

**RANCANG BANGUN APLIKASI *SPLIT BILL* BERBASIS WEB  
MENGUNAKAN METODE PXP**

**(Skripsi)**

**Oleh**

**ALJABBAR RIZAL AZHAR**

**2015061053**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2025**

**RANCANG BANGUN APLIKASI *SPLIT BILL* BERBASIS WEB  
MENGUNAKAN METODE PXP**

**Oleh**

**ALJABBAR RIZAL AZHAR**

**2015061053**

**SKRIPSI**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA TEKNIK**

**Pada**

**Program Studi**

**Teknik Informatika**

**Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2025**

## ABSTRAK

### RANCANG BANGUN APLIKASI SPLIT BILL BERBASIS WEB MENGUNAKAN METODE PXP

OLEH

ALJABBAR RIZAL AZHAR

Dalam kehidupan sehari-hari, pembagian biaya sering kali menjadi tantangan, terutama dalam situasi kelompok. Kesulitan mencatat dan mengorganisir pembagian tagihan secara manual sering memicu ketidakadilan atau kesalahpahaman. Penelitian ini bertujuan mengembangkan aplikasi *split bill* berbasis web yang membantu pengguna mencatat, membagi, dan menyesuaikan pembagian biaya dengan transparan dan efisien. Metode pengembangan yang digunakan adalah *Personal Extreme Programming (PXP)*, sebuah model yang fleksibel dan adaptif untuk pengembangan perangkat lunak oleh pengembang individu. Aplikasi ini dibangun menggunakan *library React.js* untuk memastikan antarmuka yang responsif dan efisien. Survei awal terhadap 26 responden menunjukkan tingkat kebutuhan yang tinggi terhadap aplikasi ini, dengan rata-rata skor 4,38 dari skala 5. Hasil pengujian menunjukkan bahwa aplikasi *split bill* berfungsi dengan baik berdasarkan metode *black box testing*, yang mencakup berbagai fitur utama seperti autentikasi, pembuatan *split bill*, metode pembagian, dan akses tanpa login. Skor performa akhir aplikasi menggunakan *Lighthouse* mencapai 97 untuk performa, 84 untuk aksesibilitas, 100 untuk praktik terbaik, dan 82 untuk *SEO*. Evaluasi heuristik menunjukkan bahwa terdapat beberapa aspek yang perlu diperbaiki, yaitu pada poin 2 dengan skor rata-rata 2, poin 6 dengan skor rata-rata 1,67, dan poin 7 dengan skor rata-rata 1. Meski demikian, evaluasi ini berhasil diterapkan untuk meningkatkan tingkat usability aplikasi pada iterasi pengembangan tambahan.

**Kata Kunci:** aplikasi web, *firebase*, metode PXP, *React.js*, *split bill*.

## ABSTRACT

### RANCANG BANGUN APLIKASI SPLIT BILL BERBASIS WEB MENGUNAKAN METODE PXP

OLEH

ALJABBAR RIZAL AZHAR

In everyday life, splitting expenses often becomes a challenge, especially in group situations. The difficulty of manually recording and organizing expense sharing frequently leads to unfairness or misunderstandings. This research aims to develop a web-based split bill application that helps users record, split, and adjust expenses with transparency and efficiency. The development method used is *Personal Extreme Programming (PXP)*, a flexible and adaptive model for individual software developers. The application is built using the *React.js* library to ensure a responsive and efficient user interface. An initial survey involving 26 respondents indicated a high level of demand for this application, with an average score of 4.38 out of 5. Testing results showed that the split bill application performed well based on the *black box testing* method, covering key features such as authentication, split bill creation, splitting methods, and access without login. The application's final performance score using *Lighthouse* reached 97 for performance, 84 for accessibility, 100 for best practices, and 82 for SEO. The heuristic evaluation shows that there are some aspects that need to be improved, namely at point 2 with an average score of 2, point 6 with an average score of 1.67, and point 7 with an average score of 1. Nevertheless, this evaluation was successfully applied to improve the usability level of the application in additional development iterations.

**Keywords:** Firebase, PXP method, React.js, split bill, web application.

Judul Skripsi : RANCANG BANGUN APLIKASI *SPLIT BILL*  
BERBASIS WEB MENGGUNAKAN  
METODE PXP

Nama Mahasiswa : *Aljabbar Rizal Azhar*

Nomor Pokok Mahasiswa : 2015061053

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

Fakultas : Teknik

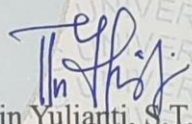
### MENYETUJUI

#### 1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

  
Dr. Eng. Ir. Mardiana, S.T., M.T.,  
I.P.M.


  
Ir. Titin Yulianti, S.T., M.Eng.  
NIP : 198807092019032015

NIP : 197216031999032002

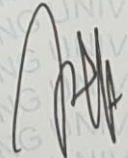
#### 2. Mengetahui

Ketua Jurusan  
Teknik Elektro

Ketua Program Studi  
Teknik Informatika

  
Herlinawati, S.T., M.T.

NIP 197103141999032001

  
Yessi Mulyani, S.T., M.T.

NIP 197312262000122001

**MENGESAHKAN**

1. **Tim Penguji**

**Ketua : Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M**

**Sekretaris : Ir. Titin Yulianti, S.T., M.Eng.**

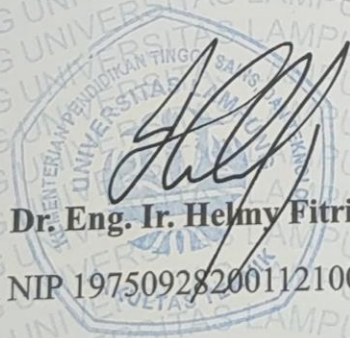
**Penguji : Ir. Gigih Forda Nama, S.T. M.T.I. I.P.M**

2. **Dekan Fakultas Teknik**

**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. )**

**NIP 197509282001121002**

**Tanggal Lulus Ujian Skripsi: 23 Januari 2025**



## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul "Rancang Bangun Aplikasi Split Bill Berbasis Web Menggunakan Metode PXP" dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 17 Februari 2025

Pembuat pernyataan,



Aljabbar Rizal Azhar

NPM 2015061053

## RIWAYAT HIDUP



Penulis dilahirkan di Jambi, pada tanggal 18 Desember 2002. Penulis merupakan anak pertama dari tiga bersaudara. Penulis menyelesaikan pendidikannya di SDN 1 Tanjung Sakti Pumi pada tahun 2014, MTs Al-Ikhlas Tanjung Sakti pada tahun 2017, dan MAN 5 Jakarta pada tahun 2020. Pada Tahun 2020, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjalani kuliah, penulis telah memfokuskan diri pada pengembangan perangkat lunak dengan mengambil konsentrasi Rekayasa Perangkat Lunak. Selama menjadi mahasiswa, penulis telah menjalani berbagai kegiatan antara lain:

1. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas Front -End Web dan dan Back-end pada tahun 2022.
2. Melaksanakan Kuliah Kerja Nyata di Desa Kesumadadi, Kecamatan Bekri, Kabupaten Lampung Tengah, Provinsi Lampung pada bulan Januari sampai dengan Februari 2023.
3. Mengikuti program Magang Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan menjadi staf magang sebagai Software Quality Assurance pada divisi IT di Vocasia.
4. Mengikuti program pelatihan Associate Data Scientist Vocational School Graduate Academy (VSGA) Gelombang ke-14 pada tahun 2024.



## **MOTTO**

*“Everybody wants the view but nobody wants the climb”*

*“I can and I will”*

*“You never lose. You either win or learn”*

## **PERSEMBAHAN**

Dengan penuh rasa syukur, saya persembahkan skripsi ini kepada:  
Kedua orang tua saya yang tercinta, Ayah dan Ibu, yang selalu memberikan doa,  
dukungan, dan cinta tiada henti.  
Seluruh pihak yang terlibat dan berkontribusi dalam penyelesaian penelitian dan  
penyusunan skripsi ini.  
Diri saya sendiri, **ALJABBAR RIZAL AZHAR**, untuk tidak pernah menyerah  
dan terus berusaha dalam setiap proses penyelesaian skripsi ini.

## SANWACANA

Alhamdulillah rabbil'alamin, Segala puji dan syukur penulis panjatkan kepada Allah SWT, Tuhan semesta alam, yang dengan rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi ini. Sholawat serta salam penulis sampaikan kepada Nabi Muhammad SAW, keluarganya, sahabatnya, dan para pengikutnya hingga akhir zaman.

Skripsi dengan judul “Rancang Bangun Aplikasi Split Bill Berbasis Web dengan Menggunakan Metode PXP” ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Universitas Lampung. Dalam proses penelitian dan penyusunan skripsi ini, penulis menerima banyak dukungan dan kontribusi dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Keluarga penulis, terutama Ayah dan Ibu yang selalu memberikan doa, dukungan, dan kasih sayang tiada henti;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung yang telah membantu proses kelancaran pengerjaan penelitian;
5. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M selaku pembimbing utama dan dosen pembimbing akademik yang telah bersedia membimbing penulis

selama melaksanakan penelitian ini dan juga melaksanakan proses perkuliahan;

6. Ibu Ir. Titin Yulianti, S.T., M.Eng. selaku pembimbing pendamping yang telah banyak memberikan masukan dalam proses penelitian ini;
7. Bapak Ir. Gigih Forda Nama, S.T, M.T.I., I.P.M., selaku Dosen Penguji yang selalu memberikan masukan dan apresiasi terhadap penelitian ini;
8. Seluruh dosen dan staf Jurusan Teknik Informatika Unila yang memberi masukan dan mempermudah proses penelitian ini;

Penulis menyadari bahwa penelitian ini masih memiliki banyak kekurangan. Oleh karena itu, penulis memohon maaf dan menerima segala bentuk kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi masyarakat luas.

Bandar Lampung, 13 Februari 2025

Penulis,

Aljabbar Rizal Azhar

## DAFTAR ISI

DAFTAR ISI .....	i
DAFTAR GAMBAR .....	iii
DAFTAR TABEL.....	vi
DAFTAR LAMPIRAN .....	vii
I. PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan.....	4
1.4 Manfaat .....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan .....	5
II. TINJAUAN PUSTAKA .....	6
2.1 <i>Split Bill</i> .....	6
2.1. UML.....	7
2.2. <i>Personal eXtreme Programming</i> .....	8
2.3. <i>Tools yang Digunakan Selama Pengembangan</i> .....	9
2.3.1 <i>Visual Studio Code</i> .....	9
2.3.2 <i>React Js</i> .....	10
2.3.3 <i>Firebase</i> .....	11
2.3.4 <i>Lighthouse</i> .....	12
2.4 <i>Black box Testing</i> .....	13
2.4 <i>Heuristic Evaluation</i> .....	14
2.5. Penelitian Terkait.....	15
III. METODOLOGI PENELITIAN.....	22
3.1.1 Waktu dan Tempat.....	22
3.2 Alat dan bahan.....	23
3.3 Tahapan Penelitian .....	24
3.3.1 <i>Requirements</i> .....	25

3.3.2	<i>Planning</i> .....	25
3.3.3	<i>Design</i> .....	26
3.3.4	<i>Implementation</i> .....	26
3.3.5	<i>Testing</i> .....	27
3.3.6	<i>Iteration</i> .....	27
3.3.7	<i>Restropective</i> .....	27
IV.	HASIL DAN PEMBAHASAN .....	29
4.1	Requierements.....	29
4.1.1	Kuesioner .....	29
4.1.2	Spesifikasi Kebutuhan.....	32
4.2	Planning .....	42
4.3	Iterasi.....	44
4.3.1	Iterasi 1 .....	44
4.3.2	Iterasi 2.....	53
4.3.3	Iterasi 3.....	64
4.3.4	Iterasi 4.....	72
4.3.5	Iterasi 5.....	78
4.3.6	Iterasi 6.....	94
4.5	Pembahasan.....	114
V.	SIMPULAN DAN SARAN .....	116
5.1	Simpulan .....	116
5.2	Saran.....	117
	DAFTAR PUSTAKA .....	118
	LAMPIRAN 1 .....	121
	LAMPIRAN 2 .....	125

## DAFTAR GAMBAR

Gambar 2.1 Proses kerja dari teknik pengujian Black box[13] .....	13
Gambar 3. 1 Tahapan Penelitian .....	24
Gambar 4. 1 Use Case Diagram .....	36
Gambar 4. 2 Activity diagram use case autentikasi .....	37
Gambar 4. 3 Activity diagram use case buat split bill baru .....	38
Gambar 4. 4 Activity diagram use case lihat dan bagikan split bill.....	39
Gambar 4. 5 Activity diagram use case melihat split bill yang dibagikan.....	39
Gambar 4. 6 Class Diagram Aplikasi.....	40
Gambar 4. 7 Desain Sign Up.....	45
Gambar 4. 8 Desain Sign In .....	45
Gambar 4. 9 Desain Halaman Home .....	46
Gambar 4. 10 integrasi dan konfigurasi firebase .....	47
Gambar 4. 11 Activity diagram proses login dengan firebase authentication.....	48
Gambar 4. 12 Implementasi daftar akun .....	49
Gambar 4. 13 Impementasi masuk akun.....	49
Gambar 4. 14 Implementasi masuk halaman utama setelah login.....	50
Gambar 4. 15 Desain buat Split Bill baru .....	54
Gambar 4. 16 Desain tambahkan anggota Split Bill .....	55
Gambar 4. 17 Desain pilih opsi pembayaran .....	55
Gambar 4. 18 Desain konfirmasi sebelum meyimpan .....	56
Gambar 4. 19 logika pembuatan untuk tiga metode .....	57
Gambar 4. 20 logika perhitungan pembagian .....	58
Gambar 4. 21 Implementasi formulir Split Bill metode sesuai porsi.....	59
Gambar 4. 22 Implementasi tambah anggota untuk metode sesuai porsi .....	59
Gambar 4. 23 Implementasi pesan peringatan gagal menambahkan anggota .....	60
Gambar 4. 24 Implementasi pilihan metode pembayaran.....	60

Gambar 4. 25 Implementasi konfirmasi Split Bill sebelum meyimpan .....	61
Gambar 4. 26 Desain buat Split Bill baru metode persenan .....	65
Gambar 4. 27 Desain tambahkan anggota Split Bill dengan metode persentase ..	66
Gambar 4. 28 diagram alir render form masukkan berdesarkan metode .....	67
Gambar 4. 29 Implementasi formulir Split Bill metode sesuai persentase .....	68
Gambar 4. 30 Implementasi tambah anggota untuk metode sesuai persentase ....	68
Gambar 4. 31 Implementasi pesan peringatan jika total pesentase tidak sama dengan 100% .....	69
Gambar 4. 32 Implementasi metode pembayaran untuk Split Bill dengan metode persentase .....	69
Gambar 4. 33 Implementasi konfirmasi Split Bill sebelum meyimpan .....	70
Gambar 4. 34 Desain buat Split Bill baru metode bagi rata.....	74
Gambar 4. 35 Desain tambahkan anggota Split Bill bagi rata .....	74
Gambar 4. 36 Implementasi formulir Split Bill metode bagi rata.....	75
Gambar 4. 37 Implementasi halaman anggota untuk metode bagi rata .....	76
Gambar 4. 38 Implementasi metode pembayaran untuk Split Bill dengan metode bagi rata.....	76
Gambar 4. 39 Implementasi konfirmasi Split Bill metode bagi rata sebelum meyimpan.....	77
Gambar 4. 40 Desain Halaman utama.....	80
Gambar 4. 41 Desain detail Split Bill .....	80
Gambar 4. 42 Implementasi daftar Split Bill di halaman utama.....	81
Gambar 4. 43 Implementasi detail Split Bill.....	82
Gambar 4. 44 Implementasi fitur bagikan tautan.....	82
Gambar 4. 45 konfirmasi sukses saat login.....	85
Gambar 4. 46 konfirmasi sukses saat menyimpan split bill.....	85
Gambar 4. 47 penerapan bahasa Indonesia pada antarmuka aplikasi .....	86
Gambar 4. 48 penerapan prinsip kontrol dan kebebasan pengguna.....	86
Gambar 4. 49 penerapan prinsip konsistensi dan standar .....	87
Gambar 4. 50 penerapan poin pencegahan kesalahan pada input harga .....	87
Gambar 4. 51 penerapan prinsip recognition rather than recall .....	88
Gambar 4. 52 tombol Kembali untuk ke halaman sebelumnya .....	88



Gambar 4. 53 ikon untuk menghapus split bill .....	89
Gambar 4. 54 pesan peringatan total tagihan tidak sama.....	89
Gambar 4. 55 contoh penerapan prinsip bantuan dan dokumentasi.....	90
Gambar 4. 56 Desain Perbaikan Format Harga .....	95
Gambar 4. 57 Desain penambahan placeholder .....	96
Gambar 4. 58 Desain perbaikan pada nama metode split bill.....	96
Gambar 4. 59 Desain sisa tagihan yang belum dialokasikan .....	97
Gambar 4. 60 Desain penambahan tombol kembali ke halaman utama .....	97
Gambar 4. 61 Implementasi perbaikan Format harga .....	98
Gambar 4. 62 Implementasi penambahan placeholder pada isian nama split bill dan total tagihan.....	98
Gambar 4. 63 Implementasi perbaikan nama metode sesuai porsi .....	99
Gambar 4. 64 Implementasi perbaikan nama metode bagi rata .....	99
Gambar 4. 65 Implementasi perbaikan nama metode .....	100
Gambar 4. 66 Implementasi penambahan informasi mengenai sisa tagihan yang belum dialokasikan.....	101
Gambar 4. 67 Implementasi penambahan tombol kembali ke halaman utama...	101
Gambar 4. 68 Hasil pengujian performa menggunakan lighthouse.....	111
Gambar 4. 69 Hasil pengujian performa setelah melakukan proses build.....	112

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	18
Tabel 3. 1 Jadwal Kegiatan Penelitian .....	22
Tabel 3. 2 Alat dan Bahan Penelitian .....	23
Tabel 4. 1 Tabel kuesioner.....	29
Tabel 4. 2 Kebutuhan Fungsional.....	32
Tabel 4. 3 Planning.....	43
Tabel 4. 4 Pengujian iterasi 1 .....	51
Tabel 4. 5 Pengujian untuk iterasi 2 .....	62
Tabel 4. 6 Pengujian untuk iterasi 3 .....	70
Tabel 4. 7 Pengujian untuk iterasi 4 .....	77
Tabel 4. 8 Pengujian Iterasi 5 .....	83
Tabel 4. 9 Daftar Evaluator dan Pengalaman.....	90
Tabel 4. 10 Rekap Hasil Pengujian Heuristik Skala Tunggal .....	91
Tabel 4. 11 Rekomendasi Perbaikan .....	92
Tabel 4. 12 Pengujian Akhir Autentikasi dan Split Bill Metode Sesuai Porsi ....	102
Tabel 4. 13 Pengujian Akhir untuk Metode Persentase.....	106
Tabel 4. 14 Pengujian Akhir untuk Metode Bagi Rata.....	108
Tabel 4. 15 Pengujian Akhir untuk Akses dan Bagikan Split bill .....	109

## DAFTAR LAMPIRAN

LAMPIRAN 1 .....	121
LAMPIRAN 2 .....	125

# I. PENDAHULUAN

## 1.1 Latar Belakang

Dalam kehidupan sehari-hari, seringkali kita menghadapi situasi di mana perlu berbagi biaya dengan teman, keluarga, atau rekan kerja. Pencatatan pembagian biaya tanpa alat bantu seringkali memunculkan kesulitan dalam mengelola informasi secara rapi, terutama dalam situasi yang tidak memungkinkan untuk membawa alat tulis, seperti saat berada di luar rumah. Tidak adanya sistem pencatatan yang jelas dapat menimbulkan kebingungan, membuat pembagian biaya terasa tidak adil, atau bahkan memicu kesalahpahaman di antara anggota kelompok. Oleh karena itu, diperlukan sebuah solusi yang dapat membantu pengguna mencatat dan mengorganisir pembagian biaya dengan mudah, tanpa menghilangkan fleksibilitas dalam penyesuaian setiap bagian tagihan.

Aplikasi *split bill* ini hadir untuk memenuhi kebutuhan tersebut dengan memberikan alat bantu sederhana yang memungkinkan pengguna mencatat, membagi, dan menyesuaikan pembagian biaya sesuai kesepakatan bersama. Dengan demikian, aplikasi ini mempermudah proses berbagi tagihan.

Seluruh pembagian biaya dicatat dengan jelas, mengurangi peluang terjadinya kesalahpahaman dan meningkatkan transparansi. Karena aplikasi ini berbasis web, pengguna dapat mengaksesnya dengan mudah melalui berbagai perangkat seperti komputer, tablet, atau *smartphone*, tanpa perlu mengunduh atau menginstal aplikasi tambahan. Hal ini memungkinkan pengguna untuk menggunakan fitur pembagian biaya kapan saja dan di mana saja selama terhubung ke internet.

Proses pengumpulan data awal dilakukan dengan metode kuesioner dalam mengumpulkan informasi tentang pengalaman dan preferensi target dari aplikasi. Kuesioner ini memiliki beberapa tujuan yaitu untuk mengetahui sejauh mana target mengalami kesulitan dalam mengelola kontribusi uang dalam kelompok, menilai ketertarikan dengan alat bantu pengelolaan, kecenderungan penggunaan aplikasi seluler, dan mendapatkan pandangan terhadap efektifitas serta kenyamanan pembagian tagihan kelompok. Kuesioner dilakukan terhadap 26 remaja berusia antara 21 sampai dengan 23 tahun, dan terdiri dari sepuluh pertanyaan dengan skala penilaian mulai dari 1 (sangat tidak setuju) hingga 5 (sangat setuju). Hasil dari kuesioner menunjukkan nilai respon rata-rata yang cukup tinggi yaitu 4.38, terkait pernyataan bahwa aplikasi pembagian tagihan dapat membuat kegiatan kelompok lebih efisien dan menyenangkan. Untuk hasil kuesioner lengkapnya dapat dilihat pada lampiran 1.

Latar belakang yang telah dijelaskan akan menjadi alasan dasar dari penelitian ini. Penelitian ini didorong oleh kesadaran akan kompleksitas pembagian biaya dalam interaksi sosial, khususnya dalam aktivitas kelompok bersama teman-teman. Masalah ini sering kali menimbulkan konflik dan ketidaknyamanan di antara individu. Oleh karena itu, pengembangan aplikasi pencatatan *Split Bill* berbasis web dianggap relevan sebagai upaya untuk mengatasi tantangan ini. Aplikasi ini diharapkan tidak hanya memberikan solusi praktis bagi pengguna, tetapi juga memiliki potensi untuk menciptakan dampak positif dengan meningkatkan transparansi pencatatan, mengoptimalkan pengelolaan waktu dan usaha dalam menghitung biaya, serta meningkatkan aksesibilitas. Dengan melihat manfaat yang dapat dihasilkan dan peluang pengembangan yang ditawarkan oleh proyek ini, penelitian ini diteruskan dengan motivasi yang kuat untuk merancang aplikasi yang relevan dan bermanfaat bagi banyak orang.

Adapun metode yang akan digunakan dalam penelitian ini adalah metode *Personal eXtreme Programming* (PXP). Metode PXP adalah model pengembangan perangkat lunak yang dirancang khusus untuk pengembang otonom, yaitu pengembang perangkat lunak yang mengimplementasikan solusi perangkat lunak tanpa menjadi bagian dari sebuah tim[1]. Dalam metode ini, setiap tahap proses

pengembangan disederhanakan agar lebih adaptif dan fleksibel. Metode ini dipilih karena PXP sangat sesuai untuk pengembangan sistem informasi dengan pemrograman tunggal, seperti pengembangan web oleh satu orang. Di mana PXP lebih mengutamakan adaptasi terhadap perubahan, mementingkan fungsional aplikasi dari pada dokumentasi, dan prinsip-prinsip Agile lainnya.

Dalam pengembangannya, aplikasi ini akan dibangun menggunakan library *React.js*. *React.js*, merupakan library *JavaScript* yang dikembangkan oleh Facebook, merupakan sebuah library *JavaScript* yang memudahkan pembuatan antarmuka pengguna yang responsif dan efisien. Dengan pendekatan komponen yang memecah antarmuka menjadi bagian-bagian yang terkelola, *React.js* memungkinkan pengembang untuk mengelola state aplikasi secara efektif dan merender tampilan secara dinamis. Menurut survei dari Statista (2023), sekitar 40.6% pengembang menggunakan *React.js*, menjadikannya salah satu framework web yang populer. Dengan dukungan yang luas dari komunitas dan kemampuannya dalam membangun aplikasi skala kecil maupun besar, *React.js* menjadi pilihan populer dalam ekosistem pengembangan web saat ini.

## 1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini antara lain :

1. Bagaimana cara mengembangkan aplikasi *Split Bill* menggunakan metode *Personal eXtreme Programming (PXP)*?
2. Bagaimana memastikan aplikasi *Split Bill* yang dibangun berjalan dengan baik?
3. Bagaimana mengukur performa aplikasi *Split Bill* yang dikembangkan untuk memastikan aplikasi berjalan secara optimal?

### 1.3 Tujuan

Adapun tujuan dari penelitian ini antara lain :

1. Mengembangkan aplikasi *Split Bill* menggunakan metode *Personal eXtreme Programming (PXP)*.
2. Menguji fungsionalitas dari aplikasi *Split Bill* dengan metode *black box Testing*.
3. Menguji performa aplikasi *Split Bill* yang dibangun dengan menggunakan *Lighthouse*.

### 1.4 Manfaat

Adapun manfaat dari penelitian ini antara lain :

1. Penelitian diharapkan berkontribusi dalam pengembangan inovasi digital dengan menyediakan solusi berbasis web yang dapat melakukan pencatatan dan pembagian biaya secara transparan.
2. Penelitian ini diharapkan menjadi landasan bagi pengembangan lebih lanjut di bidang teknologi pembagian tagihan atau fitur-fitur pendukung lainnya.
3. Penelitian diharapkan dapat menjadi manfaat dan sumber referensi mengenai metode dan *tech stack* yang digunakan dalam penelitian ini.

### 1.5 Batasan Masalah

Adapun batasan masalah pada penelitian ini antara lain :

1. Penelitian ini akan difokuskan pada pengembangan aplikasi "*Split Bill*" dengan penekanan pada fitur pencatatan saja.
2. Penelitian ini akan menggunakan *Firestore* sebagai *backend*, sehingga aplikasi akan mematuhi batasan yang telah ditetapkan oleh *Firestore* itu sendiri.

3. Aplikasi yang dikembangkan dalam penelitian ini akan bersifat berbasis web untuk meningkatkan aksesibilitas dan fleksibilitas pengguna.

## **1.6 Sistematika Penulisan**

Adapun sistematika penulisan yang digunakan pada penelitian ini antara lain :

### **BAB I PENDAHULUAN**

Berisi latar belakang, rumusan masalah, tujuan, dan manfaat penelitian.

### **BAB II TINJAUAN PUSTAKA**

Membahas alat-alat yang digunakan dan penelitian sebelumnya.

### **BAB III METODOLOGI PENELITIAN**

Menjelaskan metode Personal eXtreme Programming (PXP) yang digunakan.

### **BAB IV PEMBAHASAN**

Menyajikan hasil implementasi fitur berdasarkan iterasi dari metode PXP serta pengujian-pengujian yang dilakukan.

### **BAB V SIMPULAN DAN SARAN**

Berisi ringkasan hasil penelitian dan rekomendasi untuk pengembangan lebih lanjut.

### **DAFTAR PUSTAKA**

### **LAMPIRAN**



## II. TINJAUAN PUSTAKA

### 2.1 *Split Bill*

*Split Bill* adalah kegiatan membagi atau memisahkan tagihan dengan nominal sesuai dengan banyaknya barang atau makanan yang dibeli bersama-sama. *Split Bill* adalah istilah yang umum digunakan untuk menggambarkan situasi di mana dua orang atau lebih membagi tagihan berdasarkan jumlah pengeluaran individu [2]. Dengan metode ini, tidak perlu lagi menggunakan kalkulator untuk menghitung jumlah pembagian tagihan. *Split Bill* sering digunakan ketika kita berkumpul dengan teman-teman atau bahkan saat pertemuan pertama dengan gebetan. Metode ini memastikan bahwa setiap orang hanya membayar tagihan barang atau jasa yang mereka gunakan secara adil.

Aplikasi yang menerapkan fitur *split bill* dalam aplikasinya adalah GoPay[3]. GoPay adalah salah satu dompet digital populer di Indonesia yang menyediakan berbagai fitur untuk mempermudah transaksi keuangan penggunanya. Fitur *Split Bill* di GoPay memungkinkan pengguna untuk membagi pembayaran secara otomatis atau manual saat melakukan kegiatan bersama, seperti makan di restoran atau membayar tagihan liburan. Fitur ini terletak di menu *Split Bill* pada beranda aplikasi.

Salah satu fitur di dalam *split bill* aplikasi GoPay adalah perhitungan manual, di mana pengguna dapat mengatur jumlah pembagian tagihan secara mandiri. Melalui fitur ini, pengguna diberikan fleksibilitas untuk menentukan porsi pembayaran masing-masing anggota sesuai kebutuhan, tanpa harus mengikuti pembagian rata atau aturan baku lainnya. Hal ini serupa dengan mekanisme pembagian tagihan yang direncanakan pada aplikasi penelitian ini, yang juga

memberikan opsi pembagian secara manual sesuai porsi atau persentase yang ditentukan pengguna. Fitur ini mempermudah pengguna dalam mengelola patungan secara personal dan memberikan kontrol penuh atas proses pembagian tagihan [3].

## 2.1. UML

*Unified Modeling Language* (UML) adalah metode pemodelan visual yang digunakan sebagai alat perancangan sistem berorientasi objek. UML memungkinkan pengembang untuk menciptakan diagram yang menggambarkan aspek-aspek sistem perangkat lunak, termasuk struktur, perilaku, dan interaksi antar komponen[4]. UML menyediakan seperangkat notasi grafis dan aturan penulisan yang telah terstandarisasi, yang memungkinkan para pengembang untuk secara sistematis menggambarkan berbagai aspek dari sistem yang sedang dikembangkan. Dengan menggunakan UML, pengembang dapat menggambarkan struktur statis dan dinamis dari sistem, termasuk kelas-kelas yang terlibat, hubungan antara kelas-kelas tersebut, serta interaksi antar objek dalam sistem. Selain itu, UML juga menyediakan diagram-diagram seperti diagram kasus penggunaan, diagram kelas, diagram aktivitas, dan banyak lagi, yang membantu dalam memahami, merancang, dan berkomunikasi tentang berbagai aspek dari sistem perangkat lunak. Dengan demikian, pemahaman yang mendalam tentang UML dan penerapannya menjadi kunci dalam pengembangan perangkat lunak yang efektif dan efisien. Beberapa jenis diagram yang berguna sebagai alat bantu dalam perencanaan dan pemahaman model antara lain :

### a. *Use Case Diagram*

Diagram ini menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem[5]. *Use case diagram* membantu mengidentifikasi fungsionalitas sistem dari sudut pandang pengguna.

### b. *Class Diagram*

Diagram ini menggambarkan struktur kelas (objek) dalam sistem, termasuk atribut dan hubungan antar kelas[5]. *Class diagram* berfungsi menunjukkan kelas-kelas dalam sistem beserta atribut-atributnya dan menggambarkan hubungan antara kelas.

### c. *Activity Diagram*

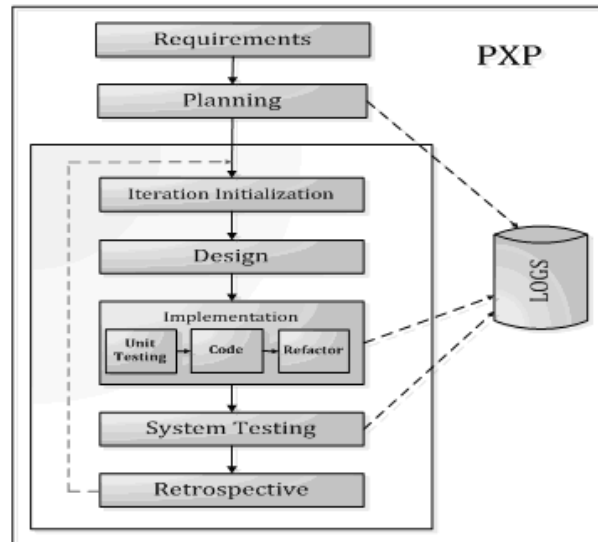
*Activity diagram* menggambarkan alur kerja atau proses dalam sistem[5]. Diagram ini berfungsi menunjukkan aktivitas-aktivitas yang terjadi dalam system dan menggambarkan urutan langkah-langkah dan keputusan yang diambil.

## 2.2. *Personal eXtreme Programming*

*Personal eXtreme Programming* (PXP) adalah sebuah metodologi pengembangan perangkat lunak yang ditujukan untuk pengembang mandiri yang ingin menerapkan prinsip-prinsip *Extreme Programming* (XP) dalam proyek mereka. PXP merupakan modifikasi dari *Personal Software Process* (PSP) yang bertujuan untuk meringankan proses pengembangan perangkat lunak dan membuatnya lebih mudah diikuti, sekaligus menjaga prinsip dasar PSP. PXP juga memperluas PSP dengan praktik pengembangan yang terbukti efisien dari XP, seperti perencanaan proyek yang adaptif, pengujian berkelanjutan, dan refaktorasi kode[1].

PXP terdiri dari tujuh fase, yaitu *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *system Testing*, dan *retrospective*. Pada fase *requirements*, pengembang mengumpulkan dan menganalisis kebutuhan pelanggan, serta membuat spesifikasi fungsional dan non-fungsional perangkat lunak. Pada fase *planning*, pengembang membuat rencana proyek yang mencakup estimasi waktu, biaya, dan sumber daya, serta menentukan prioritas fitur yang akan dikembangkan. Pada fase *iteration initialization*, pengembang memilih fitur yang akan dikerjakan dalam iterasi saat ini, dan membuat daftar tugas yang harus diselesaikan. Pada fase *design*, pengembang membuat desain rinci perangkat lunak, termasuk diagram kelas, diagram urutan, dan diagram komponen, serta menentukan standar kode yang akan digunakan. Pada fase *implementation*, pengembang menulis kode perangkat lunak sesuai dengan standar kode yang telah ditetapkan, dan melakukan pengujian unit setiap kali menambah atau mengubah kode. Pada fase *system Testing*, pengembang melakukan pengujian integrasi dan sistem untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan

kebutuhan pelanggan, serta memperbaiki bug yang ditemukan. Pada fase *retrospective*, pengembang melakukan evaluasi terhadap proses dan produk perangkat lunak, serta mencatat data dan metrik yang relevan untuk meningkatkan kinerja di masa depan.



Gambar 2. 2 Fase dalam PXP [1]

Selain itu, metode *Personal eXtreme Programming* (PXP) juga menawarkan keuntungan signifikan dengan memungkinkan pengembang untuk bekerja secara mandiri dengan konsentrasi yang tinggi, fleksibilitas jadwal, dan tanggung jawab penuh atas proyek mereka.

## 2.3. Tools yang Digunakan Selama Pengembangan

### 2.3.1 Visual Studio Code

Salah satu editor teks yang banyak digunakan oleh para pengembang perangkat lunak adalah *Visual Studio Code* (VSCode). *Visual Studio Code* (VSCode) adalah *Integrated Development Environment* (IDE) yang digunakan untuk berbagai macam bahasa pemrograman[6]. Editor teks ini mendukung berbagai macam bahasa pemrograman, termasuk *JavaScript*. VS Code memiliki banyak kelebihan, seperti gratis dan *open source*, ringan dan cepat, serta mudah dikustomisasi. Selain itu, VS Code juga memiliki banyak ekstensi dan fitur yang dapat membantu proses

pengembangan perangkat lunak, seperti *debugging*, *linting*, *code completion*, *refactoring*, *version control*, dan lain-lain. VS Code juga selalu melakukan pembaruan versi setiap bulannya, sehingga menjamin kualitas dan kinerja editor teks ini. Alasan-alasan ini membuat VS Code menjadi pilihan yang tepat untuk pengembangan perangkat lunak.

### 2.3.2 *React Js*

*React JS* adalah pustaka *JavaScript* yang digunakan untuk merancang antarmuka aplikasi web yang *responsive* [7]. Pustaka ini difokuskan pada pembuatan UI yang deklaratif, memungkinkan pengembang mendefinisikan tampilan UI sesuai dengan berbagai keadaan dan memungkinkan *React* untuk mengelola perubahan tampilan tersebut secara efisien. *React* menggunakan komponen sebagai konsep inti dalam pengembangannya, memungkinkan pengguna untuk memecah aplikasi menjadi unit-unit yang lebih kecil, mudah dikelola, dan dapat digunakan kembali. Ini mempermudah pengembangan aplikasi yang skalabel dan memungkinkan pengembang untuk memisahkan logika aplikasi dari UI, meningkatkan keterbacaan kode, dan pemeliharaan.

Ada beberapa alasan yang mendukung penggunaan *React* dalam pengembangan aplikasi web. Pertama, *React* memiliki performa tinggi, terutama dalam mengupdate UI, berkat penggunaan konsep Virtual DOM yang memungkinkan perubahan pada UI diproses secara efisien. Kedua, *React* memiliki ekosistem yang luas dengan banyak pustaka pendukung yang membuat pengembangan lebih cepat dan mudah. Sebagai contoh, *React Router* digunakan untuk manajemen rute dalam aplikasi web, *Redux* membantu dalam manajemen keadaan, dan banyak ekstensi lainnya yang dapat diintegrasikan untuk meningkatkan fungsionalitas. Selain itu, komunitas *React* yang besar menyediakan dokumentasi, tutorial, dan dukungan yang melimpah, sehingga pengembang dapat dengan mudah memecahkan masalah yang muncul selama pengembangan. Dengan platform yang kuat, fleksibel, dan efisien untuk mengembangkan aplikasi atau situs web, *React JS* adalah pilihan yang sangat baik [8].

### 2.3.3 *Firestore*

*Firestore* adalah layanan *Backend as a Service* (BaaS) yang sekarang menjadi bagian dari *Google Cloud Platform*[9]. Ini menyediakan beragam layanan dan alat yang kuat untuk membantu pengembang membangun aplikasi web dan seluler yang berkualitas tinggi dengan cepat dan efisien. Salah satu aspek utama dari *Firestore* adalah basis data *real-time* (*Realtime Database*) yang memungkinkan aplikasi untuk menyimpan dan mengakses data secara real-time, sehingga memungkinkan pengembangan aplikasi kolaboratif dan berbasis peristiwa. *Firestore* juga menyediakan layanan otentikasi yang aman dan mudah digunakan, yang memungkinkan pengguna masuk ke aplikasi dengan akun *Google*, *Facebook*, atau platform lainnya, atau menggunakan otentikasi khusus aplikasi. Selain itu, *Firestore* juga memiliki fitur hosting yang memungkinkan pengembang untuk dengan mudah menyimpan dan mengelola konten web statis serta berbagai alat analitik dan pelaporan untuk melacak kinerja aplikasi dan interaksi pengguna.

Ada beberapa alasan yang mendukung penggunaan *Firestore* dalam pengembangan aplikasi. Pertama, *Firestore* adalah platform berbasis awan yang memiliki skalabilitas tinggi, sehingga aplikasi dapat dengan mudah mengatasi peningkatan jumlah pengguna dan data. Kedua, *Firestore* memiliki otentikasi yang kuat dan aman, yang memungkinkan pengembang untuk mengintegrasikan sistem otentikasi yang andal ke dalam aplikasi mereka tanpa perlu membangunnya dari awal. Ketiga, *Firestore* menyediakan beragam layanan tambahan seperti penyimpanan berkas, pesan push, dan analitik yang mudah diintegrasikan, menghemat waktu dan upaya pengembangan. Selain itu, *Firestore* juga memiliki dokumentasi yang kaya, komunitas pengguna yang aktif, dan dukungan teknis yang baik, sehingga pengembang dapat dengan mudah memecahkan masalah dan memahami cara menggunakan platform ini secara efektif. *Firestore* memiliki pustaka yang lengkap untuk sebagian besar platform web dan mobile, dan dapat diintegrasikan dengan framework seperti *node*, *java*, dan *JavaScript*[10]. Secara keseluruhan, *Firestore* cocok untuk aplikasi yang membutuhkan skala besar, akses

data cepat, dan sinkronisasi real-time, tapi kurang cocok untuk aplikasi yang membutuhkan operasi basis data relasional yang kompleks. Untuk integrasi, *firebase* menyediakan layanan yang mendukung pengembangan aplikasi web dan mobile. Aplikasi yang akan dibangun pada penelitian ini menggunakan *reactjs*, integrasi *firebase* dengan *reactjs* sangat cocok dan cukup mudah dilakukan. Dikarenakan *firebase* menawarkan *Software Development Kit* (SDK) yang mendukung *reactjs* secara langsung.

#### 2.3.4 *Lighthouse*

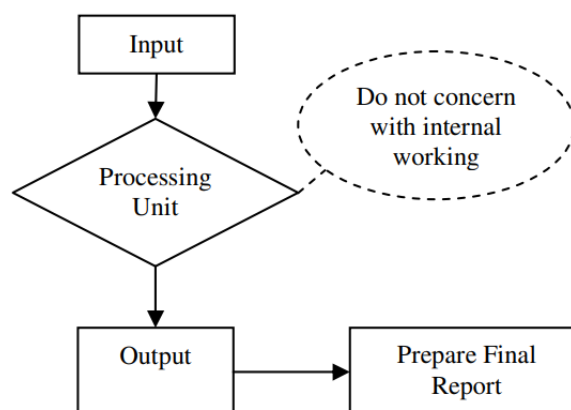
*Lighthouse* merupakan alat otomatis bersifat *open-source* yang dapat membantu dalam meningkatkan kualitas halaman web yang dikembangkan oleh *Google* pada pertengahan tahun 2018. *Lighthouse* digunakan sebagai alat pengukuran dan audit untuk performa kualitas halaman web, aksesibilitas, aplikasi web progresif, dan banyak lagi. *Lighthouse* akan melakukan serangkaian pemeriksaan untuk mengidentifikasi masalah pada halaman dan kemudian memberikan laporan mengenai sejauh mana kinerja dari halaman tersebut[11].

Ada beberapa alasan yang mendukung penggunaan *Lighthouse* dalam pengembangan aplikasi. Pertama, *Lighthouse* memberikan analisis komprehensif dalam beberapa kategori penting, seperti performa, aksesibilitas, SEO dan *best practices* sehingga dapat membantu pengembang untuk memahami aspek mana yang perlu ditindaklanjuti dan dioptimalkan. Kedua, *Lighthouse* memberikan skor hasil dalam bentuk diagram untuk setiap kategori penting dan menyertakan saran atau panduan yang relevan untuk perbaikan. Misalnya, dalam kategori performa, alat ini akan mengidentifikasi adanya masalah seperti waktu loading, gambar yang tidak teroptimasi, atau *JavaScript* yang besar. Ketiga, *Lighthouse* tersedia langsung sebagai ekstensi pada *Chrome DevTools*, sehingga memudahkan pengembang melakukan audit web tanpa perlu alat tambahan.

## 2.4 *Black box Testing*

*Black box Testing* merupakan salah satu metode pengujian perangkat lunak yang dilakukan tanpa memerhatikan struktur internal kode atau logika implementasi program. Dalam *black box Testing*, fokus utama adalah pada input dan output yang dihasilkan oleh perangkat lunak, serta perilaku fungsionalnya[12]. Para pengujian tidak memerlukan pengetahuan detail tentang bagaimana perangkat lunak bekerja di dalamnya. Sebagai gantinya, pengujian berfokus pada pemahaman terhadap spesifikasi fungsi atau fitur yang harus dipenuhi oleh perangkat lunak tersebut. Pendekatan ini memungkinkan tim pengujian untuk memeriksa apakah perangkat lunak beroperasi sesuai dengan harapan dan kebutuhan pengguna tanpa memperhatikan implementasi internalnya.

Salah satu keuntungan utama dari *black box Testing* adalah bahwa pengujian dapat dilakukan dari perspektif pengguna akhir, sehingga memungkinkan untuk mendeteksi masalah yang mungkin tidak terlihat jika hanya dilihat dari dalam kode. Hal ini membantu dalam menemukan bug atau kelemahan dalam perangkat lunak yang mungkin terlewatkan selama proses pengembangan. Selain itu, *black box Testing* juga memfasilitasi kerja tim yang terdiri dari pengembang dan pengujian, karena pengujian dapat dilakukan tanpa ketergantungan pada pengetahuan khusus tentang kode yang dimiliki oleh pengembang. Proses kerja dari teknik pengujian ini dapat dilihat pada gambar 2.1.



Gambar 2.1 Proses kerja dari teknik pengujian *Black box*[13]



## 2.4 *Heuristic Evaluation*

*Heuristic Evaluation* atau Evaluasi heuristik (HE) dapat diartikan sebagai pengujian suatu sistem yang pada dasarnya dilakukan dengan hanya melihat antarmuka pengguna dan kemudian evaluator akan mengemukakan pendapatnya tentang baik atau buruknya antarmuka tersebut sehingga secara singkat Evaluasi heuristik ini adalah metode yang menguji tingkat dari kegunaan (*usability*) dari suatu aplikasi. Sekalipun siapa pun dapat melakukan pengujian ini, evaluator yang ahli dan memiliki gelar serta pengalaman dalam bidang terkait merupakan rekomendasi terbaik untuk melakukan pengujian ini dibandingkan dengan evaluator pemula [14]. Untuk menentukan jumlah evaluator, rekomendasi terbaik untuk melakukan pengujian evaluasi heuristik adalah dengan tiga sampai lima evaluator [15]. Setiap evaluator akan melakukan proses pengujian evaluasi heuristik secara individual dan hanya diperbolehkan untuk dapat berkomunikasi satu sama lain setelah pengujian dan evaluasi selesai. Hal ini sangat penting dilakukan untuk dapat menghindari pengaruh satu evaluator terhadap evaluator lainnya.

Keuntungan menggunakan metode pengujian evaluasi heuristik antara lain biaya yang rendah, tidak memerlukan perencanaan yang rumit, dapat digunakan selama proses pengembangan, dan pengujian yang didasarkan pada sudut pandang penguji atau bersifat subjektif. *Output* yang dihasilkan dari pengujian menggunakan metode ini meliputi daftar masalah kegunaan pada antarmuka pengguna dan referensi sebagai sumber acuan dari prinsip-prinsip yang digunakan. Evaluasi heuristik tidak memberikan pendekatan sistematis untuk melakukan perbaikan terhadap masalah kegunaan (*usability*), namun perbaikannya cukup mudah dilakukan karena masalah yang ditemukan mengacu pada prinsip kegunaan (*usability*) yang dilanggar. Dari segi *severity ratings*, metode pengujian evaluasi heuristik menggunakan metode skala tunggal (*single scale*), yaitu : yaitu: 0 untuk tidak ada masalah, 1 untuk masalah kosmetik (*cosmetic problem*) yang tidak mendesak, 2 untuk masalah minor dengan prioritas rendah, 3 untuk masalah utama yang menjadi prioritas, dan 4 untuk masalah kritis (*usability catastrophe*) yang harus diselesaikan sebelum aplikasi dirilis [16].

Dalam proses pengujian menggunakan evaluasi heuristik, terdapat sepuluh prinsip yang sering dijadikan acuan, yang dikenal sebagai *Top Ten Usability Heuristics*. Prinsip-prinsip tersebut meliputi: visibilitas status sistem (*visibility of system status*), kesesuaian antara sistem dan dunia nyata (*match between system and the real world*), kontrol dan kebebasan pengguna (*user control and freedom*), konsistensi dan standar (*consistency and standards*), pencegahan kesalahan (*error prevention*), pengakuan daripada pengingatan (*recognition rather than recall*), fleksibilitas dan efisiensi penggunaan (*flexibility and efficiency of use*), desain estetis dan minimalis (*aesthetic and minimalist design*), membantu pengguna mengenali, mendiagnosis, dan memulihkan dari kesalahan (*help user recognize, diagnose, and recover from errors*), serta bantuan dan dokumentasi (*help and documentation*) [17].

## 2.5. Penelitian Terkait

Penelitian mengenai sistem informasi berbasis web oleh Suprpto dkk., pada tahun 2020 [18] membahas mengenai pengembangan sistem informasi berbasis web untuk keperluan penjualan dan pelelangan di Ricardo Corner MLG. Penelitian ini bertujuan untuk mengembangkan sistem informasi yang memudahkan proses penjualan dan pelelangan di Ricardo Corner MLG. Metode yang digunakan adalah *Personal eXtreme Programming* (PXP). Dari penelitian ini disimpulkan bahwa dengan metode PXP, dapat menghasilkan sistem informasi berbasis web yang memungkinkan pengguna untuk melakukan transaksi penjualan dan pelelangan secara efisien.

Penelitian berikutnya yang menggunakan metode PXP diteliti oleh Sandika dan Kurniawan pada tahun 2014[19] membahas pengembangan sistem informasi yang berfokus pada kualitas air sungai di Provinsi Lampung. Penelitian ini bertujuan untuk mengembangkan sistem informasi yang memantau kualitas air sungai di wilayah Provinsi Lampung. Penelitian ini menghasilkan sistem informasi berbasis web yang memungkinkan pemantauan dan evaluasi kualitas air sungai di Provinsi Lampung. Dari penelitian ini didapat bahwa metode PXP dapat

menghasilkan proses pengembangan yang lebih cepat dan efisien karena fase-fasenya tidak terlalu banyak.

Kemudian terdapat penelitian yang dilakukan oleh Asri dan Setiawan pada tahun 2017[20]. Pada penelitian ini penulis mencari model pendekatan Agile yang cocok untuk mengembangkan sistem informasi PKL online. Hasil penelitian menunjukkan bahwa model pendekatan PXP cocok digunakan untuk mengembangkan sistem informasi PKL online. Dari penelitian ini juga dapat diambil bahwa dengan menggunakan PXP, dokumentasi yang dihasilkan akan lebih ringkas ditambah pengerjaan akan lebih singkat.

Selanjutnya terdapat penelitian oleh Panjaitan dan Pakpahan pada tahun 2021 [21] yang membahas tentang pengembangan sebuah sistem *e-reporting* yang dapat memudahkan proses pelaporan transaksi oleh pegawai kantor pusat Gereja Masehi Advent Hari Ketujuh Indonesia Wilayah Barat ke departemen keuangan. Sistem ini menggunakan *React JS* sebagai *library front-end* dan *Firebase* sebagai platform back-end. Di penelitian ini dijelaskan *firebase* digunakan untuk mempercepat dan mempermudah pengembangan, dan *React JS* digunakan karena memberikan skalabilitas, kecepatan, dan kemudahan. Dan penelitian ini juga menggunakan *black box Testing* sebagai metode untuk menguji sistem yang telah dikembangkan.

Penelitian lainnya yang menggunakan *React JS* dilakukan oleh Huda dan Fernando pada tahun 2021[22]. Penelitian ini bertujuan untuk mengembangkan sistem informasi e-ticketing yang memudahkan penjualan tiket event musik di wilayah Lampung. Aplikasi yang dibangun merupakan aplikasi berbasis web, metode pengembangan yang digunakan adalah *GRAPPLE (Guidilines For Rapid Application Engineering)*. Dari hasil penelitian ini didapat bahwa dengan menggunakan *React JS*, dapat menghasilkan sistem berupa aplikasi berbasis web yang memungkinkan pengguna untuk melakukan jual-beli tiket event musik secara online.

Ditemukan lagi penelitian yang menggunakan *black box Testing* sebagai salah satu metode dalam menguji sistem yaitu penelitian oleh Nursaid dkk., pada

tahun 2020[23]. Penelitian ini bertujuan untuk mengembangkan sistem informasi yang memudahkan pengelolaan persediaan barang pada studi kasus penelitian. Untuk metode pengembangan digunakan metode *prototyping*. Pada hasil penelitian ini *black box Testing* berperan dalam pengujian validasi, di mana total 15 dari 15 kebutuhan fungsional yang diuji dalam penelitian memiliki hasil yang memenuhi ekspektasi.

Penelitian terkait selanjutnya oleh Novrina dkk., tahun 2022[24] yang membahas pengembangan aplikasi jual-beli online menggunakan *React Native* dan *Firebase*. Penelitian ini bertujuan untuk mengembangkan aplikasi jual-beli online yang memanfaatkan teknologi *React Native* dan *Firebase*. Yang menghasilkan aplikasi yang memungkinkan pengguna untuk melakukan jual-beli item game secara online. Penelitian menggunakan teknologi *React Native* karena kemampuan *React Native* dalam menghasilkan aplikasi seluler lintas platform dengan kode *JavaScript* yang sama. Dan *firebase* memudahkan pengelolaan data dan otentikasi pengguna tanpa harus membangun *backend* dari awal.

Penelitian oleh Prihandoyo tahun 2018 membahas tentang pengembangan sistem informasi akademik di Politeknik Harapan Bersama Tegal dengan menggunakan model *Unified Modeling Language* (UML) sebagai model pengembangan. Metode pengembangan yang digunakan tidak secara spesifik disebutkan, tetapi UML memberikan kerangka kerja yang komprehensif untuk merancang dan menggambarkan sistem. Jurnal ini juga membahas kelebihan UML yang dapat memudahkan developer sistem dalam merancang sistem yang berorientasi pada objek dan memberikan optimisasi layanan informasi[25].

Selanjutnya penelitian oleh Setiaji dan Sastra pada tahun 2021[26]. Tujuan dari penelitian ini adalah untuk merancang sistem informasi penggajian yang efisien dan akurat. Dengan menggunakan UML (*Unified Modeling Language*), penelitian ini bertujuan untuk menggambarkan secara visual elemen-elemen sistem penggajian, termasuk entitas, hubungan, dan interaksi di antara mereka. Metode yang digunakan dalam penelitian adalah *Iterative Waterfall Model*. Hasil dari penelitian ini adalah sistem informasi penggajian yang terstruktur dengan baik,

didokumentasikan dengan diagram UML yang jelas, dan dapat membantu manajemen dalam mengelola gaji karyawan secara efisien dan akurat.

Kemudian terdapat penelitian yang memanfaatkan UML oleh Sumiati dkk. tahun 2021[27]. Penelitian ini bertujuan untuk memberikan kontribusi keilmuan berupa rekomendasi dalam perancangan sistem informasi persewaan alat pesta menggunakan pemodelan UML. Dengan menggunakan dua jenis metode penelitian, yaitu studi lapangan, studi kepustakaan dan implementasi. Hasilnya dengan menggunakan UML untuk perancangan perangkat lunak, sistem informasi yang dibangun masih dapat dikategorikan cukup efektif dalam penggunaannya.

Tabel 2. 1 Penelitian Terkait

No	Peneliti	Metode / Alat yang Digunakan	Hasil
1	Suprpto dkk.[18]	<i>Personal eXtreme Programming (PXP)</i>	Pengembangan sistem informasi berbasis web untuk penjualan dan pelelangan di Ricardo Corner MLG. Metode PXP memungkinkan transaksi penjualan dan pelelangan secara efisien.
2	Sandika dan Kurniawan 2014 [19]	<i>Personal eXtreme Programming (PXP)</i>	Pengembangan sistem informasi pemantauan kualitas air sungai di Provinsi Lampung. Metode PXP mempercepat proses pengembangan dengan fase yang lebih efisien.
3	Asri dan Setiawan 2017 [20]	Model pendekatan PXP	Pengembangan sistem informasi PKL online. PXP cocok digunakan dan menghasilkan dokumentasi yang lebih ringkas serta

No	Peneliti	Metode / Alat yang Digunakan	Hasil
			pengerjaan yang lebih singkat.
4	Panjaitan dan Pakpahan 2021 [21]	<i>React JS (front-end) dan Firebase (back-end)</i>	Pengembangan sistem e-reporting untuk pelaporan transaksi oleh pegawai Gereja Masehi Advent Hari Ketujuh Indonesia Wilayah Barat ke departemen keuangan. <i>Firebase</i> mempercepat pengembangan, sementara <i>React JS</i> memberikan skalabilitas dan kecepatan. <i>Black box Testing</i> digunakan untuk menguji sistem.
5	Huda dan Fernando 2021 [22]	<i>React Js</i>	Dengan menggunakan <i>React JS</i> , dapat menghasilkan sistem berupa aplikasi berbasis web yang baik.
6	Nursaid dkk. 2020[23]	<i>black box Testing</i>	Pengembangan sistem informasi pengelolaan persediaan barang. <i>Black box Testing</i> berperan dalam pengujian validasi dan memastikan hasil memenuhi ekspektasi.
7	Novrina dkk. 2022 [24]	<i>React Native dan Firebase</i>	Pengembangan aplikasi jual-beli online. <i>React Native</i> memungkinkan aplikasi seluler lintas platform dengan kode <i>JavaScript</i> yang sama, sementara <i>Firebase</i> memudahkan pengelolaan

No	Peneliti	Metode / Alat yang Digunakan	Hasil
			data dan otentikasi pengguna.
8	Prihandoyo 2018 [25]	<i>Unified Modeling Language (UML)</i>	Pengembangan sistem informasi akademik di Politeknik Harapan Bersama Tegal. UML memberikan kerangka kerja komprehensif untuk merancang dan menggambarkan sistem, serta memudahkan pengembang dalam merancang sistem berorientasi pada objek.
9	Setiaji dan Sastra 2021[26]	UML	Dengan diagram UML yang jelas, hasil dari penelitian ini adalah sistem informasi penggajian yang terstruktur dengan baik dan dapat membantu manajemen dalam mengelola gaji karyawan secara efisien dan akurat
10	Sumiati dkk. 2021 [27]	UML	Dengan menggunakan UML untuk perancangan perangkat lunak, sistem informasi yang dibangun masih dapat dikategorikan cukup efektif dalam penggunaannya

Dari tabel 2.1 di atas, kemudian dipilih metode dan alat untuk mengembangkan aplikasi *Split Bill*. Pertama, penelitian ini akan menggunakan *Metode Personal eXtreme Programming (PXP)* karena metode PXP dapat mempercepat proses pengembangan dengan fase yang efisien, dokumentasi yang lebih ringkas, dan pengerjaan yang lebih singkat. Kemudian akan digunakan UML

guna memberikan kerangka kerja komprehensif untuk merancang dan menggambarkan sistem. Dalam membangun front-end aplikasi, digunakan library *React JS* karena memberikan skalabilitas dan kecepatan. Untuk *database* akan digunakan *Firebase* karena menawarkan kemudahan dalam pengelolaan data dan otentikasi pengguna sehingga proses pengembangan akan menjadi lebih cepat. Terakhir akan digunakan *Black box Testing* untuk pengujian fungsionalitas sistem dan memastikan hasil memenuhi ekspektasi.



### III. METODOLOGI PENELITIAN

#### 3.1.1 Waktu dan Tempat

Penelitian ini dilakukan di lingkungan laboratorium teknik komputer Universitas Lampung, dengan waktu penelitian dimulai dari bulan Juli 2024 sampai dengan Desember 2024 dengan rincian waktu pada tabel 3.1 sebagai berikut :

Tabel 3. 1 Jadwal Kegiatan Penelitian

No	Kegiatan	Jul	Agu	Sep	Okt	Nov	Des
1	Pengumpulan data						
2	<i>Requirements</i>						
3	Perencanaan						
5	Desain						
6	Implementasi						
7	<i>Testing</i>						
8	Restropektif						
9	Iterasi						
10	Pengujian akhir						
11	Analisis dan Penyusunan Laporan						

### 3.2 Alat dan bahan

Berikut alat dan bahan yang digunakan dalam penelitian :

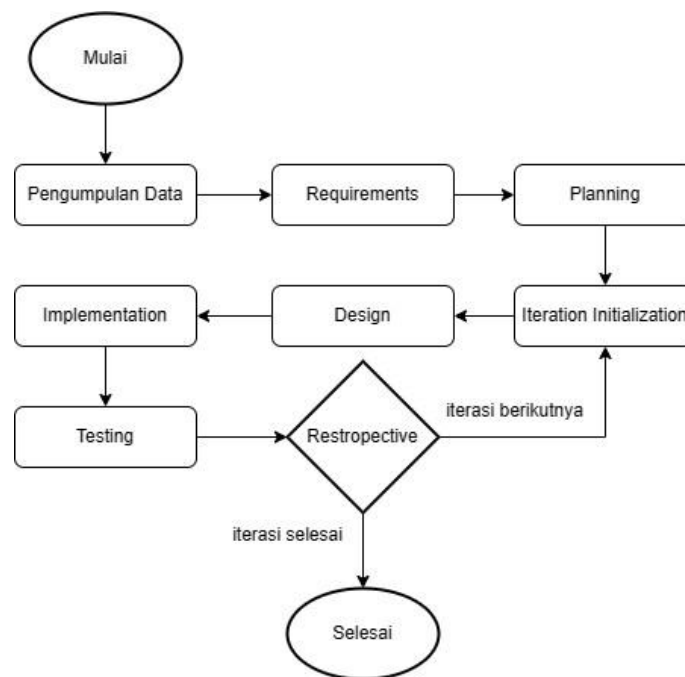
Tabel 3. 2 Alat dan Bahan Penelitian

<b>Nama Alat / Bahan</b>	<b>Spesifikasi</b>	<b>Kegunaan</b>
Laptop	Intel Core i5 Generasi ke-10	Melakukan penelitian secara umum
<i>Visual Studio Code</i>	Versi 1.87	Menulis dan melakukan debugging kode
<i>Node JS</i>	Versi 21.7.1	Pengembangan sisi server
<i>React JS</i>	Versi 18.0	Membangun antarmuka pengguna
<i>Firebase</i>	Versi 11.11.1	Integrasi dengan <i>database</i> dan otentikasi pengguna
<i>Google Chrome</i>	Versi 122.0.6261.95	Pengujian aplikasi web
Ms. Word	Ms Office <i>Home</i> and Student 2019	Dokumentasi dan komunikasi
Canva	<i>online</i>	Mendesain aplikasi
Draw.io	<i>online</i>	Membuat diagram dan flowchart

Data Kuesioner		Data awal untuk menentukan fungsi aplikasi,
----------------	--	---

### 3.3 Tahapan Penelitian

Metodologi penelitian yang digunakan pada penelitian ini adalah *Personal eXtreme Programming (PXP)*. Dengan menggunakan metode PXP, tahapan penelitian akan menjadi seperti diagram alir dibawah ini.



Gambar 3. 1 Diagram alir tahapan penelitian

Gambar 3.1 di atas mencakup tahapan-tahapan mulai dari awal pengumpulan data hingga tahapan dalam metode yang digunakan dalam penelitian. Dibuatnya diagram alir bertujuan agar tahapan-tahapan yang dilakukan dalam penelitian dapat dengan lebih mudah dibaca dan dipahami.

### 3.3.1 *Requirements*

Pada tahap *requirements* ini pertama dilakukan studi literatur tentang tren terkini dalam industri pengembangan aplikasi. Hal ini untuk mendapatkan pemahaman yang lebih mendalam tentang konsep dan teori tentang pengembangan aplikasi, menentukan metode pengembangan yang efektif, dan menentukan teknologi yang digunakan dalam pengembangan. Metode yang digunakan dalam studi literatur ini meliputi pencarian dan analisis terhadap publikasi ilmiah, jurnal akademik, artikel blog, serta sumber daya online lainnya yang relevan.

Setelah itu, digunakan metode kuesioner untuk mengumpulkan data awal. Kuesioner disebar kepada target pengguna aplikasi yang merupakan remaja berusia 21 sampai dengan 23 tahun untuk mengumpulkan data tentang preferensi pengguna. Dengan menggunakan skala penilaian dari 1 hingga 5 (sangat tidak setuju hingga sangat setuju), hasil kuesioner tersebut kemudian akan dianalisis untuk memahami kebutuhan pengguna. Hasil kuesioner dapat dilihat pada bagian lampiran.

Berdasarkan data awal dari jawaban kuesioner, beberapa fungsi potensial aplikasi kemudian diidentifikasi dan dikembangkan. Dari hasil identifikasi tersebut, dirumuskanlah kebutuhan fungsional untuk aplikasi *SplitBill*. Hal ini termasuk menentukan fitur-fitur utama yang harus ada dalam aplikasi dan dapat memenuhi kebutuhan pengguna.

Untuk memahami Untuk memodelkan berbagai aspek dalam sistem ini akan digunakan *Unified Modeling Language (UML)*. Dalam PXP, di mana pengembang hanya satu orang, UML bukan menjadi prioritas utama. Namun, UML tetap memiliki manfaat untuk memahami, merencanakan, dan mengorganisasi pengembangan perangkat lunak dengan lebih baik. *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram* adalah tiga jenis diagram yang digunakan untuk memodelkan berbagai aspek sistem ini.

### 3.3.2 *Planning*

Pada tahap *planning*, strategi yang direncanakan mencakup penentuan fitur prioritas, durasi tiap iterasi, dan tahapan yang akan dilakukan dalam setiap iterasi

untuk memastikan proses pengembangan berjalan terorganisir dan sesuai target waktu. Fitur utama seperti autentikasi pengguna dan pembuatan *split bill* akan menjadi fokus awal sebagai fondasi aplikasi, sedangkan fitur tambahan dirancang untuk dikerjakan di iterasi berikutnya. Durasi tiap iterasi akan disesuaikan dengan tingkat kompleksitas fitur yang dikerjakan. Setiap iterasi terdiri atas empat tahapan utama, yaitu inialisasi untuk menentukan apa yang akan dilakukan pada iterasi tersebut, desain untuk merancang antarmuka pengguna, implementasi untuk merealisasikan desain menjadi fungsi aplikasi, dan pengujian menggunakan metode *black box testing* untuk memastikan fitur berfungsi. Selain itu, perencanaan juga mencakup penetapan tujuan iterasi, tugas yang perlu diselesaikan, alokasi waktu untuk masing-masing tugas, dan hasil iterasi yang diharapkan.

### **3.3.3 Design**

Dalam menerapkan *Personal eXtreme Programming* (PXP) pada bangun aplikasi *Split Bill*, tahap desain ini akan dilakukan dengan membuat *low fidelity prototype* dari aplikasi. *Low fidelity prototype* adalah representasi dasar dan sederhana dari suatu produk atau desain. Pada tahap ini, fokus utamanya adalah pada fungsionalitas dan interaksi pengguna, sementara aspek estetika visual diabaikan. Dengan demikian, desain *low fidelity* akan dibuat dengan sederhana, mudah dimengerti, dan mudah diimplementasikan. Menerapkan *low fidelity design* memungkinkan untuk iterasi cepat dan adaptasi terhadap perubahan kebutuhan, yang mana ini selaras dengan tujuan digunakannya metode pengembangan PXP.

### **3.3.4 Implementation**

Setelah tahap desain, selanjutnya adalah tahap *implementation* atau implementasi. Desain yang digunakan sebelumnya adalah *prototype low-fidelity design*, sehingga memungkinkan adanya perubahan atau penyesuaian selama implementasi. Fokus utama tahap ini adalah mengubah desain menjadi fitur yang dapat digunakan dalam aplikasi, menggunakan teknologi dan alat yang telah direncanakan sebelumnya.

### 3.3.5 *Testing*

Tahap *testing* atau pengujian akan dilakukan setelah implementasi selesai. *Testing* dilakukan dengan menguji pada setiap perubahan yang dilakukan dalam setiap iterasi. Dalam penerapannya tahap pengujian ini menggunakan pendekatan pengujian *black box*. Menggunakan pendekatan *black box* berarti pengujian akan fokus pada fungsionalitas aplikasi, dengan memperhatikan pada masukan dan keluaran yang diharapkan dari sistem. Menggunakan pendekatan *black box Testing* dalam setiap iterasi, dapat memastikan bahwa aplikasi tetap berkinerja dengan baik dan memenuhi kebutuhan pengguna dengan baik. Jika tidak ditemukan masalah atau error, iterasi akan dilanjutkan ke tahap berikutnya. Namun, jika terdapat masalah atau error selama pengujian, maka akan kembali ke tahap implementasi untuk memperbaiki masalah tersebut sebelum melanjutkan ke iterasi selanjutnya.

### 3.3.6 *Iteration*

Iterasi atau *iteration* dilakukan secara berulang sesuai rencana pada tahap *planning* selama proses pengembangan aplikasi. Setiap iterasi direncanakan memiliki durasi yang singkat, seperti satu minggu atau dua minggu, tetapi dapat disesuaikan sesuai dengan kebutuhan dan kesulitan sebuah tugas dalam iterasi tersebut. Iterasi ini mencakup inisialisasi, desain, implementasi, pengujian, dan retrospektif. Jumlah iterasi akan dilaksanakan sesuai dengan rencana pada tahap *planning*, namun jika diperlukan penyesuaian, jumlah iterasi dapat bertambah untuk menerapkan penyesuaian tersebut.

### 3.3.7 *Restropective*

Tahap retrospektif dilakukan dengan mengadopsi pendekatan introspektif dengan cara merefleksikan iterasi pengembangan yang telah berlangsung, meninjau pencapaian, mengidentifikasi kendala yang dihadapi, dan memungkinkan adanya

perbaikan. serta merumuskan rencana untuk meningkatkan proses pengembangan di masa mendatang. Tahap retrospektif ini akan dilalui dengan refleksi pribadi dengan tujuan untuk terus meningkatkan kualitas dan efisiensi pengembangan. Setelah semua iterasi telah selesai, akan diadakan pengujian secara keseluruhan.

## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan, didapat kesimpulan sebagai berikut:

1. Aplikasi *split bill* dalam penelitian ini berhasil dikembangkan menggunakan metode *Personal Extreme Programming (PXP)* dalam lima iterasi awal dengan total waktu pengerjaan selama 70 hari, sesuai estimasi yang direncanakan pada tahap *planning*. Namun, adanya evaluasi *heuristik* yang dilakukan memerlukan perbaikan tambahan, sehingga iterasi bertambah satu dan proses pengembangan selesai dalam waktu tambahan 4 hari.
2. Hasil pengujian dengan metode *black box testing* menunjukkan bahwa aplikasi *split bill* yang dikembangkan telah berfungsi dengan baik. Pengujian mencakup berbagai fitur utama, seperti pembuatan akun, proses autentikasi, pembuatan *split bill*, penerapan metode pembagian tagihan, dan akses *split bill* oleh pengguna tanpa *login*. Seluruh skenario pengujian berhasil diselesaikan dengan baik, sehingga aplikasi dinyatakan sesuai dengan standar yang telah dirancang.
3. Berdasarkan hasil uji performa menggunakan *Lighthouse* pada pengujian akhir, aplikasi *Split Bill* menunjukkan hasil yang sangat baik di lingkungan produksi. Skor performa aplikasi mencapai 97, dengan aspek lainnya seperti aksesibilitas, best practice, dan SEO masing-masing mendapatkan skor 84, 100, dan 82. Hasil ini menunjukkan bahwa aplikasi telah dioptimalkan dengan baik.
4. Metode *Personal Extreme Programming (PXP)* terbukti mendukung pengembangan aplikasi secara adaptif dengan memberikan fleksibilitas untuk menambahkan iterasi guna memenuhi standar *usability* yang diidentifikasi melalui evaluasi *heuristik*. Hal ini dibuktikan pada iterasi kelima, di mana dilakukan evaluasi *heuristik* untuk menilai tingkat *usability* aplikasi *split bill* yang dikembangkan. Evaluasi tersebut menghasilkan sejumlah saran perbaikan dari para evaluator, yang kemudian diimplementasikan pada iterasi keenam dengan fokus pada peningkatan *usability*.



## 5.2 Saran

Saran yang dapat dipertimbangkan untuk penelitian selanjutnya antara lain :

1. Menambahkan fitur OCR (*Optical Character Recognition*) yang memungkinkan pengguna untuk memindai tagihan fisik yang mereka miliki, sehingga data seperti total tagihan dan item pada tagihan dapat diisi secara otomatis ke dalam aplikasi.
2. Mengembangkan aplikasi menjadi *Progressive Web App* (PWA). Dengan fitur ini, aplikasi dapat diakses secara *offline*, memberikan pengalaman seperti aplikasi *native*, dan meningkatkan performa pada perangkat dengan koneksi internet yang tidak stabil.
3. Mengimplementasikan *automated end-to-end testing* akan mempercepat proses pengujian, mengingat pengujian menyeluruh secara manual setiap kali fitur baru ditambahkan memakan waktu yang lama.
4. tools

## DAFTAR PUSTAKA

- [1] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming—An Agile Process for Autonomous Developers," in *S3T 2009 (Symposium on Emerging Trends in Software Engineering)*, Sofia, Bulgaria, 2009, pp. 44–45.
- [2] CIMB, "Apa Itu Split Bill? Ini Asal Mula, Keuntungan, dan Caranya," [cimbniaga.co.id](https://www.cimbniaga.co.id/id/inspirasi/gayahidup/split-bill). Accessed: Jul. 29, 2024. [Online]. Available: <https://www.cimbniaga.co.id/id/inspirasi/gayahidup/split-bill>
- [3] GoPay, "Split Bill Makin Gampang di Aplikasi GoPay." Accessed: Dec. 05, 2024. [Online]. Available: <https://gopay.co.id/blog/split-bill-aplikasi-gopay#cara-menggunakan>
- [4] M. T. Prihandoyo, "Unified Modeling Language (UML) model untuk pengembangan sistem informasi akademik berbasis web," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 1, pp. 126–129, 2018.
- [5] B. Tarangga, I. P. Putra, and R. N. Zhafran, "Perancangan Sistem Informasi Rental Mobil Menggunakan UML," in *Prosiding Seminar SITASI 13 November 2021*, Nov. 2021, pp. 183–190.
- [6] M. Romzi and B. Kurniawan, "Implementasi Pemrograman Python Menggunakan Visual Studio Code," *JIK*, vol. XI, no. 2, pp. 1–9, Dec. 2020.
- [7] P. S. Maratkar and P. Adkar, "React JS-An Emerging Frontend Javascript Library," *IRE Journals*, vol. 4, no. 12, pp. 99–102, Jun. 2021.
- [8] A. Sandeep, "React.js and Front-End Development," *International Research Journal of Engineering and Technology*, vol. 7, no. 4, pp. 3676–3679, Apr. 2020.
- [9] R. Leonardo, I. Arwani, and D. E. Ratnawati, "Pemanfaatan Teknologi Firebase dalam Pengembangan Aplikasi Pengelolaan Stok Barang Berbasis Mobile pada Rumah Makan Nakamse Malang," *Teknologi Informasi, dan Edukasi Sistem Informasi*, vol. 1, no. 1, pp. 1–11, Jul. 2020.
- [10] I. F. Maulana, "Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 854–863, 2020.

- [11] Lighthouse, “Lighthouse,” <https://developer.chrome.com/docs/lighthouse>. Accessed: Nov. 17, 2024. [Online]. Available: <https://developer.chrome.com/docs/lighthouse>
- [12] C. Kartiko, “Black Box Testing Boundary Value Analysis pada Aplikasi Submission System,” *Edik Informatika*, vol. 6, no. 2, pp. 15–22, Apr. 2020, doi: 10.22202/ei.2020.v6i2.3995.
- [13] M. E. Khan, “Different approaches to white box testing technique for finding errors,” *International Journal of Software Engineering and its Applications*, vol. 5, no. 3, pp. 1–14, 2011, doi: 10.5121/ijsea.2011.2404.
- [14] J. Nielsen, “Finding Usability Problems Through Heuristic Evaluation,” in *CHI '92*, Mar. 1992.
- [15] A. Abulfaraj and A. Steele, “Detailed Usability Heuristics: A Breakdown of Usability Heuristics to Enhance Comprehension for Novice Evaluators,” in *HCI International 2020 - Late Breaking Papers: User Experience Design and Case Studies*, C. Stephanidis, A. Marcus, E. Rosenzweig, P.-L. P. Rau, A. Moallem, and M. Rauterberg, Eds., Cham: Springer International Publishing, 2020, pp. 3–18.
- [16] J. Nielsen and R. L. Mack, *Usability Inspection Methods*. Wiley, 1994. [Online]. Available: <https://books.google.co.id/books?id=cuRQAAAAMAAJ>
- [17] J. Nielsen, “Enhancing the Explanatory Power of Usability Heuristics,” in *CHI '94*, Apr. 1994, pp. 152–158.
- [18] F. Rahman Suprpto, G. Indah Marthasari, and I. Nuryasin, “Sistem Informasi Penjualan dan Pelelangan Berbasis Web pada Ricardo Corner MLG Menggunakan Metode Personal eXtreme Programming (PXP),” *REPOSITOR*, vol. 2, no. 11, pp. 1535–1542, Nov. 2020.
- [19] T. Sandika and H. Kurniawan, “Rancang Bangun Sistem Informasi Kualitas Air Wilayah Sungai di Provinsi Lampung dengan Metode Personal Extreme Programming,” *Jurnal Ilmiah ESAI*, vol. 8, no. 2, 2014.
- [20] S. A. Asri and W. Setiawan, “Alternatif penggunaan model pendekatan agile pada perancangan sistem informasi PKL online,” *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 5, no. 3, p. 122, 2017.
- [21] J. Panjaitan and A. F. Pakpahan, “Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr. 2021, doi: 10.28932/jutisi.v7i1.3098.
- [22] A. Muda, S. Huda, and Y. Fernando, “E-Ticketing Penjualan Tiket Event Musik di Wilayah Lampung pada Karcismu Menggunakan Library ReactJS,” *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol. 2, no. 1, pp. 96–103, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTSI>

- [23] F. F. Nursaid, A. Hendra Brata, and A. P. Kharisma, “Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri),” vol. 4, no. 1, pp. 46–55, Jan. 2020.
- [24] R. N. Sari, N. Novrina, and A. R. Kusuma, “Pemanfaatan React Native dan Firebase untuk Pembuatan Aplikasi Jual-Beli Item Game Online,” *Jurnal Larik: Ladang Artikel Ilmu Komputer*, vol. 2, no. 1, pp. 17–22, 2022.
- [25] M. T. Prihandoyo, “Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web,” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 1, pp. 126–129, 2018.
- [26] R. Sastra, “Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Penggajian,” *Jurnal Teknik Komputer AMIK BSI*, vol. 7, no. 1, 2021, doi: 10.31294/jtk.v4i2.
- [27] M. Sumiati, R. Abdillah, and A. Cahyo, “Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta,” *Jurnal FASILKOM*, vol. 11, no. 2, pp. 79–86, Aug. 2021.