17 Skenario kasus pada sistem edit profile admin	63
Tabel 3.18 Skenario utama edit profile admin.....	64

Tabel 3.19 Skenario kasus pada sistem posting admin	65
Tabel 3.20 Skenario utama posting admin	65
Tabel 3.21 Skenario kasus pada sistem edit admin	66
Tabel 3.22 Skenario utama edit admin	67
Tabel 3.23 Skenario kasus pada sistem delete admin	67
Tabel 3.24 Skenario utama delete admin	68
Tabel 3.25 Skenario kasus pada sistem view	69
Tabel 3.26 Skenario utama view	69
Tabel 3.27 Jadwal pengembangan sistem informasi Lost and found	71
Tabel 3.28 Gantt Chart	74
Tabel 4.1 Implementasi Perangkat Lunak.....	96
Tabel 4.2 Implementasi Perangkat Keras.....	97
Tabel 4.3 Database User	98
Tabel 4.4 Database Barang.....	99
Tabel 4.5 Database FAQ	100
Tabel 4.6 Database Notifikasi	100
Tabel 4.7 Gantt Chart Perbandingan Perencanaan dan Aktual ..	119
Tabel 4.8 Hasil Pengujian Blackbox Admin	121
Tabel 4.9 Hasil Pengujian Blackbox User	122
Tabel 4.10 Bobot Penilaian Responden	126
Tabel 4.11 Kriteria Inspeksi Skor	127
Tabel 4.12 Hasil UAT Responden	127
Tabel 4.13 Hasil Skor Penghitungan UAT Responden	128
Tabel 4.14 Hasil Persentase.....	130

DAFTAR GAMBAR

Gambar 2.1 Struktur dari Model-view-controller (MVC)	32
Gambar 2.2 Teknologi Location Based Service (LBS)	34
Gambar 2.3 Tahapan pengembangan FDD	36
Gambar 3.1 Tahapan pengembangan FDD	49
Gambar 3.2 Use case diagram	54
Gambar 3.3 Gantt chart grafik	74
Gambar 3.4 Activity Diagram Login/Register User	76
Gambar 3.5 Activity Diagram View Page.....	77
Gambar 3.6 Activity Diagram Post Item	78
Gambar 3.7 Activity Diagram FAQ Admin	79
Gambar 3.8 Sequence diagram register akun user	80
Gambar 3.9 Sequence diagram login akun user	81
Gambar 3.10 Sequence diagram view page	82
Gambar 3.11 Sequence diagram post item user.....	83
Gambar 3.12 Sequence diagram FAQ user	84
Gambar 3.13 Class diagram Lost and found	85
Gambar 3.14 Desain form login user	86
Gambar 3.15 Desain form register user	87
Gambar 3.16 Desain halaman dashboard	89
Gambar 3.17 Desain halaman tentang website.....	90
Gambar 3.18 Desain halaman list item.....	92
Gambar 3.19 Desain halaman detail item	94
Gambar 3.20 Desain halaman user profile	95

Gambar 4.1 Database Lost & Found	98
Gambar 4.2 Halaman Dashboard User	101
Gambar 4.3 Halaman Login	102
Gambar 4.4 Halaman Register	103
Gambar 4.5 Halaman About.....	104
Gambar 4.6 Halaman FAQ.....	105
Gambar 4.7 Halaman Lost & Found	106
Gambar 4.8 Halaman Form Lost & Found	109
Gambar 4.9 Halaman Detail Lost & Found	111
Gambar 4.10 Halaman Profile	113
Gambar 4.11 Halaman Profile Riwayat Upload	113
Gambar 4.12 Halaman List Barang Admin	115
Gambar 4.13 Halaman List Pengguna Admin	116
Gambar 4.14 Halaman List FAQ Admin	118

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan pesat teknologi digital dalam beberapa dekade terakhir telah mengubah banyak aspek kehidupan manusia. Salah satu elemen paling signifikan dari perkembangan teknologi ini adalah internet, yang kini telah menjadi elemen esensial dalam kehidupan modern. Layanan-layanan yang dulunya hanya tersedia secara fisik kini telah beralih ke format daring, memberikan fleksibilitas dan kenyamanan bagi pengguna untuk menjalankan berbagai aktivitas mereka. *Website* adalah salah satu dari banyaknya perkembangan teknologi informasi melalui internet yang sangat cepat.

Website dapat didefinisikan sebagai kumpulan halaman yang menyajikan beragam informasi, seperti teks, gambar, animasi, suara, video, atau kombinasi dari berbagai jenis data digital lainnya. Rangkaian data ini saling terhubung, di mana setiap halaman dihubungkan melalui hyperlink yang memudahkan navigasi antar halaman (Novianto Pratama, 2021).

Website berfungsi sebagai platform digital untuk memudahkan pengguna dalam berkomunikasi, berbagi

informasi, serta melakukan transaksi. Perkembangan teknologi yang pesat saat ini membawa banyak dampak terutama pada cara manusia bekerja, kemajuan teknologi yang berhubungan dengan komputer penting untuk diikuti.

Teknologi telah menjadi suatu kebutuhan yang sangat mendasar dalam kehidupan modern, apalagi semakin berkembangnya teknologi saat ini berkaitan dengan kebutuhan mendesak akan informasi yang tepat dan akurat, menjadikan peran teknologi informasi pada suatu instansi sangat berpengaruh dan berbagai indikasinya diperlukan agar proses tersebut dapat berjalan. Salah satu inovasi dalam ranah teknologi informasi adalah pengembangan sistem informasi berbasis komputer, yang banyak diterapkan dalam berbagai konteks. Pembuatan sistem informasi tersebut memungkinkan pemanfaatan komputer atau perangkat seluler untuk memberikan solusi yang cepat, akurat, dan efisien. Di samping itu, sistem informasi mampu menjalankan instruksi dari pengguna untuk mencapai tujuan tertentu.

Teknologi informasi merupakan elemen penting yang sangat diperlukan untuk mendukung operasional dan pengambilan keputusan.

Salah satu layanan yang dapat dioptimalkan melalui teknologi digital adalah sistem pencarian dan penemuan barang hilang. Barang-barang pribadi seperti buku, tas,

alat tulis, hingga perangkat elektronik seringkali hilang dan sulit untuk ditemukan kembali. Situasi tersebut bukan hanya mengakibatkan kerugian materi, tetapi juga menghadirkan ketidaknyamanan bagi pemiliknya. Di sisi lain, ada pula individu yang dengan niat baik menemukan barang milik orang lain dan ingin mengembalikannya, namun seringkali mereka tidak tahu cara untuk menghubungi pemilik barang tersebut, seperti apa yang diucapkan pada Surat (QS. An-Nisa' Ayat 58).

إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ ۖ أَنْ تَحْكُمُوا بِالْعَدْلِ ۚ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا ﴿٥٨﴾

Artinya : Sesungguhnya Allah menyuruh kamu menyampaikan amanah kepada pemiliknya. Apabila kamu menetapkan hukum di antara manusia, hendaklah kamu tetapkan secara adil. Sesungguhnya Allah memberi pengajaran yang paling baik kepadamu. Sesungguhnya Allah Maha Mendengar lagi Maha Melihat.

Ayat ini berisi perintah Allah agar amanah disampaikan kepada yang berhak. Kata “amanah” mencakup segala bentuk titipan dan hak orang lain, termasuk barang hilang yang ditemukan. Dalam Islam, barang temuan adalah amanah yang wajib dijaga dan dikembalikan kepada pemiliknya. Mengabaikannya berarti melanggar perintah Allah. Ayat ini juga menegaskan

pentingnya keadilan dan kejujuran sebagai dasar dalam bermasyarakat.

Banyak platform yang dapat membantu seseorang dalam mempermudah untuk melakukan pencarian atau penemuan dalam barang hilang sebagai contoh sudah terdapat sarana media informasi yang bisa digunakan untuk menginformasikan kehilangan, seperti televisi, koran, radio, dan media informasi lainnya. Dengan banyaknya informasi seperti itu, maka diperlukan sebuah wadah untuk menampung data dan informasi kehilangan yang lebih terpusat agar dapat dimanfaatkan pengguna untuk mengumumkan informasi dan mencari informasi kehilangan.

Wadah tersebut adalah teknologi internet (Setiawan et al., 2021). Maka dari itu peneliti melakukan rancangan *website* sistem informasi dalam platform pencarian maupun penemuan barang di lingkungan Universitas Islam Negeri (UIN) Walisongo, sebagai media alat bantu mahasiswa Universitas Islam Negeri (UIN) Walisongo Semarang dalam melakukan pencarian maupun penemuan barang dengan memanfaatkan rancangan *website* sistem informasi yang peneliti rancang.

Universitas Islam Negeri Walisongo Semarang, merupakan Universitas dengan 8 Fakultas dan 1 Pascasarjana berisikan 39 program sarjana, 9 program

magister, 1 program doktor, dan 1 program profesi. Universitas Islam Negeri (UIN) Walisongo Semarang merupakan perguruan tinggi negeri yang berada di kota Semarang, Jawa Tengah. Sebagian besar mahasiswa yang menemukan barang cenderung mengumumkan penemuan barang mereka dengan membuat informasi di meja lobby gedung atau resepsionis tiap fakultas. Peneliti berpendapat bahwa metode ini tidak efektif, karena penjaga yang menerima informasi tersebut tidak selalu mengetahui pemilik dari barang tersebut. Selain itu ini juga terjadi sebaliknya terhadap kehilangan barang, pemilik barang juga tidak selalu mengetahui atau mengingat dimana lokasi barang yang mereka miliki hilang.

Situasi yang mungkin muncul adalah ketika seorang mahasiswa yang kehilangan barang menanyakan di salah satu fakultas, sementara orang yang menemukan barang tersebut melaporkan informasi di fakultas lain. Hal ini menyebabkan informasi tidak sampai kepada pihak yang membutuhkan dan dinilai kurang efektif. Sebagian mahasiswa UIN Walisongo juga menggunakan media sosial, seperti Facebook, Whatsapp, dan Twitter, untuk menyebarkan informasi kehilangan karena dianggap lebih praktis. Namun, kelemahan dari pendekatan ini adalah status mengenai kehilangan dapat dengan cepat tenggelam di antara pembaruan status baru dari pengguna lainnya.

Mengingat permasalahan tersebut, Peneliti merasa penting untuk memiliki sarana khusus yang dapat menampung penyebaran informasi kehilangan barang di lingkungan mahasiswa UIN Walisongo.

Maka dari itu perlu adanya platform sistem informasi kehilangan barang berbasis web yang dapat diakses kapan saja yang dapat dimanfaatkan sebagai media penyedia informasi guna untuk membantu dalam melakukan pencarian barang yang hilang dan memberikan platform bagi penemu agar lebih mudah untuk melaporkan barang yang mereka temukan, Mengingat pentingnya isu ini, penelitian ini bertujuan untuk mengembangkan sebuah solusi berbasis teknologi yang dapat memfasilitasi proses pencarian barang hilang.

Solusi yang diusulkan adalah platform "Lost and Found" yang dirancang khusus dengan menggunakan metode pengembangan Feature-Driven Development (FDD). FDD adalah salah satu variasi metode Agile yang menggunakan pendekatan langsung untuk membuat pengembangan sistem lebih sederhana, mudah dipahami, dan mudah diterapkan (Tamir, 2023).

Informasi mengenai lokasi barang yang ditemukan sangat penting untuk mempermudah pengguna dalam proses pencariannya. Untuk meningkatkan fungsionalitas *website* ini, ditambahkan fitur Location-Based Service

(LBS), yaitu teknologi yang memberikan informasi berdasarkan lokasi pengguna. Teknologi LBS dapat diakses melalui perangkat mobile dan memanfaatkan jaringan untuk menampilkan posisi geografis perangkat tersebut (Ananda et al., 2023). Menurut penelitian oleh Suciati dan Arafat (2023), LBS juga memungkinkan terjadinya komunikasi dan interaksi dua arah antara pengguna dan penyedia layanan. Dalam hal ini, pengguna memberikan informasi mengenai posisi mereka kepada penyedia layanan agar dapat memperoleh informasi yang relevan sesuai dengan lokasi yang ditentukan. Dengan dukungan fitur LBS, *website* ini diharapkan dapat menjawab tantangan yang dihadapi dalam pencarian barang hilang, serta berfungsi sebagai platform komunikasi yang efektif antara individu yang kehilangan barang dan mereka yang menemukannya.

Hal ini akan meningkatkan kemungkinan barang yang hilang dapat kembali kepada pemiliknya dengan lebih cepat dan efisien. diinginkan berdasarkan lokasi tersebut, dengan dukungan penggunaan dari fitur. LBS, *Website* ini diharapkan mampu menjawab tantangan yang ada dan menjadi wadah komunikasi efektif antara individu yang kehilangan barang dengan yang menemukannya.

B. Rumusan Masalah

1. Bagaimana merancang dan membangun sistem informasi *lost and found* untuk menampung dan menyebarkan informasi mengenai kehilangan dan penemuan barang di kalangan mahasiswa UIN Walisongo?
2. Bagaimana penerapan metode Feature-Driven Development (FDD) dalam proses pengembangan sistem serta evaluasinya melalui pengujian Blackbox dan User Acceptance Testing (UAT) untuk memastikan fungsionalitas dan tingkat kepuasan pengguna?

C. Tujuan Penelitian

1. Merancang dan membangun Sistem Informasi *lost and found* untuk dapat menampung dan menyebarkan informasi tentang kehilangan dan penemuan barang di kalangan mahasiswa UIN Walisongo.
2. Menerapkan metode Feature-Driven Development (FDD) dalam pengembangan sistem informasi *lost and found*, serta mengevaluasi hasilnya menggunakan pengujian Blackbox dan User Acceptance Testing (UAT) guna memastikan fungsionalitas sistem dan penerimaan tingkat kepuasan oleh pengguna.

D. Batasan Penelitian

Agar fokus pembahasan dalam penelitian ini tetap terjaga, maka ditentukan batasan masalah sebagai berikut:

1. Menghasilkan produk berupa *website* lost and found berbasis *Location Based System* (LBS) untuk mahasiswa Universitas Islam Negeri (UIN) Walisongo Semarang
2. Menggunakan metode *Feature Driven Development* (FDD) yang digunakan untuk mengembangkan *website* tersebut.
3. *Website* yang dirancang hanya berupa menampilkan barang yang di upload oleh *user*, dan menampilkan fitur *Location Based Service* (LBS) berdasarkan apa yang diinput oleh user
4. Apabila Warga luar UIN Walisongo Semarang ingin melaporkan mengenai kehilangan atau penemuan benda di lingkungan UIN Walisongo Semarang, maka dapat menghubungi lobby atau resepsionis tiap fakultas sebagai *admin*.

E. Manfaat Penelitian

Pada penelitian ini, penulis berharap agar hasil dari penelitian ini dapat memberikan manfaat bagi pembaca, baik dalam aspek teoritis maupun praktis.

1. Manfaat Teoritis
 - a. Penelitian ini dapat memberikan pemahaman yang lebih mendalam tentang alur pembuatan sistem informasi berbasis web, khususnya dalam konteks pencarian dan penemuan barang hilang.

- b. Penelitian ini menjelaskan penerapan metode Feature-Driven Development (FDD) dalam pengembangan sistem, sehingga pembaca dapat memahami bagaimana pendekatan Agile dapat meningkatkan efisiensi dalam proses pengembangan.
 - c. Hasil penelitian ini dapat dijadikan referensi dan masukan bagi penelitian lain yang berkaitan dengan pengembangan sistem informasi, terutama dalam konteks organisasi kampus atau lembaga pendidikan lainnya.
2. Manfaat Praktis
- a. Bagi Mahasiswa:
 - 1) Platform Sistem informasi ini membantu mahasiswa dalam mengelola pencarian dan penemuan barang hilang secara lebih efektif, sehingga mengurangi stres dan kebingungan saat kehilangan barang.
 - 2) Mahasiswa dapat dengan mudah mengakses informasi tentang barang yang hilang dan *ditemukan*, yang meningkatkan kemungkinan barang kembali kepada pemiliknya.
 - b. Bagi Pihak Universitas: Hasil dari pengembangan platform ini dapat berfungsi

sebagai alat untuk meningkatkan transparansi dan komunikasi di antara mahasiswa, dosen, dan pihak universitas terkait barang hilang, yang pada gilirannya dapat meningkatkan citra institusi.

- c. Bagi Dosen: Dosen dapat menggunakan platform ini untuk mengawasi dan menilai tingkat kesadaran mahasiswa mengenai pengelolaan barang pribadi. Ini juga memberikan mereka wawasan tentang perilaku mahasiswa dalam menangani kehilangan barang.
- d. Bagi Pengguna Umum: Platform ini juga dapat digunakan oleh pengguna di luar lingkungan kampus, seperti masyarakat umum, untuk melaporkan dan mencari barang hilang, sehingga meningkatkan manfaat sosial dari sistem yang dikembangkan.

BAB II

LANDASAN PUSTAKA

A. Kajian Pustaka

1. *Lost and Found*

Lost and found adalah istilah yang merujuk pada proses atau sistem yang digunakan untuk membantu menemukan barang yang hilang dan untuk mengembalikan barang yang telah *ditemukan* kepada pemilik yang sah (Nuryasin & Budianti, 2020). Konsep ini umum digunakan di tempat umum, seperti sekolah, universitas, pusat perbelanjaan, stasiun, bandara, dan berbagai institusi lainnya. *Lost and found* menggunakan 2 komponen penting yaitu terdapat *Lost* yang berarti kehilangan dan *Found* yang berarti menemukan.

a. *Lost*

Lost merupakan suatu kategori yang mencakup semua informasi mengenai benda yang kehilangan dengan informasi dari benda secara *detail* untuk membantu *user* yang menggunakan web “*Lost and found*” bisa mengetahui dan jika menemukan benda yang sesuai dapat diketahui oleh pengguna untuk harus dikembalikan ke yang bersangkutan.

b. *Found*

Found berisikan informasi mengenai semua benda yang telah ditemukan oleh *user* lain, dengan mencantumkan deskripsi dari barang yang ditemukan dengan lengkap agar pemilik dari benda tersebut dapat dengan mudah menemukan benda yang mereka cari sesuai dengan ciri-ciri yang telah di deskripsikan.

Lost and Found adalah sistem yang mempermudah pelacakan barang hilang dan pelaporan barang ditemukan. Sistem ini memastikan barang dikembalikan kepada pemilik yang sah melalui proses verifikasi. Dapat diimplementasikan secara manual atau digital, tujuannya adalah meningkatkan efisiensi dan transparansi dalam pengelolaan barang hilang (Wibisono, 2015).

2. Sistem Informasi

Sistem Informasi merupakan integrasi antara perangkat keras, perangkat lunak, dan sumber daya manusia yang bekerja secara terkoordinasi untuk mengubah data mentah menjadi informasi yang bermanfaat. Proses ini melibatkan tahapan-tahapan penting, seperti pengumpulan, penyimpanan, pengelolaan, pemrosesan, dan penyajian data menjadi informasi yang bernilai. Untuk menjalankan proses tersebut, diperlukan perangkat keras dan perangkat lunak yang sesuai, serta partisipasi aktif individu dalam mengoperasikan sistem secara efektif. Partisipasi ini

mencakup kemampuan untuk menggunakan sistem dengan baik dan membuat keputusan berdasarkan informasi yang dihasilkan. Dengan demikian, Sistem Informasi tidak hanya menjadi alat teknologi, tetapi juga kerangka kerja yang mendukung pengambilan keputusan berbasis data secara strategis dan efisien (Imananda, 2024).

Data dalam sistem informasi mencakup angka, teks, gambar, atau bentuk lain yang dicatat dan disimpan, sementara informasi adalah hasil pengolahan data yang diubah menjadi bentuk yang terstruktur, relevan, dan berguna untuk pengambilan keputusan. Sistem informasi membantu organisasi dalam mengumpulkan data yang relevan, menganalisisnya, menghasilkan laporan, dan menyediakan informasi yang dibutuhkan, serta mendukung komunikasi dan kolaborasi antar unit dalam organisasi (Imananda, 2024). Untuk memastikan sistem berfungsi dengan optimal, diperlukan pengawasan yang mencakup kontrol atas data masukan (input), data keluaran (output), dan keseluruhan operasi sistem. Sub-sistem input berperan menerima data dari berbagai sumber, mengatur frekuensi pemasukan, dan menentukan jenis data yang akan diolah, sehingga hasil akhirnya mampu memberikan informasi yang bernilai dalam mendukung keputusan strategis organisasi (Komara, 2024).

Dari penjelasan di atas, dapat disimpulkan bahwa Sistem Informasi adalah integrasi antara perangkat keras, perangkat lunak, dan sumber daya manusia yang bekerja secara terkoordinasi untuk mengolah data mentah menjadi informasi yang bermanfaat. Sistem ini mencakup berbagai tahapan penting, mulai dari pengumpulan, penyimpanan, pengelolaan, pemrosesan, hingga penyajian data menjadi informasi yang relevan, terstruktur, dan bernilai. Informasi yang dihasilkan berperan penting dalam mendukung pengambilan keputusan strategis, meningkatkan efisiensi operasional, serta memfasilitasi komunikasi dan kolaborasi antarunit dalam organisasi. Agar sistem dapat berfungsi optimal, diperlukan pengawasan yang mencakup kontrol atas data masukan, keluaran, serta keseluruhan proses operasionalnya. Dengan peran strategis ini, Sistem Informasi tidak hanya menjadi alat teknologi, tetapi juga kerangka kerja yang membantu organisasi mencapai tujuan dengan cara yang lebih efektif dan efisien.

3. *Website*

Website, secara umum, adalah kumpulan halaman yang terhubung dalam satu domain atau subdomain di internet, yang merupakan bagian dari World Wide Web (WWW). Setiap halaman biasanya berupa dokumen dalam format HTML (*Hyper Text Markup Language*) yang dapat diakses menggunakan protokol HTTP. Protokol ini memungkinkan

informasi dari server dikirim ke browser *user* agar dapat ditampilkan secara visual. *Website* dirancang untuk dapat diakses secara luas melalui aplikasi peramban dengan URL (Uniform Resource Locator), yang merupakan alamat unik untuk mengakses suatu situs. Salah satu contoh URL yang populer adalah <http://www.google.com>. (Anamisa & Mufarroha, 2020)

Sebuah *website* dapat menyajikan berbagai jenis informasi dalam bentuk teks, gambar, video, suara, atau gabungan dari elemen-elemen tersebut. Semua data ini disimpan di server hosting. *Website* dapat dimiliki oleh individu maupun organisasi, dengan tujuan yang bervariasi, seperti edukasi, bisnis, hiburan, atau keperluan pribadi. Struktur dan konten setiap *website* dirancang untuk memenuhi tujuan tertentu, sehingga dapat memberikan informasi atau layanan yang relevan kepada penggunaanya.

4. *Hyper Text Markup Language* (HTML)

Hyper Text Markup Language (HTML) adalah bahasa standar yang digunakan dalam pembuatan halaman web. HTML berfungsi sebagai struktur dasar dalam membangun situs web dengan elemen-elemen seperti teks, gambar, audio, video, dan elemen interaktif lainnya

Seiring dengan berkembangnya teknologi web, HTML mengalami beberapa pembaruan, dengan HTML5 sebagai

versi terbaru. HTML5 hadir dengan berbagai peningkatan dibandingkan dengan versi sebelumnya, HTML4.01, seperti dukungan bawaan untuk multimedia (audio dan video), penanganan dokumen yang lebih baik, serta pengurangan ketergantungan pada JavaScript untuk beberapa fungsi.

Salah satu fitur utama HTML5 adalah Web-Based Graphic Library (WebGL) yang memungkinkan pembuatan tampilan grafis tiga dimensi dalam web tanpa perlu plugin tambahan seperti Adobe Flash atau Microsoft Silverlight. Selain itu, HTML5 lebih kompatibel dengan berbagai perangkat dan browser modern, menjadikannya standar utama dalam pengembangan web saat ini (Mulia et al., n.d.).

HTML memiliki peran penting dalam pengembangan web modern, memungkinkan desainer dan pengembang untuk menciptakan situs web yang lebih interaktif, responsif, dan mudah diakses oleh berbagai platform.

5. *Cascading Style Sheets (CSS)*

Cascading Style Sheets (CSS) adalah bahasa yang digunakan untuk mengatur tampilan dan tata letak halaman web. CSS memungkinkan pemisahan antara struktur konten yang dibuat dengan HTML dan desain visualnya, sehingga memberikan fleksibilitas dalam pengelolaan tampilan situs web. (Kaushal et al., 2022)

CSS memiliki peran penting dalam mendukung desain responsif, yang memungkinkan halaman web menyesuaikan tampilan berdasarkan ukuran layar perangkat. Dengan fitur seperti media queries, elemen visual dapat berubah secara dinamis untuk memberikan pengalaman pengguna yang optimal di berbagai perangkat, dari desktop hingga smartphone.

Dalam pengembangan web modern, CSS juga sering digunakan bersama dengan PHP, terutama dalam aplikasi berbasis web dinamis. PHP dapat menghasilkan kode HTML yang kemudian diperindah menggunakan CSS, memungkinkan pembuatan antarmuka yang lebih menarik dan interaktif

6. *Hypertext Preprocessor (PHP)*

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang digunakan untuk membangun aplikasi web dinamis. PHP dapat diintegrasikan dengan HTML untuk menghasilkan halaman web interaktif, serta memungkinkan pengolahan data sebelum dikirim ke browser pengguna. Keunggulan utama PHP meliputi fleksibilitas, efisiensi, dan keamanan. PHP mendukung manajemen sesi, validasi formulir, manipulasi file, serta integrasi dengan layanan eksternal melalui API. Selain itu, PHP kompatibel dengan berbagai sistem operasi seperti Windows, Linux, dan macOS.

PHP juga memiliki ekosistem luas dengan banyak pustaka dan framework populer seperti Laravel, CodeIgniter, dan Symfony, yang mempercepat pengembangan dan meningkatkan keamanan aplikasi. Dengan kemudahan implementasi dan dukungan komunitas yang luas, PHP tetap menjadi pilihan utama dalam pengembangan web modern (Joseph Kurien et al., 2022).

7. *Visual Studio Code (VSCode)*

Visual Studio Code (VSCode) merupakan editor kode sumber terbuka yang dikembangkan oleh Microsoft. Editor ini dirancang untuk digunakan di berbagai sistem operasi seperti Windows, macOS, dan Linux. VSCode sangat populer di kalangan pengembang perangkat lunak karena ringan, cepat, dan memiliki berbagai fitur yang mendukung produktivitas dalam pengembangan aplikasi.

VSCode mendukung berbagai bahasa pemrograman secara default, termasuk JavaScript, TypeScript, dan Node.js. Selain itu, ekosistem ekstensi yang luas memungkinkan pengguna untuk menambahkan dukungan untuk bahasa lain seperti C++, C#, Java, Python, PHP, Go, dan .NET. Fitur utama VS Code mencakup debugging yang efektif, penyorotan sintaks, IntelliSense (penyelesaian kode cerdas), kontrol Git terintegrasi, dan terminal bawaan (Ariq Sulthon Hanif, n.d.).

VSCode memiliki kemampuan untuk diintegrasikan dengan berbagai tools dan framework yang mendukung pengembangan perangkat lunak modern. Ekstensi yang tersedia di marketplace VSCode memungkinkan pengguna untuk menyesuaikan editor sesuai kebutuhan spesifik mereka, mulai dari pengembangan web, pemodelan ilmiah, hingga pengembangan aplikasi berbasis kecerdasan buatan.

8. *Unified Model Language (UML)*

Unified Modeling Language (UML) adalah standar spesifikasi yang digunakan untuk mendokumentasikan, merancang, dan membangun perangkat lunak. UML berfungsi sebagai metodologi dalam pengembangan sistem berorientasi objek serta sebagai alat pendukung dalam proses pengembangannya. Sebagai bahasa berbasis grafik, UML memungkinkan visualisasi, spesifikasi, konstruksi, dan dokumentasi sistem perangkat lunak berbasis *Object-Oriented (OO)*. Selain itu, UML menyediakan standar dalam pembuatan blueprint sistem, mencakup pemodelan proses bisnis, penulisan kelas dalam bahasa pemrograman tertentu, perancangan skema database, serta elemen-elemen

yang diperlukan dalam pengembangan perangkat lunak (Julius, 2021).

Seiring dengan perkembangannya, terdapat berbagai jenis diagram UML yang sering digunakan oleh pengembang perangkat lunak. Beberapa di antaranya meliputi use case diagram, activity diagram, class diagram, dan sequence diagram. Dari berbagai jenis tersebut, sequence diagram dan class diagram menjadi dua yang cukup penting dan memiliki penjelasan lebih lanjut sebagai berikut:

a. *Use Case Diagram*

Pada tahap awal pengembangan, diagram *use case* sering digunakan untuk menggambarkan fitur yang diharapkan dari sebuah sistem. Diagram ini berfungsi sebagai representasi visual yang menunjukkan cara suatu sistem atau kelas bekerja serta bagaimana sistem tersebut berinteraksi dengan lingkungan eksternal. Dengan demikian, use case memberikan gambaran sistem secara fungsional dari perspektif pengguna (Julius, 2021).

Selain itu, *use case* juga merepresentasikan interaksi antara aktor dan sistem. Hal ini sangat bermanfaat dalam menyusun kebutuhan sistem, mengomunikasikan rancangan kepada klien, serta

merancang test case untuk memastikan seluruh fitur dalam sistem berfungsi dengan baik. Adapun notasi use case yang digunakan dalam penelitian ini disajikan sebagai berikut(Julius, 2021).

Tabel 2.1 Notasi *Use Case Diagram*

Notasi	Nama Elemen & Fungsi
	Aktor merupakan entitas yang bisa berupa individu, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dikembangkan. Aktor ini berada di luar sistem informasi yang dibuat, meskipun dalam diagram, mereka tetap direpresentasikan dengan simbol aktor.
	Setiap fungsionalitas yang disediakan oleh sistem direpresentasikan sebagai unit-unit yang saling berkomunikasi, baik antarunit maupun dengan aktor. Biasanya, nama use case diawali dengan kata kerja untuk menggambarkan tindakan yang dilakukan dalam sistem.
	Dalam sebuah use case, terdapat komunikasi antara aktor dan unit fungsional yang berpartisipasi. Hal

	ini menunjukkan adanya interaksi antara aktor dan use case, di mana aktor dapat berperan sebagai pemicu atau penerima dalam proses tersebut.
<i><<Include>></i> ←-----	Ada dua pandangan utama tentang include dalam use case. Pertama, include memastikan use case yang ditambahkan selalu dipanggil saat use case tambahan dijalankan. Kedua, include berfungsi sebagai pengecekan, memastikan use case yang ditambahkan telah dijalankan sebelum proses dimulai.
<i><<Extend>></i> -----→	Use case tambahan berhubungan dengan use case utama tetapi bisa berdiri sendiri. Konsep ini mirip dengan inheritance dalam pemrograman, di mana use case tambahan biasanya memakai awalan nama yang sama dengan use case utama.
←—————	Hubungan generalisasi dan spesialisasi dalam <i>use case</i> menunjukkan bahwa satu fungsi lebih umum dibandingkan yang lain.

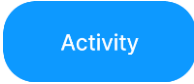
b. *Activity Diagram*

Activity Diagram digunakan untuk memvisualisasikan alur kerja atau aktivitas dalam suatu sistem maupun proses bisnis. Diagram ini menggambarkan bagaimana suatu sistem menjalankan aliran aktivitas secara terstruktur.

Selain itu, diagram ini menunjukkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, termasuk bagaimana aktivitas dimulai, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir. Setiap aktivitas dapat dikaitkan dengan satu atau lebih use case. Aktivitas merepresentasikan proses yang berjalan, sementara use case menggambarkan bagaimana aktor berinteraksi dengan sistem untuk melaksanakan aktivitas tersebut. Elemen keputusan juga digunakan untuk menggambarkan perilaku sistem dalam kondisi tertentu (Syahputra, 2023).

Tabel 2.2 Notasi *Activity Diagram*

Notasi	Nama Elemen & Fungsi
	Status awal (<i>Initial State</i>) dalam sistem menandai titik dimulainya suatu aktivitas. Setiap diagram aktivitas harus memiliki satu status awal sebagai titik awal eksekusi proses.

	Status akhir dalam suatu sistem menandakan bahwa aktivitas telah selesai. Dalam diagram aktivitas, setiap alur harus memiliki titik akhir yang menunjukkan penyelesaian proses.
	Setiap aktivitas dalam diagram umumnya diawali dengan kata kerja untuk menunjukkan tindakan atau proses yang sedang berlangsung.
	Penggabungan merupakan proses yang mengintegrasikan beberapa aktivitas menjadi satu kesatuan dalam diagram aktivitas.
	Percabangan (<i>Decision</i>) terjadi ketika suatu aktivitas memiliki lebih dari satu kemungkinan pilihan, memungkinkan sistem untuk menentukan jalur yang sesuai berdasarkan kondisi tertentu.
	Elemen <i>swimlane</i> digunakan untuk memisahkan peran atau bagian dalam organisasi yang bertanggung jawab

Nama Swimlane	atas aktivitas tertentu dalam suatu proses bisnis.
---------------	--

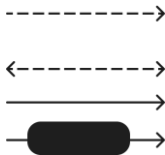
c. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antara objek yang terdapat di dalam maupun di sekitar sistem. Interaksi ini direpresentasikan dalam bentuk pesan yang dikirim dan diterima oleh objek-objek tersebut, serta diatur berdasarkan urutan waktu. Oleh karena itu, dalam pembuatan *sequence diagram*, penting untuk mengidentifikasi objek-objek yang berperan dalam proses yang akan digambarkan dalam sebuah *use case*. Diagram ini membantu dalam memahami bagaimana alur komunikasi antarobjek terjadi, sehingga memudahkan dalam analisis dan perancangan sistem secara lebih terstruktur (Komara, 2024). Berikut adalah notasi dari komponen pada *sequence diagram*.

Tabel 2.3 Notasi *Sequence Diagram*

Notasi	Nama Elemen & Fungsi
	Aktor, yaitu entitas yang berperan sebagai pengguna eksternal yang berinteraksi dengan sistem. Dalam diagram ini, aktor biasanya direpresentasikan dengan simbol figur

	berbentuk stik untuk menunjukkan keterlibatannya dalam proses komunikasi dengan sistem.
	<i>Activation box</i> menunjukkan durasi waktu yang diperlukan oleh suatu objek untuk menyelesaikan tugasnya. Semakin lama proses berlangsung, semakin panjang pula <i>activation box</i> yang digambarkan. Komponen ini direpresentasikan dalam bentuk persegi panjang untuk memvisualisasikan waktu eksekusi suatu aktivitas dalam <i>sequence diagram</i> .
	<i>Lifeline</i> direpresentasikan sebagai garis putus-putus dalam <i>sequence diagram</i> . Komponen ini biasanya dilengkapi dengan kotak yang berisi objek, yang berfungsi untuk menunjukkan aktivitas serta keberlangsungan peran objek tersebut dalam interaksi sistem.
	Penggabungan merupakan proses yang mengintegrasikan beberapa aktivitas menjadi satu kesatuan dalam diagram aktivitas.

	<i>Object</i> direpresentasikan sebagai kotak yang berisi nama objek dengan garis bawah. Komponen ini berperan dalam mendokumentasikan perilaku suatu objek dalam sistem, membantu dalam memahami interaksi dan fungsinya dalam <i>sequence diagram</i>.
	<i>Messages</i> atau pesan berfungsi untuk menunjukkan komunikasi antar objek dalam <i>sequence diagram</i>. Pesan-pesan ini muncul secara berurutan di sepanjang <i>lifeline</i> dan direpresentasikan dengan simbol anak panah. Komponen <i>messages</i> dan <i>lifeline</i> menjadi elemen utama dalam diagram ini, karena keduanya menggambarkan alur interaksi dalam sistem.

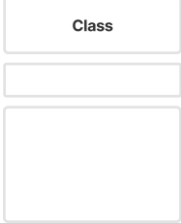
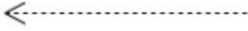
d. *Class Diagram*

Class Diagram merupakan salah satu jenis diagram struktur statis yang digunakan untuk merepresentasikan susunan sistem dengan menampilkan kelas-kelas beserta relasi di antaranya. Model diagram ini membantu pengembang perangkat lunak dalam mengidentifikasi hierarki kelas, atribut,

metode, serta berbagai jenis hubungan, seperti asosiatif, agregatif, dan komposisional dalam suatu sistem(Ariq Sulthon Hanif, n.d.).

Tabel 2.4 Notasi *Class Diagram*

Notasi	Nama Elemen & Fungsi
	Jika suatu kelas tidak dapat berdiri sendiri dan harus menjadi bagian dari kelas lain, maka kelas tersebut memiliki hubungan composition dengan kelas lain yang menjadi bagian pentingnya. Hubungan composition ini divisualisasikan dengan garis yang berujung jajaran genjang yang terisi penuh.
	<i>Aggregation</i> merepresentasikan hubungan antara keseluruhan dan bagian. Dalam banyak kasus, hubungan ini disebut sebagai relasi kepemilikan, di mana suatu objek terdiri dari bagian-bagian lainnya. Diagram <i>aggregation</i> ditunjukkan dengan garis yang memiliki ujung berbentuk jajaran genjang kosong.
	<i>Class</i> merupakan elemen fundamental dalam pemrograman berbasis objek. Sebuah <i>class</i>

	digambarkan dalam bentuk kotak yang terbagi menjadi tiga bagian utama: bagian atas berisi nama kelas, bagian tengah mendeskripsikan atribut atau properti kelas, dan bagian bawah menunjukkan metode yang dimiliki oleh kelas tersebut.
	Generalization menggambarkan hubungan pewarisan dalam konsep pemrograman berorientasi objek. Hubungan ini memungkinkan suatu kelas mewarisi atribut dan metode dari kelas lain. Dalam diagram, hubungan generalization direpresentasikan dengan panah yang memiliki ujung segitiga kosong.
	<i>Dependency</i> menunjukkan ketergantungan satu kelas terhadap kelas lainnya. Biasanya digunakan untuk menggambarkan suatu kelas yang memanfaatkan operasi atau layanan dari kelas

	lain. Simbol yang digunakan dalam diagram untuk dependency adalah panah putus-putus.
--	--

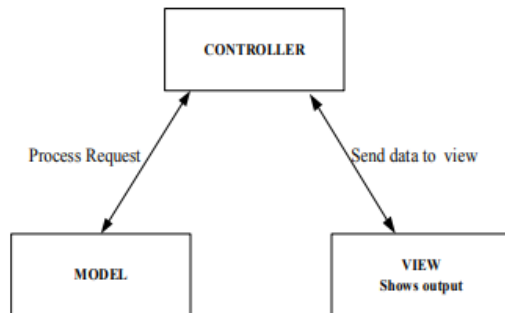
9. *Framework* Laravel

Laravel merupakan salah satu *framework* PHP yang paling banyak digunakan saat ini untuk pengembangan aplikasi berbasis web. Dengan perkembangan teknologi dan kebutuhan akan jaringan yang lebih luas, banyak individu dan perusahaan membutuhkan situs web yang dapat memberikan informasi tentang bisnis dan produk mereka. Namun, pendekatan tradisional dalam pengembangan web membutuhkan waktu yang lebih lama dan sering kali menemui berbagai kendala. Untuk mengatasi permasalahan tersebut, dikembangkan *framework* seperti Laravel yang menawarkan solusi praktis bagi pengembang.

Sebagai *framework* PHP open-source, Laravel menyediakan berbagai fitur untuk mempermudah pengembangan web, termasuk arsitektur desain Model-View-Controller (MVC), yang umum digunakan dalam pengembangan aplikasi web. Model menangani logika data dan pengelolaan database, View mengatur tampilan antarmuka pengguna, dan Controller

bertindak sebagai penghubung antara Model dan View dengan menangani permintaan pengguna serta mengelola alur kerja aplikasi.

Selain itu, Laravel juga menyediakan fitur seperti routing, templating dengan Blade, dan Eloquent ORM untuk memudahkan pengelolaan database. (Yadav et al., 2019)



Gambar 2.1 Struktur dari *Model-view-controller* (MVC)

- a. *Model*: Berhubungan dengan logika data, mengelola database, dan menangani data yang dikirim antara *View* dan *Controller*.
- b. *View*: Bertanggung jawab atas tampilan antarmuka pengguna dan menyajikan informasi kepada pengguna.
- c. *Controller*: Berperan sebagai penghubung antara *Model* dan *View*, memproses permintaan pengguna, serta mengelola komponen aplikasi

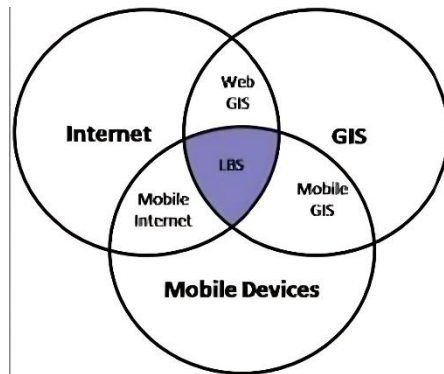
Didukung dengan sistem routing yang fleksibel, manajemen basis data yang efisien, dan berbagai alat bantu lainnya. Salah satu keunggulan utama Laravel adalah kemudahan dalam membangun situs web dengan struktur yang terorganisir serta penggunaan kembali kode yang telah tersedia, sehingga menghemat waktu pengembangan dan meningkatkan efisiensi kerja pengembang.

Selain itu, Laravel mendukung berbagai fitur keamanan untuk mencegah serangan seperti SQL injection, cross-site request forgery (CRF), dan cross-site scripting. Keamanan ini sangat penting terutama dalam pengembangan situs web e-commerce yang menangani transaksi dan data sensitif pelanggan. Dengan berbagai fitur yang ditawarkannya, Laravel menjadi pilihan utama dalam pengembangan web modern yang membutuhkan fleksibilitas, skalabilitas, dan keamanan tinggi (Yadav et al., 2019).

10. *Location Based Service (LBS)*

Location Based Service (LBS) adalah layanan informasi yang dapat diakses melalui perangkat pengguna menggunakan jaringan dan dapat menampilkan lokasi geografis perangkat tersebut. LBS memungkinkan komunikasi serta interaksi dua arah antara pengguna dan sistem.

Dengan demikian, pengguna memberikan informasi lokasi mereka kepada penyedia layanan untuk memperoleh informasi yang diperlukan. Layanan berbasis lokasi merupakan hasil integrasi tiga teknologi utama, yaitu Geographic Information System, Internet Service dan Mobile Devices.



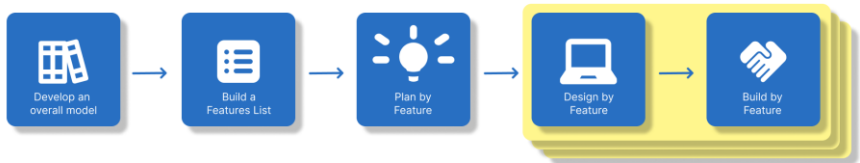
Gambar 2.2 Teknologi Location Based Service (LBS)

Location Based Service adalah layanan yang secara aktif merespons perubahan posisi suatu entitas, sehingga dapat mendeteksi lokasi objek dan memberikan layanan sesuai dengan lokasi yang telah diketahui. Dalam teknologi LBS berbasis jaringan seluler, penentuan posisi perangkat komunikasi bergerak ditentukan berdasarkan posisi relatif perangkat tersebut terhadap lokasi BTS (*Base Transceiver Station*) (Suciati & Arafat, 2023)

11. *Feature Driven Development (FDD)*

Feature Driven Development (FDD) adalah model pengembangan perangkat lunak *agile* yang sangat adaptif dan berfokus pada kualitas di semua fase pengembangan. Seperti namanya, fitur merupakan aspek yang sangat penting dalam FDD. Sebuah fitur adalah fungsi bernilai yang diinginkan oleh pengguna dalam perangkat lunak. Namun, FDD terutama berfokus pada fase perancangan (*design*) dan pembangunan (*building*).

Model ini menghasilkan hasil nyata secara berkala serta menyediakan informasi tentang kemajuan dan status proyek. Gagasan tentang FDD pertama kali muncul pada tahun 1999 oleh Jeff De Luca (Coad et al., 1999) menggunakan metode ini dalam sebuah proyek besar setelah menyadari bahwa metode pengembangan lainnya kurang efektif untuk menangani proyek besar dalam waktu tertentu (Al-Saqqa et al., 2020). Metode FDD memiliki siklus hidup yang mencakup lima tahapan berurutan, seperti terlihat dalam Gambar 2.3. Tahapan ini dijalankan secara bertahap dan berulang (iteratif).



Gambar 2.3 Tahapan pengembangan FDD

Penjelasan dalam tahapan pengembangan tersebut yaitu sebagai berikut (Al-Saqqa et al., 2020):

- a. *Develop an overall model* (Pengembangan model secara keseluruhan)

Tahap ini melibatkan pengembang dalam mendefinisikan konteks dan ruang lingkup proyek yang dibutuhkan. Berbagai model dapat dikembangkan oleh pengembang yang berbeda. Model-model tersebut kemudian dievaluasi untuk memilih yang paling sesuai dengan kebutuhan proyek.

- b. *Build a feature list* (Membangun daftar list fitur)

Pada tahap ini, model keseluruhan dan dokumen persyaratan digunakan untuk menyusun daftar fitur yang diperlukan oleh pengguna. Daftar fitur ini akan direview oleh klien sebelum akhirnya disetujui.

c. *Plan by feature* (Perencanaan berdasarkan fitur)

Rencana tingkat tinggi disusun berdasarkan daftar fitur yang telah disetujui. Rencana ini diurutkan sesuai prioritas pengguna dan keterkaitan antarfitur. Jadwal untuk *milestone* utama proyek dan detail jadwal setiap fitur juga termasuk dalam rencana ini. *Project Manager*, *Development Manager*, dan *Chief Programmer* terlibat dalam proses ini. *Chief programmer* kemudian menugaskan setiap fitur kepada pengembang tertentu, yang disebut sebagai *Class Owner*. Perancangan ini penulis merangkap keseluruhan tugas dari *Project Manager*, *Development Manager*, dan *Chief Programmer*.

d. *Design by Feature* (Desain Fitur)

Tahap ini bersifat iteratif, *chief programmer* dan *class owner* membuat *design package* untuk setiap kelas, termasuk diagram urutan. *Design package* dan diagram tersebut kemudian direview untuk mendapatkan persetujuan.

e. *Build by Feature* (Membangun fitur)

Ini adalah tahap terakhir dalam proses FDD. Pada tahap ini, desain diimplementasikan (dikodekan), kode diperiksa, dan proses pengujian

dilakukan. Tahap ini juga bersifat iteratif, mirip dengan tahap perancangan berbasis fitur. Setelah semua iterasi selesai, fitur yang telah dikembangkan dipublikasikan dalam build utama, dan proses dimulai kembali untuk serangkaian fitur baru.

Dalam metode FDD, terdapat enam peran utama. Setiap peran dapat dijalankan oleh lebih dari satu orang, dan satu anggota tim dapat memiliki lebih dari satu peran. Berikut adalah peran-peran tersebut (Al-Saqqa et al., 2020):

a. Project Manager

Bertanggung jawab sebagai pemimpin proyek, mengambil keputusan administratif, dan menciptakan lingkungan kerja yang optimal bagi tim.

b. Chief Architect

Memegang tanggung jawab utama dalam merancang keseluruhan desain perangkat lunak yang diminta. Persetujuan akhir terhadap desain juga berada di bawah kewenangannya.

c. Development Manager

Mengawasi seluruh tim pengembang dan memiliki keahlian teknis yang cukup untuk menyelesaikan masalah yang tidak dapat diatasi

oleh Chief Programmer. Selain itu, ia juga bertanggung jawab dalam menangani konflik antar tim.

d. Chief Programmer

Berperan sebagai pemimpin dalam proses pengembangan di dalam tim-tim kecil. Ia mengarahkan tim dalam seluruh tahap analisis, perancangan, dan implementasi. Selain itu, ia menentukan fitur yang harus dikerjakan dalam iterasi tertentu serta mengatasi permasalahan teknis yang muncul.

e. Class Owner

Memiliki tanggung jawab dalam merancang, mengembangkan, dan menguji fitur yang ada dalam proyek.

f. Domain Expert

Merupakan ahli dalam bidang bisnis tertentu yang memiliki pemahaman mendalam tentang cara kerja bisnis dan kebutuhan yang harus dipenuhi oleh perangkat lunak. Ia memberikan wawasan serta masukan untuk menyusun daftar fitur yang akan dikembangkan.

Metode ini kemudian dikembangkan lebih lanjut oleh Stephen Palmer dan Felsing John (Palmer & M. Felsing, 2002) agar lebih adaptif, cepat, dan dapat

diterapkan dalam pengembangan perangkat lunak berskala besar. Pengembangan ini menghasilkan tiga poin utama, yaitu:

- a. Berfokus pada kerja sama dan kolaborasi antar anggota tim.
- b. Mengelola kompleksitas perangkat lunak dengan membagi proses pengembangan berdasarkan fitur, yang kemudian diikuti dengan proses integrasi perangkat lunak.
- c. Melakukan komunikasi teknis secara intensif dan *mendetail* melalui berbagai cara, baik secara lisan, visual, maupun tertulis.

Perbandingan metode FDD dengan metode Agile lainnya dalam pengembangan software (Al-Saqqa et al., 2020):

Tabel 2.5 Perbandingan Metode *Agile Software Development*

Metode	Prinsip Utama	Keunggulan	Kelemahan
Feature Driven Development (FDD)	Pengembangan berbasis fitur, iterasi pendek dan inkremental	Cocok untuk proyek skala besar, mudah beradaptasi dengan perubahan, berfokus pada kualitas	Tidak memiliki panduan eksplisit dalam pengumpulan kebutuhan, membutuhkan tim dengan keterampilan tinggi

Scrum	Manajemen proyek berbasis iterasi, transparansi tinggi	Fleksibel terhadap perubahan, komunikasi tim yang tinggi, mudah dipahami	Tidak memberikan panduan teknis spesifik, membutuhkan disiplin tinggi
Extreme Program ming (XP)	Kolaboratif, cepat beradaptasi dengan perubahan	Meningkatkan produktivitas, mendukung pengujian berkelanjutan, meningkatkan komunikasi tim	Tidak cocok untuk tim yang terdistribusi, membutuhkan keterlibatan pelanggan yang tinggi

Inti dari FDD adalah mengelola pengembangan perangkat lunak berdasarkan daftar fitur yang dibutuhkan sesuai kebutuhan bisnis. FDD adalah metode pengembangan perangkat lunak yang sangat adaptif, sehingga dapat menerima perubahan kebutuhan perangkat lunak bahkan di tahap akhir. Fokus utama dari FDD adalah memberikan hasil berkualitas tinggi di setiap fase proses pengembangan.

12. *Black Box Testing*

Black Box Testing merupakan metode pengujian perangkat lunak yang menitikberatkan pada pengujian fungsionalitas eksternal sistem, tanpa memperhatikan

struktur internal atau kode program. Tujuan utamanya adalah untuk menemukan berbagai kesalahan seperti fungsi yang tidak berjalan sesuai harapan, error pada antarmuka pengguna, struktur data yang tidak tepat, masalah performa, serta kesalahan dalam proses inisialisasi atau penghentian sistem. Salah satu alat yang umum digunakan dalam pengujian ini adalah User Acceptance Test (UAT), yaitu dokumen berisi indikator dan prosedur untuk mengevaluasi fungsionalitas sistem (Komara, 2024)

Dalam penerapannya, black-box testing memiliki sejumlah keunggulan, antara lain:

- a. Penguji tidak perlu memahami bahasa pemrograman atau detail teknis sistem.
- b. Pengujian dilakukan berdasarkan perspektif pengguna, yang membantu mengidentifikasi ambiguitas atau ketidaksesuaian dengan spesifikasi kebutuhan.
- c. Tidak ada ketergantungan langsung antara programmer dan tester, karena mereka bekerja secara terpisah.

B. Kajian Penelitian yang Relevan

Tabel 2.6 Kajian penelitian terdahulu

No	Paper	Metode	Data (obyek)	Hasil
1	Pengembangan prototype back-end <i>website</i> Pelaporan barang hilang di pt. Presentologics Dicoding academy indonesia (Putra raharja & Wijayanto, 2023)	Metode Extreme Programming (XP)	Aplikasi ini menggunakan <i>Postman</i> sebagai pengujian unit dalam hal koneksi aplikasi, dan menggunakan <i>PostgreSQL</i> sebagai basis data nya dalam perancangan.	Hasil penelitian ini adalah pengembangan sistem pelaporan barang hilang dan ditemukan bernama "Lost and Found Finder". Sistem berbasis web ini menggunakan metode Extreme Programming (XP), teknologi <i>PostgreSQL</i> , dan API untuk memudahkan akses data. Fitur utamanya meliputi pencarian, pembuatan laporan, serta tiga peran pengguna: Finder, Seeker, dan Helper.

No	Paper	Metode	Data (obyek)	Hasil
				Hasil pengujian menunjukkan 19 halaman diuji dengan rata-rata respons 597 ms.
2	Aplikasi Web Pencarian Barang Hilang dengan Deteksi Lokasi Berbasis Google Map API (Chandra et al., 2023)	Metode Software Development Life Cycle (SDLC)	<i>Website</i> berikut menggunakan API Google Map untuk pendukung dalam rancangan nya, penyimpanan basis data menggunakan MySQL, untuk testing nya menggunakan BlackBox	Penelitian ini mengembangkan aplikasi web pencarian barang hilang berbasis Google Maps API. Menggunakan React.js untuk antarmuka yang sederhana dan efisien, aplikasi ini memungkinkan pelaporan barang hilang dan pelacakan lokasi penemuan. Fitur utamanya mencakup pendaftaran, kategori barang, verifikasi,

No	Paper	Metode	Data (obyek)	Hasil
				manajemen thread dan komentar, serta laporan. Sistem ini diharapkan membantu masyarakat menemukan barang hilang secara lebih efektif.
3	Pendekatan Metodologi Feature Driven Development Pada Aplikasi E-Commerce (Studi Kasus CV. Megajaya) (Zulvi et al., 2021)	Metode Feature Driven Development (FDD)	Aplikasi berikut melakukan pengumpulan data dengan melakukan wawancara secara langsung pada pemilik CV Mega Jaya,	Penelitian ini menunjukkan bahwa aplikasi e-commerce dengan metodologi Feature Driven Development (FDD) berhasil diselesaikan tepat waktu. Terdapat 1 perubahan fungsional dan penambahan 1 fitur (10% dari cakupan, waktu, dan biaya). FDD terbukti cocok untuk kasus

No	Paper	Metode	Data (obyek)	Hasil
				B2B dan lebih efektif jika diterapkan oleh tim berpengalaman.
4	Implementasi Konsep Business To Customers Nusa.Net Dengan Teknologi MCommerce Menggunakan Metode Feature Driven Development (Nur, 2021)	Metode Feature Driven Development (FDD)	Website ini melakukan wawancara dalam menganalisis data data yang dibutuhkan.	Penelitian ini menghasilkan strategi M-Commerce efektif bagi Nusa.net untuk meningkatkan daya saing, memperkuat hubungan pelanggan, dan menarik konsumen baru. Dikembangkan dengan metode Feature Driven Development (FDD), strategi ini mempermudah penyelesaian, pelaporan, dan pengawasan proyek. M-

No	Paper	Metode	Data (obyek)	Hasil
				Commerce diharapkan memudahkan pemesanan, pembayaran, serta mendukung keunggulan kompetitif Nusa.net.
5	Perancangan Aplikasi Pengenalan Pariwisata Danau Toba Berbasis Android Menggunakan Layanan Location Based Service (LBS) (Ananda et al., 2023)	Metode Waterfall	Data dari perancangan aplikasi ini dilakukan dengan memilih dan menentukan objek ataupun partisipan yang berpotensi untuk terlibat secara aktif dalam penelitian.	Penelitian oleh Gladian Dzaky Dirga Ananda dan tim merancang aplikasi Android berbasis LBS dan GPS untuk mengenalkan wisata Danau Toba. Aplikasi ini menyajikan informasi lokasi, jarak, fasilitas, dan jam operasional secara efisien. Pengguna cukup memasukkan email

No	Paper	Metode	Data (obyek)	Hasil
				untuk menerima PIN akses ke home yang menampilkan berbagai jenis wisata, seperti air terjun, perbukitan, sejarah, budaya, pantai, dan religi.

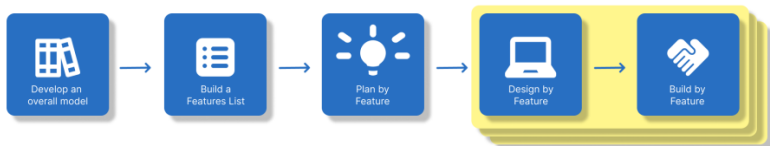
Studi menunjukkan bahwa pemilihan metodologi pengembangan perangkat lunak yang tepat sangat mempengaruhi efektivitas sistem. Metode seperti *Agile Development* terbukti mampu meningkatkan kinerja aplikasi dengan fitur-fitur utama seperti pencarian barang hilang, pembuatan laporan, dan manajemen peran pengguna (seperti Finder, Seeker, dan Helper).

Selain itu, integrasi teknologi pendukung seperti API, memungkinkan pencarian lokasi yang lebih akurat, dan meningkatkan kecepatan akses data. Hasil pengujian pada aplikasi *Lost and Found* juga menunjukkan performa yang stabil dengan respons cepat, mendukung kebutuhan pengguna dalam menemukan dan melaporkan barang secara efektif.

BAB III

METODE PENELITIAN

Penelitian ini menggunakan metode Feature Driven Development (FDD), salah satu implementasi Agile Model, untuk mengembangkan sistem dengan membagi kompleksitas aplikasi berdasarkan fitur dan menerapkan sistem iterasi dalam pembuatan fiturnya (Riady et al., 2022). FDD dipilih karena menyediakan infrastruktur pendukung, dokumentasi, serta informasi proyek secara tepat waktu. Metode ini bersifat iteratif dan inkremental dengan prosedur utama: *develop an overall model*, *build feature list*, *plan by feature list*, *design by feature*, dan *build by feature*.



Gambar 3.1 Tahapan pengembangan FDD

A. *Develop an overall model* (Pengembangan model secara keseluruhan)

Pada tahapan ini, penulis akan fokus pada perencanaan awal sistem dengan mengidentifikasi fungsi-fitur utama. Tahap pertama melibatkan pengumpulan data penelitian

melalui studi literatur, observasi, atau wawancara dengan stakeholder seperti mahasiswa, pengguna luar, dan admin. Data tersebut dianalisis untuk memahami kebutuhan pengguna.

Selanjutnya, penulis menyusun model keseluruhan sebagai kerangka dasar dan membuat daftar kebutuhan pengguna berdasarkan permasalahan yang ada. Setelah itu, penulis membuat use case untuk memvisualisasikan interaksi pengguna dengan sistem. Use case ini menjadi dasar untuk analisis lebih lanjut dan pengembangan fitur sistem secara terstruktur.

1. Pengumpulan data penelitian

a. Observasi

Pengumpulan data yang dilakukan secara langsung dan teratur. Peneliti mengamati langsung ke kampus Universitas Islam Negeri Walisongo Semarang dalam mencari informasi terkait dan permasalahan yang ada di lokasi kampus mengenai permasalahan barang hilang untuk memperoleh data yang dibutuhkan dalam penyusunan skripsi ini.

b. Wawancara

Pada tahap wawancara ini, peneliti melakukan pengumpulan data dengan memberikan pertanyaan langsung terhadap

mahasiswa jurusan untuk mendapatkan informasi mengenai pertanyaan seputar barang hilang.

Tabel 3.1 Pertanyaan Wawancara

No.	Pertanyaan
1	Apa saja kendala yang biasanya kamu hadapi ketika kehilangan atau menemukan barang di lingkungan kampus?
2	Menurut kamu, fitur apa saja yang penting untuk ada dalam sebuah website Lost and Found agar memudahkan proses pelaporan dan pencarian barang?
3	Bagaimana pendapat kamu tentang penggunaan website sebagai solusi digital dibandingkan dengan metode konvensional seperti papan pengumuman atau media sosial?

c. Studi Literatur

Setelah melakukan observasi dan wawancara, langkah berikutnya adalah melakukan studi literatur. Ini mencakup pengumpulan data, membaca dan mencatat informasi dari berbagai sumber seperti buku, e-book, e-jurnal, dan modul pembelajaran yang berkaitan dengan penelitian. Data yang dikumpulkan kemudian diolah untuk

mendukung proses penelitian ini. Prosedur pengembangan sistem

2. Daftar Kebutuhan Pengguna

Tabel 3.2 Daftar kebutuhan pengguna

No SRS	Kebutuhan	<i>Use case</i>
SRS_01	Perangkat lunak merespon <i>register</i> sebagai <i>user</i>	<i>Register</i> sebagai <i>user</i>
SRS_02	Perangkat lunak harus mampu <i>login</i> sebagai <i>user</i>	<i>Login</i> sebagai <i>user</i>
SRS_03	Mengubah data diri <i>user</i> dengan <i>edit profile</i>	<i>Edit Profile Lost and found</i> sebagai <i>user</i>
SRS_04	Melakukan laporan atau pengumuman mengenai kehilangan atau penemuan benda	<i>Posting Lost and found</i> sebagai <i>user</i>
SRS_05	Mengubah data dari barang dan status barang yang telah di <i>posting</i> <i>user</i>	<i>Edit Lost and found</i> sebagai <i>user</i>
SRS_06	Menghapus data mengenai barang yang telah di <i>posting</i> <i>user</i>	<i>Delete Lost and found</i> sebagai <i>user</i>
SRS_07	Mengirimkan pesan <i>FAQ</i> kepada <i>admin</i> ketika <i>user</i> ada kendala	Mengirimkan pesan kepada <i>admin</i> sebagai <i>user</i>

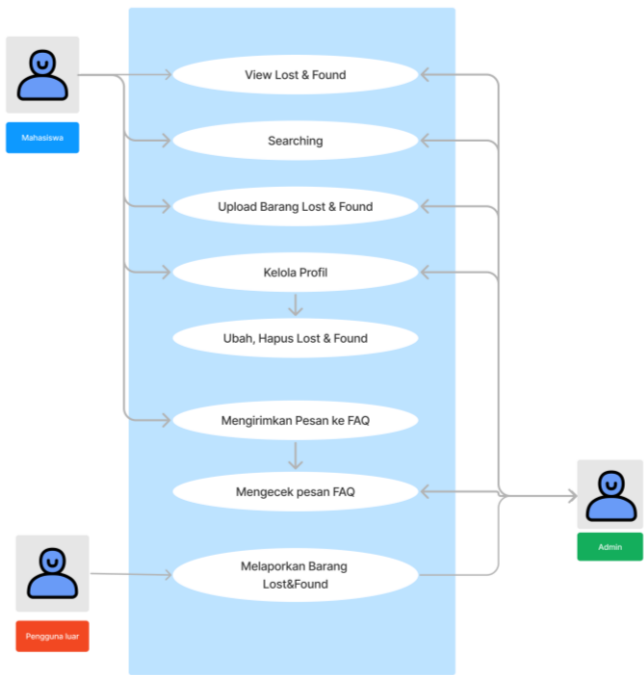
SRS_08	Perangkat lunak harus mampu <i>login</i> sebagai <i>user</i>	<i>Login</i> sebagai <i>user</i>
SRS_09	Mengubah data diri <i>user</i> dengan <i>edit profile</i>	<i>Edit Profile Lost and found</i> sebagai <i>user</i>
SRS_10	Melakukan laporan atau pengumuman mengenai kehilangan atau penemuan benda	<i>Posting Lost and found</i> sebagai <i>user</i>
SRS_11	Mengubah data dari barang dan status barang yang telah di <i>posting</i> <i>user</i>	<i>Edit Lost and found</i> sebagai <i>user</i>
SRS_12	Menghapus data mengenai barang yang telah di <i>posting</i> <i>user</i>	<i>Delete Lost and found</i> sebagai <i>user</i>
SRS_14	Melihat list keseluruhan postingan barang yang di <i>upload</i> oleh semua <i>user</i>	<i>View Posting Lost and found</i>

B. *Build a feature list* (Membangun daftar list fitur)

Setelah merancang model keseluruhan, tahap berikutnya adalah menyusun daftar fitur yang akan dibangun. Proses ini dimulai dengan merancang kebutuhan fungsional menggunakan *Use case Diagram*

Use case diagram digunakan untuk merepresentasikan perilaku sistem, di mana terdapat beberapa aktor, seperti

user, pengguna luar dan *admin*, yang memiliki peran serta aktivitas yang berbeda. Diagram ini memvisualisasikan fungsionalitas perangkat lunak Lost and Found. Selain itu, skenario sistem juga disediakan untuk menggambarkan secara *detail* aktivitas dari setiap aktor, yang disusun dalam bentuk tabel. Penjelasan mengenai *use case* diagram dapat ditemukan pada Gambar 3.2 berikut.



Gambar 3.2 *Use case diagram*

Setelahnya setelah *use case* dan daftar kebutuhan pengguna telah di rancang maka dibuatlah dua skenario dengan skenario kasus pada sistem dan skenario utama. Skenario kasus pada sistem ini akan menjelaskan mengenai deskripsi suatu *use case* berdasarkan tujuan yang akan dilakukan pengguna pada saat melakukan suatu *activity* atau kegiatan sedangkan skenario utama merupakan suatu langkah-langkah yang akan dilakukan oleh aktor.

1. Skenario *Use case Register* sebagai *user*

Tabel 3.3 Skenario kasus pada sistem *register user*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_01
Nama	<i>Register</i> sebagai <i>user</i>
Tujuan	Menambahkan akun sebagai <i>user</i>
Deskripsi	Setelah melakukan <i>Register</i> menggunakan nama, email (sesuai dengan milik kampus), dan password, maka <i>user</i> dapat memiliki hak akses untuk akun baru pada <i>website Lost and found</i>
Aktor	<i>User</i>

Tabel 3.4 Skenario utama *register user*

Skenario Utama	
Kondisi awal	<i>User</i> memasuki halaman <i>registrasi</i> akun dari <i>Lost and found</i>

Aksi Aktor	<i>User</i> memasukkan informasi data diri untuk menambahkan akun dengan menggunakan nama, email (sesuai milik kampus), dan password.
Reaksi Sistem	Sistem melakukan pemrosesan data agar dapat disimpan pada database sesuai dengan apa yang <i>user input</i> .
Kondisi Akhir	Akun untuk <i>website Lost and found user</i> berhasil di daftarkan atau <i>registrasi</i> .

2. Skenario *Use case Login* sebagai *user*

Tabel 3.5 Skenario kasus pada sistem *login user*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_02
Nama	<i>Login sebagai user</i>
Tujuan	<i>Login ke halaman utama Lost and found sebagai user</i>
Deskripsi	Setelah melakukan <i>login</i> menggunakan <i>email</i> (sesuai dengan milik kampus), dan <i>password</i> , maka <i>user</i> dapat memiliki hak akses untuk memasuki halaman <i>Lost and found</i>
Aktor	<i>User</i>

Tabel 3.6 Skenario Utama sistem *login user*

Skenario Utama	
Kondisi awal	<i>User login ke halaman dari Lost and found</i>

Aksi Aktor	<i>User</i> memasukkan informasi data diri untuk dapat masuk ke halaman <i>Lost and found</i> dengan menggunakan <i>email</i> (sesuai milik kampus), dan <i>password</i> .
Reaksi Sistem	Sistem akan memproses data dari database dan sistem selanjutnya melakukan pengecekan terhadap <i>field</i> yang diisi pada laman <i>login</i> , apakah sudah sesuai atau belum. Jika telah sesuai dengan data diri maka <i>user</i> akan dipindahkan ke laman utama <i>Lost and found</i>
Kondisi Akhir	Menampilkan <i>interface</i> halaman utama <i>Lost and found</i>

3. Skenario *Use case* terhadap *edit profile Lost and found* sebagai *user*

Tabel 3.7 Skenario kasus pada sistem *edit profile user*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_03
Nama	<i>Edit Profile Lost and found</i> sebagai <i>user</i>
Tujuan	Mengubah data diri <i>user</i> berdasarkan <i>field</i> yang berada pada <i>Edit Profile</i> berupa Tanggal Lahir, Nomor Telepon
Deskripsi	Setelah melakukan <i>login</i> , <i>user</i> juga dapat mengubah <i>profile user</i> mengenai data diri yang sebelumnya diisi
Aktor	<i>User</i>

Tabel 3.8 Skenario utama *edit profile user*

Skenario Utama	
Kondisi awal	<i>User berada di halaman profile</i>
Aksi Aktor	<i>User dapat melakukan edit pada bagian profile mengenai form data diri dan juga dapat merubah kata sandi user</i>
Reaksi Sistem	Sistem akan memproses data diri dari database lalu sistem dapat melakukan pengeditan <i>profile</i> yang dilakukan oleh <i>user</i> . Setelah pengeditan selesai <i>user</i> akan diarahkan kembali pada <i>profile user</i>
Kondisi Akhir	Menampilkan informasi data diri yang telah di <i>edit pada profile</i>

4. Skenario *Use case* terhadap *posting Lost and found* sebagai *user*

Tabel 3.9 Skenario kasus pada sistem *posting user*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_04
Nama	<i>Posting Lost and found</i> sebagai <i>user</i>
Tujuan	Melakukan laporan atau pengumuman mengenai kehilangan atau penemuan benda
Deskripsi	Setelah <i>login</i> dengan data diri yang sesuai, maka <i>user</i> dapat langsung melakukan unggahan

	mengenai laporan atau pengumuman kehilangan atau penemuan benda, dan bisa langsung dilihat pada halaman <i>view Lost and found</i> .
Aktor	<i>User</i>

Tabel 3.10 Skenario utama *posting user*

Skenario Utama	
Kondisi awal	<i>Login sebagai user dan masuk ke page view Lost and found</i>
Aksi Aktor	<i>User menekan tombol Post untuk melakukan laporan mengenai barang, lalu melakukan pilihan apakah merupakan kategori barang hilang atau barang ditemukan, setelahnya user dapat melakukan pengisian form seperti, foto barang, nama barang, deskripsi barang, kategori barang, warna barang, waktu barang, tanggal barang, dan lokasi dari barang berdasarkan pilihan kehilangan atau penemuan barang.</i>
Reaksi Sistem	<i>Sistem akan memproses data dari pengisian form Lost and found dan sistem selanjutnya akan melakukan pengecekan apakah sudah terisi lengkap atau belum, jika lengkap maka akan tersimpan dalam database dan akan ditampilkan dalam page view Lost and found.</i>
Kondisi Akhir	<i>Menampilkan informasi barang yang di unggah pada page view Lost and found</i>

5. Skenario *Use case* terhadap *edit Lost and found* sebagai *user*

Tabel 3.11 Skenario kasus pada sistem *edit user*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_05
Nama	<i>Edit Lost and found</i> sebagai <i>user</i>
Tujuan	Mengubah data dari benda yang telah di <i>posting user</i> , selain itu juga dapat digunakan untuk mengubah <i>status</i> benda yang di unggah
Deskripsi	Setelah melakukan <i>posting</i> , <i>user</i> dapat melakukan <i>edit</i> terhadap postingan yang telah dibuat, selain itu juga dapat melakukan pengubahan <i>status</i> postingan tersebut dari “belum dikembalikan atau <i>ditemukan</i> ” menjadi “telah dikembalikan atau <i>ditemukan</i> ” sesuai dengan kategori postingan yang <i>user</i> buat
Aktor	<i>User</i>

Tabel 3.12 Skenario utama *edit user*

Skenario Utama	
Kondisi awal	<i>User</i> berada di halaman <i>profile</i>
Aksi Aktor	<i>User</i> dapat melakukan <i>edit</i> pada bagian <i>history upload</i> mengenai form tentang postingan <i>Lost</i>

	<i>and found</i> dan juga dapat merubah status mengenai <i>postingan</i> tersebut.
Reaksi Sistem	Sistem akan memproses data dari database lalu sistem dapat melakukan <i>pengeditan</i> yang dilakukan oleh <i>user</i> . Setelah <i>pengeditan</i> selesai dan <i>user</i> juga dapat merubah <i>status</i> dari <i>postingan</i> tersebut
Kondisi Akhir	Menampilkan informasi barang yang telah di <i>edit</i> pada <i>history upload</i> pada <i>profile</i>

6. Skenario *Use case* terhadap *delete Lost and found* sebagai *user*

Tabel 3.13 Skenario kasus pada sistem *delete user*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_06
Nama	<i>Delete Lost and found</i> sebagai <i>user</i>
Tujuan	Menghapus data mengenai barang yang telah di <i>posting user</i>
Deskripsi	Setelah melakukan <i>posting</i> , <i>user</i> dapat menghapus mengenai benda yang telah di <i>posting</i> . Maksud dari <i>delete posting</i> disini adalah menghapus mengenai <i>postingan</i> yang telah dianggap selesai.
Aktor	<i>User</i>

Tabel 3.14 Skenario utama *delete user*

Skenario Utama	
Kondisi awal	<i>User berada di halaman profile</i>
Aksi Aktor	<i>User dapat melakukan delete pada bagian history upload mengenai postingan Lost and found untuk menghapus postingan yang telah dianggap selesai</i>
Reaksi Sistem	Sistem akan memproses data dari database, untuk melakukan penghapusan dari permintaan <i>user</i> mengenai postingan yang sebelumnya sudah ter <i>upload</i> ataupun telah dianggap selesai
Kondisi Akhir	Menampilkan notifikasi barang telah dihapus pada <i>history upload</i> pada <i>profile</i>

7. Skenario *Use case Login* sebagai *admin*

Tabel 3.15 Skenario kasus pada sistem *login admin*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_08
Nama	<i>Login sebagai admin</i>
Tujuan	<i>Login ke halaman utama Lost and found sebagai admin</i>
Deskripsi	Setelah melakukan <i>login</i> menggunakan <i>email</i> (sesuai dengan milik kampus), dan <i>password</i> ,

	maka <i>admin</i> dapat memiliki hak akses untuk memasuki halaman <i>Lost and found</i>
Aktor	<i>Admin</i>

Tabel 3.16 Skenario utama *login admin*

Skenario Utama	
Kondisi awal	<i>Admin login</i> ke halaman dari <i>Lost and found</i>
Aksi Aktor	<i>Admin</i> memasukkan informasi data diri untuk dapat masuk ke halaman <i>Lost and found</i> dengan menggunakan <i>email</i> (sesuai milik kampus), dan <i>password</i> .
Reaksi Sistem	Sistem akan memproses data dari database dan sistem selanjutnya melakukan pengecekan terhadap <i>field</i> yang diisi pada laman <i>login</i> , apakah sudah sesuai atau belum. Jika telah sesuai dengan data diri maka <i>admin</i> akan dipindahkan ke laman utama <i>Lost and found</i>
Kondisi Akhir	Menampilkan <i>interface</i> halaman utama <i>Lost and found</i>

8. Skenario *Use case* terhadap *edit profile Lost and found* sebagai *admin*

Tabel 3. 17 Skenario kasus pada sistem *edit profile admin*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_09

Nama	<i>Edit Profile Lost and found sebagai admin</i>
Tujuan	Mengubah data diri <i>admin</i> berdasarkan <i>field</i> yang berada pada <i>Edit Profile</i> berupa Tanggal Lahir, Nomor Telepon
Deskripsi	Setelah melakukan <i>login</i> , <i>admin</i> juga dapat mengubah <i>profile admin</i> mengenai data diri yang sebelumnya diisi
Aktor	<i>Admin</i>

Tabel 3.18 Skenario utama *edit profile admin*

Skenario Utama	
Kondisi awal	<i>Admin</i> berada di halaman <i>profile</i>
Aksi Aktor	<i>Admin</i> dapat melakukan <i>edit</i> pada bagian <i>profile</i> mengenai form data diri dan juga dapat merubah kata sandi <i>admin</i>
Reaksi Sistem	Sistem akan memproses data diri dari database lalu sistem dapat melakukan pengeditan <i>profile</i> yang dilakukan oleh <i>admin</i> . Setelah pengeditan selesai <i>admin</i> akan diarahkan kembali pada <i>profile admin</i>
Kondisi Akhir	Menampilkan informasi data diri yang telah di <i>edit</i> pada <i>profile</i>

9. Skenario *Use case* terhadap *posting Lost and found* sebagai *admin*

Tabel 3.19 Skenario kasus pada sistem *posting admin*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_10
Nama	<i>Posting Lost and found</i> sebagai <i>admin</i>
Tujuan	Melakukan laporan atau pengumuman mengenai kehilangan atau penemuan benda
Deskripsi	Setelah <i>login</i> dengan data diri yang sesuai, maka <i>admin</i> dapat langsung melakukan unggahan mengenai laporan atau pengumuman kehilangan atau penemuan benda, dan bisa langsung dilihat pada halaman <i>view Lost and found</i> .
Aktor	<i>Admin</i>

Tabel 3.20 Skenario utama *posting admin*

Skenario Utama	
Kondisi awal	<i>Login</i> sebagai <i>admin</i> dan masuk ke <i>page view Lost and found</i>
Aksi Aktor	<i>Admin</i> menekan tombol <i>Post</i> untuk melakukan laporan mengenai barang, lalu melakukan pilihan apakah merupakan kategori barang hilang atau barang <i>ditemukan</i> , setelahnya <i>admin</i> dapat melakukan pengisian <i>form</i> seperti, foto barang, nama barang, deskripsi barang, kategori barang, warna barang, waktu barang, tanggal

	barang, dan lokasi dari barang berdasarkan pilihan kehilangan atau penemuan barang.
Reaksi Sistem	Sistem akan memproses data dari pengisian <i>form Lost and found</i> dan sistem selanjutnya akan melakukan pengecekan apakah sudah terisi lengkap atau belum, jika lengkap maka akan tersimpan dalam <i>database</i> dan akan ditampilkan dalam <i>page view Lost and found</i> .
Kondisi Akhir	Menampilkan informasi barang yang di unggah pada <i>page view Lost and found</i>

10. Skenario *Use case* terhadap *edit Lost and found* sebagai *admin*

Tabel 3.21 Skenario kasus pada sistem *edit admin*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_11
Nama	<i>Edit Lost and found</i> sebagai <i>admin</i>
Tujuan	Mengubah data dari benda yang telah di <i>posting</i> seluruh <i>user</i> , selain itu juga dapat digunakan untuk mengubah <i>status</i> benda yang di unggah oleh seluruh <i>user</i>
Deskripsi	Setelah suatu <i>user</i> melakukan <i>posting</i> , <i>admin</i> dapat melakukan <i>edit</i> terhadap postingan yang telah dibuat oleh seluruh <i>user</i> , selain itu juga dapat melakukan pengubahan <i>status</i> postingan tersebut dari “belum dikembalikan atau

	<i>ditemukan</i> ” menjadi “telah dikembalikan atau <i>ditemukan</i> ” sesuai dengan kategori postingan yang seluruh <i>user</i> buat
Aktor	<i>Admin</i>

Tabel 3.22 Skenario utama *edit admin*

Skenario Utama	
Kondisi awal	<i>Admin</i> berada di halaman <i>profile</i>
Aksi Aktor	<i>Admin</i> dapat melakukan <i>edit</i> pada bagian <i>post Lost and found</i> di <i>dashboard</i> mengenai form tentang postingan <i>Lost and found</i> yang dibuat oleh seluruh <i>user</i> dan juga dapat merubah status mengenai <i>postingan</i> tersebut.
Reaksi Sistem	Sistem akan memproses data dari database lalu sistem dapat melakukan <i>pengeditan</i> yang dilakukan oleh <i>admin</i> . Setelah <i>pengeditan</i> selesai dan <i>admin</i> juga dapat merubah <i>status</i> dari <i>postingan</i> tersebut yang dibuat oleh seluruh <i>user</i>
Kondisi Akhir	Menampilkan informasi barang yang telah di <i>edit</i> pada <i>post Lost and found</i> pada <i>Dashboard</i>

11. Skenario *Use case* terhadap *delete Lost and found* sebagai *admin*

Tabel 3.23 Skenario kasus pada sistem *delete admin*

Skenario Kasus pada Sistem

No Use case	SRS_12
Nama	<i>Delete Lost and found sebagai admin</i>
Tujuan	Menghapus data mengenai barang yang telah di
Deskripsi	Setelah suatu <i>user</i> melakukan <i>posting</i> , <i>admin</i> dapat menghapus mengenai benda yang telah <i>diposting</i> . Maksud dari <i>delete posting</i> disini adalah menghapus mengenai postingan yang telah dianggap selesai.
Aktor	<i>Admin</i>

Tabel 3.24 Skenario utama *delete admin*

Skenario Utama	
Kondisi awal	<i>Admin</i> berada di halaman <i>profile</i>
Aksi Aktor	<i>Admin</i> dapat melakukan <i>delete</i> pada bagian <i>post Lost and found</i> di <i>dashboard</i> mengenai postingan <i>Lost and found</i> milik seluruh <i>user</i> untuk menghapus postingan yang telah dianggap selesai
Reaksi Sistem	Sistem akan memproses data dari database, untuk melakukan penghapusan dari permintaan <i>admin</i> mengenai postingan seluruh <i>user</i> yang sebelumnya sudah ter <i>upload</i> ataupun telah dianggap selesai
Kondisi Akhir	Menampilkan notifikasi barang telah dihapus pada <i>post Lost and found</i> pada <i>dashboard</i>

12. Skenario *Use case* terhadap *view Lost and found*

Tabel 3.25 Skenario kasus pada sistem *view*

Skenario Kasus pada Sistem	
No <i>Use case</i>	SRS_14
Nama	<i>View Posting Lost and found</i>
Tujuan	Melihat list keseluruhan postingan barang yang di <i>upload</i> oleh semua <i>user</i>
Deskripsi	Setelah melakukan <i>login</i> , seluruh <i>user</i> dapat mengetahui list dari barang yang di <i>upload</i> oleh seluruh <i>user</i> pada <i>page view Lost and found</i> , sehingga <i>user</i> dapat mencari ataupun membantu menginformasikan barang yang terdaftar kepada pemilik yang tepat
Aktor	<i>User, Viewer, dan Admin</i>

Tabel 3.26 Skenario utama *view*

Skenario Utama	
Kondisi awal	<i>Login sebagai user dan masuk ke page view Lost and found</i>
Aksi Aktor	Seluruh <i>user website Lost and found</i> disini akan masuk ke <i>page view Lost and found</i> untuk dapat melihat seluruh mengenai postingan yang di <i>upload</i> oleh seluruh <i>user</i>

Reaksi Sistem	Sistem akan memproses data dari database keseluruhan barang <i>Lost and found</i> dan sistem selanjutnya akan memunculkan tampilan barang sesuai yang tersedia dalam <i>database</i>
Kondisi Akhir	Menampilkan informasi barang yang di unggah seluruh <i>user</i> dengan 4 info yang ditampilkan beserta gambar pada <i>page view Lost and found</i> , untuk dapat melihat keseluruhan informasi barang <i>user</i> dapat menekan foto barang untuk menampilkan keseluruhan informasi barang

C. *Plan by feature* (Perencanaan berdasarkan fitur)

Pada tahap selanjutnya, akan dilakukan perencanaan dalam pengembangan sistem informasi *Lost and found* secara terstruktur dengan membuat susunan berdasarkan daftar fitur yang telah di buat, dengan setiap iterasi fokus terhadap pengembangan dan penyelesaian satu atau lebih dari 1 fitur (Kusnadi et al., 2024). Pembuatan jadwal perancangan ini mengutamakan berdasarkan kebutuhan dan pentingnya setiap fitur yang akan dirancang beserta sumber daya yang tersedia. Pendekatan iteratif ini membantu sistem agar dapat dikembangkan secara bertahap dan fitur fitur penting dapat segera diimplementasikan pada *website* dan diuji sebelum keseluruhan sistem dapat selesai dibangun. Hal ini

membantu dalam melakukan minimalis resiko dan memungkinkan penyesuaian cepat ketika ada perubahan kebutuhan pada pengguna. Rincian tentang jadwal pengembangan dalam sistem informasi *Lost and found* yang dibangun dapat dilihat pada Tabel 3.26 berikut ini.

Tabel 3.27 Jadwal pengembangan sistem informasi *Lost and found*

No.	Tahapan FDD	Iterasi	Deskripsi Kegiatan
1	<i>Develop an Overall Model</i>	Iterasi 1	Pengumpulan Data dan Identifikasi masalah
		Iterasi 1	Pengembangan model sistem secara keseluruhan dengan Menyusun perencanaan dalam fungsi sistem
2	<i>Build a Features List</i>	Iterasi 1	Desain <i>use case</i> diagram dengan skenario untuk menggambarkan interaksi pengguna terhadap sistem
		Iterasi 1	Menata daftar fitur dengan melakukan perancangan skenario utama dan skenario pada sistem berdasarkan <i>use case</i>

3	<i>Plan by Feature</i>	Iterasi 2	Menyusun prioritas dalam pengembangan fitur dari kebutuhan pengguna
		Iterasi 2	Pembagian fitur ke dalam beberapa iterasi pengembangan
4	<i>Design by Feature</i>	Iterasi 2	Perancangan model alur menggunakan <i>Activity Diagram</i> serta <i>Sequence Diagram</i> untuk membantu proses implementasi
		Iterasi 2	Perancangan <i>Class Diagram</i> untuk merepresentasikan susunan sistem dengan menampilkan kelas-kelas beserta relasi di antaranya.
5	<i>Build by Feature</i>	Iterasi 3	Perancangan tampilan kasar atau wireframe untuk menggambarkan <i>website</i>
		Iterasi 3	Memulai pengkodean untuk tampilan awal <i>website</i> (front-end)

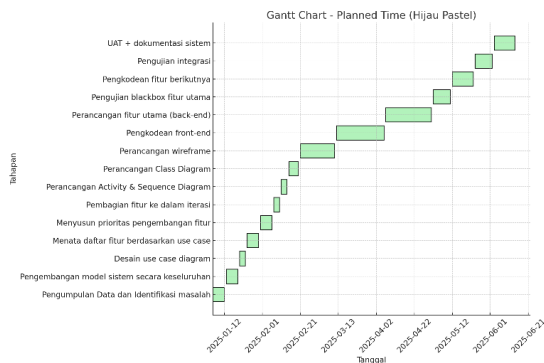
Iterasi 3	Melakukan perancangan fitur utama <i>website</i>
Iterasi 3	Melakukan pengujian blackbox terhadap fitur utama
Iterasi 4	Melanjutkan pengkodean fitur berikutnya
Iterasi 4	Melakukan pengujian integrasi dalam memastikan fitur berjalan dengan baik
Iterasi 4	Melakukan pengujian menggunakan User Acceptance Test (UAT) serta membuat dokumentasi sistem untuk persiapan peluncuran

Berikut adalah tampilan visualisasi gantt chart untuk menampilkan perencanaan waktu perancangan,

Tabel 3.28 Gantt Chart

No	Tahapan	Start Date (Planned)	End Date (Planned)	Duration (Planned)
Develop an overall model				
1	Pengumpulan Data dan Identifikasi masalah	06/Jan	12/Jan	7
2	Pengembangan model sistem secara keseluruhan	13/Jan	19/Jan	7
Build a features list				
3	Desain use case diagram	20/Jan	23/Jan	4
4	Menata daftar fitur berdasarkan use case	24/Jan	30/Jan	7
Plan by feature				
5	Menyusun prioritas pengembangan fitur	31/Jan	06/Feb	7
6	Pembagian fitur ke dalam iterasi	07/Feb	10/Feb	4
Design by feature				
7	Perancangan Activity & Sequence Diagram	11/Feb	14/Feb	4
8	Perancangan Class Diagram	15/Feb	20/Feb	6
Build by feature				
9	Perancangan wireframe	21/Feb	11/Mar	19
10	Pengkodean front-end	12/Mar	06/Apr	26
11	Perancangan fitur utama (back-end)	07/Apr	01/May	25
12	Pengujian blackbox fitur utama	02/May	11/May	10
13	Pengkodean fitur berikutnya	12/May	23/May	12
14	Pengujian integrasi	24/May	02/Jun	10
15	UAT + dokumentasi sistem	03/Jun	14/Jun	12

Selanjutnya akan dirancang bentuk tampilan secara grafik berdasarkan gantt chart berdasarkan tabel diatas.



Gambar 3.3 Gantt chart grafik

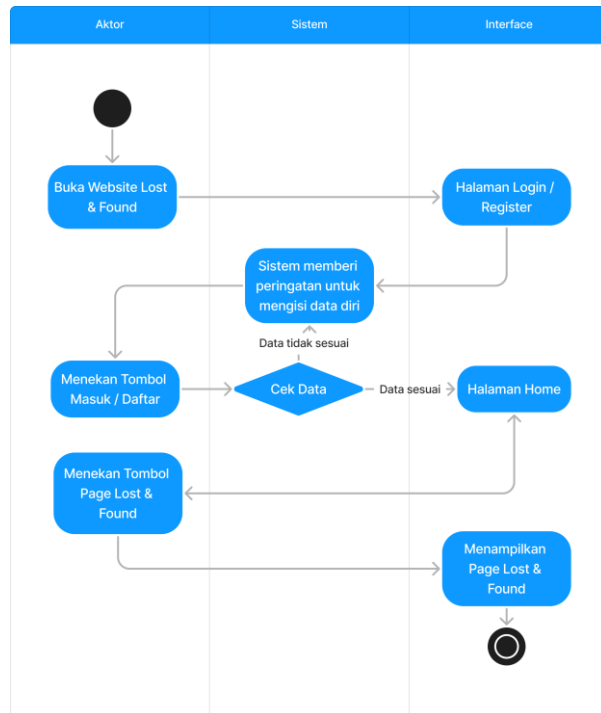
D. *Design by Feature* (Desain Fitur)

Tahap ini bertujuan untuk menggambarkan bagaimana fitur-fitur dalam sebuah website dirancang dan berfungsi, sehingga mempermudah proses perancangan lebih lanjut. Dengan menyajikan gambaran fitur secara terperinci, tahap ini membantu menjelaskan interaksi antara entitas dalam sistem serta bagaimana fitur-fitur tersebut akan diimplementasikan. Proses ini juga berperan penting dalam memastikan bahwa desain yang dihasilkan sesuai dengan kebutuhan pengguna dan mendukung tujuan utama dari sistem yang dikembangkan. Visualisasi dari hasil rancangan fitur berikut ditampilkan menggunakan activity diagram, sequence diagram dan class diagram.

1. *Activity Diagram*

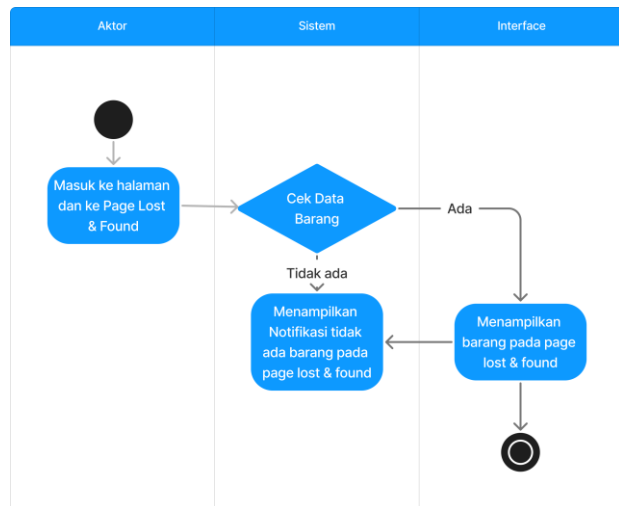
Activity diagram yang pertama ini menggambarkan alur aktivitas pengguna dalam mengakses dan menggunakan website *Lost and found*. Proses dimulai ketika pengguna membuka website, yang kemudian menampilkan halaman *login / register*. Pengguna menekan tombol masuk / daftar, lalu sistem melakukan Cek Data untuk memastikan apakah data pengguna sesuai. Jika data tidak sesuai, sistem memberikan peringatan agar pengguna mengisi data diri yang benar. Namun, jika data sesuai, pengguna diarahkan ke Halaman Dashboard. Selanjutnya,

pengguna dapat menekan tombol untuk mengakses *Page Lost and found*, dan sistem akan menampilkan halaman tersebut. Proses berakhir setelah halaman *Lost and found* berhasil ditampilkan. Diagram ini menggambarkan alur utama login dan navigasi pengguna dalam website, termasuk validasi data sebelum mengakses halaman utama.



Gambar 3.4 *Activity Diagram Login/Register User*

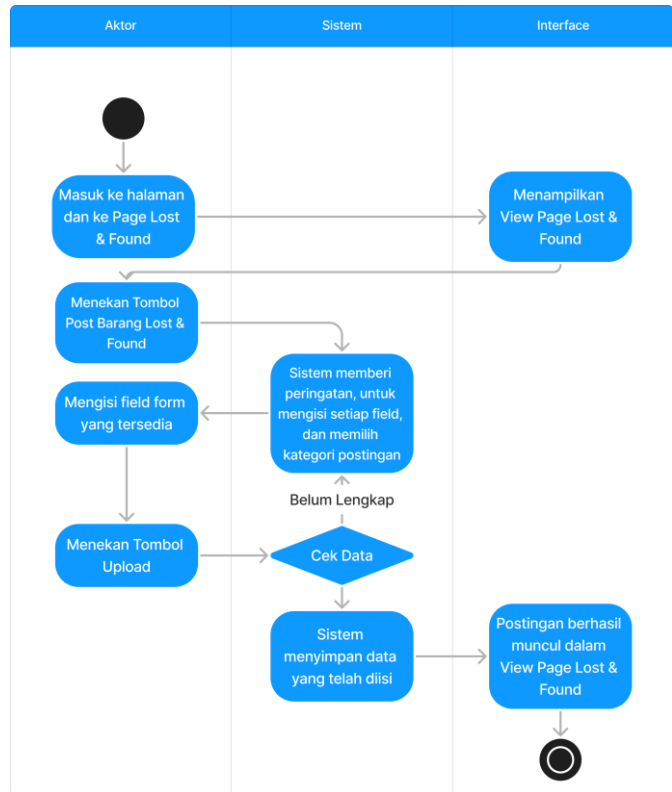
Activity diagram ini menggambarkan proses pengguna dalam mengakses dan melihat data barang pada halaman Lost and Found. Saat halaman dibuka, sistem akan memeriksa database. Jika tidak ada data barang, sistem menampilkan pemberitahuan bahwa belum ada barang tersedia. Sebaliknya, jika data tersedia, sistem langsung menampilkan daftar barang kepada pengguna.



Gambar 3.5 Activity Diagram View Page

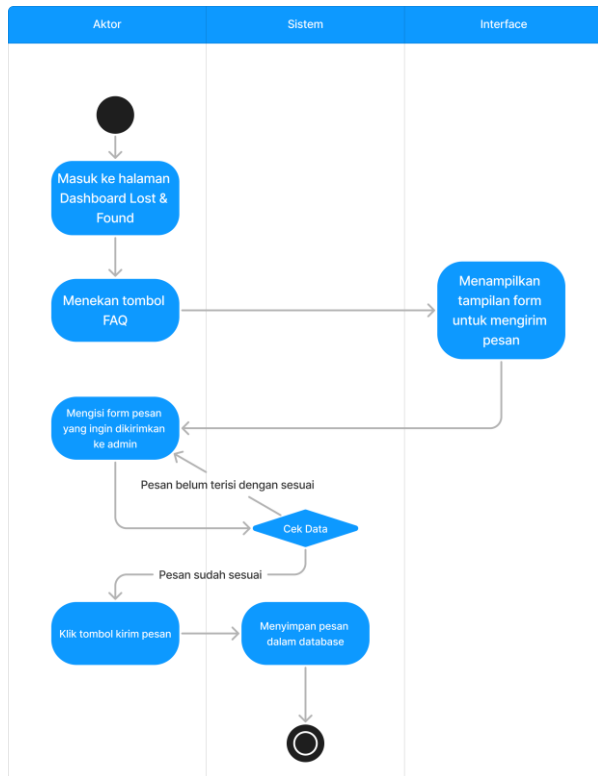
Activity diagram yang ketiga ini menunjukkan proses pengguna memposting barang hilang atau ditemukan di *Lost and found*. Pengguna masuk ke halaman, mengisi form, dan menekan Upload. Sistem

memeriksa kelengkapan data, memberi peringatan jika ada yang kurang, lalu menyimpan data jika sudah lengkap. Akhirnya, postingan tampil di halaman *Lost and found*.



Gambar 3.6 Activity Diagram Post Item

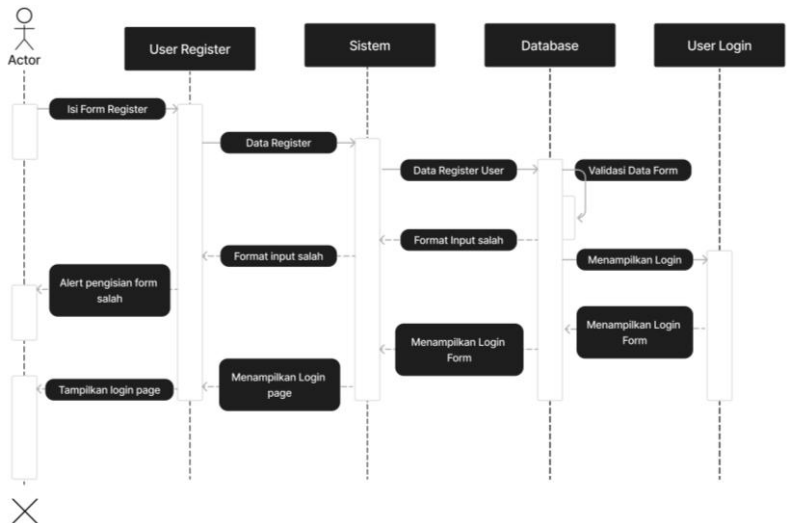
Activity diagram yang keempat ini menunjukkan alur pengguna dalam menghubungi admin melalui fitur *Lost and found*. Pengguna masuk ke Dashboard *Lost and found*, lalu menekan tombol FAQ, yang terdapat tampilan form untuk mengirim pesan. Setelah itu, pengguna mengisi form pesan yang ingin dikirim. sistem menyimpan pesan tersebut ke dalam database, menandakan bahwa proses telah selesai.



Gambar 3.7 Activity Diagram FAQ Admin

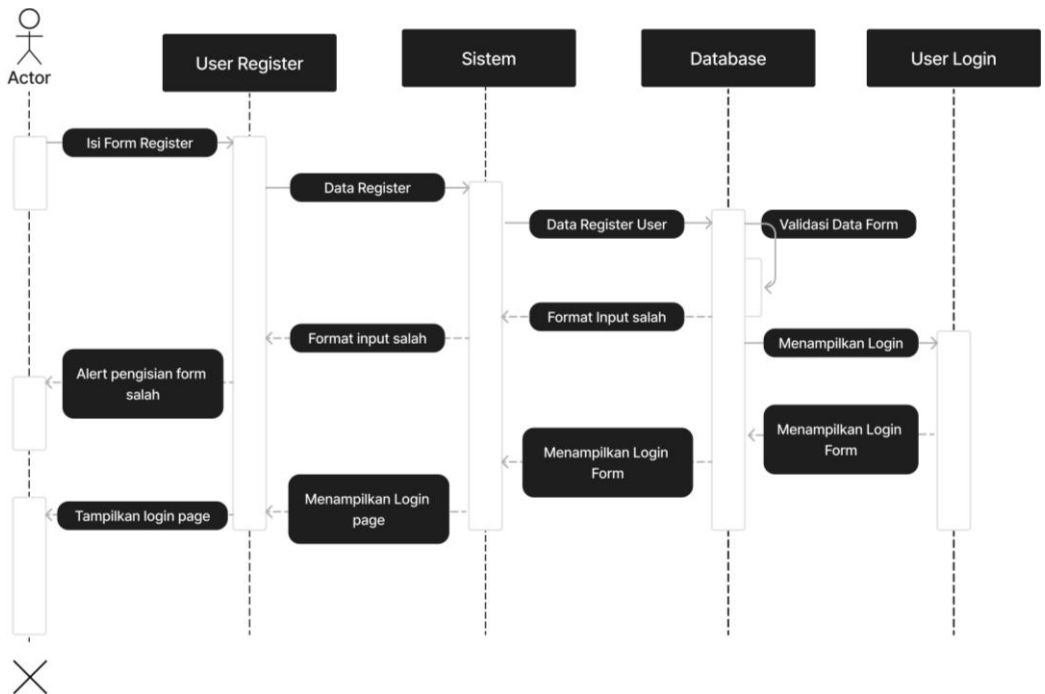
2. Sequence Diagram

Sequence Diagram yang pertama yaitu berupa menjelaskan tentang tata cara *registrasi* akun *user* seperti yang tercantum dalam gambar, akan dijelaskan langkah-langkah *user* dalam melakukan *registrasi* akun. Pada *Sequence diagram* tersebut *user* diharuskan untuk mengisi form yang tersedia dalam *register page*, form yang harus diisi *user* berupa email yang disediakan oleh universitas, password, nama asli, NIM (Nomor Induk Mahasiswa), dan nomor handphone.



Gambar 3.8 Sequence diagram register akun *user*

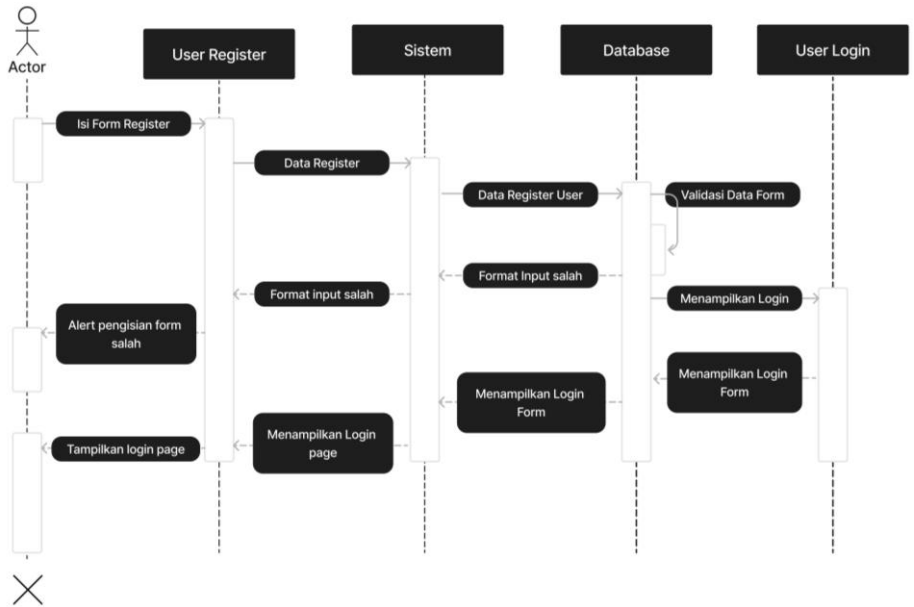
Selanjutnya pada sequence diagram yang kedua setelah *user* berhasil melakukan registrasi, maka *user* akan diarahkan pada *login page*, *user* diharuskan mengisi form yang tersedia berupa NIM (Nomor Induk Mahasiswa) dan juga password, seperti yang dijelaskan pada Gambar 3.5 berikut.



Gambar 3.9 Sequence diagram *login* akun *user*

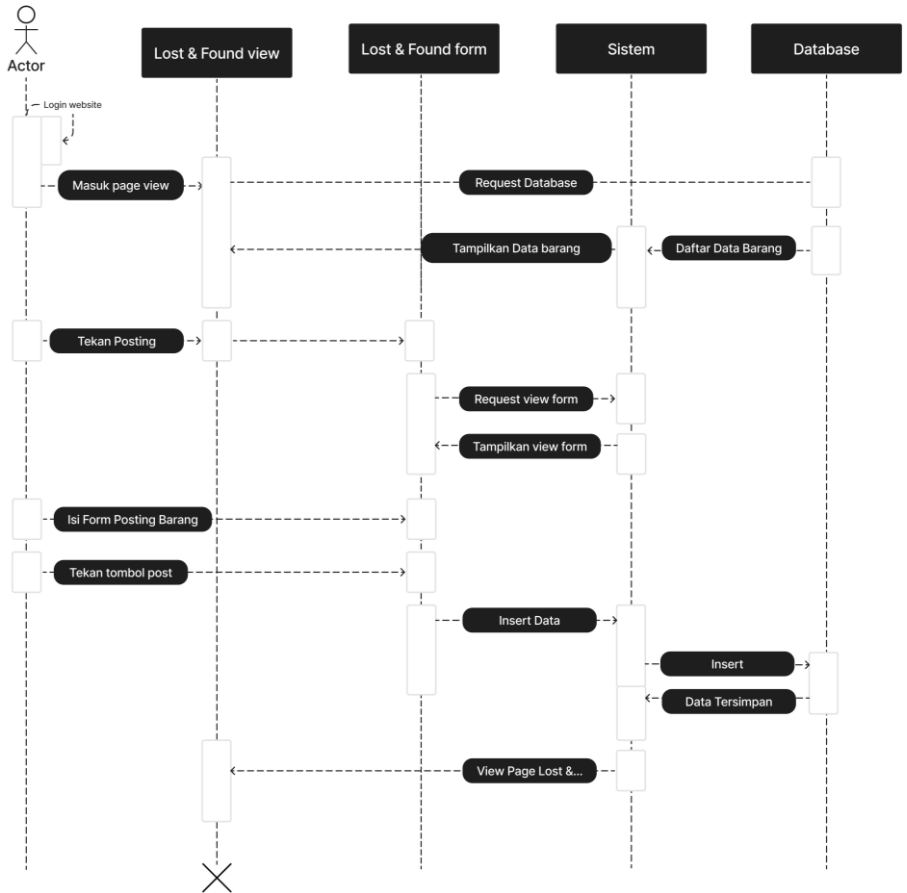
Sequence Diagram yang ketiga akan menggambarkan mengenai proses penggunaan *user* melihat informasi terkait barang yang *ditemukan* dan

barang yang kehilangan pada halaman *view Lost and found*, berikut lebih *detail* dijelaskan pada Gambar 3.6 berikut.



Gambar 3.10 Sequence diagram *view page*

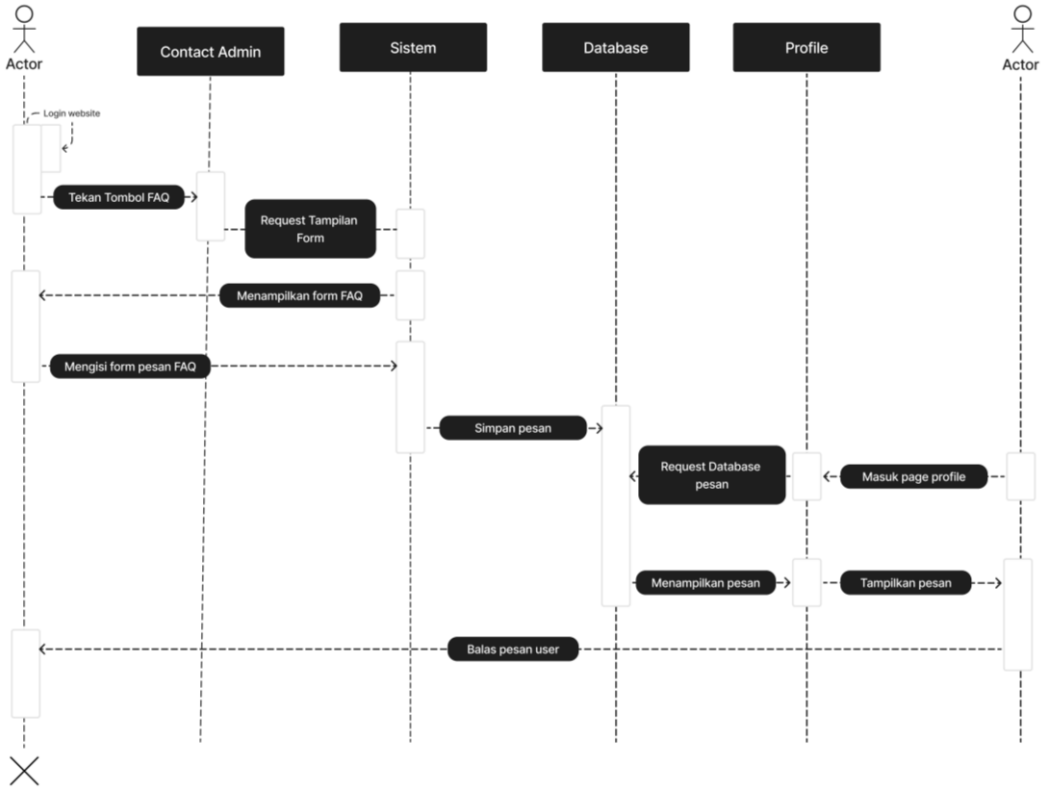
Sequence Diagram keempat akan menjelaskan mengenai informasi bagaimana cara *user* dapat melakukan *post* barang yang kehilangan ataupun barang yang *ditemukan*. Berikut lebih jelas mengenai langkah *user posting* barang pada Gambar 3.7 berikut.



Gambar 3.11 Sequence diagram *post item user*

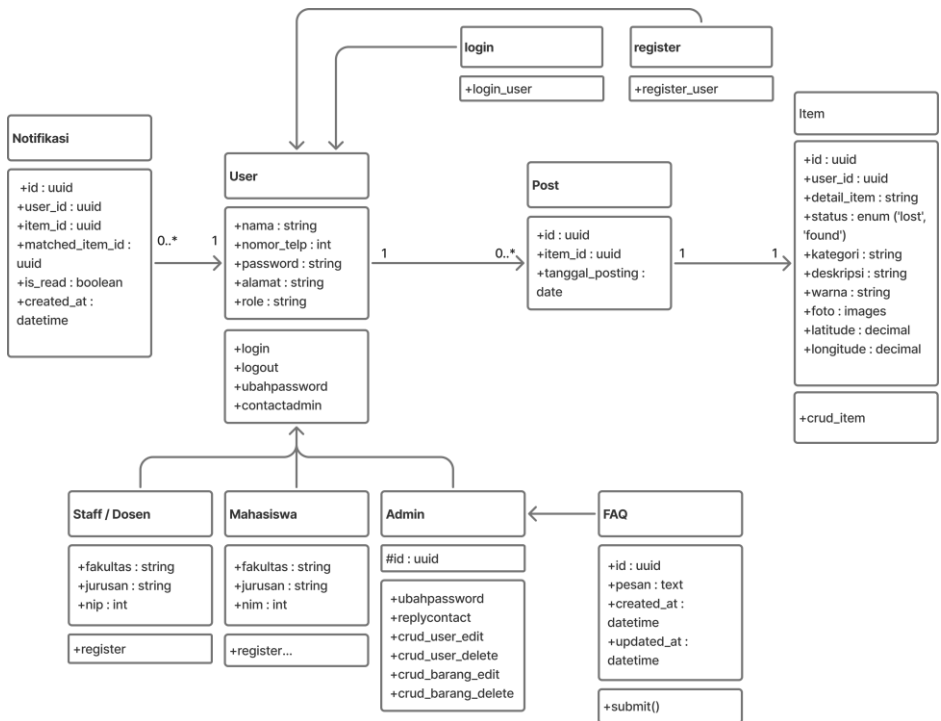
Sequence Diagram yang kelima berupa informasi bagaimana cara *user* membutuhkan bantuan ketika ada kendala dalam menggunakan *website* dan melaporkannya pada *admin* menggunakan fitur FAQ.

Penjelasan lebih *detail* terdapat pada Gambar 3.8 berikut



Gambar 3.12 Sequence diagram FAQ *user*

3. Class Diagram



Gambar 3.13 Class diagram Lost and found

E. Build by Feature (Membangun fitur)

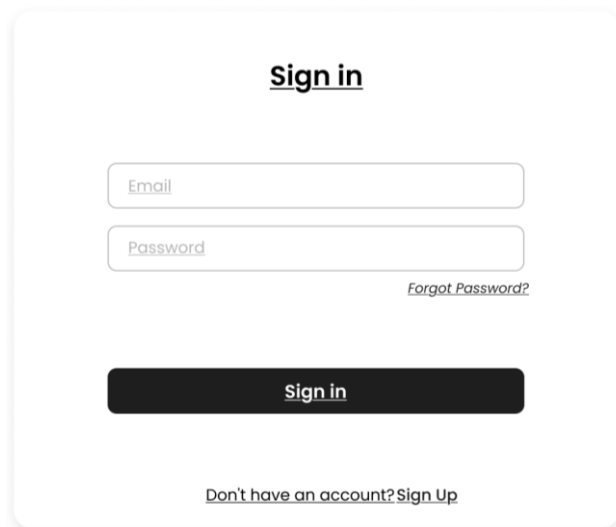
Pada Tahap akhir pengembangan metode *Feature Driven Development*, *Build by Feature*, kami merancang desain secara bertahap berdasarkan dari hasil fitur yang telah ditetapkan pada fase *Build a Features List*, dan selaras dengan rancangan yang dibuat pada fase *Design by Feature*, agar nantinya dapat diimplementasikan kepada

hasil akhir rancangan sistem informasi yang berbasis *website*.

1. Tampilan desain form *login*

Form *login* merupakan tampilan form yang muncul pertama saat *website* dijalankan, *user* diminta memasukkan NIM dan password yang valid untuk mengakses halaman *website*. Desain form *login* dapat dilihat pada Gambar 3.13.

LOGO



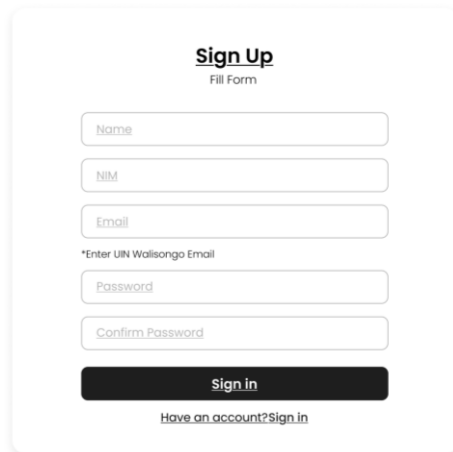
The image shows a user login form design. At the top, the text "Sign in" is displayed in a bold, black font. Below this, there are two input fields: the first is labeled "Email" and the second is labeled "Password". To the right of the "Password" field, there is a link that says "Forgot Password?". Below the input fields, there is a large, dark gray button with the text "Sign in" in white. At the bottom of the form, there is a link that says "Don't have an account? Sign Up".

Gambar 3.14 Desain form *login user*

2. Tampilan desain form register

Form *register* merupakan tampilan form yang muncul saat *user* ingin membuat akun baru di *website*. Pada form ini, *user* diminta untuk memasukkan Nama, NIM, Email Kampus, Password, dan Konfirmasi Password yang valid untuk dapat mendaftarkan akun dan mengakses halaman *website*. Jika *user* belum memiliki akun, mereka harus mengisi form registrasi ini terlebih dahulu sebelum dapat masuk dan menggunakan fitur yang tersedia. Desain form register dapat dilihat pada Gambar 3.14.

LOGO



The image shows a user registration form titled "Sign Up" with the subtitle "Fill Form". The form contains five input fields: "Name", "NIM", "Email", "Password", and "Confirm Password". Below the "Email" field, there is a note: "*Enter UIN Walisongo Email". At the bottom of the form is a black button labeled "Sign in". Below the button, there is a link: "Have an account? Sign in".

Gambar 3.15 Desain form *register user*

3. Tampilan desain *Dashboard user*

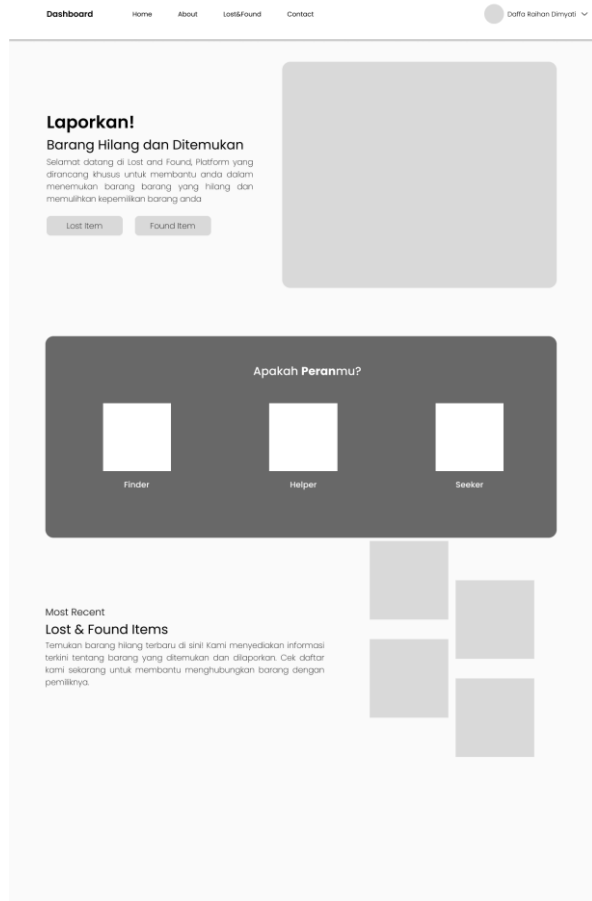
Tampilan *dashboard Lost and found* diawali dengan navigasi yang menampilkan logo dan nama *website*, diikuti oleh empat tombol utama: Home, About, *Lost and found* Page, dan FAQ untuk memudahkan akses ke berbagai halaman. Setelah itu, terdapat tampilan profil pengguna yang mencakup foto dan nama, serta dropdown menu untuk pengaturan akun.

Pada laman utama, pengguna disambut dengan teks dan gambar yang berfungsi sebagai peringatan sekaligus ajakan untuk segera melaporkan barang hilang atau ditemukan. Tersedia dua tombol utama, yaitu Lost Item dan Found Item, yang memudahkan pengguna untuk melakukan pelaporan sesuai kondisi.

Selanjutnya, dijelaskan pembagian peran pengguna, yaitu Helper (membantu menemukan pemilik barang), Seeker (mencari barang miliknya yang hilang), dan Finder (melaporkan barang yang ditemukan). Di bawahnya, ditampilkan daftar barang yang sering muncul atau paling banyak dicari berdasarkan kategori.

Pada bagian akhir halaman, terdapat footer berisi deskripsi singkat tentang platform, logo, menu pintasan (Instant Menu) untuk navigasi cepat, serta

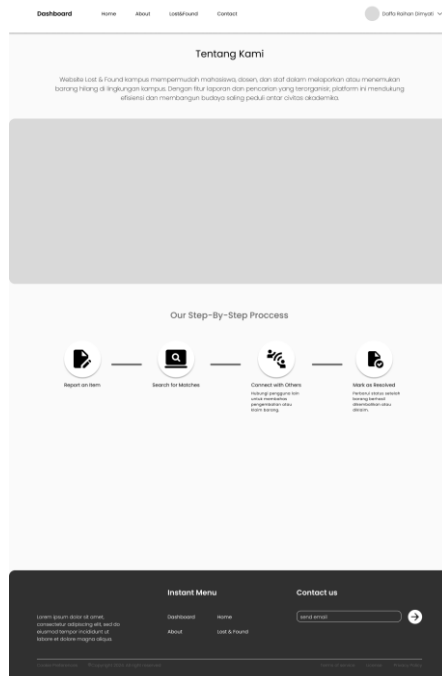
kontak admin yang dapat dihubungi untuk bantuan lebih lanjut. Seluruh elemen ini tersaji seperti ditampilkan pada Gambar 3.15.



Gambar 3.16 Desain halaman dashboard

4. Tampilan desain tentang *website*

Bagian "Tentang Kami" menjelaskan tujuan platform dalam mendukung pencarian barang hilang dan membangun kepedulian civitas akademika. Proses utama ditampilkan dalam empat langkah: melaporkan, mencari, menghubungi, dan memperbarui status barang, dengan koneksi antar pengguna dan deskripsi singkat untuk kemudahan pengguna.



Gambar 3.17 Desain halaman tentang *website*

5. Tampilan Desain Halaman *Lost and found*

Pada halaman *Lost and found* menampilkan daftar barang hilang dan ditemukan dalam platform *Lost and found* kampus. Bagian utama berisi daftar barang yang dilaporkan, ditampilkan dalam bentuk grid dengan informasi seperti kategori barang, nama barang, pemilik, dan tanggal upload *User*. Terdapat juga kolom pencarian serta filter berdasarkan tanggal terbaru dan kategori untuk mempermudah pencarian barang.

Desain ini berfokus pada kemudahan navigasi dengan menampilkan barang dalam format visual yang terorganisir dan mudah dipahami. Setiap barang disajikan dalam bentuk kartu atau grid yang rapi, dilengkapi dengan gambar, nama barang, lokasi, serta waktu pelaporan. Informasi ini dirangkum secara singkat untuk mempercepat proses identifikasi.

Desain ini mengutamakan fungsionalitas dan kenyamanan pengguna, serta tampil responsif di berbagai perangkat. Seluruh elemen tersebut disajikan seperti pada Gambar 3.17 berikut.

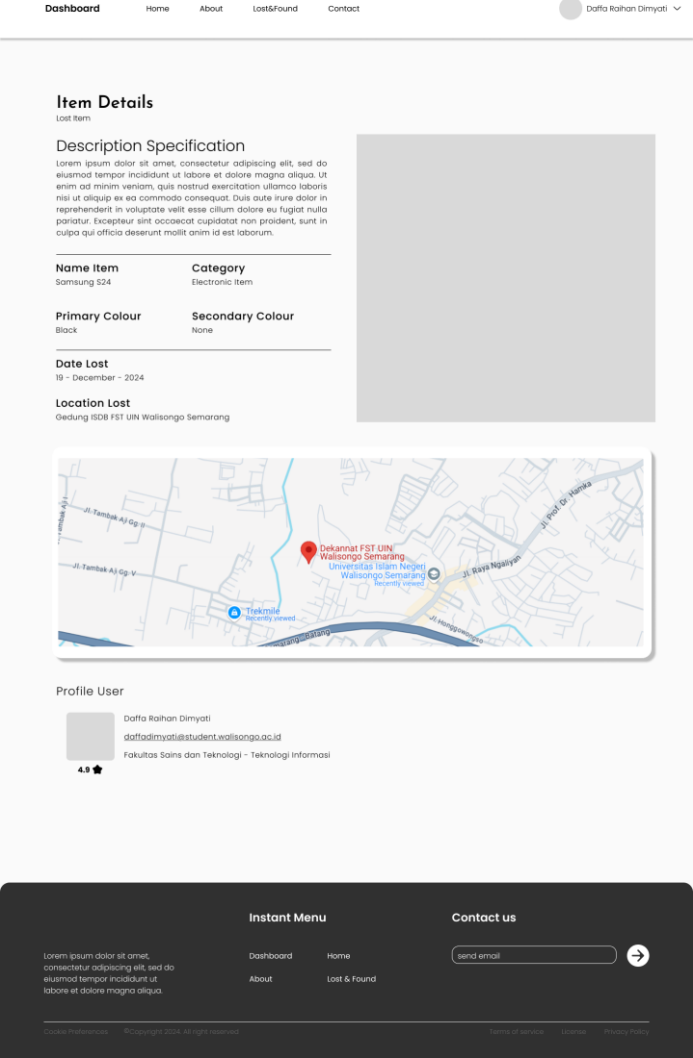


Gambar 3.18 Desain halaman list *item*

6. Tampilan desain detail item

Tampilan pada desain *detail item* menampilkan informasi *detail* mengenai barang yang hilang atau ditemukan. Bagian utama mencakup deskripsi spesifikasi, termasuk nama barang, kategori, warna utama dan sekunder, serta tanggal dan lokasi barang hilang. Desain ini mempermudah pengguna dalam memahami karakteristik barang, sehingga dapat membantu pencarian dan klaim barang dengan lebih akurat.

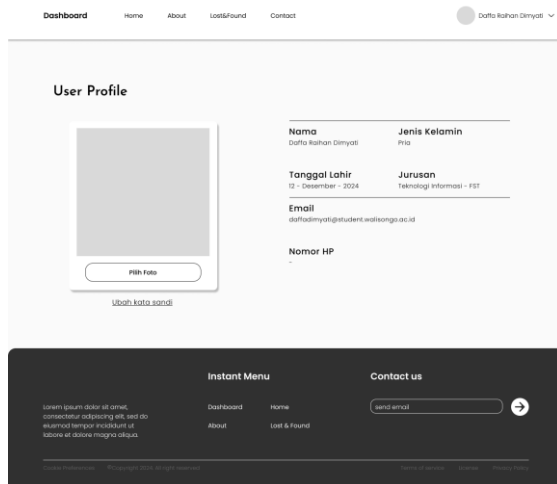
Selain itu, terdapat peta lokasi yang menggunakan fitur *Location Based Service* (LBS) untuk menunjukkan tempat barang terakhir *ditemukan*, serta informasi profil pengguna yang melaporkan barang tersebut. Dengan adanya *detail* kontak pengguna, platform ini menggunakan email yang tercantum oleh pengguna untuk mempermudah komunikasi antara pihak yang menemukan dan pemilik barang, sehingga meningkatkan peluang barang kembali ke pemiliknya. Seperti pada Gambar 3.16 berikut.



Gambar 3.19 Desain halaman *detail item*

7. Tampilan desain user profile

Tampilan pada desain *user profile* atau profil pengguna menampilkan informasi *detail* mengenai data pribadi pengguna. Bagian utama mencakup informasi dasar seperti nama lengkap, jenis kelamin, tanggal lahir, jurusan, serta email pengguna. Desain ini mempermudah pengguna untuk mengelola data pribadi mereka dengan lebih terstruktur dan mudah dipahami. Selain itu, terdapat fitur untuk mengunggah foto profil guna memperkuat identitas pengguna, serta opsi ubah kata sandi yang disediakan untuk meningkatkan keamanan akun. Seperti pada Gambar 3.17 berikut.



Gambar 3.20 Desain halaman *user profile*

BAB IV

HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil akhir dari proses pembangunan sistem yang telah dirancang sebelumnya. Pembahasan difokuskan pada implementasi perangkat lunak dan perangkat keras, perancangan serta penerapan basis data, dan antarmuka pengguna yang mendukung kinerja sistem secara menyeluruh. Selain itu, bab ini juga menjelaskan proses integrasi antar komponen sistem serta tahapan pengujian yang dilakukan untuk memastikan bahwa semua fitur berjalan sesuai dengan spesifikasi yang telah ditetapkan. Pengujian dilakukan untuk mengevaluasi performa sistem dan memastikan kesesuaian dengan kebutuhan pengguna, sekaligus menjadi dasar perbaikan atau pengembangan lebih lanjut.

A. Implementasi Perangkat Lunak

Untuk memastikan sistem yang dirancang dapat berfungsi dengan efektif dan sesuai arah yang diinginkan, sejumlah software pendukung diperlukan selama proses pengembangannya. Daftar software yang dimanfaatkan tercantum pada Tabel 4.1.

Tabel 4.1 Implementasi Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1	Integrated Development Environment (IDE)	Visual Studio Code

2	PHP, Web server, dan Database server	PHP 8.3.11, Laragon, MySQL
3	Web Frameworks	Laravel 11, Tailwind, Laravel Breeze
4	Sistem Operasi	Windows 10
5	Peta Interaktif	OpenStreetMap Leaflet
6	Hosting	Domainesia
7	Domain	Lofouin.my.id

B. Implementasi Perangkat Keras

Perangkat keras yang dimanfaatkan oleh peneliti untuk membangun sistem informasi *Lost and Found*, disajikan dalam Tabel 4.2.

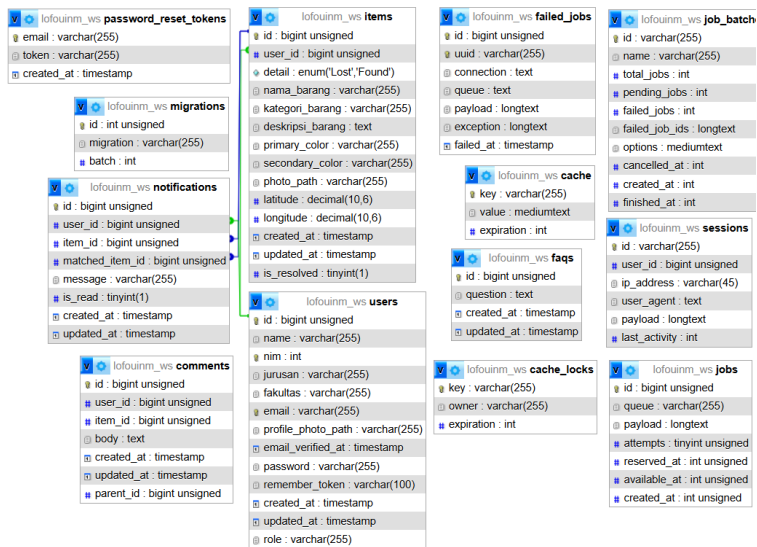
Tabel 4.2 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	AMD Ryzen 7 4800H
2	SSD	512GB
3	RAM	16GB
4	Monitor	15,6 Inch
5	Mouse	Standar
6	Keyboard	Standar

C. Implementasi Basis Data

Rancangan class diagram yang telah disusun pada tahap perancangan sistem digunakan sebagai panduan dalam implementasi basis data. Berdasarkan rancangan

tersebut, implementasi basis data direalisasikan menggunakan MySQL sebagai sistem manajemen basis data yang mendukung pengembangan sistem. Hasil dari implementasi tersebut ditampilkan pada Gambar 4.1



Gambar 4.1 Database Lost & Found

Detail tabel yang digunakan dalam platform website Lost & Found antara lain adalah

1. Tabel Users

Tabel 4.3 Database User

No	Nama	Tipe Data	Size	Keterangan
1	id	bigint		primary key
2	name	varchar	255	name
3	nim	int	10	nim (foreign key)
4	jurusan	varchar	255	jurusan
5	fakultas	varchar	255	fakultas

6	email	varchar	255	email (foreign key)
7	profile_photo_path	varchar	255	profile (null)
8	email_verified_at	timestamp		verify email (null)
9	password	varchar	255	password
10	remember_token	varchar	100	remember token (null)
11	created_at	timestamp		created_at
12	updated_at	timestamp		updated_at
13	role	varchar	255	role

2. Tabel barang

Tabel 4.4 Database Barang

No	Nama	Tipe Data	Size	Keterangan
1	id	bigint		primary_key
2	user_id	bigint		foreign_key
3	detail	enum	('Lost', 'Found')	detail
4	nama_barang	varchar	255	nama_barang
5	kategori_barang	varchar	255	kategori_barang
6	deskripsi_barang	text		deskripsi_barang
7	primary_color	varchar	255	primary_color
8	secondary_color	varchar	255	null
9	photo_path	varchar	255	photo_path
10	latitude	decimal	10,6	null
11	longitude	decimal	10,6	null
12	created_at	timestamp		null
13	updated_at	timestamp		null
14	is_resolved	tinyint	1	is_resolved

3. Tabel FAQ

Tabel 4.5 Database FAQ

No	Nama	Tipe Data	Size	Keterangan
1	id	bigint		Primary_Key
2	question	text		Question
3	created_at	timestamp		Created_at
4	updated_at	timestamp		Updated_at

4. Tabel Notifikasi

Tabel 4.6 Database Notifikasi

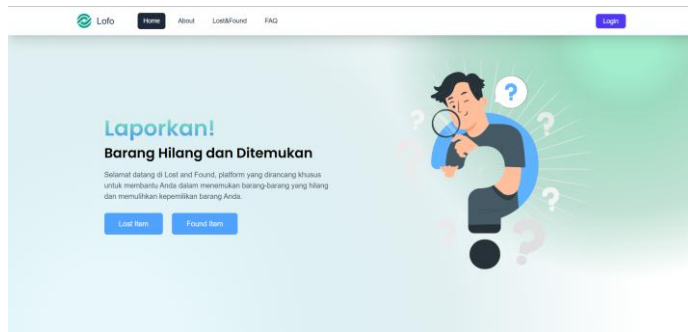
No	Nama	Tipe Data	Size	Keterangan
1	id	bigint		Primary_Key
2	user_id	bigint		Foreign_key
3	item_id	bigint		Foreign_key
4	matched_item_id	bigint		Foreign_key
5	message	varchar	255	Message
6	is_read	tinyint	1	Is_read
7	created_at	timestamp		Created_at
8	updated_at	timestamp		Updated_at

D. Implementasi Sistem

Tahap implementasi sistem adalah proses lanjutan dari perancangan, di mana rancangan antarmuka yang telah dibuat pada tahap sebelumnya mulai diwujudkan dalam bentuk nyata. Pada tahap ini, elemen-elemen desain diolah dan dikembangkan menjadi sebuah sistem yang memiliki fungsi operasional serta dapat digunakan secara langsung oleh pengguna.

1. Halaman *Dashboard*

Halaman *Dashboard User* ini dirancang sebagai titik awal interaksi pengguna, pada saat pertama kali, *user* hanya dapat melihat halaman *website*, untuk dapat mengisi *user* diharuskan *login* terlebih dahulu penjelasan peran pengguna, dan barang yang sering muncul pada web. Tampilan Dashboard pada gambar 4.2.



Gambar 4.2 Halaman Dashboard User

Untuk Kode dari Halaman Home dapat dilihat sebagai berikut:

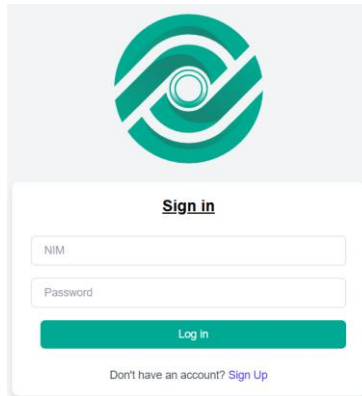
```
<div class="space-x-4">
  <a href="{ route('items.index', ['status' => 'Lost']) }"
  class="inline-block rounded-md bg-blue-400 px-8 py-3 text-white font-
  medium hover:bg-indigo-700">Lost Item</a>
  <a href="{ route('items.index', ['status' => 'Found']) }"
  class="inline-block rounded-md bg-blue-400 px-8 py-3 text-white font-
  medium hover:bg-indigo-700">Found Item</a>
</div>
```

Kode diatas berfungsi sebagai tombol untuk mengarahkan user secara langsung ke halaman lost &

found dengan jenis barang yang dicari yaitu “Lost” dan “Found”.

2. Halaman Login

Halaman Login dirancang sebagai platform user untuk dapat melakukan login dengan akun yang telah didaftarkan



Gambar 4.3 Halaman Login

Untuk Detail kode pada halaman login sebagai berikut:

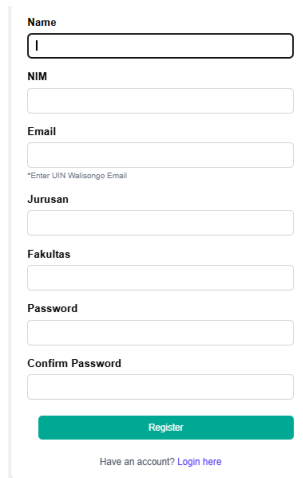
```
<x-auth-session-status class="mb-4 text-green-600 text-sm"
:status="session('status')" />
@if(session('success'))
    <div class="mb-4 text-green-600 text-sm">
        {{ session('success') }}
    </div>
@endif
```

Kode tersebut digunakan untuk mengenali autentikasi pengguna, jika sukses akan dapat login,

dan jika tidak akan mendapatkan notifikasi bahwa data tidak sesuai

3. Halaman Register

User diharapkan untuk melakukan pendaftaran akun sebelum melakukan login dalam website berikut untuk tampilan form pada register:



The image shows a registration form with the following fields and labels: Name, NIM, Email, Jurusan, Fakultas, Password, and Confirm Password. Below the fields is a green 'Register' button. At the bottom, there is a link that says 'Have an account? Login here'.

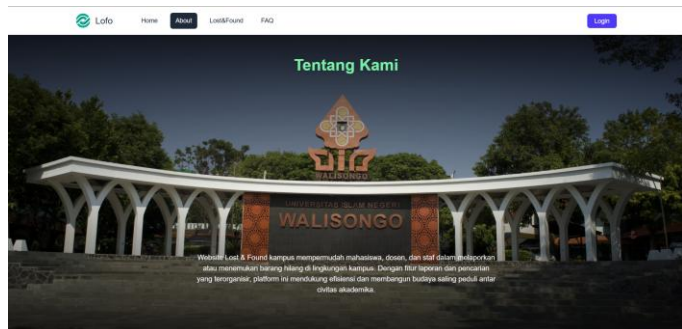
Gambar 4.4 Halaman Register

Kode berikut digunakan untuk menampilkan status pendaftaran pengguna. Jika proses registrasi berhasil, maka akan muncul pesan sukses; jika terjadi kesalahan, sistem akan menampilkan notifikasi sesuai dengan jenis error yang terjadi.

```
<form method="POST" action="{{ route('register') }}">
    @csrf
```

4. Halaman About

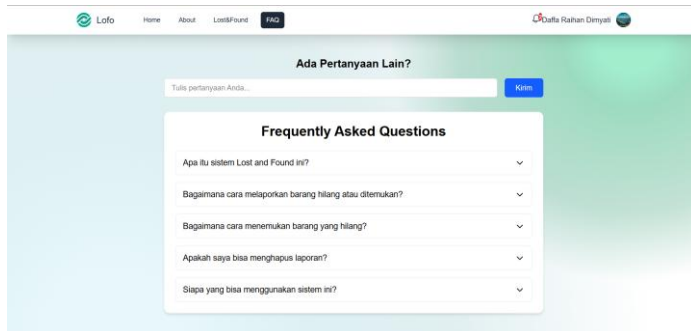
Halaman About ini dirancang untuk memberikan informasi mengenai tujuan dan fungsi dari platform Lost & Found kampus. Halaman ini menjelaskan sistem dibuat untuk memudahkan dalam melaporkan maupun mencari barang di lingkungan kampus. Tampilan halaman about pada gambar 4.3.



Gambar 4.5 Halaman About

5. Halaman FAQ

Halaman FAQ (Frequently Asked Questions) berfungsi sebagai pusat informasi untuk pertanyaan-pertanyaan umum seputar penggunaan website untuk membantu pengguna memahami cara kerja sistem, Tampilan halaman FAQ pada gambar 4.4.



Gambar 4.6 Halaman FAQ

Kode berikut digunakan user untuk menanyakan perihal kendala apa yang user dapatkan, lalu pesan tersebut akan diterima pada dashboard admin.

```
<form action="{ { route('faq.store') } }" method="POST" class="flex flex-col sm:flex-row gap-4">
    @csrf

    <input type="text" name="hero-input" placeholder="Tulis pertanyaan Anda..." class="bg-white flex-1 px-4 py-2 border border-gray-300 rounded-md focus:ring-2 focus:ring-blue-400 focus:outline-none">

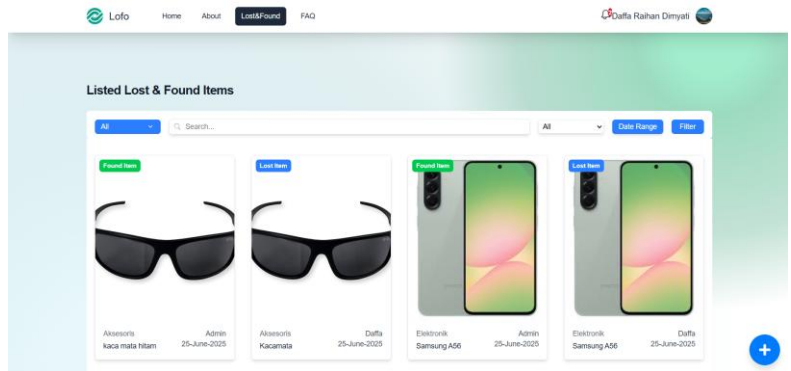
    <button type="submit" class="bg-blue-600 text-white px-6 py-2 rounded-md hover:bg-blue-700 transition">Kirim</button>

</form>
```

6. Halaman Lost & Found

Halaman Lost & Found menampilkan daftar barang yang dilaporkan hilang maupun ditemukan oleh pengguna. Halaman ini dirancang untuk memudahkan proses pencarian dan pelaporan barang

dengan fitur pencarian, filter kategori, dan rentang tanggal.



Gambar 4.7 Halaman Lost & Found

Kode berikut digunakan untuk mendapatkan informasi tentang daftar barang yang ada di database

```
<form method="GET" action="{ { route('items.index') } }">
</form>
```

User dapat melakukan filter jenis barang “Lost” atau “Found” apa yang dicari sesuai dengan kode berikut.

```
<div x-data="{ open: false, selected: '{ request('status', 'All') }' }">
    <input type="hidden" name="status" :value="selected">
    <button type="button" @click="open = !open">
        <span x-text="selected"></span>
    </button>
    <div x-show="open" @click.outside="open = false" x-transition>
        <button type="submit" @click="selected = 'All'; open = false">All</button>
        <button type="submit" @click="selected = 'Lost'; open = false">Lost</button>
```

```

<button type="submit" @click="selected =
'Found'; open = false">Found</button>
</div></div>

```

Kode selanjutnya yaitu fitur search dimana akan melakukan pencarian berdasarkan apa yang user input dan nantinya akan disesuaikan dengan barang yang tersedia

```

<div>
  <input type="text" name="search" value="{{
request('search') }}" placeholder="Search..." />
</div>

```

Fitur berikutnya yaitu adalah kategori barang, yang mana akan menampilkan sesuai dengan kategori yang user pilih, kategori telah terdaftar jenisnya pada database sesuai seperti kode berikut.

```

<select name="category">
  <option value="">All</option>
  <option value="Elektronik" {{
request('category') == 'Elektronik' ? 'selected' : ''
}}>Elektronik</option>
  <option value="Aksesoris" {{
request('category') == 'Aksesoris' ? 'selected' : ''
}}>Aksesoris</option>
  <option value="Tas" {{ request('category') ==
'Tas' ? 'selected' : '' }}>Tas</option>
  <option value="Baju" {{ request('category') ==
'Baju' ? 'selected' : '' }}>Baju</option>
  <option value="Alat Tulis" {{
request('category') == 'Alat Tulis' ? 'selected' : '' }}>Alat
Tulis</option>
</select>

```

User juga dapat melakukan filter kalender, agar user dapat melakukan pencarian sesuai tanggal kehilangan maupun menemukan sesuai kode berikut

```

<div x-data="{ open: false }">
  <button type="button" @click="open = !open">

```

```

Date Range
</button>
<div x-show="open" @click.outside="open = false" x-
transition>
  <label>From</label>
  <input type="date" name="date_from" value="{{
request('date_from') }}" />
  <label>To</label>
  <input type="date" name="date_to" value="{{
request('date_to') }}" />
</div>
</div>

```

Kode berikut akan menampilkan jika tidak ada barang yang terdaftar atau sesuai dengan yang user cari

```

@empty
    <p class="col-span-full text-center text-gray-
500">Belum ada barang yang terdaftar.</p>
@endforelse

```

Button pada kode berikut hanya akan muncul jika user telah melakukan login sehingga akan terdeteksi oleh session dan bisa melakukan upload barang

```

@auth
    <x-button-form />
@endauth

```

7. Halaman Form Lost & Found

Halaman ini menyediakan formulir bagi *user* untuk melaporkan barang yang hilang atau ditemukan. Terdapat pula fitur unggah gambar barang dan pinpoint lokasi pada peta yang ditampilkan menggunakan Leaflet OpenStreetMap. Informasi

yang diisi akan ditampilkan secara publik dan digunakan untuk membantu pencocokan barang secara otomatis oleh sistem.

Gambar 4.8 Halaman Form Lost & Found

User diminta untuk mengisi beberapa informasi penting seperti jenis laporan (lost/found), nama barang, kategori, deskripsi, warna, serta lokasi barang, untuk penambahan barang memiliki kode sebagai berikut:

```
public function store(Request $request){
    $validated = $request->validate([
        'detail' => 'required|in:Lost Item (Kehilangan
        Barang),Found Item (Penemuan Barang)',
        'nama_barang' => 'required|string|max:255']);
    $item = new Item([
        'detail' => str_contains($validated['detail'],
        'Lost') ? 'Lost' : 'Found',
        'nama_barang' => $validated['nama_barang'],
    ]);
    $item->save();
    return redirect('/items')->with('success', 'Item
    berhasil diunggah!');
}
```

Kode tersebut nantinya akan diterima oleh Sitem dan akan ditambahkan barang baru sesuai dengan validasi pada kode \$validated

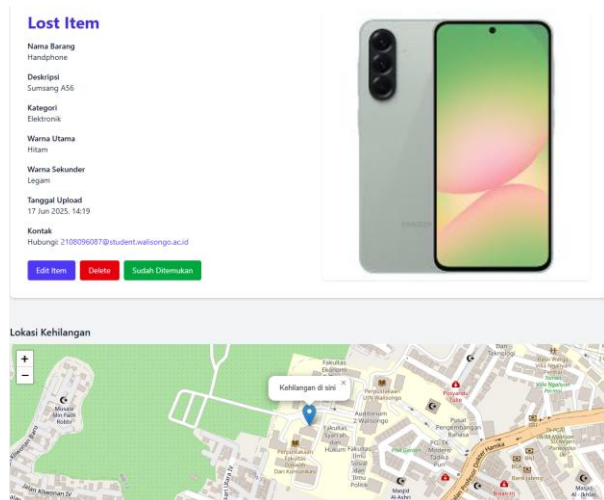
```
<script>
src="https://unpkg.com/leaflet@1.9.3/dist/leaflet.js"></script>
</script>
<script>
    // Inisialisasi peta
    var map = L.map('map').setView([-6.992741906578865,
110.35089910796111], 17);
    // Tambahkan layer OpenStreetMap
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}
}.png', {
        attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</
a> contributors'
    }).addTo(map);
    // Definisikan marker sebagai variabel kosong
    var marker;
    // Cek apakah ada nilai lama, kalau ada tampilkan marker
    var latInput = document.getElementById('latitude');
    var lngInput = document.getElementById('longitude');

    if (latInput.value && lngInput.value) {
        var lat = parseFloat(latInput.value);
        var lng = parseFloat(lngInput.value);
        marker = L.marker([lat, lng]).addTo(map);
        map.setView([lat, lng], 17);
    }
    // Saat peta diklik, taruh marker
    map.on('click', function (e) {
        var lat = e.latlng.lat;
        var lng = e.latlng.lng;
        // Jika marker sudah ada, hapus dulu
        if (marker) {
            map.removeLayer(marker);
        }
        // Tambahkan marker baru
        marker = L.marker([lat, lng]).addTo(map);
        // Set nilai ke input form
        latInput.value = lat;
        lngInput.value = lng;
    });
</script>
```

Kode diatas digunakan untuk menginisiasi API dari OpenStreetMap LeafletJS.

8. Halaman Detail Lost & Found

Halaman detail lost & found menyajikan informasi lengkap mengenai barang yang dilaporkan hilang, seperti nama barang, deskripsi singkat kejadian, kategori, warna utama, tanggal dan waktu unggah, serta kontak pelapor berupa email universitas. Selain itu, halaman ini juga menampilkan foto barang untuk membantu visualisasi dan peta lokasi kehilangan.



Gambar 4.9 Halaman Detail Lost & Found

Berikut adalah kode yang menampilkan hasil dari inputan map oleh OpenStreetMap LeafletJS.

```
<script
src="https://unpkg.com/leaflet@1.9.3/dist/leaflet.js"></script
>

<script>
    document.addEventListener("DOMContentLoaded", function
    () {

        const lat = ({ $item->latitude });
        const lng = ({ $item->longitude });
        const map = L.map('map').setView([lat, lng], 17);

        L.tileLayer('https://{s}.tile.openstreetmap.org/{z
       }/{x}/{y}.png', {
            attribution: '&copy; OpenStreetMap
            contributors'
        }).addTo(map);

        L.marker([lat, lng]).addTo(map)
            .bindPopup("{{ $item->detail === 'Found' ?
            'Nemu' : 'Kehilangan' }} di sini")
            .openPopup();

    });
</script>
```

9. Halaman Profile

Halaman *User Profile* berfungsi sebagai pusat pengelolaan data pribadi. Pengguna dapat memperbarui data diri, Fitur penggantian kata sandi, di mana pengguna perlu memasukkan kata sandi lama dan mengonfirmasi kata sandi baru untuk menjaga keamanan akun. Selain itu, terdapat bagian Riwayat Unggahan Barang yang menampilkan daftar barang yang pernah dilaporkan oleh pengguna.

User Profile



Profile Foto

Nama

Dafra Rahman (Demyati)

NIM

2108090007

Jurusan

It

Fakultas

Sains

Email

2108090007@student.walisongo.ac.id

Simpan

Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

Gambar 4.10 Halaman Profile

Riwayat Unggahan Barang

Daftar barang yang pernah kamu unggah sebagai barang hilang atau ditemukan.

Item Unggul

Baru - Hilang

17 Jan 2023, 09:47

Ubah

Handphone

Baru - Hilang

17 Jan 2023, 14:19

Ubah

Handphone Ditemukan

Baru - Hilang

17 Jan 2023, 08:58

Ubah

Delete Account

Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain.

DELETE ACCOUNT

Gambar 4.11 Halaman Profile Riwayat Upload

```

@if($items->isEmpty())
    <p>Belum ada barang yang kamu unggah.</p>
@else
    <div>
        @foreach ($items as $item)
            <div>
                <div>
                    <p>
                        {{ $item->nama_barang }}

                        @if($item->is_resolved)
                            <span>
                                Ditemukan!
                            </span>
                        @endif
                    </p>
                </div>
            </div>
        @endforeach
    </div>

```

```

{{ $item->kategori_barang }} - {{
$item->detail }}
</p>
<p>{{ $item->created_at->format('d M
Y, H:i') }}</p>
</div>
<a href="{ route('items.show', $item->id)
}">Lihat</a>
</div>
@endforeach
</div>
@endif

```

Kode tersebut merupakan kode yang menampilkan bagian history sesuai apa yang user upload, dengan tambahan jika sudah ditemukan akan ada tag nya untuk menandakan barang.

10. Halaman List Barang - Admin

Halaman ini menampilkan Admin Dashboard pada bagian List Barang, yang berisi tabel daftar barang hilang dan ditemukan. Setiap baris menampilkan foto, status (Lost/Found), nama, kategori, deskripsi, serta warna utama dan sekunder. Admin dapat mengedit atau menghapus data melalui tombol aksi sehingga memudahkan pengelolaan laporan secara efisien.

Admin Dashboard	List Barang	Log Out
Barang		
Pengguna		
FAQ		

Foto	Detail	Nama Barang	Kategori	Deskripsi	Primary Color	Secondary Color	Status	Actions
	Lost	kacamata	Aksesori	apalah	coklat	-	Belum Ditemukan	Edit Hapus
	Found	Handphone	Elektronik	Samsung A56 Casing putih	Hitam	putih	Belum Ditemukan	Edit Hapus
	Lost	Kacamata	Aksesori	Di caa ri	Coklat Item	Item	Belum Ditemukan	Edit Hapus
	Found	Buku	Alat Tulis	adadad	putih	-	Ditemukan	Edit Hapus
	Lost	Hp	Elektronik	Panjang pipih samsung z flip	Abu-abu	Abu hitam	Belum Ditemukan	Edit Hapus
	Found	Kunci Motor	Elektronik	Kunci	Hitam	Hitam	Belum Ditemukan	Edit Hapus

Gambar 4.12 Halaman List Barang Admin

Berikut merupakan kode untuk admin dapat mengedit dan menghapus barang apa yang pengguna upload, ini berfungsi untuk mencegah pengupload an yang tidak sesuai oleh pengguna.

```
<a href="{{ route('barang.edit', $item->id) }}">Edit</a>

<form action="{{ route('barang.destroy', $item->id) }}"
method="POST" onsubmit="return confirm('Yakin ingin menghapus
item ini?')">
    @csrf
    @method('DELETE')
    <button type="submit">Hapus</button>
</form>
```

```
<form action="{{ route('barang.update', $item->id) }}"
method="POST" enctype="multipart/form-data" class="bg-white
shadow-md rounded-lg p-8 space-y-6">
    @csrf
    @method('PUT')</form>
```

11. Halaman List Pengguna - Admin

Halaman ini menampilkan daftar pengguna yang terdaftar dalam sistem melalui tampilan tabel

yang informatif. Setiap baris tabel memuat data pengguna, seperti nama lengkap, NIM, alamat email, jurusan, fakultas, serta peran atau role pengguna dalam sistem, misalnya sebagai admin atau user biasa. Di sisi kanan setiap baris terdapat tombol aksi berupa “Edit” dan “Hapus” yang memungkinkan admin untuk memperbarui informasi pengguna atau menghapus akun jika diperlukan. Tampilan ini dirancang untuk memudahkan pengelolaan data pengguna secara efisien dan terstruktur oleh admin.

Nama	NIM	Email	Jurusan	Fakultas	Role	Actions
Dafra Raihan Dimiyati	2108096087	2108096087@student.walisongo.ac.id	TI	Saintek	User	Edit Hapus
Admin	12345678	Admin@student.walisongo.ac.id	Admin	Admin	Admin	Edit Hapus

Gambar 4.13 Halaman List Pengguna Admin

Berikut kode untuk action edit dan delete agar admin dapat mengubah dan menghapus data pengguna.

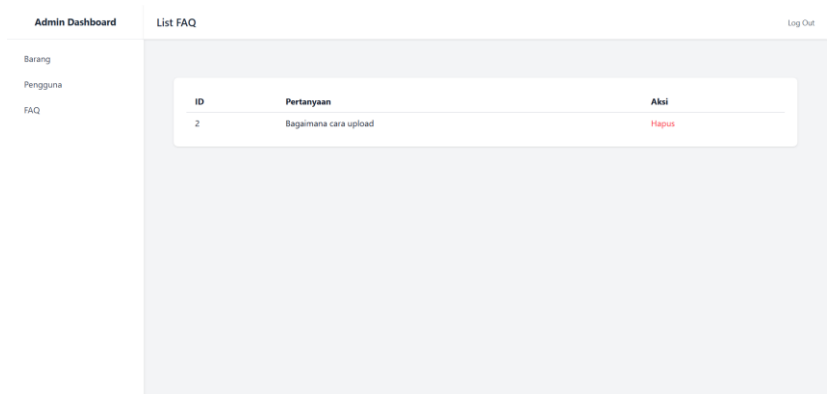
```
<a href="{ route('pengguna.edit', $user->id) }" class="text-blue-500 hover:underline">Edit</a>
```

```
<form action="{{ route('pengguna.destroy', $user->id) }}"
method="POST" class="inline-block" onsubmit="return
confirm('Yakin ingin menghapus pengguna ini?')">
    @csrf
    @method('DELETE')
    <button type="submit" class="text-red-500
hover:underline">Hapus</button>
</form>
```

```
<form action="{{ route('pengguna.update', $user->id) }}"
method="POST" class="bg-white shadow-md rounded-lg p-8 space-
y-6">
    @csrf
    @method('PUT')
</form>
```

12. Halaman List FAQ - Admin

Halaman List FAQ pada dashboard admin digunakan untuk menampilkan daftar pertanyaan yang diajukan oleh pengguna. Pada halaman ini, admin dapat melihat setiap pertanyaan beserta ID-nya dalam bentuk tabel yang sederhana dan bersih. Terdapat juga opsi untuk menghapus pertanyaan yang dianggap tidak relevan atau tidak perlu. Navigasi di sisi kiri memudahkan admin berpindah ke halaman lain seperti data barang atau pengguna. Dengan adanya halaman ini, admin dapat mengelola pertanyaan yang masuk secara terpusat dan efisien.



Gambar 4.14 Halaman List FAQ Admin

Berikut adalah kode yang dituliskan pada halaman FAQ Admin, penampilan kode berikut tidak ada data user dikarenakan pengirim tidaklah harus pengguna yang telah login, jadi tidak dapat menerima user id.

```
@forelse($questions as $question)
    <tr class="border-t border-gray-300">
        <td class="px-4 py-2">{{
            $question->id }}</td>
        <td class="px-4 py-2">{{
            $question->question }}</td>
        <td class="px-4 py-2">
            <form action="{{
                route('faq.destroy', $question->id) }}" method="POST"
                onsubmit="return confirm('Yakin ingin menghapus question
                ini?')">
                @csrf
                @method('DELETE')
                <button
                    type="submit" class="text-red-500
                    hover:underline">Hapus</button> </form>
            </td></tr>
@empty
```

E. Hasil Implementasi Perencanaan Waktu

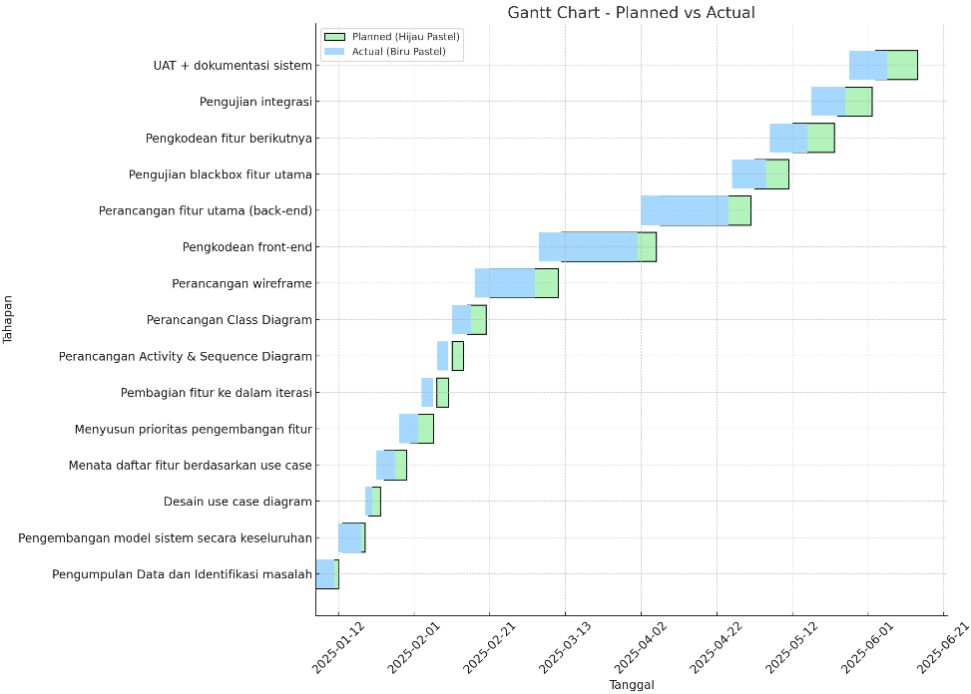
Tabel 4.7 Gantt Chart Perbandingan Perencanaan dan Aktual

No	Tahapan	Start Date (Planned)	End Date (Planned)	Duration (Planned)	Start Date (Actual)	End Date (Actual)	Duration (Actual)
Develop an overall model							
1	Pengumpulan Data dan Identifikasi masalah	06/Jan	12/Jan	7	06/Jan	11/Jan	6
2	Pengembangan model sistem secara keseluruhan	13/Jan	19/Jan	7	12/Jan	18/Jan	7
Build a features list							
3	Desain use case diagram	20/Jan	23/Jan	4	19/Jan	21/Jan	3
4	Menata daftar fitur berdasarkan use case	24/Jan	30/Jan	7	22/Jan	27/Jan	6
Plan by feature							
5	Menyusun prioritas pengembangan fitur	31/Jan	06/Feb	7	28/Jan	02/Feb	6
6	Pembagian fitur ke dalam iterasi	07/Feb	10/Feb	4	03/Feb	06/Feb	4
Design by feature							
7	Perancangan Activity & Sequence Diagram	11/Feb	14/Feb	4	07/Feb	10/Feb	4
8	Perancangan Class Diagram	15/Feb	20/Feb	6	11/Feb	16/Feb	6
Build by feature							
9	Perancangan wireframe	21/Feb	11/Mar	19	17/Feb	05/Mar	17
10	Pengkodean front-end	12/Mar	06/Apr	26	06/Mar	01/Apr	27
11	Perancangan fitur utama (back-end)	07/Apr	01/May	25	02/Apr	25/Apr	24
12	Pengujian blackbox fitur utama	02/May	11/May	10	26/Apr	05/May	10
13	Pengkodean fitur berikutnya	12/May	23/May	12	06/May	16/May	11
14	Pengujian integrasi	24/May	02/Jun	10	17/May	26/May	10
15	UAT + dokumentasi sistem	03/Jun	14/Jun	12	27/May	06/Jun	11

Berdasarkan perbandingan antara Gantt Chart jadwal rencana (planned) dan realisasi (actual), dapat disimpulkan bahwa proyek berjalan dengan sangat baik dan bahkan mengalami percepatan di sejumlah tahapan. Beberapa fase awal seperti pengumpulan data, pengembangan model sistem, dan penyusunan prioritas fitur diselesaikan lebih cepat dari yang direncanakan, mencerminkan efisiensi kerja tim dalam tahap analisis dan perencanaan.

Secara umum, durasi aktual dari berbagai tahapan cenderung lebih singkat dibandingkan estimasi awal, tanpa adanya penundaan signifikan. Bahkan, tidak terdapat satu pun tahapan yang mengalami keterlambatan dari jadwal. Hal ini menunjukkan bahwa estimasi waktu cukup konservatif, dan pelaksanaan proyek dilakukan dengan efektif.

Secara keseluruhan, proyek berhasil diselesaikan sekitar delapan hari lebih cepat dari jadwal yang telah ditentukan, menunjukkan bahwa proses pengelolaan waktu dalam proyek ini sangat efisien dan tim mampu bekerja secara optimal, berikut untuk tampilan grafik gantt chart berdasarkan perbandingan tersebut.



F. Hasil Pengujian Sistem

1. Hasil Pengujian Blackbox

Pengujian dengan metode Blackbox dilakukan untuk mengetahui hasil dari fungsional

dalam penggunaan website. Hasil pengujian tersebut disajikan pada tabel berikut.

Tabel 4.8 Hasil Pengujian Blackbox Admin

No	Form Uji	Skenario Uji	Hasil yang diharapkan	Hasil Uji
1	Login sebagai admin	Admin memasukkan username dan password dengan data admin yang terdaftar	Admin dapat login sebagai admin dengan username dan password admin yang sudah terdaftar	Valid
2		Masukkan username dan password admin yang salah	Muncul pesan error bahwa data yang dimasukkan tidak sesuai	Valid
3	Dashboard	Memasuki Dashboard Admin	Halaman utama Dashboard tampil	Valid
4	Lost & Found List sebagai Admin	Admin melihat semua hasil postingan yang telah diposting	Lost & Found List tampil dalam dashboard admin	Valid
5	Edit Lost & Found sebagai admin	Admin merubah status benda yang telah diposting	Admin dapat merubah status benda yang diposting	Valid

6	Hapus Lost & Found sebagai admin	Admin melakukan penghapusan postingan lost & found	Admin dapat menghapus postingan yang telah diposting oleh user	Valid
7	Edit Profil User sebagai admin	Admin melakukan pengubahan data pengguna	Admin dapat melakukan perubahan pada data pengguna	Valid
8	Hapus Profil User sebagai admin	Admin melakukan penghapusan data pengguna	Data User berhasil dihapuskan oleh admin	Valid
9	Pesan FAQ sebagai admin	Admin menerima pesan pertanyaan melalui FAQ yang dikirim oleh user	Pesan berhasil diterima pada dashboard admin	Valid

Tabel 4.9 Hasil Pengujian Blackbox User

No	Form Uji	Skenario Uji	Hasil yang diharapkan	Hasil Uji
1	Register sebagai User	Masukkan Data user sesuai form yang disediakan pada register	Pendaftaran akun user berhasil ditambahkan	Valid
2	Login sebagai User	User memasukkan username dan	User berhasil melakukan login, dan akan diarahkan ke halaman home	Valid

		password sesuai dengan yang didaftarkan		
3		User memasukkan username dan password yang tidak sesuai	User akan diberikan notifikasi bahwa data yang dimasukkan tidak sesuai	Valid
4	Halaman home	User memasuki halaman home setelah melakukan login	Halaman home ditampilkan dengan data yang sesuai dengan user masukkan	Valid
5	Halaman Lost & Found List sebagai user	User membuka halaman Lost & Found List	Menampilkan List barang yang telah didaftarkan/diposting user lain	Valid
6	Form Lost & Found sebagai user	User melakukan pengisian data untuk melakukan upload barang Lost & Found	Data barang yang diisi oleh user akan tampil pada halaman Lost & Found List	Valid
7	Search pada Lost & Found	User melakukan pencarian barang	Menampilkan barang yang dicari sesuai dengan inputan user	Valid
		User tidak menemukan barang yang dicari	Menampilkan pemberitahuan bahwa barang tidak ditemukan	Valid

8	Filter pada Lost & Found	User melakukan filter sesuai kategori yang ingin user cari	Menampilkan sesuai dengan kategori yang ingin user cari	Valid
9	Detail Lost & Found sebagai user	User membuka detail dari barang yang terdaftar pada Lost & Found List	Menampilkan data barang secara menyeluruh dan informasi dari user yang upload	Valid
10	Edit Lost & Found sebagai user	User melakukan edit pada barang yang telah di upload	User berhasil melakukan edit sesuai barang yang telah didaftarkan user itu sendiri	Valid
11	Hapus Lost & Found sebagai user	User melakukan penghapusan barang yang telah di upload	User berhasil melakukan hapus sesuai barang yang telah didaftarkan user itu sendiri	Valid
11	Tag Barang ditemukan	User melakukan klik pada tombol barang telah ditemukan	Barang akan hilang pada tampilan Lost & Found List, dan dinyatakan bahwa barang itu telah ditemukan pada history user profile	Valid
12	Edit User Profile	User melakukan edit data diri pada profile user	Data diri user berhasil diubah dengan data yang baru	Valid
13	Edit Password User	User melakukan perubahan	Password berhasil diubah dengan password yang baru	Valid

		password pada profile user		
14	History Upload	Menampilkan barang yang telah di upload oleh user	Menampilkan data barang yang telah user upload dan dapat dilihat detailnya	Valid
15	Hapus User Profile	Menghapus akun user pada user profile	Akun berhasil dihapus	Valid
16	Kirim pertanyaan melalui FAQ	User mengirim pertanyaan melalui FAQ	Pesan berhasil dikirimkan dan akan diterima oleh admin	Valid

Pengujian dengan metode black box terhadap karakteristik functional suitability telah dilaksanakan pada dua kategori pengguna, yaitu admin dan user. Masing-masing diuji berdasarkan jumlah skenario yang berbeda sesuai dengan peran dan tanggung jawab fungsionalnya.

- a) Pada pengguna admin, dilakukan sebanyak 9 skenario pengujian, yang seluruhnya berhasil dijalankan sesuai dengan harapan. Tingkat keberhasilan pengujian mencapai 100%.
- b) Pada pengguna user, pengujian dilakukan terhadap 16 skenario, dan seluruhnya juga menunjukkan hasil yang sesuai dengan fungsionalitas yang telah dirancang, sehingga menghasilkan persentase keberhasilan sebesar 100%.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa sistem telah memenuhi karakteristik functional suitability secara optimal, dengan tingkat keberhasilan penuh pada masing-masing jenis pengguna. Hal ini mengindikasikan bahwa seluruh fungsi sistem yang diuji telah berjalan dengan baik dan mampu memenuhi kebutuhan pengguna.

2. Hasil Pengujian *User Acceptance Test* (UAT)

Metode *User Acceptance Testing* (UAT) digunakan untuk mengevaluasi tanggapan pengguna atau responden terhadap sistem yang telah dikembangkan, biasanya dilakukan melalui penyebaran kuesioner (Bastari et al., 2022). Dalam proses ini, dilakukan survei dengan memberikan pertanyaan-pertanyaan kepada pengguna sebagai responden. Penilaian dilakukan berdasarkan skala yang tertera pada Tabel 4.5

Tabel 4.10 Bobot Penilaian Responden

Bobot Nilai	Keterangan
1	Sangat Tidak Setuju (STS)
2	Tidak Setuju (TS)
3	Netral / Ragu-ragu (RG)
4	Setuju (S)
5	Sangat Setuju (SS)

Hasil dari UAT menjadi bukti dokumentasi dalam proses pengujian sistem. Berdasarkan hasil tersebut, dapat disimpulkan apakah website yang diuji dapat diterima oleh pengguna atau tidak (Bastari et al., 2022).

Tabel 4.11 Kriteria Inspeksi Skor

Persentase	Keterangan
0% - 20%	Sangat Kurang Baik
21% - 40%	Kurang Baik
41% - 60%	Cukup Baik
61% - 80%	Baik
81% - 100%	Sangat Baik

Berikut adalah hasil dari frekuensi jawaban oleh responden yang didapatkan berjumlah 45 responden.

Tabel 4.12 Hasil UAT Responden

Pertanyaan	Frekuensi Jawaban				
	SS	S	RG	TS	STS
P1	25	18	2	0	0
P2	32	12	1	0	0
P3	31	13	1	0	0
P4	26	18	1	0	0
P5	29	13	3	0	0
P6	28	14	3	0	0
P7	27	16	2	0	0

P8	30	14	1	0	0
P9	22	19	4	0	0
P10	27	17	1	0	0
P11	25	18	2	0	0
P12	26	16	3	0	0
P13	31	13	0	0	0
P14	27	18	0	0	0
P15	25	18	2	0	0
P16	32	12	1	0	0
P17	24	15	6	0	0
P18	31	14	0	0	0

Berdasarkan pengujian UAT, diperoleh hasil distribusi tanggapan responden sebagai berikut:

Total Skor = (SS $\times$ 5) + (S $\times$ 4) + (RG $\times$ 3) + (TS $\times$ 2) + (STS $\times$ 1), sehingga persentase skor adalah:

Tabel 4.13 Hasil Skor Penghitungan UAT Responden

Pertanyaan	Frekuensi Jawaban					Bobot
	SS	S	RG	TS	STS	
P1	125	72	6	0	0	203
P2	160	48	3	0	0	211
P3	155	52	3	0	0	210
P4	130	72	3	0	0	205
P5	145	52	9	0	0	206
P6	140	56	9	0	0	205

P7	135	64	6	0	0	205
P8	150	56	3	0	0	209
P9	110	76	12	0	0	198
P10	135	68	3	0	0	206
P11	125	72	6	0	0	203
P12	130	64	9	0	0	203
P13	155	52	0	0	0	207
P14	135	72	0	0	0	207
P15	125	72	6	0	0	203
P16	160	48	3	0	0	211
P17	120	60	18	0	0	198
P18	155	56	0	0	0	211
Total Bobot						3701

Tabel di atas menyajikan hasil penghitungan berdasarkan skala Likert yang telah dikalikan dengan bobot penilaian dari para responden. Data yang ditampilkan dalam tabel tersebut kemudian dimanfaatkan untuk menghitung nilai persentase (Bastari et al., 2022). Adapun rumus untuk menghitung nilai rata-rata adalah sebagai berikut:

Rumus Nilai Rata-Rata:

$$\text{Nilai Rata - rata} = \frac{\text{Jumlah Total Nilai Bobot Responden}}{\text{Jumlah Responden}}$$

Setelah diperoleh nilai rata-rata, langkah selanjutnya adalah menentukan nilai persentase menggunakan rumus berikut:

Rumus Persentase:

$$\text{Persentase} = \left(\frac{\text{Nilai Rata - rata}}{\text{Bobot Maksimum}} \right) \times 100\%$$

Setelah memperoleh rumus nilai rata-rata, langkah selanjutnya adalah menghitung persentase dari setiap pertanyaan yang telah dijawab oleh responden. Perhitungan ini bertujuan untuk menilai apakah kualitas sistem yang diuji sudah layak atau belum. Hasil dari perhitungan persentase tersebut dapat dilihat pada Tabel 4.9.

Tabel 4.14 Hasil Persentase

Pertanyaan	Nilai Rata-rata	Persentase	Keterangan
P1	4,5	90,2%	Sangat Baik
P2	4,7	93,8%	Sangat Baik
P3	4,7	93,3%	Sangat Baik
P4	4,6	91,1%	Sangat Baik
P5	4,6	91,6%	Sangat Baik
P6	4,6	91,1%	Sangat Baik
P7	4,6	91,1%	Sangat Baik
P8	4,6	92,9%	Sangat Baik
P9	4,4	88,0%	Sangat Baik
P10	4,6	91,6%	Sangat Baik
P11	4,5	90,2%	Sangat Baik
P12	4,5	90,2%	Sangat Baik
P13	4,6	92,0%	Sangat Baik

P14	4,6	92,0%	Sangat Baik
P15	4,5	90,2%	Sangat Baik
P16	4,7	93,8%	Sangat Baik
P17	4,4	88,0%	Sangat Baik
P18	4,7	93,8%	Sangat Baik
Total Rata – rata		91,4%	Sangat Baik

Berdasarkan hasil evaluasi menggunakan metode System Usability Scale (SUS), sistem memperoleh skor rata-rata sebesar 91,4%, yang termasuk dalam kategori "Sangat Baik". Dengan demikian, Sistem Informasi Lost and Found yang menggunakan metode Feature Driven Development (FDD) dan berbasis Location Based Service (LBS) dinyatakan sangat layak dan efektif untuk digunakan.

BAB V

KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan terhadap perancangan sistem informasi *lost & found* berbasis *website*, maka diperoleh kesimpulan sebagai berikut:

1. Sistem informasi Lost & Found berbasis *website* mempermudah pencarian dan pelaporan barang hilang di lingkungan universitas. Fitur peta dengan API Leaflet memudahkan pengguna menandai lokasi, serta sistem notifikasi, dan komentar memudahkan pengguna dalam mencari barang. Hasil pengujian fungsional menggunakan metode black-box menunjukkan tingkat keberhasilan 100% dengan 25 skenario pengujian, sementara uji user acceptance oleh 45 responden mencatat tingkat kepuasan 91,4%.
2. Pengembangan sistem menggunakan pendekatan Feature Driven Development (FDD) dengan UML, Laravel, dan MySQL. Berdasarkan Gantt Chart, proyek berhasil diselesaikan 8 hari lebih cepat dari rencana. Hal ini menunjukkan bahwa FDD mendukung pengelolaan waktu yang efisien dan pengembangan sistem yang terstruktur.

B. Saran

Berdasarkan hasil penelitian yang berjudul " Rancang Bangun Sistem Informasi *Lost And Found* Menggunakan Metode *Feature Driven Development* (FDD) Berbasis *Location Based Service* (LBS)", masih terdapat beberapa keterbatasan yang perlu diperbaiki dan ditingkatkan. Oleh karena itu, peneliti menyampaikan beberapa rekomendasi sebagai berikut:

1. Karena *website lost & found* ini masih baru, maka diperlukan promosi terhadap mahasiswa agar lebih banyak yang mengetahui keberadaan website berikut.
2. Disarankan agar aplikasi dikembangkan ke dalam bentuk *mobile application* yang mendukung sistem operasi seperti iOS, Android, maupun Windows Phone.
3. Perlu ditambahkan fitur *chat online* antar pengguna untuk mempermudah proses komunikasi dan penyampaian informasi.
4. Dalam penelitian selanjutnya, disarankan untuk melakukan pembaharuan pada fitur map terhadap *website lost & found*.

DAFTAR PUSTAKA

- Al-Saqqa, S., Sawalha, S., & Abdelnabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11), 246–270. <https://doi.org/10.3991/ijim.v14i11.13269>
- Anamisa, D. R., & Mufarroha, F. A. (2020). *Dasar Pemrograman Web Teori & Implementasi (HTML, CSS, JavaScript, Bootstrap, CodeIgniter)*. Media Nusa Creative. <https://books.google.co.id/books?id=k2eEAAAQBAJ&printsec=copyright&hl=id#v=onepage&q&f=false>
- Ananda, G. D. D., Hidayatullah, M., & Nasution, V. A. (2023). *Perancangan Aplikasi Pengenalan Pariwisata Danau Toba Berbasis Android Menggunakan Layanan Location Based Service (LBS)*. <https://doi.org/10.62712/juktisi.v2i2.74>
- Ariq Sulthon Hanif, M. (n.d.). *Rancang Bangun Sistem Informasi Laporan Kunjungan Pasien Rumah Sakit Menggunakan Framework Laravel (Studi Kasus: Rumah Sakit Haji Jakarta UIN Syarif Hidayatullah)*. [Universitas Islam Negeri Syarif Hidayatullah Jakarta]. <https://repository.uinjkt.ac.id/dspace/bitstream/123456789/82358/1/MUHAMMAD%20ARIQ%20SULTHON%20HANI-F-FST.pdf>
- Bastari, M. A., Darmansah, D., & Rakhmadani, D. P. (2022). Sistem Informasi Jasa Cuci Interior Rumah dan Mobil Menggunakan Metode User Acceptance Test. *JURIKOM (Jurnal Riset Komputer)*, 9(2), 305. <https://doi.org/10.30865/jurikom.v9i2.3926>
- Chandra, I., Irawati, E., & Archie Kosasih, R. (2023). Aplikasi Web Pencarian Barang Hilang dengan Deteksi Lokasi Berbasis Google Map API. *Seminar Nasional Rekayasa, Sains Dan Teknologi*, 2(1), 1–5. <https://doi.org/10.47970/snarstek.v2i1.604>
- Coad, P., Lefebvre, E., & Luca, D. J. (1999). Java Modeling in Color with UML: In *USA: Prentice Hall PTR*. https://books.google.co.id/books/about/Java_Modeling_in_Color_with_UML.html?id=fo0_AQAAIAAJ&redir_esc=y

- Imananda, F. M. (2024). *Rancang Bangun Sistem Informasi Laboratorium Klinik Berbasis Web (Studi Kasus: Laboratorium Rumah Sakit Xyz)*. <https://repository.uinjkt.ac.id/dspace/handle/123456789/81292>
- Joseph Kurien, A. T., Mathew, S. A., & Mana, S. C. (2022). Development of PHP and MySQL based Digital Asset Management System for Secure Organizations. *2022 6th International Conference on Trends in Electronics and Informatics, ICOEI 2022 - Proceedings*, 1859–1863. <https://doi.org/10.1109/ICOEI53556.2022.9776698>
- Julius, R. F. (2021). *Rancang Bangun Portal Website Sekolah Pada SMAN 14 Muaro Jambi Sebagai Media Informasi Dengan Menggunakan Bahasa Pemrograman Php Dan Database MySQL SKRIPSI* [Universitas Putra Indonesia Padang]. http://repository.upiypk.ac.id/5267/4/SKRIPSI_RAVI%20FEBRI%20JULIUS_16101152610628_FULL%20TEXT.pdf
- Kaushal, U., Singh, G., & Parashar, T. (2022). Responsive Webpage Using HTML CSS. *International Conference on Cyber Resilience, ICCR 2022*. <https://doi.org/10.1109/ICCR56254.2022.9995922>
- Komara, A. P. (2024). *Rancang Bangun Sistem Manajemen Konferensi Ilmiah Berbasis Website Menggunakan Metode Rapid Application Development (RAD)* [Universitas Islam Negeri Syarif Hidayatullah Jakarta]. <https://repository.uinjkt.ac.id/dspace/handle/123456789/76161>
- Kusnadi, I. T., Maulana Huddin, J., Supiandi, A., & Oktapiani, R. (2024). Implementasi Feature Driven Development Pada Sistem Informasi Absensi Dan Penggajian (Sisenji) Berbasis Web. *Jurnal Responsif*, 6(2), 195–204. <https://ejurnal.ars.ac.id/index.php/jti>
- Mulia, G. S. G., Najoran, X. B. N., & Lumenta, A. S. M. (n.d.). Analisa Teknologi Hyper Text Markup Language (HTML) Versi 5. *Jurnal Teknik Informatika*. <http://repo.unsrat.ac.id/id/eprint/2977>

- Novianto Pratama, K. (2021). *Rancang Bangun Sistem Informasi E-Commerce Penjualan Baju Fashion Wanita Berbasis Website*. [Universitas Islam Negeri Syarif Hidayatullah Jakarta].
<https://repository.uinjkt.ac.id/dspace/bitstream/123456789/57156/1/KOKO%20NOVIANTO%20PRATAMA-FST.pdf>
- Nur, Q. (2021). Bulletin of Information Technology (BIT) Implementasi Konsep Business To Customers Nusa.Net Dengan Teknologi M-Commerce Menggunakan Metode Feature Driven Development. *Bulletin of Information Technology (BIT)*, 2(2), 45–50.
- Nuryasin, I., & Budianti, S. (2020). Sistem Informasi Pencarian Barang Hilang “Lost and Found” Pada Kampus 3 Universitas Muhammadiyah Malang. *REPOSITOR*, 2(5), 591–600.
<https://doi.org/10.22219/repositor.v2i5.30633>
- Palmer, S. R., & M. Felsing, J. (2002). *A Practical Guide to Feature-driven Development*. Prentice Hall PTR.
https://books.google.co.id/books/about/A_Practical_Guide_to_Feature_driven_Deve.html?id=NhlFAAAAYAAJ&redir_esc=y
- Putra Raharja, M., & Wijayanto, H. (2023). *Pengembangan Prototype Back-End Website Pelaporan Barang Hilang Di Pt. Presentologics Dicoding Academy Indonesia (Development Of A Prototype Back-End For A Lost Item Reporting Website At Pt. Presentologics Dicoding Academy Indonesia)*. <http://begawe.unram.ac.id/index.php/JBTI/>
- Setiawan, M. A., Andryana, S., & Gunaryati, A. (2021). Penerapan Algoritma Boyer Moore Dalam Pencarian Barang Hilang pada Aplikasi FindIt Berbasis Android. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(3), 945.
<https://doi.org/10.30865/mib.v5i3.3093>
- Suciati, P. E., & Arafat, M. Y. (2023). Sistem Informasi Pencarian Lokasi Gerai Umkm di Tangerang Selatan Berbasis Website Menggunakan Metode Location Based Service. In *Scientia Sacra: Jurnal Sains* (Vol. 3, Issue 2).
<http://pijarpemikiran.com/index.php/Scientia>

- Syahputra, C. (2023). *Rancang Bangun Sistem Informasi Akuntansi Berbasis Web* [Universitas Islam Negeri Syarif Hidayatullah Jakarta].
<https://repository.uinjkt.ac.id/dspace/handle/123456789/72560>
- Tamir, P. (2023). Implementasi Metode Feature Driven Development Pada Perancangan Web Portal Lomba. *Kohesi: Jurnal Multidisiplin Saintek*, 1(6), 1–10.
<https://doi.org/10.3785/kjst.v1i6.347>
- Wibisono, D. A. (2015). Rancang Bangun Sistem Informasi Pencarian Benda Hilang “Lost And Found” Berbasis Website Di Universitas Brawijaya. In *Kementrian Riset Teknologi Dan Pendidikan Tinggi Universitas Brawijaya*.
<http://repository.ub.ac.id/id/eprint/146530>
- Yadav, N., Singh Rajpoot, D., & Krishna Dhakad, S. (2019). *2019 Fifth International Conference on Image Information Processing (ICIIP)*. IEEE.
<https://ieeexplore.ieee.org/abstract/document/8985771/metrics#metrics>
- Zulvi, M. S., Sari, R. P., & Fitrisia, Y. (2021). *Pendekatan Metodologi Feature Driven Development Pada Aplikasi E-Commerce (Studi Kasus Cv. Megajaya)*.
<https://abecindonesia.org/proceeding/index.php/abec/article/view/50>

LAMPIRAN

A. Lampiran 1 Persetujuan Proposal

Proposal Skripsi ini telah disetujui oleh Pembimbing untuk dilaksanakan.

Disetujui pada

Hari : Selasa

Tanggal : 11 Februari 2025

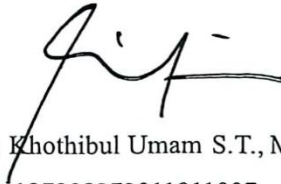
Pembimbing I,



(Hery Mustofa, M.Kom)

NIP. 198703172019031007

Pembimbing II,



(Dr. Khothibul Umam S.T., M. Kom)

NIP. 197908272011011007

Mengetahui,

Ketua Jurusan Teknologi Informasi



(Dr. Khothibul Umam S.T., M. Kom)

NIP. 197908272011011007

B. Lampiran 2 Lembar Pengesahan Proposal



**KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Prof Dr. Hamka Kampus III Ngaliyan Semarang
Telp. 7601295 Fax. 7615387

PENGESAHAN UJIAN KOMPREHENSIF

Naskah proposal skripsi berikut ini:

Judul : Rancang Bangun Sistem Informasi Lost And Found
Menggunakan Metode Feature Driven Development (FDD)
Berdasarkan Location Based Service (LBS)
Penulis : Daffa Raihan Dimiyati
NIM : 2108096087
Jurusan : Teknologi Informasi

Telah diuji dalam seminar proposal skripsi oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam bidang Ilmu Teknologi Informasi.

Semarang, 05 Juni 2025

DEWAN PENGUJI

Penguji I

Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom.
NIP. 197706222006042005

Penguji II

Dr. Khotibul Umam, M.Kom.
NIP. 197908272011011007

Penguji III

Siti Nurâ€™aini, M.Kom.
NIP. 198401312018012001

Penguji IV

Adhral Arwani Mahfudh, M.Kom.
NIP. 199107032019031006

C. Lampiran 3 Kuesioner Wawancara

Nama Lengkap	
NIM (Nomor Induk Mahasiswa)	
Email	
Apa saja kendala yang biasanya kamu hadapi ketika kehilangan atau menemukan barang di lingkungan kampus?	
Menurut kamu, fitur apa saja yang penting untuk ada dalam sebuah website Lost and Found agar memudahkan proses pelaporan dan pencarian barang?	
Bagaimana pendapat kamu tentang penggunaan website sebagai solusi digital dibandingkan dengan metode konvensional seperti papan pengumuman atau media sosial?	

D. Lampiran 4 Kuesioner Pengujian UAT

No	Pertanyaan	Pilihan Jawaban				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1	Website ini mudah dipahami oleh pengguna.					
2	Website berjalan dengan lancar tanpa hambatan.					
3	Fitur-fitur pada website dapat diakses dengan mudah.					
4	Website cukup responsif saat digunakan.					
5	Pengguna dapat melakukan login dan registrasi dengan baik.					
6	Informasi barang ditampilkan dengan benar oleh sistem.					
7	Proses input data barang oleh pengguna berjalan dengan baik.					
8	Tombol seperti “Tambah”, “Edit”, “Hapus”, dan “Simpan” berfungsi dengan baik.					
9	Fitur peta dapat berjalan dan menampilkan lokasi dengan baik.					
10	Fitur pencarian barang berfungsi dengan benar di halaman lost and found.					

11	Website mampu mengelola data barang dengan efisien.					
12	Fitur filter kategori bekerja dengan baik saat melakukan pencarian.					
13	Tampilan antarmuka (dashboard) mudah dipahami.					
14	Kombinasi warna pada website jelas dan nyaman dilihat.					
15	Tata letak tampilan website rapi dan terorganisir.					
16	Ukuran dan jenis font pada teks mudah dibaca.					
17	Desain antarmuka berfungsi baik di berbagai perangkat (desktop, tablet, ponsel).					
18	Secara keseluruhan, saya puas dengan sistem informasi Lost & Found ini.					

E. Lampiran 5 Link Form Responden

<https://forms.gle/eB7dkVPqyQZVjiGw7>

Sistem Informasi Pelaporan Barang Lost & Found

B *I* U  

Halo! Perkenalkan, saya Daffa Raihan Dimiyati, mahasiswa Universitas Islam Negeri Walisongo, Program Studi Teknologi Informasi (S1). Saya tengah menyusun tugas akhir di bawah bimbingan Bapak Hery Mustofa, M.Kom yang bertujuan untuk memperoleh data dalam rangka menyelesaikan project akhir ini. Mohon bantuannya untuk pengisi diharapkan mahasiswa UIN Walisongo Semarang yang masih aktif, terimakasih

Silahkan akses link berikut sebelum menjawab pertanyaan.
gunakan **NIM dan Data Diri Pribadi** dalam melakukan pendaftaran.
Contoh:
Nim: 2108096087
Email: 2108096087@student.walisongo.ac.id

LINK WEBSITE :
lofouin.my.id

Jika ada yang ingin ditanyakan bisa hubungi 2108096087@student.walisongo.ac.id

 Terima kasih banyak atas partisipasi dan waktu Anda!

This form is automatically collecting emails from all respondents. [Change settings](#)

F. Lampiran 6 Code Pemrograman Database

Database User

```
0001_01_01_000000_create_users_table.php X
database > migrations > 0001_01_01_000000_create_users_table.php > ...
1 </php>
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->integer('nim')->unique();
18             $table->string('jurusan');
19             $table->string('jabatan');
20             $table->string('email')->unique();
21             $table->timestamp('email_verified_at')->nullable();
22             $table->string('password');
23             $table->rememberToken();
24             $table->timestamps();
25         });
26
27         Schema::create('password_reset_tokens', function (Blueprint $table) {
28             $table->string('email')->primary();
29             $table->string('token');
30             $table->timestamp('created_at')->nullable();
31         });
32
33         Schema::create('sessions', function (Blueprint $table) {
34             $table->string('id')->primary();
35             $table->foreignId('user_id')->nullable()->index();
36             $table->string('ip_address', 45)->nullable();
37             $table->text('user_agent')->nullable();
38         });
39     }
40
41     /**
42      * Reverse the migrations.
43      */
44     public function down(): void
45     {
46         Schema::dropIfExists('users');
47         Schema::dropIfExists('password_reset_tokens');
48         Schema::dropIfExists('sessions');
49     }
50 }
```

Database Barang

```
2025_04_29_075952_create_items_table.php X
database > migrations > 2025_04_29_075952_create_items_table.php > ...
1 </php>
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('items', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId('user_id')->constrained()->onDelete('cascade');
17             $table->enum('detail', ['lost', 'found']);
18             $table->string('nama_barang');
19             $table->string('kategori_barang');
20             $table->text('deskripsi_barang');
21             $table->string('primary_color');
22             $table->string('secondary_color')->nullable();
23             $table->string('photo_path');
24             $table->decimal('latitude', 10, 6)->nullable();
25             $table->decimal('longitude', 10, 6)->nullable();
26             $table->timestamps();
27         });
28
29         /**
30          * Reverse the migrations.
31          */
32         public function down(): void
33         {
34             Schema::dropIfExists('items');
35         }
36     }
37 }
```

RIWAYAT HIDUP

A. Identitas Diri

1. Nama Lengkap : Daffa Raihan Dimyati
2. Tempat & Tanggal Lahir : Tegal, 26 Oktober 2003
3. Alamat Rumah : Perum Puri Nirwana
4. HP : 082324631383
5. E-mail : daffadimyati2610@gmail.com

B. Riwayat Pendidikan

1. Sekolah Dasar Negeri 3 Randugunting Kota Tegal
2. Sekolah Menengah Pertama Negeri 5 Kota Tegal
3. Sekolah Menengah Kejuruan 3 Kota Tegal

Semarang, 22 Juni 2025



Daffa Raihan Dimyati

NIM. 2108096087